

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait ini, peneliti akan membandingkan hasil penelitiannya dengan lima jurnal yaitu :

Berdasarkan Penelitian dari Saifur Rohman Cholil (2020) Untuk meminimalisir terjadinya kesalahan serta lama waktu yang digunakan, dibuatlah suatu Sistem Pendukung Keputusan untuk menentukan calon karyawan terbaik yang akan bekerja diperusahaan. Metode Sistem Pendukung Keputusan yang digunakan adalah Additive Ratio Assessment (ARAS). Penelitian ini telah melalui proses validasi korelasi rank spearman dan diperoleh nilaisebesar 0,950. Berdasarkan hasil tersebut, metode ARAS dapat digunakan dalam menyeleksi calon karyawan baru pada PT. Dawam prima Perkasa.

Penelitian Ketiga Liza Handayani, M. Syahrizal, Kennedi Tampubolon (2019) Hasil yang dicapai adalah dalam pemilihan kepling teladan menggunakan sistem pendukung keputusan ini dapat menentukan kriteria dan nilai bobot untuk setiap alternatif dengan menggunakan metode *rank order centroid (ROC)* dapat menentukan nilai bobot dengan tingkat prioritasnya, dan dengan menggunakan metode *additive ratio assessement (ARAS)* dinilai dapat menyelesaikan permasalahan dalam pemilihan kepling teladan di Kecamatan Medan Area.

Berdasarkan penelitian dari Heri Syahputra (2019) dalam suatu sistem keputusan. Dengan tujuan untuk membantu orang tua dalam melakukan

pengawasan untuk anak-anaknya diwebsite youtube tersebut. Sistem Pendukung Keputusan (SPK) adalah sistem yang dapat membantu seseorang, dalam mengambil suatu keputusan yang akurat dan tepat sasaran. Banyak permasalahan yang dapat diselesaikan dengan menggunakan SPK contohnya banyak orang tua yang bingung dalam memilih konten yang layak tonton untuk anak mereka, disini fungsi sistem ini dibuat.

Berdasarkan Charis Maulana, dkk (2019) bahwa penelitian dengan menerapkan metode Additive Ratio Assessment dalam suatu sistem pendukung keputusan, sehingga dapat membantu dalam menentukan penerima pinjaman koperasi untuk menghindari kredit macet. Sistem Pendukung Keputusan (SPK) adalah sistem yang dapat membantu seseorang, dalam mengambil suatu keputusan yang akurat dan tepat sasaran. Banyak permasalahan yang dapat diselesaikan dengan menggunakan SPK, contohnya membangun model sistem pendukung keputusan penentuan anggota koperasi potensial dalam pengajuan pinjaman untuk menghasilkan informasi anggota koperasi potensial untuk menghindari kredit macet.

Penelitian pertama Fadila Pratiwi, Fince Tinus Waruwu, Dito Putro Utomo, Rian Syahputra (2019) penelitian ini berisikan sistem pengambilan keputusan dengan *Aras*. Hasil yang dicapai adalah dilakukan dengan cara menghitung nilai-nilai alternatif berdasarkan algoritma *aras* yang hasilnya bertujuan untuk mendapatkan alternatif dengan ranking tertinggi.

Berdasarkan penelitian dari Hendri Susanto (2018) bahwa penelitian dengan menerapkan metode Additive Ratio Assessment dalam suatu sistem pendukung keputusan, sehingga dapat membantu dalam pemilihan Susu Gym terbaik untuk menambah massa Otot. Sistem Pendukung Keputusan (SPK) merupakan sistem yang dapat membantu seseorang, dalam mengambil suatu keputusan yang akurat dan tepat sasaran. Banyak permasalahan yang dapat diselesaikan dengan menggunakan SPK, contohnya saat para pemula fitness dalam memutuskan untuk membeli Susu Gym sebagai tambahan protein untuk membantu dalam pemulihan tubuh ataupun penambah massa otot.

Penelitian kedua Lia Ciky Lumban Gaol, Nelly Astuti Hasibuan (2018) penelitian ini membahas tentang sistem pengambilan keputusan dengan metode *Aras*. Hasil yang dicapai adalah untuk memecahkan masalah pemilihan team leader shift terbaik dengan menggunakan metode tersebut didapatkan bahwa kriteria yang paling dominan adalah kriteria Kejujuran dibandingkan dengan keempat kriteria lainnya yaitu: Pengalaman, Pelatihan, Penampilan dan Wawasan.

II.2. Landasan Teori

II.2.1 Sistem

Definisi sistem adalah “kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama.” Definisi sistem adalah “sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai suatu kesatuan yang dirancang untuk mencapai suatu tujuan. Sistem adalah penggabungan dari bagian-bagian atau komponen-komponen yang terpisah-

pisah dan disatukan menjadi satu rangkaian dan menjadi suatu fungsi yang baru (Aris, 2017)

Kata sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) yang artinya adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Secara umum sistem adalah kumpulan dari beberapa bagian tertentu yang saling berhubungan secara harmonis untuk mencapai suatu tujuan tertentu. Elemen-elemen yang mewakili suatu sistem secara umum adalah masukan (*Input*) pengolahan (*Processing*) dan keluaran (*Output*). (Sutopo et al., 2016)

II.2.1.1. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya yaitu:

1. Batasan (*Boundary*) adalah Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*Environment*) adalah Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*Input*) adalah Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.

4. Keluaran (*Output*) adalah Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar Komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*Component*) adalah Kegiatan-kegiatan atau proses dalam sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*Interface*) adalah Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*Storage*) adalah Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama (Aris, 2017)

II.2.2 Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem informasi berbasis komputer yang interaktif, dengan cara mengolah data dengan berbagai model untuk memecahkan masalah-masalah yang tidak terstruktur sehingga dapat memberikan informasi yang bisa digunakan oleh para pengambil keputusan dalam membuat sebuah keputusan. Dalam sebuah sistem pendukung keputusan, sumber daya intelektual yang dimiliki seseorang dipadukan dengan kemampuan komputer untuk membantu meningkatkan kualitas dari keputusan yang diambil. Pengambilan

keputusan merupakan sebuah proses memilih sebuah tindakan diantara beberapa alternatif yang ada, sehingga tujuan yang diinginkan dapat tercapai (Alfina Harahap : 2019)

Sistem Pendukung Keputusan adalah konsep spesifik sistem yang menghubungkan komputerisasi informasi dengan para pengambil keputusan sebagai pemakainya. Dengan demikian Sistem Pendukung Keputusan adalah seperangkat elemen yang digabungkan satu dengan yang lainnya saling bekerja sama untuk menghasilkan satu kesatuan di dalam pencapaian suatu tujuan bersama. Sistem pendukung keputusan adalah suatu sistem berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data dan model. Untuk menghasilkan keputusan yang baik di dalam sistem pendukung keputusan, perlu didukung oleh informasi dan fakta-fakta yang berkualitas antara lain :

1. Akseibilitas

Atribut ini berkaitan dengan kemudahan mendapatkan informasi, informasi akan lebih berarti bagi si pemakai kalau informasi tersebut mudah didapat, karena akan berkaitan dengan aktifitas dari nilai informasinya.

2. Kelengkapan

Atribut ini berkaitan dengan kelengkapan isi informasi, dalam hal ini isi tidak menyangkut hanya volume tetapi juga kesesuaian dengan harapan si pemakai sehingga sering kali kelengkapan ini sulit diukur secara kuantitatif

3. Ketelitian

Atribut ini berkaitan dengan tingkat kesalahan yang mungkin di dalam pelaksanaan pengolahan data dalam jumlah (volume) besar. Dua tipe kesalahan yang sering terjadi yaitu berkaitan dengan perhitungan.

4. Ketepatan

Atribut ini berkaitan dengan kesesuaian antara informasi yang dihasilkan dengan kebutuhan pemakai. Sama halnya dengan kelengkapan, ketepatan pun sangat sulit diukur secara kuantitatif.

5. Ketepatan Waktu

Kualitas informasi juga sangat ditentukan oleh ketepatan waktu penyampaian dan aktualisasinya. Misal informasi yang berkaitan dengan perencanaan harian akan sangat berguna kalau disampaikan setiap dua hari sekali.

6. Kejelasan

Atribut ini berkaitan dengan bentuk atau format penyampaian informasi. Bagi seorang pimpinan, informasi yang disajikan dalam bentuk grafik, histogram, atau gambar biasanya akan lebih berarti dibandingkan dengan informasi dalam bentuk katakata yang panjang.

7. Fleksibilitas

Atribut ini berkaitan dengan tingkat adaptasi dari informasi yang dihasilkan terhadap kebutuhan berbagai keputusan yang akan diambil dan terhadap sekelompok pengambil keputusan yang berbeda. (Arief Budiman, dkk : 2020)

II.2.3. Tujuan Sistem Pendukung Keputusan

Sistem Pendukung Keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. Tujuan dari sistem pendukung keputusan adalah:

1. Membantu manajer dalam pengambilan keputusan atas masalah semiterstruktur
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer
3. Meningkatkan efektifitas keputusan yang diambil manajer lebih daripada perbaikan efesiensinya
4. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah
5. Peningkatan produktifitas
6. Komputer bisa meningkatkan kualitas keputusan yang dibuat.
7. Berdaya saing
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan (Arief Budiman, dkk : 2020).

II.2.4. Siswa

Siswa/Siswi istilah bagi peserta didik pada jenjang pendidikan dasar dan menengah. Siswa adalah komponen masukan dalam sistem pendidikan, yang

selanjutnya diproses dalam proses pendidikan, sehingga menjadi manusia yang berkualitas sesuai dengan tujuan pendidikan nasional. Sebagai suatu komponen pendidikan, siswa dapat ditinjau dari berbagai pendekatan, antara lain: pendekatan social, pendekatan psikologis, dan pendekatan edukatif/pedagogis dengan tujuan untuk menjadi manusia yang berilmu pengetahuan, berketerampilan, berpengalaman, berkepribadian, berakhlak mulia, dan mandiri. (Dn, 2017)

II.2.5. Prestasi

Prestasi adalah suatu kegiatan yang telah dikerjakan, diciptakan baik secara individual maupun kelompok. Istilah prestasi di dalam Kamus Ilmiah Populer didefinisikan sebagai hasil yang telah dicapai. Menurut Noehi Nasution, menyimpulkan bahwa “belajar dalam arti luas dapat diartikan sebagai suatu proses yang memungkinkan timbulnya atau berubahnya suatu tingkah laku sebagai hasil dari terbentuknya respons utama, dengan syarat bahwa perubahan atau munculnya tingkah baru itu bukan disebabkan oleh adanya perubahan sementara karena sesuatu hal. (Dn, 2017)

II.2.6. Beasiswa

Beasiswa pada umumnya merupakan pemberian biaya untuk pendidikan bagi mahasiswa yang masih aktif mengikuti masa pendidikan. Dalam Kamus Besar Bahasa Indonesia, beasiswa dimaknai sebagai tunjangan yang diberikan kepada pelajar atau mahasiswa sebagai bantuan biaya belajar. Beasiswa juga bisa dimaknai lain yakni sebagai dana siswa atau dharma siswa. Beasiswa adalah bantuan untuk

membantu orang terutama bagi yang masih sekolah atau kuliah agar mereka dapat menyelesaikan tugasnya dalam rangka mencari ilmu pengetahuan hingga selesai. Bantuan ini biasanya berbentuk dana untuk menunjang biaya atau ongkos yang harus dikeluarkan oleh anak sekolah atau mahasiswa selama menempuh masa pendidikan di tempat belajar yang diinginkan. Beasiswa adalah pemberian berupa bantuan keuangan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan.

II.2.7. *Android*

Android adalah suatu sistem operasi yang berbasis *Linux* dengan perangkat *Mobile device*, *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi-aplikasi baru untuk membantu kebutuhan manusia. Awal terdirinya *android* dimulai dari *Google Inc* yang membeli *Android inc* pendatang bat dari perangkat lunak untuk ponsel/*smartphone*. Semakin banyak *smartphone* dan *PC Tablet* menggunakan sistem operasi dengan versi yang berbdeda. Semakin tinggi versinya maka fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah *HTC Dream* yang dirilis pada tanggal 22 Oktober 2008. Perkembangan sistem *Android* berkembang sesuai versi keluarnya adapun versi *Android* dimulai dari (Muhamad Zaelani ; 2017: 5)

Tabel. II.1 Versi Android

Versi	Nama	Rilis	Catatan
1.0	<i>Android 1.0</i>	23 September 2008	<i>Android</i> pertama hanya untuk <i>smartphone</i>
1.1	<i>Android 1.1</i>	9 Februari 2008	
1.5	<i>Cupcake</i>	30 April 2009	Mulai pakai kode nama
1.6	<i>Donut</i>	15 September 2009	
2.0- 2.1	<i>Éclair</i>	26 Oktober 2009 (2.0) 12 Januari 2010 (2.1)	
2.2	<i>Froyo</i>	20 Mei 2010	
2.3	<i>Gingerbread</i>	6 Desember 2010	Digunakan pada <i>smartphone</i> jenis lama
3.0- 3.2	<i>Honeycomb</i>	22 Februari 2011 (3.0) 10 Mei 2011 (3.1) 15 Juli 2011 (3.2)	Hanya untuk tablet
4.0	<i>ICS (Ice Cream Sandwich)</i>	19 Oktober 2011	<i>Smartphone</i> dan tablet
4.1- 4.3	<i>Jelly Bean</i>	9 Juli 2012 (4.1) 13 November 2012 (4.2) 24 Juli 2013 (4.3)	<i>Update</i> untuk memperbaiki dan menambah <i>fitur</i> pada <i>ICS</i>
4.4	<i>Kit kat</i>	3 September 2013	
5.0	<i>Lollipop</i>	12 november 2014 (5.0) 9 Maret 2018 (5.1)	
6.0	<i>Marshmallow</i>	5 oktober 015	Terdapat <i>daze mode</i> , <i>Do Not Disturb mode</i> , mendukung <i>USB tipe C</i> , mendukung pembacaan <i>fingerprint</i> .
7.0	<i>Nougat</i>	22 Agustus 2016	Multitasking

Sumber: Muhamad Zaelani (2017: 5)

Adapun penjelasan dari versi as android diatas adalaha sebagai berikut :

1. *Android 1.0: Astro (Alpha)*

Kedua versi awal *Android* ini mungkin agak asing kamu dengarkan. Pasalnya versi *Android 1.0 Astro (Alpha)* dan *Android 1.1* Walaupun belum

menggunakan nama makanan manis, kedua sistem operasi *Android* ini tentu menjadi *pionir*. Pasalnya di sinilah *Android* bermula lewat *smartphone* pertama, *HTC Dream*.

2. *Android 1.1 : Bender (Beta)*

Android 1.1 Bender (Beta) ini belum diluncurkan secara publik untuk kebutuhan komersil. *Platform Android* sendiri pertama kali diluncurkan pada September 2008 dengan andil Andy Rubin yang saat ini dikenal sebagai Bapak *Android*. Walaupun belum menggunakan nama makanan manis, kedua sistem operasi *Android* ini tentu menjadi *pionir*. Pasalnya di sinilah *Android* bermula lewat *smartphone* pertama, *HTC Dream*. (Muhamad Zaelani ; 2017: 5)

3. *Cupcake*

Mulai dari versi ini, *Android* menggunakan nama makanan manis untuk setiap versi yang diluncurkan. *Android 1.5 Cupcake* sendiri dirilis pada tanggal 30 April 2009 dengan berbagai fitur di sebuah perangkat *smartphone* untuk menggantikan *feature phone* kala itu. (Muhamad Zaelani ; 2017: 5)

4. *Donut*

Tentu pada awal perilisannya, sistem operasi *Android* tetap memiliki banyak *bug* yang pengembangnya perlu mengadakan perbaikan. Hal ini dilakukan pada *Android 1.6 Donut* yang dirilis pada 15 September 2009. Ya artinya belum genap setahun semenjak perilisan *Android 1.5 Cupcake* atau hanya berselang 5 bulan saja. *Android* pun menambahkan beberapa pembaruan, terutama dukungan pada layar *smartphone* yang lebih besar. (Muhamad Zaelani ; 2017: 5)

5. *Eclair*

Sama seperti versi sebelumnya, *Android 2.0 & 2.1 Eclair* masih berfungsi untuk menutupi *bug* yang masih ditemukan pada sistem operasi *mobile* ini. Di samping itu, *Android* juga menambah berbagai fitur di dalamnya. Mulai dari dukungan *Bluetooth* hingga fitur kamera yang mulai menjadi nilai jual *smartphone* kala itu. *Android 2.0 & 2.1 Eclair* digunakan pada perangkat seperti *HTC Nexus One*. (Muhamad Zaelani ; 2017: 5)

6. *Android 2.2 : Froyo (Frozen Yoghurt)*

Mulai versi ini *Android* tampaknya sudah mulai dikenal luas oleh berbagai *brand smartphone*. *Android 2.2 Frozen Yoghurt* alias *Froyo* ini dirilis pertama kali pada tanggal 20 Mei 2010. Walaupun sudah mulai dipergunakan pada beberapa *brand*, namun tetap saja *Android* masih kalah bersaing dengan *Symbian* yang mendominasi pasar *featured phone*. *Android 2.2 Froyo* memberikan peningkatan pada kecepatan kerja, fitur *USB tethering*, *WiFi hotspot*, serta fitur keamanan. (Muhamad Zaelani ; 2017: 5)

7. *Gingerbread*

Belum setahun berselang lagi, *Android 2.3 Gingerbread* kembali diluncurkan pada Desember 2010 dengan berbagai peningkatan yang cukup signifikan. Hal ini terutama pada tampilan tatap muka alias *user interface* yang digunakan. Mulai versi ini, banyak *brand smartphone* mulai melirik menggunakan sistem operasi *Android*. Salah satunya *Samsung Galaxy Series* yang populer hingga saat ini. (Muhamad Zaelani ; 2017: 5)

8. *Honeycomb*

Untuk para pengguna *smartphone* mungkin akan agak asing dengan versi *Android* yang satu ini. *Android* 3.0 & 3.2 *Honeycomb* yang menggunakan ikon lebah ini memang diperuntukkan penggunaannya khusus untuk perangkat *tablet*. Tentu perilisan *Android* 3.0 & 3.2 *Honeycomb* pada 10 Mei 2011 ini untuk mendukung *Samsung* yang mulai merilis perangkat *tablet*, *Samsung Galaxy Tab Series* untuk menyaingi *Apple iPad*. (Muhamad Zaelani ; 2017: 5)

9. *Ice Cream Sandwich (ICS)*

Selain itu, *Android* pun juga merilis versi *Android* 4.0 *Ice Cream Sandwich* yang kembali diperuntukkan untuk perangkat *smartphone*. *Android* 4.0 *Ice Cream Sandwich* sendiri dirilis pada 19 Oktober 2011 silam. Punya versi nama paling panjang hingga saat ini, *Android* 4.0 *Ice Cream Sandwich* memberikan banyak pembaruan. Mulai dari animasi yang semakin halus, sederhana, dan mudah digunakan. (Muhamad Zaelani ; 2017: 5)

10. *Jelly Bean*

Peningkatan signifikan terasa saat menggunakan *Android* 4.1 & 4.3 *Jelly Bean*. Sistem operasi ini sendiri pertama kali dirilis pada Juni 2012 dengan membawa sejumlah peningkatan terutama di sektor pengolahan grafis. Dengan begini, tentu *Android* 4.1 & 4.3 *Jelly Bean* bisa memberikan peningkatan fungsi pada *user interface* dan teknologi *Vsync* yang digunakannya. (Muhamad Zaelani ; 2017: 5)

11. *KitKat*

Menggunakan nama *brand* cemilan terkenal, *Android 4.4 KitKat* pertama kali dirilis pada Oktober 2013. Versi *Android* terbaik ini pun bisa dikatakan menjadi favorit karena mendukung hampir seluruh perangkat *smartphone* di dunia. Hal ini dikarenakan *Android 4.4 KitKat* dapat memberikan optimalisasi yang baik termasuk pada perangkat *smartphone* yang memiliki *spesifikasi* kurang mumpuni alias cukup rendah saat itu. (Muhamad Zaelani ; 2017: 5)

12. *Lollipop*

Mulai beberapa versi ke belakang, *Android* dan *Google* pun mulai secara rutin memperbarui sistem operasi mereka dalam selang waktu setahun. Termasuk *Android 5.0 & 5.1 Lollipop* yang dirilis dan diresmikan pada Juni 2014. Bisa dibilang *Android 5.0 & 5.1 Lollipop* menjadi *pionir* dibuatnya *smartphone flagship* dengan *spesifikasi* cukup mumpuni. Versi *Android* ini sudah mendukung arsitektur 64-bit yang sudah memungkinkan penggunaan RAM di atas 3GB. Salah satunya *ASUS Zenfone 2* yang sudah mengusung RAM 4GB saat itu. (Muhamad Zaelani ; 2017: 5)

13. *Marshmallow*

Android 6.0 Marshmallow menjadi suksesor dari versi *Android* sebelumnya. Sistem operasi ini sendiri pertama kali diperkenalkan pada Mei 2018 dan mulai dirilis pada Oktober 2018 silam. Sistem operasi ini secara jelas memberikan peningkatan pada sistem keamanan dengan dihadirkannya *fingerprint sensor* sebagai sistem keamanan *biometrik* yang digunakan. Selain digunakan untuk mengunci layar, *fingerprint sensor* ini dapat digunakan untuk otentikasi

Google Play Store dan pembelian dengan menggunakan *Android Pay*.
(Muhamad Zaelani ; 2017: 5)

14. *Nougat*

Untuk saat ini, sistem operasi *Android* ini masih digunakan beberapa *smartphone* yang baru dirilis belakangan ini. *Android 7.0 & 7.1 Nougat* pertama kali diperkenalkan pada Juni 2016 dengan menampilkan ikon robot *Android* dengan batangan *Nougat*. Sistem operasi *Android 7.0 & 7.1 Nougat* mengalami perubahan dari segi tampilan antarmuka. Selain itu ada juga *fitur splitscreen* untuk membagi tampilan layar untuk dua aplikasi sekaligus. (Muhamad Zaelani ; 2017: 5)

II.2.8. *Android Studio*

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi *Android*, berdasarkan IntelliJ IDEA. Konsep yang dimiliki *Android Inc* ternyata menggugah minat *Google* untuk memilikinya. Pada bulan Agustus 2005, Akhirnya *Android Inc* diakuisisi oleh *Google Inc*. seluruh sahamnya dibeli oleh *Google*. Banyak yang memperkirakan nilai pembelian *Android Inc* Oleh *Google* adalah sebesar USD 50 juta. saat itu banyak yang berspekulasi bahwa akuisisi ini adalah langkah awal yang dilakukan *Google* untuk masuk kepasar mobile phone. Maka dari situlah banyak pengembang sistem maupun software berlomba lomba untuk membuat atau merancang sistem *Android* menggunakan software – software yang support dengan *Android*, dan sebagai contoh disini kita pengenalan *Android Studio*. Selain

merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, *Android Studio* menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi *Android*, misalnya:

1. Sistem versi berbasis *Gradle* yang *fleksibel*
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat *Android*
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah - masalah lain
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan App Engine

Laman ini berisi pengantar dasar fitur-fitur *Android Studio*. Untuk memperoleh rangkuman perubahan terbaru, lihat Catatan Rilis *Android Studio* (*Google developer* ; 2017).

II.2.9 Metode ARAS (*Additive Ratio Assesment*)

ARAS merupakan metode yang didasarkan pada prinsip intuitif bahwa alternatif harus memiliki rasio terbesar untuk menghasilkan solusi yang optimal.

Metode ARAS melakukan perangkingan dengan membandingkan nilai setiap kriteria pada masing-masing alternative dengan melihat bobot masing-masing untuk memperoleh alternative yang ideal. Pada metode ARAS nilai fungsi utilitas yang menentukan efisiensi relatif kompleks dari alternatif yang layak berbanding lurus dengan efek relatif dari nilai dan bobot kriteria utama yang dipertimbangkan penentuan alternative terbaik. ARAS didasarkan pada argumen bahwa permasalahan yang rumit dapat dipahami dengan sederhana menggunakan perbandingan relatif. Pada ARAS, rasio jumlah nilai kriteria yang dinormalkan dan ditimbang, yang menggambarkan alternatif yang dipertimbangkan, dengan jumlah nilai kriteria normal dan tertimbang, yang menggambarkan alternatif yang optimal. Dalam pendekatan klasik, metode pengambilan keputusan multi-kriteria fokus pada peringkat. Metode MCDM membandingkan nilai fungsi utilitas solusi yang ada dengan nilai solusi alternatif positif yang ideal atau mengambil jarak ke solusi positif dan ideal negatif yang ideal menjadi pertimbangan. Sedangkan metode ARAS membandingkan fungsi utilitas dari alternatif dengan nilai fungsi utilitas yang optimal (Charis Maulana, dkk : 2019).

Adapun langkah-langkah dari metode *Additive Ratio Assesment* (ARAS) sebagai berikut :

1. Pembentukan Decision Making Matriks

$$X = \begin{bmatrix} X_{0i} & X_{0j} & \dots & X_{0n} \\ X_{1i} & X_{1j} & \dots & X_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{ni} & X_{nj} & \dots & X_{nn} \end{bmatrix} \quad (i = 0, m; \dots j = 1, n) \quad (1)$$

Dimana :

m = Jumlah Alternatif

n = Jumlah Kriteria

X_{ij} = Nilai performa dari alternatif ; terhadap kriteria J_{xoj} = nilai optimum dari kriteria J

Jika nilai optimum kriteria J (x_{oj}) Tidak diketahui, maka :

$$x_{oj} = \text{Max} \frac{\text{Max}}{l} = x_{ij} \text{ if } \frac{\text{Max}}{l} \cdot x_{ij} \text{ is Preference} \quad (2)$$

$$x_{oj} = \text{Max} \frac{\text{Min}}{l} = x_{ij} \text{ if } \frac{\text{Min}}{l} \cdot x_{ij} \text{ is Prefeerable} \quad (3)$$

2. Penormalisasian matriks keputusan untuk semua kriteria

a. Jika kriteria beneficial (Max) maka dilakukan normalisasi mengikuti :

$$X_{ij}^* = \frac{x_{ij}}{\sum_{i=1}^m x_{ij}} \quad \rightarrow \text{Dimana : } x_{ij}^* \text{ adalah nilai normalisasi} \quad (4)$$

b. Jika kriteria non beneficial maka dilakukan normalisasi :

$$\rightarrow \text{Tahap 1} = x_{ij} - \frac{1}{x_{ij}} \quad (5)$$

$$\rightarrow \text{Tahap 2} = R - \frac{x_{ij}}{\sum_{i=1}^m x_{ij}} \quad (6)$$

3. Menentukan bobot matriks yang sudah dinormalisasikan

$$D = [d_{ij}] m \times n = r_{ij} \cdot w_j \quad \rightarrow \text{Dimana : } w_j = \text{bobot kriteria} \quad (7)$$

4. Menentukan nilai fungsi optimalisasi (S_i)

$$S_i = \sum_{j=1}^n d_{ij} : (i = 1, 2, \dots, m; j = 1, 2, \dots, n) \quad (8)$$

Dimana S_i adalah nilai fungsi optimalisasi alternatif i. Nilai terbesar adalah nilai yang terbaik, dan nilai yang paling sedikit adalah yang terburuk. Dengan memperhitungkan proses, hubungan proporsional dengan nilai dan bobot kriteria yang diteliti berpengaruh pada hasil akhir.

5. Menentukan tingkat peringkat tertinggi dari alternative

$$K_i = \frac{S_i}{S_0} \quad (9)$$

Dimana S_i dan S_0 merupakan nilai kriteria optimalitas, diperoleh dari persamaan sudah jelas. Itu dihitung nilai U_i berada pada interval dan merupakan pesanan yang diinginkan didahulu efisiensi relatif kompleks dari alternatif yang layak bisa ditemukan sesuai dengan nilai fungsi utilitas. (Charis Maulana, dkk : 2019).

II.2.10 Website

Website merupakan sebuah media informasi yang ada di internet. Website tidak hanya dapat digunakan untuk penyebaran informasi saja melainkan bisa digunakan untuk membuat toko online. Website adalah kumpulan dari halaman-halaman situs, yang biasanya terkumpul dalam sebuah domain atau subdomain, yang tempatnya berada di dalam World Wide Web (WWW) di Internet. Sebuah halaman web adalah dokumen yang ditulis dalam format HTML (Hyper Text Markup Language), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser. Semua publikasi dari website-website tersebut dapat membentuk sebuah jaringan informasi yang sangat besar (Yunita Trimarsiah : 2018).

II.2.11 PHP

PHP adalah bahasa pelengkap HTML yang memungkinkan dibuatnya aplikasi dinamis yang memungkinkan adanya pengolahan data dan pemrosesan data. Semua syntax yang diberikan akan sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke browser hanya hasilnya saja. Kemudian merupakan bahasa berbentuk script yang ditempatkan dalam server dan diproses di server. Hasilnya akan dikirimkan ke client, tempat pemakai menggunakan browser. PHP dikenal sebagai sebuah bahasa scripting, yang menyatu dengan tag-tag HTML, dieksekusi di server, dan digunakan untuk membuat halaman web yang dinamis seperti halnya Active Server Pages (ASP) atau Java Server Pages (JSP). PHP merupakan sebuah software Open Source (Reza Hermiati : 2021).

II.2.12 Mysql

MySQL merupakan suatu jenis database server yang sangat terkenal. MySQL termasuk jenis RDBMS (Relational Database Manajement System). MySQL mendukung bahasa pemrograman PH, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI. MySQL merupakan RDBMS (Relational Database Management System) server. RDBMS adalah program yang memungkinkan pengguna database untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada database memiliki relasi antara satu tabel dengan tabel lainnya. Beberapa keunggulan dari MySQL yaitu :

- a. Cepat, handal dan mudah dalam penggunaannya. MySQL lebih cepat tiga sampai empat kali dari pada database server komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan MySQL.
- b. Didukung oleh berbagai bahasa Database Server MySQL dapat memberikan pesan Error dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.
- c. Mampu membuat tabel berukuran sangat besar. Ukuran maksimal dari setiap tabel yang dapat dibuat dengan MySQL adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.
- d. Lebih murah MySQL bersifat open source dan didistribusikan dengan gratis tanpa biaya untuk UNIX platform, OS/2 dan Windows Platform. Melekatnya integrasi PHP dengan MySQL. Keterikatan antara PHP dengan MySQL yang sama-sama Software Open-Source sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan database server lainnya. Modul MySQL di PHP telah dibuat Built-in sehingga tidak memerlukan konfigurasi tambahan pada File konfigurasi Php ini (Reza Hermiati : 2021).

II.2.13 UML (*Unified Modelling Language*)

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)

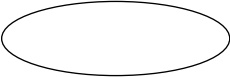
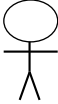
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan

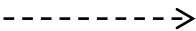
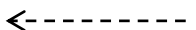
pengembangan sistem. Dalam membangun perancangan sistem dengan alat bantu perancangan *Unified Modeling Language* (UML) ada beberapa tahapan yang akan dilakukan, yaitu sebagai berikut :

II.13.1.1. *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. Adapun simbol *Use Case Diagram* yang dapat di lihat pada Tabel II.1. :

Tabel II.1. Simbol *Use Case diagram*

Gambar	Keterangan
	<i>Use case</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 adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam

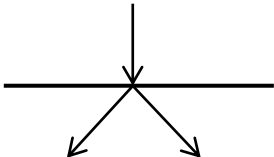
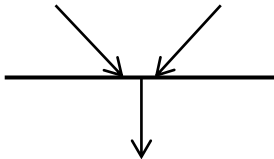
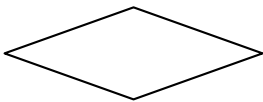
	beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan 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 : Janiver W. Janis; 2020)

II.2.13.2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Adapun simbol-simbol *activity diagram* yang dapat di lihat pada Tabel II.2 :

Tabel II.2. Simbol Activity Diagram

Gambar	Keterangan
	<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 paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, 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 : Janiver W. Janis; 2020)

II.2.13.3. Class Diagram (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint*

yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/ Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti. Adapun simbol-simbol *Multipcity class diagram* yang dapat di lihat pada Tabel II.3. :

Tabel II.3. Multiplicity Class Diagram

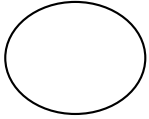
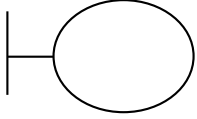
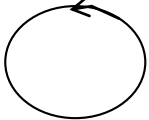
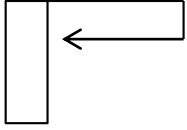
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Janiver W. Janis; 2020)

II.2.13.4. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Adapun simbol *Sequence Diagram* yang dapat di lihat pada Tabel II.4. :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</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> , <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 : Janiver W. Janis; 2020)