

BAB II

LANDASAN TEORI

2.1 Sistem Prediksi

2.1.1 Pengertian Sistem

Menurut (Tsany, Mulyawan, & Sutrisno, 2018) pengertian sistem yang dikemukakan oleh (Kristanto, 2003) adalah kata sistem berasal dari bahasa Latin *systema* dan bahasa Yunani *sistēma*. Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memperoleh masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan.

Sedangkan menurut (Romney dan Steinbart, 2015) sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari sub sistem yang lebih kecil yang mendukung sistem yang lebih besar.

Pada prinsipnya, setiap sistem selalu terdiri dari empat elemen:

1. Objek, yang dapat berupa bagian, elemen, ataupun variabel.
2. Atribut, yang menentukan kualitas atau sifat kepemilikan sistem dan objeknya.
3. Hubungan internal, diantara objek-objek di dalamnya.
4. Lingkungan, tempat dimana sistem berada.

2.1.2 Pengertian Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil.

Menurut (Utama & Watequlis, 2016) *Forecasting* bertujuan untuk mendapatkan prediksi atau prediksi yang bisa meminimumkan kesalahan dalam meramal yang biasanya diukur dengan mean square error, mean absolute *error*.

Pengertian Prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu.

Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan. Prediksi bisa berdasarkan metode ilmiah ataupun subjektif belaka. Ambil contoh, prediksi cuaca selalu berdasarkan data dan informasi terbaru yang didasarkan pengamatan termasuk oleh satelit.

2.2 Sistem Pendukung Keputusan

Menurut (Andika, Winata, & Ginting, 2019) Sistem Pendukung Keputusan yang dikemukakan oleh (Kusrini, 2007) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan manipulasi data. Sistem ini digunakan untuk pengambilan keputusan dalam situasi yang semiterstruktur dan tidak terstruktur.

SPK bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik.

SPK merupakan implementasi teori-teori pengambilan keputusan yang telah diperkenalkan oleh ilmu-ilmu seperti *operation research* dan *management science*, hanya bedanya adalah bahwa jika dahulu untuk mencari penyelesaian masalah yang dihadapi harus dilakukan perhitungan iterasi secara manual (biasanya untuk mencari nilai minimum, maksimum, atau optimum), saat ini komputer PC telah menawarkan kemampuannya untuk menyelesaikan persoalan yang sama dalam waktu relatif singkat.

2.3 Inventory

Persediaan barang (*inventory*) adalah istilah yang dipakai untuk menunjukkan barang – barang yang dimiliki untuk dijual kembali atau digunakan untuk diproses menjadi barang – barang yang akan dijual. Pengelolaan persediaan merupakan elemen penting dalam perusahaan. Manajemen persediaan yang efektif merupakan kunci keberhasilan dalam operasional perusahaan, khususnya dalam perusahaan dagang dan perusahaan manufaktur (pabrikasi). Manajemen persediaan yang efektif akan berusaha untuk mempertahankan kuantitas dan jenis persediaan yang cukup untuk memenuhi permintaan pelanggan, di sisi lain juga harus menghindari biaya penyimpanan sebagai akibat penentuan kebijakan persediaan dalam jumlah besar.

Menurut (Sulaiman, Hartono, & Raharjo, 2017) Persediaan (*inventory*) yang dikemukakan oleh (Haming & Nurnajamuddin, 2007) adalah sumber daya

ekonomi fisik yang perlu diadakan dan dipelihara untuk menunjang kelancaran produksi meliputi bahan baku (*raw material*), produk jadi (*finish product*), komponen rakitan (*component*), bahan pembantu (*substance material*) dan bahan dalam proses pengerjaan.

2.4 Algoritma Apriori

Algoritma apriori termasuk jenis aturan asosiasi pada data mining. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut affinity analysis atau market basket analysis.

Menurut (Badrul, 2016) analisis asosiasi atau *association rule mining* yang dikemukakan oleh (Larose, 2005) adalah teknik data mining untuk menemukan aturan suatu kombinasi *item*.

Analisis asosiasi atau *association rule mining* adalah teknik data mining untuk menemukan aturan asosiatif antara suatu kombinasi item. Analisis asosiasi dikenal juga sebagai salah satu teknik data mining yang menjadi dasar dari berbagai teknik data mining lainnya. Secara khusus, salah satu tahap analisis asosiasi yang menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien adalah analisis pola frekuensi tinggi (*frequent pattern mining*).

Penting tidaknya aturan asosiatif dapat diketahui dengan dua parameter, yaitu *support* dan *confidence*. *Support* (nilai penunjang) adalah persentase kombinasi item tersebut dalam database, sedangkan *confidence* (nilai kepastian) adalah kuatnya hubungan antar-item dalam aturan asosiasi.

Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan asosiasi yang memenuhi syarat minimum untuk support (*minimum support*) dan syarat minimum untuk *confidence* (*minimum confidence*).

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

1. Analisis Pola Frekuensi Tinggi dengan Algoritma Apriori

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai support dalam basis data. Nilai support sebuah item diperoleh dengan menggunakan rumus berikut :

$$\text{Support} = \frac{\text{Jumlah Transaksi mengandung A}}{\text{Total Transaksi}} \times 100\%$$

Sementara, nilai support dari 2 item diperoleh dengan menggunakan rumus :

$$\text{Support } A \cap B = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}} \times 100\%$$

Frequent itemset menunjukkan itemset yang memiliki frekuensi kemunculan lebih dari nilai minimum yang ditentukan (\varnothing). Misalkan $\varnothing = 2$, maka semua itemsets yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut frequent. Himpunan dari frequent k-itemset dilambangkan dengan F_k .

2. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan

menghitung *confidence* aturan asosiatif $A \rightarrow B$. Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh dengan rumus berikut :

$$Confidence = P(A | B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Jumlah Transaksi Mengandung A}} \times 100\%$$

Untuk menentukan aturan asosiasi yang akan dipilih maka harus diurutkan berdasarkan $Support \times Confidence$. Aturan diambil sebanyak “n” aturan yang memiliki hasil terbesar.

2.5 Metode *Moving Average*

Metode *time series* terdiri dari beberapa metode, salah satunya adalah *moving average* atau rata-rata bergerak. Metode *moving average* digunakan jika data masa lalu merupakan data yang tidak memiliki unsur trend atau faktor musiman. *Moving average* banyak digunakan untuk menentukan trend dari suatu deret waktu. Tujuan utama dari penggunaan rata-rata bergerak adalah untuk menghilangkan atau mengurangi acakan (*randomness*) dalam deret waktu. Tujuan ini dapat dicapai dengan merata-ratakan beberapa nilai data bersama-sama, dengan cara mana kesalahan-kesalahan positif dan negatif yang mungkin terjadi dapat dikeluarkan atau dihilangkan. Untuk mendapatkan nilai dari *moving average* sebelumnya ditentukan terlebih dahulu jumlah periode (T). Setelah ditentukan jumlah periode yang akan digunakan dalam observasi pada setiap rata-rata atau MA(T) dapat dihitung nilai rata-ratanya.

Hasil dari nilai rata-rata bergerak tersebut kemudian akan menjadi ramalan untuk periode mendatang. *Moving average* tidak menggunakan data yang terdahulu terus-menerus, setiap ada data yang baru, data baru tersebut digunakan

dan tidak lagi menggunakan nilai observasi yang paling lama, dikarenakan penggunaan jumlah periode selalu konstan.

Rumus *Moving average* atau Rata-rata Bergerak adalah sebagai berikut :

$$MA = \Sigma X / \text{Jumlah Periode}$$

Keterangan :

$MA = \text{Moving average}$

$\Sigma X =$ Keseluruhan Penjumlahan dari semua data periode waktu yang diperhitungkan

Jumlah Periode = Jumlah Periode Rata-rata bergerak

atau dapat ditulis dengan :

$$MA = (n_1 + n_2 + n_3 + \dots) / n$$

Keterangan

$MA = \text{Moving average}$

$n_1 =$ data periode pertama

$n_2 =$ data periode kedua

$n_3 =$ data periode ketiga dan seterusnya

$n =$ Jumlah Periode Rata-rata bergerak.

2.6 Perhitungan Nilai Akurasi

Dalam peramalan terdapat banyak metode yang dapat digunakan, namun tidak semua metode dapat sesuai dengan kasus yang ada. Secara umum ada tiga jenis perhitungan untuk melihat seberapa besar tingkat kesalahan dalam peramalan, yaitu:

2.6.1 MAD (*Mean Absolute Deviation*)

Merupakan perhitungan yang digunakan untuk menghitung rata-rata kesalahan mutlak, dengan rumus:

$$\text{MAD} = \sum | \text{Aktual} - \text{Forecast} | / n$$

Dari rumus diatas, dapat diartikan bahwa $\sum | \text{Aktual} - \text{Forecast} |$ adalah hasil pengurangan antara nilai aktual dan forecast masing-masing periode yang kemudian di absolute-kan, dan selanjutnya dilakukan penjumlahan terhadap hasil-hasil pengurangan tersebut. Dan n merupakan jumlah periode yang digunakan untuk perhitungan.

2.6.2 MSE (*Mean Square Error*)

Merupakan perhitungan yang digunakan untuk menghitung rata-rata kesalahan berpangkat, dengan rumus :

$$\text{MSE} = \sum (\text{Aktual} - \text{Forecast})^2$$

Dapat diartikan bahwa $\sum (\text{Aktual} - \text{Forecast})^2$ merupakan hasil pengurangan antara nilai aktual dan forecast yang telah dikuadratkan, kemudian dilakukan penjumlahan terhadap hasil-hasil tersebut. Dan n merupakan jumlah periode yang digunakan untuk perhitungan.

2.6.3 MAPE (*Mean Absolute Percent Error*)

Merupakan perhitungan yang digunakan untuk menghitung rata-rata persentase kesalahan mutlak, dengan rumus :

$$\text{MAPE} = \sum (|\text{Aktual} - \text{Forecast}| / \text{Aktual}) * 100 / n$$

Dapat diartikan bahwa $\sum (|\text{Aktual} - \text{Forecast}| / \text{Aktual})$ merupakan hasil pengurangan antara nilai aktual dan forecast yang telah di absolute-kan, kemudian di bagi dengan nilai aktual per periode masing-masing, kemudian dilakukan penjumlahan terhadap hasil-hasil tersebut. Dan n merupakan jumlah periode yang digunakan untuk perhitungan. Semakin rendah nilai MAPE, kemampuan dari model peramalan yang digunakan dapat dikatakan baik, dan untuk MAPE terdapat range nilai yang dapat dijadikan bahan pengukuran mengenai kemampuan dari suatu model peramalan, range nilai tersebut dapat dilihat pada tabel 1 [2].

Range Nilai MAPE	Arti
< 10 %	Kemampuan Model Peramalan Sangat Baik
10 - 20 %	Kemampuan Model Peramalan Baik
20 - 50 %	Kemampuan Model Peramalan Layak
> 50 %	Kemampuan Model Peramalan Buruk

MAD (Mean Absolute Deviation) digunakan jika seorang analis ingin mengukur kesalahan peramalan dalam unit ukuran yang sama seperti data aslinya. MSE (Mean Square Error) digunakan karena menghasilkan kesalahan yang moderat yang lebih disukai oleh suatu peramalan yang biasanya menghasilkan kesalahan yang lebih kecil tetapi kadang-kadang menghasilkan kesalahan yang sangat besar. MAPE (Mean Absolute Percent Error) digunakan jika ukuran variabel peramalan merupakan faktor penting dalam mengevaluasi akurasi peramalan tersebut. MAPE memberikan petunjuk seberapa besar kesalahan peramalan dibandingkan dengan nilai sebenarnya dari series tersebut.

(Analisa Perbandingan Nilai Akurasi Moving Average dan Exponential Smoothing untuk Sistem Peramalan Pendapatan pada Perusahaan XYZ, M. Azman Maricar 2019).

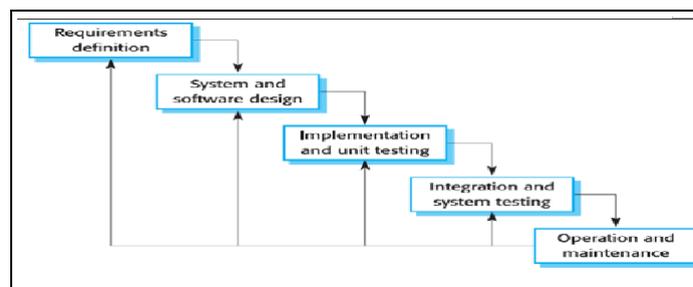
2.7 Design

Tahap ini dilakukan sebelum melakukan coding. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

2.7.1 Metodologi Waterfall

Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing* / *verification*, dan *maintenance*. Langkah demi langkah yang dilalui harus diselesaikan satu per satu dan berjalan secara berurutan, oleh karena itu disebut *waterfall* (Air Terjun).

Ian Sommerville (2011) menjelaskan bahwa ada lima tahapan pada Metode *Waterfall*, yakni *Requirements Analysis and Definition*, *System and Software Design*, *Implementation and Unit Testing*, *Integration and System Testing*, dan *Operational and Maintenance*.



Gambar 2.1 Metodologi Waterfall (Ian Sommerville, 2011)

2.7.2 UML (*Unified Modeling Language*)

Menurut (Prihandoyo, 2018) yang dikutip dari (Booch, 2005) *Unified Modeling Language* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik.

2.7.2.1 *Use Case Diagram*

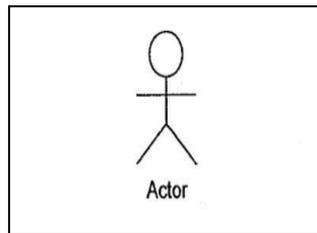
Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam use case terdapat actor yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

Menurut (Kurniawan, 2018) yang dikutip dari (Larman, 2005), Sebuah *Use Case Diagram* menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem. Diagram ini bisa menjadi gambaran yang bagus untuk menjelaskan konteks dari sebuah sistem sehingga terlihat jelas batasan dari sistem.

Komponen-komponen pembentuk diagram use case sebagai berikut :

1. Aktor

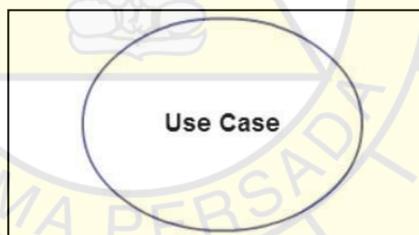
Menurut Muhamad Muslihudin & Oktafianto (2016) dalam buku yang berjudul “Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML”, aktor menggambarkan pihak-pihak yang berperan dalam sistem. Untuk lebih jelasnya dapat dilihat di gambar 2.1.



Gambar 2.2 Simbol Aktor (Kurniawan, 2018)

2. *Use case*

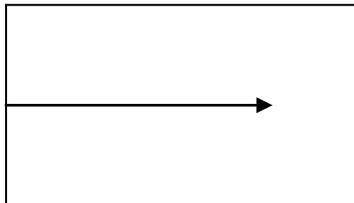
Menurut Muhamad Muslihudin & Oktafianto (2016) dalam Buku yang berjudul “Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML”, *use case* menggambarkan aktivitas / sarana yang disediakan oleh sistem. *Use case* digambarkan dalam bentuk elipse / oval.



Gambar 2.3 Simbol Use Case (M. Muslihudin & Oktafianto, 2016)

3. Relasi

Menurut Muhamad Muslihudin & Oktafianto (2016) dalam Buku yang berjudul “Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML”, merupakan garis atau simbol yang menghubungkan antara elemen *use case* dan aktor.

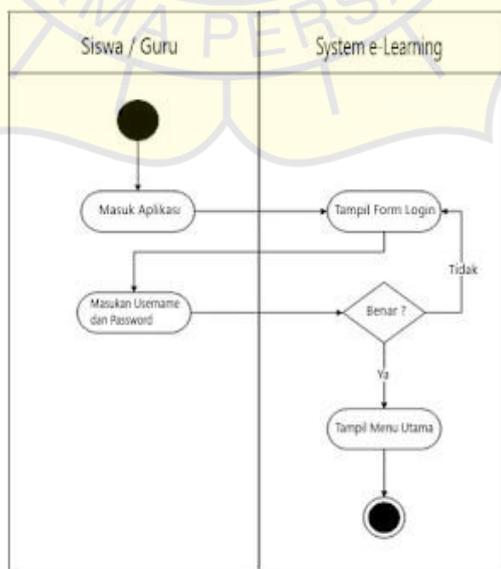


Gambar 2.4 Simbol Relasi (M. Muslihudin & Oktafianto, 2016)

2.7.2.2 Activity Diagram

Merupakan gambaran alir dari aktivitas-aktivitas didalam sistem yang berjalan. *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Menurut Murad (2013:57), “*Activity diagram* merupakan diagram yang bersifat dinamis. *Activity diagram* adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem dan berfungsi untuk menganalisa proses”.

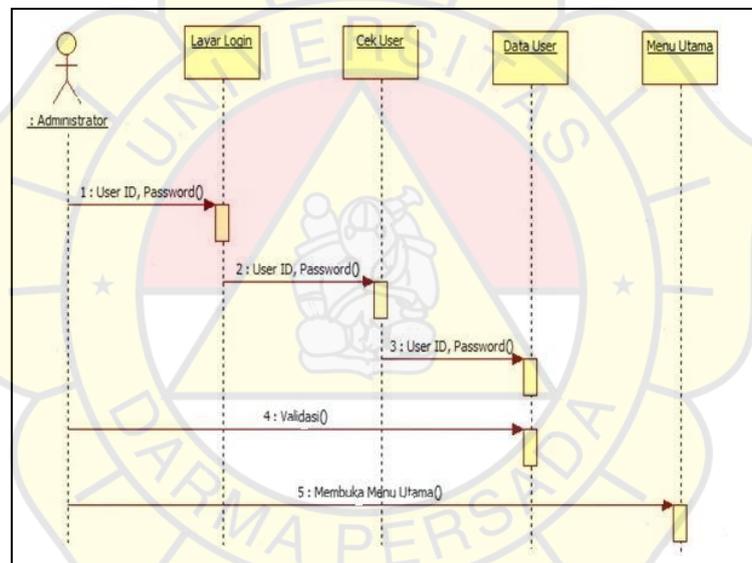


Gambar 2.5 Activity Diagram (Murad, 2013)

2.7.2.3 Sequence Diagram

Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa message yang digambarkan terhadap waktu.

Menurut Vidia (2013:20), ”*Sequence diagram* dibuat berdasarkan *activity diagram* dan *class diagram*. *Sequence diagram* menggambarkan aliran pesan yang terjadi antar kelas yang dideskripsikan pada *class diagram* dengan menggunakan operasi yang dimiliki kelas tersebut. Untuk aliran pesan, *sequence diagram* merujuk pada alur sistem *activity diagram* yang telah dibuat sebelumnya”.



Gambar 2.6 *Sequence Diagram* (Vidia, 2013)

2.8 Tool Pemograman

2.8.1 Web

Menurut (Hariyanto, 2015), *Web* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis

yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

2.8.1.1 Kelebihan Web

Aplikasi *Web* memiliki beberapa kelebihan menurut (Hidayatullah & Kawistara, 2015). yaitu:

- Kita dapat menjalankan aplikasi berbasis *web* dimanapun kapanpun tanpa harus melakukan penginstalan.
- Terkait dengan isu lisensi (hak cipta) kita tidak memerlukan lisensi ketika menggunakan *web-based application*, sebab lisensi telah menjadi tanggung jawab dari *web* penyedia aplikasi.
- Dapat dijalankan di sistem operasi manapun. Tidak peduli apakah kita menggunakan linux, windows, aplikasi berbasis *web* dapat dijalankan asalkan kita memiliki *browser* dan akses internet.
- Dapat diakses lewat banyak media seperti: komputer dan *handphone* yang sudah sesuai dengan standard *WAP (Wireless Application Protocol)*.
- Tidak perlu spesifikasi komputer yang tinggi untuk menggunakan aplikasi berbasis *web* ini.

2.8.1.2 Kekurangan Web

Web juga memiliki suatu kekurangan, yaitu :

- Cenderung lebih lambat karena memerlukan sebuah koneksi ke *server* secara berkala, sehingga butuh waktu untuk *loading* dari satu halaman ke halaman lainnya.

- Butuh pengetahuan bahasa cukup banyak. Ada beberapa bahasa yang harus dikuasai untuk membuat aplikasi berbasis *website* ini. Yang paling penting cuma, *HTML (Hypertext Markup Language)* sebagai bahasa utamakerangka *website*, dan satu bahasa pemrograman server yaitu *PHP (Hypertext Preprocessor)*, *ASP (Active Server Page)*, atau *Node.JS*. Selain itu kita juga butuh bahasa untuk pengolahan *database*, seperti *MySQL*, ada juga *PostgreSQL, MongoDB*, dsb.

2.8.2 *HTML*

Menurut Jubilee Enterprise (2016:16) mengklaim bahwa *HTML (Hyper Text Markup Language)* adalah sebuah teks berbentuk link dan mungkin juga foto atau gambar yang saat di klik akan membawa pengakses internet dari satu dokumen ke dokumen lainnya.

2.8.3 *CSS*

Menurut Jubilee Enterprise (2016:93) mengklaim bahwa *CSS (Cascading Style Sheet)* adalah kumpulan kode untuk mendefinisikan desain dan bahasa markup. Karena ada kata markup pada definisi *CSS*, maka relasi antara *CSS* dan *HTML* sangatlah dekat. Dengan *CSS* lah, sebuah desain website yang dibangun menggunakan *HTML* akan menjadi lebih menarik dan variatif. Jika didefinisikan secara bebas, *CSS* merupakan kumpulan kode untuk mendesain atau mempercantik tampilan halaman website. Dengan artian lain, dengan memanfaatkan *CSS* kita bias mengubah desain standar yang dihasilkan oleh *HTML* menjadi variasi-variasi yang lebih kompleks.

2.8.4 PHP

Menurut Agung, Gregorius (2015:1) mengklaim bahwa *PHP (Hypertext PreProcessor)* adalah Bahasa komputer / bahasa pemrograman / koding / script yang digunakan untuk mengolah data dari *server* untuk ditampilkan di *website*. *PHP* digunakan untuk membuat *website* dinamis. Dalam penggunaan murninya , kode-kode *PHP* disisipkan diantara kode *HTML*. Secara *default*, dokumen *PHP* memiliki ekstensi *.php*. Saat *server web* menemukan *file* dengan menggunakan jenis ini, *file* tersebut secara otomatis dikirim untuk diproses oleh prosesor *PHP*.

2.8.5 JavaScript

Menurut R.H. Sianipar (2015) mengklaim bahwa *javascript* adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti *Internet Explorer, Mozilla Firefox, Netscape dan Opera*. Kode *javascript* dapat disisipkan dalam halaman web menggunakan *tagscript*.

2.8.6 JQuery

Menurut Lukmanul Hakim (2016:45), dalam buku *Bikin Website Super Keren dengan PHP & JQuery*, *jquery* adalah *Javascript Library* berupa kumpulan kode / fungsi *javascript* siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode *javascript*. Secara standar, apabila kita membuat kode *javascript*, maka diperlukan kode yang cukup panjang, bahkan terkadang sangat sulit dipahami, *jquery* digunakan untuk menyederhanakan kode *javascript*. Hal ini sesuai dengan slogannya “*Write less, do more*”, cukup tulis sedikit tapi bisa melakukan banyak hal.

2.8.7 *Bootstrap*

Menurut Zaenal A. Rozi (2015:1) mengklaim bahwa *Bootstrap* adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah *website*. Bisa dikatakan, *Bootstrap* adalah *template design web* dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses *design web* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman.

2.8.8 *MySql*

Menurut R.H. Sianipar (2015) mengklaim bahwa MySQL bukan termasuk bahasa pemrograman. *MySQL* merupakan salah satu database populer dan mendunia. *MySQL* bekerja menggunakan *SQL Language (Structure Query Language)*. Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah) dan *DELETE* (menghapus).