****

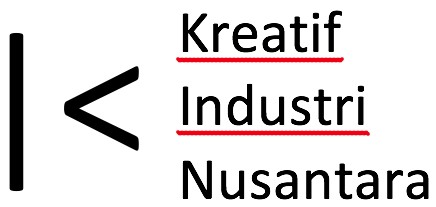
**PERANGANCAN SISTEM PERAMALAN MENGGUNAKAN REGRESI LINIER BERGANDA**

**PERANGANCAN SISTEM PERAMALAN MENGGUNAKAN REGRESI LINIER BERGANDA**

**Cahyo Prianto, S.Pd., M.T.**

**Imron Sumadireja**

Informatics Research Center



**Kreatif Industri Nusantara**

Penulis:

Cahyo Prianto, S.Pd., M.T.

Imron Sumadireja

ISBN : 978-623-7898-26-9

Editor:

Imron Sumadireja

Penyunting:

Cahyo Prianto, S.Pd., M.T.

Imron Sumadireja

Desain sampul dan Tata letak:

Alit Fajar Kurniawan

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : [awangg](mailto:awangga@kreatif.co.id)[a@kreatif.co.id](mailto:a@kreatif.co.id)

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : [irc@poltekpos.ac.id](mailto:irc@poltekpos.ac.id)

Cetakan Pertama, 2020

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara

apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat*

*menahan lelahnya*

*belajar, Maka kamu harus*

*sanggup menahan*

*perihnya Kebodohan.’*

*Imam Syafi’i*

CONTRIBUTORS

Cahyo Prianto, Imron Sumadireja,

Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesi

# 

CONTENTS IN BRIEF

|  |  |  |
| --- | --- | --- |
| **1** | **Latar Belakang Masalah** | **1** |
| **2** | **Landasan Teori** | **4** |
| **3** | **Instalasi Aplikasi Yang Digunakan** | **19** |
| **4** | **Perancangan Sistem** | **45** |
| **5** | **Penerapan Metode Regresi Linier Berganda** | **59** |
| **6** | **Tutorial Membuat Sistem Peramalan** | **68** |
| **7** | **Panduan Dalam Menggunakan Sistem Peramalan** | **184** |

# DAFTAR ISI

[Daftar Isi ix](#_Toc34293749)

[Daftar Gambar xii](#_Toc34293750)

[Daftar Tabel xvi](#_Toc34293751)

[Listings xvii](#_Toc34293752)

[Foreword xx](#_Toc34293753)

[Kata Pengantar xxi](#_Toc34293754)

[Acknowledments xxiii](#_Toc34293755)

[Acronyms xxv](#_Toc34293756)

[Introduction xvi](#_Toc34293757)

[Latar Belakang Masalah 1](#_Toc34293758)

[Landasan Teori 4](#_Toc34293759)

[*Hypertext Preprocessor(PHP)* 4](#_Toc34293760)

[*Object Oriented Programming* 10](#_Toc34293761)

[*CodeIgniter* 13](#_Toc34293762)

[Instalasi Aplikasi Yang Digunakan 19](#_Toc34293763)

[*Visual Studio Code* 19](#_Toc34293764)

[*Xampp* 23](#_Toc34293765)

*Package* [*Codeigniter* 28](#_Toc34293766)

[Perancangan Sistem 45](#_Toc34293769)

[Metode Regresi Linier Berganda 59](#_Toc34293784)

[Tutorial Membuat Sistem Peramalan 68](#_Toc34293785)

[Panduan Dalam Menggunakan Sistem Peramalan 198](#_Toc34293786)

[DAFTAR PUSTAKA 201](#_Toc34293787)

# 

# DAFTAR GAMBAR

[Gambar 1. 1 Digital Marketing. 2](#_Toc35528111)

[Gambar 1. 2 Sales. 3](#_Toc35528112)

[Gambar 2. 1 Rasmus Lerdorf. 7](#_Toc35528119)

[Gambar 2. 2 Perbandingan PHP dengan Bahsa Scripting Server-Side. 10](#_Toc35528120)

[Gambar 2. 3 Logo CodeIgniter. 14](#_Toc35528121)

[Gambar 2. 4 Konsep MVC. 15](#_Toc35528122)

[Gambar 2. 5 Alur Kerja Framework CodeIgniter. 16](#_Toc35528123)

[Gambar 3. 1 Download Visual Studio Code. 20](#_Toc35528124)

[Gambar 3. 2 Instalasi Visual Studio Code. 20](#_Toc35528125)

[Gambar 3. 3 Select Additional Tasks. 21](#_Toc35528126)

[Gambar 3. 4 Ready To Install. 21](#_Toc35528127)

[Gambar 3. 5 Proses Installasi. 22](#_Toc35528128)

[Gambar 3. 6 Installasi Berhasil. 22](#_Toc35528129)

[Gambar 3. 7 Menu Utama Visual Studio Code. 23](#_Toc35528130)

[Gambar 3. 8 Download File XAMPP. 24](#_Toc35528131)

[Gambar 3. 9 Setup Installasi XAMPP. 25](#_Toc35528132)

[Gambar 3. 10 Select Component. 25](#_Toc35528133)

[Gambar 3. 11 Installation Folder. 26](#_Toc35528134)

[Gambar 3. 12 Ready To Install. 26](#_Toc35528135)

[Gambar 3. 13 Proses Installasi. 27](#_Toc35528136)

[Gambar 3. 14 Installasi Berhasil. 27](#_Toc35528137)

[Gambar 3. 15 Pilih Bahasa Yang Akan Digunakan. 28](#_Toc35528138)

[Gambar 3. 16 Menu Utama XAMPP. 28](#_Toc35528139)

[Gambar 3. 17 Download File CodeIgniter. 29](#_Toc35528140)

[Gambar 3. 18 Ekstrak File CodeIgniter. 29](#_Toc35528141)

[Gambar 3. 19 Copy File Ke Dalam Direktori htdocs. 29](#_Toc35528142)

[Gambar 3. 20 Menjalankan Apache dan MySQL. 30](#_Toc35528143)

[Gambar 3. 21 Gambar Utama Dari CodeIgniter. 30](#_Toc35528144)

[Gambar 3. 22 Struktur Dari CodeIgniter. 31](#_Toc35528145)

[Gambar 3. 23 Konfigurasi CodeIgniter. 33](#_Toc35528146)

[Gambar 3. 24 Konfigurasi Autoload.php 33](#_Toc35528147)

[Gambar 3. 25 Konfigurasi Autoload Bagian Libraries. 33](#_Toc35528148)

[Gambar 3. 26 Menambahkan Database. 34](#_Toc35528149)

[Gambar 3. 27 Konfigurasi Autoload Helper. 34](#_Toc35528150)

[Gambar 3. 28 Buka File Config.php 34](#_Toc35528151)

[Gambar 3. 29 Konfigurasi Base\_Url. 34](#_Toc35528152)

[Gambar 3. 30 Menambahkan Base\_URL. 35](#_Toc35528153)

[Gambar 3. 31 Buka File Database.php 35](#_Toc35528154)

[Gambar 3. 32 Konfigurasi File Database.php 35](#_Toc35528155)

[Gambar 3. 33 Menambahkan Atribut Pada Database. 36](#_Toc35528156)

[Gambar 3. 34 Hasil Dari Syntax Hello 1. 38](#_Toc35528157)

[Gambar 3. 35 Syntax Untuk Memuat Model. 39](#_Toc35528158)

[Gambar 3. 36 Syntax Untuk Mengakses Objek Pada Model. 39](#_Toc35528159)

[Gambar 3. 37 Syntax Untuk Mendapatkan Data Dari Model. 40](#_Toc35528160)

[Gambar 3. 38 Hasil Dari Syntax Hello 2. 40](#_Toc35528161)

[Gambar 3. 39 Hasil Dari Syntax Hello 3. 41](#_Toc35528162)

[Gambar 3. 40 Syntax Untuk Memanggil View. 41](#_Toc35528163)

[Gambar 3. 41 Hasil Dari Syntax Hello 4. 43](#_Toc35528164)

[Gambar 4. 1 Flowmap Login. 46](#_Toc35528165)

[Gambar 4. 2 Flowmap prosedur kelola data addOn. 47](#_Toc35528166)

[Gambar 4. 3 Flowmap Prosedur Penerapan Algoritma Regresi Linier Berganda. 48](#_Toc35528167)

[Gambar 4. 4 Flowmap Prosedur Kelola Akun. 49](#_Toc35528168)

[Gambar 4. 5 Use case diagram peramalan pendapatan. 51](#_Toc35528169)

[Gambar 4. 6 Class Diagram. 57](#_Toc35528170)

[Gambar 4. 7 Component Diagram. 58](#_Toc35528171)

[Gambar 4. 8 Deployment Diagram. 58](#_Toc35528172)

[Gambar 5. 1 Persamaan Regresi Linier Berganda 60](#_Toc35528173)

[Gambar 5. 2 Persamaan Koefiein Determinasi. 60](#_Toc35528174)

[Gambar 5. 3 Persamaan Koefisien Korelasi. 61](#_Toc35528175)

[Gambar 5. 4 Data Utama. 61](#_Toc35528176)

[Gambar 5. 5 Hasil Perhitungan RLB. 62](#_Toc35528177)

[Gambar 5. 6 Rumus Mencari Matriks. 62](#_Toc35528178)

[Gambar 5. 7 Hasil Matriks. 63](#_Toc35528179)

[Gambar 5. 8 Hasl Determinan. 63](#_Toc35528180)

[Gambar 5. 9 Hasil Nilai a, b1, dan b2. 64](#_Toc35528181)

[Gambar 5. 10 Hasil Perhitungan RLB. 64](#_Toc35528182)

[Gambar 5. 11 Persamaan Regresi Linier Sederhana. 65](#_Toc35528183)

[Gambar 5. 12 Persamaan Koefisien a dan b. 65](#_Toc35528184)

[Gambar 6. 1 Hasil dari syntax Auth.php. 80](#_Toc35528185)

[Gambar 6. 2 Hasil Dari Syntax Untuk Membuat Judul. 96](#_Toc35528186)

[Gambar 6. 3 Hasil Dari Syntax View Dashboard Admin. 97](#_Toc35528187)

[Gambar 6. 4 Hasil Dari Syntax Membuat Button. 115](#_Toc35528188)

[Gambar 6. 5 Hasil Dari Syntax View Tabel Data. 117](#_Toc35528189)

[Gambar 6. 6 Hasil Dari Syntax View Form Upload Data. 118](#_Toc35528190)

[Gambar 6. 7 Hasil Dari Syntax View List Indieland. 119](#_Toc35528191)

[Gambar 6. 8 Hasil Dari Syntax View Membuat TextBox dan Button. 134](#_Toc35528192)

[Gambar 6. 9 Syntax View Untuk Membuat Tabel Data. 135](#_Toc35528193)

[Gambar 6. 10 Hasil Dari Syntax Form Upload Data Pelanggan. 137](#_Toc35528194)

[Gambar 6. 11 Hasil Dari Syntax Membuat Button. 151](#_Toc35528195)

[Gambar 6. 12 Hasil Dari Syntax Membuat Tabel. 153](#_Toc35528196)

[Gambar 6. 13 Hasil Dari Syntax Form Input Data Pegawai. 154](#_Toc35528197)

[Gambar 6. 14 Hasil Dari Syntax Detail Akun Pegawai. 156](#_Toc35528198)

[Gambar 6. 15 Hasil Dari Syntax Halaman Edit Akun Pegawai. 159](#_Toc35528199)

[Gambar 6. 16 Hasil Syntax Untuk Membuat Tab. 176](#_Toc35528200)

[Gambar 6. 17 Hasil Dari Tabel Pada Tab Pertama. 178](#_Toc35528201)

[Gambar 6. 18 Syntax Untuk Tabel Kedua dan Perhitungan RLB. 180](#_Toc35528202)

[Gambar 6. 19 Hasil Dari Syntax Untuk Menampilkan Nilai Determinan, a, b1, dan b2. 181](#_Toc35528203)

[Gambar 6. 20 Hasil Dari Syntax Untuk Menghitung Dalam RLB. 183](#_Toc35528204)

[Gambar 6. 21 Chart Data Keseluruhan. 193](#_Toc35528205)

[Gambar 6. 22 Chart Untuk Data Total Pelanggan. 195](#_Toc35528206)

[Gambar 6. 23 Chart Untuk Data Total Produk. 197](#_Toc35528207)

# DAFTAR TABEL

[Tabel 4. 1 Definisi Aktor. 51](#_Toc34252822)

[Tabel 4. 2 Skenario Use Case Login. 52](#_Toc34252823)

[Tabel 4. 3 Skenario Use Case Prediksi Pendapatan. 53](#_Toc34252824)

[Tabel 4. 4 Skenario Use Case Kelola Data addOn. 54](#_Toc34252825)

[Tabel 4. 5 Skenario Use Case Kelola Akun. 55](#_Toc34252826)

# LISTINGS

[Listings 3. 1 Syntax Controller Hello1. 37](#_Toc35528591)

[Listings 3. 2 Syntax Model Untuk Hello 2. 38](#_Toc35528592)

[Listings 3. 3 Syntax Controller Untuk Hallo 2. 39](#_Toc35528593)

[Listings 3. 4 Syntax View Pada Hello 3. 40](#_Toc35528594)

[Listings 3. 5 Syntax Controller Pada Hello 3. 41](#_Toc35528595)

[Listings 3. 6 Syntax Controller Untuk Hello 4. 43](#_Toc35528596)

[Listings 3. 7 Syntax Controller Pada Hello 4. 43](#_Toc35528597)

[Listings 3. 8 Syntax View Untuk Hello 4. 43](#_Toc35528598)

[Listings 6. 1 Syntax Controller Untuk Auth. 74](#_Toc35528609)

[Listings 6. 2 Syntax Untuk Validasi Akun Pada Sistem Peramalan. 75](#_Toc35528610)

[Listings 6. 3 Syntax Untuk Fungsi Validasi Dashboard. 76](#_Toc35528611)

[Listing 6. 4 Syntax Untuk Fungsi Logout. 76](#_Toc35528612)

[Listing 6. 5 Syntax Model Auth. 77](#_Toc35528613)

[Listing 6. 6 Syntax Fungsi Model Auth. 78](#_Toc35528614)

[Listing 6. 7 Syntax Untuk Membuat View Login. 79](#_Toc35528615)

[Listing 6. 8 Beberapa Fungsi Syntax Untuk Membuat Halaman Login. 80](#_Toc35528616)

[Listings 6. 9 Syntax Controller Untuk Dashboard Admin. 82](#_Toc35528617)

[Listings 6. 10 Syntax Untuk Validasi Akun. 82](#_Toc35528618)

[Listings 6. 11 Syntax Model Dashboard Admin. 85](#_Toc35528619)

[Listings 6. 12 Syntax Untuk Menghubungkan Dashboard Pada Database. 85](#_Toc35528620)

[Listings 6. 13 Syntax Untuk Function Indieland. 86](#_Toc35528621)

[Listings 6. 14 Syntax Function Movieyuk dan Indieland\_pantai. 86](#_Toc35528622)

[Listings 6. 15 Syntax View Pada Dashboard Admin. 95](#_Toc35528623)

[Listings 6. 16 Syntax Untuk Membuat Header Pada Dashboard Admin. 95](#_Toc35528624)

[Listings 6. 17 Syntax Untuk Membuat List Box. 96](#_Toc35528625)

[Listings 6. 18 Syntax Controller Indieland. 105](#_Toc35528626)

[Listings 6. 19 Syntax Untuk Meload Library. 105](#_Toc35528627)

[Listings 6. 20 Function Untuk Get Data Pada Database. 105](#_Toc35528628)

[Listings 6. 21 Syntax Untuk Import Data. 107](#_Toc35528629)

[Listings 6. 22 Syntax Untuk Cek Duplikasi. 108](#_Toc35528630)

[Listings 6. 23 Syntax Untuk Model Indieland. 112](#_Toc35528631)

[Listings 6. 24 Function Untuk Get Nomor Internet. 112](#_Toc35528632)

[Listings 6. 25 Syntax Model Untuk Cek Duplikasi Data. 112](#_Toc35528633)

[Listings 6. 26 Syntax View Indieland. 115](#_Toc35528634)

[Listings 6. 27 Syntax Untuk Membuat Button. 115](#_Toc35528635)

[Listings 6. 28 Syntax Untuk Membuat Form Tabel Data. 116](#_Toc35528636)

[Listing 6. 29 Syntax Untuk Membuat Form Upload Data. 118](#_Toc35528637)

[Listings 6. 30 Syntax Untuk View List Indieland. 118](#_Toc35528638)

[Listings 6. 31 Function Untuk Menampilkan Data Pada Sistem. 119](#_Toc35528639)

[Listings 6. 32 Syntax Controller Untuk Movieyuk. 123](#_Toc35528640)

[Listings 6. 33 Function Untuk Melakukan Load Library. 123](#_Toc35528641)

[Listings 6. 34 Syntax Untuk Validasi Sidebar. 124](#_Toc35528642)

[Listings 6. 35 Syntax Untuk Menampilkan Data Berdasarkan Periode. 124](#_Toc35528643)

[Listings 6. 36 Syntax Untuk Memberikan Fungsi Import File. 126](#_Toc35528644)

[Listings 6. 37 Syntax Untuk Memberikan Fungsi Download File. 127](#_Toc35528645)

[Listings 6. 38 Syntax Model Movieyuk. 129](#_Toc35528646)

[Listings 6. 39 Syntax Untuk Menampilkan Data Berdasarkan Periode.. 129](#_Toc35528647)

[Listings 6. 40 Syntax Untuk Memberikan Fungsi Download File Pada Sistem. 130](#_Toc35528648)

[Listings 6. 41 Syntax Untuk Update Data. 130](#_Toc35528649)

[Listings 6. 42 Syntax Lengkap Untuk View Movieyuk. 133](#_Toc35528650)

[Listings 6. 43 Syntax Untuk Membuat Textbox dan Button. 134](#_Toc35528651)

[Listings 6. 44 Syntax Untuk Membuat Tabel. 135](#_Toc35528652)

[Listings 6. 45 Syntax Untuk Membuat Form Upload Data. 136](#_Toc35528653)

[Listings 6. 46 Syntax View List Pada Movieyuk. 137](#_Toc35528654)

[Listinge 6. 47 Function Untuk Menampilkan Total Pelanggan. 138](#_Toc35528655)

[Listings 6. 48 Function Unutuk Menampilkan Data Berdasarkan Periode. 138](#_Toc35528656)

[Listings 6. 49 Syntax Untuk Download Data Movieyuk. 140](#_Toc35528657)

[Listings 6. 50 Syntax Untuk Membuat Judul Pada Kolom Data. 141](#_Toc35528658)

[Listings 6. 51 Syntax Controller Untuk Form Kelola Akun. 144](#_Toc35528659)

[Listings 6. 52 Syntax Untuk Validasi Akun Yang Login Pada Sistem. 144](#_Toc35528660)

[Listings 6. 53 Syntax Untuk Menambahkan Akun Baru Pada Sistem. 145](#_Toc35528661)

[Listings 6. 54 Syntax Untuk Membuat Aksi Hapus Data. 145](#_Toc35528662)

[Listings 6. 55 Syntax Untuk Mengubah Data Akun. 145](#_Toc35528663)

[Listings 6. 56 Syntax Untuk Melihat Detail Akun. 146](#_Toc35528664)

[Listings 6. 57 Syntax Untuk Model Kelola Data User. 147](#_Toc35528665)

[Listings 6. 58 Function Untuk Menghubungkan Sistem Dengan Database. 148](#_Toc35528666)

[Listings 6. 59 Syntax Untuk View Data User. 151](#_Toc35528667)

[Listings 6. 60 Syntax Untuk Membuat Judul Halaman dan Button. 151](#_Toc35528668)

[Listings 6. 61 Syntax Untuk Membuat Tabel. 153](#_Toc35528669)

[Listings 6. 62 Syntax Untuk Membuat Form Input Data User. 154](#_Toc35528670)

[Listings 6. 63 Syntax Untuk Membuat Form Detail User. 155](#_Toc35528671)

[Listings 6. 64 Function Untuk Menampilkan Detail Akun. 156](#_Toc35528672)

[Listings 6. 65 Syntax Untuk View Edit Akun. 158](#_Toc35528673)

[Listings 6. 66 Syntax Untuk Membuat Textbox Serta Button Pada Form Edit. 159](#_Toc35528674)

[Listings 6. 67 Syntax Controller Untuk Penerapan Regresi Linier Berganda. 161](#_Toc35528675)

[Listings 6. 68 Function Untuk Menampilkan Data Secara Keseluruhan Dari Database. 162](#_Toc35528676)

[Listings 6. 69 Function Untuk Import Data Pada Sistem. 163](#_Toc35528677)

[Listings 6. 70 Syntax Untuk Model Pada Penerapan Regresi Linier Berganda. 164](#_Toc35528678)

[Listings 6. 71 Syntax Untuk Cek Duplikasi dan Update Data. 165](#_Toc35528679)

[Listings 6. 72 Syntax Untuk Melakukan Perhitungan Matrix. 165](#_Toc35528680)

[Listings 6. 73 Syntax Lengkap Controller Untuk Penerapan Regresi Linier Berganda. 176](#_Toc35528681)

[Listings 6. 74 Syntax Untuk Membuat Section Tabel Pada Form RLB. 176](#_Toc35528682)

[Listings 6. 75 Syntax Untuk Membuat Tabel Serta Menampilkan Data. 177](#_Toc35528683)

[Listings 6. 76 Syntax Untuk Membuat Tabel Pada Tab Kedua. 179](#_Toc35528684)

[Listings 6. 77 Syntax Untuk Menampilkan Perhitungan Determinan. 181](#_Toc35528685)

[Listings 6. 78 Syntax Untuk Menghitung Menggunakan Rumus RLB. 182](#_Toc35528686)

[Listings 6. 79 Syntax Untuk Menampilkan Chart. 189](#_Toc35528687)

[Listings 6. 80 Syntax Untuk Menampilkan Data Secara Keseluruhan. 192](#_Toc35528688)

[Listings 6. 81 Syntax Untuk Menampilkan Informasi Jumlah Pelanggan. 194](#_Toc35528689)

[Listings 6. 82 Syntax Untuk Menampilkan Informasi Jumlah Produk. 197](#_Toc35528690)

# 

# FOREWORD

Terimakasih penulis ucapkan kepada kaprodi D-IV Teknik Informatika, bagian kemahasiswaan, serta khususnya mahasiswa D-IV Teknik Informatika yang telah membantu proses penulisan dalam buku ini.

# KATA PENGANTAR

Penulisan buku ini dimaksudkan untuk memberikan arahan atau sarana pembelajaran kepada pembaca mengenai perancangan sistem peramalan serta penerapan metode regresi linier berganda dalam kehidupan sehari-hari yang kemudian proses perancangan sistem peramalan tersebut dibuat dengan menggunakan tools bahasa pemograman php.

Cahyo Prianto, Imron Sumadireja

Bandung, Jawa Barat

Januari, 2020

# ACKNOWLEDGMENTS

Puji syukur penulis haturkan kepada Allah SWT atas berkah, rahmat dan hidayahnya sehingga penulis dapat menyelesaikan buku yang berjudul “Perancangan Sistem Peramalan Menggunakan Regresi Linier Berganda”. Dalam menulis buku ini banyak orang yang telah memotivasi, dan menyampaikan saran bagi penulis. Dalam kesempatan yang berharga ini, penulis bermaksud menyampaikan rasa terima kasih dan penghargaannya kepada mereka semua. Pertama, penghargaan terdalam dari penulis diberikan kepada orang tua tercinta untuk do’a dan dukungannya. Penulis juga ucapkan terima kasih kepada pembimbing yang mana tanpa bantuan, dukungan dan saran pembimbing penulis buku ini mungkin tidak akan terselesaikan. Buku ini jauh dari sempurna, tetapi diharapkan akan bermanfaat tidak hanya bagi penulis, tetapi juga bagi pembaca. Untuk alasan ini, saran dan kritik bijaksana yang konstruktif disambut.

I.S.

# ACRONYMS

RLB Regresi Linier Berganda

FI Form Interpreter

HTML Hypertext Markup Language

# INTRODUCTION

CAHYO PRIANTO, S.Pd., M.T, IMRON SUMADIREJA.

Informatics Research Center

Bandung, Jawa Barat, Indonesia

Sistem Peramalan merupakan sekumpulan komponen untuk mencapai tujuan pada proses penentuan peramalan jumlah pendapatan pada produk tertentu, hingga prospek yang didapatkan dari penjualan suatu produk untuk dijadikan analisis strategi pemasaran berikutnya. Untuk memberikan kemudahan dalam menentukan hal-hal tersebut, sistem peramalan ini dijalankan dengan menggunakan metode regresi linier berganda.

# LATAR BELAKANG MASALAH

Kemajuan zaman dan teknologi saat ini telah merubah pola dalam perusahaan atau pun organisasi-organisasi. Data-data yang besar dan telah terkumpul dalam waktu yang lama dapat diolah menjadi sumber informasi yang dapat membantu menganalisis eksistensi dari sebuah perusahaan atau organisasi. Analisa otomatis dari data yang berjumlah besar atau kompleks yang terbentuk sebagai *data mining*, dengan tujuan untuk menemukan pola atau kecenderungan yang penting biasanya tidak disadari keberadaannya saat ini telah banyak dilakukan[1].

Selain itu permasalahan optimasi merupakan permasalahan yang sering ditemui dalam kehidupan sehari-hari. Hal ini tidak terlepas dari sifat manusia yang ingin mendapat keuntungan maksimum dan menderita kerugian yang minimum[2].

Optimasi adalah sarana untuk mengekspresikan model yang bertujuan untuk memecahkan masalah dengan cara terbaik. Model optimasi yang ada digunakan untuk menyelesaikan berbagai permasalahan dalam pemerintah, bisnis, teknik ekonomi, ilmu-ilmu fisika dan sosial yang terkait dengan adanya keterbatasan pengalokasian sumber daya[3]. Persoalan optimasi merupakan persoalan yang sangat penting untuk diterapkan dalam segala sistem maupun organisasi.

Gambar 1. 1 Optimasi

*(Sumber: Logique Digital Indonesia)*

testing

# LANDASAN TEORI

## *HYPERTEXT PREPROCESSOR (PHP)*

PHP pertama kali dibuat oleh Rasmus Lerdorf, seorang programmer C. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data *form* dari *web*. Jadi, awal mula PHP digunakannya untuk menghitung jumlah pengunjung didalam *web*-nya.

Dengan alasan untuk meningkatkan performa, Rasmus Lerdorf membuat ulang kode program tersebut dalam bahasa C. Lerdord menyebut kode program ini sebagai *Personal Home Page*. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan skrip PERL yang dibuatnya untuk membuat halaman *web*-nya menjadi dinamis. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kedependekan dari *Hypertext Prerocessing / Form Interpreter*.

Dengan perilisan kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah dapat mengakses *database* dan dapat terintegrasi dengan *Hypertext Markup Language* (HTML). Pada rilis ini *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP / FI secara signifikan. PHP versi 2.0 ini telah menarik banyak perahtian *programmer*, namun bahasa ini memiliki masalah dengan kestabilan yang kurang bisa diandalkan. Hal ini lebih dikarenakan Lerdorf hanya bekerja sendiri untuk mengembangkan PHP[6].

Pada saat itulah Zeev Suraski dan Andi Gutmans, ikut mengambil bagian dan membuat ulang *parsing engine* yang menjadi dasar dari PHP agar lebih stabil. Dengan dukungan dari banyak *programmer* lainnya, proyek PHP secara perlahan beralih dari proyek satu orang menjadi proyek masal yang lebih akrab kita kenal sebagai *open-source project.* PHP selanjutnya dikembangkan oleh The PHP Group yang merupakan kumpulan banyak *programmer* dari seluruh dunia.

*Pada* tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang *software*-nya.

PHP versi 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya *Zend Engine* yang dibuat oleh Zeev Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP *scripting engine*, yang lainnya adalah *build in HTTP session*, tidak lagi menggunakan *library* tambahan seperti pada PHP. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi *web*. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan *web server*.

PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, *Zend* merilis PHP 5.0. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Beberapa penambahan fitur meliputi *PHP Data Objects* (PDO) untuk mengakses *database*, *closures, trait,* dan *namespaces*.

Versi lanjutan dari PHP, yakni PHP 6.x sebenarnya telah lama dikembangkan, bahkan sejak tahun 2005. Fokus pengembangan PHP 6 terutama dalam mendukung *unicode*, agar PHP bisa mendukung berbagai jenis karakter bahasa *non-latin*.

Namun dikarenakan beberapa alasan seperti kurangnya *programmer*, dan performa yang tidak memuaskan, pengembangan PHP 6 dihentikan dan fitur yang ada dimasukkan kedalama PHP 5.

Pada tahun 2014, sebuah proyek lanjutan PHP mulai mengemuka, yakni PHP 7 yang berkembang dari banyak eksperimen yang dinamakan PHP *Next Generation* (PHPNG), yang dikembangkan Dmitry Stogov, Xinchen Hui, dan Nikita Popov. Proyek ini menggunakan pendekatan modern agar PHP diproses lebih cepat seperti memakai teknik *just-in-time* (JIT) *compiler*. Proyek PHPNG bertujuan untuk menyusun ulang kode PHP untuk meningkatkan performa. Selain performa yang meningkat, terdapat beberapa fitur baru pada PHP 7, seperti *combined comparison operator* atau dikenal dengan *spaceship operator* “⬄”, *anonymous classes*, dan dukungan yang lebih stabil untuk server 64-bit.

Beberapa fitur yang sudah usang (*deprecated*) juga dihapus, seperti penulisan PHP dengan ASP *style* <% %> dan tag <script language=php></script>. Kedua cara ini sudah tidak bisa digunakan lagi. Modul *mysql extension* juga dihapus karena sudah digantin dengan *mysqli extension*. Penghapusan modul *mysql extension* ini sering menjadi masalah karena banyak kode program atau buku PHP lama yang masih menggunakannya.



Gambar 2. 1 Rasmus Lerdorf.

*(Sumber: abuadzhan(dot)com)*

PHP adalah basaha pemrogmana *script* sisi server yang didesain untuk pengembangan *web*. Dimana PHP ini merupakan singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa pemrograman umum. Selain itu PHP juga digunakan bersamaan dengan bahasa pemrograman lainnya seperti bahasa pemrograman HTML, dan Javascript.

PHP disebut sebagai bahasa pemrograman *server-side* karena, hal ini berbeda dengan bahasa pemrograman *client-side* seperti Javascript yang diproses pada *web browser* (*client*). PHP juga menjadi dasar dari aplikasi *Content Management System* (CMS) yang populer seperti Joomla, Drupal, dan Wordpress.

Salah satu fungsi dari PHP ini dapat disisipkan pada dokumen HTML. Karena kemampuan inilah PHP juga sering disebut sebagai bahasa pemrograman *script* atau *scripting language*. Berikut ini merupakan sintaksis dasar pada PHP[7].

1. Pembatas

PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun di luar pembatas tidak diproses oleh PHP. Pembatas paling umum adalah “<?php” untuk membuka dan “?>” untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, dan Javascript.

1. Variabel

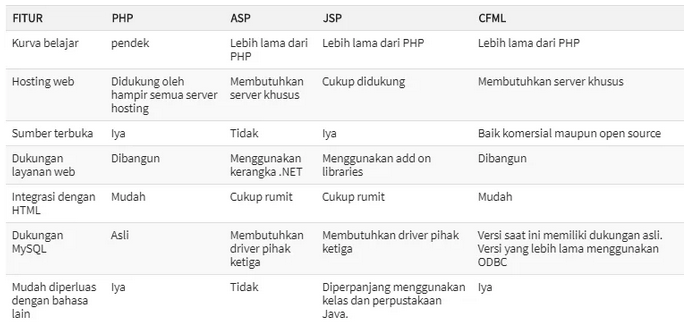
Variabel dalam PHP diawali dengan simbol dolar “$”. Pada versi PHP 5 diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa mereka menjadi parameter objek dari *class* tertentu, *array*, atau fungsi. Namun, jenis petenjuk tidak dapat digunakan dengan jenis skalar seperti angka atau *string*. Contoh variabel dapat ditulis sebagai *$nama\_variabel*.

1. Komentar

PHP memiliki 3 jenis sintaks sebagai komentar pada kode yaitu blok “/ \* \* /”, komentar 2 baris “//”, serta tanda “#” digunakan untuk komentar 1 baris. Komentar bertujuan untuk meninggalkan catatan pada kode PHP dan tidak akan diterjemahkan ke program.

1. Keunggulan bahasa pemrograman PHP, antara lain :
2. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
3. *Web server* yang mendukung PHP dapat ditemukan di mana-mana dari mulai *apache, IIS, Lightpad,* hingga *xitami* dengan konfirgurasi yang relatif mudah.
4. Dalam sisi pengembangan lebih mudah, karena banyaknya *developer* yang siap membantu dalam pengembangan.
5. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
6. PHP adalah bahasa *open source* yang digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintah-perintah sistem[8].
7. Kekurangan bahasa pemrograman PHP, antara lain :
8. Tidak ideal jika untuk pengembangan skala besar.
9. Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya.
10. Tidak dapat memisahkan antara tampilan dengan logik dengan baik.
11. PHP mempunyai kelemahan *security* tertentu yang mana jika *programmer* tidak jeli dalam melakukan pemrograman dan kurang memperhatikan isu dan konfigurasi PHP[8].

Berikut adalah perbandingan antara PHP dengan bahasa *scripting* *server-side* lainnya seperti ASP – *Active Server Pages*, JSP – *Java Server Pages*, CFML – *Cold Markup Language*.



Gambar 2. 2 Perbandingan PHP dengan Bahsa Scripting Server-Side.

*(Sumber: jagoanhosting(dot)com)*

## *OBJECT ORIENTED PROGRAMMING*

Konsep *Object Oriented Programming* (OOP) pertama kali muncul di MIT (*Massachusetts Institute of Technology*) pada era 1960-an. Sekitar beberapa tahun kemudian antara 1962-1965, sebuah basaha pemrograman yang mendasari konsep OOP diperkenalkan dengan nama bahasa pemrograman SIMULA 1, dikembangkan oleh Kristen Nygaard dan Ole-Johan yang merupakan warga negara Norwegia. Setelah itu pada tahun 1967 keluarlah SIMULA 67.

Bahasa SIMULA dengan OOP menginspirasi banyak pengembang, sehingga pada tahun 70-an tercipta bahasa SMALTALK, bahasa yang pertama kali disebut oleh dunia sebagai *object oriented programming*, dengan membawa fitur barunya yaitu *inheritance*.

Semenjak itu konsep OOP semakin dikenal, pada tahun 1980-an banyak bahasa pemrograman menggunakan konsep OOP, diperkenalkan kepada dunia, tetapi hanya empat yang cukup populer, diantaranya ADA (US Department of Defense), PROLOG (The Japanese “Fifth Generation Computer Project), Eifle dan C++. Bahasa pemrograman ADA dan PROLOG dipercaya akan bersaing ketat sebagai bahasa pemrograman yang paling dominan.

Namun pada tahun 1980-an bahasa pemrograman C++ mematahkan kepercayaan tersebut. Bahasa pemrograman C++ menjadi bahasa pemrograman yang populer dan mendominasi hingga sekarang. Bahasa pemrograman C++ yang merupakan gabungan dari 2 konsep bahasa pemrograman, yakni C dan SIMULA[9].

Semenjak C++ terkenal, banyak sekali pengembang yang terinspirasi oleh C++ dan pada tahun 1990-an, bahasa pemrograman Java diperkenalkan yang mengaku terinspirasi oleh C++, dan tahun 2002 perusahaan Microsoft juga mengeluarkan bahasa turunan dari C++ yaitu C# (C-Sharp), disusul dengan VB.Net dengan fitur OOP yang merupakan penyempurnaan dari bahasa VB 0.6 yang tidak mendukung fitur OOP.

*Object Oriented Programming* atau disingkat menjadi OOP adalah paradigma pemrograman dalam melakukan pemrograman yang berorientasi kepada objek, semua fungsi, data dan pengolahan data akan dibungkus dalam kelas-kelas dan objek-objek.

Masing-masing objek dapat memiliki sifat dan tugasnya. Pada paradigma ini, objek-objek tersebut dapat bekerja sendiri dan juga dapat saling bekerja sama dengan kemungkinan untuk saling berhubungan, seperti menerima, mengirim data kepada objek lainnya dan memproses data.

Paradigma OOP dapat dilihat sebagai interaksi dari objek yang saling berhubungan satu sama lain untuk melakukan tugasnya. Pengembangan program OOP dilakukan dengan cara mengikuti model yang telah ada di dalam dunia nyata.

OOP bertujuan untuk memberikan pola pikir dalam mengembangkan program, pola pikir tersebut dipercaya dapat memberikan kemudahan, fleksibilitas, kemudahan pembuatan, pengembangan program, dan perawatan program[10].

Karena kemudahan yang diberikan oleh konsep OOP, banyak bahasa yang membawa dukungan fitur OOP, karena hal itu OOP menjadi memiliki dan dibagi menjadi 3 jenis, yaitu :

1. Bahasa OOP murni, merupakan sebuah bahasa yang mengharuskan program ditulis hanya berupa objek saja. Contoh – Eifel, Smaltalk, Ruby, Jade, dan lain-lain.
2. Bahasa OOP *hybrid*, merupakan bahasa yang dirancang untuk pemrograman objek dengan beberapa elemen prosedural.
3. Bahasa OOP *hybrid* dalam *web*, salah seperti bahasa OOP *hybrid*, yang membedakan hanya konsep yang digunakan dalam pemrograman *web*.

Konsep *Object Oriented Programming*

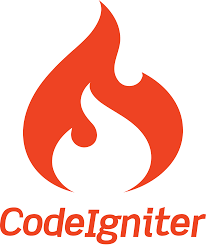
1. *Class,* adalah sebuah rancangan untuk mendefinisikan karakter dan perilaku dari objek, yang merupakan kumpulan atas definisi dan fungsi-fungsi dalam suatu unit, untuk suatu tujuan tertentu.
2. *Object*, adalah dasar dari modularitas dan struktur pada OOP, dan merupakan representasi dari *class*, objek akan memiliki sifat dan perilaku dari *class* yang digunakan.
3. *Encapsulation*, adalah konsep dalam implementasi untuk membungkus data dan fungsi menjadi satu entitas, dan membatasi akses dari luar *class*.
4. *Inheritance*, adalah konsep pewarisan *class*. *Class* juga dapat menuruni dan memiliki apa yang dimiliki oleh *class* lainnya.
5. *Abstraction,* adalah konsep untuk mendesain sebuah objek, teknik dalam menyembunyikan detail suatu proses dalam objek tersebut, dengan tujuan untuk memfokuskan pengguna pada fungsi inti objek.
6. *Polymorphism,* adalah kemampuan dalam menyampaikan pesan tertentu keluar dari hierarki objeknya, dimana objek yang berbeda memberikan tanggapan atau respon terhadap pesan yang sama sesuai dengan sifat masing-masing objek.
7. Keunggulan OOP, diantaranya :
8. OOP menyediakan struktur modular yang jelas untuk program sehingga OOP sangat bagus digunakan untuk mendefinisikan tipe data abstrak dimana detail implementasinya tersembunyi.
9. OOP akan mempermudah dalam memaintain dan memodifikasi kode yang sudah ada. Objek yang baru dapat dibuat tanpa mengubah kode yang sudah ada.
10. OOP menyediakan *framework* untuk *library* kode dimana komponen *software* yang tersedia dapat dengan mudah diadaptasi dan dimodifikasi oleh programmer. Hal ini sangat berguna untuk mengembangkan *Graphical User Interfaces* (GUI)[10].
11. Kekurangan OOP, diantaranya :
12. Tidak memperbolehkan implementasi yang kuat pada *reuse*.
13. *Property software* tidak terikat dalam satu unit fungsional sehingga harus *crosscut* di antara komponennya.
14. *Crosscut* tersebut mengakibatkan sulitnya pengembangan dan pemeliharaan[10].

## *CODEIGNITER*

Kelahiran codeignier adalah bermula dari kegalauan Rick Ellis atas banyaknya kode PHP yang harus ditulis ketika membangun salah satu CMS kesayangannya, *expression engine*. Rick Ellis ingin mempermudah penulisan kode-kode program PHP dan membuatnya lebih singkat dan cepat. Akhirnya Rick Ellis mengambil inisiatif dengan membuat sendiri kode singkat / *shortcode* dari fungsi-fungsi yang ada di PHP.

Pembuatan *shortcode* tersebut membuatnya mampu membangun *expression engine* dengan sangat bagus, efisien dan cepat. Selain itu, performanya juga sangat bagus.

Setelah tidak berapa lama, Rick Ellis melalui situsnya ellislab(dot)com membagikan *shortcode* yang dibuat sendiri itu untuk digunakan oleh *developer* lainnya. Tujuannya yaitu membantu *developer* lain dalam menangani masalah dalam *framework* yang terkenal. Banyak *developer* turun tangan membantu perkembangan CodeIgniter dan puncaknya, CI menjadi *framework* terpopuler tahun 2006.



Gambar 2. 3 Logo CodeIgniter.

*(Sumber: iconfinder(dot)com)*

Codeigniter (CI) merupakan sebuah *web application framework* yang bersifat *open source* dimana digunakan untuk membangun aplikasi php dinamis. *Framework* itu sendiri merupakan abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat *user*, sehingga dapat menyediakan perangkat lunak untuk aplikasi tertentu[11].

Codeigniter menjadi sebuah *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP yang dapat mempercepat pengembangan untuk membuat sebuah aplikasi *web*. Selain ringan dan cepat, codeigniter juga memiliki dokumentasi yang lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih codeigniter sebagai *framework* pilihannya. Berikut merupakan gambar perbandingan PHP biasa dengan Codeigniter.



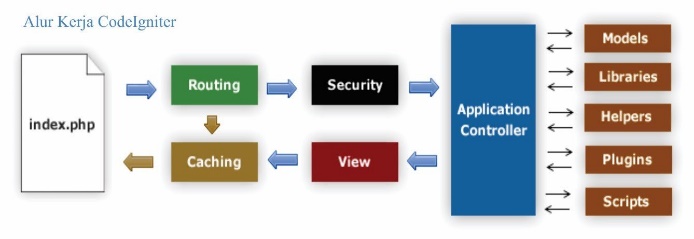
Gambar 2. 4 Konsep MVC.

*(Sumber: idcloudhost(dot)com)*

MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi, diantaranya :

1. Model, dapat berhubungan dengan *database* seperti *insert, update, delete*. Menangani validasi dari bagian controller, namun model tidak berhubungan langsung dengan *view*.
2. Controller, merupakan bagian yang mengatur hubungan antara model dan *view*, controller berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.
3. *View,* merupakan bagian yang menangani *presentation*. Pada suatu aplikasi *web* bagian ini biasanya berupa file template HTML, yang diatur oleh controller. *View* berfungsi untuk menerima dan mempresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model[12].

Alur kerja *framework* Codeigniter dapat digambarkan sebagai berikut :



Gambar 2. 5 Alur Kerja Framework CodeIgniter.

*(Sumber: idcloudhost(dot)com)*

1. Index.php: berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
2. Router: router akan memeriksa HTTP *request* untuk menentukan hal apa yang harus dilakukan oleh program.
3. *Cache* File: apabila dalam program sudah terdapat “*cache* file” maka file tersebut akan langsung dikirim ke browser. File *cache* inilah yang dapat membuat sebuah *website* dapat dibuka dengan lebih cepat. *Cache* file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.
4. *Security*: sebelum *file* *controller* di *load* keseluruhan, HTTP *request* dan data yang disubmit oleh *user* akan disaring terlebih dahulu melalui fasilitas *securtiy* yang dimiliki oleh codeigniter.
5. Controller: controller akan membuka file model, *core libraries*, *helper* dan semua *resources* yang dibutuhkan dalam program tersebut.
6. *View*: hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam *view* file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file *view* sudah ada yang di “cache” maka file *view* baru yang belum ter-*cache* akan *update* file *view* yang sudah ada.

Fungsi CodeIgniter

1. Mempercepat dan mempermudah kita dalam pembuaan *website*.
2. Menghasilkan struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya.
3. Memberikan standar *coding* sehingga memudahkan kita atau orang lain untuk mempelajari kembali sistem aplikasi yang dibangun[12].
4. Keunggulan Codeigniter, diantaranya :
5. Performa sangat cepat, salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scratch*, tetapi codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan *framework* yang paling cepat dibanding *framework* yang lainnya.
6. Konfigurasi yang sangat minim (*nearly zero configuration*), tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetapi diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan codeigniter dengan setingan standar, anda hanya perlu mengubah sedikit saja file pada folder config.
7. Dokumentasi yang lengkap, setiap paket instalasi codeigniter sudah disertai *user guide* yang bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.
8. Kekurangan Codeigniter, diantaranya :
9. Codeigniter dikembangkan oleh Ellis lab dan bukan oleh suatu komunitas, yang menyebabkan *update code engine*-nya tidak secepat *framework* lain.
10. Tidak ditujukan untuk pembuatan *web* dengan skala besar (*enterprise*) walaupun tersedia banyak *library*.
11. Masih banyak kelonggaran dalam hal *coding*, misalnya bebas dalam menambah file.

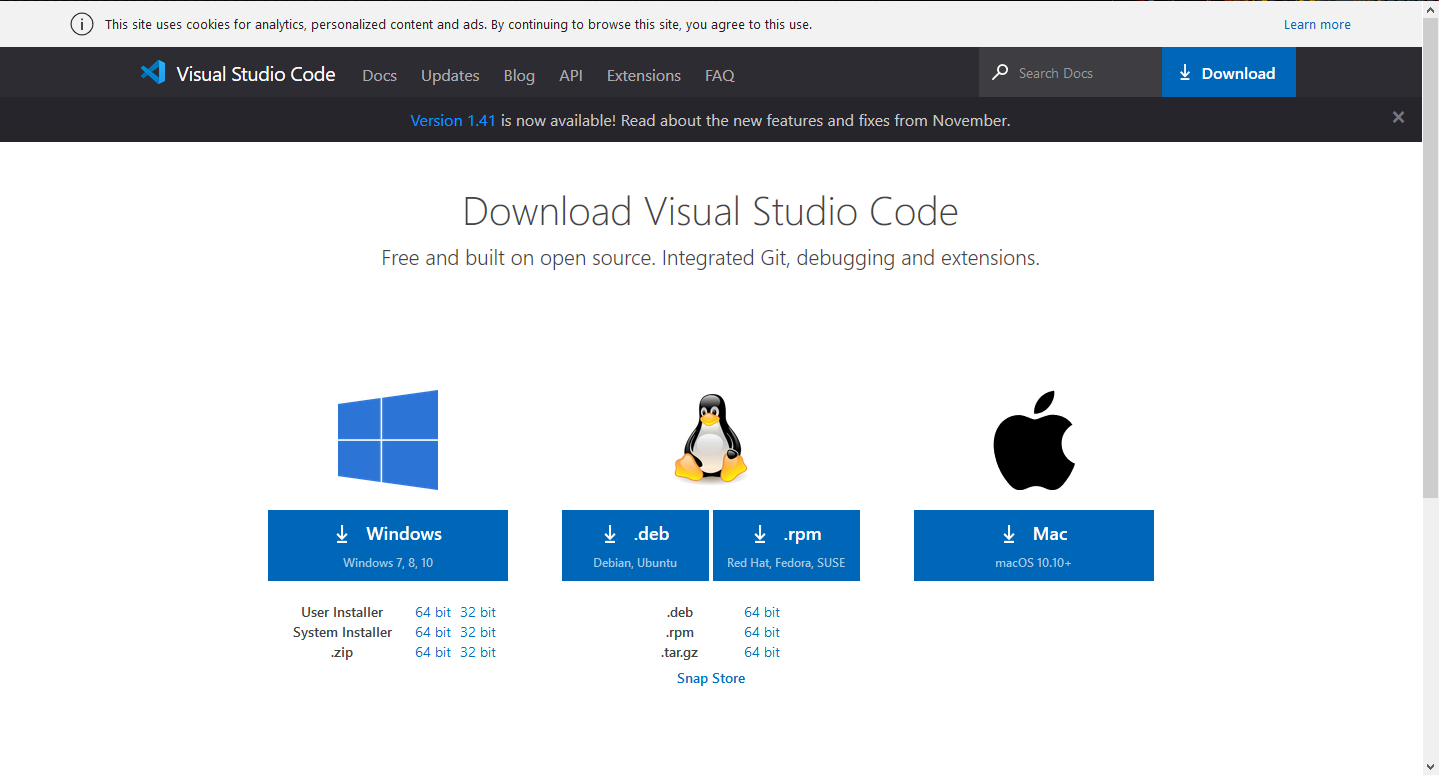
# INSTALASI APLIKASI YANG AKAN DIGUNAKAN

## *Visual Studio Code*

Visual studio code (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform,* artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via *marketplace* Visual Studio Code (seperti C++, C#, PHP, Python, Go, Java, dst).

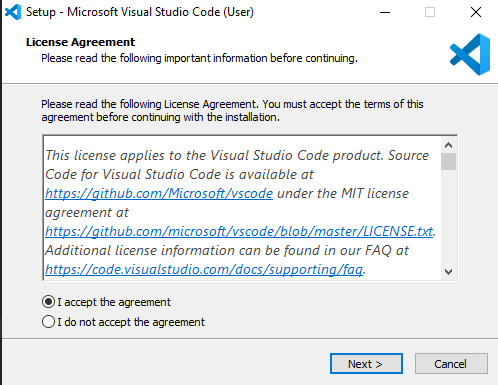
Berikut ini merupakan langkah-langkah instalasi visual studio code pada sistem operasi windows.

1. Pertama, *download* terlebih dahulu file installer visual studio code melalui situs resminya pada link berikut ini *https(:)//code(dot)visualstudio(dot)com/download*



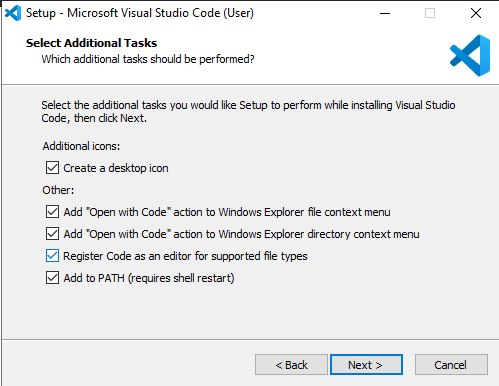
Gambar 3. 1 Download Visual Studio Code.

1. Terdapat 2 piliahn untuk sistem operasi windows, diantaranya ada yang 64 bit dan 32 bit. Pilihlah installer sesuai dengan spesifikasi laptop yang digunakan.
2. Jika, sudah berhasil di *download*, lanjut pada proses instalasi, pertama klik pada file installer vs code.
3. Pilih “I accept the aggrement” untuk menyetujui “License agreement”, kemudian klik next.



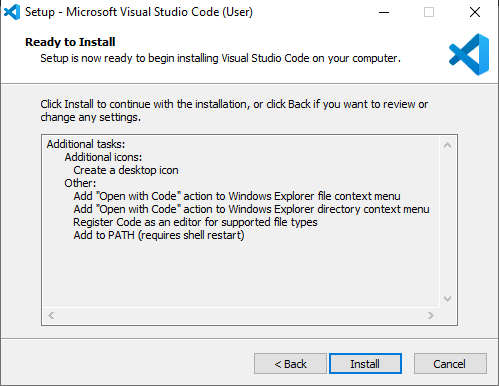
Gambar 3. 2 Instalasi Visual Studio Code.

1. Pada bagian select *additional tasks*, centang semuanya kemudian klik next.



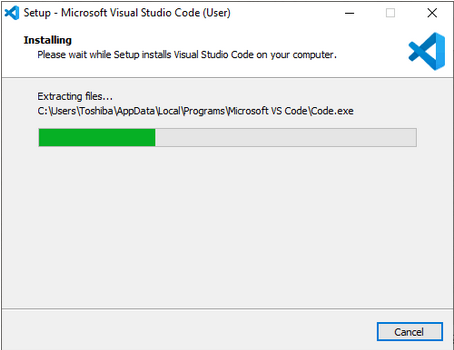
Gambar 3. 3 Select Additional Tasks.

1. Kemudian klik install untuk melanjutkan proses instalasi.



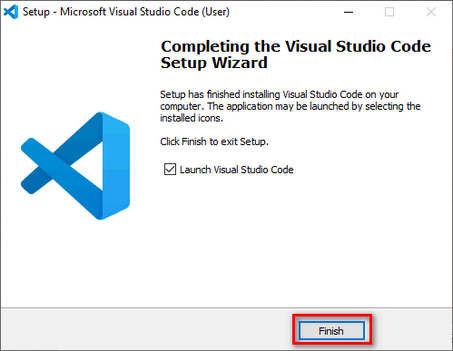
Gambar 3. 4 Ready To Install.

1. Tunggu proses instalasi hingga selesai.



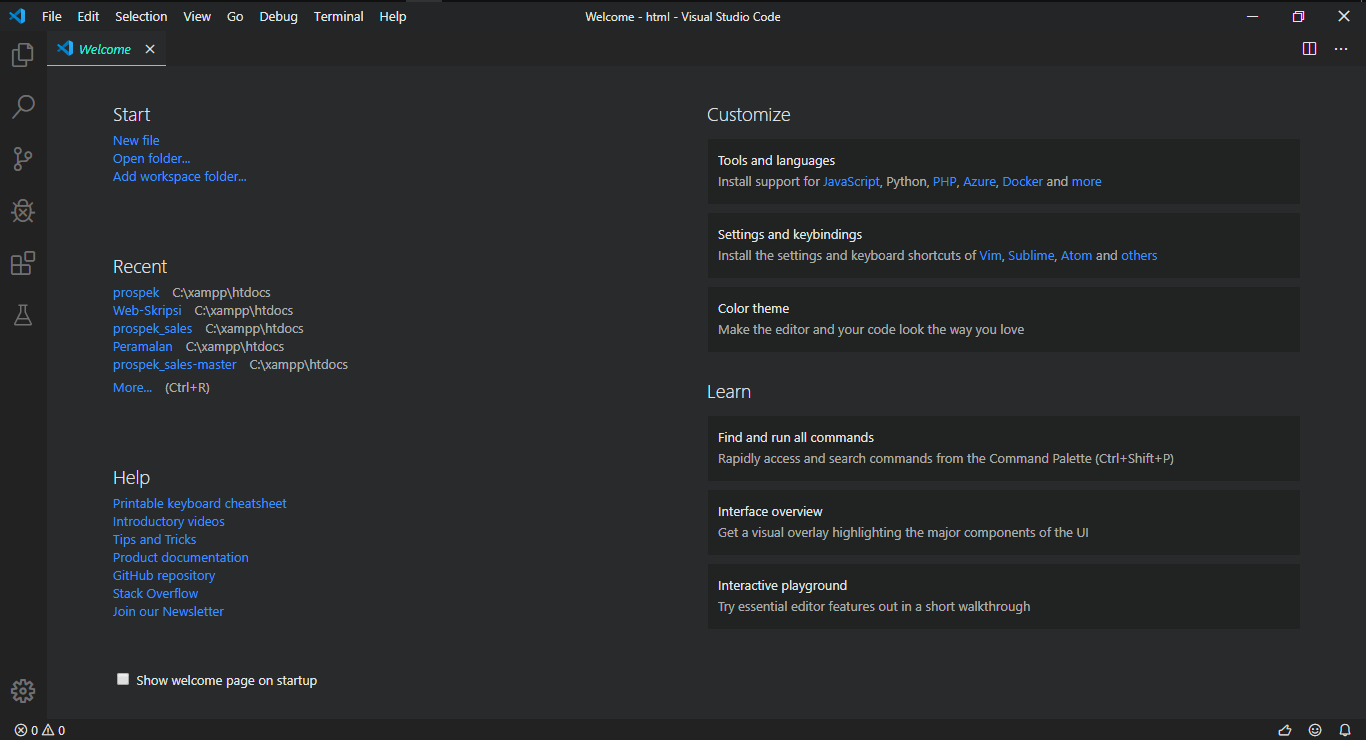
Gambar 3. 5 Proses Installasi.

1. Setelah selesai klik Finish.



Gambar 3. 6 Installasi Berhasil.

1. Jika berhasil, maka tampilan utamanya akan seperti berikut.



Gambar 3. 7 Menu Utama Visual Studio Code.

## *Xampp*

Xampp adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL / MariaDB, PHP, dan PERL. Sementara huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi yang berbeda.

Jika dijabarkan secara gamblang, masing-masing huruf yang adad di dalam nama XAMPP memiliki arti sebagai berikut.

**X = *Cross Platform*,** merupakan kode penanda untuk *software* *cross platform* atau yang bisa berjalan di banyak sistem operasi.

**A = Apache**, apache adalah aplikasi *web server* yang bersifat gratis dan bisa dikembangkan oleh banyak orang (*open source*).

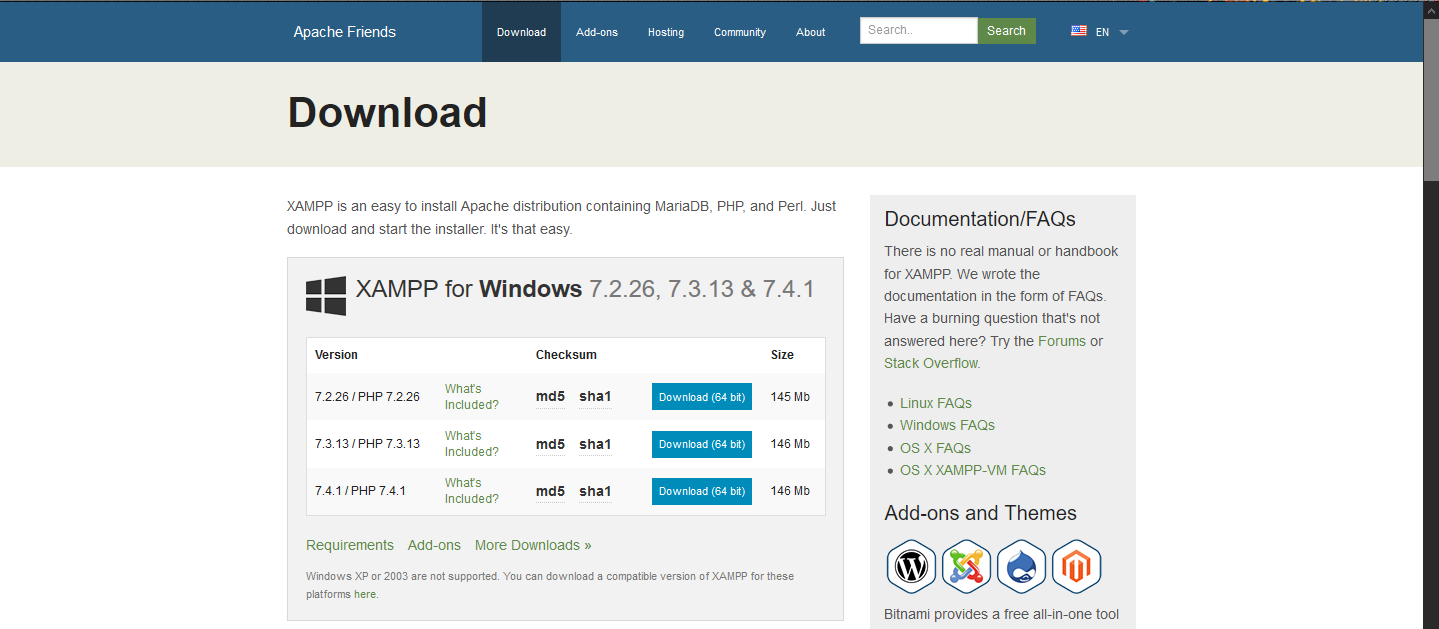
**M = MySQL / MariaDB**, MySQL atau MariaDB merupakan aplikasi *database server* yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui *database*.

**P = PHP**, huruf “P” yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat *website dinamis*.

**P = Perl**, untuk huruf “P” selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi, sehingga sangat fleksibel dan banyak digunakan.

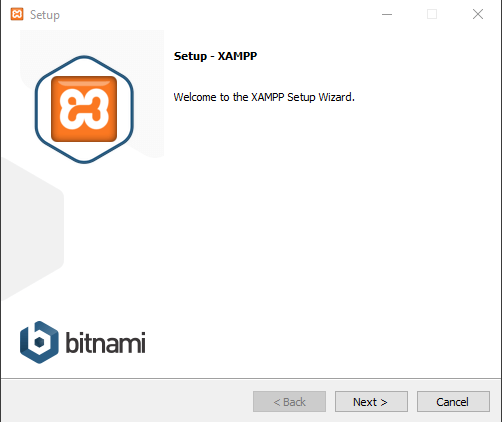
Berikut merupakan langkah-langkah untuk melakukan instalasi xampp pada sistem operasi windows.

1. Pertama, *download* terlebih dahulu file installer xampp pada link berikut ini. *https(:)//apachefriends(dot)org*



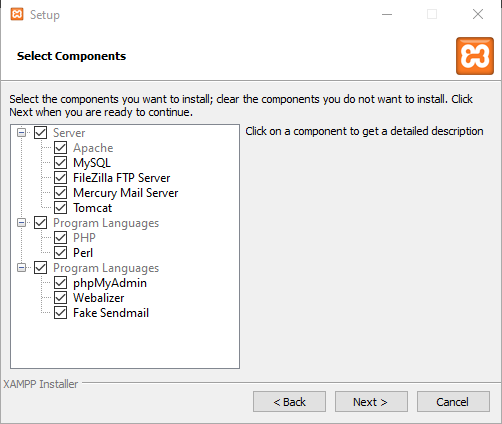
Gambar 3. 8 Download File XAMPP.

1. Pilih file installer sesuai dengan spesifikasi laptop yang digunakan.
2. Klik file xampp yang sudah berhasil di *download*, maka akan muncul tampilan seperti ini.



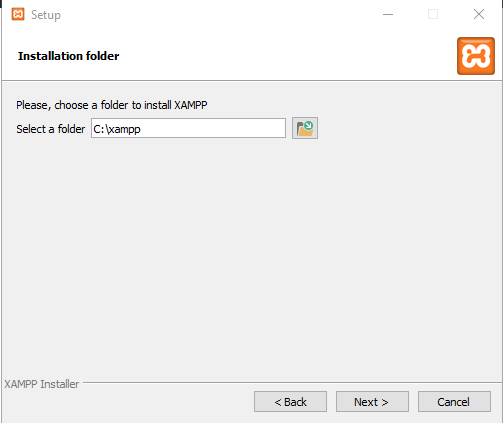
Gambar 3. 9 Setup Installasi XAMPP.

1. Kemudian, pilih next untuk melanjutkan pada proses instalasi.



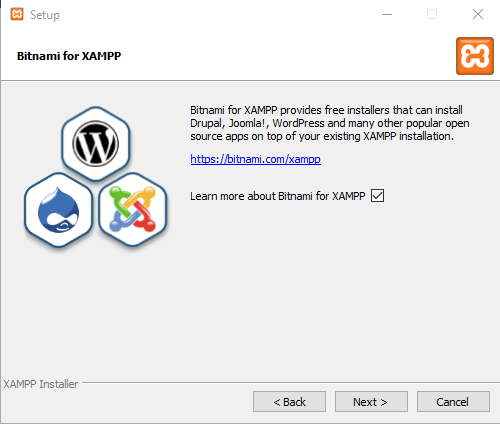
Gambar 3. 10 Select Component.

1. Pada tampilan *select component*, centang semua pilihannya agar dapat menjalankan semua program yang dipilih. Kemudian pilih *next*.



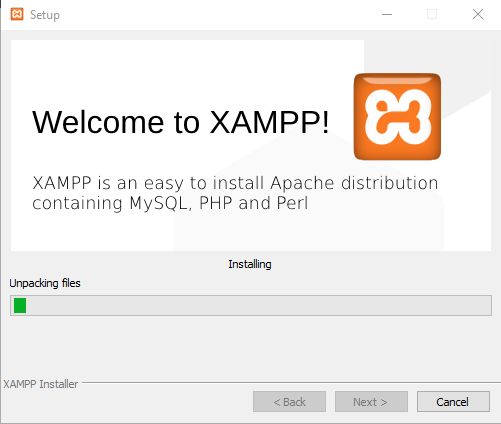
Gambar 3. 11 Installation Folder.

1. Pada, *installation folder* secara *default* sistem akan menyimpannya pada direktori C. Jika sudah klik *next* untuk melanjutkan proses instalasi.



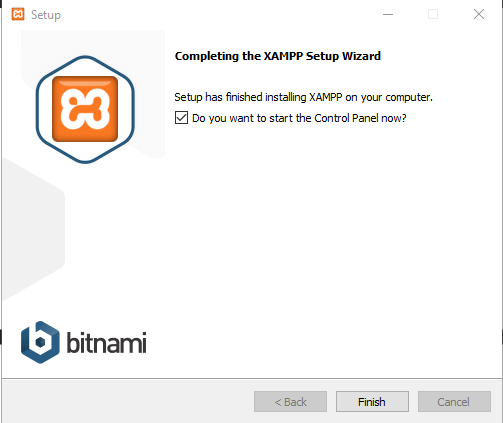
Gambar 3. 12 Ready To Install.

1. Pada tampilan ini langsung saja klik *next* untuk melanjutkan.



Gambar 3. 13 Proses Installasi.

1. Kemudian tunggu hingga proses instalasi selesai.



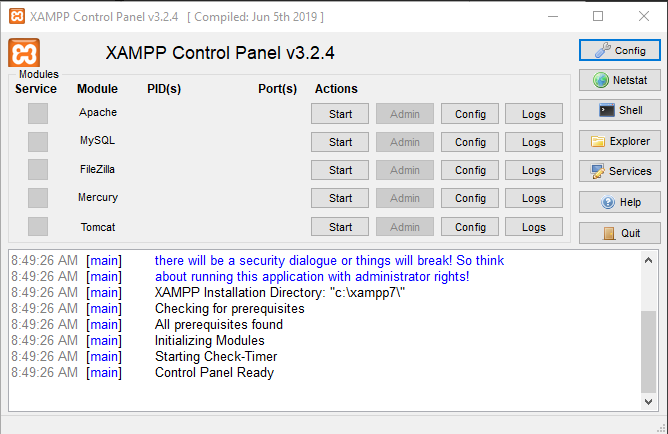
Gambar 3. 14 Installasi Berhasil.

1. Jika telah selesai, selanjutnya klik *Finish* untuk membuka *control panel*.



Gambar 3. 15 Pilih Bahasa Yang Akan Digunakan.

1. Pilih bahasa yang akan digunakan pada xampp. Lalu klik *save*.



Gambar 3. 16 Menu Utama XAMPP.

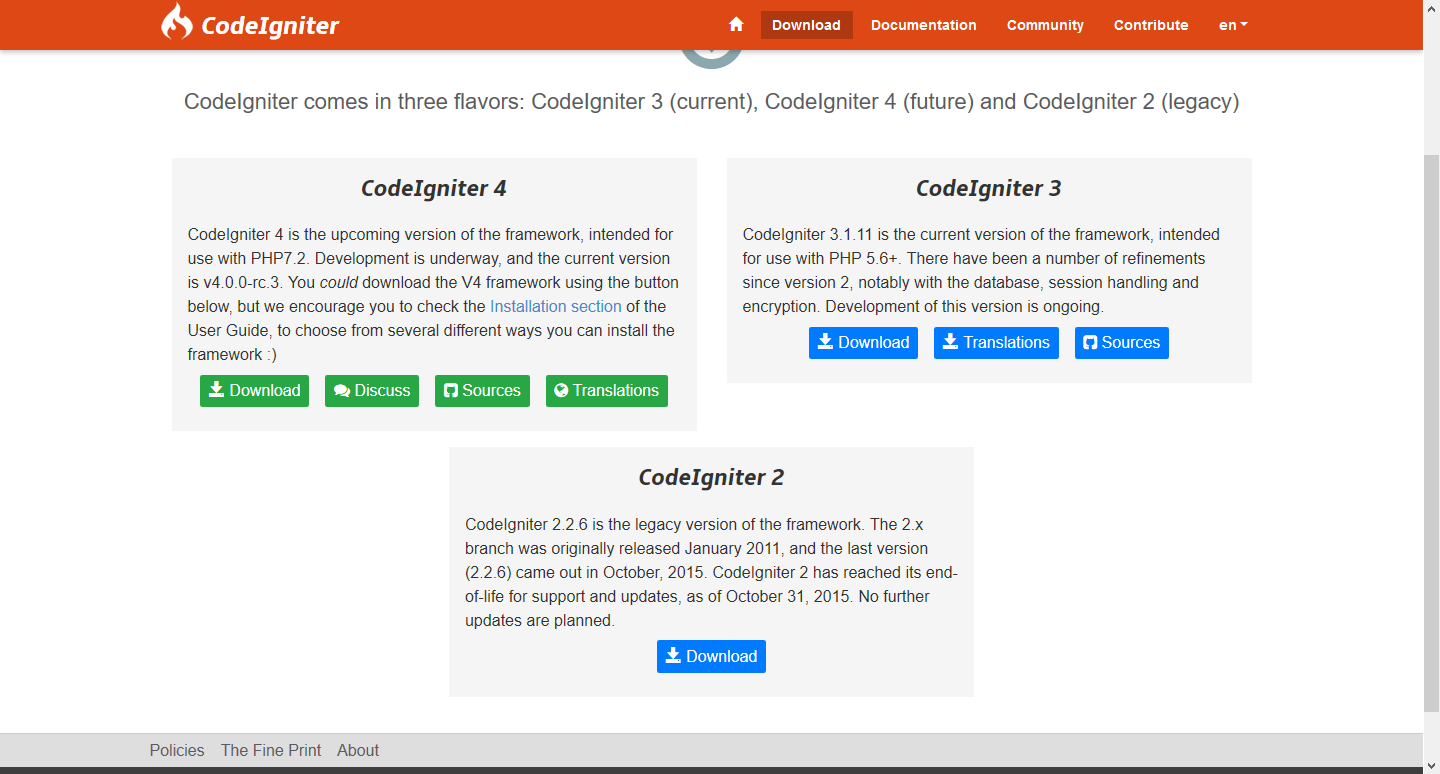
1. Jika berhasil, maka tampilan utama *control panel* dari xampp seperti berikut.

## *Codeigniter*

Codeigniter (CI) merupakan sebuah *web application framework* yang bersifat *open source* dimana digunakan untuk membangun aplikasi php dinamis. *Framework* itu sendiri merupakan abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat *user*, sehingga dapat menyediakan perangkat lunak untuk aplikasi tertentu.

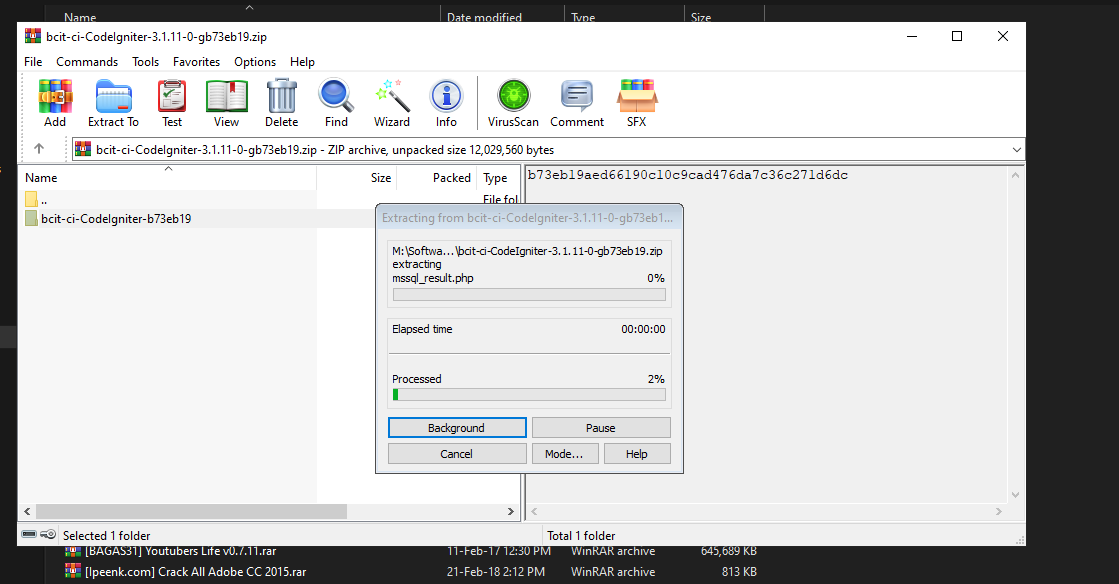
Berikut ini merupakan tahapan-tahapan untuk instalasi codeigniter pada sistem operasi windows.

1. Pertama, *download* terlebih dahulu codeigniter pada *website* resminya pada link berikut. *https(:)//codeigniter(dot)com/download*.



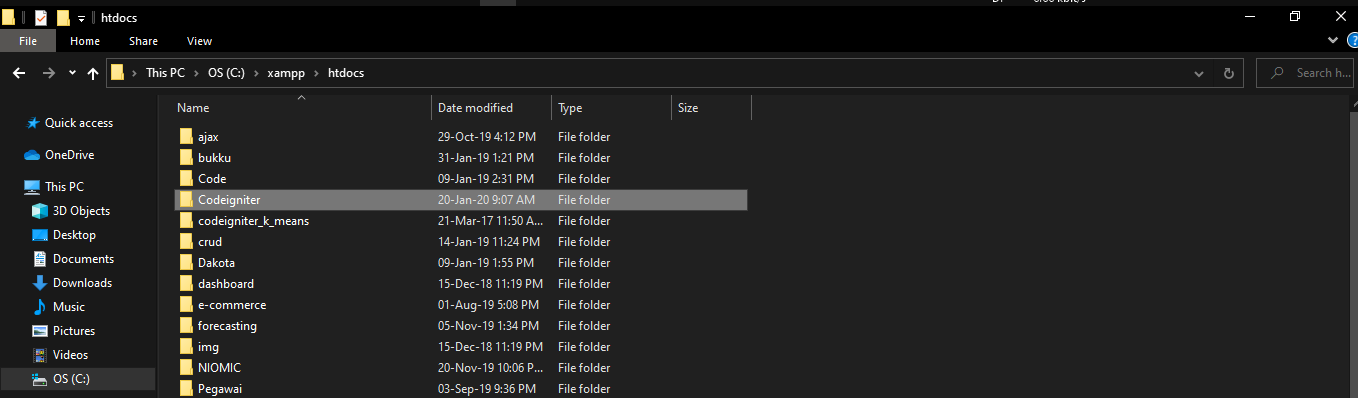
Gambar 3. 17 Download File CodeIgniter.

1. Setelah berhasil di *download*, selanjutnya ekstrak file tersebut.



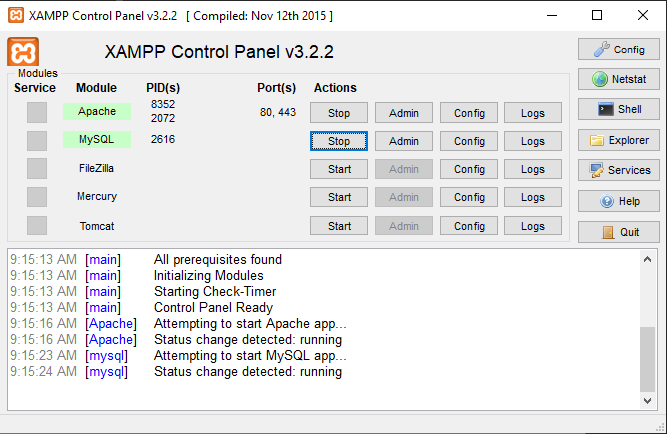
Gambar 3. 18 Ekstrak File CodeIgniter.

1. Kemudian, *copy* file yang telah di ekstrak ke dalam direktori htdocs, seperti gambar berikut.



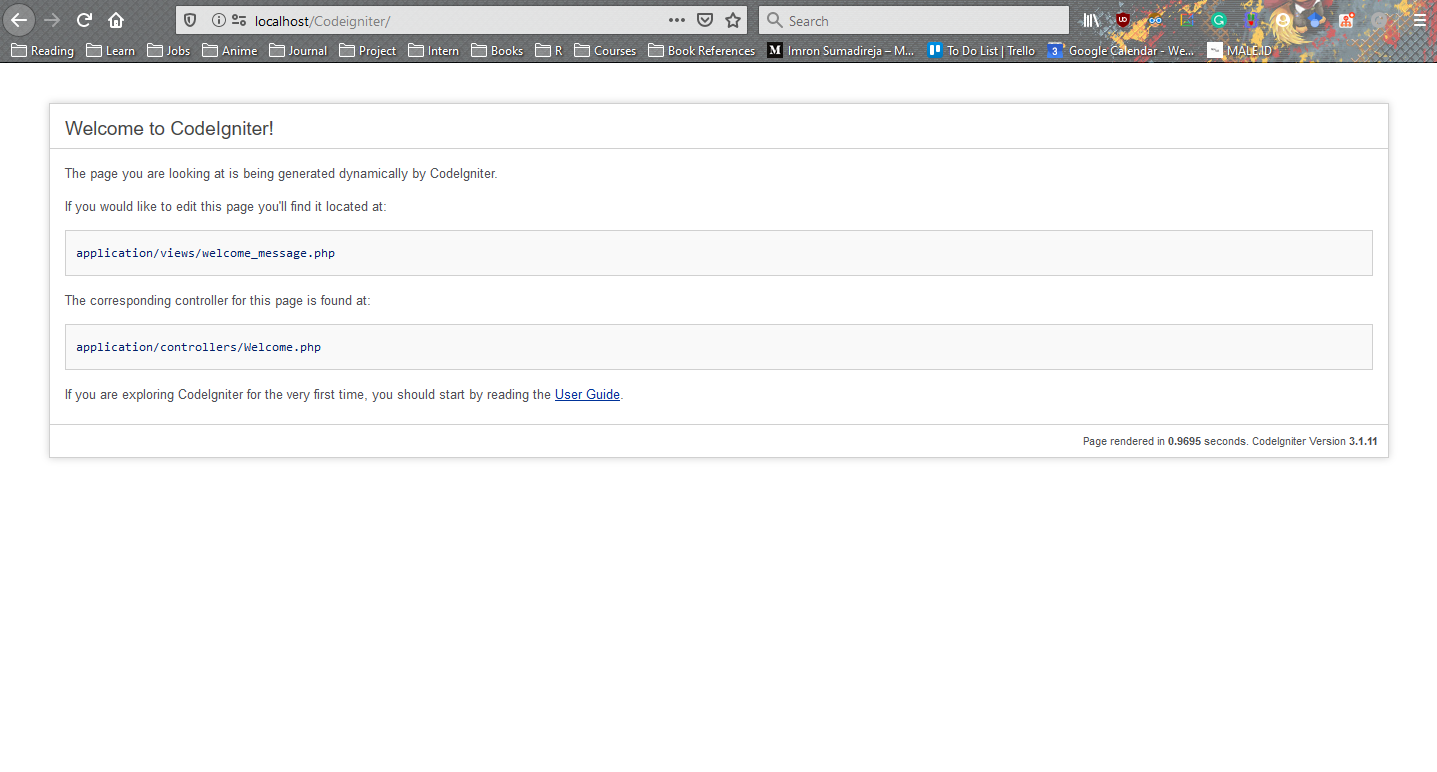
Gambar 3. 19 Copy File Ke Dalam Direktori htdocs.

1. Untuk menjalankannya, kita menggunakan xampp, untuk itu silakan jalankan terlebih dahulu aplikasi xampp-nya seperti berikut.
2. Buka aplikasi xamppnya, kemudian klik tombol start pada kolom actions baris pertama untuk menjalankan apache, dan klik tombol start pada kolom actions baris kedua untuk menjalankan MySQL.



Gambar 3. 20 Menjalankan Apache dan MySQL.

1. Setelah xamppnya berhasil dijalankan, proses selanjutnya tinggal memanggil folder yang telah disimpan pada direktori htdocs di browser. Maka tampilan utamanya seperti gambar berikut.



Gambar 3. 21 Gambar Utama Dari CodeIgniter.

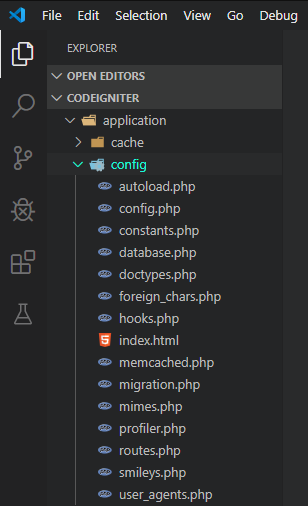
1. Berikut ini merupakan struktur folder yang terdapat pada codeigniter, diantaranya sebagai berikut.



Gambar 3. 22 Struktur Dari CodeIgniter.

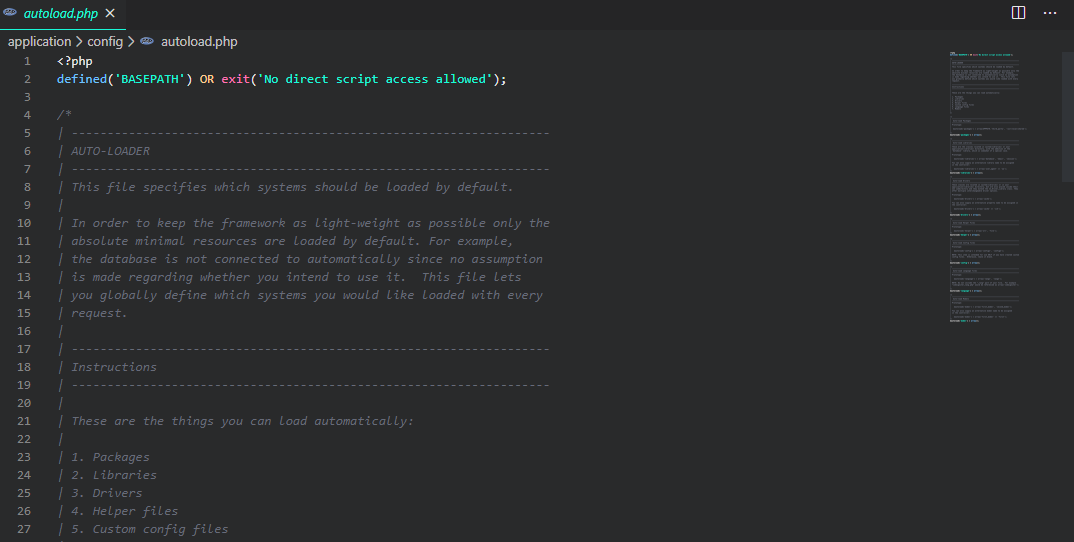
1. *Application*, merupakan folder yang pada dasarnya menyimpan aplikasi yang sedang kita buat.
2. *Cache*, merupakan folder yang menyimpan semua cache yang dibuat oleh cache library.
3. *Config,* merupakan folder yang menyimpan informasi mengenai konfigurasi aplikasi seperti autoload, database, routes, dan lainnya.
4. *Controller,* merupakan folder yang menyimpan controller-controller aplikasi yang dapat digunakan untuk menyusun aktivitas program.
5. *Core,* merupakan folder untuk memperluas *class* inti codeigniter.
6. *Helpers*, merupakan folder untuk menyimpan helpers.
7. *Hooks,* merupakan folder untuk menyimpan hooks untuk mengubah alur fungsi dari core codeigniter.
8. *Language*, merupakan folder untuk menyimpan bahasa-bahasa yang akan digunakan.
9. *Libraries*, merupakan folder untuk menyimpan library.
10. *Logs,* merupakan folder untuk menyimpan semua error log apabila error log diaktifkan.
11. *Models,* merupakan folder untuk menyimpan models yang akan didefinisikan tabel dari database yang dapat kita gunakan oleh controller yang kita buat untuk mengakses database.
12. *Third\_party,* merupakan folder untuk menyimpan fungsi-fungsi tambahan dalam cara kerja codeigniter.
13. *Views,* merupakan folder untuk menyimpan tampilan dari aplikasi yang kita buat.
14. *System*, merupakan folder untuk menyimpan sistem inti dari codeigniter.
    1. Konfigurasi dasar pada codeigniter

Dalam memulai codeigniter, ada beberapa konfigurasi dasar yang perlu kita ketahui, diantaranya autoload.php, config.php dan database.php. Semua konfigurasi pada codeigniter, terletak pada satu tempat yakni di dalam folder *application/config*.



Gambar 3. 23 Konfigurasi CodeIgniter.

1. Autoload.php, file ini digunakan untuk mengatur fungsi-fungsi yang akan dimuat otomatis di awal ketika program dijalankan. Untuk melakukan konfigurasi pada file autoload.php, silakan buka file-nya seperti gambar berikut.



Gambar 3. 24 Konfigurasi Autoload.php

1. Kemudian temukan kode berikut.



Gambar 3. 25 Konfigurasi Autoload Bagian Libraries.

1. Ubah kode tersebut menjadi seperti berikut.



Gambar 3. 26 Menambahkan Database.

Kode tersebut dapat diartikan kita dapat meload library ‘database’ secara otomatis.

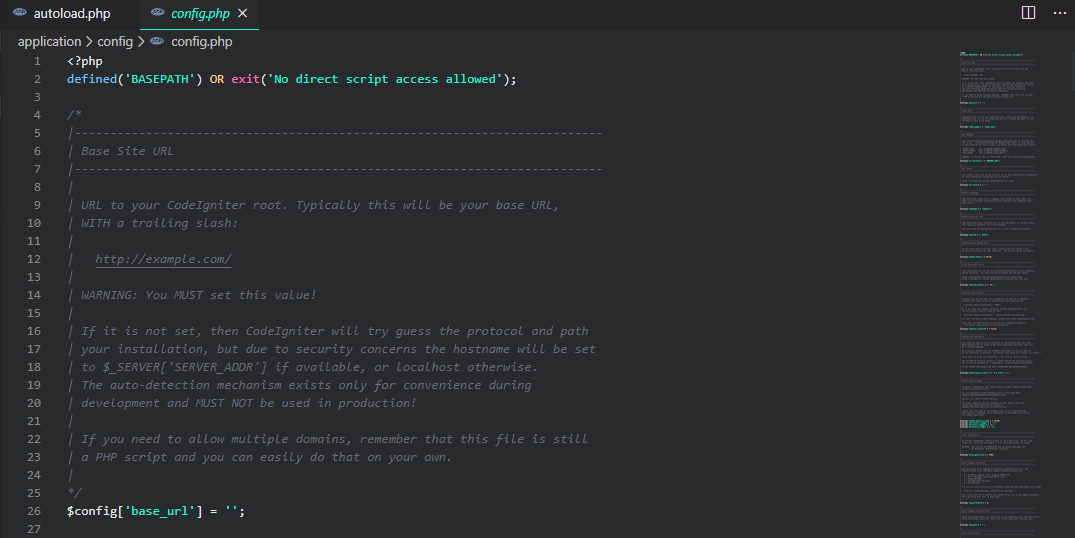
1. Selanjutnya, temukan kode berikut, dan tambahkan ‘url’ didalamnya.



Gambar 3. 27 Konfigurasi Autoload Helper.

Kode tersebut dapat diartikan kita dapat meload helper “url” secara otomatis.

1. Config.php, pada file ini terdapat beberapa konfigurasi yang secara standar sudah terkonfigurasi, namun terdapat beberapa konfigurasi yang perlu diperhatikan, untuk konfigurasi dasar, cukup mengetahui konfigurasi base\_url.



Gambar 3. 28 Buka File Config.php

1. Kemudian temukan kode berikut.



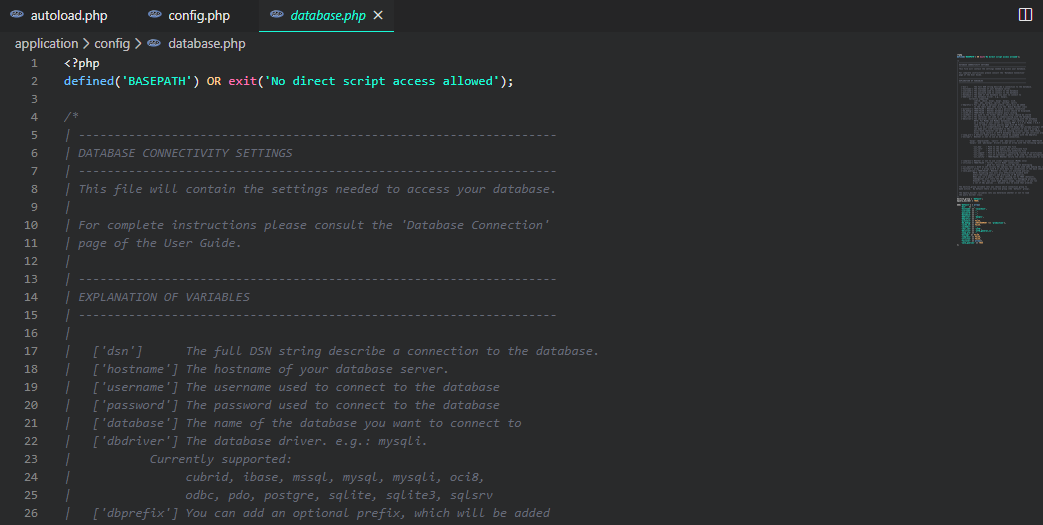
Gambar 3. 29 Konfigurasi Base\_Url.

1. Kemudian tambahkan kode tersebut, menjadi nama folder yang disimpan pada htdocs.



Gambar 3. 30 Menambahkan Base\_URL.

1. Database.php, file ini digunakan untuk melakukan konfigurasi yang berkaitan dengan konfigurasi database dari website yang akan dibuat. Adapun konfigurasi yang perlu diperhatikan, diantaranya: hostname, username, password dan database. Buka file database.php pada teks editor seperti gambar berikut.



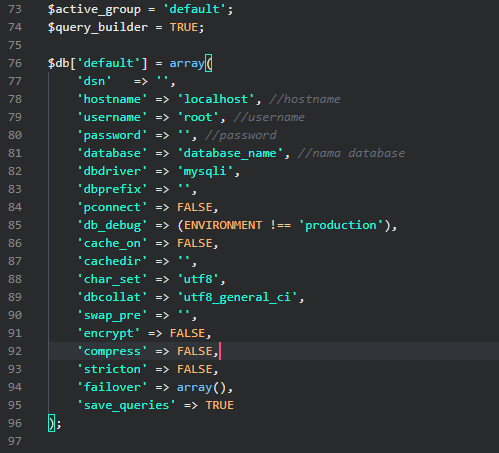
Gambar 3. 31 Buka File Database.php

1. Kemudian temukan kode seperti berikut pada file database.php.



Gambar 3. 32 Konfigurasi File Database.php

1. Kemudian tambahkan kode tersebut seperti gambar berikut.



Gambar 3. 33 Menambahkan Atribut Pada Database.

* 1. *Hello World Codeigniter*

Setelah memahami konsep dasar dari desain MVC, struktur direktori di dalam *framework* CodeIgniter, dan cara kerja dari aplikasi CodeIgniter, pada bagian ini kita akan membahas tentang cara pembuatan aplikasi paling sederhana, yaitu aplikasi dengan menampilkan teks “Hello World!” ke layar *web browser*. Untuk memahami peran dari setiap bagian di dalam pola desain MVC, aplikasi akan dibuat ke dalam 4 versi, yaitu:

1. Versi 1 : Aplikasi yang hanya menggunakan *controller.*
2. Versi 2 : Aplikasi yang menggunakan *controller* dan *model*.
3. Versi 3 : Aplikasi yang menggunakan *controller* dan *view*.
4. Versi 4 : Aplikasi yang menggunakan *controller*, *model,* dan *view*.

**Versi 1 : Hanya Menggunakan *Controller***

Dalam aplikasi dengan menggunakan pola desain MVC, *controller* merupakan komponen utama yang berperan sebagai pemroses permintaan dari *user*. Secara ekstrim, suatu aplikasi berbasis MVC bisa saja dibuat tanpa menggunakan *model* muapun *view*, tetapi tidak mungkin tanpa menggunakan komponen *controller*. Oleh karena itu, kita akan melakukan pembuatan contoh aplikasi mulai dari komponen *controller*.

Pertama, jalankan aplikasi *Text Editor* (Visual Studio Code), kemudian pada bagian *controller* buat file baru dengan nama Hello1.php lalu tuliskan kode berikut:

|  |
| --- |
| <?php  class Hello1 extends CI\_Controller {  public function index(){  echo "<h2> Hello World! (Versi 1)</h2>";  }  } |

Listings 3. 1 Syntax Controller Hello1.

Pada contoh tersebut kita mendefinisikan kelas *controller* dengan nama Hello1. Simpan file tersebut dengan nama Hello1.php (nama file harus sama dengan nama kelas) dan tempatkan di dalam direktori berikut:

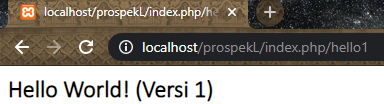
C:\xampp\htdocs\prospekL\application\controllers

Metode *index()* di dalam kelas Hello1 merepresentasikan aksi *default* yang akan dieksekusi ketika *controller* Hello1 dipanggil tanpa menyertakan nama aksi.

Anda dapat menjalankan *controller* tersebut dengan menuliskan *URI* berikut:

<http://localhost/prospekL/index.php/hello1>

Permintaan tersebut akan mengeksekusi metode *index()* yang terdapat di dalam kelas Hello1. Maka hasil yang akan diperoleh adalah sebagai berikut:



Gambar 3. 34 Hasil Dari Syntax Hello 1.

Dalam praktiknya, *controller* seharusnya tidak berisi kode-kode yang berkaitan langsung dengan tampilan karena seharusnya kode tersebut ditulis didalam *view*. Pada contoh tersebut, kita menuliskan kode-kode untuk tampilan *web* di dalam *controller* semata-mata untuk menunjukan cara kinerja *controller*.

**Versi 2 : Menggunakan *Controller* dan *Model***

Aplikasi juga dapat dibuat dengan menggunakan komponen *controller* dan *model*, tanpa *view*; meskipun pada praktiknya hal ini seharusnya tidak dilakukan. Untuk pembuktiannya silakan Anda buat *model* dengan nama M\_Hello2.php, kemudian tuliskan kode berikut:

|  |
| --- |
| <?php  class M\_hello2 extends CI\_Moedl{  //mendefinisikan properti dengan nama $mdl  public $mdl = 'Hello World! (Versi 2)';  } |

Listings 3. 2 Syntax Model Untuk Hello 2.

Kode tersebut simpan ke dalam direktori berikut:

C:\xampp\htdocs\prospekL\application\models

Selanjutnya, buatlah file *controller* baru dengan nama file Hello2.php dan tuliskan kode berikut:

|  |
| --- |
| <?php  class Hello2 extends CI\_Controller {  public function index(){  //Memuat model dengan nama Hello2\_model  $this->load->model('Hello2\_model');  //Menangkap objek dari kelas Hello2\_model  $model = $this->Hello2\_model;  echo "<html>";  echo "<head><title>Controller dan Model</title></head>";  echo "<body>";  //Memanggil properti $mdl milik objek $model  echo "<h1>".$model->mdl."</h1>";  echo "</body>";  echo "</html>";  }  } |

Listings 3. 3 Syntax Controller Untuk Hallo 2.

Pada contoh versi 2 ini yang akan ditampilkan ke laya *web browser* disimpan di dalam *model*. Agar *controller* dapat mengenal *model* tersebut, *model* harus dimuat terlebih dahulu menggunakan baris kode berikut:

# 

Gambar 3. 35 Syntax Untuk Memuat Model.

# Proses pemuatan pada kode tersebut akan disertai dengan proses pembuatan objek dari kelas Hello2\_model, yang dilakukan secara implisit oleh CodeIgniter. Untuk mengambil data dari properti *$mdl* yang didefinisikan di dalam kelas Hello2\_model, kita perlu mengakses objek dari kelas bersangkutan, dengan cara berikut:



Gambar 3. 36 Syntax Untuk Mengakses Objek Pada Model.

Selanjutnya, dapat diambil melalui objeknya seperti berikut:

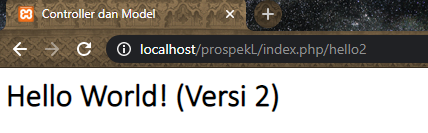


Gambar 3. 37 Syntax Untuk Mendapatkan Data Dari Model.

Jalankan aplikasi tersebut dengan menggunakan *URI* berikut:

<http://localhost/prospekL/index.php/hello2>

Hasil yang akan diperoleh adalah sebagai berikut:



Gambar 3. 38 Hasil Dari Syntax Hello 2.

**Versi 3 : Menggunakan *Controller* dan *View***

Untuk melakukan implementasi pada komponen *controller* dan *view*, buat terlebih dahulu *view* dengan kode sebagai berikut:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Controller dan View</title>  </head>  <body>  <h1>Hello World! (Versi 3)</h1>  </body>  </html> |

Listings 3. 4 Syntax View Pada Hello 3.

Simpan file tersebur dan tempatkan di dalam direktor berikut:

C:\xampp\htdocs\prospekL\application\views

Selanjutnya, buat kembali file *controller* dengan nama file Hello3.php dengan kode sebagai berikut:

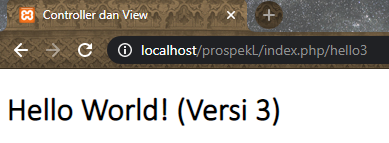
|  |
| --- |
| <?php  class Hello3 extends CI\_Controller {  public function index(){  //memuat viewhello3  $this->load->view('viewhello3');  }  } |

Listings 3. 5 Syntax Controller Pada Hello 3.

Jalankan aplikasi tersebut dengan menggunakan *URI* berikut:

<http://localhost/prospekL/index.php/hello3>

Hasilnya akan tampak sebagai berikut:



Gambar 3. 39 Hasil Dari Syntax Hello 3.

Pada contoh ini, *controller* hanya memanggil file *view*. Semua kode yang berkaitan dengan tampilan ditangani sepenuhnya oleh *view*. Proses pemuatan *view* dari *controller* dilakukan menggunakan baris kode berikut ini:



Gambar 3. 40 Syntax Untuk Memanggil View.

**Versi 4 : Menggunakan *Controller, Model*, dan *View***

Aplikasi dengan komponen *controller*, *model*, dan *view* merupakan aplikasi yang lengkap. Pada praktiknya, teknik inilah yang seharusnya digunakan oleh setiap aplikasi yang dikembangkan menggunakan CodeIgniter. Urutan proses dari aplikasi yang akan kita buat pada bagian ini adalah sebagai berikut:

1. *Controller* akan mengambil data dari *model*.
2. *Controller* akan memuat *view* dengan mengirimkan data yang diambil dari *model* ke *view*.
3. *View* akan memproses data yang dikirim oleh *controller* menjadi suatu tampilan.
4. *Controller* akan menyajikan tampilan yang dihasilkan oleh *view* ke layar *web browser*.

Untuk melakukan implementasi pada komponen *controller, model,* dan *view*. Buatlah tiga buah file dengan nama Hello4.php (*controller*), M\_hello4.php (*model*), dan viewhello4.php (*view*). Berikut merupakan kode dari masing-masing file tersebut.

File *controller* dengan nama file Hello4.php:

|  |
| --- |
| <?php  class Hello4 extends CI\_Controller {  public function index(){  //memuat model "M\_hello4"  $this->load->model('M\_hello4');  //menangkap objek dari kelas M\_hello4  $model = $this->M\_hello4;  //mengambil data dari model  $H = $model->mdl;  //membuat data yang akan dikirimkan ke view  $data['teks'] = $H;  //memuat view 'viewhello4'  $this->load->view('viewhello4', $data);  }  } |

Listings 3. 6 Syntax Controller Untuk Hello 4.

File model dengan nama file M\_hello4.php:

|  |
| --- |
| <?php  class M\_hello4 extends CI\_Model{  //mendefinisikan properti dengan nama $str  public $ = 'Hello World! (Versi 4)';  } |

Listings 3. 7 Syntax Controller Pada Hello 4.

File *view* dengan nama file viewhello4.php:

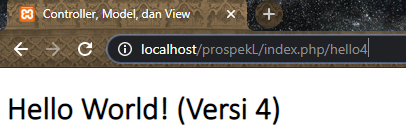
|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Controller, Model, dan View</title>  </head>  <body>  <h1><?php echo $teks; ?></h1>  </body>  </html> |

Listings 3. 8 Syntax View Untuk Hello 4.

Jalankan aplikasinya dengan menggunakan *URI* berikut:

<http://localhost/prospekL/index.php/hello4>

Hasil yang akan diperoleh adalah sebagai berikut:



Gambar 3. 41 Hasil Dari Syntax Hello 4.

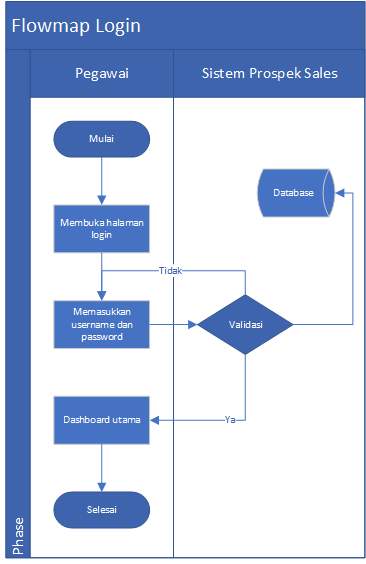
## PERANCANGAN SISTEM

Analisis merupakan penelaahan atau penelitian dilengkapi dengan hasil percobaan dengan melakukan suatu percobaan pada proses siklik dan perbandingan yang menghasilkan kesimpulan dari penguraian suatu sistem informasi yang utuh ke dalam bagian-bagian kompenennya dengan maksud untuk mengidentifikasi dan mengevaluasi segala permasalahan yang timbul, sehingga dapat diusulkan perbaikan-perbaikan yang dapat membangun.

Analisis yang dimaksud disini berupa analisis *flowmap* mengenai sistem yang akan dibangun meluputi prosedur *login,* prosedur kelola data *addOn*, prosedur daftar akun pegawai, dan prosedur prediksi pendapatan. Adapun *flowmap* yang akan dibangun adalah sebagai berikut.

1. Analisis *Flowmap* Prosedur Login

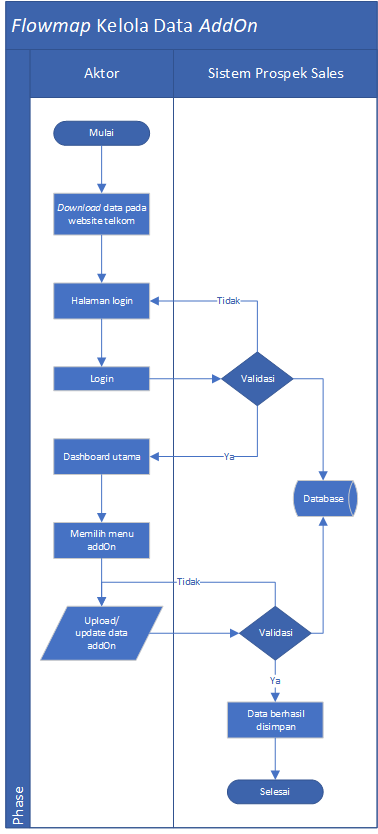
Pada prosedur *login* melibatkan dua entitas (aktor) yaitu Admin dan Pegawai. Dimana setiap entitas (aktor) tersebut memiliki hak akses yang berbeda terhadap sistem atau aplikasi yang dibuat. Admin memiliki semua hak akses menu, sedangkan Pegawai hanya memiliki hak akses prediksi pendapatan, dan *update* data *addOn* saja, adapun *flowmap* prosedur *login*, yaitu sebagai berikut:



Gambar 4. 1 Flowmap Login.

1. Analisis *Flowmap* Prosedur Kelola Data *addOn*

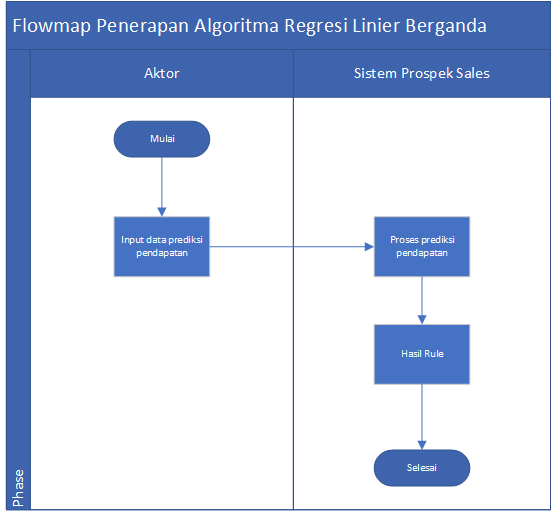
Pada prosedur kelola data *addOn*, melibatkan dua entitas (aktor) yaitu, admin dan pegawai. Pertama, aktor *login* terlebih dahulu, lalu masuk ke halaman utama. Didalam kelola data *addOn* aktor diberikan hak akses untuk menambahkan dan mengedit pelanggan yang menggunakan produk dari unit *digital service*. Adapun *flowmap* kelola data *addOn* adalah sebagai berikut.



Gambar 4. 2 Flowmap prosedur kelola data addOn.

1. Analisis *Flowmap* Prosedur Prediksi Pendapatan

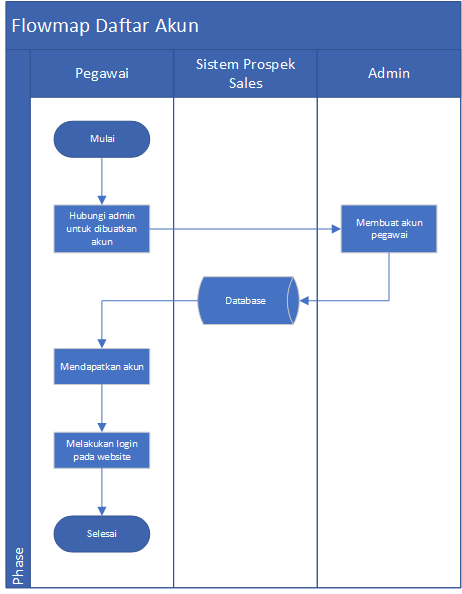
Pada prosedur penerapan algoritma regresi linier berganda melibatkan dua aktor, yaitu admin dan pegawai. Pertama, admin *login* terlebih dahulu, lalu masuk ke menu prediksi pendapatan di dalam menu prediksi admin dapat menginput data prediksi dengan ekstensi xlsx, lalu sistem akan memproses data tersebut dan menghasilkan *rule* dari data yang telah diinputkan. Adapun *flowmap* penerapan algoritma regresi linier berganda adalah sebagai berikut:



Gambar 4. 3 Flowmap Prosedur Penerapan Algoritma Regresi Linier Berganda.

1. Analisis *Flowmap* Prosedur Kelola Akun

Pada prosedur kelola akun melibatkan satu entitas (aktor) yaitu, admin (Unit *Digital Santai & Ceria* Talaka Regional II). Pertama, admin *login* terlebih dahulu, lalu masuk ke menu kelola akun. Didalam menu kelola akun admin diberikan hak akses untuk menambahkan, mengubah, dan menghapus data akun yang nantinya akan masuk ke dalam sistem *database*. Adapun *flowmap* kelola akun adalah sebagai berikut.



Gambar 4. 4 Flowmap Prosedur Kelola Akun.

Analisis dokumen aplikasi merupakan suatu kebutuhan dokumen yang berhubungan dengan dokumen sistem yang akan dibuat. Dimana menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sebuah sistem.

Adapun kebutuhan aplikasi yang akan dibuat yaitu pengelolaan data prospek *sales* dengan prediksi pendapatan sesuai dengan urutan sebagai berikut.

1. *Login* admin (*Manager Unit Digital Santai & Ceria Tareg II* yang diberi hak akses sebagai admin) dan pegawai;
2. Kelola data *addOn*;
3. Kelola data prediksi pendapatan;
4. Kelola akun.

Setiap proses memiliki representasi masing-masing pada sebuah tabel atau data yang terdapat pada *database* yang telah dirancang sebelumnya, dan setiap proses berhubungan langsung dengan entitas atau aktor.

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Tahapan ini meliputi mengkonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

**UML (*Unified Modelling Language*)**

Proses analisis menggunakan UML harus diberikan spesifikasi prosesnya dan dijelaskan dengan tulisan secara lengkap. Terdapat beberapa diagram pada UML, diantaranya:

1. *Use case diagram*
2. *Class diagram*
3. *Sequence diagram*
4. *Activity diagram*
5. *Collaboration diagram*
6. *Statechart diagram*
7. *Component diagram*
8. *Deployment diagram*

*Use case diagram* merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antara aktor dengan aktivitas yang terdapat pada sistem. Adapun *use case diaram* pada aplikasi peramalan pendapatan PT Talaka Indonesia dengan paket internet Indieland, yaitu sebagai berikut.

1. ***Use Case Diagram***



Gambar 4. 5 Use case diagram peramalan pendapatan.

Pada *use case* tersebut terdapat dua entitas (aktor) yakni, admin dan pegawai yang telah diberikan akses sesuai dengan entitasnya masing-masing. Berikut ini merupakan deskripsi dari aktor-aktor yang terlibat dalam aplikasi, diantaranya sebagai berikut.

Tabel 4. 1 Definisi Aktor.

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Admin | * 1. *Login*   2. Kelola data prospek *sales*   3. Kelola data pegawai   4. Prediksi pendapatan |
| 2 | Pegawai | 1. *Login* 2. Kelola data prospek *sales* 3. Prediksi pendapatan |

Skenario *use case* mendeskripsikan urutan langkah-langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berikut ini merupakan penjelasan dari masing-masing skenario tersebut.

1. Skenario *Use Case Login*

Adapun interaksi antara aktor pengguna dengan *use case* *login* dijelaskan dalam tabel berikut ini.

Tabel 4. 2 Skenario Use Case Login.

|  |  |
| --- | --- |
| Identifikasi | |
| Nama | *Login* |
| Tujuan | Masuk ke dalam sistem sebagai administrator atau pegawai |
| Aktor | Admin, Pegawai |
| Deskripsi | Proses login ini untuk masuk ke halaman admin jika admin yang melakukan proses login, dan jika pegawai yang melakukan proses login maka hanya akan masuk ke menu prediksi dan *dashboard* utama. |
| Skenario utama | |
| Kondisi awal | From *login* di tampilkan |
| Aksi Aktor | Reaksi sistem |
| 1. Memasukan *Username* dan *Password* | 2. Mencocokan data *login* dengan data admin dan pegawai pada basis data |
|  | 3. Bila valid akan menampilkan halaman admin untuk admin dan hanya halaman prediksi dan *dashboard* utama untuk pegawai |
| Skenario Alternatif (jika gagal) | |
| Aksi actor | Reaksi Sistem |
|  | 1. Menampilkan pesan |
| 2. Memasukan *Username* dan *Password* |  |
| Kondisi akhir | Admin dapat melakukan kegiatan pada sistem sesuai kewenangan sebagai administrator |

1. Skenario *Use Case* Prediksi Pendapatan

Adapun interaksi antara aktor dengan *use case* prediksi pendapatan dijelaskan dalam tabel sebagai berikut.

Tabel 4. 3 Skenario Use Case Prediksi Pendapatan.

|  |  |
| --- | --- |
| Identifikasi | |
| Nama | Prediksi Pendapatan |
| Tujuan | Untuk melakukan prediksi pendapatan dari produk *digital service*. |
| Aktor | Admin, Pegawai |
| Use case yang berkaitan | *Login* |
| Deskripsi | Proses login ini untuk masuk ke halaman admin jika admin yang melakukan proses login, dan jika pegawai yang melakukan proses login maka hanya akan masuk ke menu prediksi dan *dashboard* utama. |
| Skenario utama | |
| Kondisi awal | Menampilkan form upload data prediksi |
| Aksi Aktor | Reaksi sistem |
| 1. Aktor memilih menu prediksi pendapatan | 2. Menampilkan form prediksi pendapatan |
| 3. Menekan tombol *upload* untuk memasukkan data prediksi pada sistem | 4. Akan melakukan proses prediksi pendapatan pada produk *digital service* dengan algoritma regresi linier berganda |
| Kondisi akhir | Data prediksi pendapatan pada produk *digital service* berhasil diolah |

1. Skenario *Use Case* Kelola Data *addOn*

Adapun interaksi aktor dengan *use case* kelola data *addOn* dijelaskan dalam tabel sebagai berikut.

Tabel 4. 4 Skenario Use Case Kelola Data addOn.

|  |  |
| --- | --- |
| Identifikasi | |
| Nama | Kelola data prospek *sales* |
| Tujuan | Untuk mengelola data pelanggan yang menggunakan produk *digital santai*. |
| Aktor | Admin, Pegawai |
| Use case yang berkaitan | *Login* |
| Deskripsi | Proses kelola data prospek *sales* ini digunakan untuk *upload* dan *update* data pelanggan yang menggunakan produk *digital santai.* |
| Skenario utama | |
| Kondisi awal | Menampilkan data prospek *sales* produk *digital santai.* |
| Aksi Aktor | Reaksi sistem |
| 1. Aktor memilih menu *dashboard* utama | 2. Menampilkan *dashboard* utama |
| 3. Menekan link pada *box* *addOn* produk *digital santai* | 4. Akan menampilkan *form* upload data produk *digital santai* |
| Kondisi akhir | Data pelanggan telah berhasil di-*upload* pada masing-masing menu *addOn*. |

1. Skenario *Use Case* Kelola Akun

Adapun interaksi antara aktor dengan *use case* kelola akun pegawai dijelaskan dalam tabel sebagai berikut.

Tabel 4. 5 Skenario Use Case Kelola Akun.

|  |  |
| --- | --- |
| Identifikasi | |
| Nama | Kelola data akun pegawai |
| Tujuan | Untuk mengelola data akun pegawai |
| Aktor | Admin |
| Use case yang berkaitan | *Login* |
| Deskripsi | Proses kelola data akun pegawai ini digunakan untuk menambahkan *user* baru, mengubah, dan menghapus data akun pegawai |
| Skenario utama | |
| Kondisi awal | Menampilkan tabel data akun pegawai |
| Aksi Aktor | Reaksi sistem |
| 1. Aktor memilih menu tambah akun | 2. Menampilkan *form* tambah akun |
| 3. Menekan tombol tambah data, lalu mengisi informasi data, dan menekan tombol *save* | 4. Akan menyimpan data ke dalam *database* |
| 5. Menekan tombol ubah untuk mengubah data akun pegawai | 6. Akan mengubah informasi data didalam *database* |
| Kondisi akhir | Data akun pegawai berhasil diolah |

1. ***Class Diagram***

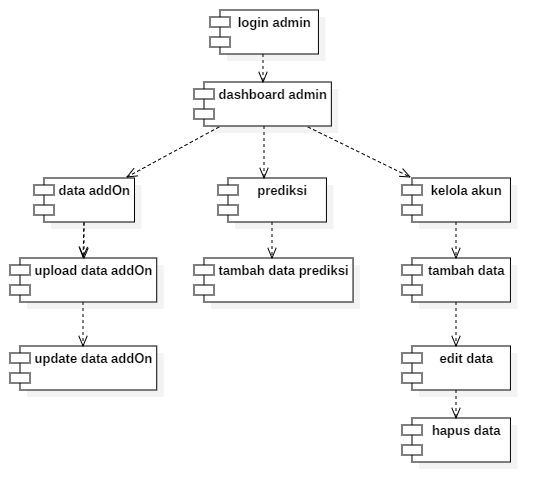
*Class diagram* menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan metode-metode yang ada pada masing-masing kelas.



Gambar 4. 6 Class Diagram.

1. ***Component Diagram***

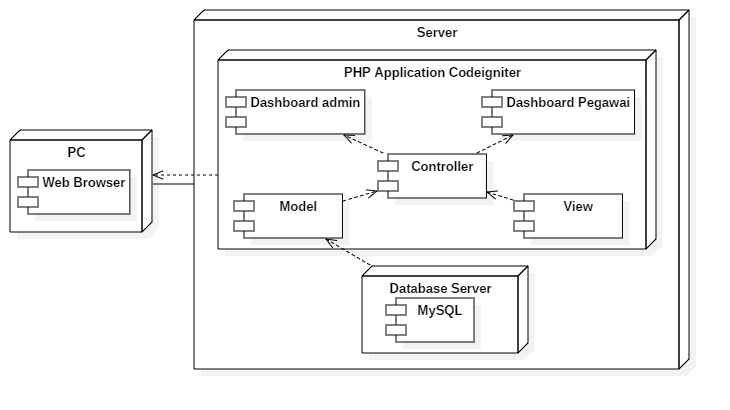
*Component diagram* adalah *diagram* yang digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen *software* sistem. *Component diagram* berguna untuk memodelkan komponen objek. Adapun *component diagram* berguna untuk memodelkan komponen objek. *Component diagram* pada aplikasi peramalan ini, yakni sebagai berikut:

****

Gambar 4. 7 Component Diagram.

1. ***Deployment Diagram***

*Deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. *Diagram deployment* juga dapat digunakan untuk memodelkan hal-hal seperti sistem tambahan (*embedded system*) yang menggambarkan rancangan *device, node,* dan *hardware*, sistem *clien/server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi.



Gambar 4. 8 Deployment Diagram.

# METODE REGRESI LINIER BERGANDA

Analisis regresi merupakan sebuah alat statistik yang memberikan penjelasan tentang pola hubungan (model) antara dua variabel atau lebih. Dalam analisis regresi, dikenal dua jenis variabel yaitu :

1. Variabel tergantung disebut juga variabel dependent yaitu variabel yang keberadaannya dipengaruhi oleh variabel lainnya yang sifatnya tidak dapat berdiri sendiri dan dinotasikan dengan Y.
2. Variabel bebas disebut juga variabel indendepen yaitu variabel yang mempengaruhi variabel lain yang sifatnya berdiri sendiri dan dinotasikan dengan X.

Analisis regresi linier berganda memberikan kemudahan bagi pengguna untuk memasukkan lebih dari satu variabel prediktor hingga p-variabel prediktor dimana banyaknya p kurang dari jumlah observasi (n). Sehingga model regresi dapat ditunjukkan sebagai berikut:



Gambar 5. 1 Persamaan Regresi Linier Berganda

Keterangan:

Y = variabel dependen (nilai yang diprediksikan)

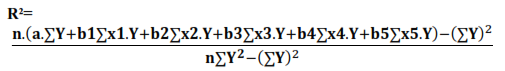
X1, x2, …, xn = variabel independen

A = konstanta (nilai yang Y apabila X1, X2, Xn = 0)

B1, b2, bn = koefisien regresi (nilai peningkatan ataupun penurunan).

**Koefisien Determinasi**

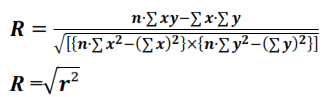
Koefisien determinasi (*R2*) ini disebut juga dengan koefisien penentu sampe artinya menyatakan proporsi variasi dalam nilai Y (peubah tidak bebas) yang disebabkan oleh hubungan linier dengan X (peubah bebas) berdasarkan persamaan (model matematis) regresi yang didapat[13].



Gambar 5. 2 Persamaan Koefiein Determinasi.

**Koefisien Korelasi**

Untuk mengetahui kuatnya hubungan antara variabel dependen dengan variabel independen diukur dengan koefisien korelasi (R) adalah suatu ukuran relatif dari asosiasi diantara dua variabel[13]. Koefisien ini bervariasi dari -1 sampai dengan +1. Angka koefisien korelasi dan determinasi dapat dihitung dengan rumus sebagai berikut:



Gambar 5. 3 Persamaan Koefisien Korelasi.

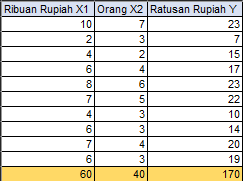
Keterangn:

Y = variabel terikat (*dependen*)

X = variabel bebas (*independen*)

N = jumlah data

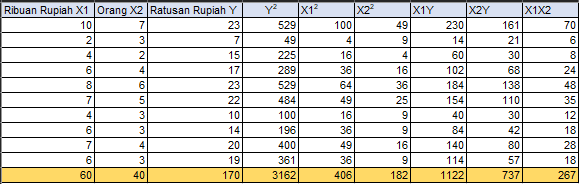
Untuk mengetahui bagaimana metode regresi linier berganda ini digunakan, berikut akan dijabarkan mengenai contoh perhitungan dengan menggunakan regresi linier berganda. Dalam contoh kasus berikut merupakan data terhadap 10 rumah tangga yang dipilih secara acak, untuk memperoleh data pengeluaran pembelian barang-barang tahan lama per minggu (Y), pendapatan per minggu (X1), dan jumlah anggota rumah tangga (X2), sebagai berikut.



Gambar 5. 4 Data Utama.

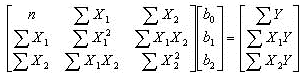
Seandainya suatu rumah tangga mempunyai X1 dan X2, dengan masing-masing 11 dan 8. Berapa besar nilai Y. Artinya, berapa ratus rupiah rumah tangga yang bersangkutan akan mengeluarkan biaya untuk pembelian barang-barang dengan kualitas tahan lama.

Langkah pertama adalah mengolah data tersebut menjadi sebagai berikut.



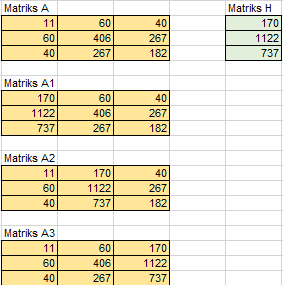
Gambar 5. 5 Hasil Perhitungan RLB.

Setelah data tersebut diolah, langkah selanjutnya adalah menentukan nilai matriks A, matriks A1, matriks A2, dan matriks A3. Kita tinggal memasukkan nilai sesuai dengan rumus seperti berikut.



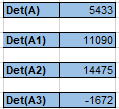
Gambar 5. 6 Rumus Mencari Matriks.

Maka, hasilnya dapat dilihat seperti berikut. Untuk matriks H, itu merupakan nilai sigma Y / total dari Y, sigma X1Y / total dari X1Y, dan sigma X2Y / total dari X2Y.



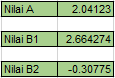
Gambar 5. 7 Hasil Matriks.

Selanjutnya melakukan perhitungan untuk mencari nilai Determinan dari setiap nilai matriks yang sudah ditentukan tersebut. Berikut untuk tahapan-tahapan dalam melakukan perhitungan deteriminannya.



Gambar 5. 8 Hasl Determinan.

Nilai tersebut didapatkan dengan menggunakan rumus yang terdapat dalam *Microsoft Excel*, yakni MDETERM dengan melakukan cara tersebut dapat dihasilkan nilai determinan dari tabel matriks yang sudah ditentukan. Selanjutnya melakukan perhitungan untuk mencari nilai A, B1, dan B2 agar dapat melakukan prediksi dengan menggunakan regresi linier berganda. Berikut merupakan tahapan perhitungannya.



Gambar 5. 9 Hasil Nilai a, b1, dan b2.

Nilai A didapatkan dengan cara membagi nilai determinan A1 dengan determinan A, sedangkan untuk mendapatkan nilai B1 nilai determinan A2 dibagi dengan nilai determinan A, begitupun dengan nilai B2 nilai determinan A3 dibagi dengan nilai determinan A. Langkah terakhir dari perhitungan regresi linier berganda ini, kita tinggal memasukkan nilai-nilai yang sudah didapatkan tersebut ke dalam rumus regresi linier berganda, yaitu seperti berikut.





Gambar 5. 10 Hasil Perhitungan RLB.

Dari model tersebut dapat disimpulkan bahwa jumlah anggota rumah tangga 3 dengan pendapatan per minggu Rp. 6.000, maka pengeluaran untuk pembelian barang-barang dengan kualitas tahan lama per minggu sebesar Rp. 1.710,63.

REGRESI LINIER SEDERHANA

Analisis regresi adalah suatu metode statistik yang mengamati hubungan antara variabel terikat Y dan serangkaian variabel bebas X1, X2, Xn. Tujuan dari metode ini adalah untuk memprediksi nilai Y untuk nilai X yang diberikan. Model regresi linier sederhana adalah model regresi yang paling sederhana yang hanya memiliki satu variabel bebas X. Analisis regresi memiliki beberapa kegunaan, salah satunya untuk melakukan prediksi terhadap variabel terikat Y[14]. Persamaan untuk model regresi linier sederhana adalah sebagai berikut.



Gambar 5. 11 Persamaan Regresi Linier Sederhana.

Y adalah variabel terikat yang diramalkan, X adalah variabel bebas, a adalah *intercep*, yaitu nilai Y pada saat X=0, dan b adalah *slope*, yaitu perubahan rata-rata Y terhadap perubahan satu unit X. Koefisien a dan b adalah koefisien regresi dimana nilai a dan b dapat dicari menggunakan persamaan berikut.



Gambar 5. 12 Persamaan Koefisien a dan b.

Nilai a adalah *slope*, b adalah *intercep* dan n adalah banyaknya data yang digunakan dalam perhitungan.

PERBANDINGAN REGRESI LINIER BERGANDA DENGAN REGRESI LINIER SEDERHANA

Sebenarnya tidak terdapat perbedaan signifikan antara analisis regresi linier sederhana dengan berganda, yaitu pada regresi berganda variabel independen adalah lebih dari satu.

Dalam analisis regresi empiris banyak dijumpai hubungan sebab akibat yang kuat antara variabel dependen dan variabel-variabel independennya.

Asumsi Regresi linier, terbagi menjadi 4 asumsi, diantaranya sebagai berikut.

1. Heteroskedastisitas

Dengan pengujian ini dapat dideteksi apakah kesalahan pengganggu dari model yang diamati tidak memiliki varians yang konstan dari satu observasi ke obeservasi lainnya.

1. Autokorelasi

Untuk mengetahui autokorelasi digunakan uji Durbin-Watson(DW-test). Adanya autokorelasi dalam regresi dapat diketahui dengan menggunakan beberapa cara antara lain metode grafik dan uji Durbin-Watson.

1. Multikolinearitas

Multikolinearitas muncul jika terdapat hubungan yang sempurna atau pasti di antara beberapa variabel atau semua variabel independen dalam model. Untuk mendeteksi multikolinearitas digunakan pengukuran terhadap nilai VIF (Variable Inflation Factor) dan nilai Tolerance.

1. Normalitas

Uji normalitas bertujuan untuk menguji apakah dalam model regresi, variabel pengganggu atau residual memiliki distribusi normal. Jika asumsi ini dilanggar, maka uji statistik menjadi tidak valid atau bias terutama untuk sampel kecil. Uji normalitas dapat dilakukan melalui dua pendekatan yaitu melalui pendekatan grafik (histogram dan P-P plot) atau uji kolmogorov-smirnov, chi-square, Liliefors maupun Shapiro-Wilk[15].

# TUTORIAL MEMBUAT SISTEM PERAMALAN

Pada tutorial membuat sistem prospek sales ini, akan dijelaskan ke dalam beberapa tahap, diantaranya.

1. Perancangan *Database*

Basis data (*database*) merupakan salah satu komponen yang penting dalam pembuatan sistem informasi, karena basis data merupakan hal pokok dalam menyediakan informasi tentang data kepada para pengguna khususnya. Berikut merupakan beberapa tabel yang terdapat pada *database* sistem.

Tabel 6. 1 Indieland.

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Kawasan | Varchar | 50 | Kawasan |
| Witel | Varchar | 50 | Wilayah Talaka |
| Datel | Varchar | 50 | Daerah Talaka |
| STO | Varchar | 20 | STO |
| NCLI | Varchar | 20 | NCLI |
| NDOS | Varchar | 20 | NDOS |
| NDEM | Varchar | 20 | NDEM |
| NO\_INET | Int | 20 | Nomor Internet |
| ND | Varchar | 20 | ND |
| CHANEL | Varchar | 50 | Chanel |
| Citem\_Cepat | Varchar | 50 | Citem Cepat |
| Kecepatan | Int | 20 | Kecepatan internet |
| Deskripsi | Varchar | 100 | Deskirpsi |
| Tgl\_Reg | Date |  | Tanggal registrasi |
| Tgl\_Etat | Date |  | Tanggal aktif |
| Status | Varchar | 50 | Status |
| Nama | Varchar | 100 | Nama |
| Kcontact | Varchar | 255 | Informasi pelanggan |
| Status\_order | Varchar | 50 | Status order |
| Alpro | Varchar | 20 | Alpro |
| CCAT | Varchar | 20 | CCAT |
| Jalan | Varchar | 50 | Alamat pelanggan |
| Nojalan | Varchar | 50 | No Jalan pelanggan |
| Distrik | Varchar | 50 | Distrik pelanggan |
| Kota | Varchar | 50 | Kota pelanggan tinggal |
| Cpack | Varchar | 50 | Paket yang digunakan |

Tabel 6. 2 Movieyuk.

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Witel | Varchar | 50 | Wilayah Talaka |
| NCLI | Varchar | 20 | NCLI |
| NDOS | Varchar | 20 | NDOS |
| NDEM | Varchar | 20 | NDEM |
| No\_Inet | Int | 20 | Nomor Internet |
| Item | Varchar | 50 | Item |
| Price | Int | 20 | Harga Produk |
| Tgl\_Va | Date | - | Tanggal Registrasi |
| Tgl\_Ps | Date | - | Tanggal Aktif |
| Kcontact | Varchar | 255 | Kontak Pelanggan |

Tabel 6. 3 Indieland Pantai.

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Witel | Varchar | 50 | Wilayah Talaka |
| NCLI | Varchar | 20 | NCLI |
| NDOS | Varchar | 20 | NDOS |
| NDEM | Varchar | 20 | NDEM |
| No\_Inet | Int | 20 | Nomor Internet |
| Item | Varchar | 50 | Item |
| Price | Int | 20 | Harga Produk |
| Tgl\_Va | Date | - | Tanggal Registrasi |
| Tgl\_Ps | Date | - | Tanggal Aktif |
| Kcontact | Varchar | 255 | Kontak Pelanggan |

Tabel 6. 4 OTM.

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Witel | Varchar | 50 | Wilayah Talaka |
| NCLI | Varchar | 20 | NCLI |
| NDOS | Varchar | 20 | NDOS |
| NDEM | Varchar | 20 | NDEM |
| No\_Inet | Int | 20 | Nomor Internet |
| Item | Varchar | 50 | Item |
| Price | Int | 20 | Harga Produk |
| Tgl\_Va | Date | - | Tanggal Registrasi |
| Tgl\_Ps | Date | - | Tanggal Aktif |
| Kcontact | Varchar | 255 | Kontak Pelanggan |

Tabel 6. 5 Indieland Jazz.

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Witel | Varchar | 50 | Wilayah Talaka |
| NCLI | Varchar | 20 | NCLI |
| NDOS | Varchar | 20 | NDOS |
| NDEM | Varchar | 20 | NDEM |
| No\_Inet | Int | 20 | Nomor Internet |
| Item | Varchar | 50 | Item |
| Price | Int | 20 | Harga Produk |
| Tgl\_Va | Date | - | Tanggal Registrasi |
| Tgl\_Ps | Date | - | Tanggal Aktif |
| Kcontact | Varchar | 255 | Kontak Pelanggan |

Tabel 6. 6 Video.

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Witel | Varchar | 50 | Wilayah Talaka |
| NCLI | Varchar | 20 | NCLI |
| NDOS | Varchar | 20 | NDOS |
| NDEM | Varchar | 20 | NDEM |
| No\_Inet | Int | 20 | Nomor Internet |
| Item | Varchar | 50 | Item |
| Price | Int | 20 | Harga Produk |
| Tgl\_Va | Date | - | Tanggal Registrasi |
| Tgl\_Ps | Date | - | Tanggal Aktif |
| Kcontact | Varchar | 255 | Kontak Pelanggan |

Tabel 6. 7 Indieland Pack.

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Witel | Varchar | 50 | Wilayah Talaka |
| NCLI | Varchar | 20 | NCLI |
| NDOS | Varchar | 20 | NDOS |
| NDEM | Varchar | 20 | NDEM |
| No\_Inet | Int | 20 | Nomor Internet |
| Item | Varchar | 50 | Item |
| Price | Int | 20 | Harga Produk |
| Tgl\_Va | Date | - | Tanggal Registrasi |
| Tgl\_Ps | Date | - | Tanggal Aktif |
| Kcontact | Varchar | 255 | Kontak Pelanggan |

Tabel 6. 8 Indieland Lokal.

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Witel | Varchar | 50 | Wilayah Talaka |
| NCLI | Varchar | 20 | NCLI |
| NDOS | Varchar | 20 | NDOS |
| NDEM | Varchar | 20 | NDEM |
| No\_Inet | Int | 20 | Nomor Internet |
| Item | Varchar | 50 | Item |
| Price | Int | 20 | Harga Produk |
| Tgl\_Va | Date | - | Tanggal Registrasi |
| Tgl\_Ps | Date | - | Tanggal Aktif |
| Kcontact | Varchar | 255 | Kontak Pelanggan |

Tabel 6. 9 Data Prediksi

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Field Size* | Keterangan |
| Id | Int | 100 | Id rlb |
| Bulan | Varchar | 10 | Bulan |
| Jumlah\_pelanggan | Int | 10 | Jumlah pelanggan |
| Jumlah\_produk | Int | 5 | Jumlah produk |
| Pendapatan | Int | 10 | Pendapatan |

1. Perancangan *Form Login*

Halaman *login* merupakan halaman untuk masuk ke masing-masing menu utama admin dan pegawai, jika yang masuk admin, akan masuk ke halaman utama dengan menu-menu seperti, kelola akun, kelola data *addOn*, dan melakukan prediksi terhadap pendapatan. Sedangkan jika pegawai yang *login* hanya akan menampilkan beberapa menu saja seperti menu kelola data *addOn*, dan prediksi terhadap pendapatan.

Berikut merupakan *syntax* yang dapat dilakukan untuk membuat *form login.*

1. Auth.php

|  |
| --- |
| <?php  class Auth extends CI\_Controller{  public function login()  {  $this->form\_validation->set\_rules('username','Username','required',[  'required' => 'Username wajib diisi!'  ]);  $this->form\_validation->set\_rules('password','Password','required',[  'required' => 'Password wajib diisi!'  ]);  if ($this->form\_validation->run() == FALSE)  {    $this->load->view('login2');      }else{  $auth = $this->model\_auth->cek\_login();  if($auth == FALSE)  {  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  <strong>Username atau Password anda Salah!!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('Auth/login');  }else{  $this->session->set\_userdata('username',$auth->username);  $this->session->set\_userdata('role\_id',$auth->role\_id);  switch($auth->role\_id){  case 1 : redirect('C\_dashboard\_admin/dashboard');  break;  case 2 : redirect('C\_dashboard\_user/dashboard');  break;  default: break;  }  }    }  }  public function logout()  {  $this->session->sess\_destroy();  redirect('Auth/login');  }  }  ?> |

Listings 6. 1 Syntax Controller Untuk Auth.

*Syntax* tersebut merupakan bagian dari *controller* untuk membuat halaman *login*, berikut ini merupakan beberapa penjelasan mengenai *syntax* tersebut.

|  |
| --- |
| <?php  class Auth extends CI\_Controller{  public function login()  {  $this->form\_validation->set\_rules('username','Username','required',[  'required' => 'Username wajib diisi!'  ]);  $this->form\_validation->set\_rules('password','Password','required',[  'required' => 'Password wajib diisi!'  ]);  if ($this->form\_validation->run() == FALSE)  {  $this->load->view('login2'); |

Listings 6. 2 Syntax Untuk Validasi Akun Pada Sistem Peramalan.

*Syntax* tersebut berfungsi untuk melakukan validasi terhadap label *username* dan *password*, jika *user* memasukkan akun yang sudah terdaftar pada *database*, maka validasinya akan berhasil dan dilanjutkan pada halaman utama.

|  |
| --- |
| }else{  $auth = $this->model\_auth->cek\_login();  if($auth == FALSE)  {  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  <strong>Username atau Password anda Salah!!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('Auth/login');  }else{  $this->session->set\_userdata('username',$auth->username);  $this->session->set\_userdata('role\_id',$auth->role\_id);  switch($auth->role\_id){  case 1 : redirect('C\_dashboard\_admin/dashboard');  break;  case 2 : redirect('C\_dashboard\_user/dashboard');  break;  default: break;  }  }  }  } |

Listings 6. 3 Syntax Untuk Fungsi Validasi Dashboard.

*Syntax* tersebut akan melakukan pengecekan terhadap model, jika *user* salah memasukkan *username* atau *password*, maka sistem akan memberikan pemberitahuan bahwa *username* atau *password* yang dimasukkan salah. Namun, jika *username* dan *password* yang dimasukkan sudah sesuai, maka akan dilanjutkan pada halaman utama. Untuk *role* admin akan dilanjutkan pada halaman utama admin, dan untuk *role* pegawai akan dilanjutkan pada halaman utama pegawai.

|  |
| --- |
| public function logout()  {  $this->session->sess\_destroy();  redirect('Auth/login');  }  }  ?> |

Listing 6. 4 Syntax Untuk Fungsi Logout.

*Syntax* tersebut memiliki fungsi untuk melakukna *logout* dari sistem, jika *user* melakukan *logout*, maka akan dikembalikan pada halaman *login*, dan *session* dari *login*-nya akan di hancurkan, untuh menghindarkan dari *user* yang kembali pada halaman utama tanpa melakukan *login* terlebih dahulu.

1. Model\_auth.php

|  |
| --- |
| <?php  class Model\_auth extends CI\_Model{  public function cek\_login()  {  $username = set\_value('username');  $password = set\_value('password');  $result = $this->db->where('username',$username)  ->where('password',$password)  ->limit(1)  ->get('login');  if($result->num\_rows() > 0){  return $result->row();  }else{  return array();  }  }  ?> |

Listing 6. 5 Syntax Model Auth.

*Syntax* tersebut merupakan bagian dari model auth, yang berguna untuk menghubungkan sistem pada *database* sistem, berikut merupakan beberapa penjelasan dari *syntax* tersebut.

|  |
| --- |
| public function cek\_login()  {  $username = set\_value('username');  $password = set\_value('password');  $result = $this->db->where('username',$username)  ->where('password',$password)  ->limit(1)  ->get('login');  if($result->num\_rows() > 0){  return $result->row();  }else{  return array();  }  } |

Listing 6. 6 Syntax Fungsi Model Auth.

*Syntax* tersebut berfungsi untuk melakukan pengecekan terhadap *database* sistem, model auth ini akan melakukan pengecekan terhadap *username* dan *password* yang dimasukkan oleh *user* dengan yang terdapat pada *database*, sistem akan melakukan pencarian dari *row* pertama hingga *row* akhir untuk validasi akun. Jika validasi tidak berhasil, maka *user* tidak dapat *login* pada sistem peramalan.

1. View\_login.php

|  |
| --- |
| <?php ?>  <head>  <title> Halaman Login </title>  <link rel="stylesheet" type="text/css" href="<?php echo base\_url()?>assets/template\_login/style1.css">  </head>  <body>  <div class="login-box">  <img src="<?php echo base\_url()?>assets/template\_login/avatar.png" class="avatar">    <h1>Login Here</h1>  <?php echo $this->session->flashdata('pesan')?>  <form method="post" action="<?php echo base\_url('Auth/login') ?>" class="user">  <p>Username</p>  <input type="text" name="username" placeholder="Enter Username">  <span class="fa fa-user form-control-feedback"></span>  <?php echo form\_error('username', '<div class="text-danger small ml-2">','</div>'); ?>  <p>Password</p>  <input type="password" name="password" placeholder="Enter Password">  <span class="glyphicon glyphicon-lock form-control-feedback"></span>  <?php echo form\_error('password', '<div class="text-danger small ml-2">','</div>'); ?>  <input type="submit" name="submit" value="Login">  </form>  </div>  </body> |

Listing 6. 7 Syntax Untuk Membuat View Login.

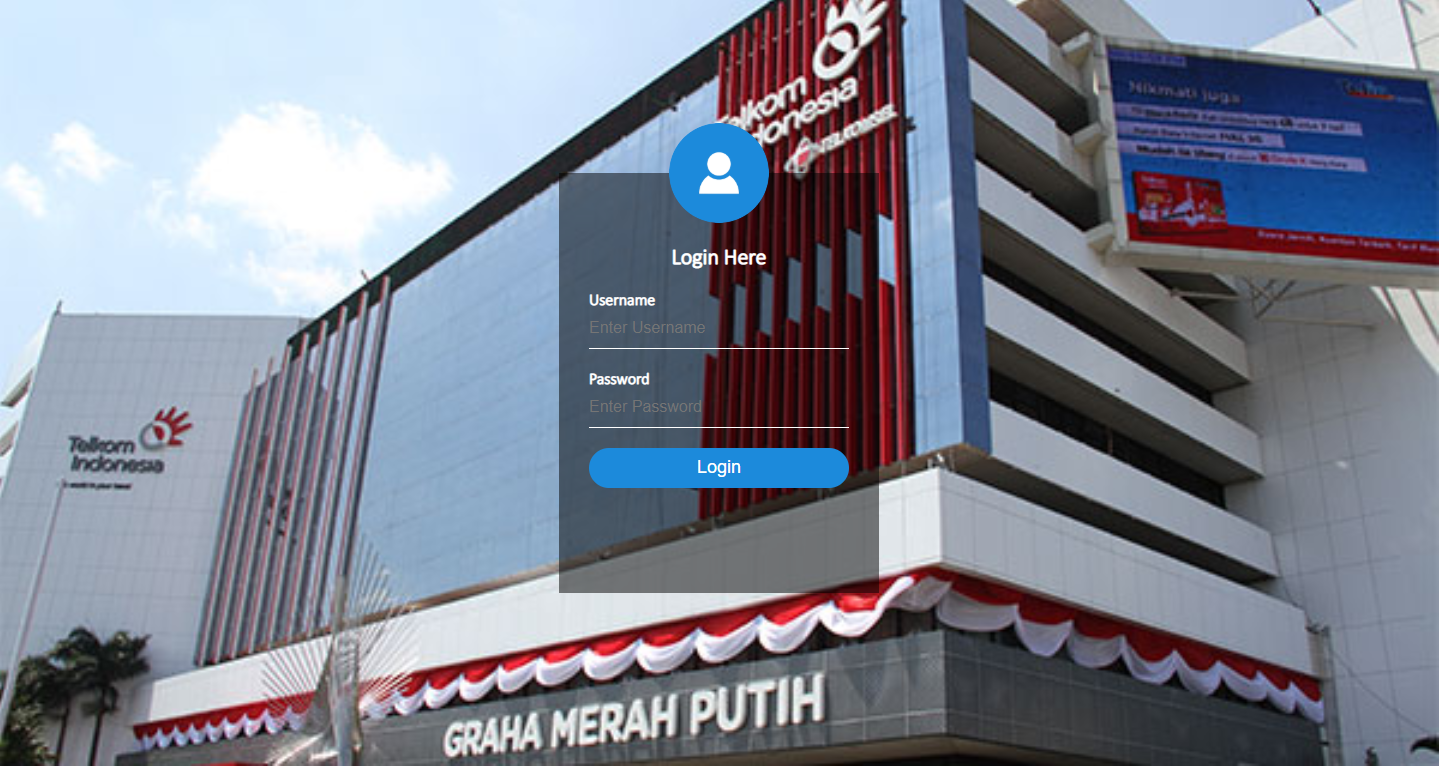
*Syntax* tersebut merupakan *view* atau tampilan pada halaman *login*, berikut ini merupakan beberapa penjelasan dari *syntax* tersebut.

|  |
| --- |
| <?php ?>  <head>  <title> Halaman Login </title>  <link rel="stylesheet" type="text/css" href="<?php echo base\_url()?>assets/template\_login/style1.css">  </head>  <body>  <div class="login-box">  <img src="<?php echo base\_url()?>assets/template\_login/avatar.png" class="avatar">  <h1>Login Here</h1>  <?php echo $this->session->flashdata('pesan')?>  <form method="post" action="<?php echo base\_url('Auth/login') ?>" class="user">  <p>Username</p>  <input type="text" name="username" placeholder="Enter Username">  <span class="fa fa-user form-control-feedback"></span>  <?php echo form\_error('username', '<div class="text-danger small ml-2">','</div>'); ?>  <p>Password</p>  <input type="password" name="password" placeholder="Enter Password">  <span class="glyphicon glyphicon-lock form-control-feedback"></span>  <?php echo form\_error('password', '<div class="text-danger small ml-2">','</div>'); ?>  <input type="submit" name="submit" value="Login">  </form>  </div>  </body> |

Listing 6. 8 Beberapa Fungsi Syntax Untuk Membuat Halaman Login.

Pada *syntax* tersebut terdapat fungsi untuk membuat judul, selain judul terdapat pula untuk menambahkan *background* gambar pada halaman *login*, untuk gambar perlu disimpan gambarnya pada direktori *assets*. Selanjutnya *syntax* untuk membuat label *username* dan *password*, label tersebut diberikan pemberitahuan tambahan jika *username* atau *password* yang dimasukkan tidak sesuai. Terakhir terdapat *button* yang berguna untuk melakukan *submit* *username* dan *password*.

Sehingga hasil dari *syntax* tersebut, sebagai berikut.



Gambar 6. 1 Hasil dari syntax Auth.php.

1. Perancangan *Form Dashboard*
2. C\_dashboard\_admin.php

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class C\_dashboard\_admin extends CI\_Controller {  public function \_\_construct(){  parent::\_\_construct();  $this->load->model('m\_dashboard\_admin');  if($this->session->userdata('role\_id')!='1'){  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  <strong>Anda belum Login !!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('auth/login');  }  }  public function dashboard()  {  $data['indieland'] = $this->m\_dashboard\_admin->indieland();  $data['movieyuk'] = $this->m\_dashboard\_admin->movieyuk();  $data['indieland\_pantai'] = $this->m\_dashboard\_admin->indieland\_pantai();  $data['otm'] = $this->m\_dashboard\_admin->otm();  $data['indieland\_jazz'] = $this->m\_dashboard\_admin->indieland\_jazz();  $data['video'] = $this->m\_dashboard\_admin->video();  $data['indieland\_pack'] = $this->m\_dashboard\_admin->indieland\_pack();  $data['indieland\_lokal'] = $this->m\_dashboard\_admin->indieland\_lokal();  $data['psy'] = $this->m\_dashboard\_admin->psy();  $data['wifi\_panjang'] = $this->m\_dashboard\_admin->wifi\_panjang();  $data['indieland\_pintar'] = $this->m\_dashboard\_admin->indieland\_pintar();  $data['indieland\_belajar'] = $this->m\_dashboard\_admin->indieland\_belajar();  $this->load->view('templates/header');  $this->load->view('templates/sidebar');  $this->load->view('v\_dashboard\_admin', $data);  $this->load->view('templates/footer');  }  } |

Listings 6. 9 Syntax Controller Untuk Dashboard Admin.

*Syntax* tersebut merupakan bagian dari *controller* untuk membuat halaman utama untuk admin, berikut merupakan beberapa penjelasan mengenai *syntax* tersebut.

|  |
| --- |
| public function \_\_construct(){  parent::\_\_construct();  $this->load->model('m\_dashboard\_admin');  if($this->session->userdata('role\_id')!='1'){  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  <strong>Anda belum Login !!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('auth/login');  }  } |

Listings 6. 10 Syntax Untuk Validasi Akun.

Pada *syntax* tersebut menjelaskan bahwa *controller* itu akan melakukan *load* model dengan nama *file* m\_dashboard\_admin, *syntax* dibawahnya menjelaskan jika ­*role\_id* yang masuk pada *dashboard* utama itu admin dan datanya sama dengan ada yang pada *database* maka akan langsung dilanjutkan pada halaman utama, namun jika tidak sesuai dengan *database* maka akan muncul *alert* dan tetap di halaman *login*.

1. M\_dashboard\_admin.php

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class M\_dashboard\_admin extends CI\_Model {  public function \_\_construct()  {  parent::\_\_construct();  }  Public function indieland() {  $wordsquery = "SELECT \* FROM indieland";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function movieyuk() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function indieland\_pantai() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from indielanad\_pantai WHERE indieland.NO\_INET = indieland\_pantai.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function otm() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from otm WHERE indieland.NO\_INET = otm.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function indieland\_jazz() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from indieland\_jazz WHERE indieland.NO\_INET = indieland\_jazz.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function video() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from video WHERE indieland.NO\_INET = video.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function indieland\_pack() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from indieland\_pack WHERE indieland.NO\_INET = indieland\_pack.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function indieland\_lokal() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from indieland\_lokal WHERE indieland.NO\_INET = indieland\_lokal.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function psy() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from psy WHERE indieland.NO\_INET = psy.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function wifi\_panjang() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from wifi\_panjang WHERE indieland.NO\_INET = wifi\_panjang.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function indieland\_pintar() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from indieland\_pintar WHERE indieland.NO\_INET = indieland\_pintar.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function indieland\_belajar() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from indieland\_belajar WHERE indieland.NO\_INET = indieland\_belajar.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  } |

Listings 6. 11 Syntax Model Dashboard Admin.

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class M\_dashboard\_admin extends CI\_Model {  public function \_\_construct()  {  parent::\_\_construct();  }  Public function indieland() {  $wordsquery = "SELECT \* FROM indieland";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function movieyuk() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  } |

Listings 6. 12 Syntax Untuk Menghubungkan Dashboard Pada Database.

*Syntax* pada gambar *6.12* merupakan model untuk menghubungkan *syntax* yang telah dibuat dengan *database* yang telah dibuat sebelumnya.

|  |
| --- |
| Public function indieland() {  $wordsquery = "SELECT \* FROM indieland";  $query = $this->db->query($wordsquery);  return $query->result();  } |

Listings 6. 13 Syntax Untuk Function Indieland.

Pada *syntax* tersebut menjelaskan bahwa *function* *indieland* akan mengambil data pada *database* sistem dengan nama tabel *indieland*, *query* tersebut akan menampilkan semua data yang terdapat pada tabel *indieland*.

|  |
| --- |
| Public function movieyuk() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function indieland\_pantai() {  $wordsquery = "SELECT \* FROM indieland WHERE NOT EXISTS (SELECT \* from indieland\_pantai WHERE indieland.NO\_INET = indieland\_pantai.NO\_INET)";  $query = $this->db->query($wordsquery);  return $query->result();  } |

Listings 6. 14 Syntax Function Movieyuk dan Indieland\_pantai.

*Syntax* tersebut berfugsi untuk menampilkan data *indieland* yang sudah berlangganan movieyuk, pada *query* tersebut terdapat fungsi untuk mencari no\_internet pada dua tabel, yakni tabel indieland dan tabel movieyuk. Jika pada kedua tabel terdapat nomor internet yang sama berarti pelanggan tersebut telah berlangganan *addOn* movieyuk. Begitupun sebaliknya jika tidak terdapat kesamaan nomot internet, maka pelanggan belum berlangganan *addOn* movieyuk.

1. V\_dashboard\_admin.php

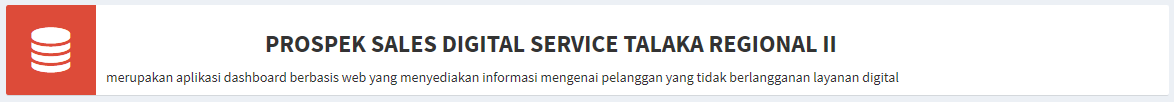
|  |
| --- |
| <div class="content-wrapper">  <!-- Content Header (Page header) -->  <section class="content-header">  <div class="row">  <div class="col-md-12 col-sm-10 col-xs-12">  <div class="info-box">  <span class="info-box-icon bg-red"><i class="fa fa-database"></i></span>  <div class="info-box-content">  <marquee direction="left"><h3><b>PROSPEK SALES DIGITAL SERVICE TALAKA REGIONAL II</b></h3></marquee>  <h7>merupakan aplikasi dashboard berbasis web yang menyediakan informasi mengenai pelanggan yang tidak berlangganan layanan digital</h7>  </div>  <!-- /.info-box-content -->  </div>  <!-- /.info-box -->  </div>    </section>  <!-- Main content -->  <section class="content">  <!-- Small boxes (Stat box) -->  <div class="row">  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-maroon">  <div class="inner">  <h3>  <?php  $indieland = count($indieland);  echo $indieland;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>INDIELAND</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Indieland</a>  <a href="<?php echo base\_url()?>c\_indieland" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-green">  <div class="inner">  <h3>  <?php  $movieyuk = count($movieyuk);  echo $movieyuk;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>MOVIEYUK</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek Movieyuk</a>  <a href="<?php echo base\_url()?>c\_movieyuk" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-aqua">  <div class="inner">  <h3>  <?php  $indieland\_pantai = count($indieland\_pantai);  echo $indieland\_pantai;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>INDIELAND PANTAI</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data IL Pantai</a>  <a href="<?php echo base\_url()?>c\_indieland\_pantai" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-yellow">  <div class="inner">  <h3>  <?php  $otm = count($otm);  echo $otm;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>OTM</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek OTM</a>  <a href="<?php echo base\_url()?>c\_otm" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-maroon">  <div class="inner">  <h3>  <?php  $indieland\_jazz = count($indieland\_jazz);  echo $indieland\_jazz;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>INDIELAND JAZZ</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek IL JAZZ</a>  <a href="<?php echo base\_url()?>c\_indieland\_jazz" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-green">  <div class="inner">  <h3>  <?php  $video = count($video);  echo $video;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>VIDEO</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek Video</a>  <a href="<?php echo base\_url()?>c\_video" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-aqua">  <div class="inner">  <h3>  <?php  $indieland\_pack = count($indieland\_pack);  echo $indieland\_pack;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>INDIELAND PACK</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek Indieland Pack</a>  <a href="<?php echo base\_url()?>c\_indieland\_pack" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-yellow">  <div class="inner">  <h3>  <?php  $indieland\_lokal = count($indieland\_lokal);  echo $indieland\_lokal;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>INDIELAND LOKAL</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek Indieland Lokal</a>  <a href="<?php echo base\_url()?>c\_indieland\_lokal" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-maroon">  <div class="inner">  <h3>  <?php  $psy = count($psy);  echo $psy;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>PSY</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek PSY</a>  <a href="<?php echo base\_url()?>c\_plc" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-green">  <div class="inner">  <h3>  <?php  $wifi\_panjang = count($wifi\_panjang);  echo $wifi\_panjang;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>WIFI PANJANG</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek Wifi Panjang</a>  <a href="<?php echo base\_url()?>c\_wifi\_panjang" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-aqua">  <div class="inner">  <h3>  <?php  $indieland\_pintar = count($indieland\_pintar);  echo $indieland\_pintar;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>INDIELAND PINTAR</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek IH PINTAR</a>  <a href="<?php echo base\_url()?>c\_indieland\_pintar" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-yellow">  <div class="inner">  <h3>  <?php  $indieland\_belajar = count($indieland\_belajar);  echo $indieland\_belajar;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>INDIELAND BELAJAR</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Prospek IL Belajar</a>  <a href="<?php echo base\_url()?>c\_indieland" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div>  <!-- ./col -->  </div>  <!-- /.row -->  <!-- Main row -->    <!-- /.row (main row) -->  </section>  <!-- /.content -->  </div>  <!-- /.content-wrapper -->  </body>  <!-- modal -->  <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog" role="document">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title" id="exampleModalLabel">FORM UPLOAD DATA </h4>  <button type="button" class="close" data-dismiss="modal" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <?= form\_open\_multipart('c\_indihome/import'); ?>  <form method="post" action="<?= base\_url().'c\_indihome/import'; ?>">  <div class="form-group">  <label for="inputEmail3" class="control-label">Upload Data Indihome</label>  <input type="file" name="file" class="form-control" id="file" required accept=".xls, .xlsx, .csv" /></p>  </div>  <?= form\_open\_multipart('c\_indihome/import'); ?>  <form method="post" action="<?= base\_url().'c\_indihome/import'; ?>">  <div class="form-group">  <label for="inputEmail3" class="control-label">Upload Data Movin</label>  <input type="file" name="file" class="form-control" id="file" required accept=".xls, .xlsx, .csv" /></p>  </div>  <?= form\_open\_multipart('c\_indieland/import'); ?>  <form method="post" action="<?= base\_url().'c\_indieland/import'; ?>">  <div class="form-group">  <label for="inputEmail3" class="control-label">Upload Data Indieland Pantai</label>  <input type="file" name="file" class="form-control" id="file" required accept=".xls, .xlsx, .csv" /></p>  <p class="help-block">Format file yang didukung .xls, .xlsx atau .csv</p>  </div>  <div class="form-group">  <input type="submit" class="btn btn-primary" value="Upload" name="upload" id="upload">  </div>  <?= form\_close(); ?>  </div>  </div>  </div>  </div>  </html> |

Listings 6. 15 Syntax View Pada Dashboard Admin.

|  |
| --- |
| <div class="content-wrapper">  <!-- Content Header (Page header) -->  <section class="content-header">  <div class="row">  <div class="col-md-12 col-sm-10 col-xs-12">  <div class="info-box">  <span class="info-box-icon bg-red"><i class="fa fa-database"></i></span>  <div class="info-box-content">  <marquee direction="left"><h3><b>PROSPEK SALES DIGITAL SERVICE TALAKAN REGIONAL II</b></h3></marquee>  <h7>merupakan aplikasi dashboard berbasis web yang menyediakan informasi mengenai pelanggan yang tidak berlangganan layanan digital</h7>  </div>  <!-- /.info-box-content -->  </div>  <!-- /.info-box -->  </div>  </section> |

Listings 6. 16 Syntax Untuk Membuat Header Pada Dashboard Admin.

*Syntax* tersebut berfungsi untuk membuat judul pada halaman utama admin, selain *syntax* untuk membuat judul terdapat juga *syntax* untuk membuat logo dan dapat kita rubah ukurannya sesuai dengan yang diiniginkan. Jika berhasil, maka hasilnya akan seperti berikut.

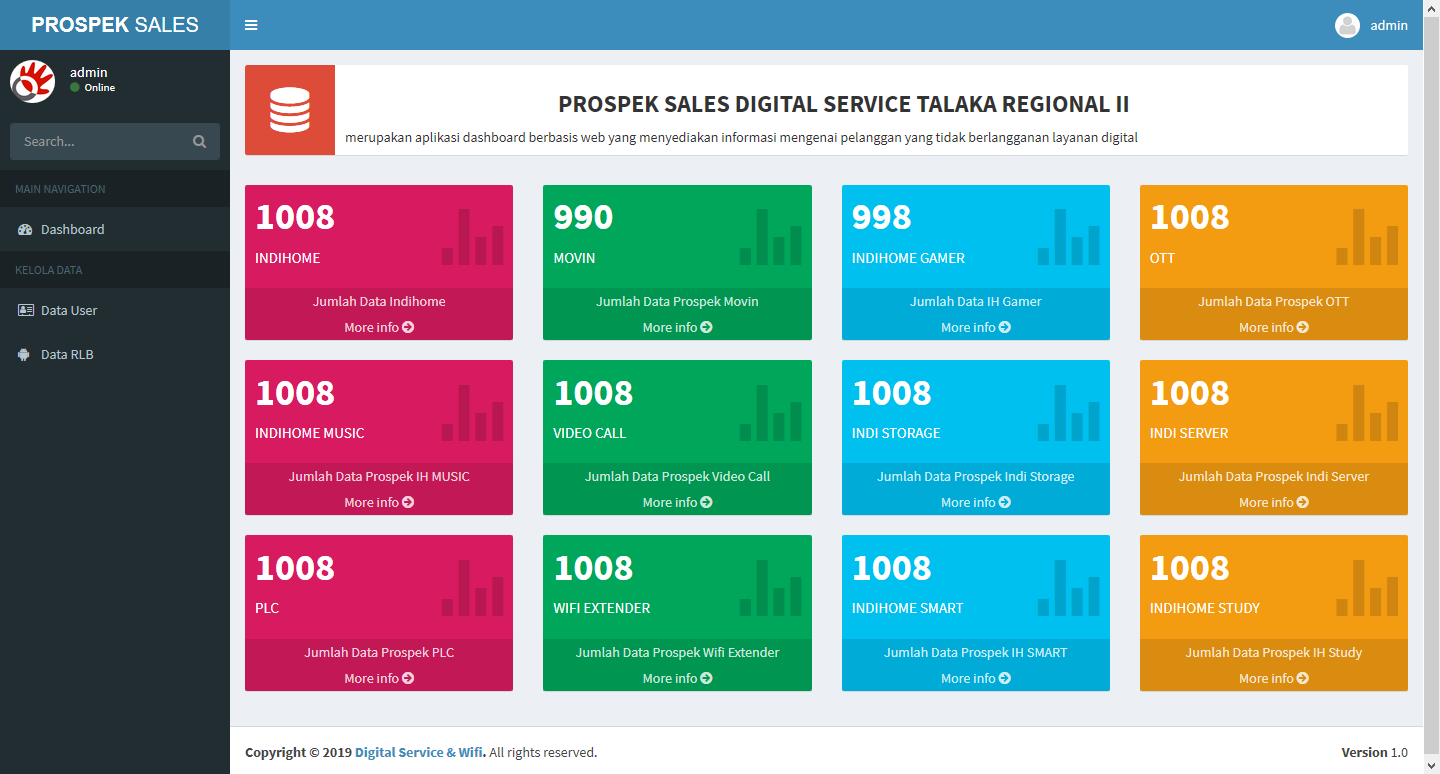


Gambar 6. 2 Hasil Dari Syntax Untuk Membuat Judul.

|  |
| --- |
| <section class="content">  <!-- Small boxes (Stat box) -->  <div class="row">  <div class="col-lg-3 col-xs-6">  <!-- small box -->  <div class="small-box bg-maroon">  <div class="inner">  <h3>  <?php  $indieland = count($indieland);  echo $indieland;  ?><sup style="font-size: 20px"></sup>  </h3>  <p>INDIELAND</p>  </div>  <div class="icon">  <i class="ion ion-stats-bars"></i>  </div>  <a class="small-box-footer">Jumlah Data Indieland</a>  <a href="<?php echo base\_url()?>c\_indieland" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>  </div>  </div> |

Listings 6. 17 Syntax Untuk Membuat List Box.

*Syntax* tersebut berfungsi untuk membuat *list box*, pada *list box* tersebut terdapat informasi mengenai jumlah pelanggan untuk setiap paket *addOn*-nya, selain itu terdapt juga *button* “*more info*” untuk melihat lebih jelas pada halaman paket *addOn*. Jika telah berhasil dilakukan, maka hasilnya akan tampak seperti berikut.



Gambar 6. 3 Hasil Dari Syntax View Dashboard Admin.

1. Perancangan *Form* Indieland, dan Movieyuk

Pada *form indieland* ini menampilkan data pelanggan yang sudah berlangganan paket utama Indieland, pada *form Indieland* ini juga terdapat *button* untuk mengunggah file pelanggan pada sistem peramalan ini. Berikut merupakan *syntax* yang digunakan untuk membuat *form indieland*.

1. C\_indieland.php

|  |
| --- |
| <?php  class C\_indieland extends CI\_Controller{    public function \_\_construct(){  parent::\_\_construct();  $this->load->model('m\_indieland);  $this->load->library('Excel');  // if($this->session->userdata('role\_id')!='1'){  // $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  // <strong>Anda belum Login !!!</strong>  // <button type="button" class="close" data-dismiss="alert" aria-label="Close">  // <span aria-hidden="true">&times;</span>  // </button>  // </div>');  // redirect('auth/login');  // }  }  public function index(){  $data['indieland1'] = $this->m\_indieland->indieland1();  $data['witel'] = $this->m\_indieland->witel();  $data['datel'] = $this->m\_indieland->datel();  if($valid->run()==FALSE)  {  $this->load->view('templates/header');  if ($this->session->userdata('role\_id') ==='1') {  $this->load->view('templates/sidebar');  }elseif($this->session->userdata('role\_id') ==='2'){  $this->load->view('templates/sidebar\_user');  }else{  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  <strong>Anda belum Login !!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('auth/login');  }  $this->load->view('indieland/v\_indieland',$data, FALSE);  $this->load->view('templates/footer');  }  $this->m\_indieland->inputpelanggan($data);  $this->session->set\_flashdata('sukses', 'Data Pelanggan Berhasil diTambahkan');  redirect(base\_url('c\_indieland'),'refresh');  }  }    public function getInet()  {  $id = $\_GET['c'];  $GetNoInet = $this->m\_indieland->getInet($id);  echo json\_encode($GetNoInet);  }  public function view\_list($id){  $explode = explode('~', $id);  $witel = $explode[0];  $datel = $explode[1];    $data['indieland'] = $this->m\_indieland->indieland($witel, $datel);  $this->load->view('indieland/view\_list', $data);  }  // public function import(){  // if(isset($\_FILES["file"]["name"]))  // {  // $path = $\_FILES["file"]["tmp\_name"];  // $object = PHPExcel\_IOFactory::load($path);  // foreach($object->getWorksheetIterator() as $worksheet)  // {  // $highestRow = $worksheet->getHighestRow();  // $highestColumn = $worksheet->getHighestColumn();  // for($row=2; $row<=$highestRow; $row++)  // {  // $kawasan = $worksheet->getCellByColumnAndRow(0, $row)->getValue();  // $witel = $worksheet->getCellByColumnAndRow(1, $row)->getValue();  // $datel = $worksheet->getCellByColumnAndRow(2, $row)->getValue();  // $sto = $worksheet->getCellByColumnAndRow(3, $row)->getValue();  // $ncli = $worksheet->getCellByColumnAndRow(4, $row)->getValue();  // $ndos = $worksheet->getCellByColumnAndRow(5, $row)->getValue();  // $ndem = $worksheet->getCellByColumnAndRow(6, $row)->getValue();  // $no\_inet = $worksheet->getCellByColumnAndRow(7, $row)->getValue();  // $nd = $worksheet->getCellByColumnAndRow(8, $row)->getValue();  // $chanel = $worksheet->getCellByColumnAndRow(9, $row)->getValue();  // $citem\_speedy = $worksheet->getCellByColumnAndRow(10, $row)->getValue();  // $kecepatan = $worksheet->getCellByColumnAndRow(11, $row)->getValue();  // $deskripsi = $worksheet->getCellByColumnAndRow(12, $row)->getValue();  // $tgl\_reg = $worksheet->getCellByColumnAndRow(13, $row)->getValue();  // $tgl\_etat = $worksheet->getCellByColumnAndRow(14, $row)->getValue();  // $status = $worksheet->getCellByColumnAndRow(15, $row)->getValue();  // $nama = $worksheet->getCellByColumnAndRow(16, $row)->getValue();  // $kcontact = $worksheet->getCellByColumnAndRow(17, $row)->getValue();  // $status\_order = $worksheet->getCellByColumnAndRow(18, $row)->getValue();  // $alpro = $worksheet->getCellByColumnAndRow(19, $row)->getValue();  // $ccat = $worksheet->getCellByColumnAndRow(20, $row)->getValue();  // $jalan = $worksheet->getCellByColumnAndRow(21, $row)->getValue();  // $nojalan = $worksheet->getCellByColumnAndRow(22, $row)->getValue();  // $distrik = $worksheet->getCellByColumnAndRow(23, $row)->getValue();  // $kota = $worksheet->getCellByColumnAndRow(24, $row)->getValue();  // $cpack = $worksheet->getCellByColumnAndRow(25, $row)->getValue();  // $data[] = array(  // 'kawasan' => $kawasan,  // 'witel' => $witel,  // 'datel' => $datel,  // 'sto' => $sto,  // 'ncli' => $ncli,  // 'ndos' => $ndos,  // 'ndem' => $ndem,  // 'no\_inet' => $no\_inet,  // 'nd' => $nd,  // 'chanel' => $chanel,  // 'citem\_speedy' => $citem\_speedy,  // 'kecepatan' => $kecepatan,  // 'deskripsi' => $deskripsi,  // 'tgl\_reg' => $tgl\_reg,  // 'tgl\_etat' => $tgl\_etat,  // 'status' => $status,  // 'nama' => $nama,  // 'kcontact' => $kcontact,  // 'status\_order' => $status\_order,  // 'alpro' => $alpro,  // 'ccat' => $ccat,  // 'jalan' => $jalan,  // 'nojalan' => $nojalan,  // 'distrik' => $distrik,  // 'kota' => $kota,  // 'cpack' => $cpack  // );  // }  // }  // $this->m\_indieland->upload($data);  // redirect('c\_dashboard\_admin/dashboard');  // }  // }  public function import(){  if(isset($\_FILES["file"]["name"]))  {  $path = $\_FILES["file"]["tmp\_name"];  $object = PHPExcel\_IOFactory::load($path);  foreach($object->getWorksheetIterator() as $worksheet)  {  $highestRow = $worksheet->getHighestRow();  $highestColumn = $worksheet->getHighestColumn();  for($row=2; $row<=$highestRow; $row++)  {  $kawasan = $worksheet->getCellByColumnAndRow(0, $row)->getValue();  $witel = $worksheet->getCellByColumnAndRow(1, $row)->getValue();  $datel = $worksheet->getCellByColumnAndRow(2, $row)->getValue();  $sto = $worksheet->getCellByColumnAndRow(3, $row)->getValue();  $ncli = $worksheet->getCellByColumnAndRow(4, $row)->getValue();  $ndos = $worksheet->getCellByColumnAndRow(5, $row)->getValue();  $ndem = $worksheet->getCellByColumnAndRow(6, $row)->getValue();  $no\_inet = $worksheet->getCellByColumnAndRow(7, $row)->getValue();  $nd = $worksheet->getCellByColumnAndRow(8, $row)->getValue();  $chanel = $worksheet->getCellByColumnAndRow(9, $row)->getValue();  $citem\_speedy = $worksheet->getCellByColumnAndRow(10, $row)->getValue();  $kecepatan = $worksheet->getCellByColumnAndRow(11, $row)->getValue();  $deskripsi = $worksheet->getCellByColumnAndRow(12, $row)->getValue();  $tgl\_reg = $worksheet->getCellByColumnAndRow(13, $row)->getValue();  $tgl\_etat = $worksheet->getCellByColumnAndRow(14, $row)->getValue();  $status = $worksheet->getCellByColumnAndRow(15, $row)->getValue();  $nama = $worksheet->getCellByColumnAndRow(16, $row)->getValue();  $kcontact = $worksheet->getCellByColumnAndRow(17, $row)->getValue();  $status\_order = $worksheet->getCellByColumnAndRow(18, $row)->getValue();  $alpro = $worksheet->getCellByColumnAndRow(19, $row)->getValue();  $ccat = $worksheet->getCellByColumnAndRow(20, $row)->getValue();  $jalan = $worksheet->getCellByColumnAndRow(21, $row)->getValue();  $nojalan = $worksheet->getCellByColumnAndRow(22, $row)->getValue();  $distrik = $worksheet->getCellByColumnAndRow(23, $row)->getValue();  $kota = $worksheet->getCellByColumnAndRow(24, $row)->getValue();  $cpack = $worksheet->getCellByColumnAndRow(25, $row)->getValue();    $DateNewTglreg = new DateTime($tgl\_reg);  $DateChangeTglreg = $DateNewTglreg->format('Y-m-d');    $DateNewTglEtat = new DateTime($tgl\_etat);  $DateChangeTglEtat = $DateNewTglEtat->format('Y-m-d H:i:s');    $cek\_duplicat = $this->m\_indieland->chek\_duplicat($ncli);  if ($cek\_duplicat != NULL) {  if ($cek\_duplicat->ncli == $ncli) {  $this->m\_indieland->update\_duplicat($kawasan, $witel, $datel, $sto, $ncli, $ndos, $ndem, $no\_inet, $nd, $chanel, $citem\_speedy, $kecepatan, $deskripsi, $DateChangeTglreg, $DateChangeTglEtat, $status, $nama, $kcontact, $status\_order, $alpro, $ccat, $jalan, $nojalan, $distrik, $kota, $cpack);  }else{  $this->m\_indieland->upload($kawasan, $witel, $datel, $sto, $ncli, $ndos, $ndem, $no\_inet, $nd, $chanel, $citem\_speedy, $kecepatan, $deskripsi, $DateChangeTglreg, $DateChangeTglEtat, $status, $nama, $kcontact, $status\_order, $alpro, $ccat, $jalan, $nojalan, $distrik, $kota, $cpack);  }  }else{  $this->m\_indieland->upload($kawasan, $witel, $datel, $sto, $ncli, $ndos, $ndem, $no\_inet, $nd, $chanel, $citem\_speedy, $kecepatan, $deskripsi, $DateChangeTglreg, $DateChangeTglEtat, $status, $nama, $kcontact, $status\_order, $alpro, $ccat, $jalan, $nojalan, $distrik, $kota, $cpack);  }    }  }  redirect('c\_dashboard\_admin/dashboard');  }  }  } |

Listings 6. 18 Syntax Controller Indieland.

|  |
| --- |
| <?php  class C\_indihome extends CI\_Controller{  public function \_\_construct(){  parent::\_\_construct();  $this->load->model('m\_indieland');  $this->load->library('Excel'); |

Listings 6. 19 Syntax Untuk Meload Library.

Pada *syntax* tersebut menggunakan sebuah model yang berfungsi untuk menghubungkan sistem dengan *database* yang telah kita buat sebelumnya. Selain itu, terdapat juga *library* excel yang berfungsi untuk melakukan *upload* file ber-ekstensi *xlsx,* dan *xls.*

|  |
| --- |
| public function getInet()  {  $id = $\_GET['c'];  $GetNoInet = $this->m\_indieland->getInet($id);  echo json\_encode($GetNoInet);  } |

Listings 6. 20 Function Untuk Get Data Pada Database.

*Syntax* tersebut berfungsi untuk mencari nomor internet yang terdapat pada *database* untuk ditampilkan pada menu *form indieland*.

|  |
| --- |
| public function import(){  if(isset($\_FILES["file"]["name"]))  {  $path = $\_FILES["file"]["tmp\_name"];  $object = PHPExcel\_IOFactory::load($path);  foreach($object->getWorksheetIterator() as $worksheet)  {  $highestRow = $worksheet->getHighestRow();  $highestColumn = $worksheet->getHighestColumn();  for($row=2; $row<=$highestRow; $row++)  {  $kawasan = $worksheet->getCellByColumnAndRow(0, $row)->getValue();  $witel = $worksheet->getCellByColumnAndRow(1, $row)->getValue();  $datel = $worksheet->getCellByColumnAndRow(2, $row)->getValue();  $sto = $worksheet->getCellByColumnAndRow(3, $row)->getValue();  $ncli = $worksheet->getCellByColumnAndRow(4, $row)->getValue();  $ndos = $worksheet->getCellByColumnAndRow(5, $row)->getValue();  $ndem = $worksheet->getCellByColumnAndRow(6, $row)->getValue();  $no\_inet = $worksheet->getCellByColumnAndRow(7, $row)->getValue();  $nd = $worksheet->getCellByColumnAndRow(8, $row)->getValue();  $chanel = $worksheet->getCellByColumnAndRow(9, $row)->getValue();  $citem\_speedy = $worksheet->getCellByColumnAndRow(10, $row)->getValue();  $kecepatan = $worksheet->getCellByColumnAndRow(11, $row)->getValue();  $deskripsi = $worksheet->getCellByColumnAndRow(12, $row)->getValue();  $tgl\_reg = $worksheet->getCellByColumnAndRow(13, $row)->getValue();  $tgl\_etat = $worksheet->getCellByColumnAndRow(14, $row)->getValue();  $status = $worksheet->getCellByColumnAndRow(15, $row)->getValue();  $nama = $worksheet->getCellByColumnAndRow(16, $row)->getValue();  $kcontact = $worksheet->getCellByColumnAndRow(17, $row)->getValue();  $status\_order = $worksheet->getCellByColumnAndRow(18, $row)->getValue();  $alpro = $worksheet->getCellByColumnAndRow(19, $row)->getValue();  $ccat = $worksheet->getCellByColumnAndRow(20, $row)->getValue();  $jalan = $worksheet->getCellByColumnAndRow(21, $row)->getValue();  $nojalan = $worksheet->getCellByColumnAndRow(22, $row)->getValue();  $distrik = $worksheet->getCellByColumnAndRow(23, $row)->getValue();  $kota = $worksheet->getCellByColumnAndRow(24, $row)->getValue();  $cpack = $worksheet->getCellByColumnAndRow(25, $row)->getValue();    $DateNewTglreg = new DateTime($tgl\_reg);  $DateChangeTglreg = $DateNewTglreg->format('Y-m-d');    $DateNewTglEtat = new DateTime($tgl\_etat);  $DateChangeTglEtat = $DateNewTglEtat->format('Y-m-d H:i:s'); |

Listings 6. 21 Syntax Untuk Import Data.

*Syntax* tersebut berfungsi untuk melakukan *import* data yang memiliki ekstensi *xlsx,* dan *xls*, *syntax* tersebut akan melakukan validasi terhadap *database* dengan file yang akan di *upload*. Jika kolom pada file dan *database* tidak sesuai, maka file tidak dapat di *upload* pada sistem.

|  |
| --- |
| $cek\_duplicat = $this->m\_indieland->chek\_duplicat($ncli);  if ($cek\_duplicat != NULL) {  if ($cek\_duplicat->ncli == $ncli) {  $this->m\_indieland->update\_duplicat($kawasan, $witel, $datel, $sto, $ncli, $ndos, $ndem, $no\_inet, $nd, $chanel, $citem\_speedy, $kecepatan, $deskripsi, $DateChangeTglreg, $DateChangeTglEtat, $status, $nama, $kcontact, $status\_order, $alpro, $ccat, $jalan, $nojalan, $distrik, $kota, $cpack);  }else{  $this->m\_indieland->upload($kawasan, $witel, $datel, $sto, $ncli, $ndos, $ndem, $no\_inet, $nd, $chanel, $citem\_speedy, $kecepatan, $deskripsi, $DateChangeTglreg, $DateChangeTglEtat, $status, $nama, $kcontact, $status\_order, $alpro, $ccat, $jalan, $nojalan, $distrik, $kota, $cpack);  }  }else{  $this->m\_indieland->upload($kawasan, $witel, $datel, $sto, $ncli, $ndos, $ndem, $no\_inet, $nd, $chanel, $citem\_speedy, $kecepatan, $deskripsi, $DateChangeTglreg, $DateChangeTglEtat, $status, $nama, $kcontact, $status\_order, $alpro, $ccat, $jalan, $nojalan, $distrik, $kota, $cpack);  }      }  }  redirect('c\_dashboard\_admin/dashboard');  } |

Listings 6. 22 Syntax Untuk Cek Duplikasi.

*Syntax* tersebut berfungsi untuk melakukan cek duplikasi data pada file yang akan di *upload* pada *database*, agar tidak terjadi redudansi data atau pengulangan data yang sama. Jika terdapat data yang sama, maka secara otomatis sistem tidak akan memasukkan data tersebut pada *database*. Jika telah berhasil melakukan *upload* data, maka akan dilanjutkan pada halaman *dashboard* utama.

1. M\_indieland.php

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class M\_indieland extends CI\_Model {  public function \_\_construct()  {  parent::\_\_construct();  }  public function getInet($id)  {  $this->db->select('\*');  $this->db->from('indieland');  $this->db->where('NO\_INET', $id);  $query=$this->db->get();  return $query->row();  }  Public function indieland1() {  // $wordsquery = "SELECT \* FROM indieland where WITEL IN ('$witel') and DATEL IN ('$datel') and TGL\_REG BETWEEN '$periode\_awal' AND '$periode\_akhir'";  // $wordsquery = "SELECT \* FROM indieland where WITEL IN ('$witel') and DATEL IN ('$datel')";  $wordsquery = "SELECT \* FROM indieland ";  $query = $this->db->query($wordsquery);  return $query->result();  }  Public function indieland($witel, $datel) {  // $wordsquery = "SELECT \* FROM indieland where WITEL IN ('$witel') and DATEL IN ('$datel') and TGL\_REG BETWEEN '$periode\_awal' AND '$periode\_akhir'";  // $wordsquery = "SELECT \* FROM indieland where WITEL IN ('$witel') and DATEL IN ('$datel')";  $wordsquery = "SELECT \* FROM indieland where WITEL IN ('$witel')";  $query = $this->db->query($wordsquery);  return $query->result();  }  public function witel(){  $wordsquery = "SELECT \* FROM tb\_witel";  $query = $this->db->query($wordsquery);  return $query->result();  }  public function datel(){  $wordsquery = "SELECT \* FROM tb\_datel";  $query = $this->db->query($wordsquery);  return $query->result();  }  // function select(){  // $this->db->order\_by('id', 'DESC');  // $query = $this->db->get('import');  // return $query;  // }  public function chek\_duplicat($id){  $QuerySql = $this->db->query("SELECT ncli FROM indieland WHERE ncli IN ('$id')");  return $QuerySql->row();  }  public function update\_duplicat($kawasan, $witel, $datel, $sto, $ncli, $ndos, $ndem, $no\_inet, $nd, $chanel, $citem\_speedy, $kecepatan, $deskripsi, $tgl\_reg, $tgl\_etat, $status, $nama, $kcontact, $status\_order, $alpro, $ccat, $jalan, $nojalan, $distrik, $kota, $cpack) {  $this->db->query('UPDATE indieland SET kawasan = "'.$kawasan.'",  witel = "'.$witel.'",  datel = "'.$datel.'",  sto = "'.$sto.'",  ncli = "'.$ncli.'",  ndos = "'.$ndos.'",  ndem = "'.$ndem.'",  no\_inet = "'.$no\_inet.'",  nd = "'.$nd.'",  chanel = "'.$chanel.'",  citem\_speedy = "'.$citem\_speedy.'",  kecepatan = "'.$kecepatan.'",  deskripsi = "'.$deskripsi.'",  tgl\_reg = "'.$tgl\_reg.'",  tgl\_etat = "'.$tgl\_etat.'",  status = "'.$status.'",  nama = "'.$nama.'",  kcontact = "'.$kcontact.'",  status\_order = "'.$status\_order.'",  alpro = "'.$alpro.'",  ccat = "'.$ccat.'",  jalan = "'.$jalan.'",  nojalan = "'.$nojalan.'",  distrik = "'.$distrik.'",  kota = "'.$kota.'",  cpack = "'.$cpack.'" WHERE ncli = "'.$ncli.'"');  }  public function upload($kawasan, $witel, $datel, $sto, $ncli, $ndos, $ndem, $no\_inet, $nd, $chanel, $citem\_speedy, $kecepatan, $deskripsi, $tgl\_reg, $tgl\_etat, $status, $nama, $kcontact, $status\_order, $alpro, $ccat, $jalan, $nojalan, $distrik, $kota, $cpack)  {  // $this->db->insert\_batch('indieland', $data);  $this->db->query('REPLACE INTO indihome (kawasan, witel, datel, sto, ncli, ndos, ndem, no\_inet, nd, chanel, citem\_speedy, kecepatan, deskripsi, tgl\_reg, tgl\_etat, status, nama, kcontact, status\_order, alpro, ccat, jalan, nojalan, distrik, kota, cpack)  VALUES("'.$kawasan.'",  "'.$witel.'",  "'.$datel.'",  "'.$sto.'",  "'.$ncli.'",  "'.$ndos.'",  "'.$ndem.'",  "'.$no\_inet.'",  "'.$nd.'",  "'.$chanel.'",  "'.$citem\_speedy.'",  "'.$kecepatan.'",  "'.$deskripsi.'",  "'.$tgl\_reg.'",  "'.$tgl\_etat.'",  "'.$status.'",  "'.$nama.'",  "'.$kcontact.'",  "'.$status\_order.'",  "'.$alpro.'",  "'.$ccat.'",  "'.$jalan.'",  "'.$nojalan.'",  "'.$distrik.'",  "'.$kota.'",  "'.$cpack.'")  ');  }  } |

Listings 6. 23 Syntax Untuk Model Indieland.

|  |
| --- |
| public function getInet($id)  {  $this->db->select('\*');  $this->db->from('indieland');  $this->db->where('NO\_INET', $id);  $query=$this->db->get();  return $query->row();  } |

Listings 6. 24 Function Untuk Get Nomor Internet.

Fungsi dari *syntax* tersebut untuk mendapatkan nomor internet yang diinputkan pada *textbox* sistem peramalan, *syntax* tersebut akan mencari nomor internet pada tabel indieland. Jika data ditemukana, maka data akan dimunculkan. Jika data tidak ditemukan, maka akan muncul pemberitahuan bahwa data yang dicari tidak ditemukan.

|  |
| --- |
| public function chek\_duplicat($id){  $QuerySql = $this->db->query("SELECT ncli FROM indieland WHERE ncli IN ('$id')");  return $QuerySql->row();  } |

Listings 6. 25 Syntax Model Untuk Cek Duplikasi Data.

Fungsi dari *syntax* tersebut untuk melakukan cek duplikasi data pada file yang akan di *upload* dengan data pada *database*.

1. V\_indieland.php

|  |
| --- |
| <!-- Content Wrapper. Contains page content -->  <div class="content-wrapper">  <!-- Content Header (Page header) -->  <section class="content-header">  <h1>  DATA INDIELAND  <small>advanced tables</small>  </h1>  <ol class="breadcrumb">  <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>  <li><a href="#">Tables</a></li>  <li class="active">Data tables</li>  </ol>  </section>  <!-- Main content -->  <section class="content">  <div>  <button class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"><i class="fa fa-plus"></i> Upload Data</button>  </div><br>  <!-- box -->  <div class="row">  <div class="col-xs-12">  <div class="box">  <div class="box-header">  <center><h3 class="box-title"><b>KAWASAN TALAKA REGIONAL II</b></h3></center>  </div>  <div class="box-body">  <table id="example1" class="table table-bordered table-striped" style="width:100%">  <thead>  <tr>  <th>NO</th>  <th>KAWASAN</th>  <th>WITEL</th>  <th>DATEL</th>  <th>STO</th>  <th>NCLI</th>  <th>NDOS</th>  <th>NDEM</th>  <th>NO\_INET</th>  </tr>  </thead>  <tbody>  <?php $no = 1; foreach($indieland1 as $ihs){ ?>  <tr>  <td> <?php echo $no++ ?> </td>  <td><?php echo $ihs->KAWASAN?></td>  <td><?php echo $ihs->WITEL?></td>  <td><?php echo $ihs->DATEL?></td>  <td><?php echo $ihs->STO?></td>  <td><?php echo $ihs->NCLI?></td>  <td><?php echo $ihs->NDOS?></td>  <td><?php echo $ihs->NDEM?></td>  <td><?php echo $ihs->NO\_INET?></td>  </tr>  <?php } ?>  </tbody>  </table>  </div>  </div>  </div>  </div>  <!-- End Box -->  </section>  <!-- /.content-wrapper -->  </body>  <!-- Modal --->  <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog" role="document">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title" id="exampleModalLabel">FORM UPLOAD DATA PELANGGAN INDIELAND</h4>  <button type="button" class="close" data-dismiss="modal" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <?= form\_open\_multipart('c\_indieland/import'); ?>  <form method="post" action="<?= base\_url() . 'c\_indieland/import'; ?>">  <div class="form-group">  <label for="inputEmail3" class="control-label">Lampirkan File</label>  <input type="file" name="file" class="form-control" id="file" required accept=".xls, .xlsx, .csv" /></p>  <p class="help-block">Format file yang didukung .xls, .xlsx atau .csv</p>  </div>  <div class="form-group">  <input type="submit" class="btn btn-primary" value="Upload" name="upload" id="upload">  </div>  <?= form\_close(); ?>  </div>  </div>  </div>  </div>  </div> |

Listings 6. 26 Syntax View Indieland.

|  |
| --- |
| <div>  <button class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"><i class="fa fa-plus"></i> Upload Data</button>  </div><br> |

Listings 6. 27 Syntax Untuk Membuat Button.

*Syntax* tersebut berfungsi untuk membuat sebuah *button*, jika telah berhasil, maka hasilnya akan seperti berikut.

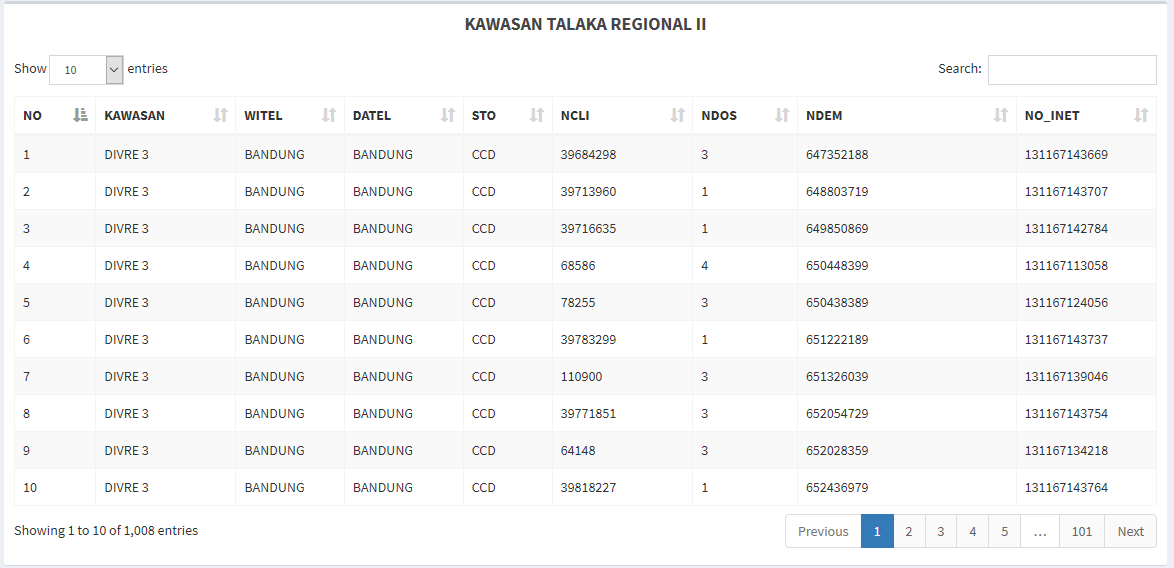


Gambar 6. 4 Hasil Dari Syntax Membuat Button.

|  |
| --- |
| <div class="row">  <div class="col-xs-12">  <div class="box">  <div class="box-header">  <center><h3 class="box-title"><b>KAWASAN TALAKA REGIONAL II</b></h3></center>  </div>  <div class="box-body">  <table id="example1" class="table table-bordered table-striped" style="width:100%">  <thead>  <tr>  <th>NO</th>  <th>KAWASAN</th>  <th>WITEL</th>  <th>DATEL</th>  <th>STO</th>  <th>NCLI</th>  <th>NDOS</th>  <th>NDEM</th>  <th>NO\_INET</th>  </tr>  </thead>  <tbody>  <?php $no = 1; foreach($indieland1 as $ihs){ ?>  <tr>  <td> <?php echo $no++ ?> </td>  <td><?php echo $ihs->KAWASAN?></td>  <td><?php echo $ihs->WITEL?></td>  <td><?php echo $ihs->DATEL?></td>  <td><?php echo $ihs->STO?></td>  <td><?php echo $ihs->NCLI?></td>  <td><?php echo $ihs->NDOS?></td>  <td><?php echo $ihs->NDEM?></td>  <td><?php echo $ihs->NO\_INET?></td>    </tr>  <?php } ?>  </tbody>  </table>  </div>  </div>  </div>  </div> |

Listings 6. 28 Syntax Untuk Membuat Form Tabel Data.

*Syntax* tersebut berfungsi untuk membuat sebauh *form* tabel data, untuk hasilnya akan seperti berikut.

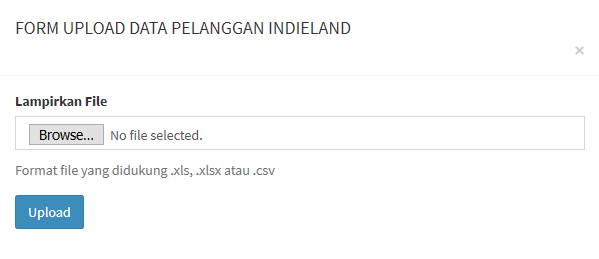


Gambar 6. 5 Hasil Dari Syntax View Tabel Data.

|  |
| --- |
| <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog" role="document">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title" id="exampleModalLabel">FORM UPLOAD DATA PELANGGAN INDIELAND</h4>  <button type="button" class="close" data-dismiss="modal" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <?= form\_open\_multipart('c\_indieland/import'); ?>  <form method="post" action="<?= base\_url() . 'c\_indieland/import'; ?>">  <div class="form-group">  <label for="inputEmail3" class="control-label">Lampirkan File</label>  <input type="file" name="file" class="form-control" id="file" required accept=".xls, .xlsx, .csv" /></p>  <p class="help-block">Format file yang didukung .xls, .xlsx atau .csv</p>  </div>  <div class="form-group">  <input type="submit" class="btn btn-primary" value="Upload" name="upload" id="upload">  </div>  <?= form\_close(); ?>  </div>  </div>  </div>  </div>  </div> |

Listing 6. 29 Syntax Untuk Membuat Form Upload Data.

Pada *syntax* tersebut berfungsi untuk membuat modal *form upload* data. *Form* ini akan muncul pada layar jika *user* menekan tombol *upload* data.Jika berhasil maka hasilnya akan seperti berikut.



Gambar 6. 6 Hasil Dari Syntax View Form Upload Data.

1. *View\_list.php*

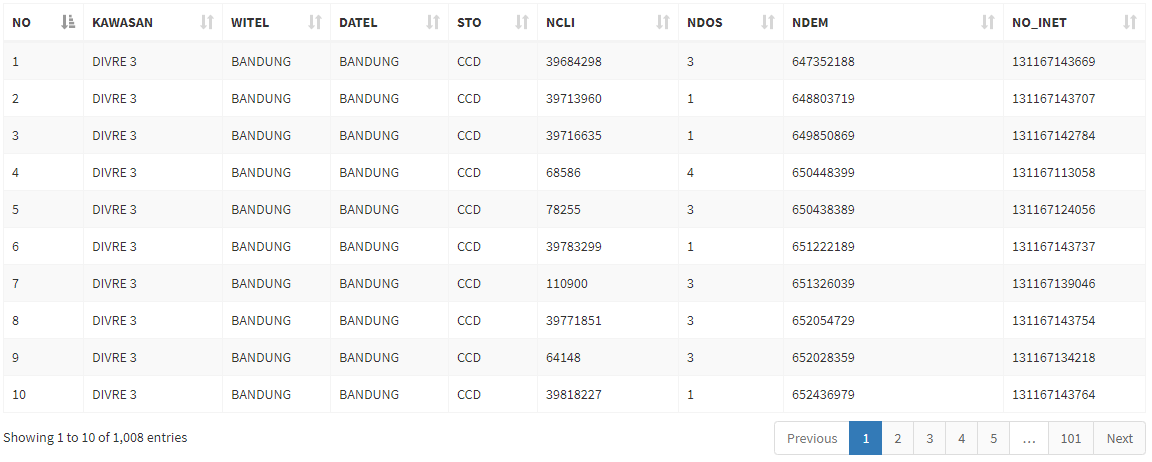
|  |
| --- |
| <?php $no = 1; foreach($indieland as $ihs){ ?>  <tr>  <td> <?php echo $no++ ?> </td>  <td><?php echo $ihs->KAWASAN?></td>  <td><?php echo $ihs->WITEL?></td>  <td><?php echo $ihs->DATEL?></td>  <td><?php echo $ihs->STO?></td>  <td><?php echo $ihs->NCLI?></td>  <td><?php echo $ihs->NDOS?></td>  <td><?php echo $ihs->NDEM?></td>  <!-- <td><?php echo $ihs->NO\_INET?></td> -->    </tr>  <?php } ?> |

Listings 6. 30 Syntax Untuk View List Indieland.

|  |
| --- |
| <?php $no = 1; foreach($indieland as $ihs){ ?>  <tr>  <td> <?php echo $no++ ?> </td>  <td><?php echo $ihs->KAWASAN?></td>  <td><?php echo $ihs->WITEL?></td>  <td><?php echo $ihs->DATEL?></td>  <td><?php echo $ihs->STO?></td>  <td><?php echo $ihs->NCLI?></td>  <td><?php echo $ihs->NDOS?></td>  <td><?php echo $ihs->NDEM?></td>  <!-- <td><?php echo $ihs->NO\_INET?></td> -->    </tr>  <?php } ?> |

Listings 6. 31 Function Untuk Menampilkan Data Pada Sistem.

*Syntax* tersebut berfungsi menampilkan data pada sistem peramalan yang diambil dari *database*. Jika berhasil, maka hasilnya akan seperti berikut.



Gambar 6. 7 Hasil Dari Syntax View List Indieland.

*Form* indieland movieyuk ini akan menampilkan jumlah pelanggan yang tidak berlangganan pada *addOn* movieyuk ini, selain jumlah pelanggan terdapat juga *form* untuk *upload* data movieyuk, periode untuk melihat jumlah pelanggan yang tidak berlanggan pada bulan tertentu, dan *download* file jumlah pelanggan yang tidak berlangganan *addOn* movieyuk ini. Berikut merupakan *syntax* yang digunakan untuk membuat *form* indihome movieyuk.

1. C\_movieyuk.php

|  |
| --- |
| <?php  class C\_movieyuk extends CI\_Controller{  public function \_\_construct(){  parent::\_\_construct();  $this->load->model('m\_movieyuk');  $this->load->library('Excel');  // if($this->session->userdata('role\_id')!='1' & '2'){  // $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  // <strong>Anda belum Login !!!</strong>  // <button type="button" class="close" data-dismiss="alert" aria-label="Close">  // <span aria-hidden="true">&times;</span>  // </button>  // </div>');  // redirect('auth/login');  //}  }  public function index (){    $this->load->view('templates/header');  //$this->load->view('templates/sidebar');  if ($this->session->userdata('role\_id') ==='1') {  $this->load->view('templates/sidebar');  }elseif($this->session->userdata('role\_id') ==='2'){  $this->load->view('templates/sidebar\_user');  }else{  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  <strong>Anda belum Login !!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('auth/login');  }  $this->load->view('movieyuk/v\_movieyuk');  $this->load->view('templates/footer');  }  public function view\_movieyuk($id){  $explode = explode('~', $id);  $periode\_awal = $explode[0];  $periode\_akhir = $explode[1];  $data['periode\_awal'] = $periode\_awal;  $data['periode\_akhir'] = $periode\_akhir;  $data['movieyuk'] = $this->m\_movieyuk->movieyuk($periode\_awal, $periode\_akhir);  $this->load->view('movieyuk/view\_list', $data);  }    public function import(){  if(isset($\_FILES["file"]["name"]))  {  $path = $\_FILES["file"]["tmp\_name"];  $object = PHPExcel\_IOFactory::load($path);  foreach($object->getWorksheetIterator() as $worksheet)  {  $highestRow = $worksheet->getHighestRow();  $highestColumn = $worksheet->getHighestColumn();  for($row=2; $row<=$highestRow; $row++)  {  $witel = $worksheet->getCellByColumnAndRow(0, $row)->getValue();  $ncli = $worksheet->getCellByColumnAndRow(1, $row)->getValue();  $ndos = $worksheet->getCellByColumnAndRow(2, $row)->getValue();  $ndem = $worksheet->getCellByColumnAndRow(3, $row)->getValue();  $no\_inet = $worksheet->getCellByColumnAndRow(4, $row)->getValue();  $item = $worksheet->getCellByColumnAndRow(5, $row)->getValue();  $price = $worksheet->getCellByColumnAndRow(6, $row)->getValue();  $tgl\_va = $worksheet->getCellByColumnAndRow(7, $row)->getValue();  $tgl\_ps = $worksheet->getCellByColumnAndRow(8, $row)->getValue();  $kcontact = $worksheet->getCellByColumnAndRow(9, $row)->getValue();  $cek\_duplicat = $this->m\_movieyuk->chek\_duplicat($no\_inet);  if ($cek\_duplicat != NULL) {  if ($cek\_duplicat->no\_inet == $no\_inet) {  $this->m\_movieyuk->update\_duplicat($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact);  }else{  $this->m\_movieyuk->upload($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact);  }  }else{  $this->m\_movieyuk->upload($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact);  }  }  }  // $this->m\_movieyuk->upload($data);  redirect('c\_dashboard\_admin/dashboard');    }  }  public function downloadexcel($id) {  $explode = explode('~', $id);  $periode\_awal = $explode[0];  $periode\_akhir = $explode[1];  $data['excelmovieyuk'] = $this->m\_movieyuk->downloadmovin($periode\_awal, $periode\_akhir);    $this->load->view('movieyuk/v\_downloadexcel', $data);  }  } |

Listings 6. 32 Syntax Controller Untuk Movieyuk.

|  |
| --- |
| public function \_\_construct(){  parent::\_\_construct();  $this->load->model('m\_movin');  $this->load->library('Excel'); |

Listings 6. 33 Function Untuk Melakukan Load Library.

Pada *syntax* tersebut akan menggunakan satu buah model yang berhubungan langsung dengan *database* dan satu buah *library* untuk melakukan *upload* dan *download* data excel yang ber-ekstensi *xlsx,* dan *xls.*

|  |
| --- |
| public function index (){    $this->load->view('templates/header');  //$this->load->view('templates/sidebar');  if ($this->session->userdata('role\_id') ==='1') {  $this->load->view('templates/sidebar');  }elseif($this->session->userdata('role\_id') ==='2'){  $this->load->view('templates/sidebar\_user');  }else{  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  <strong>Anda belum Login !!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('auth/login');  }  $this->load->view('movieyuk/v\_movieyuk');  $this->load->view('templates/footer');  } |

Listings 6. 34 Syntax Untuk Validasi Sidebar.

Pada *syntax* tersebut terdapat *if* dan *else* yang berfungsi untuk menentukan *sidebar* yang akan ditampilkan. Jika *user* dengan *role\_id* = 1 yang *login*, maka akan menampilkan *sidebar* admin. Jika *user* dengan *role\_id* = 2 yang *login*, maka akan menampilkan *sidebar* *user*. Lalu jika *session* *login*-nya telah habis, maka sistem akan secara otomatis melakukan *logout* akun, sehingga akan kembali pada halaman *login*.

|  |
| --- |
| public function view\_movieyuk($id){  $explode = explode('~', $id);  $periode\_awal = $explode[0];  $periode\_akhir = $explode[1];  $data['periode\_awal'] = $periode\_awal;  $data['periode\_akhir'] = $periode\_akhir;  $data['movieyuk'] = $this->m\_movieyuk->movieyuk($periode\_awal, $periode\_akhir);  $this->load->view('movieyuk/view\_list', $data);  } |

Listings 6. 35 Syntax Untuk Menampilkan Data Berdasarkan Periode.

Pada *syntax* tersebut berfungsi untuk membuat pencarian data pada *database* dengan menggunakan periode awal dan periode akhir.

|  |
| --- |
| public function import(){  if(isset($\_FILES["file"]["name"]))  {  $path = $\_FILES["file"]["tmp\_name"];  $object = PHPExcel\_IOFactory::load($path);  foreach($object->getWorksheetIterator() as $worksheet)  {  $highestRow = $worksheet->getHighestRow();  $highestColumn = $worksheet->getHighestColumn();  for($row=2; $row<=$highestRow; $row++)  {  $witel = $worksheet->getCellByColumnAndRow(0, $row)->getValue();  $ncli = $worksheet->getCellByColumnAndRow(1, $row)->getValue();  $ndos = $worksheet->getCellByColumnAndRow(2, $row)->getValue();  $ndem = $worksheet->getCellByColumnAndRow(3, $row)->getValue();  $no\_inet = $worksheet->getCellByColumnAndRow(4, $row)->getValue();  $item = $worksheet->getCellByColumnAndRow(5, $row)->getValue();  $price = $worksheet->getCellByColumnAndRow(6, $row)->getValue();  $tgl\_va = $worksheet->getCellByColumnAndRow(7, $row)->getValue();  $tgl\_ps = $worksheet->getCellByColumnAndRow(8, $row)->getValue();  $kcontact = $worksheet->getCellByColumnAndRow(9, $row)->getValue();  $cek\_duplicat = $this->m\_movin->chek\_duplicat($no\_inet);  if ($cek\_duplicat != NULL) {  if ($cek\_duplicat->no\_inet == $no\_inet) {  $this->m\_movieyuk->update\_duplicat($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact);  }else{  $this->m\_movieyuk->upload($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact);  }  }else{  $this->m\_movieyuk->upload($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact);  }  }  }  // $this->m\_movieyuk->upload($data);  redirect('c\_dashboard\_admin/dashboard');    }  } |

Listings 6. 36 Syntax Untuk Memberikan Fungsi Import File.

*Syntax* tersebut berfungsi untuk *upload* data yang memiliki ekstensi *xlsx,* dan *xls*. Selain itu pada saat akan melakukan *upload* data, sistem akan melakukan pengecekan duplikasi data, agar pada saat dilakukan *upload* pada *database* tidak terjadi redudansi data.

|  |
| --- |
| public function downloadexcel($id) {  $explode = explode('~', $id);  $periode\_awal = $explode[0];  $periode\_akhir = $explode[1];  $data['excelmovieyuk'] = $this->m\_movieyuk->downloadmovieyuk($periode\_awal, $periode\_akhir);    $this->load->view('movieyuk/v\_downloadexcel', $data);  }  } |

Listings 6. 37 Syntax Untuk Memberikan Fungsi Download File.

*Syntax* tersebut berfungsi untuk men-*download* file excel yang terdapat pada sistem peramalan untuk mengetahui detail mengenai pelanggan yang tidak berlangganan pada *addOn* movieyuk ini.

1. M\_movieyuk.php

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class M\_movieyuk extends CI\_Model {  public function \_\_construct()  {  parent::\_\_construct();  }  Public function movieyuk($periode\_awal, $periode\_akhir) {  if ($periode\_awal==null||$periode\_akhir==null){  $wordsquery = "SELECT WITEL, COUNT(\*) as jtb FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET) GROUP BY WITEL";  }else{  $wordsquery = "SELECT WITEL, COUNT(\*) as jtb FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET) AND TGL\_REG BETWEEN '$periode\_awal' AND '$periode\_akhir' GROUP BY WITEL";  }    $query = $this->db->query($wordsquery);  return $query->result();  }  // public function upload($data)  // {  // $this->db->insert\_batch('movieyuk', $data);  // }  public function chek\_duplicat($id){  $QuerySql = $this->db->query("SELECT no\_inet FROM movieyuk WHERE no\_inet IN ('$id')");  return $QuerySql->row();  }  Public function downloadmovieyuk($periode\_awal, $periode\_akhir) {  if ($periode\_awal==null||$periode\_akhir==null){  $wordsquery = "SELECT KAWASAN,WITEL,DATEL,STO,NCLI,NDOS,NDEM,NO\_INET,ND,CHANEL,CITEM\_SPEEDY,KECEPATAN,DESKRIPSI,TGL\_REG,TGL\_ETAT,STATUS,NAMA,KCONTACT,STATUS\_ORDER,ALPRO,CCAT,JALAN,NOJALAN,DISTRIK,KOTA,CPACK FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET)";  }else{  $wordsquery = "SELECT KAWASAN,WITEL,DATEL,STO,NCLI,NDOS,NDEM,NO\_INET,ND,CHANEL,CITEM\_SPEEDY,KECEPATAN,DESKRIPSI,TGL\_REG,TGL\_ETAT,STATUS,NAMA,KCONTACT,STATUS\_ORDER,ALPRO,CCAT,JALAN,NOJALAN,DISTRIK,KOTA,CPACK FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET) AND TGL\_REG BETWEEN '$periode\_awal' AND '$periode\_akhir'";  }    $query = $this->db->query($wordsquery);  return $query->result();  }  public function update\_duplicat($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact) {  $this->db->query('UPDATE movieyuk SET witel = "'.$witel.'",  ncli = "'.$ncli.'",  ndos = "'.$ndos.'",  ndem = "'.$ndem.'",  no\_inet = "'.$no\_inet.'",  item = "'.$item.'",  price = "'.$price.'",  tgl\_va = "'.$tgl\_va.'",  tgl\_ps = "'.$tgl\_ps.'",  kcontact = "'.$kcontact.'" WHERE no\_inet = "'.$no\_inet.'"');  }    public function upload($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact)  {  $this->db->query('REPLACE INTO movieyuk (witel, ncli, ndos, ndem, no\_inet, item, price, tgl\_va, tgl\_ps, kcontact)  VALUES("'.$witel.'",  "'.$ncli.'",  "'.$ndos.'",  "'.$ndem.'",  "'.$no\_inet.'",  "'.$item.'",  "'.$price.'",  "'.$tgl\_va.'",  "'.$tgl\_ps.'",  "'.$kcontact.'")  ');  }  } |

Listings 6. 38 Syntax Model Movieyuk.

|  |
| --- |
| Public function movieyuk($periode\_awal, $periode\_akhir) {  if ($periode\_awal==null||$periode\_akhir==null){  $wordsquery = "SELECT WITEL, COUNT(\*) as jtb FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET) GROUP BY WITEL";  }else{  $wordsquery = "SELECT WITEL, COUNT(\*) as jtb FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET) AND TGL\_REG BETWEEN '$periode\_awal' AND '$periode\_akhir' GROUP BY WITEL";  }    $query = $this->db->query($wordsquery);  return $query->result();  } |

Listings 6. 39 Syntax Untuk Menampilkan Data Berdasarkan Periode..

Fungsi dari *syntax* tersebut untuk mencari beberapa data pada bulan tertentu yang dicari pada 2 tabel, yakni tabel indieland dan tabel movieyuk, dengan nomor internet yang sama atau terdapat pada kedua tabel tersebut. Fungsi lain dari *syntax* tersebut yakni, untuk melihat jumlah pelanggan yang tidak berlangganan pada *addOn* tersebut pada periode tertentu.

|  |
| --- |
| Public function downloadmovieyuk($periode\_awal, $periode\_akhir) {  if ($periode\_awal==null||$periode\_akhir==null){  $wordsquery = "SELECT KAWASAN,WITEL,DATEL,STO,NCLI,NDOS,NDEM,NO\_INET,ND,CHANEL,CITEM\_SPEEDY,KECEPATAN,DESKRIPSI,TGL\_REG,TGL\_ETAT,STATUS,NAMA,KCONTACT,STATUS\_ORDER,ALPRO,CCAT,JALAN,NOJALAN,DISTRIK,KOTA,CPACK FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET)";  }else{  $wordsquery = "SELECT KAWASAN,WITEL,DATEL,STO,NCLI,NDOS,NDEM,NO\_INET,ND,CHANEL,CITEM\_SPEEDY,KECEPATAN,DESKRIPSI,TGL\_REG,TGL\_ETAT,STATUS,NAMA,KCONTACT,STATUS\_ORDER,ALPRO,CCAT,JALAN,NOJALAN,DISTRIK,KOTA,CPACK FROM indieland WHERE NOT EXISTS (SELECT \* from movieyuk WHERE indieland.NO\_INET = movieyuk.NO\_INET) AND TGL\_REG BETWEEN '$periode\_awal' AND '$periode\_akhir'";  }  $query = $this->db->query($wordsquery);  return $query->result();  } |

Listings 6. 40 Syntax Untuk Memberikan Fungsi Download File Pada Sistem.

Fungsi dari *syntax* tersebut untuk men-*download* file excel yang isinya mengenai keterangan pelanggan.

|  |
| --- |
| public function update\_duplicat($witel, $ncli, $ndos, $ndem, $no\_inet, $item, $price, $tgl\_va, $tgl\_ps, $kcontact) {  $this->db->query('UPDATE movieyuk SET witel = "'.$witel.'",  ncli = "'.$ncli.'",  ndos = "'.$ndos.'",  ndem = "'.$ndem.'",  no\_inet = "'.$no\_inet.'",  item = "'.$item.'",  price = "'.$price.'",  tgl\_va = "'.$tgl\_va.'",  tgl\_ps = "'.$tgl\_ps.'",  kcontact = "'.$kcontact.'" WHERE no\_inet = "'.$no\_inet.'"');  } |

Listings 6. 41 Syntax Untuk Update Data.

Fungsi dari *syntax* tersebut untuk melakukan *update* data pada *database* agar tidak terjadi redudansi data pada saat melakukan *upload* data baru.

1. V\_movin.php

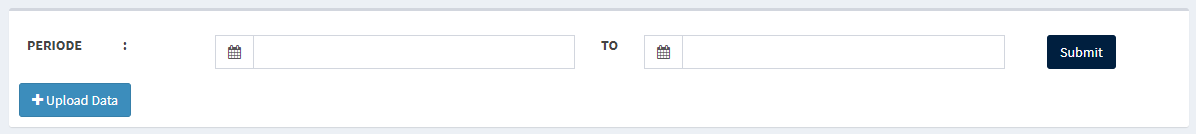
|  |
| --- |
| <div class="content-wrapper">  <!-- Content Header (Page header) -->  <section class="content-header">  <h1>  DATA PROSPEK  </h1>  <ol class="breadcrumb">  <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>  <li><a href="#">Data Prospek</a></li>  <li class="active">Movieyuk</li>  </ol>  </section>  <!-- Main content -->  <section class="content">  <div class="row">  <div class="col-xs-12">  <div class="box">  <div class="box-header">  <table id="example1" class="table no-border" >  <tr>  <th>PERIODE&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;:</th>  <td><div class="input-group date">  <div class="input-group-addon">  <i class="fa fa-calendar"></i>  </div>  <input type="text" class="form-control pull-right periode\_awal" id="datepicker" name="periode\_awal">  </div>  </td>  <th><center>TO</center></th>  <td><div class="input-group date">  <div class="input-group-addon">  <i class="fa fa-calendar"></i>  </div>  <input type="text" class="form-control pull-right periode\_akhir" id="datepicker1" name="periode\_akhir">  </div>  </td>  <td></td>  <td><button id="idMovieyuk" type="submit" class="btn bg-navy" name="submit">Submit</button></td>    </tr>  </table>  <div>  <button class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"><i class="fa fa-plus"></i> Upload Data</button>  </div>  </div>  </div>  <div class="box">  <div class="box-header">  <h3 class="box-title">MOVIEYUK</h3>  </div>  <!-- /.box-header -->  <div class="box-body">  <table id="example2" class="table table-bordered table-hover">  <thead>  <tr>  <th>NO</th>  <th>WITEL</th>  <th>Jumlah tidak berlangganan</th>  </tr>  </thead>  <tbody id="view\_movin">  </tfoot>  </table>  </div>  <!-- /.box-body -->  </div>  <!-- /.box -->  </div>  <!-- /.col -->  </div>  <!-- /.row -->  </section>  <!-- /.content -->  </div>  <!-- /.content-wrapper -->  </body>  <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog" role="document">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title" id="exampleModalLabel">FORM UPLOAD DATA PELANGGAN MOVIEYUK</h4>  <button type="button" class="close" data-dismiss="modal" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <?= form\_open\_multipart('c\_movieyuk/import'); ?>  <form method="post" action="<?= base\_url() . 'c\_movieyuk/import'; ?>">  <div class="form-group">  <label for="inputEmail3" class="control-label">Lampirkan File</label>  <input type="file" name="file" class="form-control" id="file" required accept=".xls, .xlsx, .csv" /></p>  <p class="help-block">Format file yang didukung .xls, .xlsx atau .csv</p>  </div>  <div class="form-group">  <input type="submit" class="btn btn-primary" value="Upload" name="upload" id="upload">  </div>  <?= form\_close(); ?>  </div>  </div>  </div>  </div> |

Listings 6. 42 Syntax Lengkap Untuk View Movieyuk.

|  |
| --- |
| <section class="content">  <div class="row">  <div class="col-xs-12">  <div class="box">  <div class="box-header">  <table id="example1" class="table no-border" >  <tr>  <th>PERIODE&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;:</th>  <td><div class="input-group date">  <div class="input-group-addon">  <i class="fa fa-calendar"></i>  </div>  <input type="text" class="form-control pull-right periode\_awal" id="datepicker" name="periode\_awal">  </div>  </td>  <th><center>TO</center></th>  <td><div class="input-group date">  <div class="input-group-addon">  <i class="fa fa-calendar"></i>  </div>  <input type="text" class="form-control pull-right periode\_akhir" id="datepicker1" name="periode\_akhir">  </div>  </td>  <td></td>  <td><button id="idMovieyuk" type="submit" class="btn bg-navy" name="submit">Submit</button></td>  </tr>  </table>  <div>  <button class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"><i class="fa fa-plus"></i> Upload Data</button>  </div>  </div>  </div> |

Listings 6. 43 Syntax Untuk Membuat Textbox dan Button.

Fungsi dari *syntax* tersebut untuk membuat *textbox* dan *button*, *textbox* itu diperuntukkan tanggal dalam mencari jumlah data pada periode tertentu. Jika berhasil, maka hasilnya akan tampak seperti berikut.



Gambar 6. 8 Hasil Dari Syntax View Membuat TextBox dan Button.

|  |
| --- |
| <div class="box">  <div class="box-header">  <h3 class="box-title">MOVIEYUK</h3>  </div>  <!-- /.box-header -->  <div class="box-body">  <table id="example2" class="table table-bordered table-hover">  <thead>  <tr>  <th>NO</th>  <th>WITEL</th>  <th>Jumlah tidak berlangganan</th>  </tr>  </thead>  <tbody id="view\_movieyuk">  </tfoot>  </table>  </div>  <!-- /.box-body -->  </div>  <!-- /.box -->  </div>  <!-- /.col -->  </div>  <!-- /.row -->  </section>  <!-- /.content -->  </div>  <!-- /.content-wrapper -->  </body> |

Listings 6. 44 Syntax Untuk Membuat Tabel.

Pada *syntax* berikut berfungsi untuk membuat sebuah tabel dengan menampilkan total pelanggan dengan WITEL (Wilayah Talaka). Jika berhasil, maka tampak akan seperti berikut.



Gambar 6. 9 Syntax View Untuk Membuat Tabel Data.

|  |
| --- |
| <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog" role="document">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title" id="exampleModalLabel">FORM UPLOAD DATA PELANGGAN MOVIEYUK</h4>  <button type="button" class="close" data-dismiss="modal" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <?= form\_open\_multipart('c\_movieyuk/import'); ?>  <form method="post" action="<?= base\_url() . 'c\_movieyuk/import'; ?>">  <div class="form-group">  <label for="inputEmail3" class="control-label">Lampirkan File</label>  <input type="file" name="file" class="form-control" id="file" required accept=".xls, .xlsx, .csv" /></p>  <p class="help-block">Format file yang didukung .xls, .xlsx atau .csv</p>  </div>  <div class="form-group">  <input type="submit" class="btn btn-primary" value="Upload" name="upload" id="upload">  </div>  <?= form\_close(); ?>  </div>  </div>  </div>  </div> |

Listings 6. 45 Syntax Untuk Membuat Form Upload Data.

Pada *syntax* tersebut berfungsi untuk membuat modal *form upload* data pelanggan, format file yang didukung dalam *upload* *form* tersebut adalah, *xlsx,* dan *xls*. Jika telah berhasil, maka hasilnya akan seperti berikut.



Gambar 6. 10 Hasil Dari Syntax Form Upload Data Pelanggan.

1. View\_list.php

|  |
| --- |
| <?php $no = 1; foreach($movieyuk as $ihs){  $array[]=array(  'TOTAL\_MOVIEYUK' => $ihs->jtb  );  $sum\_movieyuk = array\_sum(array\_column($array, 'TOTAL\_MOVIEYUK'))  ?>  <tr>  <td> <?php echo $no++ ?> </td>  <td><?php echo $ihs->WITEL?></td>  <td><?php echo $ihs->jtb?></td>    </tr>  <?php } ?>  <tr>  <td><b>TOTAL</b></td>  <td colspan="2" style="text-align: center"><a href="<?php echo base\_url('C\_movieyuk/downloadexcel/'.$periode\_awal.'~'.$periode\_akhir) ?>"><b><?php if ($movieyuk != NULL) {  echo $sum\_movieyuk;}else{echo "DATA TIDAK TERSEDIA";}?></b></td>  </tr> |

Listings 6. 46 Syntax View List Pada Movieyuk.

|  |
| --- |
| <?php $no = 1; foreach($movieyuk as $ihs){  $array[]=array(  'TOTAL\_MOVIEYUK' => $ihs->jtb  );  $sum\_movieyuk = array\_sum(array\_column($array, 'TOTAL\_MOVIEYUK'))  ?>  <tr>  <td> <?php echo $no++ ?> </td>  <td><?php echo $ihs->WITEL?></td>  <td><?php echo $ihs->jtb?></td>    </tr> |

Listinge 6. 47 Function Untuk Menampilkan Total Pelanggan.

*Syntax* tersebut berfungsi untuk menampilkan total pelanggan yang tidak berlangganan pada produk *addOn* movieyuk.

|  |
| --- |
| <?php } ?>  <tr>  <td><b>TOTAL</b></td>  <td colspan="2" style="text-align: center"><a href="<?php echo base\_url('C\_movieyuk/downloadexcel/'.$periode\_awal.'~'.$periode\_akhir) ?>"><b><?php if ($movieyuk != NULL) {  echo $sum\_movieyuk;}else{echo "DATA TIDAK TERSEDIA";}?></b></td>  </tr> |

Listings 6. 48 Function Unutuk Menampilkan Data Berdasarkan Periode.

*Syntax* tersebut berfungsi untuk menampilkan total pelanggan pada periode tertentu, jika pada periode tertentu tidak terdapat pelanggan, maka sistem akan menampilkan pemberitahuan pada *user*.

1. View\_downloadExcel.php

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  </head>  <body>  <table border="1">  <tr>  <th>KAWASAN</th>  <th>WITEL</th>  <th>DATEL</th>  <th>STO</th>  <th>NCLI</th>  <th>NDOS</th>  <th>NDEM</th>  <th>NO\_INET</th>  <th>ND</th>  <th>CHANEL</th>  <th>CITEM\_SPEEDY</th>  <th>KECEPATAN</th>  <th>DESKRIPSI</th>  <th>TGL\_REG</th>  <th>TGL\_ETAT</th>  <th>STATUS</th>  <th>NAMA</th>  <th>KCONTACT</th>  <th>STATUS\_ORDER</th>  <th>ALPRO</th>  <th>CCAT</th>  <th>JALAN</th>  <th>NOJALAN</th>  <th>DISTRIK</th>  <th>KOTA</th>  <th>CPACK</th>  </tr>  <?php  header ("Content-type: application/octet-stream");  header ("Content-Disposition: attachment; filename=Data Movin TOTAL.xls");  header ("Pragma: no-cache");  header ("Expires: 0");  foreach ($excelmovieyuk as $key => $value) { ?>  <tr>  <td><?= $value->KAWASAN ?></td>  <td><?= $value->WITEL ?></td>  <td><?= $value->DATEL ?></td>  <td><?= $value->STO ?></td>  <td><?= $value->NCLI ?></td>  <td><?= $value->NDOS ?></td>  <td><?= $value->NDEM ?></td>  <td><?= $value->NO\_INET ?></td>  <td><?= $value->ND ?></td>  <td><?= $value->CHANEL ?></td>  <td><?= $value->CITEM\_SPEEDY ?></td>  <td><?= $value->KECEPATAN ?></td>  <td><?= $value->DESKRIPSI ?></td>  <td><?= $value->TGL\_REG ?></td>  <td><?= $value->TGL\_ETAT ?></td>  <td><?= $value->STATUS ?></td>  <td><?= $value->NAMA ?></td>  <td><?= $value->KCONTACT ?></td>  <td><?= $value->STATUS\_ORDER ?></td>  <td><?= $value->ALPRO ?></td>  <td><?= $value->CCAT ?></td>  <td><?= $value->JALAN ?></td>  <td><?= $value->NOJALAN ?></td>  <td><?= $value->DISTRIK ?></td>  <td><?= $value->KOTA ?></td>  <td><?= $value->CPACK ?></td>  </tr>  <?php } ?>  </table>  </body>  </html> |

Listings 6. 49 Syntax Untuk Download Data Movieyuk.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title></title>  </head>  <body>  <table border="1">  <tr>  <th>KAWASAN</th>  <th>WITEL</th>  <th>DATEL</th>  <th>STO</th>  <th>NCLI</th>  <th>NDOS</th>  <th>NDEM</th>  <th>NO\_INET</th>  <th>ND</th>  <th>CHANEL</th>  <th>CITEM\_SPEEDY</th>  <th>KECEPATAN</th>  <th>DESKRIPSI</th>  <th>TGL\_REG</th>  <th>TGL\_ETAT</th>  <th>STATUS</th>  <th>NAMA</th>  <th>KCONTACT</th>  <th>STATUS\_ORDER</th>  <th>ALPRO</th>  <th>CCAT</th>  <th>JALAN</th>  <th>NOJALAN</th>  <th>DISTRIK</th>  <th>KOTA</th>  <th>CPACK</th>  </tr> |

Listings 6. 50 Syntax Untuk Membuat Judul Pada Kolom Data.

Pada *syntax* tersebut, sistem akan membuat judul kolom pada file yang akan *user download*, file tersebut berisi mengenai total pelanggan yang tidak berlangganan dan beberapa informasi lainnya.

1. Perancangan *Form* Kelola Akun

Ini merupakan *syntax* untuk membuat *form* kelola akun.

1. C\_data\_user.php

|  |
| --- |
| <?php  defined('BASEPATH') or exit('No direct script access allowed');  class C\_data\_user extends CI\_Controller  {  public function \_\_construct()  {  parent::\_\_construct();  $this->load->model('m\_data\_user');  if ($this->session->userdata('role\_id') != '1') {  $this->session->set\_flashdata('pesan', '<div class="alert alert-danger" role="alert">  <strong>Anda belum Login !!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('auth/login');  }  }  public function index()  {  $data['data\_user'] = $this->m\_data\_user->tampil\_data()->result();  $this->load->view('templates/header');  $this->load->view('templates/sidebar');  $this->load->view('data\_user/v\_data\_user', $data);  $this->load->view('templates/footer');  }  public function tambah\_aksi()  {  $nama = $this->input->post('nama');  $username = $this->input->post('username');  $password = $this->input->post('password');  $role\_id = $this->input->post('role\_id');  $data = array(  'nama' => $nama,  'username' => $username,  'password' => $password,  'role\_id' => $role\_id  );  $this->m\_data\_user->input\_data($data, 'login');  // $this->session->set\_flashdata('mesage','  // <div class="alert alert-success alert-dismissible" role="alert">  // <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>  // Data Berhasil Ditambahkan  // </div>');  redirect('c\_data\_user/index');  }  public function hapus($id)  {  $where = array('id' => $id);  $this->m\_data\_user->hapus\_data($where, 'login');  // $this->session->set\_flashdata('mesage','  // <div class="alert alert-danger alert-dismissible" role="alert">  // <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>  // Data Berhasil Dihapus  // </div>');  redirect('c\_data\_user/index');  }  public function edit($id)  {  $where = array('id' => $id);  $data['data\_user'] = $this->m\_data\_user->edit\_data($where, 'login')->result();  $this->load->view('templates/header');  $this->load->view('templates/sidebar');  $this->load->view('data\_user/v\_edit\_user', $data);  $this->load->view('templates/footer');  }  public function update()  {  $id = $this->input->post('id');  $nama = $this->input->post('nama');  $username = $this->input->post('username');  $password = $this->input->post('password');  $role\_id = $this->input->post('role\_id');  $data = array(  'nama' => $nama,  'username' => $username,  'password' => $password,  'role\_id' => $role\_id  );  $where = array('id' => $id);  $this->m\_data\_user->update\_data($where, $data, 'login');  $this->session->set\_flashdata('mesage', '  <div class="alert alert-info alert-dismissible" role="alert">  <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>  Data Berhasil Diupdate  </div>');  redirect('c\_data\_user/index');  }  public function detail($id)  {  $detail = $this->m\_data\_user->detail\_data($id);  $data['detail'] = $detail;  $this->load->view('templates/header');  $this->load->view('templates/sidebar');  $this->load->view('data\_user/v\_detail\_user', $data);  $this->load->view('templates/footer');  }  } |

Listings 6. 51 Syntax Controller Untuk Form Kelola Akun.

|  |
| --- |
| public function \_\_construct()  {  parent::\_\_construct();  $this->load->model('m\_data\_user');  if ($this->session->userdata('role\_id') != '1') {  $this->session->set\_flashdata('pesan', '<div class="alert alert-danger" role="alert">  <strong>Anda belum Login !!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('auth/login');  }  } |

Listings 6. 52 Syntax Untuk Validasi Akun Yang Login Pada Sistem.

*Syntax* tersebut berfungsi untuk melakukan validasi terhadap *user* yang akan *login* pada sistem, untuk *form* kelola akun ini yang dapat mengaksesnya hanya admin saja. Jika pegawai atau *role\_id* = 2, maka sistem akan menolak *login* tersebut, karena *user* dengan *role\_id* = 2 tidak diberikan hak akses.

|  |
| --- |
| public function tambah\_aksi()  {  $nama = $this->input->post('nama');  $username = $this->input->post('username');  $password = $this->input->post('password');  $role\_id = $this->input->post('role\_id');  $data = array(  'nama' => $nama,  'username' => $username,  'password' => $password,  'role\_id' => $role\_id  );  $this->m\_data\_user->input\_data($data, 'login');  redirect('c\_data\_user/index');  } |

Listings 6. 53 Syntax Untuk Menambahkan Akun Baru Pada Sistem.

*Syntax* tersebut berfungsi untuk membuat akun baru.

|  |
| --- |
| public function hapus($id)  {  $where = array('id' => $id);  $this->m\_data\_user->hapus\_data($where, 'login');    redirect('c\_data\_user/index');  } |

Listings 6. 54 Syntax Untuk Membuat Aksi Hapus Data.

*Syntax* tersebut berfungsi untuk menghapus akun *user* yang sudah tidak digunakan.

|  |
| --- |
| public function edit($id)  {  $where = array('id' => $id);  $data['data\_user'] = $this->m\_data\_user->edit\_data($where, 'login')->result();  $this->load->view('templates/header');  $this->load->view('templates/sidebar');  $this->load->view('data\_user/v\_edit\_user', $data);  $this->load->view('templates/footer');  } |

Listings 6. 55 Syntax Untuk Mengubah Data Akun.

*Syntax* tersebut untuk melakukan *edit* data akun, seperti merubah *password*, *role\_id*, *username*, dan nama.

|  |
| --- |
| public function detail($id)  {  $detail = $this->m\_data\_user->detail\_data($id);  $data['detail'] = $detail;  $this->load->view('templates/header');  $this->load->view('templates/sidebar');  $this->load->view('data\_user/v\_detail\_user', $data);  $this->load->view('templates/footer');  } |

Listings 6. 56 Syntax Untuk Melihat Detail Akun.

*Syntax* tersebut berfungsi untuk memberikan aksi melihat detail akun yang telah dibuat.

1. M\_data\_user.php

|  |
| --- |
| <?php  defined('BASEPATH') OR exit('No direct script access allowed');  class M\_data\_user extends CI\_Model {  public function \_\_construct()  {  parent::\_\_construct();  }  public function tampil\_data()  {  return $this->db->get('login');  }  public function input\_data($data,$table)  {  $this->db->insert($table, $data);  }  public function hapus\_data($where, $table)  {  $this->db->where($where);  $this->db->delete($table);  }  public function edit\_data($where, $table)  {  return $this->db->get\_where($table, $where);  }    public function update\_data($where, $data, $table)  {  $this->db->where($where);  $this->db->update($table,$data);  }  public function detail\_data($id = NULL)  {  $query = $this->db->get\_where('login', array('id' => $id))->row();  return $query;  }  } |

Listings 6. 57 Syntax Untuk Model Kelola Data User.

|  |
| --- |
| public function tampil\_data()  {  return $this->db->get('login');  }  public function input\_data($data,$table)  {  $this->db->insert($table, $data);  }  public function hapus\_data($where, $table)  {  $this->db->where($where);  $this->db->delete($table);  }  public function edit\_data($where, $table)  {  return $this->db->get\_where($table, $where);  }    public function update\_data($where, $data, $table)  {  $this->db->where($where);  $this->db->update($table,$data);  }  public function detail\_data($id = NULL)  {  $query = $this->db->get\_where('login', array('id' => $id))->row();  return $query;  } |

Listings 6. 58 Function Untuk Menghubungkan Sistem Dengan Database.

*Syntax* tersebut berfungsi untuk menghubungkan sistem dengan *database*, karena pada *form* kelola akun ini berhubungan langsung dengan akun yang diperlukan untuk *login* pada sistem peramalan.

1. View\_data\_user.php

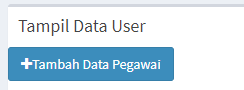
|  |
| --- |
| <!-- Content Wrapper. Contains page content -->  <div class="content-wrapper">  <!-- Content Header (Page header) -->  <section class="content-header">  <h1>  DATA USER  </h1>  <ol class="breadcrumb">  <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>  <li><a href="#">Kelola Data</a></li>  <li class="active">Data User</li>  </ol>  </section>  <!-- Main content -->  <section class="content">  <div class="row">  <div class="col-xs-12">  <div class="box">  <div class="box-header">  <h3 class="box-title">Tampil Data User</h3>  </div>  <!-- /.box-header -->  <button class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"><i class="fa fa-plus"></i>Tambah Data Pegawai</button>  <div class="box-body">    <table id="example1" class="table table-bordered table-hover">  <thead>  <tr>  <th style="width : 5px">NO</th>  <th style="width : 100px">Nama</th>  <th style="width : 100px">Username</th>  <th style="width : 100px">Role</th>  <th style="width : 10px">Aksi</th>  </tr>  </thead>  <tbody>  <?php $no = 1; foreach($data\_user as $ihs){ ?>  <tr>  <td> <?php echo $no++ ?> </td>  <!-- <td><?php echo $ihs->id?></td> -->  <td><?php echo $ihs->nama?></td>  <td><?php echo $ihs->username?></td>  <td><?php echo $ihs->role\_id?></td>  <!-- <td><?php echo $ihs->password?></td> -->  <!-- <td><?php echo $ihs->role\_id?></td> -->  <td>  <?= anchor('c\_data\_user/detail/'.$ihs->id,'<div class="btn btn-success btn-sm"><i class="fa fa-search-plus"></i></div>') ?>  <?= anchor('c\_data\_user/edit/'.$ihs->id, '<div class="btn btn-primary btn-sm"><i class="fa fa-edit"></i></div>') ?>  <?= anchor('c\_data\_user/hapus/'.$ihs->id,'<div class="btn btn-danger btn-sm"><i class="fa fa-trash"></i></div>') ?>  </td>  </tr>  <?php } ?>  </tbody>  </tfoot>  </table>  </div>  <!-- /.box-body -->  </div>  <!-- /.box -->  </div>  <!-- /.col -->  </div>  <!-- /.row -->  </section>  <!-- Modal -->  <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog" role="document">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title" id="exampleModalLabel">FORM INPUT DATA PEGAWAI</h4>  <button type="button" class="close" data-dismiss="modal" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <?= form\_open\_multipart('c\_data\_user/tambah\_aksi'); ?>  <div class="form-group">  <div class="form-group">  <label>Nama User</label>  <input type="text" name="nama" class="form-control" required autocomplete="off">  </div>  <div class="form-group">  <label>Username</label>  <input type="text" name="username" class="form-control" required autocomplete="off">  </div>  <div class="form-group">  <label>Password</label>  <input type="text" name="password" class="form-control" required autocomplete="off">  </div>  <div class="form-group">  <label>Role</label>  <select class="form-control" name="role\_id">  <option selected>--Pilih Role--</option>  <option>1</option>  <option>2</option>  </select>  </div>  <div class="form-group">  <button type="reset" class="btn btn-danger">Reset</button>  <button type="submit" class="btn btn-primary">Simpan</button>  <?= form\_close(); ?>  </div>  </div>  </div>  </div>  </div>  </div>  <!-- /.content -->  </div>  <!-- /.content-wrapper -->  </body> |

Listings 6. 59 Syntax Untuk View Data User.

|  |
| --- |
| <section class="content">  <div class="row">  <div class="col-xs-12">  <div class="box">  <div class="box-header">  <h3 class="box-title">Tampil Data User</h3>  </div>  <!-- /.box-header -->  <button class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"><i class="fa fa-plus"></i>Tambah Data Pegawai</button>  <div class="box-body"> |

Listings 6. 60 Syntax Untuk Membuat Judul Halaman dan Button.

*Syntax* ini berfungsi untuk membuat tampilan judul dan *button*. Jika berhasil, maka hasilnya akan seperti berikut.

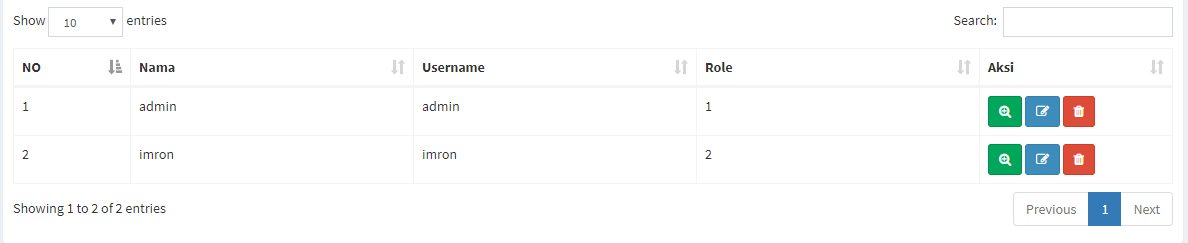


Gambar 6. 11 Hasil Dari Syntax Membuat Button.

|  |
| --- |
| <table id="example1" class="table table-bordered table-hover">  <thead>  <tr>  <th style="width : 5px">NO</th>  <th style="width : 100px">Nama</th>  <th style="width : 100px">Username</th>  <th style="width : 100px">Role</th>  <th style="width : 10px">Aksi</th>  </tr>  </thead>  <tbody>  <?php $no = 1; foreach($data\_user as $ihs){ ?>  <tr>  <td> <?php echo $no++ ?> </td>  <!-- <td><?php echo $ihs->id?></td> -->  <td><?php echo $ihs->nama?></td>  <td><?php echo $ihs->username?></td>  <td><?php echo $ihs->role\_id?></td>  <!-- <td><?php echo $ihs->password?></td> -->  <!-- <td><?php echo $ihs->role\_id?></td> -->  <td>  <?= anchor('c\_data\_user/detail/'.$ihs->id,'<div class="btn btn-success btn-sm"><i class="fa fa-search-plus"></i></div>') ?>  <?= anchor('c\_data\_user/edit/'.$ihs->id, '<div class="btn btn-primary btn-sm"><i class="fa fa-edit"></i></div>') ?>  <?= anchor('c\_data\_user/hapus/'.$ihs->id,'<div class="btn btn-danger btn-sm"><i class="fa fa-trash"></i></div>') ?>  </td>  </tr>  <?php } ?>  </tbody>  </tfoot>  </table>  </div>  <!-- /.box-body -->  </div>  <!-- /.box -->  </div>  <!-- /.col -->  </div>  <!-- /.row -->  </section> |

Listings 6. 61 Syntax Untuk Membuat Tabel.

Pada *syntax* tersebut berfungsi untuk membuat tabel untuk menampilkan data akun yang telah terdaftar pada *database*. Jika berhasil, maka tampilannya akan seperti berikut.

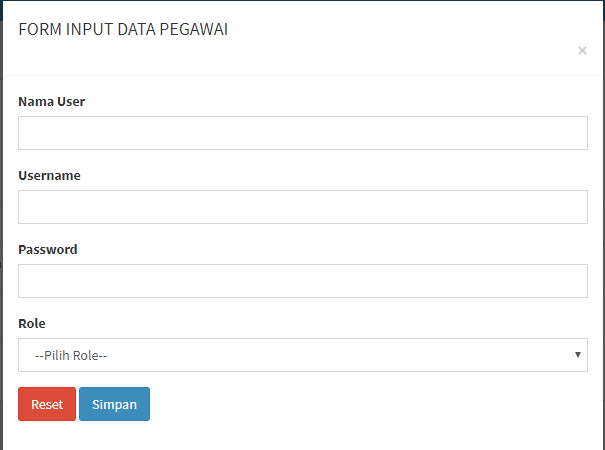


Gambar 6. 12 Hasil Dari Syntax Membuat Tabel.

|  |
| --- |
| <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog" role="document">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title" id="exampleModalLabel">FORM INPUT DATA PEGAWAI</h4>  <button type="button" class="close" data-dismiss="modal" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <?= form\_open\_multipart('c\_data\_user/tambah\_aksi'); ?>  <div class="form-group">  <div class="form-group">  <label>Nama User</label>  <input type="text" name="nama" class="form-control" required autocomplete="off">  </div>  <div class="form-group">  <label>Username</label>  <input type="text" name="username" class="form-control" required autocomplete="off">  </div>  <div class="form-group">  <label>Password</label>  <input type="text" name="password" class="form-control" required autocomplete="off">  </div>  <div class="form-group">  <label>Role</label>  <select class="form-control" name="role\_id">  <option selected>--Pilih Role--</option>  <option>1</option>  <option>2</option>  </select>  </div>  <div class="form-group">  <button type="reset" class="btn btn-danger">Reset</button>  <button type="submit" class="btn btn-primary">Simpan</button>  <?= form\_close(); ?>  </div>  </div>  </div>  </div> |

Listings 6. 62 Syntax Untuk Membuat Form Input Data User.

*Syntax* tersebut berfungsi untuk membuat *form input* data pegawai, dengan animasi *pop up* dan beberapa *button* yang berfungsi untuk me-*reset* dan menyimpan data yang telah diinputkan. Jika berhasil, maka tampilannya akan seperti berikut.



Gambar 6. 13 Hasil Dari Syntax Form Input Data Pegawai.

1. View\_detail\_user.php

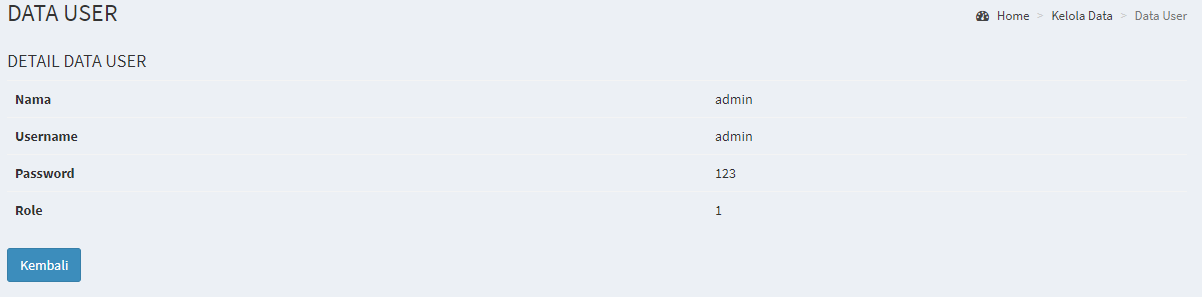
|  |
| --- |
| <div class="content-wrapper">  <!-- Content Header (Page header) -->  <section class="content-header">  <h1>  DATA USER  </h1>  <ol class="breadcrumb">  <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>  <li><a href="#">Kelola Data</a></li>  <li class="active">Data User</li>  </ol>  </section>  <section class="content">  <h4>DETAIL DATA USER</h4>  <table class="table">  <tr>  <th>Nama</th>  <td><?= $detail->nama ?></td>  </tr>  <tr>  <th>Username</th>  <td><?= $detail->username ?></td>  </tr>  <tr>  <th>Password</th>  <td><?= $detail->password ?></td>  </tr>  <tr>  <th>Role</th>  <td><?= $detail->role\_id ?></td>  </tr>  </table>  <a href="<?= base\_url('c\_data\_user/index'); ?>" class="btn btn-primary">Kembali</a>  </section>  </div> |

Listings 6. 63 Syntax Untuk Membuat Form Detail User.

|  |
| --- |
| <section class="content">  <h4>DETAIL DATA USER</h4>  <table class="table">  <tr>  <th>Nama</th>  <td><?= $detail->nama ?></td>  </tr>  <tr>  <th>Username</th>  <td><?= $detail->username ?></td>  </tr>  <tr>  <th>Password</th>  <td><?= $detail->password ?></td>  </tr>  <tr>  <th>Role</th>  <td><?= $detail->role\_id ?></td>  </tr>  </table>  <a href="<?= base\_url('c\_data\_user/index'); ?>" class="btn btn-primary">Kembali</a>  </section>  </div> |

Listings 6. 64 Function Untuk Menampilkan Detail Akun.

*Syntax* tersebut berfungsi untuk memberikan label pada *form* detail akun, agar lebih jelas informasi yang ditampilkan pada layar. Jika berhasil, maka tampilannya akan seperti berikut.



Gambar 6. 14 Hasil Dari Syntax Detail Akun Pegawai.

1. View\_edit\_user.php

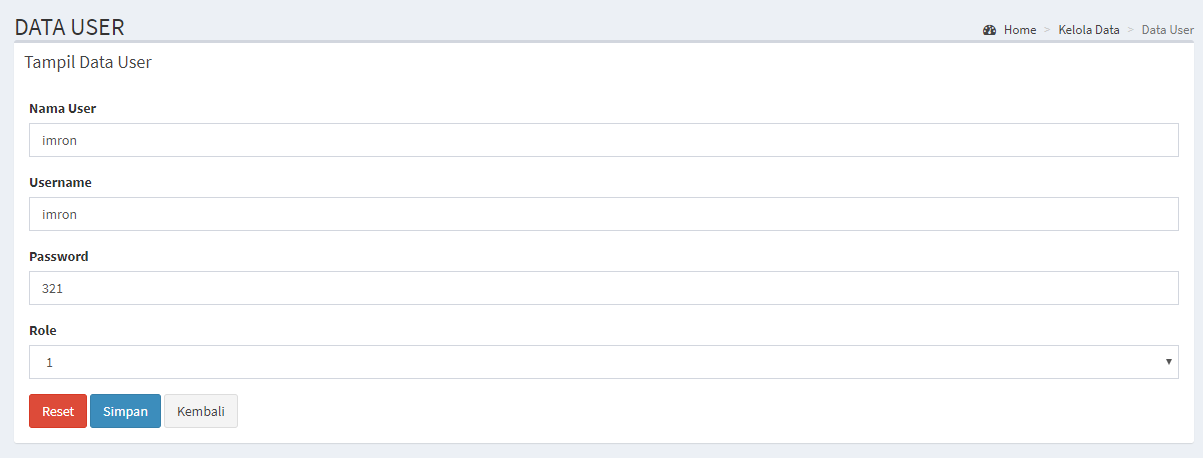
|  |
| --- |
| <div class="content-wrapper">  <section class="content-header">  <h1>  DATA USER  </h1>  <ol class="breadcrumb">  <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>  <li><a href="#">Kelola Data</a></li>  <li class="active">Data User</li>  </ol>  </section>  <div class="col-xs-12">  <div class="box">  <div class="box-header">  <h3 class="box-title">Tampil Data User</h3>  </div>  <section class="content">  <?php foreach ($data\_user as $ihs) : ?>  <?= form\_open\_multipart('c\_data\_user/update'); ?>  <div class="form-group">  <label>Nama User</label>  <input type="hidden" name="id" class="form-control" value="<?= $ihs->id ?>">  <input type="text" name="nama" class="form-control" value="<?= $ihs->nama ?>">  </div>  <div class="form-group">  <label>Username</label>  <input type="text" name="username" class="form-control" value="<?= $ihs->username ?>">  </div>  <div class="form-group">  <label>Password</label>  <input type="text" name="password" class="form-control" value="<?= $ihs->password ?>">  </div>  <div class="form-group">  <label>Role</label>  <select class="form-control" name="role\_id" value="<?= $ihs->role\_id ?>">  <option>1</option>  <option>2</option>  </select>  </div>  <button type="reset" class="btn btn-danger">Reset</button>  <button type="submit" class="btn btn-primary">Simpan</button>  <a href="<?= base\_url('c\_data\_user/index'); ?>" class="btn btn-default">Kembali</a>  <?= form\_close(); ?>  <!-- </form> -->  <?php endforeach; ?>  </section>  </div>  </div>  </div> |

Listings 6. 65 Syntax Untuk View Edit Akun.

|  |
| --- |
| <section class="content">  <?php foreach ($data\_user as $ihs) : ?>  <?= form\_open\_multipart('c\_data\_user/update'); ?>  <div class="form-group">  <label>Nama User</label>  <input type="hidden" name="id" class="form-control" value="<?= $ihs->id ?>">  <input type="text" name="nama" class="form-control" value="<?= $ihs->nama ?>">  </div>  <div class="form-group">  <label>Username</label>  <input type="text" name="username" class="form-control" value="<?= $ihs->username ?>">  </div>  <div class="form-group">  <label>Password</label>  <input type="text" name="password" class="form-control" value="<?= $ihs->password ?>">  </div>  <div class="form-group">  <label>Role</label>  <select class="form-control" name="role\_id" value="<?= $ihs->role\_id ?>">  <option>1</option>  <option>2</option>  </select>  </div>  <button type="reset" class="btn btn-danger">Reset</button>  <button type="submit" class="btn btn-primary">Simpan</button>  <a href="<?= base\_url('c\_data\_user/index'); ?>" class="btn btn-default">Kembali</a>  <?= form\_close(); ?>  <!-- </form> -->  <?php endforeach; ?>  </section> |

Listings 6. 66 Syntax Untuk Membuat Textbox Serta Button Pada Form Edit.

*Syntax* tersebut berfungsi untuk membuat *textbox* pada *form* *edit* akun pegawai. Jika kita menekan tombol *edit* pada salah satu akun yang akan diubah datanya, maka *textbox* pada *form* *edit* akun pegawai sudah terisi oleh informasi akun tersebut. Jika berhasil, maka tampilannya akan seperti berikut.



Gambar 6. 15 Hasil Dari Syntax Halaman Edit Akun Pegawai.

1. Perancangan *Form* Prediksi

Berikut ini merupakan *syntax* untuk membuat halaman prediksi beserta penerapan metode Regresi Linier Berganda.

1. C\_rlb.php

|  |
| --- |
| <?php  class C\_rlb extends CI\_Controller{  public function \_\_construct(){  parent::\_\_construct();  if($this->session->userdata('role\_id') == NULL){  $this->session->set\_flashdata('pesan','<div class="alert alert-danger" role="alert">  <strong>Anda belum Login !!!</strong>  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>');  redirect('auth/login');  }  $this->load->library('Excel');  $this->load->model('m\_rlb');  }  public function index(){  $data = [  'rlb' => $this->m\_rlb->getAll(),  ];  $this->load->view('templates/header');  $this->load->view('templates/sidebar');  $this->load->view('rlb/index', $data);  $this->load->view('templates/footer');  }  public function json(){  $data = $this->m\_rlb->getAll();  //var\_dump($query);  echo json\_encode($data);  }    public function import(){  if(isset($\_FILES["file"]["name"]))  {  $path = $\_FILES["file"]["tmp\_name"];  $object = PHPExcel\_IOFactory::load($path);  foreach($object->getWorksheetIterator() as $worksheet)  {  $highestRow = $worksheet->getHighestRow();  $highestColumn = $worksheet->getHighestColumn();  for($row=2; $row<=$highestRow; $row++)  {  $bulan = $worksheet->getCellByColumnAndRow(0, $row)->getValue();  $jumlah\_pelanggan = $worksheet->getCellByColumnAndRow(1, $row)->getValue();  $jumlah\_produk = $worksheet->getCellByColumnAndRow(2, $row)->getValue();  $pendapatan = $worksheet->getCellByColumnAndRow(3, $row)->getValue();    $cek\_duplicat = $this->m\_rlb->chek\_duplicat($bulan);  if ($cek\_duplicat != NULL) {  if ($cek\_duplicat->bulan == $bulan) {  $this->m\_rlb->update\_duplicat($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan);  }else{  $this->m\_rlb->upload($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan);  }  }else{  $this->m\_rlb->upload($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan);  }      }  }  redirect('C\_rlb');  }  }  } |

Listings 6. 67 Syntax Controller Untuk Penerapan Regresi Linier Berganda.

|  |
| --- |
| public function json(){  $data = $this->m\_rlb->getAll();  //var\_dump($query);  echo json\_encode($data);  } |

Listings 6. 68 Function Untuk Menampilkan Data Secara Keseluruhan Dari Database.

*Syntax* tersebut berfungsi untuk mengambil seluruh data indihome yang telah di-*upload* dan tersimpan pada *database*.

|  |
| --- |
| public function import(){  if(isset($\_FILES["file"]["name"]))  {  $path = $\_FILES["file"]["tmp\_name"];  $object = PHPExcel\_IOFactory::load($path);  foreach($object->getWorksheetIterator() as $worksheet)  {  $highestRow = $worksheet->getHighestRow();  $highestColumn = $worksheet->getHighestColumn();  for($row=2; $row<=$highestRow; $row++)  {  $bulan = $worksheet->getCellByColumnAndRow(0, $row)->getValue();  $jumlah\_pelanggan = $worksheet->getCellByColumnAndRow(1, $row)->getValue();  $jumlah\_produk = $worksheet->getCellByColumnAndRow(2, $row)->getValue();  $pendapatan = $worksheet->getCellByColumnAndRow(3, $row)->getValue();    $cek\_duplicat = $this->m\_rlb->chek\_duplicat($bulan);  if ($cek\_duplicat != NULL) {  if ($cek\_duplicat->bulan == $bulan) {  $this->m\_rlb->update\_duplicat($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan);  }else{  $this->m\_rlb->upload($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan);  }  }else{  $this->m\_rlb->upload($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan);  }    }  }  redirect('C\_rlb');  }  }  } |

Listings 6. 69 Function Untuk Import Data Pada Sistem.

*Syntax* tersebut berfungsi untuk melakukan *import* data pada sistem dengan format *xlsx,* dan *xls.*

1. *M\_rlb.php*

|  |
| --- |
| <?php  class M\_rlb extends CI\_Model  {  public function \_\_construct(){  parent::\_\_construct();  }  public function getAll(){  $this->db->from('data\_rlb');  $this->db->order\_by('bulan','ASC');  return $this->db->get()->result();  }  public function count(){  return $this->db->count\_all\_results('data\_rlb');  }  public function chek\_duplicat($id){  $QuerySql = $this->db->query("SELECT bulan FROM data\_rlb WHERE bulan IN ('$id')");  return $QuerySql->row();  }  public function update\_duplicat($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan) {  $this->db->query('UPDATE data\_rlb SET jumlah\_pelanggan = "'.$jumlah\_pelanggan.'",  jumlah\_produk = "'.$jumlah\_produk.'",  pendapatan = "'.$pendapatan.'" WHERE bulan = "'.$bulan.'"');  }  public function upload($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan)  {  $this->db->query('REPLACE INTO data\_rlb (bulan, jumlah\_pelanggan, jumlah\_produk, pendapatan)  VALUES("'.$bulan.'",  "'.$jumlah\_pelanggan.'",  "'.$jumlah\_produk.'",  "'.$pendapatan.'")  ');  }  public function matrixdet($data){  $rumus = ($data[0][0] \* $data[1][1] \* $data[2][2]) + ($data[0][1] \* $data[1][2] \* $data[2][0]) + ($data[0][2] \* $data[1][0] \* $data[2][1]) - ($data[0][1] \* $data[1][0] \* $data[2][2]) - ($data[0][0] \* $data[1][2] \* $data[2][1]) - ($data[0][2] \* $data[1][1] \* $data[2][0]);  return $rumus;  }  }  ?> |

Listings 6. 70 Syntax Untuk Model Pada Penerapan Regresi Linier Berganda.

|  |
| --- |
| public function count(){  return $this->db->count\_all\_results('data\_rlb');  }  public function chek\_duplicat($id){  $QuerySql = $this->db->query("SELECT bulan FROM data\_rlb WHERE bulan IN ('$id')");  return $QuerySql->row();  }  public function update\_duplicat($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan) {  $this->db->query('UPDATE data\_rlb SET jumlah\_pelanggan = "'.$jumlah\_pelanggan.'",  jumlah\_produk = "'.$jumlah\_produk.'",  pendapatan = "'.$pendapatan.'" WHERE bulan = "'.$bulan.'"');  }  public function upload($bulan, $jumlah\_pelanggan, $jumlah\_produk, $pendapatan)  {  $this->db->query('REPLACE INTO data\_rlb (bulan, jumlah\_pelanggan, jumlah\_produk, pendapatan)  VALUES("'.$bulan.'",  "'.$jumlah\_pelanggan.'",  "'.$jumlah\_produk.'",  "'.$pendapatan.'")  ');  } |

Listings 6. 71 Syntax Untuk Cek Duplikasi dan Update Data.

*Syntax* tersebut berfungsi untuk melakukan perhitungan jumlah data yang tersimpan pada *database* dan akan ditampilkan pada sistem. Fungsi dari cek duplikasi untuk melihat data yang sudah disimpan pada *database* dengan data yang akan di *upload*, agar tidak terjadi redudansi data atau pengulangan data yang sama.

|  |
| --- |
| public function matrixdet($data){  $rumus = ($data[0][0] \* $data[1][1] \* $data[2][2]) + ($data[0][1] \* $data[1][2] \* $data[2][0]) + ($data[0][2] \* $data[1][0] \* $data[2][1]) - ($data[0][1] \* $data[1][0] \* $data[2][2]) - ($data[0][0] \* $data[1][2] \* $data[2][1]) - ($data[0][2] \* $data[1][1] \* $data[2][0]);  return $rumus;  } |

Listings 6. 72 Syntax Untuk Melakukan Perhitungan Matrix.

*Syntax* tersebut berfungsi untuk membuat perhitungan dengan rumus matrix.

1. Index.php

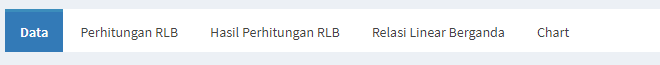
|  |
| --- |
| <style type="text/css">  th{  text-align: center;  background-color: black;  color: white;  }  </style>  <!-- Content Wrapper. Contains page content -->  <div class="content-wrapper">  <!-- Content Header (Page header) -->  <section class="content-header">  <ol class="breadcrumb">  <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>  <li class="active">Data RLB</li>  </ol>  </section>  <!-- Main content -->  <section class="content">  <div class="row">  <div class="col-xs-12">  <div class="card">  <div class="card-header d-flex p-0">  <h3 class="card-title p-3">Data RLB</h3>  <ul class="nav nav-pills ml-auto p-2" style="background-color: white;">  <li class="nav-item active"><a class="nav-link active" href="#tab\_1" data-toggle="tab">Data</a></li>  <li class="nav-item"><a class="nav-link" href="#tab\_2" data-toggle="tab">Perhitungan RLB</a></li>  <li class="nav-item"><a class="nav-link" href="#tab\_3" data-toggle="tab">Hasil Perhitungan RLB</a></li>  <li class="nav-item"><a class="nav-link" href="#tab\_y" data-toggle="tab">Relasi Linear Berganda</a></li>  <li class="nav-item"><a class="nav-link" href="#chart" data-toggle="tab">Chart</a></li>  </ul>  </div><!-- /.card-header -->  <br/>  <div class="card-body">  <div class="tab-content" style="background-color: white;">  <div class="tab-pane active" id="tab\_1">  <div>  <button class="btn btn-primary" style="margin: 10px 10px 10px 10px;" data-toggle="modal" data-target="#exampleModal"><i class="fa fa-plus"></i> Upload Data</button>  </div>  <table class="table table-bordered">  <thead>  <th>No</th>  <th>Bulan</th>  <th>Jumlah Pelanggan</th>  <th>Jumlah Produk</th>  <th>Pendapatan</th>  </thead>  <tbody>  <?php  $n = 1;  $pelanggan = 0;  $produk = 0;  $pendapatan = 0;  foreach ($rlb as $data) {  ?>  <tr>  <td><?php echo $n++; ?></td>  <td><?php echo date('M-Y', ($data->bulan - 25569) \* 86400); ?></td>  <td><?php echo $x1 = $data->jumlah\_pelanggan; ?></td>  <td><?php echo $x2 = $data->jumlah\_produk; ?></td>  <td><?php echo $y = $data->pendapatan; ?></td>  </tr>  <?php  $pelanggan = $x1+$pelanggan;  $produk = $x2 + $produk;  $pendapatan = $y + $pendapatan;  }  ?>  </tbody>  <tfoot>  <th colspan="2">n = <?php echo $n-1; ?></th>  <th><?php echo $pelanggan; ?></th>  <th><?php echo $produk; ?></th>  <th><?php echo $pendapatan; ?></th>  </tfoot>  </table>  </div>  <!-- /.tab-pane -->  <div class="tab-pane" id="tab\_2">  <table class="table table-bordered">  <thead>  <th>No</th>  <th>Bulan</th>  <th>Jumlah Pelanggan (X1)</th>  <th>Jumlah Produk (X2)</th>  <th>Pendapatan (Y)</th>  <th>Y^2</th>  <th>X1^2</th>  <th>X2^2</th>  <th>X1.Y</th>  <th>X2.Y</th>  <th>X1.X2</th>  </thead>  <tbody>  <?php  $n = 1;  $pelanggan = 0;  $produk = 0;  $pendapatan = 0;  $toty2 = 0;  $totx12 = 0;  $totx22 = 0;  $totx1xy = 0;  $totx2xy = 0;  $totx1x2 = 0;  foreach ($rlb as $data) {  ?>  <tr>  <td><?php echo $n++; ?></td>  <td><?php echo date('M-Y', ($data->bulan - 25569) \* 86400); ?></td>  <td><?php echo number\_format($x1 = $data->jumlah\_pelanggan); ?></td>  <td><?php echo number\_format($x2 = $data->jumlah\_produk); ?></td>  <td><?php echo number\_format($y = $data->pendapatan); ?></td>  <td><?php echo number\_format($y2 = pow($y, 2)); ?></td>  <td><?php echo number\_format($x12 = pow($x1, 2)); ?></td>  <td><?php echo number\_format($x22 = pow($x2, 2)); ?></td>  <td><?php echo number\_format($x1xy = $x1\*$y); ?></td>  <td><?php echo number\_format($x2xy = $x2\*$y); ?></td>  <td><?php echo number\_format($x1x2 = $x1\*$x2); ?></td>  </tr>  <?php  $pelanggan = $x1+$pelanggan;  $produk = $x2 + $produk;  $pendapatan = $y + $pendapatan;  $toty2 = $y2 + $toty2;  $totx12 = $x12 + $totx12;  $totx22 = $x22 + $totx22;  $totx1xy = $x1xy + $totx1xy;  $totx2xy = $x2xy + $totx2xy;  $totx1x2 = $x1x2 + $totx1x2;  }  ?>  </tbody>  <tfoot>  <th colspan="2">n = <?php echo $n = $n-1; ?></th>  <th><?php echo number\_format($pelanggan); ?></th>  <th><?php echo number\_format($produk); ?></th>  <th><?php echo number\_format($pendapatan); ?></th>  <th><?php echo number\_format($toty2); ?></th>  <th><?php echo number\_format($totx12); ?></th>  <th><?php echo number\_format($totx22); ?></th>  <th><?php echo number\_format($totx1xy); ?></th>  <th><?php echo number\_format($totx2xy); ?></th>  <th><?php echo number\_format($totx1x2); ?></th>  </tfoot>  </table>  </div>  <!-- /.tab-pane -->  <div class="tab-pane" id="tab\_3">  <?php  //rumus tahap2  $sigmax12 = (pow($pelanggan, 2)/$n) - $totx12;  $sigmax22 = (pow($produk, 2)/$n) - $totx22;  $sigmay2 = (pow($pendapatan, 2)/$n) - $toty2;  $sigmax1y = (($pelanggan \* $pendapatan)/$n) - $totx1xy;  $sigmax2y = (($produk\*$pendapatan)/$n) - $totx2xy;  $sigmax1x2 = (($pelanggan \* $produk)/$n) - $totx1x2;  $matrixdeta = array(  array($n, $pelanggan, $produk),  array($pelanggan, $totx12, $totx1x2),  array($produk, $totx1x2, $totx22)  );  $deta = $this->m\_rlb->matrixdet($matrixdeta);  //echo "<script>alert('".$deta."')</script>";  $matrixdeta1 = array(  array($pendapatan, $pelanggan, $produk),  array($totx1xy, $totx12, $totx1x2),  array($totx2xy, $totx1x2, $totx22)  );  $deta1 = $this->m\_rlb->matrixdet($matrixdeta1);  //echo "<script>alert('".number\_format($deta1)."')</script>";  $matrixdeta2 = array(  array($n, $pendapatan, $produk),  array($pelanggan, $totx1xy, $totx1x2),  array($produk, $totx2xy, $totx22)  );  $deta2 = $this->m\_rlb->matrixdet($matrixdeta2);  //echo "<script>alert('".number\_format($deta2)."')</script>";  $matrixdeta3 = array(  array($n, $pelanggan, $pendapatan),  array($pelanggan, $totx12, $totx1xy),  array($produk, $totx1x2, $totx2xy)  );  $deta3 = $this->m\_rlb->matrixdet($matrixdeta3);  //echo "<script>alert('".number\_format($deta3)."')</script>";  $a = $deta1/$deta;  $b1 = $deta2/$deta;  $b2 = $deta3/$deta;  ?>  <table class="table">  <thead>  <th>Nama</th>  <th>Nilai</th>  </thead>  <tbody>  <!-- <tr>  <td>Sigma X1^2</td>  <td><?php echo number\_format($sigmax12); ?></td>  </tr>  <tr>  <td>Sigma X2^2</td>  <td><?php echo number\_format($sigmax22); ?></td>  </tr>  <tr>  <td>Sigma Y^2</td>  <td><?php echo number\_format($sigmay2); ?></td>  </tr>  <tr>  <td>Sigma X1\*Y</td>  <td><?php echo number\_format($sigmax1y); ?></td>  </tr>  <tr>  <td>Sigma X2^Y</td>  <td><?php echo number\_format($sigmax2y); ?></td>  </tr>  <tr>  <td>Sigma X1\*X2</td>  <td><?php echo number\_format($sigmax1x2); ?></td>  </tr> -->  <tr>  <td>Det(A)</td>  <td><?php echo number\_format($deta); ?></td>  </tr>  <tr>  <td>Det(A1)</td>  <td><?php echo number\_format($deta1); ?></td>  </tr>  <tr>  <td>Det(A2)</td>  <td><?php echo number\_format($deta2); ?></td>  </tr>  <tr>  <td>Det(A3)</td>  <td><?php echo number\_format($deta3); ?></td>  </tr>  <tr>  <td>a</td>  <td><?php echo $a; ?></td>  </tr>  <tr>  <td>B1</td>  <td><?php echo $b1; ?></td>  </tr>  <tr>  <td>B2</td>  <td><?php echo $b2; ?></td>  </tr>  </tbody>  </table>  </div>  <div class="tab-pane" id="tab\_y">  <table class="table">  <tr>  <td>Masukan nilai X1 <input type="number" name="nilaix1" id="nilaix1" value="0" class="form-control"></td>  </tr>  <tr>  <td>Masukan nilai X2 <input type="number" name="nilaix2" id="nilaix2" value="0" class="form-control"></td>  </tr>  <tr>  <td>  <!-- <img src="<?php echo base\_url('assets/img/rumusY.png') ?>"> -->  <a onclick="munculY()" class="btn btn-warning">Lihat Hasil dari Y</a>  <input type="text" disabled id="output" value="0" class="form-control">  <script>  function munculY(){  var a = Number('<?php echo $a ?>');  var b1 = Number('<?php echo $b1 ?>');  var b2 = Number('<?php echo $b2 ?>');  var nilaix1 = Number(document.getElementById("nilaix1").value);  var nilaix2 = Number(document.getElementById("nilaix2").value);  rumus = a+(b1\*nilaix1)+(b2\*nilaix2);  //rumus = nilaix2+nilaix1;  document.getElementById("output").value = parseFloat(rumus);  }  </script>  </td>  </tr>  <tr>  <td>  <!-- <img src="<?php echo base\_url('assets/img/rumusSeSyx.png') ?>"> -->  <a onclick="munculSeSyx()" class="btn btn-warning">Lihat Standart Error Estimate</a>  <input type="text" disabled id="output2" value="0" class="form-control">  <script>  function munculSeSyx(){  var sigmay2 = Number('<?php echo $toty2 ?>');  var a = Number('<?php echo $a ?>');  var pendapatan = Number('<?php echo $pendapatan ?>');  var b1 = Number('<?php echo $b1 ?>');  var sigmax1y = Number('<?php echo $sigmax1y ?>');  var b2 = Number('<?php echo $b2 ?>');  var sigmax2y = Number('<?php echo $sigmax2y ?>');  var n = Number('<?php echo $n ?>');  var rumus = (sigmay2-(a\*pendapatan)-(b1\*sigmax1y)-(b2\*sigmax2y))/n-3;  document.getElementById("output2").value = Math.sqrt(parseFloat(rumus))/10000;  }  </script>  </td>  </tr>  </table>  </div>  <div class="tab-pane" id="chart">  <div class="container">  <br />  <div class="row">  <div class="col-md-1"></div>  <div class="col-md-10">  <!-- Chart.js Canvas Tag -->  <canvas id="barChart"></canvas>  </div>  <div class="col-md-1"></div>  </div>  <div class="col-md-1"></div>  <div class="col-md-10">  <canvas id="barChart1"></canvas>  </div>  <div class="col-md-1"></div>  <div class="col-md-10">  <canvas id="barChart2"></canvas>  </div>  </div>  </div>  <!-- /.tab-pane -->  </div>  <!-- /.tab-content -->  </div><!-- /.card-body -->  </div>  <!-- ./card -->  </div>  <!-- /.col -->  </div>  <!-- /.row -->  </section>  <!-- /.content -->  </div>  <!-- /.content-wrapper -->    </body>  <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">  <div class="modal-dialog" role="document">  <div class="modal-content">  <div class="modal-header">  <h4 class="modal-title" id="exampleModalLabel">FORM UPLOAD DATA RLB</h4>  <button type="button" class="close" data-dismiss="modal" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <?= form\_open\_multipart('c\_rlb/import'); ?>  <form method="post" action="<?= base\_url() . 'c\_rlb/import'; ?>">  <div class="form-group">  <label for="inputEmail3" class="control-label">Lampirkan File</label>  <input type="file" name="file" class="form-control" id="file" required accept=".xls, .xlsx, .csv" /></p>  <p class="help-block">Format file yang didukung .xls, .xlsx atau .csv</p>  </div>  <div class="form-group">  <input type="submit" class="btn btn-primary" value="Upload" name="upload" id="upload">  </div>  <?= form\_close(); ?>  </div>  </div>  </div>  </div>  <script src="<?php echo base\_url()?>assets/chartjs/Chart.js"></script>  <?php  $this->load->view('rlb/chart');  ?> |

Listings 6. 73 Syntax Lengkap Controller Untuk Penerapan Regresi Linier Berganda.

|  |
| --- |
| <section class="content">  <div class="row">  <div class="col-xs-12">  <div class="card">  <div class="card-header d-flex p-0">  <h3 class="card-title p-3">Data RLB</h3>  <ul class="nav nav-pills ml-auto p-2" style="background-color: white;">  <li class="nav-item active"><a class="nav-link active" href="#tab\_1" data-toggle="tab">Data</a></li>  <li class="nav-item"><a class="nav-link" href="#tab\_2" data-toggle="tab">Perhitungan RLB</a></li>  <li class="nav-item"><a class="nav-link" href="#tab\_3" data-toggle="tab">Hasil Perhitungan RLB</a></li>  <li class="nav-item"><a class="nav-link" href="#tab\_y" data-toggle="tab">Relasi Linear Berganda</a></li>  <li class="nav-item"><a class="nav-link" href="#chart" data-toggle="tab">Chart</a></li>  </ul>  </div><!-- /.card-header --> |

Listings 6. 74 Syntax Untuk Membuat Section Tabel Pada Form RLB.

*Syntax* tersebut berfungsi untuk membuat tab pada sistem. Jika berhasil, maka tampilannya akan seperti berikut.

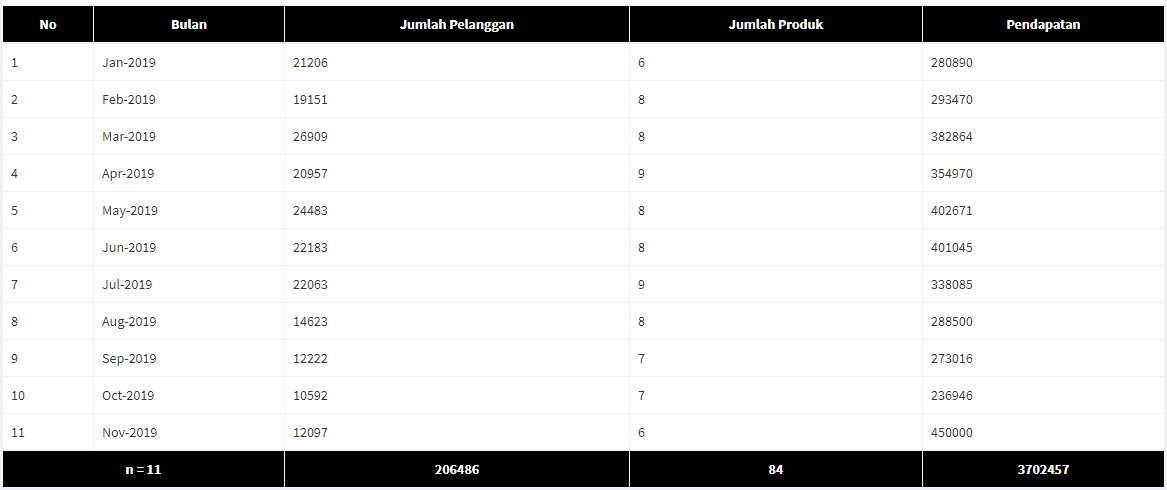


Gambar 6. 16 Hasil Syntax Untuk Membuat Tab.

|  |
| --- |
| <div class="card-body">  <div class="tab-content" style="background-color: white;">  <div class="tab-pane active" id="tab\_1">  <div>  <button class="btn btn-primary" style="margin: 10px 10px 10px 10px;" data-toggle="modal" data-target="#exampleModal"><i class="fa fa-plus"></i> Upload Data</button>  </div>  <table class="table table-bordered">  <thead>  <th>No</th>  <th>Bulan</th>  <th>Jumlah Pelanggan</th>  <th>Jumlah Produk</th>  <th>Pendapatan</th>  </thead>  <tbody>  <?php  $n = 1;  $pelanggan = 0;  $produk = 0;  $pendapatan = 0;  foreach ($rlb as $data) {  ?>  <tr>  <td><?php echo $n++; ?></td>  <td><?php echo date('M-Y', ($data->bulan - 25569) \* 86400); ?></td>  <td><?php echo $x1 = $data->jumlah\_pelanggan; ?></td>  <td><?php echo $x2 = $data->jumlah\_produk; ?></td>  <td><?php echo $y = $data->pendapatan; ?></td>  </tr>  <?php  $pelanggan = $x1+$pelanggan;  $produk = $x2 + $produk;  $pendapatan = $y + $pendapatan;  }  ?>  </tbody> |

Listings 6. 75 Syntax Untuk Membuat Tabel Serta Menampilkan Data.

*Syntax* tersebut berfungsi untuk membuat tabel pada tab pertama, dan menampilkan keseluruhan data indieland pada *database*. Jika berhasil, maka tampilannya akan seperti berikut.

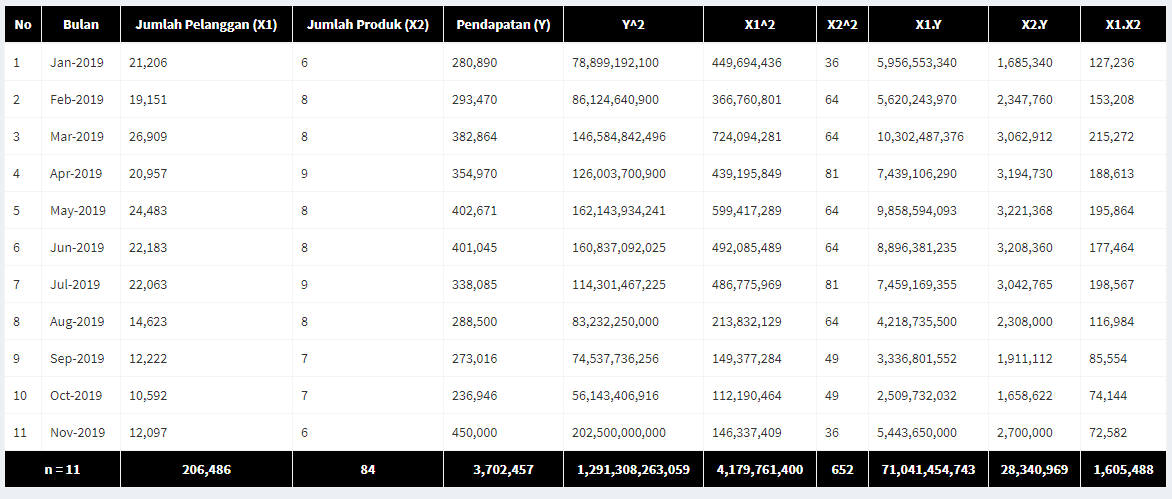


Gambar 6. 17 Hasil Dari Tabel Pada Tab Pertama.

|  |
| --- |
| <tfoot>  <th colspan="2">n = <?php echo $n-1; ?></th>  <th><?php echo $pelanggan; ?></th>  <th><?php echo $produk; ?></th>  <th><?php echo $pendapatan; ?></th>  </tfoot>  </table>  </div>  <!-- /.tab-pane -->  <div class="tab-pane" id="tab\_2">  <table class="table table-bordered">  <thead>  <th>No</th>  <th>Bulan</th>  <th>Jumlah Pelanggan (X1)</th>  <th>Jumlah Produk (X2)</th>  <th>Pendapatan (Y)</th>  <th>Y^2</th>  <th>X1^2</th>  <th>X2^2</th>  <th>X1.Y</th>  <th>X2.Y</th>  <th>X1.X2</th>  </thead>  <tbody>  <?php  $n = 1;  $pelanggan = 0;  $produk = 0;  $pendapatan = 0;  $toty2 = 0;  $totx12 = 0;  $totx22 = 0;  $totx1xy = 0;  $totx2xy = 0;  $totx1x2 = 0;  foreach ($rlb as $data) {  ?>  <tr>  <td><?php echo $n++; ?></td>  <td><?php echo date('M-Y', ($data->bulan - 25569) \* 86400); ?></td>  <td><?php echo number\_format($x1 = $data->jumlah\_pelanggan); ?></td>  <td><?php echo number\_format($x2 = $data->jumlah\_produk); ?></td>  <td><?php echo number\_format($y = $data->pendapatan); ?></td>  <td><?php echo number\_format($y2 = pow($y, 2)); ?></td>  <td><?php echo number\_format($x12 = pow($x1, 2)); ?></td>  <td><?php echo number\_format($x22 = pow($x2, 2)); ?></td>  <td><?php echo number\_format($x1xy = $x1\*$y); ?></td>  <td><?php echo number\_format($x2xy = $x2\*$y); ?></td>  <td><?php echo number\_format($x1x2 = $x1\*$x2); ?></td>  </tr> |

Listings 6. 76 Syntax Untuk Membuat Tabel Pada Tab Kedua.

*Syntax* tersebut berfungsi untuk membuat tabel pada tab kedua, tabel kedua ini diberikan beberapa fungsi perhitungan Regresi Linier Berganda, yakni untuk mencari nilai X1X2, Y2, dan lain sebagainya. Sehingga hasil tampilannya akan seperti berikut.

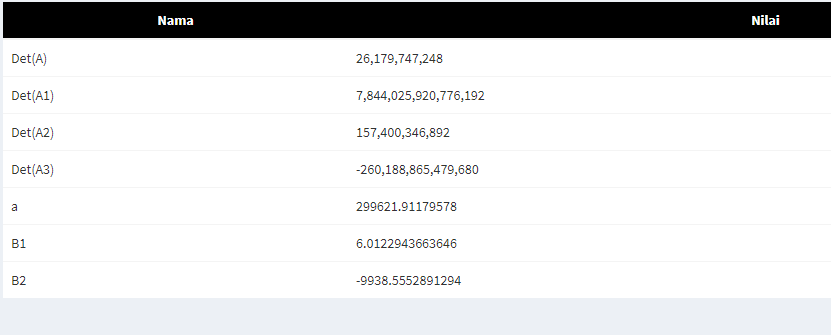


Gambar 6. 18 Syntax Untuk Tabel Kedua dan Perhitungan RLB.

|  |
| --- |
| <div class="tab-pane" id="tab\_3">  <?php  //rumus tahap2  $sigmax12 = (pow($pelanggan, 2)/$n) - $totx12;  $sigmax22 = (pow($produk, 2)/$n) - $totx22;  $sigmay2 = (pow($pendapatan, 2)/$n) - $toty2;  $sigmax1y = (($pelanggan \* $pendapatan)/$n) - $totx1xy;  $sigmax2y = (($produk\*$pendapatan)/$n) - $totx2xy;  $sigmax1x2 = (($pelanggan \* $produk)/$n) - $totx1x2;  $matrixdeta = array(  array($n, $pelanggan, $produk),  array($pelanggan, $totx12, $totx1x2),  array($produk, $totx1x2, $totx22)  );  $deta = $this->m\_rlb->matrixdet($matrixdeta);  //echo "<script>alert('".$deta."')</script>";  $matrixdeta1 = array(  array($pendapatan, $pelanggan, $produk),  array($totx1xy, $totx12, $totx1x2),  array($totx2xy, $totx1x2, $totx22)  );  $deta1 = $this->m\_rlb->matrixdet($matrixdeta1);  //echo "<script>alert('".number\_format($deta1)."')</script>";  $matrixdeta2 = array(  array($n, $pendapatan, $produk),  array($pelanggan, $totx1xy, $totx1x2),  array($produk, $totx2xy, $totx22)  );  $deta2 = $this->m\_rlb->matrixdet($matrixdeta2);  //echo "<script>alert('".number\_format($deta2)."')</script>";  $matrixdeta3 = array(  array($n, $pelanggan, $pendapatan),  array($pelanggan, $totx12, $totx1xy),  array($produk, $totx1x2, $totx2xy)  );  $deta3 = $this->m\_rlb->matrixdet($matrixdeta3);  //echo "<script>alert('".number\_format($deta3)."')</script>";  $a = $deta1/$deta;  $b1 = $deta2/$deta;  $b2 = $deta3/$deta;  ?> |

Listings 6. 77 Syntax Untuk Menampilkan Perhitungan Determinan.

Pada tabel ketiga ini berfungsi untuk menampilkna nilai dari perhitungan determinan dan perhitungan untuk mencari nilai a, b1, dan b2. Jika berhasil, maka tampilannya akan seperti berikut.

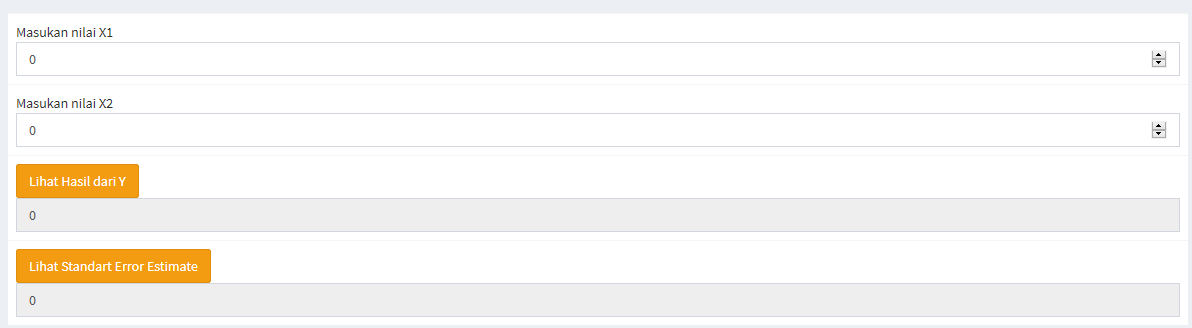


Gambar 6. 19 Hasil Dari Syntax Untuk Menampilkan Nilai Determinan, a, b1, dan b2.

|  |
| --- |
| <div class="tab-pane" id="tab\_y">  <table class="table">  <tr>  <td>Masukan nilai X1 <input type="number" name="nilaix1" id="nilaix1" value="0" class="form-control"></td>  </tr>  <tr>  <td>Masukan nilai X2 <input type="number" name="nilaix2" id="nilaix2" value="0" class="form-control"></td>  </tr>  <tr>  <td>  <!-- <img src="<?php echo base\_url('assets/img/rumusY.png') ?>"> -->  <a onclick="munculY()" class="btn btn-warning">Lihat Hasil dari Y</a>  <input type="text" disabled id="output" value="0" class="form-control">  <script>  function munculY(){  var a = Number('<?php echo $a ?>');  var b1 = Number('<?php echo $b1 ?>');  var b2 = Number('<?php echo $b2 ?>');  var nilaix1 = Number(document.getElementById("nilaix1").value);  var nilaix2 = Number(document.getElementById("nilaix2").value);  rumus = a+(b1\*nilaix1)+(b2\*nilaix2);  //rumus = nilaix2+nilaix1;  document.getElementById("output").value = parseFloat(rumus);  }  </script>  </td>  </tr> |

Listings 6. 78 Syntax Untuk Menghitung Menggunakan Rumus RLB.

*Syntax* tersebut berfungsi untuk menerapkan metode regresi linier berganda pada sistem prospek *sales*, data yang dipakai pada perhitungan ini terdapat pada tab kedua dan tab ketiga, yakni nilai X1, X2, a, b1, dan b2. Kemudian nilai-nilai tersebut dihitung menggunakan rumus regresi linier berganda, untuk menentukan prediksi pendapatan pada bulan berikutnya atau yang akan datang. Berikut merupakan hasil dari *syntax* tersebut.



Gambar 6. 20 Hasil Dari Syntax Untuk Menghitung Dalam RLB.

|  |
| --- |
| <script type="text/javascript">  var canvas = document.getElementById("barChart");  var ctx = canvas.getContext('2d');  // Global Options:  Chart.defaults.global.defaultFontColor = 'black';  Chart.defaults.global.defaultFontSize = 16;  var data = {  labels: [  <?php  foreach ($rlb as $data) {  echo '"'.date('M-Y', ($data->bulan - 25569) \* 86400).'",';  }  ?>  ],  datasets: [{  label: "Jumlah Pelanggan",  fill: false,  lineTension: 0.1,  //backgroundColor: "rgba(225,0,0,0.4)",  borderColor: "red", // The main line color  borderCapStyle: 'square',  borderDash: [], // try [5, 15] for instance  borderDashOffset: 0.0,  borderJoinStyle: 'miter',  pointBorderColor: "black",  pointBackgroundColor: "white",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "yellow",  pointHoverBorderColor: "brown",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: true  data: [<?php  foreach ($rlb as $data) {  echo $data->jumlah\_pelanggan.',';  }  ?>],  spanGaps: true,  }, {  label: "Jumlah Produk",  fill: true,  lineTension: 0.1,  //backgroundColor: "rgba(167,105,0,0.4)",  borderColor: "blue",  borderCapStyle: 'butt',  borderDash: [],  borderDashOffset: 0.0,  borderJoinStyle: 'miter',  pointBorderColor: "white",  pointBackgroundColor: "black",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "brown",  pointHoverBorderColor: "yellow",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: false  data: [  <?php  foreach ($rlb as $data) {  echo $data->jumlah\_produk.',';  }  ?>  ],  spanGaps: true,  }, {  label: "Pendapatan",  fill: true,  lineTension: 0.1,  //backgroundColor: "rgba(167,105,0,0.4)",  borderColor: "green",  borderCapStyle: 'butt',  borderDash: [],  borderDashOffset: 0.0,  borderJoinStyle: 'miter',  pointBorderColor: "white",  pointBackgroundColor: "black",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "brown",  pointHoverBorderColor: "yellow",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: false  data: [  <?php  foreach ($rlb as $data) {  echo $data->pendapatan.',';  }  ?>  ],  spanGaps: true,  }  ]  };  // Notice the scaleLabel at the same level as Ticks  var options = {  responsive: true,  title: {  display: true,  text: 'Relasi Linear Berganda'  },  scales: {  yAxes: [{  ticks: {  beginAtZero:true  },  scaleLabel: {  display: true,  labelString: 'Relasi Linear Berganda',  fontSize: 20  }  }]  }  };  // Chart declaration:  var myBarChart = new Chart(ctx, {  type: 'line',  data: data,  options: options  });  </script>  <!-- barchart1 -->  <script type="text/javascript">  var canvas = document.getElementById("barChart1");  var ctx = canvas.getContext('2d');  // Global Options:  Chart.defaults.global.defaultFontColor = 'black';  Chart.defaults.global.defaultFontSize = 16;  var data = {  labels: [  <?php  foreach ($rlb as $data) {  echo $data->jumlah\_pelanggan.",";  }  ?>  ],  datasets: [{  label: "Pendapatan",  fill: false,  lineTension: 0.1,  backgroundColor: "rgba(225,0,0,0.4)",  borderColor: "red", // The main line color  borderCapStyle: 'square',  borderDash: [], // try [5, 15] for instance  borderDashOffset: 0.0,  borderJoinStyle: 'miter',  pointBorderColor: "black",  pointBackgroundColor: "white",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "yellow",  pointHoverBorderColor: "brown",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: true  data: [<?php  foreach ($rlb as $data) {  echo $data->pendapatan.',';  }  ?>],  spanGaps: true,  }  ]  };  // Notice the scaleLabel at the same level as Ticks  var options = {  responsive: true,  title: {  display: true,  text: 'Jumlah Pelanggan'  },  scales: {  yAxes: [{  ticks: {  beginAtZero:true  },  scaleLabel: {  display: true,  labelString: 'Relasi Linear Berganda',  fontSize: 20  }  }]  }  };  // Chart declaration:  var myBarChart = new Chart(ctx, {  type: 'line',  data: data,  options: options  });  </script>  <!-- barchart2 -->  <script type="text/javascript">  var canvas = document.getElementById("barChart2");  var ctx = canvas.getContext('2d');  // Global Options:  Chart.defaults.global.defaultFontColor = 'black';  Chart.defaults.global.defaultFontSize = 16;  var data = {  labels: [  <?php  foreach ($rlb as $data) {  echo $data->jumlah\_produk.",";  }  ?>  ],  datasets: [{  label: "Pendapatan",  fill: false,  lineTension: 0.1,  backgroundColor: "rgba(0, 0, 255, 0.4)",  borderColor: "blue", // The main line color  borderCapStyle: 'square',  borderDash: [], // try [5, 15] for instance  borderDashOffset: 1.0,  borderJoinStyle: 'miter',  pointBorderColor: "black",  pointBackgroundColor: "white",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "yellow",  pointHoverBorderColor: "brown",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: true  data: [<?php  foreach ($rlb as $data) {  echo $data->pendapatan.',';  }  ?>],  spanGaps: true,  }  ]  };  // Notice the scaleLabel at the same level as Ticks  var options = {  responsive: true,  title: {  display: true,  text: 'Jumlah Produk'  },  scales: {  yAxes: [{  ticks: {  beginAtZero:true,  },  scaleLabel: {  display: true,  labelString: 'Relasi Linear Berganda',  fontSize: 20  }  }]  }  };  // Chart declaration:  var myBarChart = new Chart(ctx, {  type: 'line',  data: data,  options: options  });  </script> |

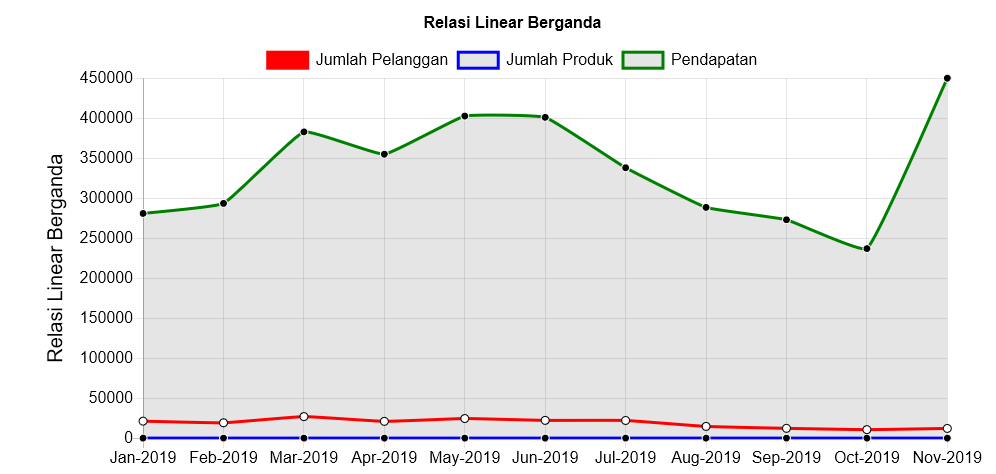
Listings 6. 79 Syntax Untuk Menampilkan Chart.

Pada *syntax* tersebut diperuntukkan informasi menganai data pelanggan yang ditampilkan dalam bentuk *chart*. Berikut merupakan beberapa penjelasan dari *syntax* tersebut.

|  |
| --- |
| <script type="text/javascript">  var canvas = document.getElementById("barChart");  var ctx = canvas.getContext('2d');  // Global Options:  Chart.defaults.global.defaultFontColor = 'black';  Chart.defaults.global.defaultFontSize = 16;  var data = {  labels: [  <?php  foreach ($rlb as $data) {  echo '"'.date('M-Y', ($data->bulan - 25569) \* 86400).'",';  }  ?>  ],  datasets: [{  label: "Jumlah Pelanggan",  fill: false,  lineTension: 0.1,  //backgroundColor: "rgba(225,0,0,0.4)",  borderColor: "red", // The main line color  borderCapStyle: 'square',  borderDash: [], // try [5, 15] for instance  borderDashOffset: 0.0,  borderJoinStyle: 'miter',  pointBorderColor: "black",  pointBackgroundColor: "white",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "yellow",  pointHoverBorderColor: "brown",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: true  data: [<?php  foreach ($rlb as $data) {  echo $data->jumlah\_pelanggan.',';  }  ?>],  spanGaps: true,  }, {  label: "Jumlah Produk",  fill: true,  lineTension: 0.1,  //backgroundColor: "rgba(167,105,0,0.4)",  borderColor: "blue",  borderCapStyle: 'butt',  borderDash: [],  borderDashOffset: 0.0,  borderJoinStyle: 'miter',  pointBorderColor: "white",  pointBackgroundColor: "black",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "brown",  pointHoverBorderColor: "yellow",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: false  data: [  <?php  foreach ($rlb as $data) {  echo $data->jumlah\_produk.',';  }  ?>  ],  spanGaps: true,  }, {  label: "Pendapatan",  fill: true,  lineTension: 0.1,  //backgroundColor: "rgba(167,105,0,0.4)",  borderColor: "green",  borderCapStyle: 'butt',  borderDash: [],  borderDashOffset: 0.0,  borderJoinStyle: 'miter',  pointBorderColor: "white",  pointBackgroundColor: "black",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "brown",  pointHoverBorderColor: "yellow",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: false  data: [  <?php  foreach ($rlb as $data) {  echo $data->pendapatan.',';  }  ?>  ],  spanGaps: true,  }  ]  };  // Notice the scaleLabel at the same level as Ticks  var options = {  responsive: true,  title: {  display: true,  text: 'Relasi Linear Berganda'  },  scales: {  yAxes: [{  ticks: {  beginAtZero:true  },  scaleLabel: {  display: true,  labelString: 'Relasi Linear Berganda',  fontSize: 20  }  }]  }  };  // Chart declaration:  var myBarChart = new Chart(ctx, {  type: 'line',  data: data,  options: options  });  </script> |

Listings 6. 80 Syntax Untuk Menampilkan Data Secara Keseluruhan.

Jika menuliskan *syntax* tersebut, lalu menjalankannya di *browser* maka akan tampil informasi mengenai total pelanggan, total produk, dan total pendapatan yang didapatkan dalam beberapa bulan. Berikut merupakan hasil dari *chart* yang dimaksud.

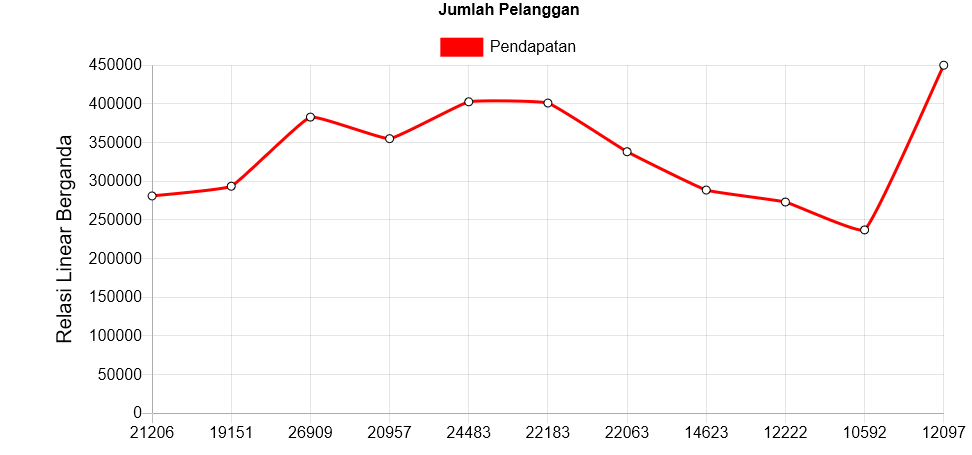


Gambar 6. 21 Chart Data Keseluruhan.

|  |
| --- |
| <script type="text/javascript">  var canvas = document.getElementById("barChart1");  var ctx = canvas.getContext('2d');  // Global Options:  Chart.defaults.global.defaultFontColor = 'black';  Chart.defaults.global.defaultFontSize = 16;  var data = {  labels: [  <?php  foreach ($rlb as $data) {  echo $data->jumlah\_pelanggan.",";  }  ?>  ],  datasets: [{  label: "Pendapatan",  fill: false,  lineTension: 0.1,  backgroundColor: "rgba(225,0,0,0.4)",  borderColor: "red", // The main line color  borderCapStyle: 'square',  borderDash: [], // try [5, 15] for instance  borderDashOffset: 0.0,  borderJoinStyle: 'miter',  pointBorderColor: "black",  pointBackgroundColor: "white",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "yellow",  pointHoverBorderColor: "brown",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: true  data: [<?php  foreach ($rlb as $data) {  echo $data->pendapatan.',';  }  ?>],  spanGaps: true,  }  ]  };  // Notice the scaleLabel at the same level as Ticks  var options = {  responsive: true,  title: {  display: true,  text: 'Jumlah Pelanggan'  },  scales: {  yAxes: [{  ticks: {  beginAtZero:true  },  scaleLabel: {  display: true,  labelString: 'Relasi Linear Berganda',  fontSize: 20  }  }]  }  };  // Chart declaration:  var myBarChart = new Chart(ctx, {  type: 'line',  data: data,  options: options  });  </script> |

Listings 6. 81 Syntax Untuk Menampilkan Informasi Jumlah Pelanggan.

Jika menuliskan *syntax* tersebut, lalu menjalankannya di *browser* maka akan tampil informasi mengenai total pelanggan terhadap total pendapatan yang didapatkan dalam beberapa bulan. Berikut merupakan hasil dari *chart* yang dimaksud.

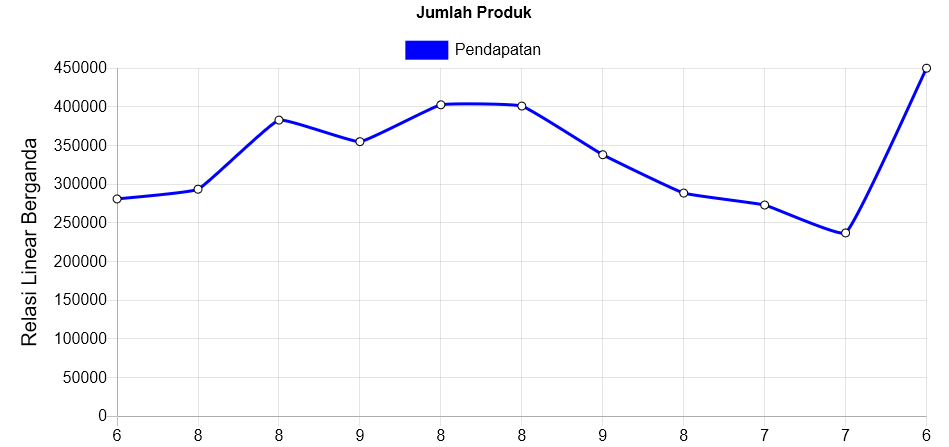


Gambar 6. 22 Chart Untuk Data Total Pelanggan.

|  |
| --- |
| <script type="text/javascript">  var canvas = document.getElementById("barChart2");  var ctx = canvas.getContext('2d');  // Global Options:  Chart.defaults.global.defaultFontColor = 'black';  Chart.defaults.global.defaultFontSize = 16;  var data = {  labels: [  <?php  foreach ($rlb as $data) {  echo $data->jumlah\_produk.",";  }  ?>  ],  datasets: [{  label: "Pendapatan",  fill: false,  lineTension: 0.1,  backgroundColor: "rgba(0, 0, 255, 0.4)",  borderColor: "blue", // The main line color  borderCapStyle: 'square',  borderDash: [], // try [5, 15] for instance  borderDashOffset: 1.0,  borderJoinStyle: 'miter',  pointBorderColor: "black",  pointBackgroundColor: "white",  pointBorderWidth: 1,  pointHoverRadius: 8,  pointHoverBackgroundColor: "yellow",  pointHoverBorderColor: "brown",  pointHoverBorderWidth: 2,  pointRadius: 4,  pointHitRadius: 10,  // notice the gap in the data and the spanGaps: true  data: [<?php  foreach ($rlb as $data) {  echo $data->pendapatan.',';  }  ?>],  spanGaps: true,  }  ]  };  // Notice the scaleLabel at the same level as Ticks  var options = {  responsive: true,  title: {  display: true,  text: 'Jumlah Produk'  },  scales: {  yAxes: [{  ticks: {  beginAtZero:true,  },  scaleLabel: {  display: true,  labelString: 'Relasi Linear Berganda',  fontSize: 20  }  }]  }  };  // Chart declaration:  var myBarChart = new Chart(ctx, {  type: 'line',  data: data,  options: options  });  </script> |

Listings 6. 82 Syntax Untuk Menampilkan Informasi Jumlah Produk.

Jika menuliskan *syntax* tersebut, lalu menjalankannya di *browser* maka akan tampil informasi mengenai total produk terhadap total pendapatan yang didapatkan dalam beberapa bulan. Berikut merupakan hasil dari *chart* yang dimaksud.

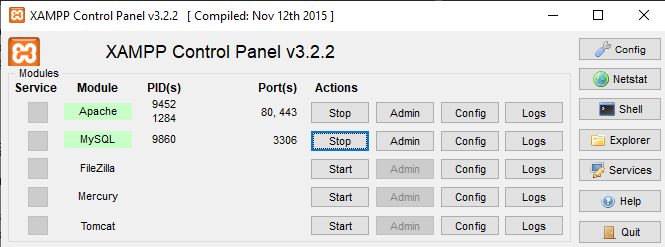


Gambar 6. 23 Chart Untuk Data Total Produk.

# PANDUAN DALAM MENGGUNAKAN SISTEM PERAMALAN

Pada bab kali ini kita akan membahas mengenai tata cara menggunakan sistem peramalan ini, berikut merupakan tahapan-tahapan dalam menjalankan atau menggunakan sistem permalan:

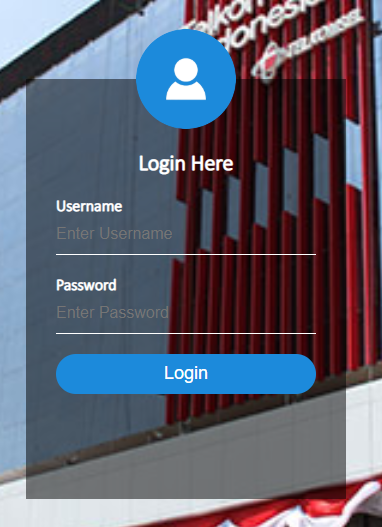
1. Tahap pertama buka aplikasi XAMPP, lalu hidupkan *apache* dan *MySQL*, fungsi dari menghidupkan *apache* dan *MySQL* ini sebagai *server local* agar sistem yang telah di buat dapat kita jalankan.



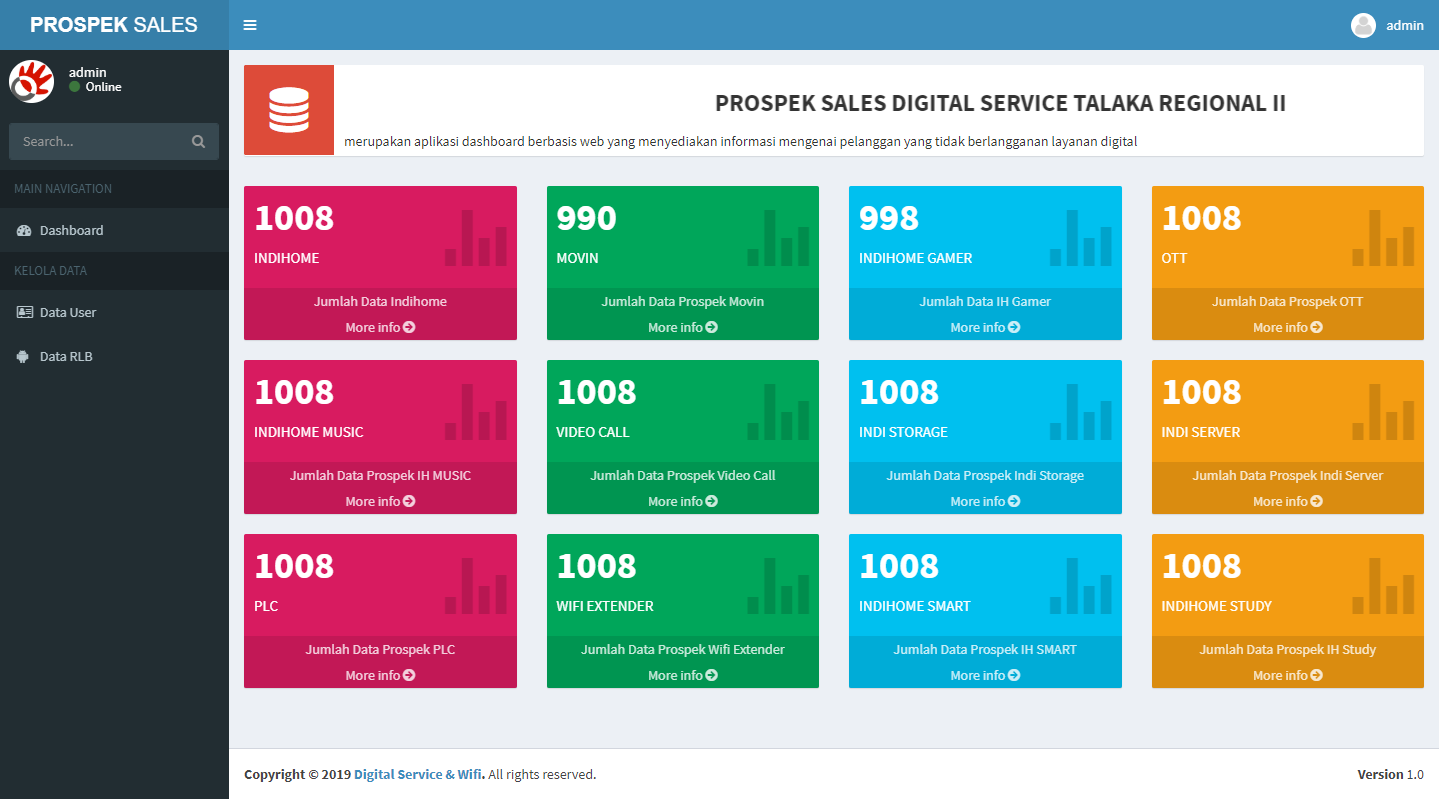
1. Lalu, buka *browser* dan masukan *URI* sebagai berikut:

<http://localhost/prospekL/>, *URI* tersebut disesuaikan dengan nama file yang kita simpan pada direktori *htdocs*.

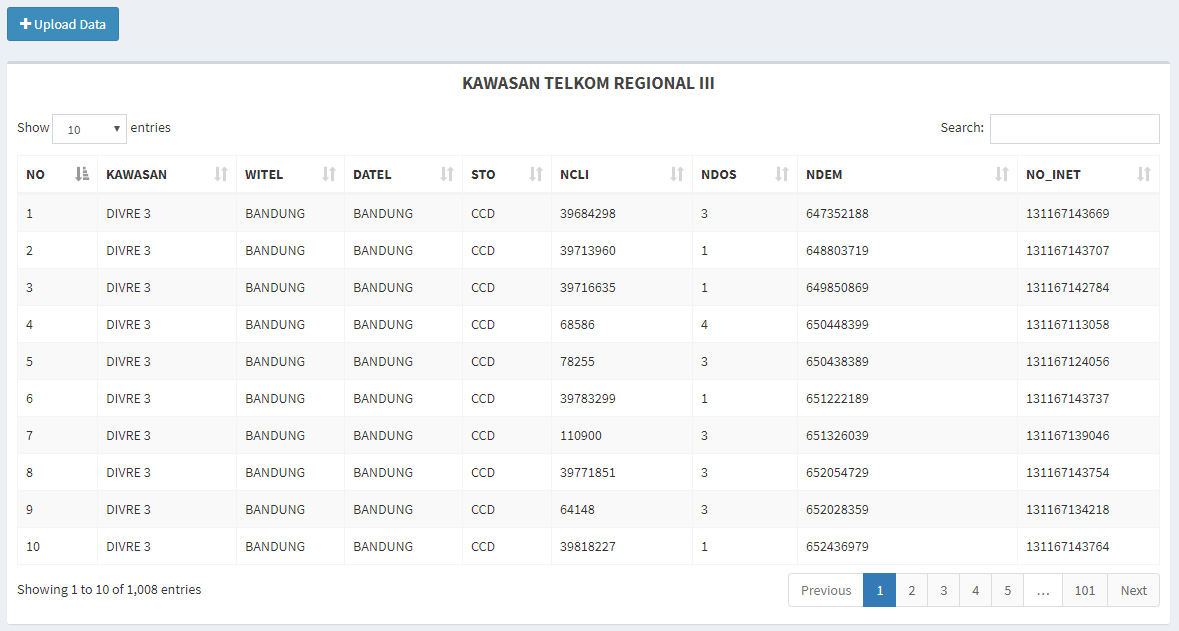
1. Jika telah berhasil menampilkan halaman *login*, maka selanjutnya kita tinggal memasukkan *username* dan *password* yang sudah terdaftar pada *database* sistem.



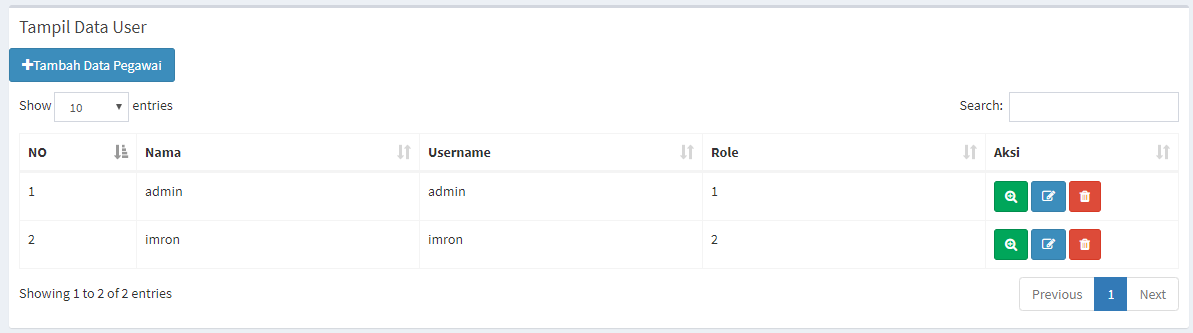
1. Pada halaman utama ini, terdapat beberapa informasi mengenai total pelanggan pada setiap *addOn* yang ada, selain itu untuk pegawai dengan *role­\_id­* admin, diberikan akses untuk mengelola data *user*.



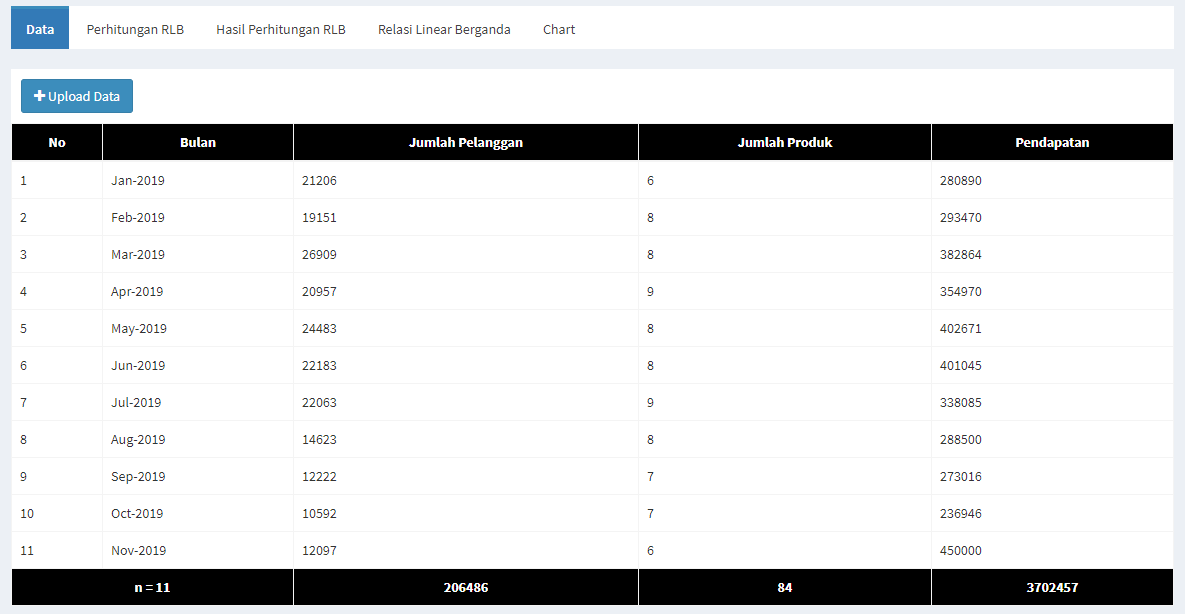
1. Jika kita masuk pada halaman detail salah satu *addOn*, maka didalamnya terdapat informasi mengenai nomor internet serta beberapa informasi lainnya, selain informasi terdapat juga satu *button* yang berfungsi untuk melakukan *upload* data pada sistem.



1. Pada halaman selanjutnya terdapat halaman untuk mengelola data pegawai, halaman ini dapat diakses hanya oleh admin, fungsi dari halamn ini untuk memberikan akses pada pegawai dalam mengakses sistem peramalan ini.



1. Pada halaman terakhir ini merupakan halaman untuk melakukan peramalan mengenai pendapatan pada PT TALAKA, selain dapat melakukan peramalan pada halaman ini juga dapat menampilkan *chart* sebagai gambaran terhadap pendapatan yang didapatkan oleh PT TALAKA.



# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | D. P. ES, Rahmi and S. Aditya, "Pemanfaatan Digital Marketing Bagi Usaha Mikro, Kecil, dan Menengah (UMKM) di Kelurahan Malaka Sari, Duren Sawit," *Jurnal Pemberdayaan Masyarakat Madani (JPMM),* vol. 1, no. 1, pp. 1-17, 2017. |
| [2] | T. Pradiani, "Pengaruh Sistem Pemasaran Digital Marketing Terhadap Peningkatan Volume Penjualan Hasil Industri Rumahan," *JIBEKA,* vol. 11, no. 2, pp. 46-53, 2017. |
| [3] | I. M. Ridwan, A. Fauzi, I. Aisyah, Susilawati and I. Sofyan, "Penerapan Digital Marketing Sebagai Peningkatan Pemasaran Pada UKM Warung Angkringan "WAGE" Bandung," *Jurnal Abdimas BSI,* vol. 2, no. 1, pp. 137-142, 2017. |
| [4] | N. Aghdaei, G. Kokogiannakis, D. Daly and T. McCarthy, "Linear Regression Models for Prediction of Annual Heating and Cooling Demand in Representative Australian Residential Dwellings," Energy Procedia, pp. 79-86, 2017. |
| [5] | J. S. Sebayang and B. Yuniarto, "Perbandingan Model Estimasi Artificial Neural Network Optimasi Genetic Algorithm dan Regresi Linier Berganda," Media Statiska, vol. I, no. 10, pp. 13-23, 2017. |
| [6] | Andre, "Dunia Ilkom," 9 October 2015. [Online]. Available: https://www.duniailkom.com. [Accessed 19 January 2020]. |
| [7] | F. Romadhoni, "Jagoan Hosting," 14 November 2019. [Online]. Available: https://jagoanhosting.com. [Accessed 19 January 2020]. |
| [8] | M. Ilham, "Materi Belajar," 14 October 2019. [Online]. Available: https://materibelajar.co.id. [Accessed 19 January 2020]. |
| [9] | Fajar, "Belajar CPP," 11 Maret 2018. [Online]. Available: https://www.belajarcpp.com. [Accessed 19 January 2020]. |
| [10] | Fajar, "Belajar CPP," 13 November 2019. [Online]. Available: https://www.belajarcpp.com. [Accessed 19 January 2020]. |
| [11] | P. A. Syah, "fdokumen," 27 April 2017. [Online]. Available: https://www.fdokumen.com. [Accessed 19 January 2020]. |
| [12] | Ridwan, "gitbooks," 12 November 2019. [Online]. Available: https://www.gitbooks.io. [Accessed 19 January 2020]. |
| [13] | E. S. Tataming, "Analisis Besar Kontribusi Hambatan Samping Terhadap Kecepatan Dengan Menggunakan Model Regresi Linier Berganda (Studi Kasus: Ruas Jalan dalam Kota Segmen Ruas Jalan Sarapung)," Jurnal Sipil Statik, vol. 2, no. 1, pp. 29-36, 2014. |
| [14] | A. Hijriani, K. Muludi and E. A. Andini, "Implementasi Metode Regresi Linier Sederhana Pada Penyajian Hasil Prediksi Pemakaian Air Bersih PDAM Way Rilau Kota Bandar Lampung Dengan Sistem Informasi Geografis," Jurnal Informatika Mulawarman, vol. 11, no. 2, pp. 37-42, 2016. |
| [15] | M. G. Mona, J. S. Kekenusa and J. D. Prang, "Penggunaan Regresi Linier Berganda Untuk Menganalisis Pendapatan Petani Kelapa Studi Kasus: Petani Kelapa di Desa Beo, Kecamatan Beo Kabupaten Talaud," vol. 4, no. 2, pp. 196-203, 2015. |

