



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Rianto (2021:3), “Komputer merupakan suatu perangkat elektronika yang memiliki kemampuan untuk menerima dan mengolah data menjadi informasi, menjalankan program yang tersimpan dalam memori, serta dapat bekerja secara otomatis berdasarkan perangkat aturan tertentu.”

Prawiro dalam Harmayani, dkk, (2021:1) mengemukakan bahwa, “Komputer adalah perangkat elektronik yang dapat digunakan untuk mengolah data sesuai dengan prosedur yang telah dirumuskan sebelumnya sehingga menghasilkan informasi bermanfaat bagi penggunanya.”

2.1.2 Pengertian Perangkat Lunak (*Software*)

Rosa dan Salahuddin (2018:2), mengatakan bahwa, “Perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Menurut Rianto (2021:5), “Perangkat lunak merupakan suatu data yang diprogram sedemikian rupa dan disimpan dalam bentuk digital yang tidak terlihat secara fisik tetapi tersimpan dalam media penyimpanan komputer.”

2.1.3 Pengertian Data

Menurut Pane, dkk, (2020:7), "Data adalah kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh dari hasil pengamatan (observasi), dan dapat diolah menjadi bentuk yang lebih kompleks, seperti: informasi, database, atau solusi untuk masalah tertentu".

Menurut Kristianto (2018:7), “Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata”.



Sedangkan, menurut Indrajani (2018:2), “Data adalah fakta atau observasi mentah yang biasanya mengenai fenomena fisik atau transaksi bisnis”.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Menurut Habibi dan Karnovi (2020:14), “Aplikasi adalah program siap pakai yang bisa dipakai untuk menjalankan sejumlah perintah dari pengguna aplikasi dengan tujuan untuk memperoleh hasil yang lebih akurat dengan tujuan pembuatan aplikasi tersebut.”

Menurut Setyawan dan Munari (2020), “Aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.”

2.2.2 Pengertian Permintaan

Menurut Himayati dalam Putra (2019:12), “Permintaan barang atau jasa disebut *Material Requisition* adalah transaksi intern perusahaan atas permintaan suatu barang/jasa. Transaksi ini ditujukan untuk manager yang bertanggung jawab terhadap pengadaan barang suatu barang/jasa.”

Sedangkan, menurut Putong dalam Putra (2019:11) “Permintaan (*demand*) adalah banyaknya jumlah barang yang diminta pada suatu pasar tertentu dengan tingkat harga tertentu pada tingkat pendapatan tertentu dan dalam periode tertentu.”

2.2.3 Pengertian Produk

Menurut Stantaon dalam Firmansyah, (2019:3), produk dapat diartikan “Sebagai sekumpulan atribut fisik yang secara nyata terkait dalam sebuah bentuk dapat diidentifikasi. Sedangkan secara luas, produk merupakan sekumpulan atribut yang nyata dan tidak nyata yang didalamnya mencakup warna, kemasan, harga, presise pengencer, dan pelayanan dari pabrik dan pengencer yang mungkin diterima oleh pembeli sebagai sebuah hal yang dapat memberikan kepuasan atas keinginannya”



Menurut Kotler dan Armstrong, (2018:79), “Produk berarti kombinasi barang dan jasa yang ditawarkan oleh perusahaan kepada pasar sasaran.” Produk Barang adalah seluruh produk yang memiliki bentuk fisik, bisa dilihat, disentuh, diraba, dipindahkan, dan juga memiliki perlakuan fisik lainnya.

2.2.4 Pengertian Kinerja

Menurut Yuniarti, dkk (2021:2) “Kinerja merupakan suatu hasil kerja yang dicapai oleh pegawai tersebut dalam pekerjaannya menurut kriteria tertentu yang berlaku untuk suatu pekerjaan tertentu.”

Menurut Rismawati dan Mattalata (2018:2) “Kinerja merupakan suatu kondisi yang harus diketahui dan dikonfirmasi kepada pihak tertentu untuk mengetahui tingkat pencapaian hasil suatu instansi dihubungkan dengan visi yang diemban suatu perusahaan atau perusahaan serta mengetahui dampak positif dan negatif dari suatu kebijakan operasional.”

2.2.5 Pengertian Website

Menurut Yuhefizar dalam Sa’ad (2020:3), “*Website* merupakan keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman web yang saling berhubungan.”

Menurut Abdulloh (2018:1), “*Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara, dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang diseluruh dunia.”

2.2.6 Pengertian Aplikasi Pengajuan Permintaan Produk untuk Meningkatkan Kinerja di Bagian Gudang pada Bapenda (Badan Pendapatan Daerah) Kota Palembang

Aplikasi Pengajuan Permintaan Produk untuk Meningkatkan Kinerja di Bagian Gudang pada Bapenda (Badan Pendapatan Daerah) Kota Palembang merupakan aplikasi yang berfungsi untuk membantu proses penginputan,



menyetujui, pencarian, pencetakan surat, dan laporan rekap oleh admin gudang dan masing-masing bidang yang ingin melakukan pengajuan permintaan produk di Bagian Gudang pada Bapenda (Badan Pendapatan Daerah) Kota Palembang.

2.3 Teori Khusus

2.3.1 Pengertian DFD (*Data Flow Diagram*)

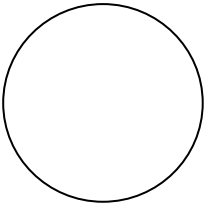


Menurut Sukamto dan Shalahudin (2018:70), “*Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) keluaran (*output*)”.

Pendapat tersebut selaras dengan Kristanto (2018:61), “DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar sistem, dimana data disimpan, proses apa yang menghasilkan tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.”

Adapun beberapa simbol yang biasanya digunakan dalam perancangan DFD, diantaranya seperti pada tabel dibawah ini.




Tabel 2.1 Simbol-simbol pada DFD

No	Notasi	Keterangan
1.		<p>Proses (<i>Process</i>) atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang seharusnya jadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
2.		<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi ilmiah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel pada basis data (<i>Entity Relationship Data</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)).</p> <p>Catatan: nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>
3.		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>Catatan: nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>



Lanjutan Tabel 2.1 Simbol-simbol pada DFD

4		<p>Aliran data; merupakan data yang dikirim antar-proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p> <p>Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.</p>
---	---	--

Sumber: Sukamto dan Shalahuddin (2018:71-72)




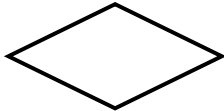


2.3.2 Pengertian *Flowchart*

Menurut Hanief dan Jepriana, (2020), “*Flowchart* adalah suatu teknik untuk menggambarkan urutan logika dari suatu prosedur penyelesaian masalah. Dengan kata lain, flowchart merupakan langkah-langkah penyelesaian masalah yang disajikan dalam bentuk simbol-simbol tertentu.”

Menurut Hendraputra, dkk (2021: 65) “*Flowchart* diawali dengan penerimaan *input* dan diakhiri dengan menampilkan *output*. Secara garis besar *flowchart* merupakan suatu gambaran yang menjelaskan urutan: (1). Pembacaan data; (2). Pemrosesan data; (3). Pengambilan keputusan terhadap data; (4). Penyajian hasil pemrosesan data.”

Berikut adalah simbol-simbol yang digunakan untuk menggambarkan algoritma dalam bentuk diagram alir dan kegunaan dari simbol-simbol yang bersangkutan.

Tabel 2.2 Simbol-simbol pada *Flowchart*

No	Simbol	Nama	Fungsi
1.		<i>Terminal</i>	Menyatakan pemulaan atau akhir suatu program
2.		<i>Input/ Output</i>	Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatanya.
3.		<i>process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
4.		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya / tidak.
5.		<i>Display</i>	Menyatakan piranti keluaran, seperti layar monitor, printer, dll.
6.		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer).

2.3.3 Pengertian ERD (*Entity Relationship Diagram*)

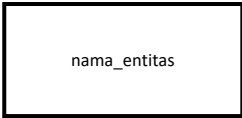
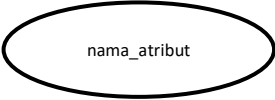
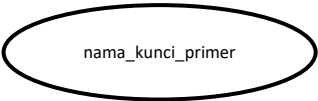
Menurut Sukamto dan Shalahuddin (2018:50-51), “Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD digunakan untuk pemodelan basis data *relational* sehingga jika penyimpanan basis data menggunakan OODBMS (*Object Oriented Database*



Management System) maka perancangan basis data tidak perlu menggunakan ERD”.

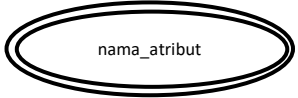
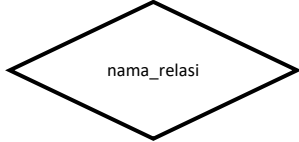

Menurut Rusmawan (2019:64), “ ERD merupakan gambaran grafis dari suatu model data yang menyertakan deskripsi detail dari seluruh entitas (*entity*), hubungan (*relationship*), dan batasan (*constrain*) untuk memenuhi kebutuhan sistem analis dalam menyelesaikan pengembangan sebuah sistem.”

Tabel 2.3 Simbol-simbol pada ERD

No	Simbol	Deskripsi
1.	Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa <i>id</i> ; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).



Lanjutan Tabel 2.3 Simbol-simbol pada ERD

4.	<p>Atribut Multinilai / <i>Multivalued</i></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p>
5.	<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p>
6.	<p>Asosiasi/<i>Association</i></p> 	<p>Penghubung antara relasi dan entitas di mana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas yang lain disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.</p>

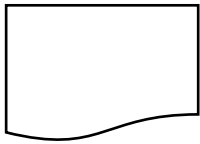
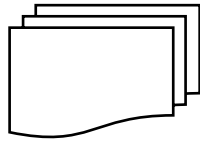
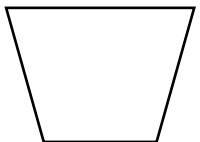
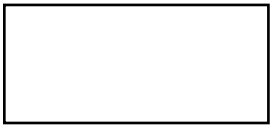
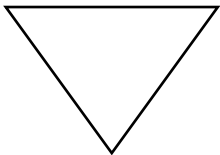
Sumber: Sukamto dan Shalahuddin (2018:50-51)



2.3.4 Pengertian *Block Chart*

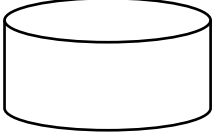

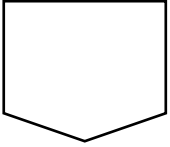
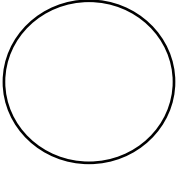

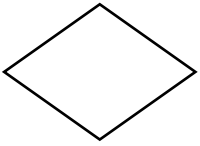
Kristanto (2018:75) menjelaskan, “*Block Chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Block Chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Tabel 2.4 Simbol-simbol pada *Block Chart*

No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen
3.		Proses manual
4.		Proses yang dilakukan oleh komputer
		Menandakan dokumen yang diarsipkan (arsip manual)





Lanjutan Tabel 2.4 Simbol-simbol pada *Block Chart*

6.		Data penyimpanan (<i>data storage</i>)
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran
11.		Pengambilan keputusan (<i>decision</i>)



Lanjutan Tabel 2.4 Simbol-simbol pada *Block Chart*

12.		Layar peraga (<i>monitor</i>)
13.		Pemasukan data secara manual

Sumber: Kristanto (2018:75-77)

2.3.5 Pengertian Kamus Data

Menurut Sukamto dan Shalahuddin (2018:73), mengungkapkan bahwa Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Ia juga menyatakan kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Menurut Kristanto (2018:72), “Kamus data adalah kumpulan elemen–elemen atau simbol–simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau *file* didalam sistem.”



Tabel 2.5 Simbol-simbol pada Kamus Data

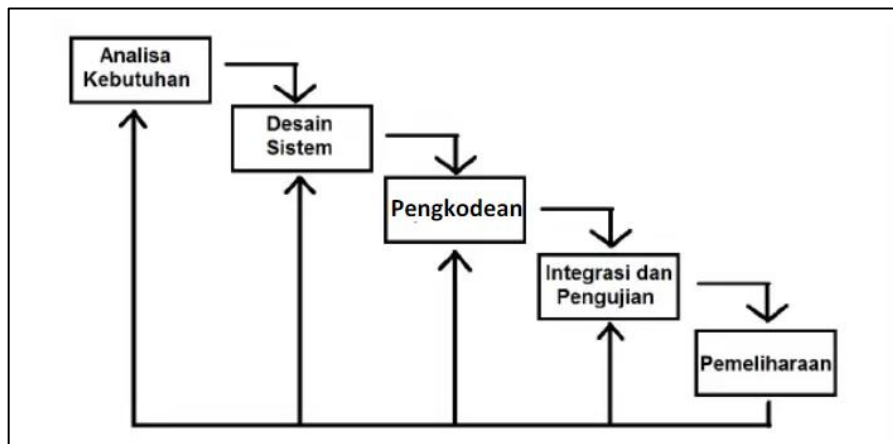
No	Simbol	Arti
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik ...atau...
4.	{ } ⁿ	N kali diulang / bernilai banyak
5.	()	Data opsional
6.	*...*	Batas Komentar

Sumber: Sukamto dan Shalahuddin (2018:74)

2.3.6 Metode Pengembangan Sistem

Metode yang penulis pilih sebagai dasar untuk menentukan pembuatan aplikasi berdasarkan model pengembangan perangkat lunak yaitu model air terjun atau *Waterfall*. Menurut Zendrato (2020) ” *Waterfall* atau AIR terjun adalah model yang dikembangkan untuk pengembangan perangkat lunak, membuat perangkat lunak. Model berkembang secara sistematis dari satu tahap ke tahap lain dalam mode seperti air terjun. Model pengembangan ini bersifat linear dari tahap awal pengembangan system yaitu tahap perencanaan sampai tahap akhir pengembangan system yaitu tahap pemeliharaan”.

Menurut Rosa (2022), “Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*)”.



Gambar 2.1 Ilustrasi Metode *Waterfall* (Rosa, 2022)

Adapun metode air terjun menurut Rosa (2022) yaitu:

2.3.6.1 Analisis Kebutuhan Perangkat Lunak

Kebutuhan Proses analisis kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2.3.6.2 Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang akan dihasilkan pada tahap ini juga perlu didokumentasikan.

2.3.6.3 Pembuatan Kode Program

Desain ini harus ditranslasikan ke dalam perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.



2.3.6.4 Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logika dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

2.3.6.5 Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi atau perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.4 Teori Program

2.4.1 HTML



Gambar 2.2 Logo HTML

Menurut (Anhar, 2020), “HTML (*Hypertext Markup Language*) adalah bahasa pemrograman web yang memiliki sintak atau aturan tertentu dalam menuliskan *script* atau kode-kode, sehingga *browser* dapat menampilkan informasi dengan membaca kode-kode HTML.”



Sedangkan menurut Abdulloh (2018:7), “HTML merupakan bahasa standar *web* yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website*.”

2.4.2 CSS



Gambar 2.3 Logo CSS

Menurut Yudhanto dan Prasetyo (2019:6), “CSS adalah *Cascading Stylesheet*, yaitu bahasa yang digunakan untuk HTML agar menjadi lebih bagus dan efektif dalam tampilan”

Sedangkan menurut Abdulloh (2018:45), “CSS merupakan suatu dokumen *web* yang berfungsi mengatur elemen HTML dengan berbagai properti yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan.”

2.4.3 JavaScript



Gambar 2.4 Logo JavaScript

Menurut Pahlevi, dkk., (2018:28), “JavaScript adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada *web browser* seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini* dan sebagainya.”



Sedangkan menurut Abdulloh (2018:193), “*Javascript* adalah Bahasa pemrograman *web* yang pemrosesannya dilakukan di sisi *client*.”

2.4.4 PHP



Gambar 2.5 Logo PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang digunakan secara luas untuk pengembangan aplikasi berbasis website.

Menurut (Anhar, 2020), “Bahasa pemrograman PHP (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman web berupa *script* yang dapat diintegrasikan dengan HTML.”

Menurut (Habibi dan Sandi, 2020), “*PHP Hypertext Pre-processor* atau *Personal Home Page* adalah suatu bahasa pemrograman bersifat *server side* yang digunakan untuk menerjemahkan sejumlah baris kode (bisa ditambah ke dalam HTML) program menjadi kode mesin dimengerti oleh mesin komputer.”

2.4.5 XAMPP



Gambar 2.6 Logo XAMPP



XAMPP merupakan *software* yang dikembangkan oleh sekelompok tim *Apache Friend* pada tahun 2002 dan bisa didapatkan secara gratis dengan label *General Public License* (GNU)

Menurut Iqbal (2019:15), “XAMPP merupakan sebuah *software web server Apache* yang di dalamnya sudah tersedia *database server MySQL* dan *support PHP programming*.”

Sedangkan, menurut Yudhanto dan Prasetyo (2019:17), “XAMPP adalah kompilasi program aplikasi gratis terfavorit dikalangan *developer/programmer* yang berguna untuk pengembangan *website* berbasis PHP dan MySQL.”

2.4.6 MySQL



Gambar 2.7 Logo MySQL

MySQL merupakan *software database* yang digunakan untuk pengembangan aplikasi web dalam bentuk tabel yang saling berhubungan. Menurut Anhar (2020), “MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS seperti *Oracle*, *MS SQL*, *Postagre SQL*, dan lain-lain.

Sedangkan menurut Chandra dan Suroto (2018), “MySQL adalah RDBMS (*Relation Database Management System*) yang cepat dan mudah digunakan, serta sudah banyak digunakan sebagai kebutuhan.”



2.4.7 Visual Studio Code

2.4.7.1 Pengertian Visual Studio Code



Gambar 2.8 Logo Visual Studio Code

Visual Studio Code adalah aplikasi *cross-platform* yang dikembangkan oleh Microsoft untuk berbagai sistem operasi seperti Windows, Linux dan Mac OS. Muncul dengan *built-in* dukungan untuk *JavaScript*, naskah dan *Node.js* dan memiliki *array* beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C #, Python, dan PHP. *Visual Studio* memiliki lintas platform kode editor yang ringan, dapat digunakan oleh siapa saja untuk membuat atau membangun aplikasi *web*.

2.4.7.2 Komponen pada Visual Studio Code

- *Customize*: Digunakan untuk menambahkan ekstensi Bahasa Pemrograman. Konfigurasi dan Kustomisasi template dengan menambahkan ekstensi Bahasa pemrograman maka tidak perlu selalu mengingat fungsi Bahasa.
- *Command Palette*: *Command Palette* menyediakan banyak akses perintah, kita bisa memberikan perintah editor membuka file, mencari file dan sebagainya dengan cepat dan mudah. untuk membuka *command palette* bisa dengan tekan Ctrl+Shift+p.
- *Integrated Terminal*: *Integratal Terminal* digunakan untuk mengeksekusi skrip di editor. Kita bisa mengeksekusi skrip editor secara langsung di terminal tanpa harus membuka terminal tambahan, ini adalah salah satu kelebihan dari *visual studio code*.



- *Extension*: *Extension* adalah fungsi tambahan dalam yang berfungsi untuk memperluas kemampuan dari editor yang dapat membantu *developer* dalam melakukan *programming*.
- *Search*: Fitur *search Visual Studio Code* juga sangat cepat nya kemudahan yang diberikan selain kecepatan *query* pencarian data juga dia akan mencari sampai ke level kontennya.
- *Grid Editor Layout*: Kita juga mudah dalam manajemen *layout visual studio code*, kita dapat dengan mudah mengatur grup editor dalam tata letak apapun baik secara vertikal maupun horizontal.
- *Color Themes*: *Color Themes* digunakan untuk memodifikasi warna dalam antarmuka *Visual studio Code* agar sesuai dengan selera yang diinginkan caranya dengan pilih *File >Preferences>Color Theme*, lalu geser *cursor* keatas dan kebawah untuk memilih tema yang diinginkan.
- *Cloud Environment*: Kita juga bisa melakukan sesuatu di lingkungan *cloud* melalui *Visual Studio Code* seperti membuat *database*, melakukan perintah, *insert, update, delete*, dan sebagainya di *cloud*.

2.4.7.3 Macam-macam *Extension* pada *Visual Studio Code*

- *Live server*: memungkinkan kita untuk memuat ulang halaman *web* secara otomatis ketika kamu mengubah di *Visual Studio Code*.
- *GitLens*: *GitLens* membantu kita untuk memvisualisasikan kepengarangan kode sekilas melalui anotasi tanpa hambatan.
- *Prettier (Code Formatter)*: Digunakan untuk menata format *code prettier* mendorong gaya penulisan *code* yang konsisten dengan memprasing kode anda dan memencetnya kembali dengan aturannya sendiri.
- *Auto Rename Tag*: *Auto rename tag* membantu kita ketika ingin mengganti sebuah tag dan juga untuk mengganti tag pembuka dan tag penutup bersamaan pada HTML dan XML.
- *Beautify*: *Beautify* membantu anda dalam *formatting* codingan kita, kita tidak perlu merapikan codingan kita secara manual cukup dengan *extension* ini akan otomatis terformat.



- *Open In Browser*: Digunakan untuk membuat langsung *project* anda pada *browser*.
- *CSS Peek*: berguna untuk melihat sebuah element HTML, yang terkena CSS, *CSS Peek* juga sangat mempermudah bagi *web designer*.
- *Indent Rainow*: Digunakan untuk untuk membantu kita memberikan warna dan juga mengelompokkan warnai sesuai tag awal dari struktur dari pemrograman yang kita buat.
- *SQL server*: adalah *extension* dimana kita adalah seorang *database engineer* *SQL server*, *extension* ini juga mempermudah anda bekerja dalam *Visual studio Code*.