

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian terkait bertujuan untuk mendapatkan bahan perbandingan dan acuan. Selain itu, untuk menghindari anggapan kesamaan dengan penelitian ini. Maka dalam tinjauan pustaka ini peneliti mencantumkan hasil-hasil penelitian terdahulu sebagai berikut.

Berdasarkan penelitian Daniel Theodarus (2021) dengan judul “*Machine Learning* Rekomendasi Produk Dalam Penjualan Menggunakan Metode *Item-Based Collaborative Filtering*” Analisa Data menjadi *point* penting dalam penelitian ini, karena memberikan pelayanan terbaik (*user experience*) kepada pelanggan. Dengan banyaknya data yang tersedia, menyebabkan mengalami kesulitan dalam memberikan Rekomendasi produk kepada pelanggan. Sistem rekomendasi muncul sebagai solusi dalam merekomendasikan produk pada penelitian ini yang menggunakan metode *Collaborative Filtering*. Berdasarkan interaksi antara pelanggan dengan pelanggan lainnya yang terdapat didalam histori penjualan. Hasil dari penelitian ini ialah perusahaan berhasil meningkatkan penjualan, memudahkan pelanggan untuk menemukan produk yang dibutuhkan dan meningkatkan layanan terbaik kepada pelanggan.

Berdasarkan penelitian dari Muhammad Busyro (2020) dengan judul “Perancangan Aplikasi Wisata Kuliner Menggunakan Algoritma *Collaborative*

Filtering dan *Algoritma Fp-Growth* Berbasis Android”. Rekomendasi wisata kuliner merupakan hal yang penting dalam penelitian ini karena dengan menggunakan rekomendasi wisata kuliner dapat membantu wisatawan menentukan wisata kuliner yang akan dikunjungi terkhusus yang ada di kota medan, dengan banyaknya wisata kuliner yang ada di kota medan terkadang dapat membuat wisatawan kesulitan dalam menentukan tujuannya karena beragamnya wisata kuliner yang tersedia.

Berdasarkan penelitian dari Kevin Hartarto Muliadi (2019) dengan judul “Rancang Bangun Sistem Rekomendasi Tempat Makan Menggunakan Algoritma *typicality based Collaborative Filtering*” penelitian ini merekomendasikan tempat makan dengan metode *collaborative filtering*. Peneliti hanya menggunakan empat item untuk rating rasa dan porsi makanan hanya dengan menggunakan metode *collaborative filtering*, tempat makan baru semakin banyak dan hal ini menyebabkan seseorang semakin sering makan diluar. Dalam memilih tempat makan baru biasanya seseorang akan mencari rekomendasi agar tempat makan baru yang akan dikunjungi sesuai dengan preferensi dan tidak mengecewakan. namun berdasarkan pengujian yang telah dilakukan diketahui bahwa aplikasi ini memiliki nilai rata-rata Mean Absolute Error sebesar 1.366 yang disebabkan karena kurangnya data training. Untuk itu diperlukan penambahan data training pada penelitian ini untuk memperbaiki tingkat akurasi dari prediksi *rating* pada sistem rekomendasi.

Berdasarkan penelitian dari Mahardika Kharisma Adjie (2020) dengan judul “Penerapan Algoritma *FP-Growth* Pada Hasil Penimbangan Kendaraan

Angkutan Barang (Studi Kasus: UPPKB Balai Raja Bengkalis)” permasalahan pada penelitian ini adalah hasil penimbangan kendaraan angkutan barang pada UPPKB balai raja bengkalis menggunakan metode *fp-growth* data hasil penimbangan kendaraan angkutan barang di UPPKB dengan jumlah data awal dan atribut yang digunakan sebagai pola adalah jenis kendaraan, dan jenis pelanggaran. Selanjutnya proses Implementasi identifikasi pola menggunakan algoritma *FP-Growth* dilakukan dengan menggunakan bahasa pemrograman PHP dan diuji dengan tools SPMF. Penelitian ini berhasil dalam pengujian hasil penimbangan kendaraan angkutan barang pada UPPKB Balai Raja Bengkalis.

Berdasarkan penelitian dari Bondan Prasetyo (2019) dengan judul “Implementasi metode item-based *Collaborative Filtering* dalam pemberian Rekomendasi calon pembeli aksesoris *smartphone*” (Studi Kasus : Flazzstore) flazzstore bergerak dibidang penjualan casing *smartphone*, permasalahan yang umum dihadapi oleh para karyawan toko maupun user adalah bagaimana menentukan pilihan produk casing hp, terdapat banyak produk yang berbeda-beda dengan banyak tema yang berbeda pula. Hal ini membuat beberapa user kesulitan dalam menentukan pilihan mengenai produk yang akan dipilih. Perlunya sebuah sistem *rekomendasi* yang mampu memberikan rekomendasi produk kepada user untuk memudahkan user dalam membelinya. Maka dengan algoritma *collaborative filtering* ini dapat dirancanglah aplikasi yang memudahkan *user* dalam menentukan desain casing secara praktis.

Pada penelitian ini penulis menerapkan metode *Collaborative Filtering* dan *FP-Growth* untuk mengatasi masalah yang dihadapi oleh wisatawan yang

ingin berkunjung ke ekowisata yang ada di daerah kabupaten langkat karena dengan mengimplementasikan metode *Collaborative Filtering* dan *Fp-Growth* maka perhitungan *similarity* antar ekowisata yang telah di-rating oleh *user* dan prediksi rating dapat dilakukan dengan metode CF. dan metode *Fp-Growth* menjadi solusi untuk menyiapkan data set pola ekowisata. Hasil dari rekomendasi merupakan informasi alamat, input rating dan review yang telah diberikan oleh user sebelumnya. Maka kedua metode ini dapat memberikan sistem rekomendasi dan informasi kepada wisatawan maupun masyarakat yang ingin berkunjung ke ekowisata yang ada di kabupaten langkat dengan praktis dan tanpa membutuhkan waktu yang lama.

II.2. Uraian Teoritis

Berikut ini merupakan uraian teoritis yang berkaitan dengan penelitian yang diambil dari beberapa jurnal yang peneliti kumpulkan.

II.2.1. Implementasi

Implementasi adalah suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan terperinci biasanya dilakukan setelah perencanaan sudah dianggap sempurna. Implementasi dimaksudkan suatu kegiatan yang terencana, bukan hanya suatu aktifitas dan dilakukan secara sungguh-sungguh berdasarkan acuan norma-norma tertentu untuk mencapai tujuan kegiatan. Oleh karena itu, implementasi tidak berdiri sendiri tetapi dipengaruhi oleh objek berikutnya yaitu kurikulum. implementasi kurikulum merupakan proses pelaksanaan ide, program atau aktivitas baru dengan harapan

orang lain dapat menerima dan melakukan perubahan dan memperoleh hasil yang diharapkan. implementasi adalah perluasan aktivitas yang saling menyesuaikan proses interaksi antara tujuan dan tindakan untuk mencapainya serta memerlukan jaringan pelaksana birokrasi yang efektif (Guntur Setiawan, 2004:39).

Implementasi dalam ilmu komputer adalah realisasi dan spesifikasi teknis atau algoritma sebagai program, komponen perangkat lunak, atau sistem komputer lainnya melalui pemrograman dan penyebaran komputer, banyak implementasi mungkin ada untuk spesifikasi atau standar tertentu.

Implementasi bermuara pada aktivitas, aksi, tindakan atau adanya mekanisme suatu sistem, implementasi bukan sekedar aktivitas, tapi suatu kegiatan yang terencana dan untuk mencapai tujuan kegiatan (Nurdin Usman, 2002:70).

II.2.2. Collaborative Filtering

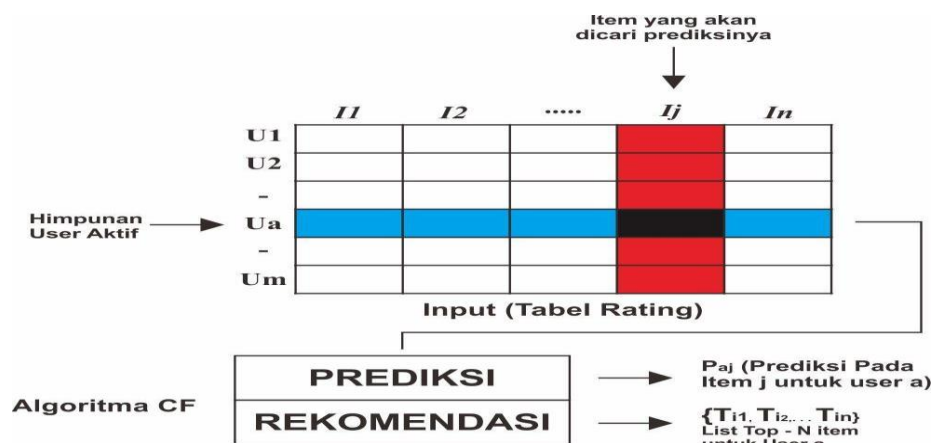
Sistem rekomendasi dengan metode *Collaborative filtering* adalah metode yang digunakan untuk memprediksi kegunaan item berdasarkan penilaian pengguna sebelumnya. *Collaborative Filtering* (CF) adalah suatu konsep dimana opini dari pengguna lain yang ada digunakan untuk memprediksi item yang mungkin disukai/diminati oleh seorang pengguna. Kualitas rekomendasi yang diberikan dengan menggunakan metode ini sangat bergantung dari opini pengguna lain (*neighbor*) terhadap suatu item. Belakangan diketahui bahwa melakukan reduksi *neighbor* (yaitu dengan memotong *neighbor* sehingga hanya beberapa pengguna yang memiliki kesamaan/similarity tertinggi sajalah yang akan

digunakan dalam perhitungan) mampu meningkatkan kualitas rekomendasi yang diberikan(Hartijah,2019).

Collaboartive filtering memberikan rekomendasi berdasarkan kumpulan dari pendapat, minat dan ketertarikan beberapa *user* yang biasanya diberikan dalam bentuk rating pada suatu item. Setiap pengguna akan mendapatkan rekomendasi dengan melakukan dua hal utama dalam menyelesaikan algoritma *collaborative filtering*, yaitu:

- Prediction*, melakukan prediksi opini yang akan diberikan oleh pengguna dalam skala bilangan yang sama.
- Recommendation*, memberikan rekomendasi berupa daftar item dengan nilai prediksi tertinggi. Hal yang perlu dicatat adalah item-item yang direkomendasikan belum pernah dibeli, dilihat atau di-rating oleh pengguna tersebut.

Berikut merupakan skema dan user item matriks dari *Collaborative filtering* dalam pemberian rekomendasi kepada user aktif dapat digambarkan dengan Tabel II.1



Tabel II.1. Rating Matriks dan skema *collaborative filtering*

Keterangan:

Ua : Himpunan user aktif

Ij : Item yang akan dicari prediksinya

CF : Collaborative Filtering

Terdapat dua metode atau pendekatan yang digunakan dalam *collaborative filtering* yaitu *user-based collaborative filtering* dan *item-based collaborative filtering* yaitu :

1. *User Based Collaborative* menggunakan teknik statistika untuk menemukan sekumpulan pengguna, dikenal sebagai tetangga (*neighbour*), yang memiliki sejarah setuju dengan pengguna yang menjadi sasaran. Setelah sekumpulan tetangga terbentuk, sistem menggunakan algoritma yang berbeda untuk menggabungkan kesukaan *neighbours* untuk menghasilkan prediksi atau rekomendasi N-teratas untuk aktif *user*.
2. *Item Based Collaborative* merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian rating terhadap suatu produk dengan produk yang dibeli. Dari tingkat kesamaan produk, kemudian dibagi dengan parameter kebutuhan pelanggan untuk memperoleh nilai kegunaan produk. Produk yang memiliki nilai kegunaan tertinggi lah yang kemudian dijadikan rekomendasi.

Metode *Collaborative Filtering* secara umum terdiri dari tiga langkah dalam pemberian rekomendasi kepada pengguna, yaitu: penemuan similar user, pembuatan ketetanggaan (*neighborhood*), dan perhitungan prediksi berdasarkan

tentangga yang dipilih . Untuk menghitung similar user, menggunakan persamaan *adjusted cosine similiarity* berikut:

$$SiN(i, j) = \frac{\sum (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}$$

Keterangan:

$SiN(i, j)$ = Nilai kemiripan *item i* dengan *item j*

$u \in U$ = Nilai *user* yang me-rating baik *item i* maupun *item j*

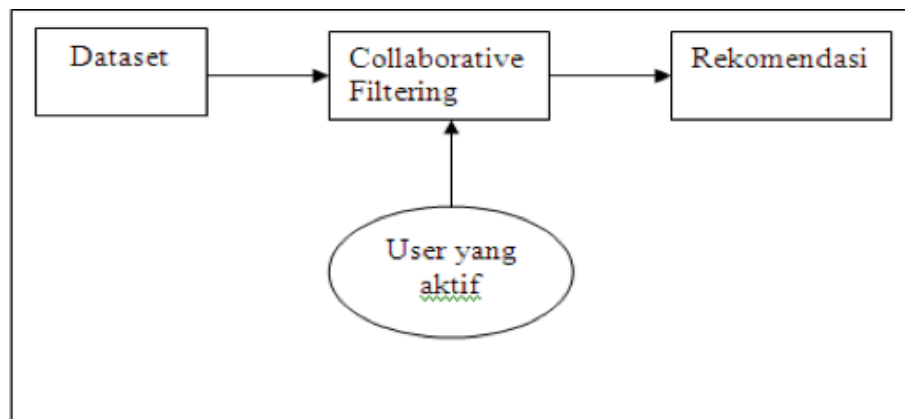
$R_{u,i}$ = Rating *user u* pada *item i*

$R_{u,j}$ = Rating *user u* pada *item j*

$\bar{R}_u$ = Nilai rating rata-rata *user u*

Bagi pengguna atau pelanggan yang ditujuh terhadap satu item atau lebih. Item dapat terdiri atas apa saja yang dapat disediakan manusia seperti misalnya buku, film, seni, artikel, atau tujuan wisata. Rating dalam *collaborative filtering* dapat berbentuk, (a) model rating skalar yang terdiri atas rating numerik seperti 1 sampai 5 (b) model rating biner dengan memilih antara setuju atau tidak setuju, atau dapat pula baik atau buruk (c) rating *unary* dapat mengindikasikan bahwa pengguna telah mengobservasi atau membeli item atau merating item dengan positif. Tidak tersedianya rating mengindikasikan tidak terdapat informasi yang menghubungkan pengguna dengan item. Rating dapat dikumpulkan secara *eksplisit, implisit*, ataupun gabungan antara eksplisit dan implisit. Rating eksplisit

yaitu rating yang didapatkan pada saat pelanggan/pengguna diminta menyediakan opini terhadap item tertentu. Rating implisit yaitu rating yang didapatkan melalui aksi yang dilakukan pelanggan. Untuk memperjelas penjelasan bagaimana proses *collaborative filtering* memberikan rekomendasi, maka dapat dilihat proses *collaborative filtering* pada gambar berikut ini :



Gambar II.1. *Collaborative Filtering* Proses

II.2.3. FP-Growth (*Frequent Pattern*)

Fp-Growth adalah salah satu dari beberapa metodologi pengembangan pola yang sering berpengaruh, di mana pola (misalnya, kumpulan item, urutan, subpohon, atau substruktur) sering terjadi sehingga frekuensi kemunculannya dalam database tidak kurang dari ambang minimum *support* yang ditentukan. Metode pertumbuhan pola (*Frequent*) menambang kumpulan data dengan cara *divide-end-conquer*. Pertama-tama memperoleh kumpulan pola frekuensi ukuran dan untuk setiap pola, ini memperoleh database yang diproyeksikan (atau bersyarat) oleh data secara rekursif. *Fp-Growth* merupakan penemuan informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data yang sangat besar (Raharjo, Yanti, Kudus, 2017). Algoritma fp-growth merupakan

pengembangan dari algoritma *Apriori*, sehingga kekurangan dari algoritma *apriori* diperbaiki di algoritma *fp-growth*. *Fp-growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset*, tidak menggunakan *generate candidate* seperti algoritma *apriori*. Hal ini lah yang menyebabkan Algoritma *FP-Growth* lebih cepat.

Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan berebentuk *tree* yang disebut dengan *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *Fp-Growth* dapat langsung mengekstrak *frequent itemset* dari *Fp-Tree*. Penggalian *itemset* yang *frequent* dengan menggunakan algoritma *FP-Growth* akan dilakukan dengan cara membangkitkan struktur data *tree* atau disebut dengan *FP-Tree* ng-ulang. Data akan dipresentasikan dalam bentuk *FP-Tree*.

Setelah *FP-Tree* terbentuk, digunakan pendekatan *divide and conquer* untuk memperoleh *Frequent itemset*. *FP-Tree* merupakan struktur data yang baik sekali untuk *Frequent Pattern Mining*. Struktur ini memberikan informasi yang lengkap untuk membentuk *Frequent Pattern*. Item - item yang tidak *frequent* (*infrequent*) sudah tidak ada dalam *FP-Tree*.

Metode *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu sebagai berikut :

1. Melakukan scan terhadap database , dalam tahap ini dihitung jumlah dari masing-masing *item* dalam transaksi. Dilakukan *scan* terhadap *database* untuk yang pertama kali. Hasil yang didapat adalah 1-*itemset* yang paling sering muncul dalam transaksi.

2. Membuat *FP-Tree* berdasarkan *1-itemset frequent* yang didapat sebelumnya. *FP-Tree* yang dibentuk dinamakan *root*. *Scan* kedua dilakukan pada *database* untuk membangun *FP-Tree*. Untuk setiap transaksi dilakukan pengurutan kembali *item* berdasarkan *1-itemset* yang didapat.
3. Tahap pencarian *frequent itemset*.

II.2.4. Pengenalan

Definisi Pengenalan dalam kamus bahasa indonesia (KBBI) adalah pe.nge.nal.an [n] proses, cara, perbuatan, mengenal atau mengenali, pengenalan dapat memudahkan dalam berkomunikasi maupun dalam menyampaikan pendapat yang ingin disampaikan kepada orang tertentu.

II.2.5. Ekowisata

Ekowisata merupakan suatu bentuk wisata yang bertanggung jawab terhadap kelestarian area yang masih alami (natural aren), memberi manfaat secara ekonomi dan mempertahankan keutuhan budaya bagi masyarakat setempat.

Ekowisata lebih populer dan banyak dipergunakan dibanding dengan terjemahan yang seharusnya dari istilah *ecotourism*, yaitu ekoturisme. Terjemahan yang seharusnya dari *ecotourism* adalah wisata ekologis. Yayasan Alam Mitra Indonesia (1995) membuat terjemahan *ecotourism* dengan ekoturisme. Di dalam tulisan ini dipergunakan istilah ekowisata yang banyak digunakan oleh para rimbawan. Hal ini diambil misalnya dalam salah satu seminar dalam Reuni

Fakultas Kehutanan Universitas Gadjah Mada (Fandeli, 1998). Kemudian Nasikun (1999) mempergunakan istilah ekowisata untuk menggambarkan adanya bentuk wisata yang baru muncul pada dekade delapan puluhan.

Definisi ekowisata yang pertama diperkenalkan oleh organisasi *The Ecotourism Society* (1990) sebagai berikut: Ekowisata adalah suatu bentuk perjalanan wisata ke area alami yang dilakukan dengan tujuan konservasi lingkungan dan melestarikan kehidupan dan kesejahteraan penduduk setempat. Semula ekowisata dilakukan oleh wisatawan pecinta alam yang menginginkan di daerah tujuan wisata tetap utuh dan lestari, di samping budaya dan kesejahteraan masyarakatnya tetap terjaga.

Namun dalam perkembangannya ternyata bentuk ekowisata ini berkembang karena banyak digemari oleh wisatawan. Wisatawan ingin berkunjung ke area alami, yang dapat menciptakan kegiatan bisnis. Ekowisata kemudian didefinisikan sebagai berikut: Ekowisata adalah bentuk baru dari perjalanan bertanggungjawab ke area alami dan berpetualang yang dapat menciptakan industri pariwisata (Eplerwood, 1999). Dari kedua definisi ini dapat dimengerti bahwa ekowisata dunia telah berkembang sangat pesat. Ternyata beberapa destinasi dari taman nasional berhasil dalam mengembangkan ekowisata ini. Bahkan di beberapa wilayah berkembang suatu pemikiran baru yang berkaitan dengan pengertian ekowisata. Fenomena pendidikan diperlukan dalam bentuk wisata ini. Hal ini seperti yang didefinisikan oleh *Australian Department of Tourism* (Black, 1999) yang mendefinisikan ekowisata adalah wisata berbasis pada alam dengan mengikuti aspek pendidikan dan interpretasi terhadap

lingkungan alami dan budaya masyarakat dengan pengelolaan kelestarian ekologis. Definisi ini memberi penegasan bahwa aspek yang terkait tidak hanya bisnis seperti halnya bentuk pariwisata lainnya, tetapi lebih dekat dengan pariwisata minat khusus, *alternative tourism* atau *special interest tourism* dengan obyek dan daya tarik wisata alam.

II.2.6. Aplikasi

Aplikasi ialah suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia.

1. Menurut Hengky W.Pramana, aplikasi adalah suatu unit perangkat lunak yang sengaja dibuat untuk memenuhi kebutuhan akan berbagai aktivitas ataupun pekerjaan, seperti aktivitas perniagaan, periklanan, pelayanan masyarakat, game, dan berbagai aktivitas lainnya yang dilakukan oleh manusia.
2. Menurut Ali Zaki dan Smadev *Community*, aplikasi merupakan komponen yang bermanfaat sebagai media untuk menjalankan pengolahan data ataupun berbagai kegiatan lainnya seperti pembuatan ataupun pengolahan dokumen dan file.
3. Menurut Abdurrahman dan Riswaya dalam (Riyowati & Fadlilah, 2019)
“Aplikasi merupakan program yang siap dipakai dan digunakan untuk menjalankan perintah *user* dengan tujuan untuk mendapatkan hasil yang tepat dan akurat sesuai dengan tujuan dari pembuatan aplikasi tersebut”

Aplikasi merupakan salah satu proses pemecahan salah satu masalah yang menggunakan teknik komputasi yang diinginkan dalam pemrosesan data.

Berdasarkan definisi tersebut dapat disimpulkan bahwa aplikasi merupakan sebuah program yang memiliki komponen-komponen yang bermanfaat sebagai media untuk menjalankan perintah user untuk mendapatkan hasil yang tepat dan akurat sesuai dengan tujuan dari pembuatan aplikasi.

II.2.7. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Aplikasi *Android* yang dikembangkan menggunakan *Java* dan mudah menyesuaikan ke platform baru. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari *boot loader*, *device driver*, dan fungsi-fungsi pustaka, hingga perangkat lunak API (*Application Programming Interface*), termasuk aplikasi SDK (*Software Development Kit*). *Android* bukanlah satu perangkat tertentu, melainkan sebuah *platform* yang dapat digunakan dan diadaptasikan untuk mendukung berbagai konfigurasi perangkat keras. Sistem operasi yang mendasari *Android* merupakan lisensi di bawah naungan GNU, *General Public License Versi 2 (GPLv2)*, yang biasa dikenal dengan istilah *Copy left*. Istilah *copy left* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh di bawah terms. Distribusi *Android* berada di bawah lisensi *Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua atau seterusnya. Pengembang aplikasi *Android* diperbolehkan

untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan. (Ganda Yoga Swara, dkk; 2017) .

II.2.8. Sejarah Android

Android adalah sistem operasi berbasis linux yang di modifikasi untuk perangkat bergerak (mobile devices) yang terdiri dari sistem operasi, middleware, dan aplikasi-aplikasi utama. Awalnya, Android dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. Sistem operasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisasi Open Handset Alliance tahun 2007. Selain Google, beberapa nama-nama besar juga ikut serta dalam Open Handset Alliance, antara lain Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba, dan Intel (Zamrony P. Juhara, 2016 : 1). Penamaan Sistem Operasi Android sangat unik karena menggunakan nama-nama makanan dan diawali dengan urutan alfabet. Untuk sekarang, versi sistem operasi terbaru dari Android adalah Android P 9.0 atau Pie.

Berikut merupakan perkembangan versi-versi Android dari masa ke masa, mulai dari versi 1.0 hingga versi 9.0 Pie.

1. *Android v1.0 Alpha* (28 September 2008)
2. *Android v1.1 Beta* (9 Februari 2009)
3. *Android v1.5 Cupcake* (27 April 2009)
4. *Android v1.6 Donut* (15 September 2009)
5. *Android v2.0-v2.5 Éclair* (26 Oktober 2009)

6. *Android v2.2-v2.2.3 Froyo* (20 Mei 2010)
7. *Android v2.3-v2.3.7 Gingerbread* (6 Desember 2010)
8. *Android v3.0-v3.2.6 Honeycomb* (22 Februari 2011)
9. *Android v4.0-v4.0.4 Ice Cream Sandwich* (18 Oktober 2011)
10. *Android v4.1-v4.3.1 Jelly Bean* (9 Juli 2012)
11. *Android v4.4-v4.4.4 KitKat* (31 Oktober 2013)
12. *Android v5.0-5.1.1 Lollipop* (12 November 2014)
13. *Android v6.0-v6.0.1 Marshmallow* (5 Oktober 2015)
14. *Android v7.0-7.1.2 Nugget* (22 Agustus 2016)
15. *Android v8.0-v8.1 Oreo* (21 Agustus 2017)
16. *Android v9.0 Pie* (6 Agustus 2018) (*Android.com*, 2019)
17. *Android 10*

II.2.9. Visual

Studio

Code

Visual Studio Code adalah editor *source code* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk *debugging*, *GIT Control* yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi. Visual Studio Code gratis dan *open-source*, meskipun unduhan resmi berada di bawah *lisensi proprietary*.

Teks editor Visual Studio Code juga bersifat open source, yang mana kode sumbernya dapat dilihat dan dapat berkontribusi untuk pengembangnya. Visual

Studio Code telah dirancang untuk bekerja dengan alat-alat yang ada, dan Microsoft menyediakan dokumentasi untuk membantu pengembang bersama, dengan bantuan untuk bekerja dengan ASP.NET 5, Node js, dan Microsoft naskah, serta alat-alat yang digunakan untuk membantu membangun dan mengolah aplikasi Node js.

II.2.10.

Flutter

Flutter merupakan sebuah SDK untuk pengembangan aplikasi *mobile* yang dikembangkan oleh Google untuk membangun aplikasi yang memiliki kinerja tinggi serta dapat dipublikasi ke *platform* Android dan iOS dari codebase tunggal. *Flutter* dapat dengan mudah dipelajari karena menggunakan bahasa pemrograman *Dart* yang pastinya terasa familiar jika sudah terbiasa menggunakan bahasa 12 pemrograman Java atau *Javascript*. Selain itu *flutter* juga menyertakan kerangka *reactive-functional*, mesin render 2D, widget siap pakai, dan tools untuk pengembangan. (CrossTechno, 2019).

II.2.11.

Xampp

Xampp adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah cross platform sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS

Windows, Mac OS, dan juga Solaris. Xampp bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website kepada orang lain tanpa harus terkoneksi dengan internet, cukup akses melalui Xampp control panel, atau istilahnya website offline.

II.2.12. *Laravel*

Laravel adalah framework aplikasi web kontemporer, open source dan digunakan secara luas untuk perancangan aplikasi web yang cepat dan mudah. *Laravel* dibuat oleh Taylor Otwell pada tahun 2011. Framework ini dibuat untuk pengembangan aplikasi website dimana mengikuti arsitektur MVC (model view controller).

MVC itu sendiri adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, controller, *user interface*.

II.2.13. Unified Modeling Language (UML)

Menurut Urva dan Siregar (2015:93), Unified Modelling Language adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan, dan membangun perangkat lunak. *Unified Modelling Language* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *Unified Modelling Language* saat ini sangat banyak digunakan dalam dunia industri yang

merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

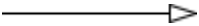
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *Unified Modelling Language* adalah sebagai berikut:

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. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel II.3. Simbol *Use Case Diagram*

Gambar	Nama	Keterangan
	Aktor	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> , tapi tidak memiliki kontrol terhadap <i>Use Case</i> .

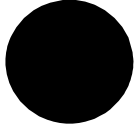
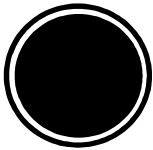
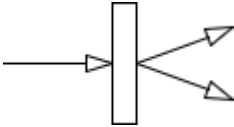
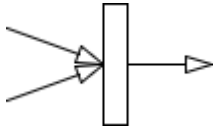
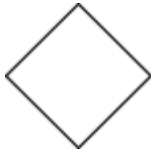
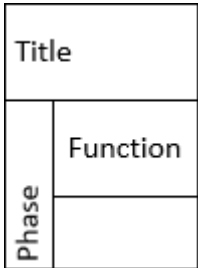
	<i>Use Case</i>	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bentukan pesan antar unit dengan Aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>Use Case</i> .
	<i>Association</i>	Asosiasi antara Aktor dan <i>Use Case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	<i>Generalization</i>	Asosiasi antara Aktor dan <i>Use Case</i> yang menggunakan panah terbuka untuk mengindikasikan bila Aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	Include, merupakan di dalam <i>Use Case</i> lain (required) atau pemanggilan <i>Use Case</i> oleh <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.
	<i>Extend</i>	Extend, merupakan perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Urva dan Siregar (2015:94))

2. *Activity Diagram* (Diagram Aktivitas)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau properti bisnis. Simbol-simbol yang digunakan dalam *Activity Diagram* yaitu:

Tabel II.4. Simbol Activity Diagram

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

(Sumber : Urva dan Siregar (2015:94))

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 Agregation, Atribut (Attributes), Operasi (Operation/Method), Visibility, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan multiplicity atau kardinaliti.

Tabel II.5. Multiple Class Diagram

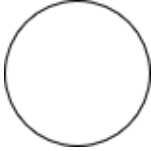
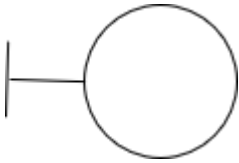
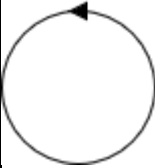
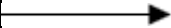
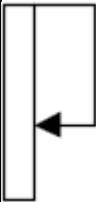
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau satu atau lebih
1..*	Satu atau lebih
0..1	Boleh tidak ada, maksimal satu
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Urva dan Siregar (2015:95))

4. *Sequence Diagram* (Diagram Urutan)

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 yaitu :

Tabel II.6. Simbol Sequence Diagram

Gambar	Nama	Keterangan
	<i>Entity Class</i>	<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>	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control Class</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>	<i>Message</i> , simbol mengirim pesan antar class.
	<i>Recursive</i>	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirimkan 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>	<i>Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.</i>
--	------------------------	---

(Sumber : Urva dan Siregar (2015:95))