

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Dibawah ini beberapa penelitian terdahulu yang pernah dilakukan berkaitan dengan penelitian yang akan dilakukan dapat dilihat pada Tabel dibawah ini:

Tabel 2. 1 Tinajaun Pustaka

| No | Nama (Tahun) | Judul | Metode | Hasil |
|----|-----------------------|--|--|--|
| 1 | Ulfa et al (2020) | Penerapan <i>Metode VlseKriterijumska Optimizacija I Kompromisno Resenje</i> (VIKOR) dalam Pemilihan Air Conditioner Terbaik | <i>Metode VlseKriterijumska Optimizacija I Kompromisno Resenje</i> (VIKOR) | Hasil penelitian yaitu sistem pemilihan air conditioner yang mempermudah pihak perusahaan |
| 2 | Lengkong et al (2015) | Implementasi Metode VIKOR untuk Seleksi Penerima Beasiswa. | Metode VIKOR | Hasil penelitian menunjukkan bahwa metode VIKOR dapat membantu proses seleksi dan menentukan penerima beasiswa yang tepat. Selain itu, metode VIKOR dapat membuat perangkan kompromi alternatif dari sejumlah alternatif yang ada. |
| 3 | Mutiara (2020) | Sistem Pendukung Keputusan Pemilihan Santri Terbaik Tahfidzh Qur ' an Pada Yayasan Islamic Center | Metode VIKOR | Hasil penelitian yaitu sistem yang dibangun menggunakan metode Visekriterijumsko Kompromisno |

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

| No | Nama (Tahun) | Judul | Metode | Hasil |
|----|---|--|--|---|
| | | Menggunakan Metode VIKOR | | Rangiranje (VIKOR) yang diharapkan mampu untuk membantu Yayasan Islamic Center dalam pemilihan santri terbaik tahfidzh qur'an |
| 4 | Primanda, Santoso and Afirianto (2018) | Pemilihan Kost di Sekitar Universitas Brawijaya menggunakan Metode Analytical Hierarchy Process (AHP) dan Simple Additive Weighting (SAW). | Metode Analytical Hierarchy Process (AHP) dan Simple Additive Weighting (SAW). | Hasil penelitian yaitu sistem yang dibangun untuk pemilihan tempat kost ini dapat menggunakan metode Analytic Hierarchy Process (AHP) dan Simple Additive Weighting (SAW) untuk mendapatkan tempat kost yang baik. |
| 5 | Fadillah <i>et al</i> (2019) | Penerapan Metode Simple Multi-Attribute Rating Technique untuk Pemilihan Lokasi Kos Terbaik di Kawasan UIN Suska Riau. | Metode Simple Multi-Attribute Rating Technique | hasil penelitian dan pembahasan yang telah dilakukan pada penerapan metode Simple Multi-Attribute Rating Technique (SMART) untuk melakukan penelitian terhadap pemilihan lokasi kos-kosan terbaik pada kawasan kampus UIN Suska Riau. |
| 6 | Primanda, Santoso and Afirianto (2018), | Pemilihan Kost di Sekitar Universitas Brawijaya menggunakan | SAW | hasil perbandingan. Persentase kecocokan yang |

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

| No | Nama (Tahun) | Judul | Metode | Hasil |
|----|------------------------------|--|--------|---|
| | | Metode Analytical Hierarchy Process (AHP) dan Simple Additive Weighting (SAW) | | paling tinggi terhadap 7 kriteria adalah 6 kriteria dengan 8,831% kemudian 5 kriteria dengan 7,999%, 4 kriteria dengan 4,761%, 3 kriteria dengan 3,58%, 2 kriteria dan 1 kriteria dengan 0%. Dalam pengujian tersebut menunjukkan bahwa banyak kriteria yang digunakan berpengaruh pada hasil perancangan kost yang direkomendasikan berdasarkan banyaknya kriteria yang digunakan. |
| 7 | Fadillah <i>et al</i> (2019) | Penerapan Metode Simple Multi-Attribute Rating Technique untuk Pemilihan Lokasi Kos Terbaik di Kawasan UIN Suska Riau. | SMART | hasil penelitian dan pembahasan yang telah dilakukan pada penerapan metode Simple Multi-Attribute Rating Technique (SMART) untuk melakukan penelitian terhadap pemilihan lokasi kos-kosan terbaik pada kawasan kampus UIN Suska Riau didapatkan perancangan |

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

| No | Nama (Tahun) | Judul | Metode | Hasil |
|----|-------------------------|---|--------|---|
| | | | | <p>nilai kriteria berdasarkan alternatif maka didapatkan Jalan Mustamindo merupakan lokasi terbaik untuk pemilihan lokasi kos-kosan terbaik di kawasan UIN Suska Riau dengan nilai 0,64, kemudian 0,63 pada Gang Iman, 0,61 pada Jalan Garuda Sakti, dan seterusnya hingga yang terakhir yaitu Jalan Perumahan dengan nilai 0,30.</p> |
| 8 | Wardhani and Nur (2017) | Sistem Pendukung Keputusan Pemilihan Tempat Kos Untuk Mahasiswa Di Luwuk Banggai Dengan Metode SAW (Simple Additive Weighting). | SAW | Hasil penelitian ini adalah sistem pendukung keputusan pemilihan tempat kos dengan menggunakan metode Simple Additive Weighting (SAW). |
| 9 | Pramudhita (2017) | Sistem pendukung keputusan pemilihan rumah kost putra untuk mahasiswa di kota malang dengan menggunakan metode saw. | SAW | Hasil menunjukkan bahwa sistem pencarian kost ini dapat membantu calon penyewa kost untuk mendapatkan alternatif terbaik tempat kost yang dapat direkomendasikan berdasarkan kriteria yang |

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

| No | Nama (Tahun) | Judul | Metode | Hasil |
|----|----------------|--|--------|--|
| | | | | dipilih pengguna dalam memilih tempat kost dengan metode Simple Additive Weighting (SAW), dengan hasil pengujian fungsional sistem sebesar 100% untuk proses data dan fungsi utama sistem. |
| 10 | Asniadi (2015) | Sistem Pendukung Keputusan Pemilihan Kost di Sekitar Kampus Unmuh Jember Menggunakan Metode Simple Additive Weighting (SAW). | SAW | Hasil menunjukkan bahwa sistem pencarian kost ini dapat membantu calon penyewa kost untuk mendapatkan alternatif tempat kost yang dapat direkomendasikan berdasarkan kriteria yang dipilih pengguna dalam memilih tempat kost dengan metode Simple Additive Weighting (SAW). |

2.2. Sistem Penunjang Keputusan

Sistem pendukung keputusan SPK merupakan suatu sistem interaktif berbasis komputer, yang membantu pengambil keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-

masalah yang sifatnya semi terstruktur dan tidak terstruktur, yang intinya mempertinggi efektifitas pengambil keputusan (Setiyaningsih, Arosyid and Fachtur, 2015)

2.3. Metode *Visekriterijumsko Kompromisno Rangiranje* (VIKOR)

Metode VIKOR adalah metode pengoptimalan multi-kriteria yang digunakan dalam sistem yang kompleks. Metode ini berfokus pada peringkat dan pemilihan dari sekumpulan alternatif, dan mengidentifikasi solusi kompromi untuk masalah yang bertentangan dengan kriteria, yang dapat membantu pembuat keputusan membuat keputusan akhir. Di sini, solusi kompromi adalah solusi yang layak, yang paling mendekati ideal, dan kompromi berarti kesepakatan dengan konsesi bersama (Kristyawan, 2018). Langkah-langkah yang digunakan dalam metode VIKOR adalah sebagai berikut:

1. Normalisasi matrik dengan cara nilai terbaik dalam satu kriteria dikurangi dengan nilai data sampel i kriteria j . Lalu dibagi dengan nilai terbaik dalam satu kriteria di kurangi dengan nilai terjelek dalam satu kriteria.

$$R_{ij} = \frac{(\max X_{ij} - X_{ij})}{(\max X_{ij} - \min X_{ij})} \dots \dots \dots (1)$$

Dimana : R_{ij} = nilai normalisaai sampel i kriteria j

X_{ij} = nilai data sampel i kriteria j

Max X_{ij} = nilai terbaik dalam satu kriteria

Min X_{ij} = nilai terendah dalam satu kriteria

i = alternatif

j = kriteria

2. Menghitung nilai *Utility Measure (S)* dan *Regret Measure (R)*.

Menghitung *utility measure* dengan cara menjumlah hasil perkalian bobot dengan hasil normalisasi matrik, menghitung *regret measure* dengan cara nilai maksimal dari perkalian bobot dengan hasil normalisasi.

$$W_j * R_{ij} \dots \dots \dots (2)$$

$$S_j = \sum_{j=1}^n W_j * R_{ij} \dots \dots \dots (3)$$

$$R_i = \max[W_j * R_{ij}] \dots \dots \dots (4)$$

$W_j = \text{Bobot Kriteria}$

3. Menghitung indeks vikor dengan cara nilai S dikurangi nilai S terbesar dikurangi dengan nilai S terkecil dan dikali v dan dijumlahkan dengan nilai R dikurangi nilai R terkecil lalu di bagi dengan nilai R terbesar dikurangi nilai R

terkecil R dan dikali dengan 1 dikurangi v

$$Q_i = \frac{(S_i - S_i \text{ min})}{(S_i \text{ max} - S_i \text{ min})} * v + \frac{(R_i - R_i \text{ min})}{(R_i \text{ max} - R_i \text{ min})} * (1 - v) \dots \dots \dots (5)$$

$V = 0.5$

$S_i \text{ min} = \text{nilai } S \text{ kecil}$

$S_i \text{ max} = \text{nilai } S \text{ besar}$

$R_i \text{ min} = \text{nilai } R \text{ kecil}$

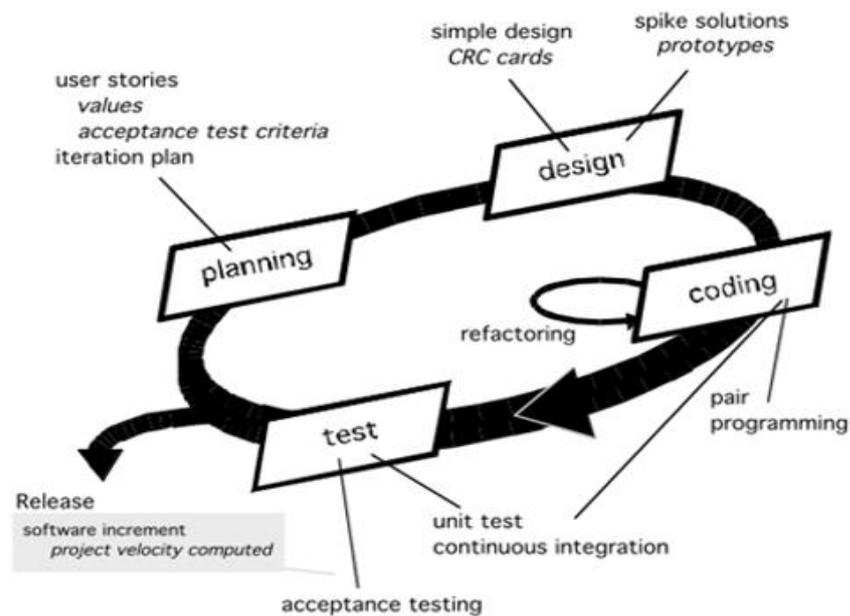
$R_i \text{ max} = \text{nilai } R \text{ besar}$

Sampel dengan nilai Q terkecil merupakan sampel terbaik

2.4. Metode Pengembangan Sistem *Extreme Programming*

Menurut Pressman (2015) *Extreme Programming (XP)* adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis

pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan di mana persyaratan pelanggan baru dapat diadopsi. Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah *testing*.



Gambar 2. 1 Model Extreme Programming (XP)

Sumber : Pressman (2015)

Menurut Pressman (2012) Proses *Extreme Programming* (XP)

1. *Planning* : Tahap *planning* dimulai dengan membuat *user stories* yang menggambarkan output, fitur, dan fungsi - fungsi dari *software* yang akan dibuat. *User stories* tersebut kemudian diberikan bobot seperti prioritas dan dikelompokkan untuk selanjutnya dilakukan proses *delivery* secara incremental.

2. *Design* : *Design* di *Extreme Programming* mengikuti prinsip *Keep It Simple* (KIS). Untuk *design* yang sulit, *Extreme Programming* akan menggunakan *Spike Solution* dimana pembuatan *design* dibuat langsung ke tujuannya. *Extreme Programming* juga mendukung adanya *refactoring* dimana *software system* diubah sedemikian rupa dengan cara mengubah stuktur kode dan menyederhanakannya namun hasil dari kode tidak berubah.
3. *Coding* : Proses *coding* pada *Extreme Programming* diawali dengan membangun serangkaian unit test. Setelah itu pengembangan akan berfokus untuk mengimplementasikannya. Dalam *Extreme Programming* diperkenalkan istilah *Pair Programming* dimana proses penulisan program dilakukan secara berpasangan. Dua orang *Programmer* saling bekerjasama di satu komputer untuk menulis program. Dengan melakukan ini akan didapat *real-time problem solving* dan *real-time quality assurance*.
4. *Testing* : Tahap ini dilakukan pengujian kode pada unit test. Dalam *Extreme Programming*, diperkenalkan *Extreme Programming acceptance test* atau biasa disebut *customer test*. Tes ini dilakukan oleh customer yang berfokus kepada fitur dan fungsi sistem secara keseluruhan. *Acceptance test* ini berasal dari *user stories* yang telah diimplementasikan.

2.5. Pemodelan Berbasis Objek (UML)

Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri

untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Rosa and Shalahudin, 2018). Beberapa jenis diagram *UML* antara lain sebagai berikut:

2.5.1. Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa and Shalahudin, 2018), simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel 2.2 di bawah ini:

Tabel 2. 2 Simbol Diagram Use Case (Rosa and Shalahudin, 2018)

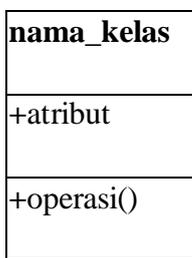
| Simbol | Deskripsi |
|---|--|
| <i>Use Case</i>  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> |
| Aktor/ <i>actor</i>  | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> |
| Asosiasi/ <i>association</i>  | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> |
| Ekstensi/ <i>extend</i> | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan |

| Simbol | Deskripsi |
|--|--|
| <<extend>>  | dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan |
| Generalisasi/ <i>generalization</i>  | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| Menggunakan/ <i>Include/uses</i> <<include>>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya |

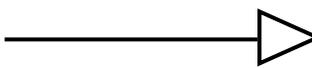
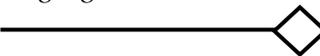
2.5.2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa and Shalahudin, 2018), simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.3 di bawah ini:

Tabel 2. 3 Simbol Class Diagram (Rosa and Shalahudin, 2018)

| Simbol | Deskripsi |
|--|--|
| Kelas  | Kelas pada struktur sistem |
| Antarmuka/ <i>Interface</i>  nama_ <i>interface</i> | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek |

Tabel 2.3 Simbol *Class Diagram* (Lanjutan)

| Simbol | Deskripsi |
|--|---|
| Asosiasi/ <i>association</i>  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
| Asosiasi berarah/ <i>directed association</i>  | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> |
| Generalisasi  | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
| Kebergantungan/ <i>dependecy</i>  | Relasi antar kelas dengan makna kebergantungan antar kelas |
| Agregasi/ <i>agregation</i>  | Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>) |

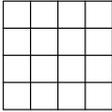
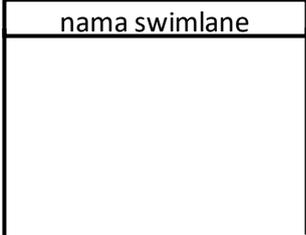
2.5.3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa and Shalahudin, 2018), simbol-simbol yang ada pada *activity diagram* dapat dilihat pada Tabel 2.4 di bawah ini :

Tabel 2. 4 Simbol *Activity Diagram* (Rosa and Shalahudin, 2018)

| Simbol | Deskripsi |
|--|---|
| Status awal  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |

Tabel 2.4 Simbol *Activity Diagram* (Lanjutan)

| Simbol | Deskripsi |
|---|--|
| Aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan/ <i>decision</i>  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| Penggabungan/ <i>join</i>  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
| Tabel  | Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis |
| Dokumen  | Menunjukkan dokumen sumber atau laporan |
| Status akhir  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| <i>Swimlane</i>  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

2.6. Alat Implementasi Program

2.6.1. Pengertian SQL

Menurut Rosa and Shalahudin (2018) SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada *Relation DBMS (Database Management System)*.

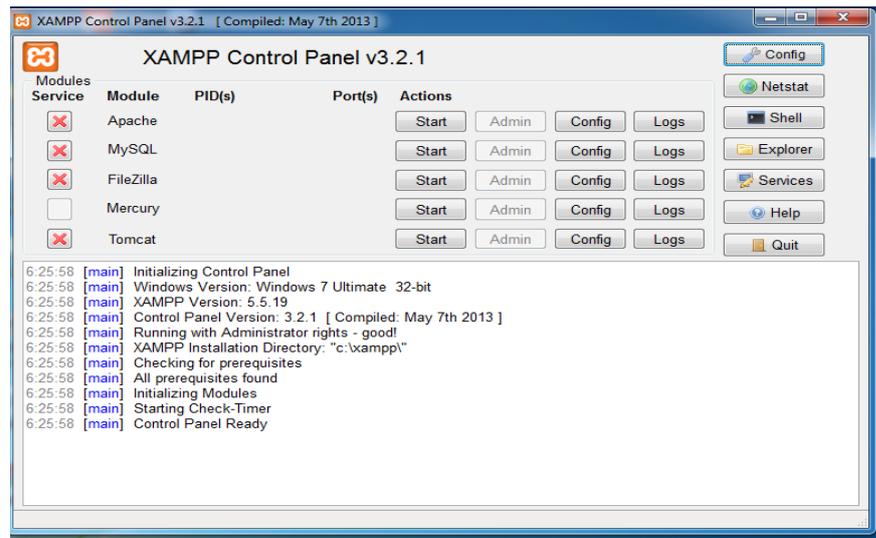
2.6.2. PHP (*Hypertext Preprocessor*)

PHP (PHP: *Hypertext apareprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. PHP termasuk dalam *Open Source Product*, sehingga source code PHP dapat diubah dan didistribusikan secara bebas. Versi terbaru PHP dapat diunduh secara gratis di situs resmi PHP: <http://www.php.net>. PHP juga dapat berjalan pada berbagai web server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami. PHP juga mampu lintas platform. Artinya PHP dapat berjalan di banyak sistem operasi yang beredar saat ini, di antaranya : Sistem Operasi Microsoft Windows (semua versi), Linux, Mac OS, Solaris. PHP dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI (*Common Gateway*

Interface). PHP dapat mengirim HTTP header, dapat mengatur cookies, mengatur *authentication* dan *redirect user*.

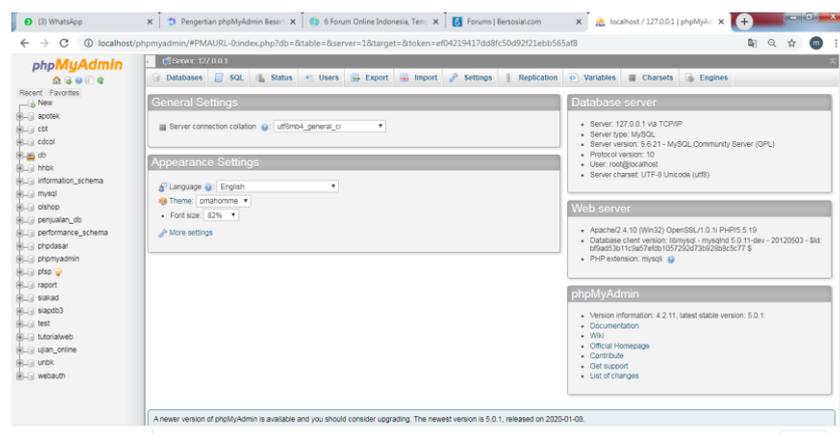
2.6.3. XAMPP

XAMPP merupakan perangkat lunak bebas (*open source*) yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL *database*, Apache HTTP *Server*, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU (*General Public License*) dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Menurut (Hidayatullah and Kawistara, 2017) XAMPP *support* untuk banyak sistem operasi seperti *Windows*, *Linux*, *Mac OS* dan *Solaris* sehingga tidak terdapat masalah ketika melakukan perpindahan sistem operasi dan Menurut (Nugroho, 2015) XAMPP adalah paket web programming, akan tetapi kita bisa memanfaatkan *database MySQL server*-nya untuk belajar Programming Visual, juga disana telah tersedia *tools PHP Myadmin* yang hanya berjalan disisi *server web* seperti *Apache Server*.



Gambar 2. 2 Gambar Xampp

PhpMyAdmin merupakan salah satu software penting dalam pengelolaan database dengan menggunakan MySQL (Bahasa SQL). Sesuai dengan namanya *phpMyAdmin* ini ditulis dalam Bahasa pemrograman PHP. *PhpMyAdmin* khusus menangani pengelolaan database dalam lingkup website (World Wide Web). Sama halnya dengan MySQL, software yang rilis perdana pada tahun 1998 ini juga berlisensi GNU (General Public License) dan sudah mendukung *multilingual* (multi bahasa) dalam user interfacenya.



Gambar 2. 3 Gambar PhpMyadmin

2.7. Pengujian *Black – Box*

Pendekatan pengujian *Black-Box* adalah metode pengujian di mana data tes berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir. Karena hanya fungsi dari modul perangkat lunak yang menjadi perhatian, pengujian *Black-Box* juga mengacu pada uji fungsional, metode pengujian menekankan pada menjalankan fungsi dan pemeriksaan inputan dan data output (Howden, 2017).

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- a. Fungsi – fungsi yang tidak benar atau hilang,
- b. Kesalahan interface
- c. Kesalahan dalam struktur data atau akses eksternal
- d. Kesalahan kinerja
- e. Inisialisasi dan kesalahan terminasi

Pada *black box testing* terdapat jenis teknik design tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan, diantaranya sebagai berikut:

1. *Equivalence Class Partitioning*
2. *Boundary Value Analysis*
3. *State Transitions Testing*
4. *Cause-Effect Graphing*