

BAB II

LANDASAN TEORI

2.1 Lelang

Lelang atau Penjualan dimuka umum adalah suatu penjualan barang yang dilakukan didepan khalayak ramai dimana harga barang- barang yang ditawarkan kepada pembeli setiap saat semakin meningkat. Serangkaian kegiatan untuk menyediakan barang/jasa dengan cara menciptakan persaingan yang sehat diantara penyedia barang/jasa yang setara dan memenuhi syarat, berdasarkan metode dan tata cara tertentu yang telah ditetapkan dan diikuti oleh pihakpihak yang terkait secara taat asas sehingga terpilih penyedia yang baik.

Lelang adalah penjualan barang secara terbuka untuk umum dengan penawaran harga secara tertulis dan lisan yang semakin meningkat atau menurun untuk mencapai harga tertinggi yang didahului dengan pengumuman lelang. Lelang konvensional biasanya diadakan oleh sebuah perusahaan yang bergerak dibidang jual beli barang lelang dimana proses lelang diadakan di sebuah tempat sehingga konsumen atau penawar bisa berdatangan untuk menjadi peserta dari lelang tersebut. Di Indonesia, terdapat lembaga yang terkait dengan proses lelang, diantaranya Kantor Pelayanan Kekayaan Negara dan Lelang (KPKNL) dan balai lelang baik milik pemerintah maupun swasta. (Andre, 2017)

2.2 Marketplace

Marketplace adalah sebuah website atau aplikasi online yang memfasilitasi proses jual beli dari berbagai toko. Sebenarnya online marketplace memiliki konsep yang kurang lebih sama dengan pasar tradisional. Pada dasarnya, pemilik

marketplace tidak bertanggung jawab atas barang-barang yang dijual karena tugas mereka adalah menyediakan tempat bagi para penjual yang ingin berjualan dan membantu mereka untuk bertemu pelanggan dan melakukan transaksi dengan lebih simpel dan mudah. Transaksinya sendiri memang diatur oleh marketplaceny. Kemudian setelah menerima pembayaran, penjual akan mengirim barang ke pembeli. Salah satu alasan mengapa marketplace terkenal adalah karena kemudahan dan kenyamanan dalam penggunaan. Banyak yang menggambarkan online marketplace seperti department store.

Marketplace (dalam bahasa Indonesia, pasar) adalah tempat bertemunya pembeli dan penjual untuk melakukan sebuah transaksi. Online marketplace sendiri merupakan sebuah platform untuk mempertemukan pembeli dan penjual secara daring. Online marketplace merupakan jenis e-commerce yang bersifat C2C (consumer to consumer), dimana proses transaksi terjadi antara sesama individu yang memiliki akun di online marketplace tersebut. Setiap individu dapat bertindak sebagai penjual dan pembeli. Penyedia servis online marketplace dalam hal ini hanya bertindak sebagai perantara antar individu. (Andre, 2017)

2.3 Android

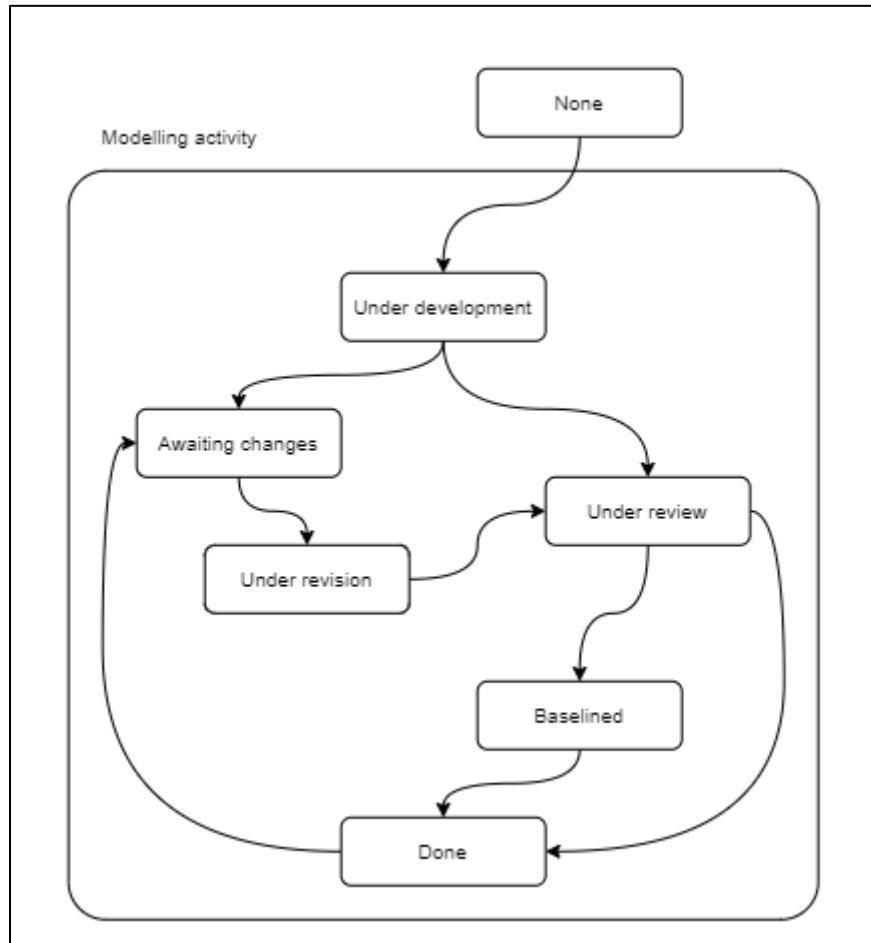
Android adalah sistem operasi yang khusus dirancang untuk smartphone dan tablet, Sistem Android ini memiliki basis Linux yang mana dijadikan sebagai pondasi dasar dari sistem operasi Android.

2.4 *Concurrent Development Model*

Menurut (Roger S. Pressman, 2005) dalam bukunya yang berjudul “Software Engineering: A Practitioner’s Approach The 6th Edition.McGraw-Hill”. Concurrent

Development Model (CDM) Adalah proses pengembangan yang pada dasarnya adalah rangkaian dari kegiatan teknis, tugas, dan hubungan saling terkait untuk merepresentasikan secara berulang-ulang dan element-element searah yang mengenai segala model proses yang concurrent.

Pada jurnal (Rijati & Nugraha, 2015), *concurrent development model* adalah pada model tersebut mendefinisikan serangkaian acara yang akan memicu adanya hubungan dari bagian ke bagian dalam kegiatan rekayasa perangkat lunak. Model concurrent sering digunakan sebagai paradigma untuk pengembangan client / server. Sebuah system client / server terdiri dari suatu komponen fungsional. Metode ini membentuk suatu skema model yang memperlihatkan suatu proses kerja pengembangan perangkat lunak yang dilakukan secara bersama dan tetap efektif dalam menyelesaikan masalah sesuai dengan kebutuhan yang ada. Sistem client / server terdiri atas satu set komponen yang fungsional, dengan menggunakan concurrent model nantinya proses kerja akan digambarkan menjadi aktivitas di dua dimensi yaitu dimensi system dan dimensi komponen.



Gambar 2.1 *Concurrent Development Modelling*

1. Dimensi Sistem; dimensi ini di tunjukan pada tiga aktifitas yaitu : desain, perakitan, dan penggunaan.
2. Dimensi Komponen; dimensi ditujukan dengan dua aktifitas yaitu : desain dan realisasi.

2.5 Correlation Based Similarity

Correlation-based similarity adalah sebuah teori yang dapat digunakan untuk menentukan bobot kemiripan antara 2 objek yang sedang dibandingkan dengan menghitung jarak kosinus vektor X_u dan X_y . Objek yang dimaksud dapat berupa pengguna atau *item*. Dalam kasus ini kemiripan yang dicari dengan *Correlation-*

based similarity adalah pengguna. Pengguna akan memberikan *rating* pada *item – item*. *Rating* dapat berupa bagus, sedang, buruk, atau suka, biasa saja, tidak suka dan sejenisnya. Kemiripan pengguna – pengguna dapat dihitung dengan menggunakan rumus berikut : (Ramadhani, Amalia, & Robbani, 2018)

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Keterangan :

n = banyaknya sampel

(x_i, y_i) = produk atau obyek yang dibandingkan

\bar{x} = rata-rata obyek 1

\bar{y} = rata-rata obyek 2

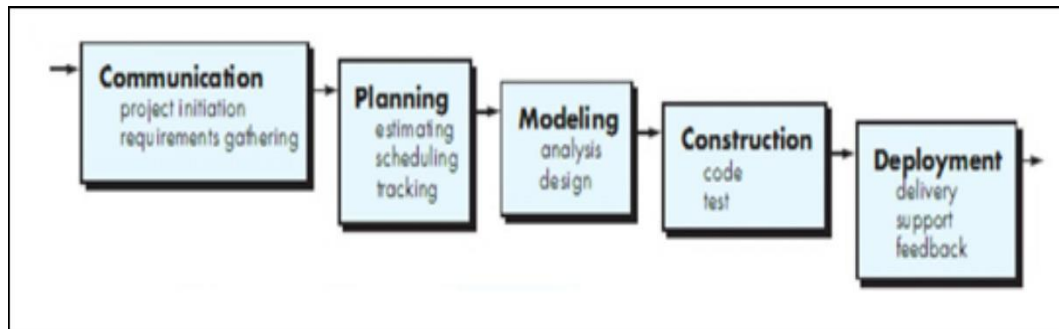
r = *rating*

2.6 Model Waterfall

Menurut Pressman (2015 : 42), metode *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan

waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Fase – fase dalam *Waterfall Model* menurut Pressman sebagai berikut :



Gambar 2.2 Ilustrasi Model *Waterfall* (Sumber : Pressman, (2015 : 42))

a. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel dan internet.

b. *Planning (Estimating, Scheduling, Tracking)*

Tahap berikutnya adalah tahapan *Planning* / perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan dan *tracking* proses pengerjaan sistem.

c. *Modeling (Analysis & Design)*

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface* dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

d. *Construction (Code & Test)*

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

e. *Deployment (Delivery, Support, Feedback)*

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software* dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015:17)

Metode *waterfall* dianggap pendekatan yang lebih cocok digunakan untuk proyek pembuatan sistem baru dan juga pengembangan *software* dengan tingkat resiko yang kecil serta waktu pengembangan yang cukup lama. Tetapi salah satu kelemahan paling mendasar adalah menyamakan pengembangan *hardware* dan *software* dengan meniadakan perubahan saat pengembangan. Padahal, *error* diketahui saat *software* dijalankan dan perubahan-perubahan akan sering terjadi.

Keuntungan menggunakan metode *waterfall* adalah prosesnya lebih terstruktur, hal ini membuat kualitas *software* baik dan tetap terjaga. Dari sisi *user* juga lebih menguntungkan, karena dapat merencanakan dan menyiapkan kebutuhan data dan proses yang diperlukan sejak awal. Penjadwalan juga menjadi lebih menentu, karena jadwal setiap proses dapat ditentukan secara pasti. Sehingga dapat dilihat jelas target penyelesaian pengembangan program. Dengan adanya urutan yang pasti, dapat dilihat pula perkembangan untuk setiap tahap secara pasti. Dari sisi lain, model ini merupakan jenis model yang bersifat dokumen lengkap sehingga proses pemeliharaan dapat dilakukan dengan mudah.

Kelemahan menggunakan metode *waterfall* adalah bersifat kaku, sehingga sulit melakukan perubahan di tengah proses. Jika terdapat kekurangan proses/prosedur dari tahap sebelumnya, maka tahapan pengembangan harus dilakukan mulai dari awal lagi. Hal ini akan memakan waktu yang lebih lama. Karena jika proses sebelumnya belum selesai sampai akhir, maka proses selanjutnya juga tidak dapat berjalan. Oleh karena itu, jika terdapat kekurangan dalam permintaan *user* maka proses pengembangan harus dimulai kembali dari awal. Karena itu, dapat dikatakan proses pengembangan *software* dengan metode *waterfall* bersifat lambat.

Kelemahan lainnya menggunakan metode *waterfall* adalah membutuhkan daftar kebutuhan yang lengkap sejak awal. Tetapi, biasanya jarang sekali *customer* yang dapat memenuhi itu. Untuk menghindari pengulangan tahap dari awal, *user* harus memberikan seluruh prosedur, data dan laporan yang diinginkan mulai dari tahap awal pengembangan. Tetapi pada banyak kondisi, *user* sering melakukan permintaan di tahap pertengahan pengembangan sistem. Dengan metode ini, maka

development harus dilakukan mulai lagi dari tahap awal. Karena *development* disesuaikan dengan desain hasil *user* pada saat tahap pengembangan awal. Di sisi lain, *user* tidak dapat mencoba sistem sebelum sistem benar-benar selesai. Selain itu, kinerja personil menjadi kurang optimal karena terdapat proses menunggu suatu tahap selesai terlebih dahulu. Oleh karena itu, seringkali diperlukan personil yang “*multi-skilled*” sehingga minimal dapat membantu pengerjaan untuk tahapan berikutnya. (Pressman, 2015:42-43)

2.7 Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan sistem arsitektur yang bekerja dalam *OOAD (Object-Oriented Analysis/Design)* dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*, dapat berupa model, *deskripsi*, atau *software*) yang terdapat dalam sistem *software*. (Fowler, 2015)

UML merupakan bahasa pemodelan yang paling sukses dari tiga metode OO yang telah ada sebelumnya, yaitu *Booch*, *OMT (Object Modeling Technique)*, dan *OOSE (Object-Oriented Software Engineering)*. (Munawar. 2015)

Dengan adanya *UML*, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. *UML* diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini. (Nugroho, 2013)


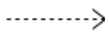
Tujuan *UML* diantaranya adalah:

1. Memberikan model yang siap pakai, bahasa pemodelan *visual* yang *ekspresif* untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai Bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan



a. Use Case Diagram

Use case diagram adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Berikut pengertian *use case diagram*. *Use Case Diagram* merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili *user* atau sistem lain yang berinteraksi dengan sistem yang dimodelkan. (Connolly, 2010)

Tabel 2. 1 **Tipe Relasi pada Use Case Diagram**

Simbol	Nama Elemen	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya

		elemen yang tidak mandiri (<i>independent</i>).
←	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
- - - - ->	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
⊲	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
—	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
□	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
○	<i>Use Case</i>	<i>Deskripsi</i> dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor


	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

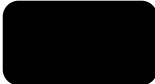



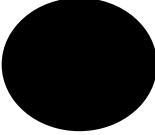
(Connolly, 2010)

b. *Activity Diagram*

Activity Diagram atau Diagram Aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram aktivitas mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah diagram aktivitas bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Sebuah *activity diagram* memiliki: (Connolly, 2010)

Tabel 2. 2 Tipe Relasi pada *Activity Diagram*

Simbol	Nama Element	Penjelasan
	<i>Initial State</i>	Mempresentasikan dimulainya alur kerja suatu sistem dalam <i>activity diagram</i> .

	<i>Action State</i>	Sebuah <i>state</i> yang menggambarkan eksekusi dari aksi <i>atomic</i> .
	<i>Transition Between Activities</i>	Mengidentifikasi bahwa suatu objek dari <i>state</i> pertama akan menampilkan aksi-aksi tertentu dan memasuki <i>state</i> kedua ketika peristiwa terjadi pergerakan dari aksi ke aksi lainnya.
	<i>Decision Point</i>	Menentukan kapan alur dalam aktivitas menjadi bercabang.
	<i>Fork</i>	Adanya percabangan paralel dari aktivitas
	<i>Final State</i>	Mempresentasikan bahwa telah diakhirinya alur suatu sistem dalam <i>activity diagram</i> .

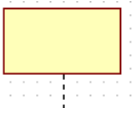

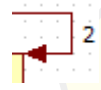

(Connolly, 2010)

c. *Sequence Diagram*

Merupakan diagram yang menunjukkan aliran fungsionalitas dalam *use case*. *Sequence* adalah satu dari dua interaksi diagram yang mengilustrasikan objek-objek yang berhubungan dengan *use case* dan *message* atau pesan-pesannya. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan

dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress *vertical*. (Connolly, 2010)

Tabel 2. 3 Tipe Relasi pada *Sequence Diagram*

Simbol	Nama Elemen	Keterangan
	Objek	Komponen yang menjadikan sebuah objek dalam membuat diagram
	<i>Stimulus</i>	Untuk menandakan hubungan komunikasi antar objek
	<i>Self Stimulus</i>	Fungsi sama dengan <i>Stimulus</i> , tetapi pesan yang disampaikan dikirimkan untuk objek itu sendiri
	<i>Focus Control</i>	Sebagai tempat untuk hasil <i>input</i> atau <i>output</i> dari sebuah proses yang dilakukan oleh objek ataupun aktor yang ada dalam sistem




(Connolly, 2010)

d. *Class Diagram*

Diagram kelas atau *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan *property*, operasi dan relasi yang sama. Disamping itu diagram kelas bisa memberikan pandangan global atas sebuah

sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan lainnya. Itulah sebabnya diagram kelas menjadi diagram yang paling populer di *UML*. (Connolly, 2010)

Tabel 2. 4 Tipe Relasi pada *Class Diagram*

Simbol	Nama Elemen	Keterangan
	Kelas	<ul style="list-style-type: none"> • Simbol untuk membangun sebuah pemrograman dengan objek • Terdiri 3 bagian, bagian atas adalah nama kelas, bagian tengah adalah atribut, dan bagian bawah adalah metode dari kelas tersebut
	Garis Asosiasi	<ul style="list-style-type: none"> • Simbol yang menggambarkan adanya hubungan antara satu kelas dengan kelas lainnya
	Generalisasi	<ul style="list-style-type: none"> • Simbol yang menandakan adanya generalisasi dari kelas input untuk menghasilkan data yang dibutuhkan

(Connolly, 2010)

Unified Modeling Language (UML) adalah bahasa pemodelan umum yang digunakan untuk melakukan spesifikasi, visualisasi, konstruksi dan dokumentasi artifak dari software system. UML bukanlah sebuah standar proses pengembangan dalam metode pengembangan sistem tertentu, namun pada umumnya UML dipakai dalam memodelkan sistem yang dibangun berbasiskan objek.

2.8 XAMPP

XAMPP merupakan perangkat lunak yang dikembangkan dari LAMP yang terdiri dari beberapa perangkat lunak seperti (*Linux, Apache, MySQL, PHP, dan PERL*) sebagai *project non profit* yang dikembangkan oleh *Apache Friends*. *Apache Friends* sendiri terdiri dari Tim Inti (*Core Team*), Tim Pengembang (*Development Team*) dan Tim Dukungan (*Support Tim*) yang didirikan Kai Oswald Seidler dan Kay Vogelgesang pada tahun 2002. Project ini berguna untuk mempromosikan penggunaan *Apache web browser*.

XAMPP merupakan singkatan dari masing-masing huruf yang memiliki makna tersendiri, makna dari masing-masing huruf tersebut adalah:

1) **X**

X, sebagai program yang paling banyak dijalankan pada sistem operasi, baik pada *windows, Linux, Mac, OS* ataupun *Solaris*.

2) **A**

Apache, sebagai aplikasi *web server*, yang mana bertugas dalam menghasilkan halaman *web* yang benar kepada *user* melalui kode PHP yang telah dituliskan pada halaman *web*.

3) **M**

MySQL, merupakan aplikasi *database server* yang dapat berguna untuk menambahkan, menghapus, dan mengubah data yang ada pada *database*. SQL kepanjangan dari *Structured Query Language* yaitu sebagai bahasa yang terstruktur dalam mengolah *database*.

4) **P**

PHP, merupakan bahasa pemrograman yang biasa digunakan untuk membuat *web* yang bersifat *server side*.

5) **P**

Perl, yaitu bahasa pemrograman.

Jadi kalau digabung XAMPP itu singkatan dari X, *Apache*, MySQL, PHP dan *Perl*. XAMPP tersedia untuk sistem operasi *Microsoft Windows*, *Linux*, *Sun Solaris* dan *Mac OS*.

XAMPP adalah paket program *web* lengkap yang dapat Anda pakai untuk belajar pemrograman *web*, khususnya PHP dan MySQL. (Nugroho, 2013)

XAMPP adalah perangkat lunak *opensource* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi seperti *windows*, *linux*, *solaris*, dan *mac*. (Buana, 2014)

2.9 MySQL

MySQL diciptakan di Negara Swedia oleh perusahaan MySQL AB adapun masing-masing nama yang berjasa dalam menciptakan MySQL adalah David Axmark, Allan Lason, dan Michael “*monty*” Widenius. Perangkat lunak ini tersebar luas secara garis besar karena memiliki lisensi *GNU General Public License*. Sampai sekarang, tercatat ada beberapa bahasa pemrograman yang populer

yang bisa bersinkronisasi dengan MySQL, seperti C<, C++, C#, bahasa pemrograman eiffel, bahasa pemrograman *Smaltalk*, bahasa pemrograman *java*, bahasa pemrograman *lisp*, *perl*, PHP, bahasa pemrograman *python*, *ruby*, *REALbasic*, dan Tcl. (Adelheid, 2012).

MySQL adalah *Relational Database Management (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *General Public License (GPL)*. Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. Saat ini *MySQL* merupakan *software database* yang memegang kendali terbesar pada aplikasi *database* berbasis *web*. Arsitektur *databasenya* sangat cocok bekerja pada lingkungan tersebut, didukung kemampuannya dalam berinteraksi dengan berbagai bahasa *programmeran side-server* seperti *PHP*, *ASP*, *CGI* dan *JSP* (Prasetyo, 2003).

MySQL Merupakan *database server* yang paling sering digunakan dalam pemrograman PHP. MySQL digunakan untuk menyimpan data dalam *database* dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam *database*. (Buana, 2014)

Berdasarkan pengertian yang ada diatas dapat penulis simpulkan bahwa pengertian *MySQL* adalah sebuah *server* basis data yang digunakan untuk membangun basis data pada aplikasi-aplikasi yang didukung oleh basis data.

2.10 PHPMysqlAdmin

Menurut Rozaq. A, Lestari (2015), PHPMysqlAdmin adalah perangkat lunak yang bebas ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MYSQL melalui Jejaring jagat Jembar (World Wide Web). PHPMysqlAdmin mendukung berbagai operasi MySQL, diantaranya (mengolah basis data, tabel-tabel, bidang (fields), relasi (relation), indeks, pengguna (users), perjanjian (permissions), dan lain-lain.

Perbedaan PHPMysqlAdmin dengan MySQL terletak pada fungsi. PhpMyAdmin merupakan alat untuk memudahkan dalam mengoperasikan database MySQL, sedangkan MySQL adalah database tempat penyimpanan data. Phpmysqladmin sendiri digunakan sebagai alat untuk mengolah/ mengatur data pada MySQL. (Rahmawati, 2017)

2.11 PHP

PHP adalah bahasa pemrograman *script server-side* yang di desain untuk pengembangan *web*. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group.

“PHP (Hypertext Preprocessor) adalah bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis”. Dengan menggunakan program PHP, sebuah website akan lebih interaktif dan *dinamis*.

Adapun kelebihan-kelebihan dari PHP yaitu:

1. Banyaknya *web server* yang mendukung bahasa pemrograman PHP sehingga konfigurasinya semakin mudah. Beberapa *web server* misalnya Apache, IIS, Xitami, dan Lain-Lain.

2. Pengembangan bahasa pemrograman PHP tergolong lebih mudah, karena banyak *developer* yang membantu mengembangkannya maupun menggunakannya.
3. *Relatif* mudah untuk dipahami, karena sekarang ini banyak sekali tersebar materi-materi/*referensi* untuk mempelajari PHP.
4. Bahasa pemrograman PHP juga dapat di sisipkan kedalam HTML.
5. Cocok digunakan untuk pemrograman *web* dinamis, walau bisa juga untuk membuat program komputer lainnya.
6. PHP merupakan bahasa pemrograman bersifat *Open Source*, sehingga dapat di gunakan di banyak sistem operasi komputer dan tentunya gratis.

2.12 Internet

Jaringan antar komputer yang saling berkaitan secara terus menerus baik melalui *email*, *transmisi file*, dan komunikasi dua arah antar individu atau kelompok. *Internet* atau *international network* merupakan rangkaian jaringan terbesar didunia dimana semua jaringan yang berada pada semua organisasi dihubungkan dengan suatu jaringan terbesar sehingga dapat saling berkomunikasi. (Mulyanto, 2009)

Menurut (Sibero, 2013), *Internet (interconnected network)* adalah jaringan komputer yang menghubungkan antar jaringan secara *global*, *internet* dapat juga disebut jaringan dalam suatu jaringan yang luas.

Sedangkan Menurut (Afrianto, 2011), *Internet* merupakan kependekan dari kata "*Internetwork*", yang berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan. Berdasarkan pendapat yang dikemukakan dapat

disimpulkan bahwa, *Internet* adalah suatu rangkaian atau jaringan komputer yang menghubungkan jaringan komputer dalam suatu jaringan yang luas.

2.13 Penelitian Terkait

Penelitian yang dilakukan oleh Bondan Prasetyo (2018) dengan judul “Implementasi Metode *Item-Based Collaborative Filtering* dalam Pemberian Rekomendasi Calon Pembeli Aksesoris *Smartphone*”. Flazzstore merupakan sebuah toko yang bergerak dibidang penjualan casing smartphone. 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 memilih produk yang akan dibelinya. Penelitian ini menggunakan metode *Item-Based Collaborative Filtering*, metode ini mencari similarity/kesamaan item dengan item lainnya. Sistem akan mencari rating tiap item dan menghitung nilai similarity menggunakan persamaan *pearson correlation-based similarity*. Kemudian nilai dari hasil perhitungan similarity akan digunakan untuk menghitung nilai prediksi tiap produk dengan menggunakan persamaan *weighted average of deviation*. Sebelum direkomendasikan kepada pelanggan dari hasil prediksi tersebut dihitung nilai *Mean Absolute Error (MAE)* dihitung selisih antara nilai rating sebenarnya dengan prediksi, dan kemudian diurutkan mulai dari terkecil ke terbesar untuk direkomendasikan kepada user. Hasil dari penelitian menunjukkan kecilnya nilai rata-rata MAE 0,572039 namun untuk proses eksekusi, waktu yang dibutuhkan cukup lama yaitu 6,4 detik.

Penelitian yang dilakukan oleh Ramadhanuz A Djamal (2010) dengan judul “Analisis Dan Implementasi Metode Item-Based Clustering Hybrid Pada Recommender System”. Sistem rekomendasi adalah aplikasi untuk mencari dan merekomendasikan item dengan memprediksi peringkat berdasarkan kesamaan informasi karakteristik pengguna. Metode Pengelompokan Hibrid Berbasis Item (ICHM) adalah salah satu sistem rekomendasi hibrid yang menggabungkan pemfilteran kolaboratif dan pemfilteran berbasis konten. Tujuan kombinasi antara pemfilteran berbasis konten dan pemfilteran kolaboratif di ICHM adalah untuk mengatasi setiap kekurangan metode penyaringan. Sistem rekomendasi ICHM memiliki keunggulan lain sehingga dapat memprediksi item baru yang tidak memiliki peringkat sama sekali. Makalah ini menjelaskan tentang implementasi dan analisis hasil ICHM yang diterapkan pada data rekomendasi film MovieLens.org. Pelaksanaannya dilakukan berdasarkan kajian literatur yang dilakukan. Analisis dilakukan dengan membandingkan Means Absolute Error (MAE) di bawah beberapa skenario pengujian. Analisis dilakukan untuk dua jenis kasus, yang merupakan masalah awal yang dingin dan masalah awal yang tidak dingin. Semakin tinggi jumlah kluster, semakin rendah MAE. c koefisien hanya mempengaruhi MAE pada masalah awal non-dingin.