

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada tinjauan pustaka ini, penulis melakukan tinjauan pustaka pada penelitian sebelumnya yang menggunakan metode yang sama. Sebagai salah satu pendukung dalam penelitian yang dilakukan penulis. Berikut ini merupakan tinjauan yang diambil yaitu sebagai berikut:

1. Yuyi Andrika, 2020, Otomatisasi Pendistribusian..Zakat..Pada.Badan.Amil Zakat Nasional Prov. Kep. Bangka Belitung, Metode *Extreme programming*, Sistem Informasi manajemen pendistribusian zakat yang dibuat hanya mencakup ruang lingkup kegiatan di bidang pendistribusian zakat Sistem yang dibangun membantu bidang pendistribusian dalam proses pencatatan pendistribusian dalam proses pencatatan proposal, hingga proses pelaporan hasil distribusi, Sistem informasi manajemen pendistribusian zakat sudah merupakan sistem yang terkomputerisasi dan menjadi solusi pencatatan data yang lebih rapi.
2. Muhammad Ardiansyah, 2022, Rancang bangun fitur *chat* pada *ecommerce* berbasis syariah (studi kasus: akad murabahah ijabqabul.id), Metode *Extreme programming*, Berdasarkan hasil pembahasan dan uji coba yang telah dilakukan, aplikasi yang yang dikembangkan dapat digunakan untuk membantu pengguna dalam bertransaksi dan bertukar informasi antar pengguna. Adapun saran yang dapat penulis sampaikan setelah menyelesaikan skripsi ini bahwa aplikasi ini masih memiliki banyak kekurangan..dan.jauh dari kata sempurna, serta tidak menutup kemungkinan untuk mengembangkan dan memperbaiki kekurangan yang ada seperti tampilan yang belum responsif, sehingga belum

dapat diakses pada smartphone secara maksimal dan belum adanya fitur untuk melakukan panggilan suara.

3. Bahar, 2020 Pengembangan model sistem informasi manajemen pengelolaan artikel ilmiah berbasis *web* menggunakan metode *extreme programming*, Metode *Extreme programming*, Berdasarkan hasil uji fungsionalitas sistem yang dilakukan dengan melibatkan perwakilan masing-masing pengguna, terlihat bahwa fungsi-fungsi yang diharapkan berdasarkan hasil analisis kebutuhan, telah dipenuhi oleh model sistem aplikasi berbasis *Web* yang dikembangkan dan dapat difungsikan dengan baik. Namun demikian, pengguna terlihat masih bingung menjalankan beberapa fitur fungsional yang tersedia pada aplikasi, sehingga diharapkan sistem perlu dilengkapi dengan file manual pengoperasian yang lebih lengkap, runut sesuai prosedur bisnis sistem, serta mudah dan praktis dipahami oleh pengguna, jika dibandingkan dengan manual pengoperasian yang telah ada saat ini.

4. Goldie Gunadi, 2021 Rancang Bangun Sistem Peminjaman Laptop dengan Metode *Extreme programming* Menggunakan Framework Bootstrap, Metode *Extreme programming*, Dari penelitian yang dilakukan ini dapat diambil kesimpulan bahwa pengembangan sistem informasi peminjaman laptop berbasis *web* dengan menggunakan metode XP dapat menjadi salah satu kekuatan bagi PT Gramedia dalam menghadapi tantangan bisnis yang diakibatkan adanya pandemi COVID-19. Dengan adanya sistem informasi peminjaman laptop, proses administrasi dapat dilakukan dengan lebih efektif dan efisien, sehingga setiap kebutuhan karyawan terhadap perangkat laptop dapat terpenuhi dengan baik.

5. Muhammad wahyudi, 2021 Rancang bangun aplikasi pembelajaran praktikum sistem informasi akuntansi di smks barunawati surabaya, Metode *Extreme programming*, aplikasi pembelajaran praktikum sistem informasi akuntansi di smks barunawati surabaya telah berhasil dirancang sebaik-baiknya. Fitur dan fungsi pada aplikasi SIA terbukti berfungsi semua dengan hasil dari uji coba aplikasi dengan menggunakan *Black Box* Testing yang menghasilkan 41 Test Case dan semuanya Pass,

Kesimpulan yang dapat penulis ambil dari tinjauan pustaka di atas adalah bahwa metode pengembangan sistem yang digunakan penulis berbeda-beda, dalam penelitian yang akan diteliti, penulis menggunakan metode *Extreme programming* sebagai metode penelitian dan pengembangan, karena selain fleksibel, cara ini juga dapat mempermudah dan mempercepat proses pengembangan sistem yang sedang berjalan.

2.2 Dasar Teori

2.2.1 Sistem

Dalam merancang sistem informasi, maka terlebih dahulu kita harus mengetahui pengertian tentang sistem. Adapun beberapa pengertian sistem antara lain, Menurut Indrajit mengemukakan bahwa sistem mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya. Menurut Hutahaean (2014), supaya sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu :

1. Komponen

Suatu sistem terdiri dari beberapa komponen yang saling berinteraksi yang bersama-sama membentuk satu kesatuan. Komponen sistem

terdiri dari komponen-komponen yang berupa subsistem atau bagian dari suatu sistem..

2. Batasan sistem (*boundary*)

Batas sistem adalah area yang memisahkan suatu sistem dari sistem lain atau lingkungan luarnya. Kendala sistem ini memungkinkan kita untuk melihat sistem sebagai satu kesatuan. Batas sistem menunjukkan ruang lingkup sistem.

3. Lingkungan luar sistem (*environment*)

Lingkungan eksternal sistem berada di luar batas sistem yang memengaruhi perilakunya. Lingkungan dapat berupa lingkungan menguntungkan yang perlu dipertahankan dan lingkungan merugikan yang perlu dipertahankan dan dikendalikan jika tidak mempengaruhi kelangsungan hidup sistem..

4. Penghubung sistem (*interface*)

Sistem penghubung merupakan media penghubung antara suatu subsistem dengan subsistem lainnya. Koneksi ini memungkinkan sumber daya mengalir dari salah satu subsistemnya ke subsistem lainnya. *Output* subsistem memasuki subsistem lain melalui koneksi.

5. Masukkan sistem (*input*)

Masukan adalah data yang dimasukkan ke dalam sistem dan dapat berupa pemeliharaan (*maintenance input*) dan masukan sinyal (*signal input*). Pekerjaan pemeliharaan adalah energi yang dikeluarkan untuk menjaga sistem tetap berjalan. Sinyal *input* adalah energi yang diproses untuk mendapatkan *output*. Contoh program sistem komputer adalah

maintenance input dan data adalah sinyal *input* yang diolah menjadi informasi..

6. Keluaran sistem (*output*)

Keluaran adalah hasil dari energi yang diproses dan dibagi menjadi keluaran yang bermanfaat .

7. Pengolah sistem

Suatu sistem menjadi bagian pemrosesan yang mengubah *input* menjadi *output*. Sistem produksi mengolah bahan mentah menjadi bahan jadi, sistem akuntansi mengolah data menjadi laporan keuangan.

8. Sasaran sistem

Sistem itu harus mempunyai sasaran atau tujuan. Objek dari sistem mendefinisikan masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem.

2.2.2 Informasi

Menurut Romney & Steinbart (2015) Informasi adalah “informasi yang telah dikelola dan diproses untuk meningkatkan dan memberi makna pada proses pengambilan keputusan”.

2.2.3 Sistem Informasi

Abdul Kadir mengungkapkan bahwa dalam praktiknya, istilah sistem informasi cenderung digunakan tanpa embel-embel komputer, padahal komputer merupakan bagian yang penting. Mengenai konsep sistem informasi, dapat disimpulkan bahwa sistem informasi adalah sekumpulan dari beberapa proses operasional, seperti: Penyimpanan, analisis dan pengolahan data untuk menghasilkan informasi. (Abdul Kadir, 2014)

2.2.4 Sewa

Sewa adalah kontrak antara penyewa dan pihak penyedia jasa penyewaan. Penyewaan sepeda motor adalah transaksi hukum antara dua pihak dengan sepeda motor, yang masing-masing memiliki hak dan kewajiban berdasarkan perjanjian ini, baik secara tertulis maupun dalam bentuk perjanjian..

Berikut definisi-definisi penyewaan menurut para ahli : Richard Eddy (2010) mengungkapkan bahwa “Sewa adalah perjanjian di mana satu pihak setuju untuk menawarkan pihak lain kenikmatan barang untuk jangka waktu tertentu dengan imbalan pembayaran harga yang disepakati oleh pihak terakhir.”.

Kimmel dalam penelitian Aang Viki Amanda (2011) mengungkapkan bahwa “penyewaan adalah pengaturan dimana pemilik aset perusahaan memberikan kesempatan kepada pihak lain untuk menggunakan aset yang ada untuk jangka waktu tertentu dengan harga yang disepakati. Dari definisi di atas dapat disimpulkan bahwa sewa adalah kontrak antara 2 pihak untuk jangka waktu dan harga yang disepakati bersama..

2.3 SDLC (*Software Development Life Cycle*)

SDLC adalah kependekan dari *software development life cycle* atau dalam bahasa Indonesia disebut siklus hidup pengembangan sistem. SDLC merupakan siklus yang digunakan dalam membuat atau mengembangkan sistem informasi yang ditujukan untuk memecahkan masalah secara efektif. Dengan kata lain, SDLC adalah tahapan pekerjaan yang bertujuan untuk menghasilkan sistem berkualitas tinggi yang memenuhi keinginan pelanggan atau tujuan penggunaan sistem tersebut. SDLC adalah framework yang berisi langkah-langkah yang harus diselesaikan untuk pengembangan perangkat lunak.

tahapan SDLC secara global adalah sebagai berikut :

1. Inisiasi (*initiation*).

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*system concept development*).

Menentukan ruang lingkup konsep, termasuk dokumentasi di seluruh sistem, analisis biaya-manfaat, manajemen rencana, dan pembelajaran kenyamanan sistem.

3. Perencanaan (*planning*).

Kembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Memberikan dasar untuk memperoleh sumber daya (*resources*) yang diperlukan untuk solusi tersebut.

4. Analisis kebutuhan (*Requirements analysis*).

Menganalisis persyaratan sistem perangkat lunak (*User*) dan mengembangkan persyaratan pengguna. Buat dokumen persyaratan fungsional.

5. Desain (*design*).

Mengubah persyaratan rinci menjadi persyaratan lengkap, dokumen desain sistem berfokus pada cara mencapai fungsi yang diperlukan.

6. Pengembangan (*development*).

Ubah desain menjadi sistem informasi yang lengkap, termasuk cara mendapatkan dan menginstal lingkungan sistem yang diperlukan, membuat *database* dan menyiapkan kasus uji, menyiapkan satu atau lebih file uji, kode, menyusun, memperbaiki dan membersihkan program, memeriksa dan menguji.

7. Integrasi dan pengujian (*integration and test*).

Buktikan bahwa sistem perangkat lunak memenuhi persyaratan yang ditentukan dalam dokumen persyaratan fungsional. Di bawah bimbingan personel penjaminan mutu (*quality assurance*) dan pengguna. Buat laporan analisis pengujian.

8. Implementasi (*Implementation*).

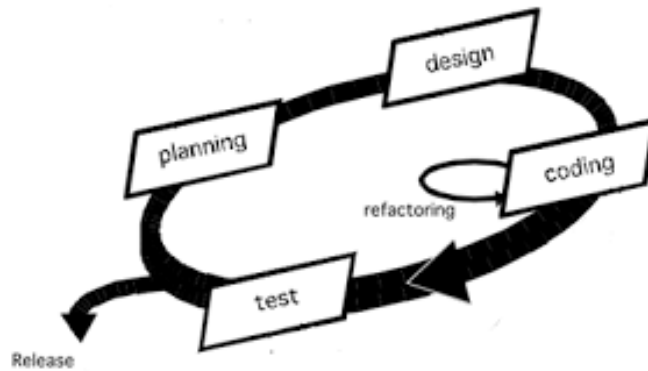
Ini termasuk mempersiapkan implementasi, mengimplementasikan persyaratan perangkat lunak dalam lingkungan produksi, dan menjalankan solusi untuk masalah yang ditemukan dari fase integrasi dan pengujian.

9. Operasi dan Pemeliharaan (*operations and maintenance*).

Menjelaskan pekerjaan pengoperasian dan pemeliharaan sistem informasi di lingkungan produksi (lingkungan pengguna), termasuk implementasi akhir dan memasuki proses audit.

2.4 *Extreme programming*

Extreme programming Ini adalah salah satu metode pengembangan perangkat lunak dan banyak digunakan oleh pengembang aplikasi atau sistem informasi. Menurut Dwiky Andika, XP adalah suatu pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai fase proses pengembangan sehingga metodenya menjadi lebih adaptif dan fleksibel. XP menggunakan pendekatan "ekstrim" untuk pengembangan iteratif, atau teknik pengembangan siklus iterative atau berulang.



Gambar 2.1 *Extreme programming*

Pada tahap perancangan, *Extreme programming* memiliki 4 tahap yang harus dikerjakan, *Extreme programming* terdiri dari beberapa tahap yaitu sebagai berikut.:

2.4.1 Perencanaan (*Planing*)

Sesuai dengan kebutuhan pengguna (spesifikasi pengguna), studi kelayakan dan teknis dan teknologi serta perencanaan proyek dan/atau perangkat lunak sistem informasi.

2.4.2 Perancangan (*Design*)

Mencari solusi dari permasalahan yang dapat diperoleh dari tahap analisis, pada tahap ini dibagi menjadi dua bagian.:

1. Pada tahap perencanaan lebih ditekankan pada *platform* yang hasil dari tahap analisis diimplementasikan.
2. Tahap desain, dimana kategori yang diperoleh pada tahap analisis disempurnakan dan kategori yang membuat sistem/perangkat lunak menjadi lebih efektif dan efisien ditambahkan dan dimodifikasi.

2.4.3 Pengkodean (*Coding*)

Tahapan ini merupakan kegiatan penerapan pemodelan yang sudah dibuat kedalam bentuk *User interface* dengan menggunakan bahasa pemrograman.

2.4.4 Pengujian (*Testing*)

Sebelum sistem diserahkan kepada pelanggan, terlebih dahulu harus dilakukan pengujian. Hasil kontrol script dari deskripsi *use case* yang disepakati dengan klien. Pengujian merupakan elemen penting dan harus dilakukan. Implementasi Di akhir fase implementasi, berbagai mesin klien yang akan digunakan dipasang. Ini juga termasuk manajemen transisi dari sistem lama ke sistem baru.

2.5 Analisi *Pieces*

Analisis pieces merupakan analisis standar diperlukan untuk menentukan apakah sistem baru ini layak atau tidak. *PIECES* terdiri dari aktivitas, pengetahuan, ekonomi, manajemen, efisiensi dan layanan. Keras (Hanif Al Fatta, 2007:51) Analisis kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan layanan pelanggan harus dilakukan untuk mengidentifikasi masalah.

2.6 UML (*Unified Modeling Language*)

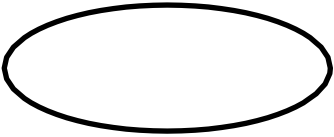
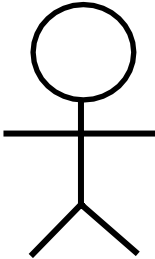
Tujuan pemodelan dalam kerangka pengembangan sistem adalah sarana analisis, pemahaman, visualisasi dan komunikasi antara beberapa anggota tim pengembangan. Beberapa diagram UML untuk mendukung pengembangan sistem.

2.6.1 *Usecase Diagram*



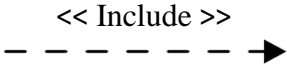
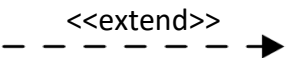
Use case diagram adalah fungsi atau interaksi terkait antara aktor dan sistem. Atau lebih umum dapat diartikan sebagai suatu teknik yang digunakan

perangkat lunak untuk menentukan persyaratan fungsional suatu sistem. Mendefinisikan *use case* diagram sendiri merupakan proses deskriptif yang dilakukan untuk merepresentasikan hubungan antara pengguna dengan sistem yang dirancang. Representasi grafik yang dihasilkan dirancang secara sederhana dan dirancang untuk memudahkan pengguna dalam membaca informasi yang diberikan.

Tabel 2.1 Simbol *Use case diagram*

| No | Nama | Simbol | Keterangan |
|----|-----------------|---|--|
| 1 | <i>Use case</i> |  | Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan kata kerja diawal frase nama <i>use case</i> |
| 2 | Aktor |  | Aktor adalah orang, atau proses yang berinteraksi dengan sistem yang dibuat, yang biasanya menggunakan kata benda di awal fase. |

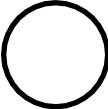

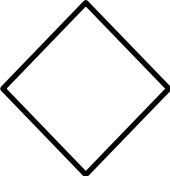

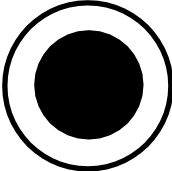
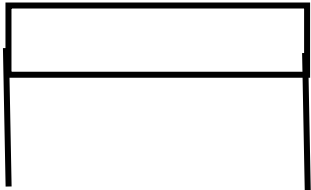
Tabel 2.3 Lanjutan tabel simbol *use case diagram*

| No | Nama | Simbol | Keterangan |
|----|----------------|---|---|
| 3 | Asosiasi |  | Asosiasi merupakan symbol yang digunakan sebagai komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case diagram</i> atau <i>use case</i> memiliki interaksi dengan aktor. |
| 4 | Generalisasi |  | Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dalam <i>usecase</i> . |
| 5 | <i>Include</i> |  | Menunjukkan bahwa suatu <i>usecase</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya |
| 5 | <i>Extend</i> |  | Mempesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan. |

2.6.2 Activity Diagram

Diagram yang dapat memodelkan proses-proses yang berjalan dalam sistem. Alur proses tanaman digambarkan secara vertikal. Diagram aktivitas adalah pengembangan dari *use case* yang memiliki aliran aktivitas.



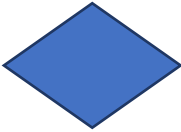


Tabel 2.2 Simbol *activity diagram*

| No | Simbol | Keterangan |
|----|---|--|
| 1 |  | Status awal sebuah diagram yang memiliki status awal. |
| 2 |  | Aktivitas yang dilakukan oleh sistem biasanya diawali dengan kata kerja. |
| 3 |  | <i>Decision</i> merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4 |  | <i>Join</i> merupakan asosiasi yang mana terdapat lebih dari satu aktivitas yang digabungkan menjadi satu. |
| 5 |  | Status akhir yang mana sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6 |  | <i>Swimlane</i> merupakan pemisah organisasi bisnis yang bertanggung jawab dalam aktivitas yang terjadi. |


2.6.3 Flowchart

Flowchart adalah deskripsi grafis dari langkah-langkah program dan aliran proses. Diagram alir membantu analis dan pengembang memecah masalah menjadi potongan-potongan kecil dan menganalisis alternatif dalam tindakan. *Flowchart* biasanya dapat memecahkan masalah dengan lebih mudah, terutama yang membutuhkan penelitian dan evaluasi lebih lanjut. Diagram, nama dan propertinya disajikan pada tabel di bawah ini:

Tabel 2.3 *Flowchart*

| No | Simbol | Gambar | Keterangan |
|----|----------------------|---|--|
| 1 | Garis |  | Digunakan untuk menunjukkan urutan eksekusi atau aliran. |
| 2 | <i>Input /Output</i> |  | Digunakan untuk merepresentasikan proses <i>input</i> / baca dan <i>output</i> / tulis |
| 3 | Decision |  | Digunakan untuk merepresentasikan operasi perbandingan logis |
| 4 | Proses |  | Digunakan untuk pemrosesan aritmatika dan transmisi data |
| 5 | Preparation |  | Digunakan untuk menetapkan nilai awal ke variable |

Tabel 2.4 Tabel lanjutan *flowchart*

| No | Nama | Simbol | Keterangan |
|----|------------|---|--|
| 6 | Terminator |  | Digunakan untuk menunjukkan awal dan akhir program |

2.7 Pengertian Pemodelan Berorientasi Objek

Object oriented programming adalah sebuah metode pemrograman dimana pengembang aplikasi tidak hanya mendefinisikan variabel yang berisi state dari sebuah struktur data, tetapi juga mendefinisikan fungsi untuk menunjukkan *behavior* yang diaplikasikan pada struktur data.

2.8 Bahasa Pemrograman dan Perangkat Lunak Pendukung

Bahasa pemrograman yang digunakan adalah PHP sedangkan perangkat lunak pendukung yang digunakan adalah, *Xampp, MySQL, Visual studio code*.

2.8.1 PHP (*Hypertext Preprocessor*)

PHP adalah skrip yang terintegrasi dengan HTML dan berada di *server* (skrip tertanam HTML sisi *server*). Dengan PHP kita dapat membuat berbagai aplikasi berbasis web, mulai dari halaman web sederhana hingga aplikasi kompleks yang membutuhkan koneksi ke *database*. Kode PHP selalu dimulai dengan tag `<?php`. File yang berisi tag HTML dalam kode PHP ini menggunakan ekstensi file `.php` atau didefinisikan di *Apache/plugin server web* lainnya. Jika file tersebut digunakan dalam plugin ini, hapus server tempat file tersebut berisi. Kode `.php`. *Server* mengkompilasi kode dan menghasilkan keluaran berupa *markup HTML* yang dikirim ke browser klien yang mengakses file. (Abdul Kadir, 2003).

Menurut Madcoms (2011:11) PHP adalah bahasa pemrograman yang dapat dijalankan pada web server dan melakukan pengolahan data pada server tersebut. Dengan bantuan program PHP, website menjadi lebih interaktif dan dinamis.

PHP adalah bahasa pemrograman yang banyak digunakan untuk situs web baru-baru ini. Dengan PHP kita dapat membuat website dinamis dengan *database* ke *database*. Beberapa perintah dasar yang sering digunakan PHP untuk mengakses *database* adalah:

1. Hubungkan ke *database*.
2. Memasukkan data ke dalam *database*.
3. Menampilkan data dari *database*.
4. Hapus data dari *database*.
5. Memodifikasi data *database*. Sampai PHP 5.1, *driver* lokal seperti ekstensi

MySQL Pada PHP 5.4, penggunaan ekstensi *MySQL* sudah mulai ditinggalkan dan akan sepenuhnya dihapus di PHP 5.5.

PHP adalah bahasa pemrograman berorientasi objek yang berjalan dalam sistem pemrograman *web* yang berfungsi sebagai penanganan data pada *server web*.

2.8.2 HTML (*Hyper Text Markup Language*)

HTML adalah singkatan dari *Hyper Text Markup Language*, sebuah skrip yang mengontrol *form* at tag untuk membuat dan memelihara struktur situs *web*. (Rosa AS, M.Shalahuddin, 2014) Beberapa fitur utama HTML saat membuat halaman *web* adalah sebagai berikut:

1. Menentukan *layout website*.

2. Memform at text dasar seperti pengaturan paragraf, dan *form* at font.
3. Membuat list.
4. Membuat tabel.
5. Menyisipkan gambar, video, dan audio.
6. Membuat link.
7. Membuat *form* ulir

2.8.3 CSS

CSS adalah singkatan dari *Cascading Style Sheet*, yang mendefinisikan tampilan elemen yang ditulis dalam bahasa markup. Fungsi CSS yang lain adalah memisahkan konten dari tampilan visual website. Awalnya dikembangkan pada tahun 1996, CSS sendiri digunakan untuk menata halaman.

2.8.4 MySQL

MySQL, atau dibaca *My Sequel*, adalah sistem manajemen basis data, atau sering disingkat DBMS, yang bekerja dengan perintah SQL (*Structured Query Language*) yang sering digunakan untuk membuat aplikasi berbasis web. Selain itu, SQL dipuji karena kesederhanaannya dalam sintaks pendek dan mudah dipahami.



Gambar 2.2 Logo MySQL

2.8.5 *Xampp*

XAMPP adalah perangkat lunak yang memungkinkan Anda menjalankan *server web* dan *database* di komputer host lokal. Dengan *XAMPP* Anda dapat membuat versi offline situs web dan dengan mudah mengakses *server web* dan *database* dari komputer lokal Anda.



Gambar 2.3 Logo *xampp*

2.8.6 *Visual studio code*

Visual Code Studio adalah sebuah code editor gratis yang bisa dijalankan di perangkat desktop berbasis Windows, Linux, dan macos. Code editor ini dikembangkan oleh salah satu raksasa teknologi dunia, tapi tetap ringan ketika digunakan. bisa dipakai untuk membuat dan mengedit *source code* berbagai Bahasa pemrograman Misalnya, seperti *javascript*, *type script*, dan *Node.js*.



.Gambar 2.4 Logo *Visual studio code*