

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Berdasarkan penelitian dari (Asrul & Hidayatullah, 2018) dengan judul “Implementasi Metode *Trend Projection* Untuk Peramalan Persediaan Ikan Hias Air Tawar Studi Kasus: Batara Indo *Aquatic*” Penelitian ini bertujuan untuk merancang Sistem Pendukung Keputusan Peramalan Persediaan Ikan Hias Air Tawar Menggunakan Metode *Trend Projection*, untuk mengimplementasikan Sistem Pendukung Keputusan Peramalan Persediaan Ikan Hias Air Tawar Menggunakan Metode *Trend Projection*. Desain penelitian yang digunakan adalah UML yang didesain secara terstruktur yang terdiri dari rancangan model *use case diagram*, *class diagram*, *activity diagram* dan *sequence diagram*.

Berdasarkan penelitian yang dilakukan oleh (Mulyanto, 2019) dengan judul “Penerapan Forecasting Menggunakan Metode Trend Moment Untuk Menentukan Target Pengeluaran Barang Di PT.Telkom Akses” PT Telkom Akses (PTTA) merupakan anak perusahaan PT Telekomunikasi Indonesia, Tbk (Telkom) yang sahamnya dimiliki sepenuhnya oleh Telkom. PTTA bergerak dalam bisnis penyediaan layanan konstruksi dan pengelolaan infrastruktur jaringan. Pendirian PTTA merupakan bagian dari komitmen Telkom untuk terus melakukan pengembangan jaringan broadband untuk menghadirkan akses informasi dan

komunikasi tanpa batas bagi seluruh masyarakat Indonesia. Telkom berupaya menghadirkan koneksi internet berkualitas dan terjangkau untuk meningkatkan kualitas sumber daya manusia sehingga mampu bersaing di level dunia. PT Telkom Akses sering dihadapkan dengan pengiriman barang ke setiap cabang atau wilayah yang tidak beraturan dan berdampak menjadi barang yang banyak hilang atau tidak terrecondisi.

Berdasarkan penelitian dari (La Sufu et al., 2020) dengan judul “Implementasi Metode *Trend Projection* Dengan Algoritma *Trend Least Square* Pada Sistem *Inventory* Barang” PT. Pinus Merah Abadi Kendari memerlukan suatu alat bantu berupa perangkat lunak atau sistem yang dapat membantu memudahkan dan memaksimalkan kinerja pegawai administrasi gudang atau manajer perusahaan dalam meramalkan atau memprediksi jumlah barang yang harus di pesan ke pabrik untuk periode berikutnya. Untuk memudahkan dalam memprediksi persediaan barang tersebut maka dibuat penelitian yang bertujuan untuk membangun sebuah sistem yang dapat memprediksi penjualan barang dengan cara mengimplementasikan metode *Trend Projection*.

Berdasarkan penelitian dari (Wardah & Iskandar, 2017) dengan judul “Analisis Peramalan Penjualan Produk Keripik Pisang Kemasan Bungkus (Studi Kasus : *Home Industry* Arwana Food Tembilahan)” Peramalan yang dilakukan tiga metode yaitu metode *Moving Average*, metode *Exponential Smoothing with Trend* dan metode *Trend Anayisis* dengan membandingkan tingkat kesalahan (error) terkecil, maka metode peramalan yang terpilih yaitu metode *Trend Analysis*, dengan nilai MAD sebesar 161,3539, MSE sebesar 55744,16, dan

standar error sebesar 242,947. Dari analisis pengolahan data yang telah dilakukan berdasarkan metode peramalan yang terpilih, peramalan penjualan terhadap produk kripik pisang jenis kemasan bungkus adalah sebanyak 1121,424 atau 1122 bungkus/bulan, artinya pihak *Home Industry Arwana Food* Tembilahan harus menyediakan produk kripik pisang kemasan bungkus adalah sebanyak 1122 bungkus untuk tiap bulannya.

Berdasarkan penelitian dari (Iwan et al., 2018) dengan judul “Analisa Peramalan Permintaan Mobil Mitsubishi Xpander dengan Tiga Metode *Forecasting*” Untuk melakukan *forecasting* kita dapat menggunakan metode *Moving Average*, *Exponential smoothing* dan *Trend Analysis* untuk mendapatkan nilai terkecil dari peramalan. Berdasarkan pengumpulan dan pengolahan data serta analisa yang sudah dilakukan maka dapat kita bandingkan bahwa dari segi tingkat kesalahan serta perangkungan *Exponential smoothing* mempunyai tingkat kesalahan yang lebih kecil dibandingkan dengan metode lainnya.

Menurut (Mulyanto, 2019) Permasalahan yang umum dihadapi penjual kubis adalah bagaimana meramalkan atau memperkirakan penjualan kubis di masa mendatang berdasarkan data yang telah direkam sebelumnya. Peramalan tersebut sangat berpengaruh pada penjual kubis untuk sebagai bahan acuan memperkirakan penjualan kubis yang harus disediakan guna memenuhi permintaan pasar yang ada. Metode peramalan *Trend Moment* digunakan dalam peramalan penjualan kubis. Peramalan menggunakan data penjualan yang sudah ada selama 3 bulan, dengan data tersebut bisa diperoleh siklus penjualan yang akan dijadikan acuan sistem peramalan ini.

Menurut penelitian (Yulian et al., 2020) Metode *Trend Moment* ini digunakan untuk dapat mengatasi permasalahan yaitu untuk meramalkan penjualan, sehingga pengaruh unsur subyektif dalam menentukan peramalan penjualan nantinya dapat di hindarkan. Semakin banyak data-data historis untuk perhitungan peramalan menggunakan metode *Trend Moment* semakin akurat pula peramalan yang dihasilkan, sehingga hasil peramalan tersebut dapat membantu pelaku pengusaha bisnis dalam mempertimbangkan berapa banyak persediaan stok barang di bulan berikutnya, agar tidak mengalami kekurangan atau kelebihan stok barang.

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)

Perkembangan berbagai perangkat elektronik di sekitar kita sudah memiliki kemampuan komputasi dan komunikasi wireless. Tetapi ada juga perangkat elektronik yang masih manual.(Yusfrizal, 2022). Semakin berkembang teknologi khususnya bidang robotika dapat dirasakan oleh semua kalangan, dimana teknologi merupakan hasil dari peradaban manusia yang semakin maju, yang sangat membantu dan mempermudah manusia dalam memenuhi kebutuhan hidupnya (Yusfrizal, 2021)

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 Informasi

Sistem Informasi merupakan kombinasi teratur dari orang-orang, perangkat keras (hardware), perangkat lunak (software, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Sistem Informasi sebagai suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan. Sistem informasi adalah gabungan yang terorganisasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi dan sumber data dalam mengumpulkan,

mengubah, dan menyebarkan informasi dalam organisasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Rusmawan, 2020)

II.2.3. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang saling berinteraksi membentuk satu kesatuan untuk mencapai sasaran. Komponen-komponen sistem informasi sering disebut dengan istilah blok bangunan (building block) yang terdiri dari :

1. Blok Masukan (Input Block)

Blok ini merupakan jalur input data yang masuk ke dalam sistem informasi.

2. Blok Model (Model Block)

Blok ini merupakan kombinasi dari prosedur, logika, dan model matematik yang tugasnya memanipulasi data input dan data yang tersimpan agar menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (Output Block) Blok ini merupakan jalur output atau hasil dari model yang diolah menjadi informasi yang diinginkan oleh penerima.

4. Blok Teknologi (Technology Block)

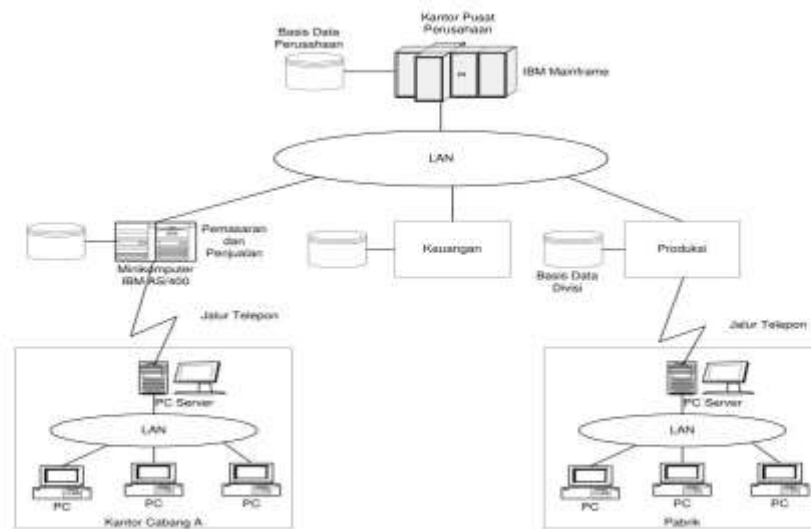
4. Blok teknologi merupakan jalur yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan output, mengirimkan output dan mengendalikan sistem secara keseluruhan

5. Blok Database (Database Block)

Blok database merupakan jalur pengumpulan data yang saling berhubungan satu dengan yang lain yang tersimpan didalam perangkat keras dan dimanipulasi menggunakan perangkat lunak yang selanjutnya diorganisasikan dengan sedemikian rupa sehingga menghasilkan informasi yang berkualitas.

6. Blok Kendali (Control Block)

Blok kendali merupakan jalur untuk mengambil tindakan dalam menjaga sistem agar tidak mengalami gangguan, seperti bencana alam, air, api, debu, suhu, kecurangan-kecurangan, sabotase, kesalahan-kesalahan, ketidakefisienan, dan kegagalan pada sistem itu sendiri. (Rusdi et al., 2020)



Gambar II.1 Arsitektur Sistem Infomasi

II.2.4. Pembelian

Pembelian adalah suatu kegiatan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan”. Dalam sistem pembelian terdapat dua macam, yaitu pembelian tunai dan kredit. Sistem pembelian tunai merupakan sistem yang diberlakukan oleh perusahaan untuk pengadaan barang yang diperlukan perusahaan”. Untuk mendapatkan barang tersebut harus melakukan pembayaran terlebih dahulu. (Rusdi et al., 2020)

Pembelian adalah suatu usaha yang dilakukan untuk pengadaan barang yang diperlukan oleh perusahaan. Pembelian dibedakan berdasarkan dari jenis pembelian yang dibedakan dalam 2 bagian, yaitu : Jenis pembelian berdasarkan pemasok : Pembelian lokal adalah pembelian dari pemasok yang berasal dari dalam negeri, Pembelian impor adalah pembelian dari pemasok yang berasal dari

luar negeri. Jenis pembelian berdasarkan transaksi : Transaksi pembelian tunai adalah jenis transaksi dimana pembayaran dilakukan secara langsung pada saat barang diterima. Transaksi pembelian kredit adalah jenis transaksi dimana pembayarannya tidak dilakukan secara langsung pada saat barang diterima, tetapi dilakukan selang beberapa waktu setelah barang diterima (tempo), sesuai perjanjian kedua belah pihak (Rusmawan, 2020)

II.2.5. Aplikasi

Pengertian aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer. (Listianto et al., 2017)

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. (Setiaji, 2020)

II.2.6. Peramalan

Peramalan atau forecasting adalah memperkirakan keadaan dimasa yang akan datang melalui pengujian keadaan dimasa lalu. Dalam kehidupan sosial segala sesuatu itu serba tidak pasti dan sukar diperkirakan secara tepat, sehingga

diperlukan peramalan. Peramalan yang dibuat selalu diupayakan agar dapat meminimumkan pengaruh ketidakpastian ini terhadap sebuah masalah. Dengan kata lain peramalan bertujuan mendapatkan peramalan yang bisa meminimumkan kesalahan meramal (*forecast error*) yang biasanya diukur dengan mean square error, mean absolute error, dan sebagainya. Ada beberapa pola data yang harus diperhatikan untuk peramalan, yaitu:

1. Pola Data Horizontal
2. Pola Data Musiman
3. Pola Data Sklis
4. Pola Data Trend (La Sufu et al., 2020)

II.2.7. Trend Projection

Metode peramalan dengan proyeksi *trend* ini adalah mencocokkan garis trend ke rangkaian titik data historis kemudian memproyeksikan garis tersebut ke masa depan dengan *horizon* waktu menengah dan panjang. Metode proyeksi pada penelitian ini melihat trend pada garis lurus (linier). Pendekatan yang tepat untuk trend linier adalah metode kuadrat terkecil (*least square*). Metode kuadrat terkecil adalah langkah untuk menentukan garis lurus yang paling sesuai untuk pola data yang muncul. Tujuan utama metode ini adalah meminimalkan jumlah kuadrat kesalahan atau selisih dari persamaan regresi linier dengan masing-masing data sebenarnya. Metode trend projection bisa disebut juga metode tren garis lurus. Adapun persamaan *trend linier* dapat ditulis sebagai berikut : (La Sufu et al., 2020)

$$Y' = a + bX \dots\dots\dots(1)$$

Dimana :

Y' = data berkala (time series data),

a dan b = konstanta

X = waktu (hari, minggu, bulan, tahun)

Pada umumnya yang diberi nilai 0 adalah *variable* waktu yang letaknya ditengah Untuk mencari nilai *variable* waktu adalah sebagai berikut :

1. Untuk nilai *variable* waktu (= X) adalah nol (0)

a. Untuk nilai n ganjil adalah :

$$n = 2k + 1 \quad k = \frac{n-1}{2} \dots\dots\dots(2)$$

$$\frac{Xk+1}{2} = 0 \dots\dots\dots(3)$$

b. Untuk nilai n genap adalah :

$$n = 2k \quad k = \frac{n}{2} \dots\dots\dots(4)$$

$$\frac{Xk - (k+1)}{3} = 0 \dots\dots\dots(5)$$

(La Sufu : 2020)

II.2.8. Metode Trend Moment

Metode *trend moment* adalah merupakan metode untuk mencari garis trend dengan perhitungan statistika dan matematika tertentu guna mengetahui fungsi garis lurus sebagai pengganti garis patah-patah yang dibentuk oleh data historis. Dengan demikian pengaruh unsur subjektif dapat dihindarkan.

Persamaan trend dengan metode *trend moment* adalah sebagai berikut

$$Y = a + bX$$

Dimana :

Y : nilai trend (Peramalan)

a : bilangan konstant

b : slope atau koefisien kecondongan garis trend

X : indeks waktu (x = 0, 1, 2, 3, ..., n)

Sedangkan untuk menghitung nilai a dan b digunakan rumus pada persamaan II.1 dan persamaan II.2.

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{(\sum X^2) - (\sum X)^2} \dots\dots\dots(\text{II.1})$$

$$a = \frac{\sum Y - b(\sum X)}{n} \dots\dots\dots(\text{II.2})$$

Dimana :

$\sum X$: Jumlah kumulatif dari periode waktu

$\sum Y$: Jumlah kumulatif data penjualan

$\sum XY$: Jumlah kumulatif dari jumlah periode dikalikan jumlah penjualan

n : Banyaknya periode waktu (bulan)

Setelah nilai ramalan yang telah diperoleh dari hasil peramalan dengan metode Trend Moment akan dikoreksi terhadap pengaruh musiman dengan menggunakan indeks musim dengan rumus :

Rata – Rata Permintaan Tertentu / rata – rata permintaan perbulan

Untuk mendapatkan hasil ramalan akhir setelah dipengaruhi indek musim maka akan menggunakan perhitungan sebagai berikut

$$Y^* = \text{Indeks Musim} \times Y$$

Dimana :

Y^* = Hasil ramalan dengan menggunakan metode Trend Moment yang telah dipengaruhi oleh indeks musim.

Y = Hasil ramalan dengan menggunakan Trend Moment. (Mulyanto, 2019)

II.2.9 *Android*

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. 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 perilisan 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 (Samosir, 2018)

Android merupakan sistem operasi *mobile*. *Android* tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data data ponsel sekalipun, atau data sistem sendiri. *Android* merupakan sebuah sistem

operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Beberapa pengertian lain dari *Android*, yaitu :

1. Merupakan *platform* terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli *Google Inc.* dari *Android Inc.*
3. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time enviroment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil. (Ni Kadek Ceryna, dkk: 2018 : 101)

Android merupakan sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh (*mobile*) seperti *smartphone* dan *tablet*. *Android* adalah sistem operasi yang bersifat *open source* dan *google* merilis kodenya di bawah lisensi *apache*. *Android* awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari *google* dan kemudian *google* membelinya pada tahun 2005. *Android* dibuat menggunakan bahasa pemrograman C(core), C++, Java(UI) dan dukungan berbagai bahasa pemrograman lain. Berdasarkan *dashboard developer android*, *Android 8.0/1 Oreo* adalah versi *Android* yang paling banyak digunakan, yakni sekitar 28.3% dari keseluruhan perangkat *Android* di seluruh dunia. Hingga saat ini, *Android* sistem operasi berbasis *mobile* yang paling banyak digunakan di seluruh dunia dan android sudah di rilis versi 10 pada 3 september 2019. (Kaban et al., 2019)

II.2.10. Andorid Studio

Android Studio adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi *Android* yang dibangun berdasarkan perangkat lunak IntelliJ IDEA. Selain editor kode dan alat pengembang yang didesain khusus untuk pengembangan *android*, *Android studio* merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi *android*. *Android studio* menyediakan banyak fitur untuk meningkatkan produktivitas dalam pengembangan aplikasi berbasis android seperti tersedianya sistem pembangunan berbasis *gradle* yang *fleksibel*, emulator yang memiliki banyak fitur, integrasi dengan GitHub, dukungan bawaan untuk *Google Cloud Platform* sehingga mempermudah pengembang aplikasi untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine*. (Kaban et al., 2019)

II.2.11 Tools Pembangunan Android

Untuk membangun sebuah sistem operasi *Android* dapat menggunakan Mac, Windows PC, ataupun *Linux*. *Tools* yang dibutuhkan gratis dan dapat di *download* dari *web*. Berikut adalah beberapa *tools* yang digunakan untuk membangun aplikasi *android*.

1. JDK (*Java Development Kit*)
2. Android SDK
3. ADT (*Android Development Tools*) (Ceryna Dewi et al., 2018)

II.2.12. Perkembangan Android

Adapun perkembangan Android adalah sebagai berikut :

a. Android versi 1.1

Android memang diluncurkan pertama kali pada tahun 2007, namun sistem operasi ini mulai dirilis dan diterapkan ke berbagai gadget pada tanggal 9 Maret 2009 silam. Android versi 1.1 merupakan Android awal yang dimana versi ini baru memberikan sentuhan di beberapa aplikasinya seperti sistem antar muka bagi pengguna (user interface) yang lebih baik, serta beberapa aplikasi yang lain.

b. Android versi 1.5 (Cupcake)

Pada bulan Mei 2009 Android kembali mengalami perubahan versi. Android versi 1.1 kemudian disempurnakan dengan Android versi 1.5 atau yang dikenal sebagai Android Cupcake.

c. Android versi 1.6 (Donut)

Donut (versi 1.6) diluncurkan dalam tempo kurang dari 4 bulan semenjak peluncuran perdana Android Cupcake, yaitu pada bulan September 2009.

d. Android versi 2.0/2.1 (Eclair)

Masih ditahun yang sama, Android kembali merilis operating sistem versi terbarunya, yaitu Android versi 2.0/2.1 Eclair. Android Eclair diluncurkan oleh Google 3 bulan setelah peluncuran.

e. Android versi 2.2 (Froyo: Frozen Yoghurt)

Butuh 5 bulan bagi Google untuk melakukan regenerasi dari Android Eclair versi sebelumnya ke versi Froyo Frozen Yoghurt. Pada tanggal 20 Mei 2010, Android versi 2.2 alias Android Froyo ini dirilis.

f. Android versi 2.3 (Gingerbread)

7 bulan kemudian Android kembali melakukan gebrakan dengan merilis kembali Android versi 2.3 atau yang dikenal sebagai Android Gingerbread.

g. Android versi 3.0/3.1 (Honeycomb)

Pada bulan Mei 2011 Android versi 3.0/3.1 atau Android Honeycom dirilis. Android Honeycomb merupakan sebuah sistem operasi Android yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis Android.

h. Android versi 4.0 (ICS: Ice Cream Sandwich)

Android ICS atau Ice Cream Sandwich juga dirilis pada tahun yang sama dengan Honeycomb, yaitu pada bulan Oktober 2011.

i. Android versi 4.1 (Jelly Bean)

Android Jelly Bean merupakan versi Android yang terbaru pada saat ini. Salah satu gadget yang menggunakan sistem operasi Jelly Bean adalah Google Nexus 7 yang diprakarsai oleh ASUS, vendor asal Taiwan yang juga menjadi teman satu kampung halaman dengan Acer.

j. Android versi 4.4 (Kit Kat)

Kehadiran android kitkat merupakan peluncuran produk OS anyar yang diluncurkan pada 4 september 2013, sebelumnya banyak kabar beredar jikalau android akan meluncurkan OS baru yang bernama Android Key Lime Pie

namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan OS Android KitKat yang sebagian besar orang sudah familiar dengan itu.

k. Android versi 5.0.2 (Lollipop)

Android Lollipop merupakan keberadaan OS Android yang memang saat ini sudah menjadi trend baru di industri smartphone, hal ini tak lepas dari keunikan dan kelebihan yang banyak di miliki dari OS tersebut. Kehadiran android versi ini amat di nanti oleh sekian banyak orang karna diharapkan sistem operasi Lollipop ini bias lebih baik dibandingkan versi-versi sebelumnya.

l. Android versi 6.0 (Marshmallow)

Android 6.0 Marshmallow adalah versi dari sistem operasi mobile Android. Pertama kali diperkenalkan Mei 2015 di Google I / O di bawah kode nama Android M, secara resmi dirilis pada Oktober 2015. Android Marshmallow memperkenalkan model izin aplikasi didesain ulang sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. (Samosir, 2018)

II.2.13 Basis Data (*Database*)

Basis data atau *database* merupakan koleksi dari data-data yang terorganisir dengan rapi sehingga data dapat dengan mudah disimpan dan dimanipulasi. Kita dapat menjumpai pemanfaatan *database* dalam kehidupan sehari-hari, seperti penggunaan mesin ATM, sistem akademi kuniversitas/sekolah,

sistem informasi penjualan. Salah satu tujuan dari *database* adalah memberikan pengguna suatu pandangan abstrak dari data, yaitu sistem menyembunyikan rincian bagaimana data disimpan dan dipelihara. Sistem *database* harus dibuat semudah mungkin untuk dimengerti karena kebanyakan pengguna sistem *database* adalah orang-orang yang kurang terlatih di bidang teknologi. (Ceryna Dewi et al., 2018)

II.2.14. Java

Java adalah salah satu bahasa pemrograman yang berbasis objek secara murni/ Object Oriented Programming (OOP). Semua tipe data diturunkan dari kelas dasar yang disebut Object. Dengan konsep seperti itu, pemrograman dalam hal mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java, dapat dilakukan secara mudah, cepat, tepat dan lebih terorganisir (Putra Yudha et al., 2018)

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan

dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi (Udaksana & Kusaeri, 2018)

II.2.15. *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 (Hermiati et al., 2021)

II.2.16. 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 (Cholil & Prisiswo, 2020)

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 .(Hermiati et al., 2021)

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bias digunakan pada HTML. PHP merupakan singkatan dari “PHP : *Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen HTML, sekaligus bekerja di sisi server (*server-side HTML-embedded scripting*). Artinya sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada halaman HTML biasa, sehingga script-nya tak tampak disisi client. PHP dirancang untuk dapat bekerja sama dengan database *server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses database menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi di mana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*.(Susilo, 2018)

II.2.17. UML (*Unified Modelling Language*)

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. (Janis et al., 2020)

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*.

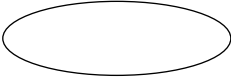
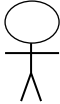
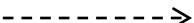
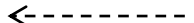
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*. (Alfina & Harahap, 2019)

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 :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk melakukan (*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. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 sebagai berikut :

Tabel II.1. Simbol Use Case

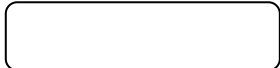
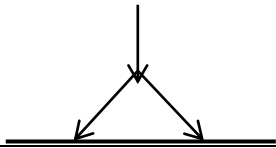
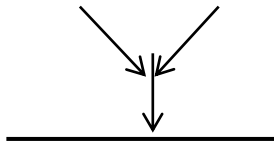
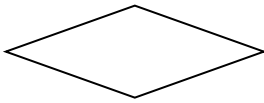
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi 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.
 <<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.
 <<extend>>	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Janiver W. Janis : 2020)

2. Diagram Aktivitas (*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 sebagai berikut:

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 diagram</i> untuk menunjukkan siapa melakukan apa.

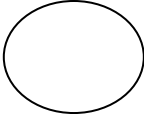
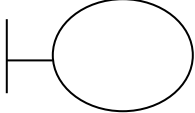
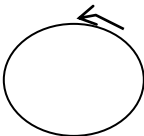
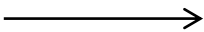
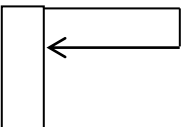
(Sumber : Janiver W. Janis : 2020)

3. 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. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 sebagai berikut:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	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>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry 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)

4. *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 yang dapat dilihat pada tabel II.4 sebagai berikut:

Tabel II.4. *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)