

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Beberapa hasil penelitian terdahulu yang relevan dapat dilihat pada tabel II.1.

Tabel II.1. Penelitian Terdahulu

NO	Nama Penulis	Judul Penelitian	Hasil Penelitian
1.	Sandy Purnama, Dyah Ayu Megawaty, Yusra Fernando (2018)	PENERAPAN ALGORITMA A STAR (A*) UNTUK PENENTUAN JARAK TERDEKAT WISATA KULINER DI KOTA BANDAR LAMPUNG	Hasil dari penelitian ini adalah membuat sebuah aplikasi Mobile <i>Android</i> untuk memberikan informasi mengenai jalur terdekat menuju wisata kuliner dan juga memberikan informasi mengenai kuliner yang terdapat di sekitar pengguna.

2	<p>Ida Bagus Gede Wahyu Antara Dalem (2018)</p>	<p>PENERAPAN ALGORITMA A* (STAR) MENGUNAKAN GRAPH UNTUK MENGHITUNG JARAK TERPENDEK</p>	<p>Hasil dari penelitian ini yaitu mempelajari cara kerja algoritma A* dalam mencari jarak tercepat, yang disimulasikan seperti kondisi ketika seorang mencari rute dalam keadaan jalanan macet. Simulasi ini memberikan gambaran yang lebih realistis terhadap perilaku algoritma A* dalam pencarian jarak rute terpendek.</p>
3	<p>Mulyono, Rochmad, Rosita Ayu Nugraeni (2015)</p>	<p>PENERAPAN ALGORTIMA A* DALAM PENYELESAIAN RUTE TERPENDEK PENDISTRIBUSIAN BARANG</p>	<p>Tujuan penelitian ini adalah untuk mengetahui dasar-dasar algoritma A*, untuk meneliti penentuan rute terpendek pendistribusian barang dengan algoritma A*, dan untuk meneliti penentuan rute terpendek pendistribusian barang diaplikasikan dengan berbantuan software pada CV Mitra Adi Busana Semarang.</p>
4	<p>Pratama Juliantono Taufiq, Agung</p>	<p>IMPLEMENTASI DAN ANALISIS ALGORITMA</p>	<p>Hasil dari penelitian ini yaitu pembuatan NPC yang pintar dengan algoritma yang bisa</p>

	Toto Wibowo, ST., MT., Gia Septiana, SSi., MSc (2015)	A*(STAR) UNTUK MENENTUKAN JALUR DENGAN MULTIPLE GOAL PADA PERGERAKAN NPC (NON-PLAYABLE CHARACTER)	menentukan rute yang optimal yang bisa diimplementasikan untuk kasus lebih dari satu karakter. Pada umumnya algoritma A*(A Star) sering digunakan pada game untuk kasus pencarian jalur. Penerapan algoritma A*(Star) dengan menggunakan nilai heuristik yang didapat dari mengkombinasikan jarak garis lurus antar masing-masing tujuan mampu menyelesaikan kasus multiple-goal dengan hasil yang complete dan optimal.
5	Muh. Yamin, Moh. Bandrigo Talai (2015)	APLIKASI PENCARIAN JALUR TERPENDEK PADA RUMAH SAKIT UMUM BAHTERAMAS MENGGUNAKAN ALGORITMA A* (A-STAR)	Hasil dari penelitian ini adalah aplikasi yang dapat menentukan jalur terpendek antara gedung dan antara ruangan yang diimplementasikan pada Operating System <i>Android</i> dan dibangun dengan menggunakan bahasa pemrograman Actionscript 3.

II.2. Landasan Teori

II.2.1. Pengertian Aplikasi

Aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi user dengan menggunakan kemampuan computer (Fergiawan Listianto, dkk, 2017).

Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain lain tanpa terjadi pemutusan atau terputusnya komunikasi (Fitriani Sahara dan Rena Adriana, 2016).

Berdasarkan pengertian diatas dapat disimpulkan bahwa aplikasi adalah suatu perangkat lunak yang bekerja pada suatu sistem seperti komputer dan telepon.

II.2.2. Pengertian Sistem

Sistem pada dasarnya adalah sekelompok unsur yang erat berhubungan satu dengan lainnya, yang berfungsi bersamasama untuk mencapai tujuan tertentu (Mulyadi, 2016: 1).

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar (Romney dan Steinbart, 2015 : 3).

Berdasarkan pengertian di atas dapat disimpulkan bahwa sistem adalah kumpulan dari unsur ataupun komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan.

II.2.3. Pengertian Rute

Rute merupakan ruas – ruas jalan yang dilalui dalam suatu trayek sehingga satu trayek dapat memiliki lebih dari satu rute. Rute angkutan umum biasanya ditempatkan di lokasi yang memang diperkirakan ada calon penumpang yang akan dilayani.

II.2.4. Pengertian Rumah Sakit

Pengertian rumah sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jalan dan gawat darurat (Kemenkes RI, 2015).

Berdasarkan pengertian di atas dapat disimpulkan bahwa rumah sakit adalah sebuah institusi penyedia pelayanan perawatan kesehatan.



Gambar II.1. Rumah Sakit
(Sumber : Rumah.com)

II.2.5. Algoritma Pencarian

Algoritma Pencarian adalah algoritma yang menghasilkan sebuah solusi untuk masalah yang sedang dikaji, biasanya didapat dari evaluasi beberapa kemungkinan solusi.

Dalam ilmu komputer, sebuah algoritma pencarian dijelaskan secara luas adalah sebuah algoritme yang menerima masukan berupa sebuah masalah dan menghasilkan sebuah solusi untuk masalah tersebut, yang biasanya didapat dari evaluasi beberapa kemungkinan solusi. Sebagian besar algoritma yang dipelajari oleh ilmuwan komputer adalah algoritma pencarian. Himpunan semua kemungkinan solusi dari sebuah masalah disebut ruang pencarian. Algoritma pencarian *brute-force* atau pencarian naif menggunakan metode yang sederhana dan sangat intuitif pada ruang pencarian, sedangkan algoritma

pencarian *informed* menggunakan heuristik untuk menerapkan pengetahuan tentang struktur dari ruang pencarian untuk berusaha mengurangi banyaknya waktu yang dipakai dalam pencarian.

II.2.6. Algoritma A-Star

Algoritma A-Star merupakan salah satu algoritma yang termasuk dalam kategori metode pencarian yang memiliki informasi (*informed search method*). Algoritma ini sangat baik sebagai solusi proses *pathfinding* (pencari jalan). Algoritma ini mencari jarak rute terpendek yang akan ditempuh suatu point awal (*starting point*) sampai ke objek tujuan. Teknik pencarian yang digunakan dalam simulasi ini adalah menggunakan Algoritma A-Star dengan fungsi heuristic. Tujuan utama penelitian ini mempelajari cara kerja algoritma A-Star dalam mencari jarak tercepat, yang disimulasikan seperti kondisi ketika seorang mencari rute dalam keadaan jalanan macet. Simulasi ini memberikan gambaran yang lebih realistis terhadap perilaku algoritma A-Star dalam pencarian jarak rute terpendek (Ida Bagus Gede Wahyu Antara Dalem, 2018 : 1).

Contoh penerapan algoritma A-Star untuk menyelesaikan masalah sehari-hari seperti pencarian tempat tinggal sementara seperti indkos. Algoritma ini meminimalkan total biaya lintasan, dan pada kondisi yang tepat akan memberikan solusi yang terbaik dalam waktu yang optimal. Algoritma A-Star membutuhkan dua antrian, yaitu *OPEN* dan *CLOSED*. *OPEN* adalah senarai (*list*) yang digunakan untuk menyimpan simpul-simpul yang pernah dibangkitkan dan nilai heuristiknya telah dihitung tetapi belum dipilih sebagai

simpul terbaik (*best node*). Dengan kata lain, *OPEN* berisi simpul-simpul yang masih memiliki peluang untuk terpilih sebagai simpul terbaik, sedangkan *CLOSED* adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah dipilih sebagai simpul terbaik. Artinya, *CLOSED* berisi simpul-simpul yang tidak mungkin dipilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup). Selain antrean tersebut, ada juga fungsi heuristik yang memprediksi keuntungan setiap *node* yang dibuat.

Algoritma ini memeriksa *node* dengan menggabungkan $g(n)$, yaitu *cost* yang dibutuhkan untuk mencapai sebuah *node* dan $h(n)$, yaitu *cost* yang didapat dari *node* awal ke *node* n . Sehingga didapatkan rumus dasar dari algoritma A-Star ini adalah:

$$f(n) = g(n) + h(n)$$

dimana:

$h(n)$ = nilai heuristik antar koordinat

$g(n)$ = jarak koordinat ke titik tujuan

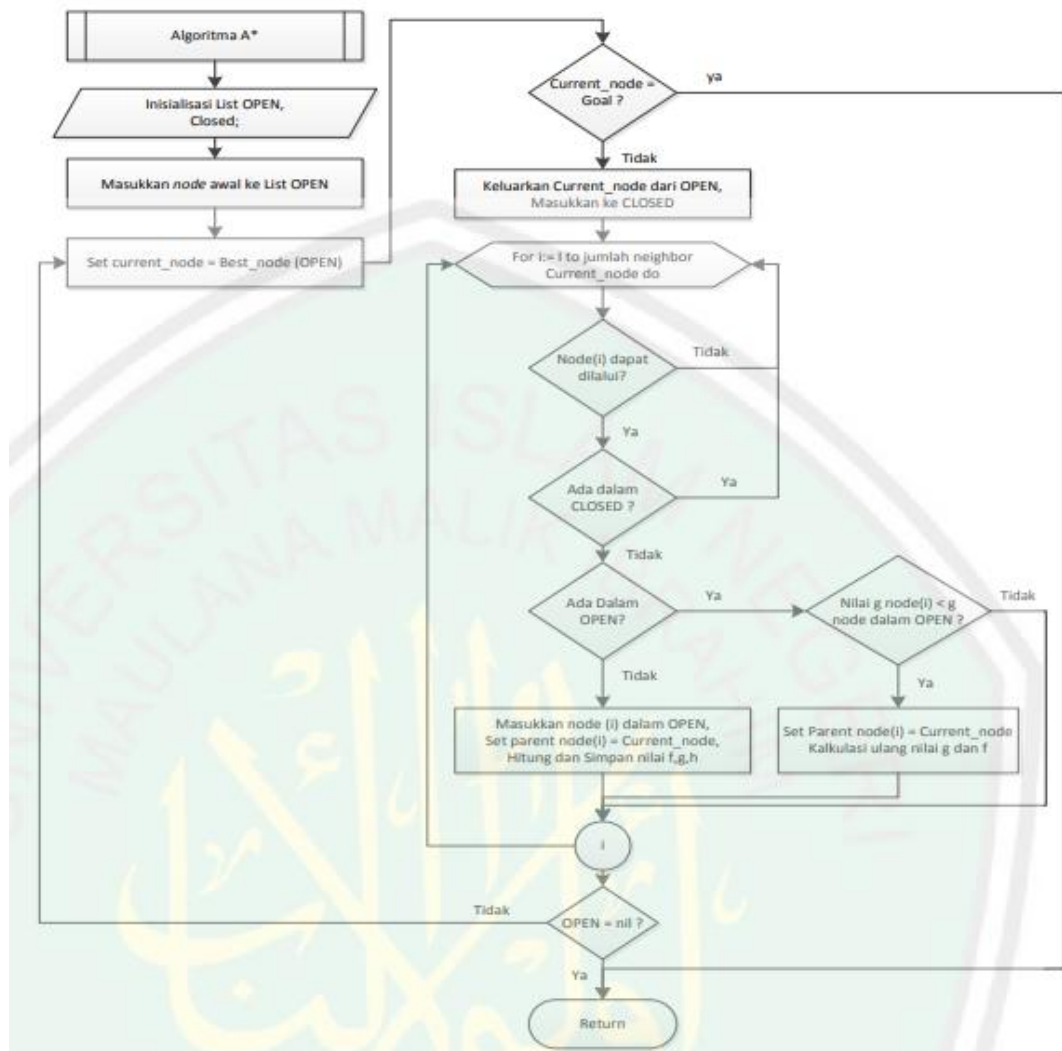
Algoritma A-Star memiliki 2 fungsi utama dalam menentukan solusi terbaik. Fungsi pertama disebut sebagai $g(x)$ merupakan fungsi yang digunakan untuk menghitung total *cost* yang dibutuhkan dari *node* awal menuju *node* tertentu. Fungsi kedua yang biasa disebut sebagai $h(x)$ merupakan fungsi perkiraan total *cost* yang diperkirakan dari suatu *node* ke *node* akhir.

$$\sqrt{Xn^2 + Yn^2}$$

$G(n) =$

$H(n) = [X(\text{target}) - X(n) [+] Y(\text{target}) - Y(n)]$

$F(n) = G(n) + f(n)$



Gambar II.2. Algoritma A-Star
(Sumber :Naufal Wafiqurrahman, 2015)

II.2.7. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi sendiri (Rusdi Efendi, dkk, 2015).

Android awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel *Android* pertama mulai dijual pada bulan Oktober 2008. *Android, Inc.* didirikan di Palo Alto, California, pada bulan Oktober 2003 oleh Andy Rubin (pendiri Danger), Rich Miner (pendiri Wildfire Communications, Inc.), Nick Sears (mantan VP T-Mobile), dan Chris White (kepala desain dan pengembangan antarmuka WebTV) untuk mengembangkan "perangkat seluler pintar yang lebih sadar akan lokasi dan preferensi penggunanya". Tujuan awal pengembangan *Android* adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan *Android* lalu dialihkan bagi pasar telepon pintar untuk menyaingi *Symbian* dan *Windows Mobile* (iPhone Apple belum dirilis pada saat itu). Meskipun para pengembang *Android* adalah pakar-pakar teknologi yang berpengalaman, *Android Inc.* dioperasikan secara diam-diam, hanya diungkapkan bahwa para pengembang sedang menciptakan sebuah perangkat lunak yang diperuntukkan bagi telepon seluler. Masih pada tahun yang sama, Rubin kehabisan uang. Steve Perlman, seorang teman dekat Rubin, meminjaminya \$10.000 tunai dan menolak tawaran saham di perusahaan. Google mengakuisisi *Android Inc.* pada tanggal 17 Agustus 2005, menjadikannya sebagai anak

perusahaan yang sepenuhnya dimiliki oleh Google. 5 6 Pendiri *Android Inc.* seperti Rubin, Miner dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google. Setelah itu, tidak banyak yang diketahui tentang perkembangan *Android Inc.*, namun banyak anggapan yang menyatakan bahwa Google berencana untuk memasuki pasar telepon seluler dengan tindakannya ini. Di Google, tim yang dipimpin oleh Rubin mulai mengembangkan platform perangkat seluler dengan menggunakan kernel Linux. Google memasarkan platform tersebut kepada produsen perangkat seluler dan operator nirkabel, dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bisa diperbarui. Google telah memilih beberapa mitra perusahaan perangkat lunak dan perangkat keras, serta mengisyaratkan kepada operator seluler bahwa kerjasama ini terbuka bagi siapapun yang ingin berpartisipasi. Spekulasi tentang niat Google untuk memasuki pasar komunikasi seluler terus berkembang hingga bulan Desember 2006. BBC dan *Wall Street Journal* melaporkan bahwa Google sedang bekerja keras untuk menyertakan aplikasi dan mesin pencariinya di perangkat seluler. Berbagai media cetak dan media daring mengabarkan bahwa Google sedang mengembangkan perangkat seluler dengan merek Google. Beberapa di antaranya berspekulasi bahwa Google telah menentukan spesifikasi teknisnya, termasuk produsen telepon seluler dan operator jaringan. Pada bulan Desember 2007, *Information Week* melaporkan bahwa Google telah mengajukan beberapa aplikasi paten di bidang telepon seluler. Pada tanggal 5 November 2007, *Open Handset Alliance* (OHA) didirikan. OHA adalah konsorsium dari perusahaan-perusahaan teknologi seperti Google, produsen perangkat seluler

seperti HTC, Sony dan Samsung, operator nirkabel seperti *Sprint Nextel* dan *T-Mobile*, serta produsen chipset seperti *Qualcomm* dan Texas Instruments. OHA sendiri bertujuan untuk mengembangkan standar terbuka bagi perangkat seluler. Saat itu, *Android* diresmikan sebagai produk pertamanya; sebuah platform perangkat seluler yang menggunakan kernel Linux versi 2.6. Telepon seluler komersial pertama yang menggunakan sistem operasi *Android* adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008. Pada tahun 2010, Google merilis seri Nexus; perangkat telepon pintar dan tablet dengan sistem operasi *Android* yang diproduksi oleh mitra produsen telepon seluler seperti HTC, LG, dan Samsung. HTC bekerjasama dengan Google dalam merilis produk telepon pintar Nexus pertama, yakni Nexus One. Seri ini telah diperbarui dengan perangkat yang lebih baru, misalnya telepon pintar Nexus 4 dan tablet Nexus 10 yang diproduksi oleh LG dan Samsung. Pada 15 Oktober 2014, Google mengumumkan Nexus 6 dan Nexus 9 yang diproduksi oleh Motorola dan HTC. Pada 13 Maret 2013, Larry Page mengumumkan dalam postingan blognya bahwa Andy Rubin telah pindah dari divisi *Android* untuk mengerjakan proyekproyek baru di Google. Ia digantikan oleh Sundar Pichai, yang sebelumnya menjabat sebagai kepala divisi Google Chrome, yang mengembangkan Chrome OS. Sejak tahun 2008, *Android* secara bertahap telah melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi, menambahkan fitur baru, dan memperbaiki bug yang terdapat pada versi sebelumnya. Setiap versi utama yang dirilis dinamakan secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula; misalnya, versi 1.5 bernama Cupcake, yang kemudian diikuti

oleh versi 1.6 Donut. Versi terbaru adalah 5.0 Lollipop, yang dirilis pada 15 Oktober 2014.

Sejak dirilisnya *Android* pada tahun 2008, *Android* kerap mengeluarkan versi terbarunya. Berikut adalah perkembangan versi *Android* hingga saat ini :

1. *Android* 1.0 (API level 1)
2. *Android* 1.1 (API level 2)
3. *Android* 1.5 Cupcake (API level 3)
4. *Android* 1.6 Donut (API level 4)
5. *Android* 2.0 Eclair (API level 5)
6. *Android* 2.2–2.2.3 Froyo (API level 8)
7. *Android* 2.3–2.3.2 Gingerbread (API level 9)
8. *Android* 2.3.3–2.3.7 Gingerbread (API level 10)
9. *Android* 3.0 Honeycomb (API level 11)
10. *Android* 3.2 Honeycomb (API level 13)
11. *Android* 4.0–4.0.2 Ice Cream Sandwich (API level 14)
12. *Android* 4.0.3–4.0.4 Ice Cream Sandwich (API level 15)
13. *Android* 4.1 Jelly Bean (API level 16)
14. *Android* 4.2 Jelly Bean (API level 17)
15. *Android* 4.3 Jelly Bean (API level 18)
16. *Android* 4.4 KitKat (API level 19)
17. *Android* 5.0 Lollipop (API level 21)
18. *Android* 6.0 Marshmallow (API level 23)
19. *Android* 7.0 Nougat (API 24)

20. *Android 7.1 Nougat* (API 25) 21. *Android 8.0 "OREO"* (API 26)

21. *Android 9.0 Pie* (API 27)



Gambar II.3. Logo *Android*
(Sumber : id.wikipedia.org)

II.2.8. *Android Studio*

Android Studio merupakan sebuah Integrated Development Environment (IDE) khusus untuk membangun aplikasi yang berjalan pada platform *android*. *Android studio* ini berbasis pada IntelliJ IDEA, sebuah IDE untuk bahasa pemrograman Java. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau layout, digunakan bahasa XML. *Android studio* juga terintegrasi dengan *Android Software Development Kit* (SDK) untuk deploy ke perangkat *android*. *Android Studio* juga merupakan pengembangan dari eclipse, dikembangkan menjadi lebih kompleks dan profesional yang telah tersedia didalamnya *Android Studio IDE*, *Android SDK tools*.

II.2.9. MySQL

MySQL adalah sistem manajemen *Database* SQL yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database* MySQL mendukung beberapa fitur seperti *multithreaded*, *multiuser* dan *SQL Database managemen system* (DBMS) (MADCOMS, 2016).

Ketika dibandingkan antara MySQL dengan sistem database yang lain, maka perlu difikirkan apa yang paling penting sesuai kebutuhan. Apakah tampilan, support, fitur-fitur SQL, kondisi keamanan dalam lisensi, atau masalah harga. Dengan pertimbangan tersebut, MySQL memiliki banyak hal yang bisa ditawarkan, antara lain :

1. Berdasarkan kecepatannya, banyak ahli memberikan pendapat bahwa MySQL merupakan server tercepat.
2. MySQL memiliki performa tinggi namun merupakan database yang simpel sehingga mudah di-*setup* dan dikonfigurasi.
3. MySQL cenderung gratis untuk penggunaan tertentu.
4. MySQL mengerti bahasa SQL (*Structured Query Language*) yang merupakan pilihan sistem database moderen.
5. Banyak klien dapat mengakses server dalam satu waktu. Mereka dapat menggunakan banyak database secara simultan.
6. Database MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu.
7. MySQL dapat berjalan dalam banyak varian Unix dengan baik, sebaik seperti saat berjalan di sistem non-Unix.

8. MySQL mudah didapatkan dan memiliki *source code* yang boleh disebarluaskan sehingga bisa dikembangkan lebih lanjut.
9. Dapat dikoneksikan pada bahasa C, C++, Java, Perl, PHP dan Python.

II.2.10. UML

Unified Modeling Language (UML) 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 merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada 26 kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahudin, 2015:133).


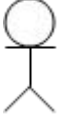


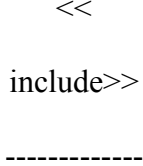
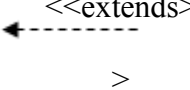
II.2.10.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (behavior) 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. (Ade Hendini, 2016).

Simbol-simbol *Use Case Diagram* dapat dilihat pada tabel II.2

Tabel II.2. Simbol-simbol *Use Case Diagram*



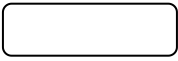
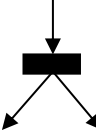
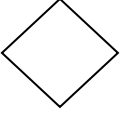
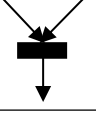
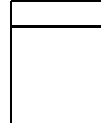
Simbol	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada kontekstarget sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap <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.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
 << include>>	<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
 <<extends>> >	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat.

(Sumber : Ade Hendini, 2016)

II.2.10.2. Activity Diagram

Activity Diagram menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem (Yunahar Heriyanto, 2018). Simbol-simbol *Activity Diagram* dapat dilihat pada tabel II.3.

Tabel II.3. Simbol-simbol *ActivityDiagram*





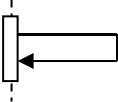
Simbol	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa


(Sumber : Jurnal Urva dan Fauzi, 2015)

II.2.10.3. *Sequence Diagram*

Sequence Diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram* (Suendri, 2018). Simbol-simbol *Sequence Diagram* dapat dilihat pada tabel II.4.

Tabel II.4. Simbol-simbol *Sequence Diagram*

Simbol	Keterangan
	<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>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<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

	<p><i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi</p>
---	--