

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian Terdahulu**

Penelitian terdahulu ini menjadi salah satu acuan dalam melakukan penelitian sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, tidak ditemukan kesamaan judul dengan judul yang akan dibuat. Namun menggunakan beberapa penelitian sebagai referensi untuk memperkaya bahan kajian pada penelitian ini. Berikut merupakan penelitian terdahulu berupa jurnal terkait dengan penelitian yang dilakukan.

Pada penelitian sebelumnya (Wahyudi, 2020) dalam jurnal yang berjudul **“Rancang Bangun Aplikasi Pembayaran Iuran Komite Berbasis Web Di SMK Taman Ilmu Kromengan”**, memiliki permasalahan pada pembayaran iuran komite yang masih dilakukan secara manual dengan cara mencatat di buku besar. Permasalahan yang timbul dari pencatatan di buku besar ini yaitu proses transaksi yang lama, kesalahan *input* data pembayaran, pengelolaan data yang sulit, dan terlambatnya penyampaian laporan ke kepala sekolah. Oleh karena itu, dibutuhkan suatu pengalihan dan perubahan proses pengelolaan data pembayaran yang dilakukan secara manual menjadi pengelolaan data berbasis komputer. Aplikasi yang dibuat menggunakan bahasa pemrograman PHP dan *database* MySQL. Pada implementasinya, aplikasi yang dibuat dapat mengelola data siswa, mengimpor data siswa, serta dapat mengelola data pembayaran. Dalam hasil pengujian yang dilakukan, aplikasi pembayaran iuran komite yang dibuat berhasil dijalankan dengan baik.

Penelitian berikutnya (Hutabri, dkk. 2021) dengan jurnal berjudul **“Perancangan Sistem Informasi Pembayaran Uang Komite Sekolah Berbasis Web”**. Tujuan penelitian ini adalah untuk merancang sistem informasi berbasis *web* yang dapat membantu bendahara sekolah untuk melihat atau mencetak laporan pembayaran, melihat atau mencetak data siswa yang sudah atau belum melakukan pembayaran, sehingga informasi yang dihasilkan menjadi lebih cepat, akurat, dan

valid. Adapun aplikasi yang dibuat menggunakan metode SDLC (*System Development Life Cycle*), bahasa pemrograman PHP, dan *database* menggunakan MySQL. Pada implementasinya, bendahara akan mendata pembayaran pada aplikasi dan otomatis data tersebut akan dikumpulkan berupa laporan harian dan harian bulanan. Dalam hasil pengujian yang dilakukan, sistem informasi pembayaran uang komite yang dibuat berhasil dijalankan dengan baik.

Penelitian yang lain yaitu (Yolanda, dkk. 2021) dengan judul **“Perancangan Sistem Informasi Pembayaran SPP Berbasis Website Pada SMA Fajrul Islam Jakarta”**, memiliki permasalahan pada proses pembayaran yang masih dilakukan secara manual, yaitu dengan cara mengantri lalu menyerahkan kartu pembayaran SPP kepada bendahara, sehingga mengalami kesulitan dalam menangani antrian dan laporan keuangan yang disajikan kurang memadai. Sehingga menyebabkan kesalahan dalam perhitungan data dan pembuatan laporan pembayaran komite. Untuk itu, pada penelitian ini bertujuan merancang sistem berbasis *online* yang terintegrasi sehingga proses transaksi menjadi lebih efisien dan efektif dalam program pelayanan. Metode pengembangan perangkat lunak pada penelitian ini menggunakan model *waterfall*. Sistem informasi yang dibuat menggunakan bahasa pemrograman PHP dan *database* MySQL. Dalam hasil pengujian yang dilakukan, sistem informasi pembayaran komite yang dibuat berhasil dijalankan dengan baik.

Berdasarkan penelitian terdahulu pada jurnal yang telah dipublikasikan, terdapat persamaan antara penelitian yang telah dilakukan dengan penelitian yang akan dilakukan, yaitu bahasa pemrograman yang digunakan dalam pembuatan aplikasi menggunakan bahasa pemrograman PHP dan *database management system* MySQL. Akan tetapi pada penelitian yang akan dilakukan, ditambahkan pula pembayaran buku LKS sehingga aplikasi tidak hanya menerima pembayaran sumbangan komite dan digunakan pula *framework* Laravel sehingga pembuatan aplikasi akan lebih mudah, cepat, dan banyak fitur yang dapat dikembangkan. Selain itu terdapat perbedaan pada metode yang digunakan, pada penelitian yang akan dilakukan menggunakan metode *prototyping* sehingga aplikasi pembayaran yang dibuat dapat lebih menyesuaikan dengan kebutuhan pengguna.

## **2.2. Teori Umum**

### **2.2.1. Aplikasi**

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju (Nurchayono, 2017).

Menurut KBBI (Kamus Besar Bahasa Indonesia) dalam jurnal yang ditulis oleh Juansyah (2015) “Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju”.

### **2.2.2. Komite**

Komite merupakan badan mandiri yang mewadai peran serta masyarakat dalam rangka meningkatkan mutu, pemerataan, dan efisiensi di satuan pendidikan, baik pada jalur pra pendidikan sekolah maupun jalur pendidikan luar sekolah. (Sinaga, 2017).

### **2.2.3. LKS**

LKS merupakan suatu bahan ajar cetak berupa lembar-lembar kertas yang berisi materi, ringkasan, dan petunjuk-petunjuk pelaksanaan tugas pembelajaran yang harus dikerjakan oleh peserta didik, yang mengacu pada kompetensi dasar yang harus dicapai (Prastowo, 2013).

### **2.2.4. Website**

*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 di seluruh dunia. Halaman *website* dibuat menggunakan bahasa standar yaitu HTML. *Script* HTML ini akan diterjemahkan oleh *website browser* sehingga dapat ditampilkan dalam bentuk informasi yang

dapat dibaca oleh semua orang (Abdulloh, 2018).

### 2.2.5. Aplikasi Web

Aplikasi *web* merupakan aplikasi yang berjalan di berbagai *browser* seperti Internet Explorer, Mozilla, Opera dan sebagainya (Nugroho, 2010). Aplikasi web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web (Simarmata, 2010).

### 2.2.6. Database

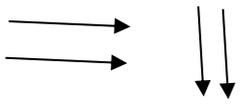
*Database* adalah kumpulan banyak data yang saling terkait dan terkumpul dalam satu tempat yang sama dan dipakai oleh sistem aplikasi yang dikontrol secara terpusat serta memiliki nilai yang berharga bagi pemilik (Setiawan, 2017).

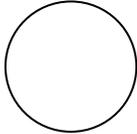
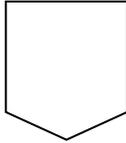
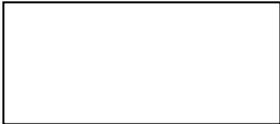
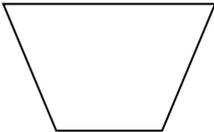
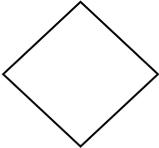
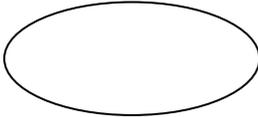
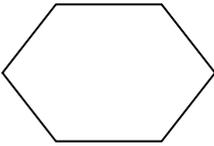
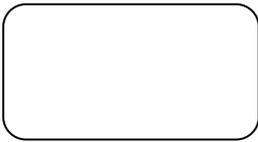
Basis data atau *database* adalah kumpulan data terstruktur. Agar dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam *database* komputer, dibutuhkan sistem manajemen basis data (*database management system*) (Suharyanto, dkk, 2017).

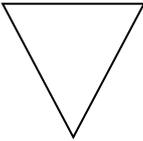
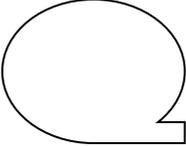
### 2.2.7. Flowchart

*Flowchart* atau diagram alir merupakan representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri dari sekumpulan simbol, dimana masing-masing simbol mempresentasikan suatu kegiatan tertentu. Simbol-simbol *flowchart* yang biasanya dipakai adalah simbol standar yang dikeluarkan oleh ANSI dan ISO. (Sari, 2017).

**Tabel 2.1** Simbol-Simbol *Flowchart*

NO	SIMBOL	KETERANGAN
1		Simbol arus/ <i>flow</i> , berfungsi untuk menyatakan jalannya arus suatu proses.

2		<p>Simbol <i>connector</i>, berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.</p>
3		<p>Simbol <i>offline connector</i>, berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.</p>
4		<p>Simbol <i>process</i>, berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.</p>
5		<p>Simbol <i>manual</i>, berfungsi untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.</p>
6		<p>Simbol <i>decision</i>, berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya/tidak.</p>
7		<p>Simbol <i>terminal</i>, berfungsi untuk menyatakan permulaan atau akhir suatu program.</p>
8		<p>Simbol <i>predefined process</i>, berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</p>
9		<p>Simbol <i>keying operation</i>, berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>.</p>

10		Simbol <i>offline-storage</i> , berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
11		Simbol manual input, berfungsi untuk memasukkan data secara manual dengan menggunakan online <i>keyboard</i> .
12		Simbol <i>input/output</i> , berfungsi untuk menyatakan proses input atau output tanpa tergantung jenis peralatannya.
13		Simbol <i>magnetic tape</i> , berfungsi untuk menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis.
14		Simbol <i>disk storage</i> , berfungsi untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
15		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui printer).

### 2.2.8. Entity Relationship Diagram (ERD)

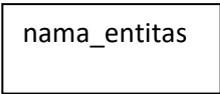
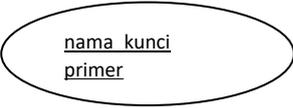
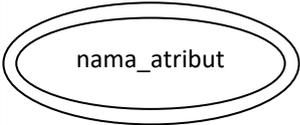
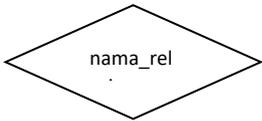
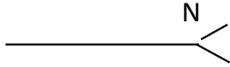
*Entity Relationship Diagram* adalah sebuah pendekatan *top-bottom* dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut dengan entitas, dan hubungan antara entitas-entitas tersebut yang digambarkan dalam suatu model (Christian, dkk, 2018).

ERD (*Entity Relationship Diagram*) merupakan model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data

(*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya (Fridayanthie dan Mahdiati, 2016).

Berikut pada Tabel 2.1 adalah simbol-simbol yang digunakan dalam *entity relationship diagram* disertai dengan keterangan fungsinya:

**Tabel 2.2** Simbol-Simbol *Entity Relationship Diagram*

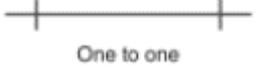
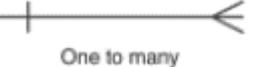
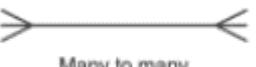
No.	Simbol	Keterangan
1.	Entitas / <i>Entity</i> 	Entitas merupakan data inti yang akan disimpan.
2.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan
3.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
4.	Atribut multi nilai / <i>Multi value</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas.
6.	Asosiasi / <i>Association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.

Pada *entity relationship diagram* terdapat *cardinality ratio constraint* atau kardinalitas relasi, yaitu relasi yang menunjukkan jumlah maksimum *entity* atau entitas yang dapat berelasi dengan *entity* atau entitas pada himpunan *entity* atau

entitas lain. Terdapat tiga jenis kardinalitas relasi yaitu satu ke satu (1:1), satu ke banyak atau banyak ke satu (1:N atau N:1) dan banyak ke banyak (N:N) (Jayanti dan Sumiari, 2018).

Berikut pada Tabel 2.3 adalah simbol-simbol pada kardinalitas dan keterangannya:

**Tabel 2.3** Simbol-Simbol Kardinalitas

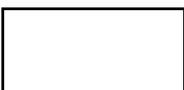
No.	Simbol	Keterangan
1.	 One to one	Merupakan simbol untuk kardinalitas satu ke satu.
2.	 One to many	Merupakan simbol untuk kardinalitas satu ke banyak.
3.	 Many to many	Merupakan simbol untuk kardinalitas banyak ke banyak.

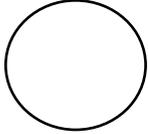
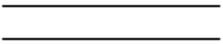
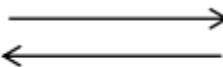
### 2.2.9. Data Flow Diagram

*Data flow diagram* (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun secara komputerisasi. (Sipayung dan Novichasari, 2021).

Berikut pada Tabel 2.4 adalah simbol-simbol yang digunakan dalam *data flow diagram* disertai dengan keterangannya:

**Tabel 2.4** Simbol-Simbol *Data Flow Diagram*

No.	Simbol	Keterangan
1.		Menggambarkan asal atau tujuan data di luar sistem dan berinteraksi dengan sistem.

2.		Menunjukkan transformasi dari masukan menjadi keluaran.
3.		Menunjukkan tempat data disimpan.
4.		Menggambarkan aliran data yang masuk ke proses atau keluar dari proses.

### 2.3. Teori Program

#### 2.3.1. HyperText Markup Language (HTML)

HTML merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman *website* yang dapat diakses atau ditampilkan menggunakan *web browser*. (Setiawan, 2017).

HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar *website* yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa *tag-tag* yang menyusun setiap elemen dari *website*. HTML berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website* sesuai *layout* yang diinginkan. (Abdulloh, 2018).

#### 2.3.2. Cascading Style Sheets (CSS)

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen *website* yang berfungsi mengatur elemen HTML dengan berbagai *property* yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan. Cara kerja CSS dalam memodifikasi HTML dengan memilih elemen HTML yang akan diatur kemudian memberikan *property* yang sesuai dengan tampilan yang diinginkan (Abdulloh, 2018).

#### 2.3.3. JavaScript

JavaScript merupakan bahasa pemrograman *website* yang pemrosesannya dilakukan di sisi *client*. Karena berjalan disisi *client*, JavaScript dapat dijalankan

hanya dengan menggunakan *browser*. Berbeda dengan PHP yang bekerja disisi *server*, untuk menjalankan *script* JavaScript tidak memerlukan *refresh* pada *browser*. JavaScript biasanya dijalankan ketika ada *event* tertentu yang terjadi pada halaman *website*. Baik *event* yang dilakukan oleh *user*, maupun *event* yang terjadi karena adanya perubahan pada halaman *website* (Abdulloh, 2018).

#### 2.3.4. Hypertext Preprocessor (PHP)

PHP atau *Hypertext Preprocessor* adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server-side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP juga dapat diintegrasikan dengan HTML, JavaScript, JQuery, atau Ajax. Namun, pada umumnya PHP lebih banyak digunakan bersamaan dengan *file* bertipe HTML. Dengan menggunakan PHP Anda bisa membuat *website* yang dinamis dengan disertai manajemen *database*-nya (Hidayatullah, 2017).

#### 2.3.5. Framework

*Framework* adalah komponen pemrograman yang siap *re-use* (bisa digunakan ulang) kapan saja, sehingga *programmer* tidak harus membuat skrip yang sama untuk tugas yang sama. Misalkan *programmer* ingin halaman-halaman web menampilkan data dengan paginasi (*paging*) halaman, *framework* telah menyediakan fungsi *paging* tersebut sedangkan *programmer* cukup menggunakan fungsi tersebut pada saat *coding*, tetapi tentu dengan kaidah-kaidah yang ditetapkan oleh masing-masing *framework* (Kasman, 2015).

#### 2.3.6. Bootstrap

Bootstrap merupakan salah satu *framework* CSS paling populer dari sekian banyak *framework* CSS yang ada. Bootstrap memungkinkan desain sebuah *website* menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran perangkat dengan tampilan yang menarik. Bootstrap juga membuat proses pengaturan desain menjadi lebih cepat karena tidak perlu bagi menulis CSS. Bootstrap telah didukung hampir semua *browser* pada setiap *desktop* maupun *mobile*. (Abdulloh, 2018).

### 2.3.7. Laravel

Laravel adalah sebuah *framework* web berbasis PHP yang *open-source* dan tidak berbayar, dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC (*Model-View-Controller*). Struktur pola MVC pada Laravel sedikit berbeda pada struktur pola MVC pada umumnya. Di Laravel terdapat *routing* yang menjembatani antara *request* dari *user* dan *controller*. Jadi *controller* tidak langsung menerima *request* tersebut (Yudanto dkk, 2017).

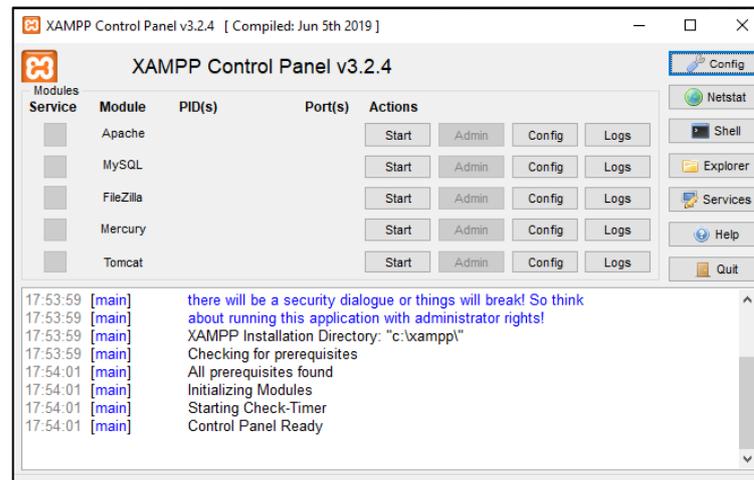
### 2.3.8. MySQL

MySQL merupakan sebuah *Relational Database Management System* (RDBMS) yang bersifat *open-source*. Perangkat lunak *database* pada umumnya disandingkan dengan bahasa pemrograman *server web* seperti PHP atau JSP. MySQL (*My Structured Query Language*) adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut dengan DBMS (*Database Management System*), sifat DBMS ini ialah *open-source* (Josi, 2017).

Salah satu keunggulan MySQL adalah kemudahan penggunaan dan pengelolaannya, untuk mengakses *database* MySQL dapat dilakukan dengan menggunakan berbagai MySQL *client*. Selain menggunakan MySQL *client* bawaan berbasis *command line*, saat ini banyak sekali *tools* yang dikembangkan oleh pihak ketiga untuk mempermudah pengelolaan *database* MySQL (Solichin, 2016).

### 2.3.9. XAMPP

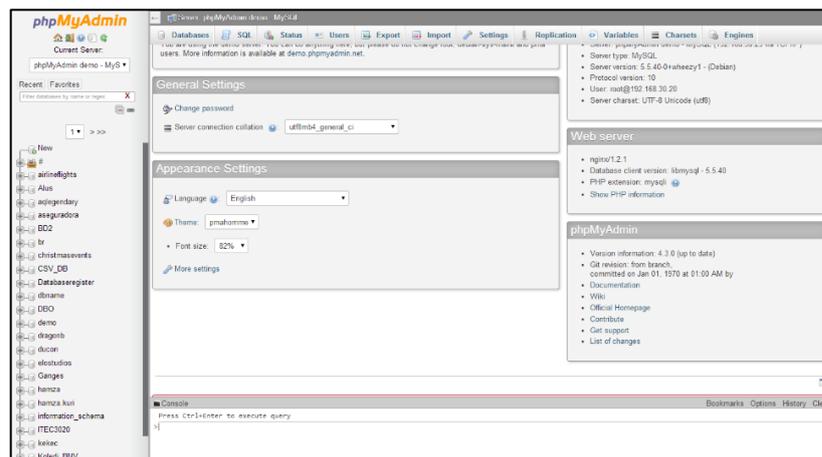
XAMPP adalah perangkat lunak gratis, yang mendukung banyak sistem operasi. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL, *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (*General Public License*) dan gratis (Hakim, 2008).



**Gambar 2.1** Tampilan XAMPP Control Panel  
(Sumber: Dokumen Pribadi)

### 2.3.10. PHPMYAdmin

*PHPMyAdmin* adalah sebuah perangkat lunak bebas (*open-source*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi *database* MySQL melalui jaringan lokal maupun *internet*. *PHPmyAdmin* mendukung berbagai operasi MySQL. Diantaranya (mengelola basis data, tabel-tabel, *fields*, relasi (*relations*), indeks, pengguna (*user*), perijinan (*permissions*), dan lain-lain) (Rozaq, 2015).

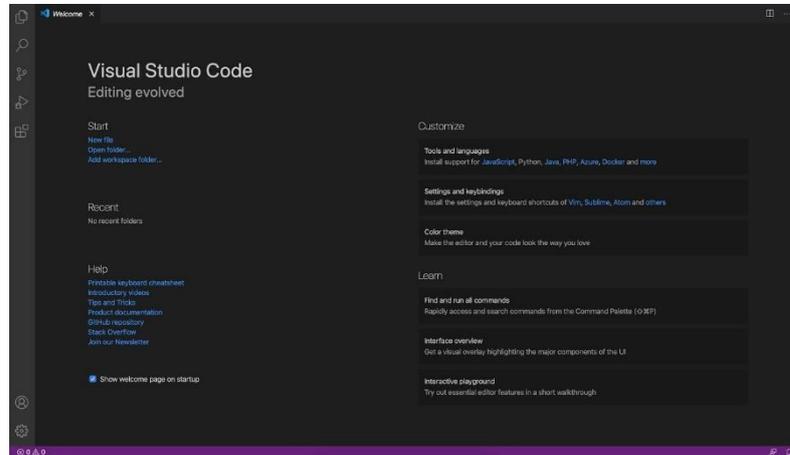


**Gambar 2.2** Tampilan Dashboard PHPMYAdmin  
(Sumber: Dokumen Pribadi)

### 2.3.11. Visual Studio Code

*Visual Studio Code* atau *VS Code* adalah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga

untuk versi Linux, Mac, dan Windows. *Visual Studio Code* mendukung banyak bahasa pemrograman. (Munari, 2020).



**Gambar 2.3** Tampilan *Visual Studio Code*  
(Sumber: Dokumen Pribadi)