BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau *variabel* yang terorganisir dan saling berkaitan satu sama lain.

2.1.1 Pengertian Sistem

Sistem merupakan sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan (Kadir, 2014).

Dalam sebuah sistem setiap elemen atau komponen harus saling memberikan manfaat demi tercapainya tujuan dari sistem itu sendiri. Jika dalam sebuah sistem terdapat komponen atau elemen yang tidak memberikan manfaat dalam mencapai tujuan, maka elemen atau komponen tersebut bukan bagian dari sistem (Fendi Hidayat, 2019).

2.1.2 Klasifikasi Sistem

Klasifikasi sistem adalah suatu bentuk kesatuan antara satu komponen dengan komponen lainnya, karena tujuan dari sistem tersebut memiliki akhir tujuan yang berbeda untuk setiap perkara atau kasus yang terjadi dalam setiap sistem tersebut. Sehingga, sistem tersebut dapat diklasifikasikan sebagai berikut (Fendi Hidayat, 2019) :

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem theologia, yaitu sistem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan. Sedangkan sistem fisik merupakan sistem yang ada secara fisik, mislanya sistem komputer, sistem produksi dan lain-lain.

2. Sistem Alamiah dan Sistem Buatan

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang dan malam, pergnatian cuaca, dan sebagainya. Sedangkan sistem buatan merupakan sistem yang melibatkan interaksi manusia dengan mesin, yang disebut human manchine system.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah salah satu contoh sistem yang tingkah lakunya dapat diprediksi berdasarkan program komputer yang dijalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena terdapat unsur probabilistik.

4. Sistem Terbuka dan Sistem Tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

2.2 Konsep Sistem Informasi

Sistem informasi dipahami sebagai kumpulan atau suatu himpunan dari kelompok orang-orang yang bekerja, prosedur-prosedur, dan sumber daya peralatan yang mengumpulkan data dan mengolahnya menjadi sebuah informasi, merawat, dan menyebarkan informasi tersebut dalam suatu organisasi atau perusahaan.

Sistem informasi adalah sebuah alat atau sarana yang bertujuan untuk mengolah data menjadi informasi, yang dapat dimanfaatkan oleh pengambil keputusan. Sistem informasi juga dapat diartikan sebagai sebuah media untuk membagikan dan menyebarluaskan informasi kepada pengguna informasi secara cepat dan tepat (Fendi Hidayat, 2019).

2.3 Sistem Rekomendasi

Sistem rekomendasi merupakan suatu aplikasi untuk menyediakan dan merekomendasikan suatu item dalam membuat suatu keputusan yang diinginkan oleh pengguna (Ungkawa, et al., 2013). Penerapan rekomendasi didalam sebuah sistem biasanya melakukan prediksi suatu *item*, seperti rekomendasi film, musik, buku, berita dan lain sebagainya yang menarik *user*. Sistem ini berjalan dengan mengumpulkan data dari *user* secara langsung maupun tidak (Fadlil & Mahmudy, 2010).

Pengumpulkan data secara langsung dapat dilakukan sebagai berikut :

1. Meminta *user* untuk melakukan *rating* pada sebuah *item*.
2. Meminta *user* untuk melakukan rangking pada *item* favorit setidaknya memilih satu *item* favorit.
3. Memberikan beberapa pilihan *item* pada *user* dan memintanya memilih yang terbaik.
4. Meminta *user* untuk mendaftar *item* yang paling disukai atau *item* yang tidak disukainya. Pengumpulan data dengan tidak langsung berhubungan dengan seorang *user*, dilakukan dengan cara mengamati *item* yang dilihat oleh seorang *user*.

Dari data hasil yang dikumpulkan tersebut, kemudian diolah dengan menggunakan algoritma tertentu.

2.4 Data Mining

Data Mining merupakan proses penggalian informasi dan pola yang bermanfaat dari data yang sangat besar. *Data Mining* mencakup pengumpulan data, ekstraksi data, analisis data, dan statistik data. *Data Mining* juga dikenal sebagai *Knowledge Discovery, Knowledge Extraction, data/pattern analysis, Information harvesting,* dan lain-lain (Muh Nasir, 2020).

Data mining juga merupakan proses logis untuk menemukan informasi yang berguna. Setelah ditemukan informasi dan pola dapat digunakan untuk alat pendukung dalam

pengambilan keputusan dalam mengembangkan bisnis. *Data mining* bertujuan untuk menemukan pola yang sebelumnya tidak diketahui. Jika pola tersebut telah diperoleh maka dapat digunakan untuk menyelesaikan berbagai macam permasalahan (Muh Nasir, 2020).

2.5 Algoritma Apriori

Algoritma Apriori adalah algoritma klasik dalam data mining. Algoritma ini digunakan untuk melihat intensitas kemunculan *itemset* atau *frequent itemset* dan aturan asosiasi yang relevan (Muh Nasir, 2020). Algoritma Apriori diperkenalkan pertama kali oleh R. Agrawal dan R. Srikant pada tahun 1994 untuk mencari *item item* yang sering keluar dalam *dataset* untuk aturan asosiasi boolean. Penamaan ini karena menggunakan pengetahuan sebelumnya tentang sifat-sifat *itemset* yang sering diambil dalam transaksi.

Dua tolok ukur penting tidaknya asosiasi adalah *support* dan *confidence*. *Support* adalah nilai penunjang sedangkan *confidence* adalah nilai kepastian. Untuk memperoleh ketentuan asosiatif dibutuhkan pencarian ketentuan yang mempunyai pola frekuensi besar (PFT). PFT dicari dengan cara mencari ketentuan yang penuhi nilai *support minimum* (Iswandi et al., 2020). Nilai *support* (penunjang) merupakan persentase *item* ataupun campuran *item* yang terdapat pada totalitas informasi. Adapun dua tolok ukur dalam membentuk *rules* atau aturan dalam penerapan algoritma apriori adalah sebagai berikut:

1. *Support*

Support atau bisa juga disebut nilai penunjang adalah persentase dari laporan atau *record* yang didalamnya mengandung kombinasi *item*. Persamaan (1) adalah rumus untuk mendapatkan nilai *support*.

$$\textbf{Support}\ (\mathbf{A}) = \frac{\textbf{Jumlah Transaksi Mengandung A}}{\textbf{Total Transaksi}}$$

Persamaan (2) adalah rumus untuk mendapatkan nilai *support* dari suatu kombinasi *item.*

$$\textbf{Support}\ (\mathbf{A}, \mathbf{B}) = \frac{\textbf{Jumlah Transaksi Mengandung A dan B}}{\textbf{Total Transaksi}}$$

2. *Confidence*

Confidence atau biasa disebut nilai kepastian adalah kuatnya hubungan antar *item* dalam aturan asosiasi. Adapun rumus untuk mendapatkan nilai *confidence* ialah:

$$\textbf{Confidence}\ (\mathbf{A}, \mathbf{B}) = \frac{\textbf{Jumlah Transaksi Mengandung A dan B}}{\textbf{Total Transaksi Mengandung A}}$$

Atau,

$$\textbf{Confidence}\ (\mathbf{A} => \mathbf{B}) = \frac{\textbf{Support}(\mathbf{A}, \mathbf{B})}{\textbf{Support}(\mathbf{A})}$$

Sedangkan rumus mendapatkan nilai persentase *confidence* adalah :

$$\textbf{Confidence}\ (\mathbf{A} => \mathbf{B}) = \frac{\textbf{Support}(\mathbf{A}, \mathbf{B})}{\textbf{Support}(\mathbf{A})}\ \textbf{x 100\%}$$

Ada dua proses yang cukup penting pada algoritma apriori ialah:

1. *Join* (Penggabungan)

Pada proses ini satu *item* dikombinasikan dengan *item* lain sampai tidak ada lagi kombinasi yang bisa terbentuk.

2. *Pruning* (Pemangkasan)

Pada proses ini dilakukan pemangkasan terhadap kombinasi sesuai dengan *minimum support* yang sebelumnya telah ditetapkan.

Langkah-langkah pada proses algoritma apriori adalah sebagai berikut:

1. Pertama *scan database* guna menemukan kandidat 1-*itemset* (C1) dan juga menghitung nilai *support*-nya. Setelah itu bandingkan antara nilai *support* dengan *minimum support* yang sebelumnya telah ditentukan, apabila nilai *support* lebih besar atau nilainya sama dengan *minimum support*, *itemset* terhitung dalam *large-itemset set* 1 (L1).

2. *Itemset* yang tidak terhitung dalam *large-itemset* tidak dipakai untuk melakukan iterasi berikutnya. (Proses *pruning*).

3. *Large-itemset set* 1 (L1) digunakan untuk proses iterasi yang berikutnya.

Pada *large-itemset set* 1 (L1) dilakukan proses *join* pada dirinya sendiri untuk menghasilkan kandidat 2-*itemset* (C2). Setelah itu bandingkan nilai *support* dari semua *item* yang ada pada C2 dengan *minimum support*, jika nilainya lebih atau sama dengan *minimum support* maka akan masuk kedalam *large-itemset* L2. Ulangi langkah yang sama seperti mencari *large-itemset* yang sebelumnya.

4. Pembentukan kandidat (*joining*) dan pembentukan *large-itemset* (*Pruning*) dilakukan secara terus-menerus sampai tidak ada lagi kandidat yang bisa terbentuk.

5. Langkah selanjutnya yaitu untuk semua *large-itemset* yang terbentuk atau memenuhi nilai *minimum support* akan dibentuk *association rule* setelah itu dicari juga nilai *confidence*-nya. Nantinya seluruh aturan yang terbentuk jika nilai *confiedence*-nya kurang dari nilai minimum *confidence* yang ditetapkan, maka aturan tersebut tidak akan dipakai atau tidak termasuk dalam *association rule* yang dipakai

2.6 Unified Modelling Language (UML)

Unified Modelling Language adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil Analisa dan desain yang berisi sintaks dalam memodelkan sistem secara visual (Haviluddin, 2011). UML digunakan untuk menentukan atau menggambarkan sebuah sistem yang terkait dengan objek.

Penggunaan UML bertujuan untuk memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa. Serta memberikan model yang siap pakai. UML memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya : (Rosa & Shalahuddin, 2014).

2.6.1 Use Case Diagram

Diagram use case memperlihatkan himpunan use case dan aktor. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan

serta diharapkan pengguna. Komponen pembentuk diagram *use case* adalah : (Widodo & Herlawati, 2011)

1. Aktor, menggambarkan pihak yang berperan dalam sistem.
2. *Use Case*, aktivitas yang disiapkan oleh bisnis/sistem.
3. Hubungan (Link), aktor mana saja yang terlibat dalam *use case*.

2.6.2 Activity Diagram

Activity diagram adalah tipe khusus dari diagram *state* yang memperlihatkan aliran dari suatu aktifitas ke aktifitas yang lain dari suatu sistem. *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa & M.Shalahuddin, 2014).

2.7 Website

Website adalah kumpulan dokumen berupa halaman web yang berisi teks dalam format *hypertext markup language* (HTML). Website disimpan di server *hosting* yang dapat diakses menggunakan *browser* dengan jaringan internet melalui alamat internet berupa *Uniform Resource Locator* atau URL (Dewa Made & Salnan Ratih, 2021).

Website terdiri dari dua jenis yaitu *website* statis dan *website* dinamis. *Website* statis adalah website yang apabila ingin merubah konten di dalamnya harus diedit secara manual artinya harus mengubah *source code*. Biasanya halaman website yang statis masih menggunakan tag HTML dan data masih belum tersimpan dalam *database*. Sedangkan *website* dinamis, apabila ingin mengubah konten yang terdapat dalam website tersebut dapat dilakukan dengan mudah tanpa harus membuka *source code* dan dapat diperbarui secara berkala. Hal ini disebabkan konten *website* disimpan di *database*. Halaman *website* yang dinamis biasanya ditulis dalam bahasa pemrograman *server side* seperti PHP, ASP, JSP dan lainnya (Dewa Made & Salnan Ratih, 2021).

2.8 Hypertext Preprocessor (PHP)

PHP atau *Hypertext Preprocessor* adalah bahasa berbentuk skrip yang ditempatkan dalam server dan diproses didalam server dan hasilnya akan dikirimkan ke user dalam bentuk halaman web yang diakses menggunakan browser (Dewa Made & Salnan Ratih, 2021). PHP juga merupakan *script* yang digunakan untuk membuat halaman *website* yang sangat dinamis, dinamis berarti halaman tampilan yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*.

Bahasa pemrograman PHP juga sering digunakan karena PHP adalah bahasa *open source* yang memiliki kesederhanaan dan memiliki beberapa fitur *built-in* yang berfungsi untuk menangani kebutuhan standart dalam pembuatan aplikasi web. PHP juga merupakan bahasa *script* yang paling mudah dipahami karena memiliki beberapa referensi. PHP dapat digunakan untuk berbagai sistem operasi antara lain : Unix, Macintosh, dan Windows. PHP dapat dijalankan secara rutin melalui console serta dapat menjalankan perintah sistem

2.9 MySQL

MySQL merupakan salah satu *software* untuk *database server* yang banyak digunakan dan bersifat *open source* (Heni A. Puspitosari, 2011). Pada umumnya perintah yang sering digunakan dalam MySQL adalah *select, insert, update,* dan *delete.* SQL juga menyediakan perintah untuk membuat *database, field,* atau untuk menambah data. MySQL memiliki beberapa kelebihan, yaitu :

1. Bersifat *open source,* memiliki kemampuan untuk dikembangkan lagi.
2. Mudah dipelajari, dan *multiuser,* artinya dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan.
3. Super *performance* yang *reliable,* tidak dapat diragukan, pemrosesan *database*nya sangat cepat dan stabil.

Jadi dapat disimpulkan MySQL adalah suatu *database server* yang membantu pengolahan basis data dengan sangat cepat menggunakan SQL *Language.*

2.10 Twitter Bootstrap

Bootstrap adalah *framework front-end* yang intuitif dan powerful untuk pengembangan aplikasi web yang lebih cepat dan mudah. Bootstrap menggunakan HTML, CSS, dan JavaScript. Bootstrap memiliki fitur-fitur komponen *interface* seperti Typography, Forms, Buttons, Tables, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel dan lainnya. Bootstrap dapat membantu membuat layout situs yang responsif dan modern dengan mudah (Jubilee Enterprise, 2016)

Bootstrap diciptakan oleh dua orang programmer twitter, yakni Mark Otto dan Jacob Thornton pada tahun 2011. Pada tahun tersebut, mereka tergerak untuk menciptakan *framework* yang dapat digunakan di lingkungan internal Twitter. Oleh karena itu, walau nama resminya Bootstrap, namun dikenal sebagai Twitter Bootstrap di kalangan developer. Twitter Bootstrap memiliki beberapa keuntungan, diantaranya : (Dzikri MS, 2020).

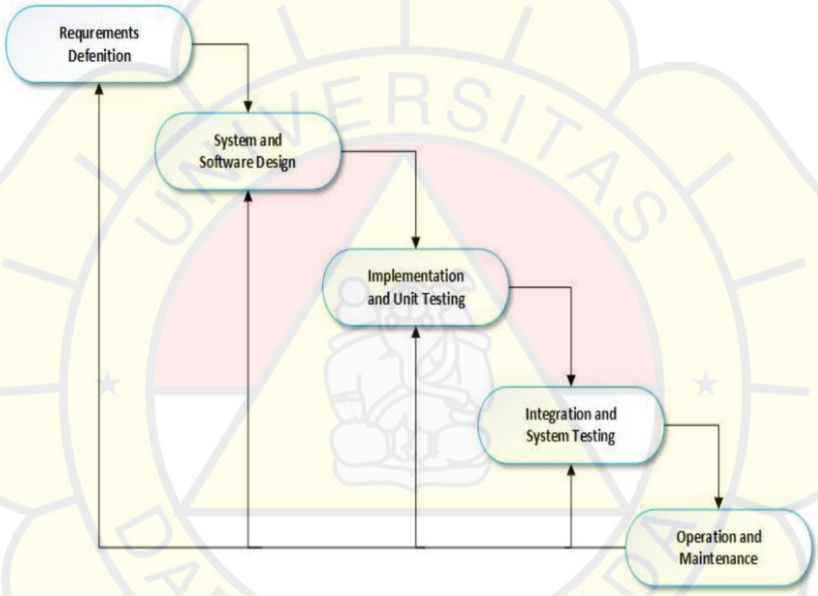
1. Bisa mempercepat waktu untuk pengembang memproses pembuatan tampilan *front-end* sebuah website karena tidak harus mengetik *syntax coding* CSS dari awal.
2. Menampilkan sisi *website* yang lebih modern.
3. Tampilan sangat *responsive*, sehingga mendukung untuk berbagai *resolusi*, mulai dari tablet, smartphone, hingga PC dan Laptop.
4. *Website* yang menggunakan bootstrap pada umumnya lebih ringan karena memiliki *library* CSS yang terstruktur.

2.11 Metode Waterfall

Metode *waterfall* atau yang sering disebut siklus hidup klasik (*Classic Life Cycle*), dimana metode ini menggambarkan pendekatan yang sistematis dan juga berurutan pada

pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan, permodelan, konstruksi, dan penyerahan sistem ke para pelanggan yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012).

Dalam pengembangannya, metode *waterfall* memiliki beberapa tahapan yang berurut yaitu :R*equirement Analisis, System Design, Implementation and Unit Testing, Integration and testing, Operation and Maintenance*. Berikut ini merupakan tahapan metode *waterfall* :



Gambar 2. 1 Tahapan Metode Waterfall

(Sumber : Muharto. 2016. Metode Penelitian Sistem Informasi; Mengatasi Kesulitan Mahasiswa dalam Menyusun Proposal Penelitian. Yogyakarta: Deepublish.)

1. *Requirement Analisis*, merupakan tahap dimana pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Pada tahap ini dilakukan dengan wawancara dan observasi untuk mendapatkan informasi yang akan dianalisis.

2. *System Design*, merupakan spesifikasi kebutuhan dari tahap sebelumnya. Dalam tahap ini akan dipelajari dan desain sistem dipersiapkan. Desain sistem membantu dalam menentukan perangkat keras dan sistem persyaratan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*, pada tahap ini sistem pertama kali dikembangkan pada program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

4. *Integration & Testing*, seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan atau kesalahan.

5. *Operation & Maintenance*, merupakan tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

Metode *waterfall* memiliki kelebihan seperti memungkinkan untuk departementalisasi dan kontrol. Proses pengembangan model fase *one by one*, sehingga meminimalisir kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir pada operasi dan pemeliharaan. Sedangkan kekurangannya, metode ini tidak memungkinkan untuk banyak revisi jika terjadi kesalahan dalam prosesnya. Karena setelah aplikasi ini dalam tahap pengujian, sulit untuk kembali untuk mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya. (Teduh Sanubari, 2020).