

BAB II

LANDASAN TEORI

2.1 Profil Perusahaan

PT. Pelabuhan Indonesia II (persero) cabang panjang (2019). Pada abad ke-17, pemerintah hindia belanda membangun pelabuhan panjang dengan dermaga sepanjang 200 meter, menggunakan konstruksi *caisson* dengan kedalaman -7 m.LWS beserta satu unit gudang seluas 1.000 m². Pelabuhan panjang kemudian menjadi pelabuhan besar dipulau sumatra dan berperan sebagai urat nadi pertumbuhan ekonomi di provinsi Lampung hingga saat ini.

Berada pada titik persilangan pulau sumatra dan pulau jawa , pelabuhan panjang merupakan pelabuhan yang sangat prospektif dimasa mendatang. Luas areal lahan pelabuhan masih dapat dikembangkan untuk kejasama pembangunan berbagai terminal guna melayani kebutuhan pengguna jasa kepelabuhan. Disamping itu lahan pertanian dan lahan perkebunan yang membentang di privinsi Lampung sangat subur untuk kegiatan agrobisnis.

Tersedianya jalan akses yang menghubungkan lokasi pelabuhan dengan hinterlandnya memudahkan transportasi dan distribusi berbagai komoditi hasil pertambangan dan agrobisnis baik untuk keperluan ekspor,impor maupun domestik.

Saat ini lingkungan hinterland tersebut masih menunggu para investor untuk dapat mengembangkannya secara optimal. Pelabuhan penjang melayani kapal dengan berbagai jenis barang, seperti barang umum, barang dalam kantung, curah cair, curah kering, dan petikemas. Arus petikemas terus tumbuh setiap tahunnya seiring pertumbuhan kegiatan industri, pertambangan dan perkebunan diwilayah itu. Dengan tersedianya petikemas yang dilengkapi dengan 3 *container crane*, 5 *transtainer*, dan *top leader* serta dukungan terminal khusus curah yang ada saat

ini, pelabuhan panjang siap memberikan pelayanan yang lebih baik kepada pengguna jasa.

2.2 Definisi Sistem Informasi

Menurut Winda Aprianti dan Umi Maliha (2016). Sistem informasi adalah kumpulan dari sub sistem yang saling berhubungan satu sama lain dan bekerja sama secara harmonis, yang bersifat untuk mendukung pembuatan keputusan dan pengawasan dalam organisasi. pengambilan keputusan yang berguna mempunyai nilai nyata yang dapat dirasakan akibatnya baik pada saat itu juga maupun dimasa mendatang, mendukung kegiatan operasional, manajerial, dan strategis organisasi, dengan memanfaatkan berbagai sumber daya yang ada dan tersedia bagi fungsi tersebut guna mencapai tujuan.

2.3 Konsep Dasar Sistem Informasi

Agustina Simangunsong (2018) menjelaskan, Sistem merupakan kumpulan elemen -elemen yang saling terkait dan bekerjasama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan suatu sistem pada dasarnya sekelompok unsur - unsur yang erat hubungannya satu dengan yang lain berfungsi untuk mencapai tujuan tertentu. Sistem informasi memiliki makna sistem yang bertujuan menampilkan informasi.

Ketersediaan komputer telah menjadikan peran informasi berbeda dari zaman sebelumnya, komputer dapat dimanfaatkan untuk mengelola informasi dalam jumlah yang sangat besar dan dalam waktu yang tepat. Saat informasi dikelola secara manual, jumlah dan peran informasi tidak seluar biasa seperti sekarang. Informasi dikelola dalam bentuk sistem, sehingga disebut sistem informasi.

2.4 Definisi Kerjasama

Menurut Risa Kartika Lubis (2015). kerjasama merupakan salah satu faktor pendukung yang penting bagi setiap kegiatan maupun aktifitas seorang manajer dalam mengemban tugasnya sehari – hari untuk mencapai tujuan organisasi yang telah ditetapkan. Komunikasi sebagai media kerjasama merupakan suatu alat yang menyampaikan pesan dari satu pihak ke pihak lain sehingga yang bersangkutan mengerti dan menyadari secara persis apa yang akan diharapkan darinya pada saat itu maupun pada saat yang akan datang.

Kerjasama yaitu suatu strategi bisnis yang dilakukan oleh dua pihak atau lebih dalam jangka waktu tertentu, kerjasama dapat terjadi ketika pihak yang bersangkutan memiliki kepentingan dan kesadaran yang sama untuk mencapai tujuan dan kepentingan bersama.

2.5 Konsep Dasar Basis Data

Dede Wira Trise Putra dan Jendra Jonika Putra (2018) menjelaskan, Basis data terdiri dari dua kata yaitu basis dan data. Basis bisa diartikan sebagai markas atau gudang, tempat berkumpul. Sedangkan data adalah representasi factor dunia nyata yang mewakili suatu obyek seperti manusia (pegawai, mahasiswa, pembeli, pelanggan, barang, peristiwa dan lainnya) yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. *Database* adalah kumpulan data / informasi yang teratur berdasarkan kriteria tertentu yang saling berhubungan.

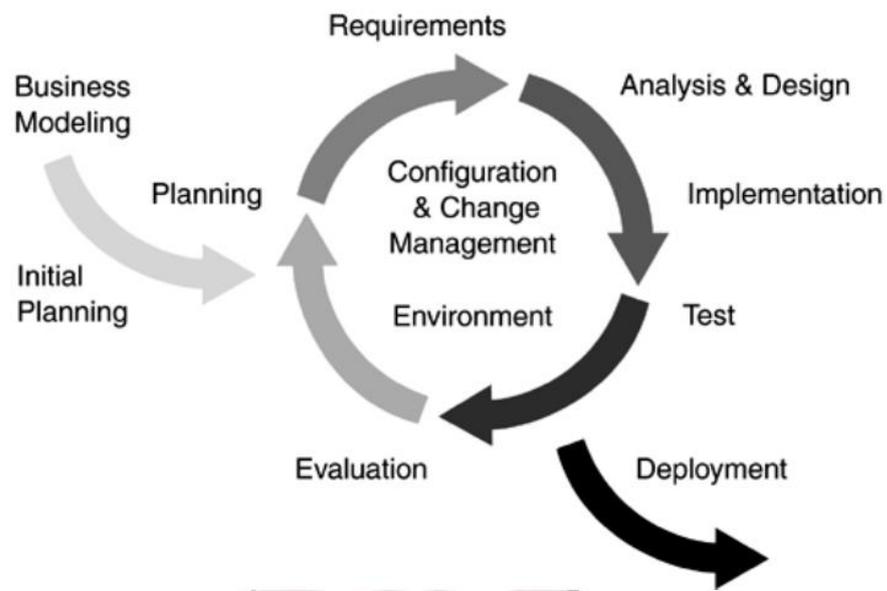
2.6 Definisi Website

Menurut Agustina Simangunsong (2018). *Website* adalah keseluruhan halaman halaman yang terdapat dari sebuah domain yang mengandung informasi atau yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, dan

video). Domain adalah nama unik yang dimiliki oleh sebuah institusi sehingga bisa diakses melalui internet.

2.7 Metode Pengembangan Sistem

Pengembangan sistem informasi yang dilakukan yaitu menggunakan *Rational Unified Process* (RUP). Proses pengulangan / *intetif* pada RUP secara global dapat dilihat pada gambar 2.1.



Gambar 2.1 Proses *Rational Unified Process* (RUP)

Menurut Rosa A. S., dan Shalahuddi M. (2014). *Rational Unified Process* (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*) lebih diarahkan berdasarkan menggunakan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). berikut adalah penjelasan untuk setiap

fase pada *Rational Unified Process* (RUP) dalam pengembangan sistem informasi kerjasama vendor :

a. *Inception*

Pada tahap ini penulis menentukan ruang lingkup pengembangan sistem dari hasil wawancara dan observasi yang dilakukan, meliputi dari hasil penelitian atau skripsi-skripsi terdahulu.

b. *Elaboration*

Pada tahap *elaboration* hasil dari observasi dan wawancara dilanjutkan dengan melakukan identifikasi masalah pada sistem yang dibuat. Didalam *elaboration* terdapat dua tahap yaitu :

- Analisis

Terdapat tiga fase dalam tahapan analisis sistem pada alur pengembangan sistem RUP, yaitu analisis permasalahan, dan analisis persyaratan.

- Perancangan

Pada tahap perancangan terdiri dari : perancangan aplikasi, menggunakan diagram *Unified Modelling Language* (UML) meliputi *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

c. *Construction*

Pada tahap ini menjelaskan bagaimana mengimplementasi dan melakukan uji coba terhadap aplikasi yang telah dibuat. Dalam tahapan implementasi dijelaskan perangkat keras dan perangkat lunak apa saja yang dibutuhkan untuk mengimpelementasi apikasi yang telah dibuat. Sedangkan pada tahap uji coba dilakukan *testing*. *Testing* diperlukan untuk menjamin

kualitas aplikasi yang telah dibuat apakah telah sesuai dengan yang diharapkan.

d. *Transition*

pada tahap *transition* yaitu membuat panduan penggunaan aplikasi yang telah dibuat.

2.8 Konsep dasar berorientasi objek

Menurut Lis Suryadi dan Johan Wahyudi (2018). Berorientasi obyek atau *object oriented* merupakan paradigma. Paradikma ini memandang sistem sebagai kumpulan obyek-obyek diskrit yang saling berinteraksi satu sama lain. Berorientasi obyek ini juga bermakna kegiatan mengorganisasikan perangkat lunak sebagai kumpulan obyek-obyek diskrit yang bekerja sama antara informasi (struktur data) dan perilaku yang mengaturnya.

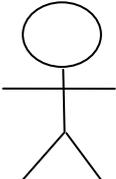
2.9 Alat Perancangan Sistem

Alat bantu perancangan yang digunakan dalam penelitian ini adalah *Unified Modelling Language* (UML).

Menurut Faizal Ari Prabowo dan Mamay Syani (2017). *Unified Modelling Language* (UML) adalah bahasa untuk visualisasi, spesifikasi, membangun sistem perangkat lunak, serta dokumentasi . UML menyediakan model-model yang tepat, tidak ambigu, dan lengkap, secara khusus UML menspesifikasi langkah-langkah penting dalam pengembangan keputusan analisis, perancangan, serta implementasi dalam sistem perangkat lunak.

2.9.1 Use Case Diagram

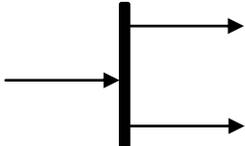
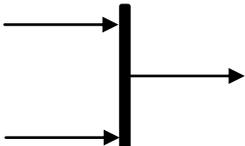
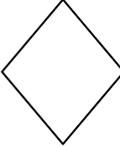
Menurut Dede Wira Trise Putra dan Rahmi Andriani (2019). *Use Case Diagram* merupakan pemodelan untuk melakukan sistem informasi yang akan dibuat. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai. Simbol yang digunakan adalah sebagai berikut :

Simbol	Penjelasan
Simbol <i>use case</i> 	Berisi kejadian yang berhubungan dengan database
simbol aktor 	Adalah orang / bagian (<i>database external</i>) yang berhubungan dengan sistem
Simbol <i>system boundary</i> 	Adalah sebagai wadah yang berisi sekumpulan <i>use case</i>
simbol <i>communicates</i> 	Berfungsi untuk menghubungkan <i>actor</i> dengan <i>use case</i>

Tabel 2.1 *use case diagram*

2.9.2 Activity Diagram

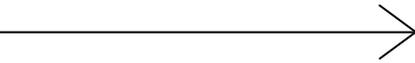
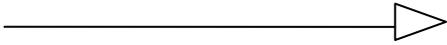
Menurut Dede Wira Trise Putra dan Rahmi Andriani (2019). *Activity Diagram* merupakan program yang menggambarkan *workflow* atau aktivitas dari sebuah sistem yang ada dari perangkat lunak. Berikut simbol yang digunakan pada *activity diagram*.

Simbol	Keterangan
	<i>Start point</i>
	<i>End point</i>
	<i>Activities</i>
	<i>Fork (percabangan)</i>
	<i>Join (penggabungan)</i>
	<i>Decision</i>

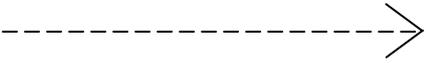
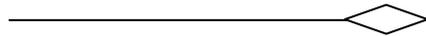
Tabel 2.2 *activity diagram*

2.9.3 Class Diagram

Menurut Dede Wira Trise Putra dan Rahmi Andriani (2019). *Class* merupakan gambaran struktur sistem dari segi pedefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* terdiri dari atribut dan operasi dengan tujuan dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan *obyek* beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Simbol	Keterangan
Kelas 	Kelas pada struktur system
antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .
asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan dengan kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> .
generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).

Tabel 2.3 *class diagram*

Simbol	Keterangan
kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian.

Tabel 2.3 *class diagram* (lanjutan)

2.9.4 *Sequence Diagram*

Menurut Dede Wira Trise Putra dan Rahmi Andriani (2019). *Sequence diagram* menggambarkan aktifitas obyek pada *use case* dengan mendeskripsikan waktu hidup obyek dan pesanyang dikirim dan diterima antar obyek. Gambaran *Sequence diagram* dibuat minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. *Sequence diagram* biasa digunakan untuk menggambarkan *scenario* atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan *output* tertentu.

2.10 Perangkat Lunak Pendukung

Perangkat lunak pendukung merupakan *software* yang akan digunakan dalam proses analisis dan perancangan sistem yang diusulkan pada, diantaranya sebagai berikut:

2.10.1 HTML5

Mhd Arief Hasan dan Nurliana Nasution (2018) menjelaskan, HTML5 merupakan salah satu karya Konsortium Waring Wera Wanua (World Wide Web Consortium, W3C) untuk mendefinisikan sebuah bahasa markah tunggal yang

dapat ditulis dengan cara HTML ataupun XHTML. HTML5 memiliki fitur yang lebih efisien, simpel dan dapat mendukung semua perangkat serta dapat beradaptasi dengan perubahan teknologi yang ada. Untuk memanfaatkan fitur-fitur HTML5 yaitu menggunakan peramban *web* yang modern.

2.10.2 Java Script

Menurut sidik dalam (Agus Prayitno, 2015) menjelaskan bahwa “Java Script” adalah bahasa pemrograman yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan dalam *browser* menjadi lebih interaktif, tidak sekedar indah saja”.

2.10.3 PHP (*HyperText Preprocessor*)

Menurut Asep Toyib Hidayat (2019), PHP (*HyperText Preprocessor*) yang digunakan sebagai bahasa *script server-side* dalam pengembangan web yang disisipkan dalam dokumen HTML. Penggunaan PHP memungkinkan web dapat dinamis sehingga *maintenance* situs web tersebut menjadi lebih mudah dan efisien. Kode-kode PHP akan diinterpretasikan pada *server web* dan menghasilkan HTML atau *output* lainnya yang akan dilihat oleh pengguna *web*.

2.10.4 MySQL

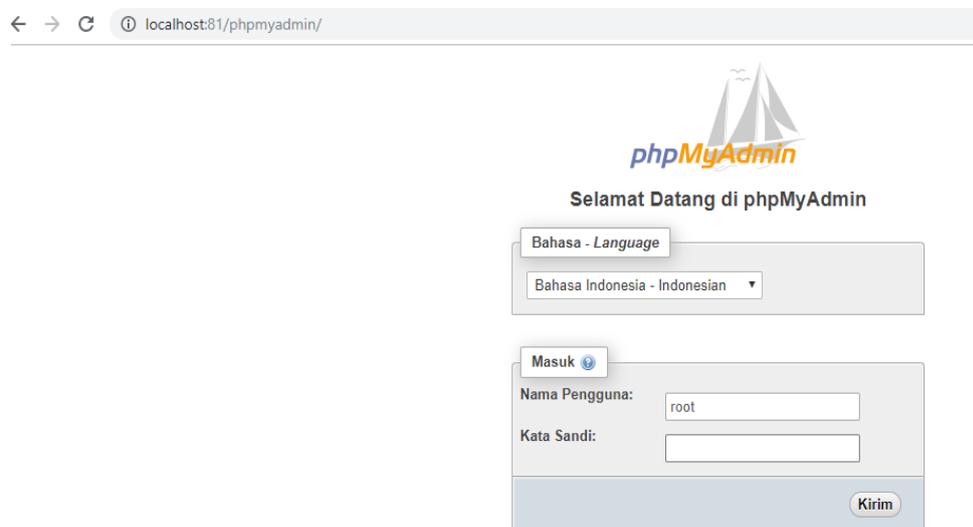
Menurut Sri Lestari dan Ardina Desi Susana (2016). MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya SQL (Structur Query Language). SQL adalah sebuah konsep pengoperasian basisdata,

terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.10.5 PhpMyAdmin

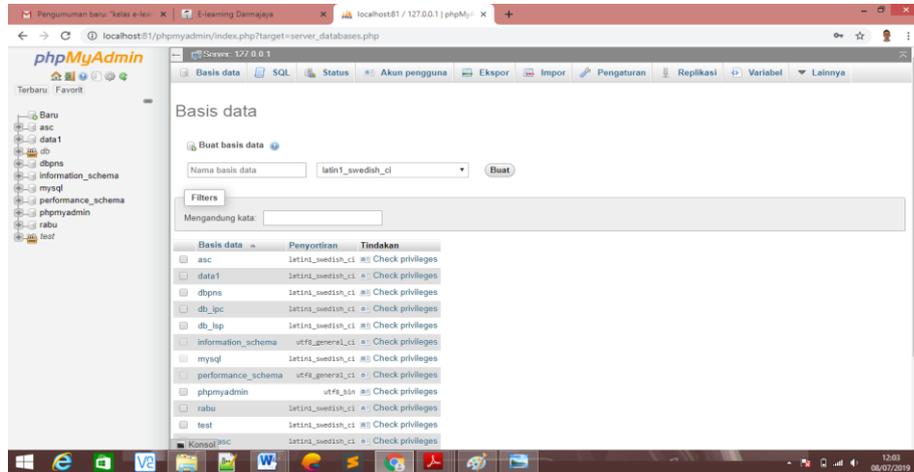
Agustina Simangunsong (2018) menjelaskan, PhpMyAdmin adalah aplikasi web yang dibuat oleh phpmyadmin.net. PhpMyAdmin digunakan untuk administrasi *database* MySQL. Berikut adalah langkah-langkah menggunakan PhpMyAdmin :

- Langkah pertama, yaitu membuka web *browser* yang akan digunakan (firefox atau Google Chrome).
- Untuk masuk ke PhpMyAdmin ketikkan <http://localhost/phpmyadmin> pada web *browser* kemudian tekan enter.



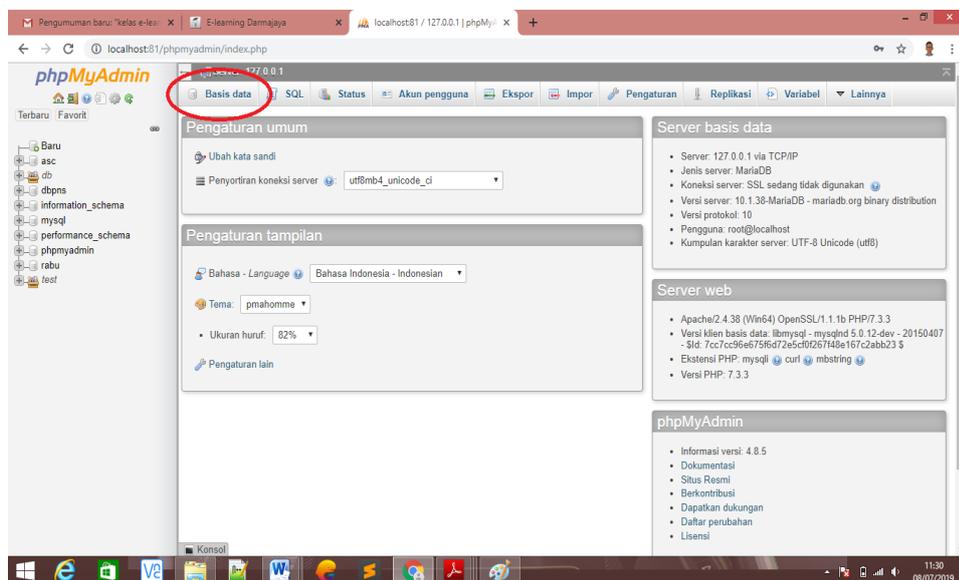
Gambar 2.2 login phpmyadmin.

- Akan muncul halaman login, setelah login maka akan masuk ke halaman PhpMyAdmin.



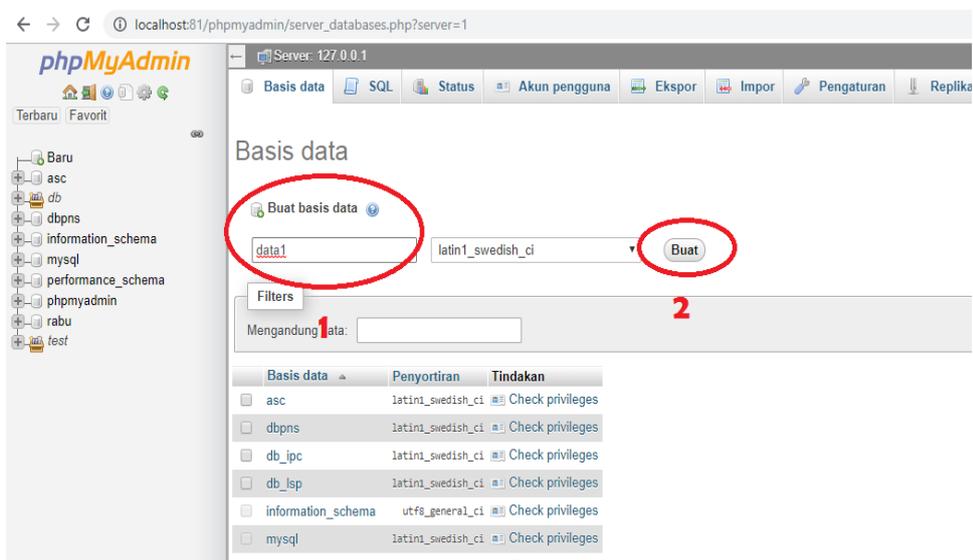
Gambar 2.3 dashboard phpmyadmin.

- Untuk membuat *database* baru, klik basisdata, dan isikan sesuai dengan nama *database* yang akan dibuat.



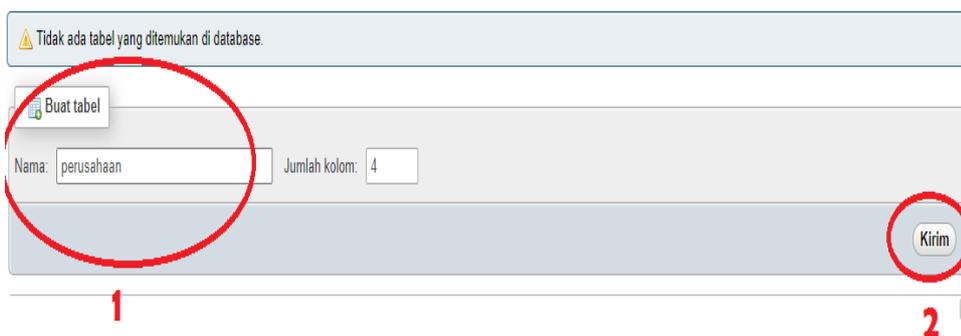
Gambar 2.4 membuat *database*.

- Setelah mengisi nama *database*, kemudian klik buat.



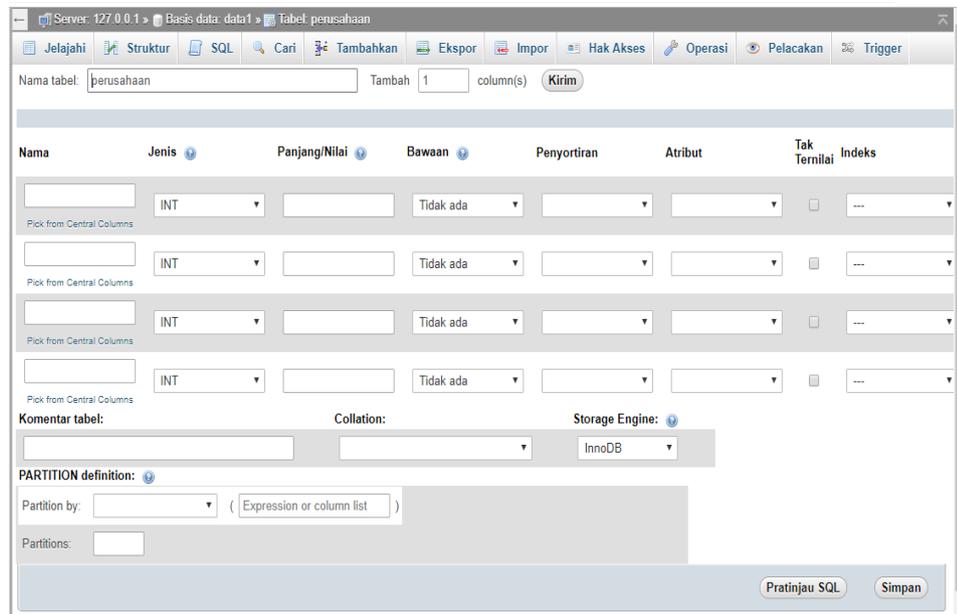
Gambar 2.5 memberi nama *database*.

- Langkah selanjutnya yaitu membuat tabel, isikan nama tabel yang akan dibuat kemudian klik kirim.



Gambar 2.6 membuat tabel.

- Maka akan tampil halaman seperti berikut ini.



Gambar 2.5 membuat tabel (lanjutan)

- Kemudian isikan nama *field*, tipe data, *length*, *primary key*, lalu klik *button* simpan.

2.10.6 Laravel

Menurut Naista (2017). *framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* akan lebih singkat dan memudahkan dalam melakukan perbaikan. Laravel adalah *framework* berbasis PHP yang sifatnya open *source*, dan menggunakan konsep *model-view-controler*, laravel berada dibawah lisensi MIT *licence* dengan menggunakan github sebagai tempat berbagai *code* menjalankannya. Berikut adalah dasar-dasar dalam penggunaan laravel:

a. Artisan

Artisan adalah *command line* atau perintah yang dijalankan melalui terminal dan disediakan beberapa perintah yang dapat digunakan selama melakukan pengembangan dan pembuatan aplikasi. Salah satu fungsi dari php artisan yaitu "*php artisan serve*". *php artisan serve* berfungsi untuk membuka *website* yang telah dibuat tanpa menggunakan *web server* lokal.

b. Routing

Routing adalah suatu proses yang bertujuan agar suatu *item* yang ingin disampaikan sampai ke tujuan. Dengan menggunakan *routing* dapat ditentukan halaman halaman yang akan muncul ketikadibuka oleh *user*. Pengaturan *routing* di laravel terletak pada file *web.php*.

c. Controller

Controller adalah suatu proses yang bertujuan untuk mengambil permintaan, menginisialisasi, memanggil model untuk dikirim ke *view*. Ada dua cara Membuat *Controller* di laravel. Cara pertama adalah dibuat *file controller* secara manual dan dituliskan *code extends Controller* didalamnya. Cara kedua adalah dibuat *file Controller* dengan menggunakan *command line* dengan menuliskan "*php artisan make:controller filename_Controller*". permintaan yang dibuat dalam laravel harus berada didalam *Controller*, kemudian diberikan melalui *routing* untuk mendapat permintaan yang diinginkan.

d. View (*blade templating*)

Blade adalah *template engine* bawaan dari laravel. *Blade* memiliki kode-kode yang lebih mudah untuk menghasilkan laravel. Cara membuat file *blade* dilakukan dengan cara manual dengan membuat *filename.blade.php* didalam folder *views*.

e. *Session*

Session adalah sebuah cara yang digunakan untuk penyimpanan pada *server* dan penyimpanan tersebut digunakan pada beberapa halaman termasuk halaman itu sendiri. Terdapat dua cara dalam penggunaan *session*. Cara yang pertama yaitu dengan menggunakan *request*. Cara yang kedua yaitu dengan menggunakan fungsi *global helper session*.