

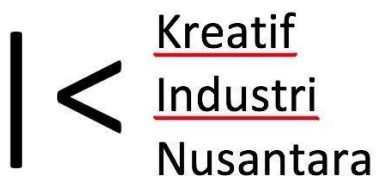
**APLIKASI *CLUSTERING* TAGIHAN RUMAH
SAKIT BERDASARKAN TINGKAT
KEPENTINGAN MENGGUNAKAN *K-MEANS*.**

APLIKASI *CLUSTERING* TAGIHAN RUMAH SAKIT BERDASARKAN TINGKAT KEPENTINGAN MENGGUNAKAN *K-MEANS*.

Woro Isti Rahayu

Ema Ainun Novia

Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Woro Isti Rahayu

Ema Ainun Novia

ISBN : xxx-xxx-xxxx-x-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2020

Hakcipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan
dengan cara apapun tanpa ijin tertulis dari penerbit

CONTRIBUTORS

Woro Isti Rahayu, Ema Ainun Novia,

Informatics Research Center, Politeknik Pos Indonesia, Bandung,
Indonesia

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	iii
DAFTAR TABEL.....	viii
KATA PENGANTAR.....	ix
BAB I.....	1
DASAR PEMROGRAMAN WEBSITE.....	1
1.1 Pengertian Website	1
1.2 Jenis – Jenis Website.....	1
1.3 Pengertian Pemrograman Web.....	2
1.4 Bahasa Pemrograman Web	2
1.5 Sejarah.....	3
1.6 Framework	36
BAB II.....	44
DEFINISI	44
2.1 <i>Data Mining</i>	44
2.2 Clustering.....	46
2.3 Algoritma <i>K-Means</i>	50
2.4 Sistem	54
BAB III	60
TOOLS DAN BAHASA PEMROGRAMAN YANG DIGUNAKAN	61
3.1 Tools Yang Digunakan	61

3.2 Bahasa Pemrograman Yang Digunakan	81
BAB IV	83
INSTALASI TOOLS YANG DIGUNAKAN	83
4.1 Instalasi Sublime Text	83
4.2 Instalasi XAMPP	87
4.3 Instalasi <i>Codeigniter</i>	93
BAB V	96
ANALISIS DAN PERANCANGAN PEMBUATAN APLIKASI.....	96
5.1 Perancangan Aplikasi	96
5.2 Perancangan Database	102
5.3 Perancangan User Interface	102
5.4 Perancangan Arsitektur Sistem.....	110
5.5 Perhitungan K-means	111
BAB VI	119
TUTORIAL PEMBUATAN APLIKASI.....	119
6.1 Mempersiapkan Tools	119
6.2 Membuat Database	120
6.3 Konfigurasi Database	126
6.4 Konfigurasi <i>Template CSS Bootstrap</i>	133
6.5 Tahap Pengkodean.....	145
DAFTAR PUSTAKA	244

DAFTAR GAMBAR

Gambar 1. 1 Logo HTML	4
Gambar 1. 2 Logo CSS	10
Gambar 1. 3 Logo PHP	13
Gambar 1. 4 Logo MySQL	24
Gambar 1. 5 Logo Java Script.....	27
Gambar 1. 6 Logo Basis Data	29
Gambar 1. 7 Logo Framework Electron	38
Gambar 1. 8 Logo Framework Proton Native	39
Gambar 1. 9 Logo Framework Django	39
Gambar 1. 10 Logo Framework Angular JS	40
Gambar 1. 11 Logo Framework Laravel.....	41
Gambar 1. 12 Logo Framework Rails.....	42
Gambar 1. 13 Logo Framework Spring	42
Gambar 1. 14 Logo Framework Codeigniter	43
Gambar 2. 1 Bidang Ilmu Pada Data Mining	45
Gambar 2. 2 Hierarchical Clustering	49
Gambar 2. 3 Partitional Clustering	50
Gambar 2. 4 Elemen Sistem.....	56
Gambar 2. 5 File Codeigniter.....	95
Gambar 3. 1 Logo Sublime Text.....	61
Gambar 3. 2 Tampilan Sublime Text.....	62
Gambar 3. 3 Logo XAMPP	64
Gambar 3. 4 Logo Codeigniter	66
Gambar 3. 5 Perbandingan PHP Biasa dan Codeigniter	67
Gambar 3. 6 Tampilan Controller Admedika	68
Gambar 3. 7 Tampilan Model Admedika	68

Gambar 3. 8 Tampilan View Tagihan.....	69
Gambar 3. 9 File Hasil Ekstrak.....	69
Gambar 3. 10 Folder Application	70
Gambar 3. 11 Folder System	71
Gambar 3. 12 Folder User_guide.....	72
Gambar 3. 13 Logo Bootstrap.....	74
Gambar 3. 14 Logo Google Chrome.....	78
Gambar 4. 1 Download File Sublime Text	83
Gambar 4. 2 Download File Sublime Text	84
Gambar 4. 3 File Sublime Text.....	84
Gambar 4. 4 Proses Instalasi Sublime Text	85
Gambar 4. 5 Proses Instalasi Sublime Text	85
Gambar 4. 6 Proses Instalasi Sublime Text	86
Gambar 4. 7 Tampilan Sublime Text.....	86
Gambar 4. 8 Download File XAMPP	87
Gambar 4. 9 Tampilan Installer Language	87
Gambar 4. 10 Proses Instalasi XAMPP	88
Gambar 4. 11 Proses Instalasi XAMPP	88
Gambar 4. 12 Proses Instalasi XAMPP.....	89
Gambar 4. 13 Proses Instalasi XAMPP	89
Gambar 4. 14 Proses Instalasi XAMPP.....	90
Gambar 4. 15 Proses Instalasi XAMPP	91
Gambar 4. 16 Proses Instalasi XAMPP	91
Gambar 4. 17 Proses Menjalankan XAMPP.....	92
Gambar 4. 18 Proses Menjalankan XAMPP.....	92
Gambar 4. 19 Tampilan XAMPP.....	93
Gambar 4. 20 Proses Instalasi Codeigniter	93
Gambar 4. 21 File Download Codeigniter	94

Gambar 4. 22 Hasil Ekstrak File Codeigniter	94
Gambar 4. 23 File Codeigniter.....	95
Gambar 5. 1 Analisis Prosedur Yang Sedang Berjalan	97
Gambar 5. 2 Analisis Prosedur Yang Akan Dibangun	98
Gambar 5. 3 Use Case Diagram.....	99
Gambar 5. 4 Class Diagram	100
Gambar 5. 5 Component Diagram	101
Gambar 5. 6 Deployment Diagram	102
Gambar 5. 7 Perancangan Database.....	102
Gambar 5. 8 User Interface Login	103
Gambar 5. 9 User Interface Halaman Utama Medical.....	103
Gambar 5. 10 <i>User Interface</i> Halaman Utama Keuangan	104
Gambar 5. 11 <i>User Interface</i> Tambah Data Tagihan.....	104
Gambar 5. 12 <i>User Interface</i> Tambah Data <i>User</i>	105
Gambar 5. 13 <i>User Interface</i> Halaman Tambah Data Rumah Sakit.....	105
Gambar 5. 14 <i>User Interface Edit</i> Data Tagihan	106
Gambar 5. 15 <i>User Interface Edit</i> Data <i>User</i>	106
Gambar 5. 16 <i>User Interface Edit</i> Data Rumah Sakit	107
Gambar 5. 17 <i>User Interface</i> Kelola Data Tagihan	107
Gambar 5. 18 <i>User Interface</i> Kelola Data <i>User</i>	108
Gambar 5. 19 <i>User Interface</i> Kelola Data Rumah Sakit	108
Gambar 5. 20 <i>User Interface</i> Perhitungan	109
Gambar 5. 21 User Interface Hasil Cluster.....	109
Gambar 6. 1 Menjalankan XAMPP	120
Gambar 6. 2 Tampilan Localhost.....	121
Gambar 6. 3 Membuat Database Baru	121
Gambar 6. 4 Membuat Database Baru	122
Gambar 6. 5 Membuat Table Database.....	122

Gambar 6. 6 Struktur Tabel Layanan.....	123
Gambar 6. 7 Tabel RS.....	123
Gambar 6. 8 Struktur Tabel Tagihan	124
Gambar 6. 9 Struktur Tb_centroid	124
Gambar 6. 10 Struktur Tb_cluster	125
Gambar 6. 11 Struktur Tb_klasifikasi.....	125
Gambar 6. 12 Struktur Tb_kriteria.....	125
Gambar 6. 13 Struktur Tabel User	126
Gambar 6. 14 Struktur Tb_hasil.....	126
Gambar 6. 15 Rename Nama Folder.....	127
Gambar 6. 16 Open Folder.....	127
Gambar 6. 17 Konfigurasi Config	128
Gambar 6. 18 Tampilan Login.....	129
Gambar 6. 19 Konfigurasi Autoload (Libraries).....	130
Gambar 6. 20 Konfigurasi Autoload (Helper)	131
Gambar 6. 21 Konfigurasi Database	132
Gambar 6. 22 Konfigurasi Routes	133
Gambar 6. 23 Folder Template	134
Gambar 6. 24 File Template	134
Gambar 6. 25 File Template	135
Gambar 6. 26 Tampilan View Yang Dibutuhkan	144
Gambar 6. 27 Tampilan Login.....	145
Gambar 6. 28 Tampilan Data User	152
Gambar 6. 29 Tampilan Tambah User.....	155
Gambar 6. 30 Tampilan Edit User	158
Gambar 6. 31 Tampilan Data Rumah Sakit	163
Gambar 6. 32 Tampilan Tambah Rumah Sakit	166
Gambar 6. 33 Tampilan Edit Rumah Sakit.....	169

Gambar 6. 34 Tampilan Data Tagihan.....	175
Gambar 6. 35 Tampilan Tambah Tagihan	178
Gambar 6. 36 Tampilan Edit Tagihan.....	182
Gambar 6. 37 Tampilan Kriteria.....	189
Gambar 6. 38 Tampilan Tambah Kriteria.....	191
Gambar 6. 39 Tampilan Edit Kriteria	194
Gambar 6. 40 Tampilan Centeroid.....	199
Gambar 6. 41 Tampilan Tambah Centeroid	202
Gambar 6. 42 Tampilan Edit Centeroid.....	205
Gambar 6. 43 Tampilan Klasifikasi	211
Gambar 6. 44 Tampilan Tambah Klasifikasi	214
Gambar 6. 45 Tampilan Tambah Klasifikasi	214
Gambar 6. 46 Tampilan Edit Klasifikasi	220
Gambar 6. 47 Perhitungan K-Means	224
Gambar 6. 48 Perhitungan K-Means	224
Gambar 6. 49 Perhitungan K-Means	225
Gambar 6. 50 Perhitungan K-Means	225
Gambar 6. 51 Tampilan Pengelompokkan.....	239
Gambar 6. 52 Tampilan Pengelompokkan.....	239
Gambar 6. 53 Tampilan Pengelompokkan.....	240

DAFTAR TABEL

Table 1. 1 Arithmetic Operator	20
Table 1. 2 Relational Operator	21
Table 1. 3 Logical Operator	22
Tabel 5. 1 Arsitektur Perangkat Lunak Sistem	110
Tabel 5. 2 Arsitektur Perangkat Keras Sistem	110
Tabel 5. 3 Data Tagihan	111
Tabel 5. 4 Centroid Awal	112
Tabel 5. 5 Hasil Jarak Data Terhadap Cluster Iterasi I	115
Tabel 5. 6 Centroid Baru Untuk Iterasi II	116
Tabel 5. 7 Hasil Jarak Data Terhadap Cluster Pada Iterasi II	117
Tabel 5. 8 Centroid Baru Untuk Iterasi III	117
Tabel 5. 9 Hasil Clustering Akhir	118

KATA PENGANTAR

Penulisan buku ini dimaksudkan untuk memberikan arahan atau sarana pembelajaran kepada pembaca mengenai data mining serta penerapan data mining dalam kehidupan sehari-hari yang kemudian proses data mining tersebut dibuat dengan menggunakan tools bahasa pemrograman php.

Woro Isti Rahayu, Ema Ainun Novia

Bandung, Jawa Barat

Januari, 2020

BAB I

DASAR PEMOGRAMAN WEBSITE

1.1 Pengertian Website

Website merupakan sebuah halaman web yang saling berhubungan yang umumnya berisikan tentang kumpulan informasi berupa teks, gambar, animasi, audio, video maupun gabungan dari semuanya yang biasanya dibuat untuk personal, organisasi dan juga perusahaan.

1.2 Jenis – Jenis Website

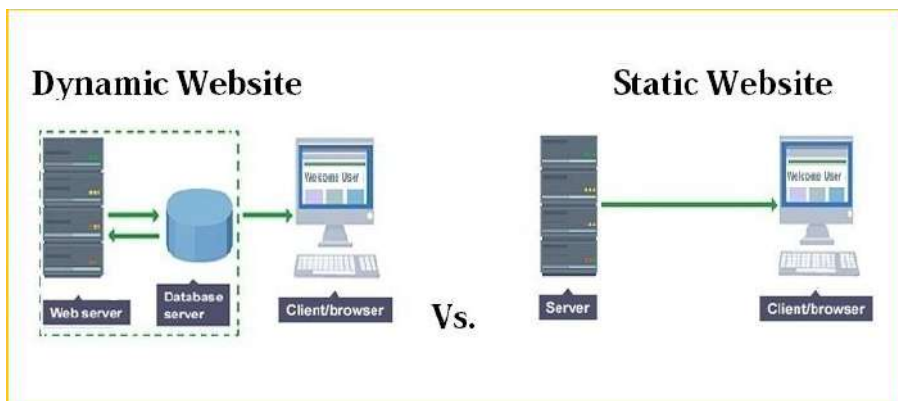
Berdasarkan sifatnya *website* dibagi menjadi 2 jenis yaitu *website* statis dan *website* dinamis.

1.2.1 Website Statis

Web statis ialah web yang berisi tentang informasi-informasi yang memiliki sifat statis (tetap) atau pengguna tidak dapat berinteraksi dengan website tersebut, web statis dapat dilihat dari tampilan website tersebut jika suatu web hanya berhubungan dengan halaman web lain yang berisi informasi tetap maka web tersebut termasuk kedalam kategori web statis, pada web statis pengguna hanya dapat melihat isi web tersebut dan jika di klik hanya akan berpindah pada halaman lainnya. Dalam web statis interaksi pengguna sangatlah terbatas.

1.2.2 Website Dinamis

Web dinamis adalah web yang dapat menampilkan informasi serta dapat membuat pengguna berinteraksi seperti dengan form input, button sehingga dapat mengolah informasi yang di tampilkan pada web tersebut, web dinamis bersifat tidak kaku dan terlihat lebih enak dipandang.



1.3 Pengertian Pemrograman Web

Pemrograman web terdiri dari dua kata yaitu pemrograman dan web. Pemrograman adalah suatu proses atau cara pembuatan program, yaitu sekumpulan instruksi atau perintah kepada komputer untuk melakukan langkah atau proses tertentu dalam rangka menyelesaikan suatu masalah. Sedangkan web merupakan jaringan komputer yang terdiri dari sekumpulan situs internet yang menampilkan teks, grafik, gambar maupun suara melalui protokol transfer hypertext.

1.4 Bahasa Pemrograman Web

Bahasa pemrograman web terdiri dari beberapa unsur bahasa. Setidaknya terdapat 5 bahasa utama yang digunakan untuk membuat website dimana masing-masing memiliki perannya sendiri-sendiri.

1. HTML, merupakan pembentuk struktur sebuah halaman website yang menempatkan setiap elemen website sesuai dengan layout yang diinginkan.
2. CSS, merupakan pembentuk desain website dengan cara mengatur setiap elemen HTML agar tampil semenarik mungkin pada browser.
3. PHP, merupakan pemrosesan data pada sisi server sesuai dengan apa yang diminta oleh client menjadi informasi siap ditampilkan. Dan juga sebagai penghubung aplikasi web dengan *database*.
4. SQL, merupakan Bahasa yang mengatur transaksi data antara aplikasi dengan database sebagai penyimpanan data. Database yang digunakan biasanya MySQL, Oracle, SQL Server dan sebagainya.
5. Java Script, merupakan Bahasa yang memproses data pada sisi client serta dapat juga memanipulasi HTML dan CSS secara dinamis.

1.5 Sejarah

Cabang ilmu pemrograman cukup luas, dan erat kaitannya dengan disiplin ilmu yang lainnya. Hal ini bisa dilihat dari berbagai aplikasi yang merupakan hasil kombinasi dari berbagai macam ilmu.

1.5.1 Sejarah HTML



Gambar 1. 1 Logo HTML

HTML merupakan singkatan dari *Hypertext Markup Language* yaitu Bahasa standar web yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusup pada setiap elemen pada *website*. HTML berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen website sesuai dengan layout yang diinginkan. HTML atau *Hypertext Markup Language* telah menjadi teknologi utama untuk Web sejak awal tahun 1990-an. Tim Berners-Lee menciptakan HTML pada tahun 1989 sebagai cara sederhana namun efektif untuk mengkodekan dokumen elektronik. Bahkan, tujuan awal dari web browser adalah untuk melayani pembaca untuk membuka dokumen berformat HTML.

1.5.1.1 Perkembangan HTML

Didalam HTML telah mengalami berbagai perkembangan dari tiap-tiap versinya. Berikut adalah versi-versi dari perkembangan HTML tersebut:

1. HTML Versi 1.0

HTML Versi 1.0 merupakan pionir yang di dalamnya masih terdapat banyak sekali kelemahan hingga wajar jika tampilan yang dihasilkan sangat sederhana. Kemampuan yang dimiliki versi 1.0 ini antara lain heading, paragraf, hypertext, list, serta cetak tebal dan miring pada teks. Versi ini juga mendukung peletakan image pada dokumennya tanpa memperbolehkan teks di sekelilingnya (wrapping).

2. HTML Versi 2.0

Pada HTML Versi ini, penambahan kualitas HTML terletak pada kemampuannya untuk menampilkan suatu form pada dokumen. Dengan adanya form ini, kita dapat memasukkan nama, alamat, serta saran dan kritik. HTML versi 2.0 ini merupakan pionir dari adanya web interaktif.

3. HTML Versi 3.0

Versi HTML 3.0 menambahkan beberapa fasilitas baru seperti FIGURE yang merupakan perkembangan dari IMAGE untuk meletakkan gambar dan tabel. Selain itu, HTML ini juga mendukung adanya rumus-rumus matematika dalam dokumennya. Versi ini yang disebut HTML+- tidak bertahan lama dan segera digantikan dengan versi 3.2.

4. HTML Versi 3.2

HTML versi ini merupakan HTML yang sering digunakan. Di dalamnya terdapat suatu teknologi untuk meletakkan teks di sekeliling gambar, gambar sebagai latar belakang, tabel, frame, style sheet dan lain-lain. Selain itu pada HTML versi ini Kita bisa menggunakan script di luar HTML untuk mendukung kinerja HTML kita tersebut, seperti Javascript, VBScript dan lain-lain.

5. HTML Versi 4

HTML ini memuat banyak sekali perubahan dan revisi dari pendahulunya yaitu HTML 3.2. Perubahan ini hampir terjadi di segala perintah HTML seperti tabel, image, link, text, meta, imagemaps, form dan lain-lain.

6. HTML Versi 4.01

HTML versi 4.01 merupakan revisi dari HTML 4.0. Versi terbaru ini memperbaiki kesalahan-kesalahan kecil (minor errors) pada versi terdahulunya. HTML 4.01 ini juga menjadi standarisasi untuk elemen dan atribut dari script XHTML 1.0.

7. HTML Versi 5.0

Teknologi ini mulai diluncurkan pada tahun 2009, tetapi pada tanggal 4 Maret 2010 terdapat sebuah informasi bahwa W3C (World Wide Web Consortium) dan IETF (Internet Engineering Task Force) yaitu sebuah organisasi yang menangani HTML sejak versi 2.0 telah mengembangkan versi HTML terbaru, yaitu versi 5.0.

HTML 5 adalah sebuah prosedur pembuatan tampilan web baru yang merupakan penggabungan antara CSS, HTML itu sendiri dengan JavaScript.

Kelebihan - kelebihan HTML antara lain:

1. Mewakili bahasa penkodean yang lintas platform (lintas platform), maksudnya HTML dapat digunakan pada berbagai jenis mesin komputer yang berbeda dan berbagai jenis sistem operasi yang berbeda. Editorial ASCII dengan karakter fleksibel.
2. Dapat disisipi gambar baik gambar statik atau dinamis (animasi) termasuk menggunakan gambar untuk dijadikan hyperlink. Gambar di sini digunakan untuk mengubah pada suatu halaman web, di mana setiap titik-titik yang telah didefinisikan sebagai

persegi panjang (kotak), poligon (kurva tak beraturan) atau lingkaran yang digunakan untuk 'melompat' ke halaman lain, atau tautan ke halaman di luar web yang memang.

3. Dapat disisipi animasi berupa Java Applet atau file-file animasi dari Macromedia Flash atau Macromedia Shockwave (untuk keperluan ini, browser harus memiliki plug-in khusus untuk menggunakan file-file animasi ini).
4. Dapat disisipi bahasa pemrograman untuk mempercantik halaman web seperti Javascript, VBScript, Active Server Pages, Perl, Tcl, PHP dan sebagainya.

Kekurangan – kekurangan dari HTML ini antara lain :

1. Menghasilkan halaman yang statistik, yang saya tahu untuk memperoleh halaman yang dinamis harus menggunakan bahasa pemrograman tertentu seperti Javascript atau VBScript dan animasi seperti Flash atau Shockwave.
2. Memiliki tag-tag yang sangat banyak dilakukan untuk yang masih awam. Tidak dapat menghasilkan halaman yang interaktif. Interaktif di sini maksudnya klien dapat memfokuskan dengan server. Untuk keperluan itu, HTML harus disisipi bahasa pemrograman yang dapat membantu hal tersebut, contohnya Perl atau Tcl.

1.5.1.2 Unsur HTML

1.5.1.2.1 Tag

Tag adalah simbol khusus (markup) berupa 2 karakter “<” dan “>” yang mengapit suatu teks sebagai nama tag. Contohnya tag <body> adalah tag dengan nama body.

1.5.1.2.2 Atribut

Atribut yaitu property yang mengatur bagaimana elemen dari suatu tag akan ditampilkan. Atribut ada yang memiliki nilai dan ada yang tidak memiliki nilai. Nilai suatu atribut ditulis di dalam tanda petik ganda (“...”), dipisahkan dengan symbol sama dengan (=) dari nama. Contohnya, `<p align=”center”>`.

1.5.1.2.3 Element

Element adalah bagian dari skrip HTML yang terdiri dari tag pembuka, isi element dan tag tutup. Jika element ditampilkan di browser maka hanya isi element saja yang akan tampil. Contoh nya, `<p>` Politeknik Pos Indonesia `</p>` maka jika ditampilkan pada browser akan tampil hanya tulisa Politeknik Pos Indonesia.

1.5.1.3 Struktur Dasar HTML

Dalam penulisan HTML ada beberapa tag yang wajib dituliskan dengan struktur yang telah ditentukan. Tag tersebut adalah sebagai berikut :

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Poltekpos</title>
  </head>
  <body>
    Politeknik Pos Indonesia
  </body>
</html>
```

- `<!DOCTYPE HTML>` adalah tag awal dari setiap dokumen HTML yang berfungsi untuk menginformasikan pada browser bahwa dokumen yang sedang dibuka adalah dokumen HTML.
- `<html>...</html>` adalah tag yang menunjukkan pembuka dan penutup dokumen HTML.

- `<head>....</head>` adalah tag yang digunakan untuk menyimpan berbagai informasi tentang dokumen HTML.
- `<title>....</title>` adalah tag yang digunakan untuk membuat judul website yang nanti akan muncul di browser.
- `<body>....</body>` Adalah tag yang menunjukkan bagian utama website. Semua yang akan ditampilkan pada halaman browser dituliskan di tag ini.

1.5.1.4 Hubungan HTML Dan PHP

Halaman web biasanya disusun dari kode-kode html yang disimpan dalam sebuah file berekstensi .html. File html ini dikirimkan oleh server (atau file) ke browser, Kemudian browser menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan program php, program ini harus diterjemahkan oleh web-server sehingga menghasilkan kode html yang dikirim ke browser agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode html sehingga dapat langsung ditampilkan bersama dengan kode-kode html tersebut. Program php dapat ditambahkan dengan mengapit program tersebut diantara tanda.

Tanda-tanda tersebut biasanya disebut tanda untuk escaping (kabur) dari kode html. File html yang telah dibubuhi program php harus diganti ekstensi-nya menjadi .php3 atau php. Php merupakan bahasa pemrograman web yang bersifat server-side HTML=embedded scripting, di mana script-nya menyatu dengan HTML dan berada di server. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di server tetapi disertakan HTML biasa. PHP dikenal sebagai bahasa scripting yang menyatu dengan tag HTML, dieksekusi di server dan digunakan untuk membuat halaman web yang dinamis seperti ASP (Active Server Pages) dan JSP (Java Server Pages).

1.5.2 Sejarah CSS



Gambar 1. 2 Logo CSS

Pada perkembangannya CSS sudah masuk level 3 untuk sekarang, dimana dimulai CSS level 1 atau yang sering di sebut CSS aja, kemudian level 2 yang merupakan penyempurnaan dari CSS level sebelumnya, yaitu CSS level 1. CSS merupakan alternatif bahasa pemrograman web masa yang akan datang, dimana mempunyai banyak keuntungan, diantaranya :

- Ukuran file lebih kecil
- Load file lebih cepat
- Dapat berkolaborasi dengan JavaScript
- Pasangan setia XHTML
- Menghemat pekerjaan tentunya, dimana hanya membuat 1 halaman CSS.
- Mudah mengganti-ganti tampilan dengan hanya merubah file CSS nya saja.

Kelebihan penggunaan CSS :

1. halaman web menjadi tidak padat, hingga halaman web menjadi SEO friendly.
2. anda dapat dengan mudah mengganti tampilan web dengan cukup mengganti nilai dan definisi pada pada satu file CSS saja.

Kekurangannya penggunaan CSS :

1. Susah untuk dimengerti dibandingkan dengan bahasa pemrograman lainnya contohnya: HTML, PHP, Java Script, dan lain-lain dan CSS ini memerlukan waktu yang sangatlah lama untuk bisa mahir untuk mempelajari CSS ini.

1.5.2.1 Perkembangan CSS

1. CSS 1

Pada tanggal 17 Agustus 1996 World Wide Web Consortium (W3C) menetapkan CSS sebagai bahasa pemrograman standard dalam pembuatan web. Tujuannya adalah untuk mengurangi pembuatan tag-tag baru oleh Netscape dan Internet Explorer, karena kedua browser tersebut sedang bersaing mengembangkan tag sendiri untuk mengatur tampilan web.

CSS 1 mendukung pengaturan tampilan dalam hal :

1. Font (Jenis ketebalan).
2. Warna, teks, background dan elemen lainnya.
3. Text attributes, misalnya spasi antar baris, kata dan huruf.
4. Posisi teks, gambar, table dan elemen lainnya.
5. Margin, border dan padding.

2. CSS 2

Pada tahun 1998, W3C menyempurnakan CSS tahap awal dengan menciptakan standard CSS 2 yang menjadi standard hingga saat ini. Pada level CSS 2 ini, dimasukkan semua atribut dari CSS 1 dan diperluas dengan penekanan pada International Accessibility and Capabilty khususnya media-specific CSS. CSS 2 dikembangkan untuk memenuhi kebutuhan terhadap format dokumen agar bisa ditampilkan di printer.

3. CSS 3

CSS 3 adalah versi terbaru dari CSS yang mampu melakukan banyak hal dalam mendesain website. CSS 3 dapat melakukan animasi pada halaman website, diantaranya animasi warna dan animasi 3D. Dengan CSS 3 desainer dimudahkan dalam hal kompatibilitas websitenya pada smartphone dengan dukungan fitur baru yakni media query. Selain itu, banyak fitur baru pada CSS 3 yaitu : Multiple background, border-radius, drop-shadow, border-image, CSS-Math dan CSS Object Model.

Fitur terbaru CSS 3 :

1. Animasi, sehingga pembuatan animasi tidak memerlukan program sejenis Adobe Flash dan Microsoft Silverlight.
2. Beberapa efek teks, seperti teks berbayang, kolom koran dan “Word-Wrap”.
3. Beberapa efek pada kotak, seperti kotak yang ukurannya dapat diubah-ubah, transformasi 2 dimensi dan 2 dimensi, sudut-sudut yang tumpul dan bayangan.

Kelebihan dari CSS antara lain :

1. Memisahkan desain dengan konten halaman web.
2. Mengatur desain seefisien mungkin.

3. Jika kita ingin mengubah suatu tema halaman web, cukup modifikasi pada css saja.
4. Menghadirkan sesuatu yang tidak dapat dilakukan oleh HTML.
5. Lebih mudah didownload karena lebih ringan ukuran filenya.
6. Satu CSS dapat digunakan banyak halaman web.

Kekurangan dari CSS antara lain :

1. Tampilan pada browser berbeda-beda.
2. Kadang juga terdapat browser yang tidak support CSS (browser lama).
3. Harus tahu cara menggunakannya.
4. Dibutuhkan waktu lebih lama dalam membuatnya.
5. Belum lagi ada bug/error dalam CSS.

1.5.3 Sejarah PHP



Gambar 1. 3 Logo PHP

PHP adalah singkatan dari “PHP: Hypertext Preprocessor”, yaitu bahasa pemrograman disisi server yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan

bersamaan dengan HTML. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server.

Pada awalnya PHP muncul pada tahun 1995, PHP ini dibuat oleh Rasmus Lerdor. Saat itu, PHP masih bernama Form Interpreted (FI) yang berwujud dalam bentuk sekumpulan skrip yang digunakan untuk pengolahan data formulir dari web. Kemudian Rasmus merilis kode sumber yang sesuai untuk umum dan menamakannya PHP / FI. Dengan rilis kode sumber tersebut menjadi sumber terbuka (open source), maka banyak pemrograman yang tertarik untuk dikembangkan PHP.

Pada November 1997, dikeluarkan PHP / FI 2.0. Saat rilis, interpreter PHP telah diimplementasikan dalam program C. Pada rilis ini dilepaskan juga modul-modul ekstensi yang meningkatkan kemampuan PHP / FI secara signifikan. Pada tahun 1997, perusahaan bernama Zend melakukan penerjemah perilsan ulang PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Selanjutnya, pada Juni 1998, perusahaan Zend sebelumnya merilis interpreter baru untuk PHP dan meresmikan rilis tersebut menjadi PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak digunakan di awal abad ke-21. Versi ini sering digunakan karena kemampuannya untuk membangun aplikasi kompleks namun tetap memiliki kecepatan dan kehandalan tinggi.

Pada Juni 2014, Zend merilis PHP 5.0. Pada versi ini, inti dari juru bahasa PHP mengubah perubahan besar. Versi ini juga menyetujui model pemrograman dengan objek wisata yang diusulkan di dalam PHP untuk menjawab perkembangan Bahasa pemrograman pada saat objek. Server objek bawaan ditambahkan pada versi 5.4 untuk menyediakan pengembang dalam kode PHP tanpa melakukan instalasi perangkat lunak server. Versi terbaru dan stabil dari PHP sekarang ini adalah versi 7.0.16 dan 7.1.2 yang sudah resmi dirilis pada 17 Februari 2017.

Untuk menjelaskan cara penulisan kode PHP, bisa dilihat ada empat macam cara penulisan kode PHP :

1. `<? echo ("Politeknik Pos Indonesia\n"); ?>`
2. `<?php echo ("Politeknik Pos Indonesia\n"); ?>`
3. `<script language="php">`
`echo ("Politeknik Pos Indonesia");`
`</script>`

1.5.3.1 Tipe Data

Dalam Bahasa pemrograman ada berbagai macam tipe data, seperti berikut :

1. Integer

Integer adalah type data pada php yang berupa angka bulat seperti 1, 22, 100, 1000, type data ini sangat umum digunakan di bahasa pemrograman khususnya berkaitan dengan angka bulat. Nilai integer bisa bernilai negatif atau positif dan jika tidak diberi tanda (–) maka diasumsikan sebagai nilai positif. Nilai dari tipe data integer dapat ditulis dalam bentuk decimal, octal, maupun heksadesimal.

Contoh bilangan bulat yang ditulis dalam bentuk decimal :

3000
-578
+98

Contoh bilangan bulat yang ditulis dalam bentuk oktal :

0755	# desimal = 493
+010	# desimal = 8

Contoh bilangan bulat yang ditulis dalam bentuk heksadisimal :

0xFF	# desimal = 255
------	-----------------

0x10	# desimal = 10
------	----------------

2. Float

Float atau nama lainnya adalah floating point atau real number adalah type data pada php yang memiliki bagian desimal di akhir angka contohnya adalah 3,21 atau 4,5 dalam penulisan type data float didalam php bukan menggunakan koma (,) tetapi menggunakan titik (.). nilai yang diperbolehkan pada tipe float adalah dari 1.7E-308 sampai dengan 1.7E+308 dengan 15 digit akurasi.

Bilangan float dapat di tulis dalam notasi sains (*scientific*) dan notasi normal. Contoh bilangan float sebagai berikut :

3.14

0.017

0.314E1

17.0E-3

3. String

String adalah type data pada php yang berisi text dan karakter dimana bentuknya bisa kata atau kalimat. Bahasa PHP mendukung dua tipe *String* yaitu diapit oleh petik tunggal dan juga petik ganda.

String yang dibuat dengan petik tunggal tidak mendukung proses interpolasi variabel. Interpolasi variable adalah proses mengganti nama variabel yang disisipkan didalam *string* dengan nilai dari variable tersebut. Interpolasi variable hanya dapat dilakukan jika pembuatan *String* menggunakan tanda petik ganda.

Contoh :

<?php

\$nama = "Niko";

```
Echo "Hallo $nama, apa kabar?";  
?>
```

4. Boolean

Type boolean adalah tipe data pada php yang paling sederhana dalam bahasa pemrograman apapun. karena tipe data ini hanya memiliki dua nilai yaitu true dan false (huruf besar atau kecil tidak berpengaruh). tipe data boolean sering kali digunakan pada operasi logika seperti kondisi if, while, looping dll.

Pada beberapa bahasa pemograman nilai true bisa digantikan 1 dan nilai false digantikan 0. Nilai yang ekuivalen dengan false adalah:

- False
- Zero
- "0"
- NULL
- array kosong
- string kosong

Untuk mencetak nilai tipe data boolean dapat kita gunakan keyword `var_dump()`.

```
<?php  
$var1=True;  
$var2=TRUE;  
  
$var3=False;  
$var4=FALSE;  
  
var_dumb ($var1);echo "<br>";
```

```
var_dump ($var2);echo "<br>";  
var_dump ($var3);echo "<br>";  
var_dump ($var4);echo "<br>";  
  
?>
```

Maka outputnya :

```
bool(true)  
bool(true)  
bool(false)  
bool(false)
```

Selain menggunakan `var_dump()`, kita juga dapat mencetak nilai tipe data boolean dengan keyword `echo` seperti contoh dibawah ini :

```
<?php  
$var1=True;  
$var2=TRUE;  
  
$var3=False;  
$var4=FALSE;  
  
//untuk nilai true  
echo $var1."<br>";  
echo $var2."<br>";  
  
//untuk nilai false
```

```
echo $var3."<br>";  
echo $var4."<br>";  
  
?>
```

Maka outputnya :

```
1  
1
```

5. Array

Array berbeda dengan type data pada php seperti integer atau boolean, karena array adalah sebuah tipe data yang didalamnya terdiri dari kumpulan tipe data. penulisan array terdiri dari kunci (key) dan nilai (value). key berfungsi sebagai penunjuk posisi dimana value tersimpan. tanda => berfungsi untuk memberikan nilai kepada key. untuk mengakses nilai dari array kita menggunakan kombinasi nama variable dan nilai key agar mempermudah.

Beberapa jenis array yaitu: Numeric Array, Associative Array, dan Multidimensional Array. Berikut ini masing-masing penjelasan singkat masing-masing jenis array:

1. Numeric Array

Numeric array merupakan jenis array dimana index elemen yang terdapat didalamnya di definisikan dengan angka, biasanya index pertama dimulai dengan angka "0" dan seterusnya.

2. Associative Array (array asosiatif)

Array associative berbeda dengan jenis array numeric, array jenis ini mendefinisikan index-nya menggunakan name atau nama untuk dapat mengakses nilai dari elemen array tersebut.

3. Multidimensional Array

Array multidimensional adalah metode penyimpanan nilai data array lebih dari satu array, ini artinya array didalam array, array multidimensi sangat berguna untuk menyimpan nilai variabel dengan banyak data meskipun lebih sulit untuk menggunakannya tetapi array jenis ini lebih efisien dalam pemakai karna mampu menampung banyak data.

6. Operator

Operator digunakan untuk memanipulasi nilai suatu variable. Variable yang nilainya dimodifikasi oleh operator disebut operand. Operator diklasifikasikan menjadi beberapa antara lain :

1. Arithmetic Operator

Arithmetic operator digunakan untuk melakukan perhitungan matematika. Ada beberapa arithmetic operator yaitu :

Table 1. 1 Arithmetic Operator

Operator	Penjelasan	Contoh	Hasil
+	Penjumlahan	<code>\$a = 5 + 1;</code> <code>Echo \$a;</code>	6
-	Pengurangan	<code>\$a = 5 - 2;</code> <code>Echo \$a;</code>	3
*	Perkalian	<code>\$a = 5*5;</code> <code>Echo \$a;</code>	25
/	Pembagian	<code>\$a = 50/5;</code>	10

		Echo \$a;	
%	Sisa Hasil Pembagian	\$a = 9/2; Echo \$a;	1
++	Increment	\$a = 1; Echo \$a++;	2
--	Decrement	\$a = 1; Echo \$a--;	0

2. Relational Operator

Relational operator digunakan untuk membandingkan nilai dari 2 operand. Hasil perbandingannya dinyatakan dalam nilai Boolean. TRUE berarti benar dan FALSE berarti salah. Beberapa jenis relational operator adalah sebagai berikut :

Table 1. 2 Relational Operator

Operator	Penjelasan	Contoh	Hasil
>	Lebih besar dari	5<10	True
<	Lebih kecil dari	10<5	False
==	Apakah sama	7=6	False
>=	Lebih besar atau sama dengan	4>=6	False
<=	Lebih kecil atau sama dengan	2<=5	True
!=	Tidak sama	3!=7	True

3. Logical Operator

Logical operator digunakan untuk membandingkan dua nilai variable yang bertipe Boolean. Hasil yang didapat dari penggunaan logical operator adalah Boolean (True, False). Berikut macam-macam logical operator :

Table 1. 3 Logical Operator

Operator	Penjelasan	Contoh	Hasil
&&	Dan	\$ x && \$ y	Benar jika \$ x dan \$ y benar
	Atau	\$ x \$ y	Benar jika \$ x atau \$ y benar
Xor	Xor	\$ x xor \$ y	Benar jika \$ x atau \$ y benar tetapi tidak keduanya
!	Tidak	! \$ x	Benar jika \$ x tidak benar
Dan	Dan	\$ x dan \$ y	Benar jika \$ x dan \$ y benar
Atau	Atau	\$ x dan \$ y	Benar jika \$ x atau \$ y benar.

1.5.3.2 Kekurangan dan Kelebihan PHP

Kelebihan dari PHP antara lain :

1. Dapat membuat Web menjadi lebih Dinamis.

2. PHP sifatnya Open Source yang artinya bisa digunakan oleh siapa saja secara gratis.
3. Program yang dibuat dengan PHP dapat digunakan atau dijalankan oleh Semua Sistem Operasi karena PHP berjalan secara Web Base yang berarti semua Sistem Operasi bahkan HP yang memiliki Web Browser bisa menggunakan program PHP.
4. Aplikasi PHP lebih cepat dan mudah jika dibandingkan dengan ASP dan Java.
5. Mendukung beberapa paket Database seperti MySQL, Oracle, PostgreSQL, dan lain-lain.
6. Bahasa pemrograman PHP tidak membutuhkan Kompilasi / Compile dalam penggunaannya.
7. Banyak Web Server yang mendukung PHP seperti Apache, Lighttpd, IIS dan yang lainnya.
8. Dalam pengembangan Aplikasi PHP mudah karena banyak Dokumentasi, Refrensi & Developer yang mendukung dalam pengembangannya.
9. Banyak bertebaran Aplikasi dan Program PHP yang Gratis & Siap pakai seperti PrestaShop, WordPress, dan masih banyak lagi yang lainnya.

Kekurangan dari PHP antara lain :

1. Tidak ideal jika untuk pengembangan skala besar.
2. Tidak mempunyai sistem pemrograman berorientasi objek yang sesungguhnya.
3. Tidak dapat memisahkan antara tampilan dengan logik dengan baik (Meskipun penggunaan template bisa memperbaikinya)

4. PHP mempunyai kelemahan security tertentu yang mana jika programmer tidak jeli dalam melakukan pemrograman dan kurang memperhatikan isu dan konfigurasi PHP
5. Kode PHP bisa dibaca semua orang, dan kompilasi hanya bisa dilakukan dengan tool yang mahal dari Zend (\$2000).

1.5.4 Sejarah MySQL



Gambar 1. 4 Logo MySQL

MySQL adalah tool yang digunakan khusus untuk mengolah SQL (Structured Query Language). SQL sendiri merupakan sebuah bahasa yang digunakan untuk mengakses baris data relasi. Mudah-mudahan adalah untuk mengakses bahasa dalam komputer. Karena SQL dan MySQL sifatnya khusus, maka hanya orang-orang yang berkecimpung dalam dunia IT lah yang familiar dengan bahasa ini.

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia.

TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggap tidak cocok untuk database yang dinamis seperti web. TcX kemudian mencoba mencari alternatif sistem database lainnya, salah satunya adalah mSQL (miniSQL). Namun mSQL versi 1 ini juga memiliki kekurangan, yaitu tidak mendukung indexing, sehingga performanya tidak terlalu bagus.

Dengan tujuan memperbaiki performa mSQL, Monty mencoba menghubungi David Hughes (programmer yang mengembangkan mSQL) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di mSQL yang dapat dihubungkan dengan UNIREG ISAM sehingga mendukung indexing. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi indexing yang independen untuk mSQL versi 2.

Dikarenakan penolakan tersebut, David Hughes, TcX (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem database baru. Sistem ini merupakan gabungan dari UNIREG dan mSQL (yang source codenya dapat bebas digunakan). Sehingga pada May 1995, sebuah RDBMS baru, yang dinamakan MySQL dirilis.

David Axmark dari Detron HB, rekanan TcX mengusulkan agar MySQL di 'jual' dengan model bisnis baru. Ia mengusulkan agar MySQL dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya di dapat dari menjual jasa "support" untuk perusahaan yang ingin mengimplementasikan MySQL. Konsep bisnis ini sekarang dikenal dengan istilah Open Source.

Pada tahun 1995 itu juga, TcX berubah nama menjadi MySQL AB, dengan Michael Widenius, David Axmark dan Allan Larsson sebagai pendirinya. Titel "AB" di belakang MySQL, adalah singkatan dari "Aktiebolag", istilah PT (Perseroan Terbatas) bagi perusahaan Swedia.

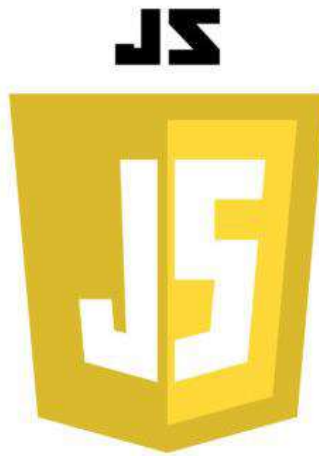
Kelebihan dari MySQL adalah :

1. MySQL Merupakan Multi-user
2. Dapat Diintegrasikan Dengan Bahasa Pemrograman Lainnya.
3. Tidak Perlu Memiliki RAM yang Besar
4. Software yang Portable
5. Struktur Tabelnya Lebih Fleksibel
6. DBMS yang Open Source
7. Bisa Digunakan Dengan Spesifikasi Hardware Yang Rendah
8. Administrative Tools Sangat Lengkap
9. Tipe Data-nya yang Bervariasi.
10. Keamanan yang Terjamin.
11. Interface yang Baik.
12. Performa yang Sangat Baik.

Kekurangan dari MySQL adalah :

1. Technical Support Kurang Baik.
2. Sulit Diaplikasikan dengan Database yang Besar.
3. Tidak Populer Untuk Aplikasi Game dan Mobile.

1.5.5 Sejarah Java Script



Gambar 1. 5 Logo Java Script

JavaScript dibuat dan didesain selama sepuluh hari oleh Brandan Eich, seorang karyawan Netscape, pada bulan September 1995. Awalnya bahasa pemrograman ini disebut Mocha, kemudian diganti ke Mona, lalu LiveScript sebelum akhirnya resmi menyandang nama JavaScript. Versi pertama dari bahasa ini hanya terbatas di kalangan Netscape saja. Fungsionalitas yang ditawarkan pun terbatas. Namun, JavaScript terus dikembangkan oleh komunitas developer yang tak henti-hentinya mengerjakan bahasa pemrograman ini.

Pada tahun 1996, JavaScript secara resmi disebut sebagai ECMAScript, di mana ECMAScript 2 diluncurkan pada tahun 1998 dan ECMAScript 3 diperkenalkan pada tahun 1999. ECMAScript tersebut dikembangkan hingga akhirnya menjadi JavaScript sebagaimana yang kita kenal saat ini. Tak hanya lintas browser, JavaScript juga bisa digunakan di berbagai perangkat, termasuk perangkat mobile dan komputer.

Sejak saat itu, JavaScript terus bertumbuh dan berkembang. Pada akhirnya di tahun 2016, sebanyak 92% website diketahui menggunakan

JavaScript. Hanya dalam kurun waktu dua puluh tahun, JavaScript telah beralih dari bahasa pemrograman yang serba terbatas dan ‘primitif’ menjadi salah satu tool terpenting bagi web developer. Jika sebagian besar waktu dihabiskan untuk berselancar di internet, maka Anda pastinya sudah sangat familiar dengan JavaScript.

Kelebihan dari JavaScript adalah :

1. Anda tidak membutuhkan compiler karena web browser mampu menginterpretasikannya dengan HTML
2. Lebih mudah dipelajari jika dibandingkan dengan bahasa pemrograman lainnya
3. Error atau kesalahan lebih mudah dicari dan ditangani
4. Dapat dialih tugaskan ke elemen halaman web atau even tertentu, misalnya klik atau mouseover.
5. JS dapat digunakan di berbagai browser, platform, dan lain-lain.
6. Dapat menggunakan JavaScript untuk memvalidasi input dan mengurangi keinginan untuk mengecek data secara manual.
7. Dengan JavaScript, website Anda menjadi lebih interaktif dan juga mampu menarik perhatian lebih banyak pengunjung.
8. Lebih cepat dan ringan jika dibandingkan dengan bahasa pemrograman lainnya.

Kekurangan dari JavaScript adalah :

1. Berisiko tinggi terhadap eksploitasi.
2. Dapat dimanfaatkan untuk mengaktifkan kode berbahaya di komputer pengguna.
3. Tidak selalu didukung oleh berbagai browser dan perangkat
4. JS code snippet agak banyak.

5. Dapat di-render secara berbeda pada masing-masing perangkat yang malah mengarah ke inkonsistensi.

1.5.6 Sejarah Basis Data



Gambar 1. 6 Logo Basis Data

Konsep basis data (database) berawal dari proses penyimpanan berkas secara manual. Sistem pemrosesan manual (berbasis kertas) merupakan bentuk pemrosesan yang menggunakan dasar berupa setumpuk data yang disimpan pada lemari arsip. Jika suatu berkas diperlukan, berkas tersebut harus dicari pada lemari arsip tersebut.

Salah satu tujuan penggunaan komputer adalah sebagai tempat penyimpanan dan pengelolaan data. Pada awal tahun 1960, Charles Bachman di perusahaan General Electric merancang DBMS generasi pertama yang disebut Penyimpanan Data Terintegrasi (Integrated Data Store). Pada akhir tahun 1960-an, IBM mengembangkan system manajemen informasi (Information Manajemen System) DBMS. IMS dibentuk dari representasi data pada kerangka kerja yang disebut model data hierarki.

Pada tahun 1970, Edgar Codd di laboratorium penelitian di San Jose mengusulkan suatu representasi data baru yang disebut model data relasional.

Pada tahun 1980, model relasional menjadi DBMS paling dominan. Bahasa query SQL di standarisasi pada akhir tahun 1980. Sejak saat itu mulailah dikembangkan sistem basis data di berbagai bidang. Pengembangan di bidang basis data meliputi bahasa query yang powerful, model data yang lengkap dan penekanan pada dukungan analisis data yang kompleks di semua bagian organisasi.

1.5.6.1 Pengertian Basis Data

Basis data adalah kumpulan data yang disimpan dalam komputer atau media penyimpanan dengan cara tertentu sehingga kumpulan data tersebut dapat dengan mudah untuk digunakan atau ditampilkan. Penyimpanan data dengan cara-cara tertentu dilakukan agar proses penggunaan atau modifikasi data dapat dilakukan dengan mudah dan terkontrol. Basis data dapat digunakan oleh satu atau lebih program aplikasi secara optimal.

Basis Data dalam kehidupan sehari-hari dapat dibayangkan sebagai lemari arsip. Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan atau pengelolaan data. Perbedaannya adalah pada lemari arsip menggunakan bahan besi atau kayu sebagai wadah penyimpanan data. Data yang disimpan pada lemari arsip pun biasanya berupa data dalam lembaran-lembaran kertas yang kemudian dikelompokkan sesuai kebutuhan dalam map-map. Sedangkan basis data menggunakan suatu media penyimpanan (disk) sebagai wadah untuk menyimpan data. Data dalam basis data berbentuk file dalam komputer yang juga dapat dikelompokkan sesuai kebutuhan.

Dari penjelasan tentang definisi basis data, dapat disimpulkan bahwa basis data memiliki sifat sebagai berikut :

- a. Bersifat data oriented, bukan program oriented

- b. Dapat digunakan oleh beberapa program aplikasi
- c. Dapat berkembang dengan mudah, baik volume maupun strukturnya
- d. Meminimalisir kerangkapan data
- e. Up to date, informasi yang tersedia selalu berubah dan akurat
- f. Pengontrolan data terpusat.

Untuk dapat mengelola basis data dalam komputer dibutuhkan suatu perangkat lunak yang disebut sistem manajemen basis data (database management system, DBMS). Salah satu DBMS yang paling berkembang dan paling sering digunakan hingga saat ini adalah DBMS dengan model Relasional atau RDBMS (Relational Database Management System). Beberapa contoh RDBMS diantaranya : MySQL, Ms. Access, Oracle, Foxpro dan PostgreSQL.

1.5.6.2 Pengertian Sistem Basis Data

Sistem basis data adalah sistem yang terdiri atas kumpulan tabel data yang saling berhubungan dan kumpulan program yang memungkinkan beberapa pemakai atau program lain untuk mengakses dan memanipulasi tabel tabel data tersebut.

1.5.6.3 Komponen Sistem Basis Data

Sistem basis data adalah suatu sistem yang dibangun oleh beberapa komponen diantaranya ada enam komponen pokok antara lain :

- a. Perangkat keras (hardware).

Perangkat keras yang biasanya terdapat dalam sistem basis data adalah Komputer, memori (tempat penyimpanan) dan perangkat komunikasi (untuk sistem jaringan).

- b. Sistem Operasi (operating system).

Sistem operasi merupakan perangkat lunak sistem yang mengelola sumber daya perangkat keras dan perangkat lunak aplikasi didalamnya. Sistem operasi berperan sebagai jembatan antara pengguna dengan perangkat keras ataupun perangkat lunak aplikasi komputer sehingga pengguna dapat mengoperasikan komputer. Sistem operasi yang banyak digunakan diantaranya adalah sistem operasi Microsoft Windows dan Linux.

c. Perangkat Lunak (software).

Perangkat lunak digunakan untuk mendukung proses pengelolaan basis data. Misalnya Sistem Informasi Akademik berbasis web, dalam proses pembuatannya tidak cukup hanya menggunakan DBMS MySQL, tetapi diperlukan software bahasa pemrograman ASP, PHP atau Javascript dan software lain untuk membuat desain tampilannya.

d. Basis Data.

Suatu sistem basis data terbentuk dari satu basis data atau lebih yang didalamnya terdapat objek-objek basis data, seperti tabel, query, indeks dan lain-lain.

e. DBMS (Database Management System)

DBMS atau database management system adalah program aplikasi khusus yang dirancang untuk membuat dan juga mengelola database yang tersedia. Sistem ini berisi koleksi data dan set program yang digunakan untuk mengakses database tersebut.

DBMS adalah software yang berperan dalam mengelola, menyimpan, dan mengambil data kembali. Adapun mekanisme yang digunakan sebagai pelengkap adalah pengaman data, konsistensi data dan pengguna data bersama. Beberapa perangkat lunak yang termasuk DBMS adalah : Oracle, MySql dan Ms.Access.

f. Pemakai (user)

Pengguna yang terlibat dalam pengelolaan dan penggunaan basis data.

User basis data dikelompokkan menjadi :

- Database Administrator
Pengguna yang mengorganisasi seluruh sistem basis data. Tugas seorang database administrator, diantaranya : mengatur hak akses user basis data, melakukan koordinasi dan monitoring basis data, serta bertanggung jawab terhadap kebutuhan hardware dan software.
- Database Designer
Pengguna yang bertugas merancang dan mengembangkan database. Tugas seorang database designer adalah mengidentifikasi data yang tersimpan dalam database dan menentukan struktur data yang tepat untuk disimpan dalam database. Database designer memerlukan koordinasi akan kebutuhan user database.
- Application Programmer
Pengguna yang berinteraksi dengan basis data melalui Data Manipulation Language (DML) yang disertakan dalam pemrograman induk yang dipakai. Contoh : Seorang programmer menyertakan kode SQL dalam bahasa pemrograman PHP dalam membangun sebuah web.
- End user
Pengguna yang memanfaatkan atau membutuhkan akses ke database melalui query, menambah, merubah menghapus maupun membuat report database.

- System Analyst

Pengguna yang merencanakan dan menentukan kebutuhan sebelum dilakukan suatu pembangunan sistem yang menggunakan basis data didalamnya.

- g. Aplikasi atau Perangkat Lain

Aplikasi ini tergantung kebutuhan, pemakai basis data bisa dibuatkan program khusus untuk melakukan pengisian, pengubahan atau pengambilan data yang mudah dalam pemakaiannya. Program tersebut ada yang tersedia langsung dalam DBMS atau dibuat menggunakan aplikasi lain seperti misalnya Visual Basic.

1.5.6.4 Bahasa Basis Data

Bahasa database merupakan bahasa data yang dapat ditempelkan kedalam bahasa pemrograman yang lain, sebut saja Java, Pascal, Fortran dst. Bahasa dimana instruksi data base menempel disebut inang. Beberapa komponen Bahasa data base menurut fungsinya dibagi menjadi:

- a. Data Definition Language

Data definition language adalah sekumpulan definisi yang disimpan di dalam data dictionary.

- b. Data Manipulation Language

Data Manipulation Language berisi akumulasi dari operasi manipulasi basis data yang dilakukan. Ini biasa disebut dengan bahasa query sebab biasanya digunakan untuk meminta informasi yang ada dari basis data tersebut.

1.5.6.5 Fungsi Dan Tujuan Basis Data

Fungsi basis data cukup banyak dan cakupannya pun luas dalam mendukung keberadaan lembaga atau organisasi, diantaranya adalah:

a. *Ketersediaan/ Availability*

Fungsi basis data yang pertama adalah untuk menyediakan data-data penting saat sedang diperlukan. Ya, ini adalah fungsi penting dari basis data yang meskipun tidak terletak dalam satu lokasi, dan tersimpan dalam bentuk disk, akan tetapi dengan cara penyimpanan yang sistematis, informasi tersebut mudah untuk didapatkan.

b. *Mudah dan Cepat/ Speed*

Selanjutnya, fungsi dari basis data ini adalah agar Anda sebagai pengguna bisa dengan mudah mengaksesnya saat sedang membutuhkan. Tidak perlu tunggu nanti, apalagi harus mengalokasikan waktu tertentu untuk memanggilnya.

c. *Kelengkapan/ Completeness*

Basis data harus menyimpan data yang lengkap, yang bisa melayani keperluan penggunanya secara keseluruhan. Meski kata lengkap yang dipakai disini sifatnya relatif, namun setidaknya data tersebut membantu memudahkan untuk menambah koleksi data, dan menjamin mudahnya pengguna untuk memodifikasi struktur data yang ada, sebut saja field-field data yang tersedia.

d. *Accuracy dan Security*

Fungsi data base selanjutnya adalah untuk accuracy atau keakuratan. Jadi, agar kesalahan dapat ditekan semaksimal mungkin, Anda bisa lakukan pengorganisasian file-file database dengan baik untuk menghindari kesalahan pada proses data entry dan juga dalam proses penyimpanan atau datastore.

Selain itu, fungsi database adalah untuk security atau keamanan. Ada fasilitas pengaman data yang disediakan oleh sistem basis data yang baik sehingga data tidak bisa dimodifikasi, diakses, diubah

maupun dihapus oleh yang tidak mendapatkan hak untuk melakukannya.

e. Storage Efficiency

Pengorganisasian data dilakukan dengan baik dengan tujuan untuk menghindari duplikasi data yang berpengaruh pada bertambahnya ruang penyimpanan dari basis data tersebut. pengkodean dan juga relasi data bermanfaat untuk menghemat space penyimpanan dalam basis data.

1.6 Framework

Framework disebut sebagai kerangka kerja. Para developer menggunakan framework untuk memudahkan mereka dalam membuat dan mengembangkan aplikasi atau software. Pada framework itu sendiri berisi kumpulan fungsi-fungsi dasar atau perintah yang biasa digunakan dalam mengembangkan suatu software, dengan harapan agar software yang dibangun menjadi lebih cepat dan terstruktur.

Ada juga yang mengatakan bahwa framework adalah komponen-komponen yang sudah siap dipakai oleh developer untuk menangani berbagai permasalahan dalam pemrograman, seperti pemanggilan variabel, file, koneksi ke database dan sebagainya. Dengan begitu, developer akan lebih fokus dan mampu menyelesaikan software menjadi lebih cepat dan efektif.

1.6.1 Fungsi Framework

1. Menghemat Waktu Pengembangan

Penggunaan framework dalam pengembangan suatu software akan mengurangi beban kerja developer, sehingga tidak ada waktu yang terbuang untuk memikirkan fungsi-fungsi umum yang akan digunakan. Selain itu, developer akan lebih fokus pada alur cerita pada

aplikasi seperti yang dibutuhkan oleh pengguna. Dengan begitu, waktu pengembangan software akan berjalan lebih cepat dan dapat diserahkan kepada pengguna sesuai dengan waktu yang telah disepakati bersama.

2. Pemrograman menjadi Lebih Terstruktur

Ketika developer menghadapi suatu proyek pengembangan software yang besar, maka akan terdapat banyak program yang ditulis didalamnya. Terkadang, semakin banyaknya program akan menjadikan proses debugging semakin lambat. Selain itu, untuk mengecek kode program yang menjadi penyebab terjadinya error akan semakin sulit karena program tidak terstruktur dengan baik. ada framework yang menerapkan konsep MVC (Model View Control) yang memudahkan developer untuk memisahkan dan menyusun program berdasarkan bagiannya, yaitu Model, View ataupun Control.

3. Pengulangan Kode

Seperti yang sudah disinggung pada poin pertama bahwa pembuatan software membutuhkan waktu yang lama, apalagi Anda berperan sebagai programmer tunggal. Maka Anda dapat menggunakan framework untuk mengurangi beban tersebut.

Sebab, framework sudah menanamkan berbagai fungsi-fungsi umum yang bisa Anda gunakan tanpa harus mengulangi pembuatan kode dari awal. Pastinya, akan memudahkan Anda juga untuk menggunakannya kembali di proyek-proyek berikutnya.

4. Meningkatkan Keamanan

Keamanan menjadi suatu hal yang sangat vital dalam pengembangan software atau aplikasi. Apalagi, software yang memuat data pengguna yang privasinya harus dilindungi. Disini, framework terus

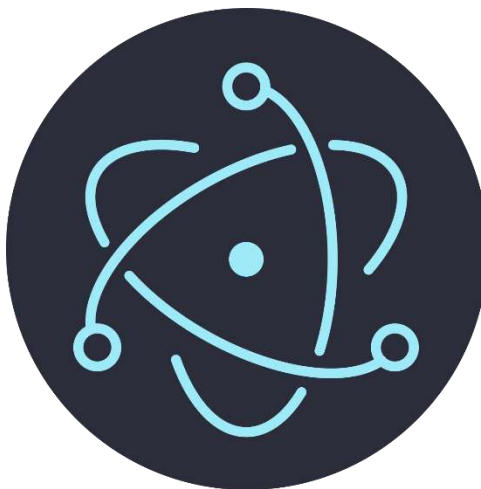
memperbarui versinya yaitu menawarkan fitur yang handal dalam menangani berbagai jenis ancaman yang menyerang sistem keamanan.

1.6.2 Jenis – Jenis Framework

Terdapat dua jenis *framework* yaitu *Desktop Framework* dan *Web Framework*. Kedua jenis *framework* tersebut memiliki kegunaannya masing-masing. *Desktop framework* merupakan *framework* yang digunakan untuk membangun aplikasi berbasis desktop, sedangkan *web framework* merupakan *framework* yang digunakan untuk membangun aplikasi berbasis web.

1. Desktop Framework

a. Electron



Gambar 1. 7 Logo Framework Electron

Electron merupakan suatu *framework JavaScript* yang digunakan untuk mengembangkan aplikasi berbasis desktop. Pada umumnya, *JavaScript* identik dengan pembuatan website. Oleh karena itu, *JavaScript* yang digunakan *electron* ini dirancang dengan menggunakan teknologi HTML, CSS dan *JavaScript*. Padahal sebenarnya, *electron* juga dapat ditulis dengan menggunakan bahasa C++. Beberapa aplikasi desktop

populer yang dibangun oleh *Electron* seperti *Atom*, *Slack*, *WordPress*, *Skype* dan *Github* desktop.

b. *Proton Native*



Gambar 1. 8 Logo Framework Proton Native

Proton Native merupakan *framework JavaScript* yang dikembangkan oleh Gustav Hansen untuk merancang aplikasi berbasis desktop. *Proton Native* memiliki fitur *native tools* yang mempunyai ukuran yang kecil dan penggunaan sumber kekuatan yang lebih sedikit.

2. *Web Framework*

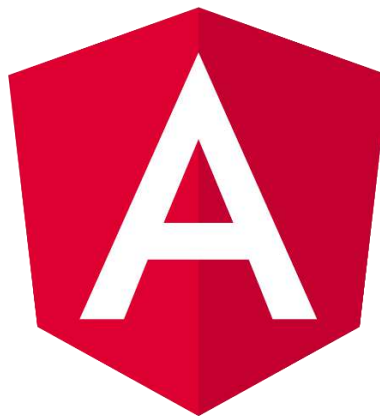
a. Django



Gambar 1. 9 Logo Framework Django

Django merupakan suatu framework yang berbasis Python dengan menggunakan pola MTV, yaitu Model, Template dan View. Django menawarkan berbagai fitur untuk mengembangkan aplikasi web yang berkualitas, terutama bagian keamanan yang digunakan untuk mencegah terjadinya eksekusi program dilapisan Template. Web framework ini sudah banyak digunakan oleh perusahaan besar seperti Disqus, Pinterest, Instagram dan Quora.

b. Angular JS



Gambar 1. 10 Logo Framework Angular JS

Angular JS merupakan framework JavaScript yang sengaja dirilis oleh Google dalam mengembangkan aplikasi web. Bahkan, angular dapat membangun halaman web di sisi client dengan kinerja yang tinggi. Framework ini memiliki konsep MVC (Model, View, Controller). Web framework ini sudah banyak digunakan oleh perusahaan besar seperti Youtube on PS3, Weather dan Netflix.

c. Laravel



Gambar 1. 11 Logo Framework Laravel

Laravel merupakan *framework* MVC yang dibuat oleh Taylor Otwell pada 2011 dengan menggunakan PHP. Framework ini adalah pengembangan *website* berbasis MVP yang dirancang untuk meningkatkan kualitas perangkat lunak, yaitu dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, meningkatkan pengalaman bekerja dengan aplikasi yang menyediakan sintaks ekspresif, jelas dan menghemat waktu. *Web framework* ini sudah banyak digunakan oleh perusahaan seperti *Deltanet Travel* dan *Neighbourhood Lender*.

d. Rails



Gambar 1. 12 Logo Framework Rails

Rails adalah web framework Model-View-Controller yang ditulis oleh David Heinemeier Hansson. Selain itu, Rails mencakup semua yang diperlukan untuk membuat aplikasi web berbasis database menggunakan Ruby. Web framework ini sudah banyak digunakan oleh perusahaan seperti AirBnB, GitHub, UrbanDictionary, GroupOn dan Shopify.

e. Spring



Gambar 1. 13 Logo *Framework Spring*

Spring adalah web framework Model-View-Controller yang dikembangkan oleh Pivotal Software. Framework ini dirancang untuk menciptakan aplikasi web berbasis JVM yang sederhana, portabel, cepat dan fleksibel. Web framework ini sudah banyak digunakan oleh perusahaan seperti Mascus dan Allocine.

f. *Codeigniter*



Gambar 1. 14 Logo *Framework Codeigniter*

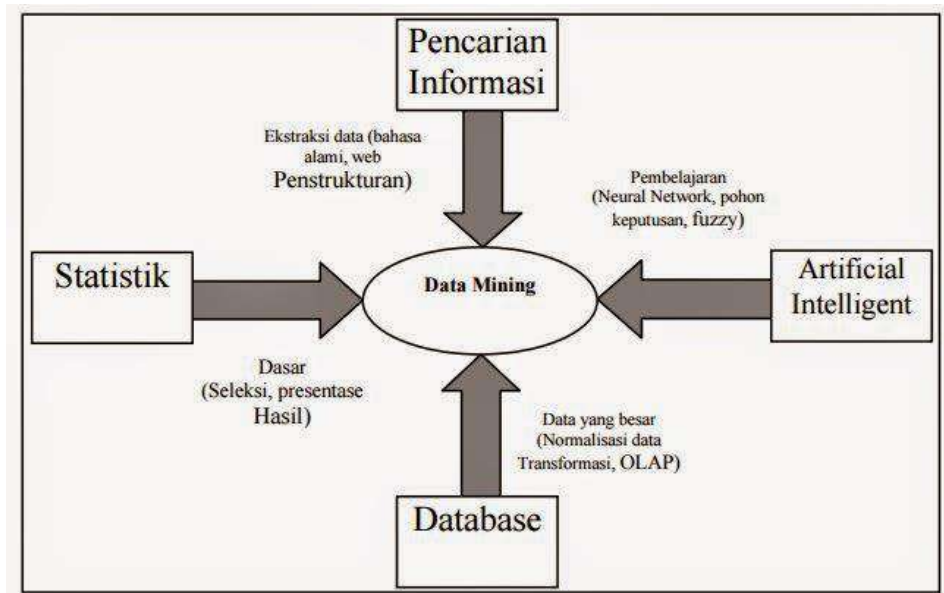
Codeigniter merupakan framework yang memiliki konsep MVC (Model, View, Controller) untuk membangun sebuah website yang dinamis menggunakan PHP. Dengan menggunakan *codeigniter*, developer akan semakin cepat dalam membangun aplikasi web, walaupun memulainya dari awal. Web framework ini sudah banyak digunakan oleh perusahaan seperti Bufferapp dan The Mail and Guardian.

BAB II

DEFINISI

2.1 *Data Mining*

Data mining adalah proses yang menggunakan teknik statistic, matematika, kecerdasan buatan dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dengan database besar. *Data Mining* melakukan proses pemodelan tanpa supervisi (*unsupervised*) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi.



Gambar 2. 1 Bidang Ilmu Pada Data Mining

2.2.1 Pengelompokkan Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan yaitu :

1. Deskripsi

Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi kecuali variable target estimasi lebih ke arah numerik daripada ke arah kategori.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi kecuali dalam prediksi nilai dari hasil akan ada dimasa mendatang.

4. Klasifikasi

Dalam klasifikasi terdapat target variable kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam 3 kategori, yaitu pendapatan tinggi, pendapatan sedang dan pendapatan rendah.

5. Pengklusteran

Pengklusteran merupakan pengelompokkan record, pengamatan atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan.

2.2 Clustering

Clustering atau klasterisasi adalah metode pengelompokan data. Menurut Tan, 2006 clustering adalah sebuah proses untuk mengelompokkan data ke dalam beberapa cluster atau kelompok sehingga data dalam satu cluster memiliki tingkat kemiripan yang maksimum dan data antar cluster memiliki kemiripan yang minimum.

Analisis Pengelompokan / *Clustering* merupakan proses membagi data dalam suatu himpunan ke dalam beberapa kelompok yang kesamaan datanya dalam suatu kelompok lebih besar daripada kesamaan data tersebut dengan data dalam kelompok lain. Potensi *clustering* adalah dapat digunakan untuk mengetahui struktur dalam data yang dapat dipakai lebih lanjut dalam berbagai aplikasi secara luas seperti klasifikasi, pengolahan gambar, dan pengenalan pola. Teknik *cluster* mempunyai dua metode untuk melakukan pengelompokkan yaitu *hierarchical clustering* dan *non-hierarchical clustering*. *hierarchical clustering* merupakan suatu metode pengelompokkan data yang cara kerjanya dengan mengelompokkan dua data atau lebih yang mempunyai kesamaan atau kemiripan, kemudian proses dilanjutkan ke objek lain yang memiliki kedekatan dua, proses ini terus berlangsung hingga *cluster* membentuk semacam *tree* dimana ada hirarki atau tingkatan yang jelas antar objek dari yang paling mirip hingga yang paling tidak mirip. Namun secara logika semua objek pada akhirnya hanya akan membentuk sebuah *cluster*.

2.2.1 Manfaat Clustering

1. Clustering merupakan metode segmentasi data yang sangat berguna dalam prediksi dan analisa masalah bisnis tertentu. Misalnya Segmentasi pasar, marketing dan pemetaan zonasi wilayah.
2. Identifikasi obyek dalam bidang berbagai bidang seperti computer vision dan image processing.

2.2.2 Syarat Clustering

Menurut Han dan Kamber, 2012, syarat sekaligus tantangan yang harus dipenuhi oleh suatu algoritma clustering adalah:

1. Skalabilitas
Suatu metode clustering harus mampu menangani data dalam jumlah yang besar. Saat ini data dalam jumlah besar sudah sangat umum digunakan dalam berbagai bidang misalnya saja suatu database. Tidak hanya berisi ratusan objek, suatu database dengan ukuran besar bahkan berisi lebih dari jutaan objek.
2. Kemampuan analisa beragam bentuk data
Algoritma klasteriasi harus mampu dimplementasikan pada berbagai macam bentuk data seperti data nominal, ordinal maupun gabungannya.
3. Menemukan cluster dengan bentuk yang tidak terduga
Banyak algoritma clustering yang menggunakan metode Euclidean atau Manhattan yang hasilnya berbentuk bulat. Padahal hasil clustering dapat berbentuk aneh dan tidak sama antara satu dengan yang lain. Karenanya dibutuhkan kemampuan untuk menganalisa cluster dengan bentuk apapun pada suatu algoritma clustering.
4. Kemampuan untuk dapat menangani noise

Data tidak selalu dalam keadaan baik. Ada kalanya terdapat data yang rusak, tidak dimengerti atau hilang. Karena system inilah, suatu algoritma clustering dituntut untuk mampu menangani data yang rusak.

5. Sensitifitas terhadap perubahan input

Perubahan atau penambahan data pada input dapat menyebabkan terjadi perubahan pada cluster yang telah ada bahkan bisa menyebabkan perubahan yang mencolok apabila menggunakan algoritma clustering yang memiliki tingkat sensitifitas rendah.

6. Mampu melakukan clustering untuk data dimensi tinggi

Suatu kelompok data dapat berisi banyak dimensi ataupun atribut. Untuk itu diperlukan algoritma clustering yang mampu menangani data dengan dimensi yang jumlahnya tidak sedikit.

7. Interpretasi dan kegunaan

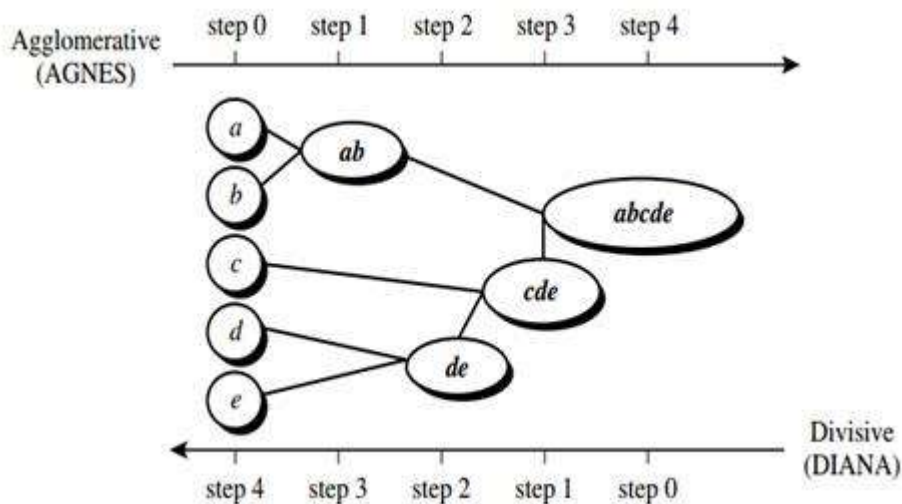
Hasil dari clustering harus dapat diinterpretasikan dan berguna.

2.2.3 Metode Clustering

Metode *clustering* secara umum dapat dibagi menjadi dua yaitu *hierarchical clustering* dan *partitional clustering* (Tan, 2011). Sebagai tambahan, terdapat pula metode Density-Based dan Grid-Based yang juga sering diterapkan dalam implementasi *clustering*. Berikut penjelasannya:

1. Hierarchical *clustering*

Pada *hierarchical clustering* data dikelompokkan melalui suatu bagan yang berupa hirarki, dimana terdapat penggabungan dua grup yang terdekat disetiap iterasinya ataupun pembagian dari seluruh set data kedalam *cluster*.



Gambar 2. 2 Hierarchical Clustering

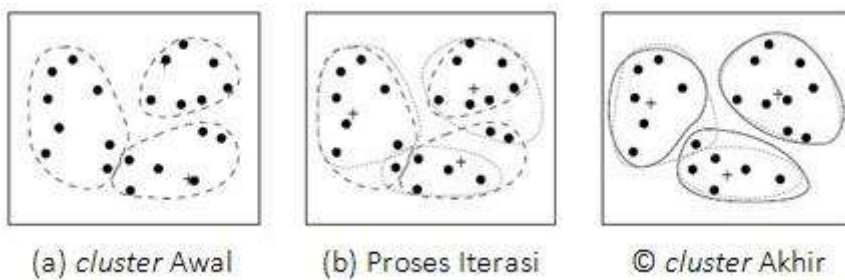
Langkah melakukan *Hierarchical clustering*:

1. Identifikasi *item* dengan jarak terdekat
2. Gabungkan *item* itu kedalam satu *cluster*
3. Hitung jarak antar *cluster*
4. Ulangi dari awal sampai semua terhubung

Contoh metode hierarchy clustering: *Single Linkage, Complete Linkage, Average Linkage, Average Group Linkage*.

2. *Partitional clustering*

Partitional clustering yaitu data dikelompokkan ke dalam sejumlah *cluster* tanpa adanya struktur hirarki antara satu dengan yang lainnya. Pada metode *partitional clustering* setiap *cluster* memiliki titik pusat *cluster* (centroid) dan secara umum metode ini memiliki fungsi tujuan yaitu meminimumkan jarak (*dissimilarity*) dari seluruh data ke pusat *cluster* masing-masing. Contoh metode *partitional clustering*: K-Means, Fuzzy K-means dan *Mixture Modelling*.



Gambar 2. 3 Partitional Clustering

2.3 Algoritma *K-Means*

Metode K-means merupakan metode clustering yang paling sederhana dan umum. Hal ini dikarenakan K-means mempunyai kemampuan mengelompokkan data dalam jumlah yang cukup besar dengan waktu komputasi yang cepat dan efisien. K-Means merupakan salah satu algoritma klastering dengan metode partisi (partitioning method) yang berbasis titik pusat (centroid) selain algoritma k-Medoids yang berbasis obyek. Algoritma ini pertama kali diusulkan oleh MacQueen (1967) dan dikembangkan oleh Hartigan dan Wong tahun 1975 dengan tujuan untuk dapat membagi M data point dalam N dimensi kedalam sejumlah k cluster dimana proses klastering dilakukan dengan meminimalkan jarak sum squares antara data dengan masing masing pusat cluster (centroid-based). Algoritma k-Means dalam penerapannya memerlukan tiga parameter yang seluruhnya ditentukan pengguna yaitu jumlah cluster k, inisialisasi klaster, dan jarak system, Biasanya, k-Means dijalankan secara independen dengan inisialisasi yang berbeda menghasilkan cluster akhir yang berbeda karena algoritma ini secara prinsip hanya mengelompokkan data menuju local minimal. Salah satu cara untuk mengatasi local minima adalah dengan mengimplementasikan algoritma k-Means, untuk K yang diberikan, dengan beberapa nilai initial

partisi yang berbeda dan selanjutnya dipilih partisi dengan kesalahan kuadrat terkecil (Jain, 2009).

K-means merupakan salah satu algoritma *clustering*. Tujuan algoritma ini yaitu untuk membagi data menjadi beberapa kelompok. Algoritma ini menerima masukan berupa data tanpa label kelas. Hal ini berbeda dengan supervised learning yang menerima masukan berupa vektor $(-x-1, y1)$, $(-x-2, y2)$, ..., $(-x-i, yi)$, di mana x_i merupakan data dari suatu data pelatihan dan y_i merupakan label kelas untuk x_i .

Pada algoritma pembelajaran ini, komputer mengelompokkan sendiri data-data yang menjadi masukannya tanpa mengetahui terlebih dulu target kelasnya. Pembelajaran ini termasuk dalam unsupervised learning. Masukan yang diterima adalah data atau objek dan k buah kelompok (cluster) yang diinginkan. Algoritma ini akan mengelompokkan data atau objek ke dalam k buah kelompok tersebut. Pada setiap cluster terdapat titik pusat (centroid) yang merepresentasikan cluster tersebut.

. *K-means* merupakan salah satu metode *clustering* non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*. Metode ini mempartisi data ke dalam *cluster* sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu *cluster* yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam *cluster* yang lain.

Algoritma *K-means* merupakan algoritma pembelajaran yang sederhana yang dapat menyelesaikan sebuah permasalahan untuk meminimalkan kesalahan ganda.

2.3.1 Cara Kerja Algoritma *K-means*

Algoritma *K-means* merupakan metode pengclusteran secara *partitioning* yang memisahkan data ke dalam kelompok yang berbeda. Dengan *partitioning*

secara iteratif, *K-means* mampu meminimalkan rata-rata jarak setiap data ke *clusternya*. Dalam algoritma *K-means*, setiap data harus termasuk ke *cluster* tertentu pada suatu tahapan proses, pada tahapan proses berikutnya dapat berpindah ke *cluster* yang lain. Pada dasarnya penggunaan algoritma *K-means* dalam melakukan proses *clustering* tergantung dari data yang ada dan konklusi yang ingin dicapai. Untuk itu digunakan algoritma *K-means* yang didalamnya memuat aturan sebagai berikut :

- a. Jumlah *cluster* yang perlu di inputkan
- b. Hanya memiliki atribut bertipe numerik

Algoritma *K-means* pada awalnya mengambil sebagian dari banyaknya komponen dari populasi untuk dijadikan pusat *cluster* awal. Pada step ini pusat *cluster* dipilih secara acak atau random dari sekumpulan populasi data yang ada. Selanjutnya *K-means* menguji dari masing-masing komponen didalam populasi data dan menandai komponen tersebut ke salah satu pusat *cluster* yang telah di definisikan tergantung dari jarak minimum antar komponen dengan tiap-tiap pusat *cluster*. Posisi pusat *cluster* nanti akan dihitung kembali sampai semua komponen pada data digolongkan kedalam tiap-tiap *cluster* dan terakhir akan terbentuk sebuah posisi *cluster* baru.

2.3.2 Rumus Algoritma K-means

Tahapan untuk *algoritma K-means* yang digunakan sebagai berikut :

1. Tentukan k sebagai jumlah *cluster* yang akan dibentuk.
2. Tentukan k *centroid* awal secara *random* atau acak.
3. Hitung jarak setiap objek ke masing-masing *centroid* dari masing-masing *cluster* dengan menggunakan metode *Euclidian Distance*.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

4. Alokasikan masing-masing objek ke dalam *centroid* yang paling dekat.
5. Lakukan iterasi, kemudian tentukan posisi *centroid* baru dengan menggunakan rumus sebagai berikut :

$$C = \frac{\sum m}{n}$$

Keterangan :

C : *centroid* data

m : anggota data yang termasuk ke dalam jarak *cluster* terdekat

n : jumlah data yang menjadi anggota *cluster* tertentu.

6. Ulangi langkah 3 jika posisi *centroid* baru tidak sama.

2.3.3 Kelebihan dan Kekurangan Algoritma K-means

A. Kelebihan dari Algoritma K-means antara lain :

1. Mudah untuk diimplementasikan dan dijalankan.
2. Waktu yang dibutuhkan untuk menjalankan pembelajaran ini relatif cepat.
3. Mudah untuk diadaptasi.
4. Umum digunakan.
5. Menggunakan prinsip yang sederhana dapat di jelaskan dalam non-statistik.

B. Kekurangan dari Algoritma K-means antara lain :

1. Apabila hanya ada terdapat beberapa buah titik sampel data yang ada, maka hal yang mudah untuk melakukan penghitungan dan mencari jarak titik terdekat dengan k titik yang telah di lakukan inisialisasi yang secara acak.

2. Adanya penggunaan k buah random, tidak ada jaminan untuk menemukan kumpulan cluster yang optimal.

2.4 Sistem

Sistem merupakan suatu kumpulan atau kesatuan dari jaringan kerja dari prosedur-prosedur yang saling berhubungan, elemen-elemen yang saling berkaitan untuk mencapai suatu tujuan tertentu.

Suatu sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu:

- a) Memiliki komponen, Sistem biasanya terdiri dari sejumlah komponen yang saling berhubungan satu sama lain yang bekerja sama membentuk satu kesatuan. Setiap sistem tidak peduli. betapapun kecilnya, selalu mengandung komponen-komponen. Setiap subsistem memiliki sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.
- b) Lingkungan luar sistem (*environment*), Lingkungan luar sistem merupakan sesuatu di luar batas dari sistem yang mempengaruhi operasi sistem.
- c) Batas sistem (*boundary*), Batas sistem yaitu pembatas antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini bisa menjadi suatu sistem yang dipandang sebagai suatu kesatuan. Batas dari suatu sistem menunjukkan ruang lingkup (*scope*) dari sebuah sistem tersebut.
- d) Penghubung sistem (*interface*), Penghubung sistem yaitu media penghubung antara satu komponen dengan komponen yang lainnya pada suatu *system*.
- e) Masukan sistem (*input*), Masukan sistem merupakan energi yang dimasukkan ke dalam suatu sistem. Masukan juga dapat berupa

sebuah masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* merupakan energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses.

2.4.1 Klasifikasi Sistem

Klasifikasi sistem diantaranya meliputi sebagai berikut:

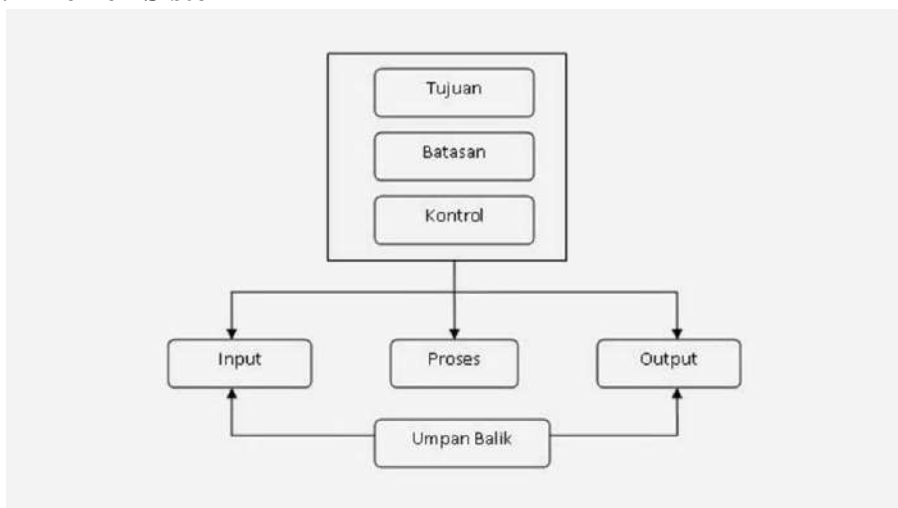
1. Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Salah satu contohnya adalah sistem teologia.
2. Sistem fisik merupakan sistem yang ada secara fisik. Salah satu contohnya adalah sistem komputer, sistem akuntansi, sistem produksi dll.
3. Sistem alamiah merupakan sistem yang terjadi melalui proses alam. Salah satu contohnya adalah sistem matahari, sistem luar angkasa, sistem reproduksi dll.
4. Sistem buatan manusia merupakan sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system*. Salah satu contohnya adalah contoh istem informasi.
5. Sistem tertentu (*deterministic system*) merupakan beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Salah satu contohnya adalah sistem komputer.
6. Sistem tak tentu (*probabilistic system*) merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
7. Sistem tertutup (*close system*) merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini

bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system*.

8. Sistem terbuka (*open system*) merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern.

2.4.2 Elemen Sistem



Gambar 2. 4 Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, berikut penjelasannya sebagai berikut:

1. Tujuan

Setiap sistem memiliki tujuan, entah hanya satu atau lebih. Tujuan inilah yang memotivasi dan mengarahkan sistem. Tanpa tujuan, sistem

menjadi tidak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

2. Masukan

Masukan (*input*) sistem merupakan segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan bisa berupa hal yang berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa pelanggan).

3. Proses

Proses adalah bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai, misalnya berupa informasi dan produk, tetapi juga dapat berupa hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

4. Keluaran

Keluaran (*output*) adalah hasil dari proses. Pada sistem informasi berupa informasi atau laporan, dan lain-lain.

5. Batas

Yang disebut batas sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem.

6. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian diwujudkan dengan menggunakan umpan balik, yang mencuplik keluaran. Umpan balik ini digunakan untuk mengontrol baik masukan maupun proses.

7. Lingkungan

Lingkungan bisa berpengaruh terhadap operasi sistem, dalam arti bisa merugikan atau menguntungkan sistem itu sendiri.

2.4.3 Jenis Sistem

Ada berbagai tipe sistem berdasarkan kategori:

1. Atas dasar keterbukaan:
 - a. Sistem terbuka, di mana pihak luar dapat mempengaruhinya.
 - b. Sistem tertutup.
2. Atas dasar komponen:
 - a. Sistem fisik, dengan komponen materi dan energi.
 - b. Sistem non-fisik atau konsep, berisikan ide-ide.

2.4.4 Sistem Terotomasi

Sistem terotomasi mempunyai sejumlah komponen yaitu diantaranya:

1. Perangkat lunak yang meliputi sistem operasi, sistem database, program pengontrol komunikasi, program aplikasi.
2. Perangkat keras yang meliputi CPU, *disk*, *printer*, *tape*.
3. Personil yang mengoperasikan sistem, menyediakan masukan, mengkonsumsi keluaran dan melakukan aktivitas manual yang mendukung sistem.
4. Data yang harus tersimpan dalam sistem selama jangka waktu tertentu.
5. Prosedur instruksi dan kebijakan untuk mengoperasikan sistem.

Sistem terotomasi terbagi dalam sejumlah kategori :

1. *On-line systems*.

Sistem *on-line* adalah sistem yang menerima langsung *input* pada area, dimana *input* tersebut direkam dan menghasilkan *output* yang dapat

menghasilkan komputasi pada area. Area dapat dipisah-pisah dalam skala. Salah satu contohnya adalah ratusan kilometer. Biasanya digunakan bagi reservasi angkutan udara, reservasi kereta api, perbankan, dll.

2. *Real-time systems.*

Sistem *real-time* adalah mekanisme pengontrolan, pemrosesan, dan perekaman data yang sangat cepat sehingga output yang dihasilkan dapat diterima dalam waktu yang relatif sama. Perbedaannya dengan sistem *on-line* adalah satuan waktu yang digunakan *real-time* biasanya seperseratus atau seperseribu detik, sedangkan *on-line* masih dalam skala detik atau bahkan kadang memerlukan waktu beberapa menit. Perbedaan lainnya, *on-line* biasanya hanya berinteraksi dengan pemakai, sedangkan *real-time* berinteraksi langsung dengan pemakai dan lingkungan yang dipetakan.

3. *Decision support system + strategic planning system.*

Sistem yang memproses transaksi organisasi secara harian dan membantu para manajer mengambil keputusan, mengevaluasi dan menganalisa tujuan organisasi. Digunakan untuk sistem penggajian, sistem pemesanan, sistem akuntansi dan sistem produksi. Biasanya berbentuk paket statistik, paket pemasaran, dll. Sistem ini tidak hanya merekam dan menampilkan data tetapi juga fungsi matematik, data analisa statistik dan menampilkan informasi dalam bentuk grafik, tabel, chart sebagaimana laporan konvensional.

4. *Knowledge-based system.*

Program komputer yang dibuat mendekati kemampuan dan pengetahuan. Umumnya menggunakan perangkat keras dan perangkat lunak seperti PROLOG dan LISP.

BAB III

TOOLS DAN BAHASA PEMROGRAMAN YANG DIGUNAKAN

3.1 Tools Yang Digunakan

3.1.1 Sublime Text

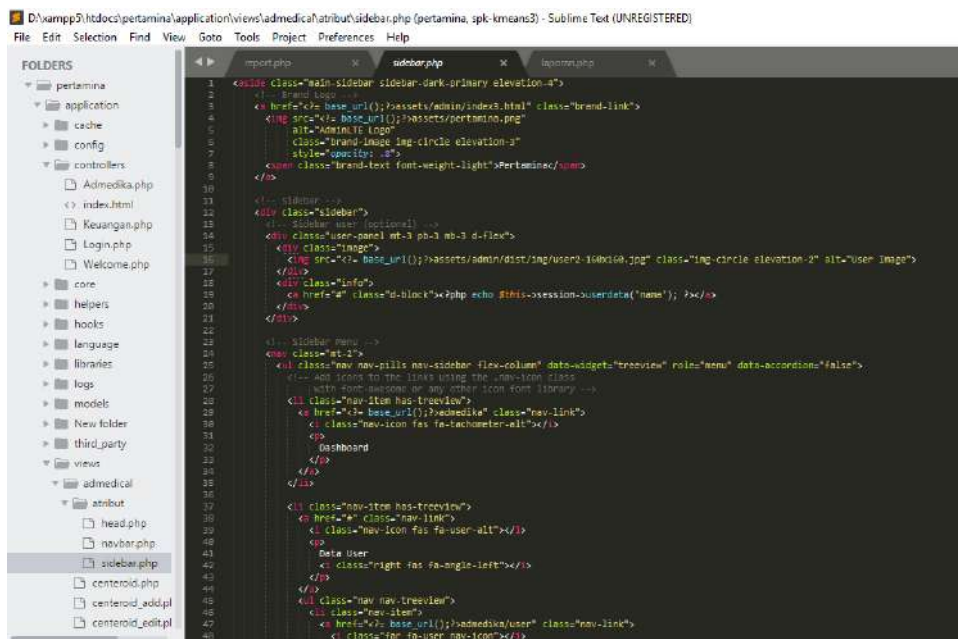


Gambar 3. 1 Logo Sublime Text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages. Sublime Text bukanlah aplikasi open source dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan

mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan pengguna.



Gambar 3. 2 Tampilan *Sublime Text*

Berikut beberapa fitur yang diunggulkan dari aplikasi Sublime Text:

a. Goto Anything

Fitur yang sangat membantu dalam membuka file ataupun menjelajahi isi dari file hanya dengan beberapa keystrokes.

b. Multiple Selections

Fitur ini memungkinkan user untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi file lebih cepat dari sebelumnya.

c. Command Pallette

Dengan hanya beberapa keystrokes, user dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu.

d. Distraction Free Mode

Bila user memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu user dengan memberikan tampilan layar penuh.

e. Split Editing

Dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan editing perpecahan. Mengedit sisi file dengan sisi, atau mengedit dua lokasi di satu file. Anda dapat mengedit dengan banyak baris dan kolom yang user inginkan.

f. Instant Project Switch

Menangkap semua file yang dimasukkan kedalam project pada aplikasi ini. Terintegrasi dengan fitur Goto Anything untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam project lainnya dengan cepat.

g. Plugin API

Dilengkapi dengan plugin API berbasis Python sehingga membuat aplikasi ini sangat tangguh.

h. Customize Anything

Aplikasi ini memberikan user fleksibilitas dalam hal pengaturan fungsional dalam aplikasi ini.

i. Cross Platform

Aplikasi ini dapat berjalan hampir disemua operating system modern seperti Windows, OS X, dan Linux based operating system.

3.1.2 XAMPP



Gambar 3. 3 Logo XAMPP

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah cross platform sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris.

Sejarah mencatat, software XAMPP pertama kali dikembangkan oleh tim proyek bernama Apache Friends dan sampai saat ini sudah masuk dalam rilis versi 7.3.9 yang bisa didapatkan secara gratis dengan label GNU (General Public License).

Jika dijabarkan secara gamblang, masing-masing huruf yang ada di dalam nama XAMPP memiliki arti sebagai berikut ini:

a. X = Cross Platform

Merupakan kode penanda untuk software cross platform atau yang bisa berjalan di banyak sistem operasi.

b. A = Apache

Apache adalah aplikasi web server yang bersifat gratis dan bisa dikembangkan oleh banyak orang (open source).

c. 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.

d. 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, contohnya dalam website berbasis CMS WordPress.

e. P = Perl

Sementara itu, 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.

3.1.3 Framework Codeigniter



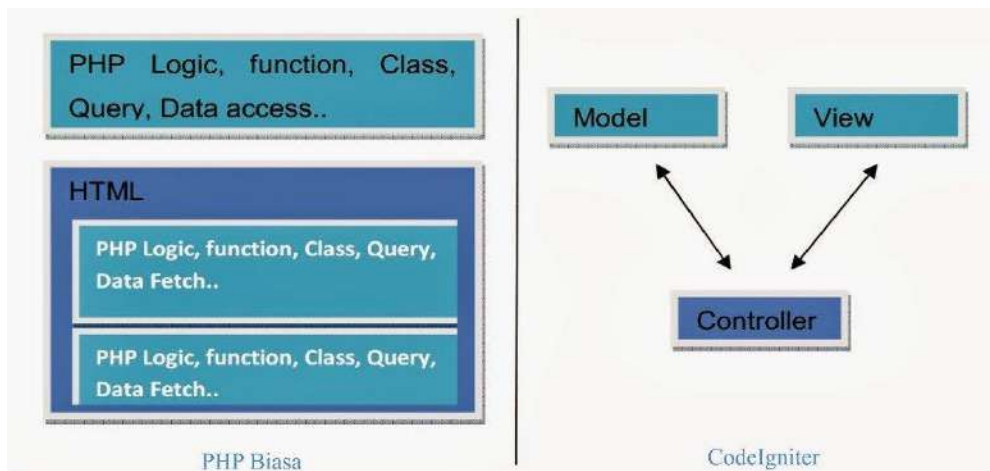
Gambar 3. 4 Logo Codeigniter

CodeIgniter merupakan sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun sebuah aplikasi PHP yang dinamis. Tujuan utama dalam mengembangkan *CodeIgniter* ini adalah untuk membantu para *developer* untuk mengerjakan atau membuat sebuah aplikasi lebih cepat daripada menulis semua *code* dari awal. *CodeIgniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan. *Codeigniter* menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, *Codeigniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih *Codeigniter* sebagai framework pilihannya. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu:

1. View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan

merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.

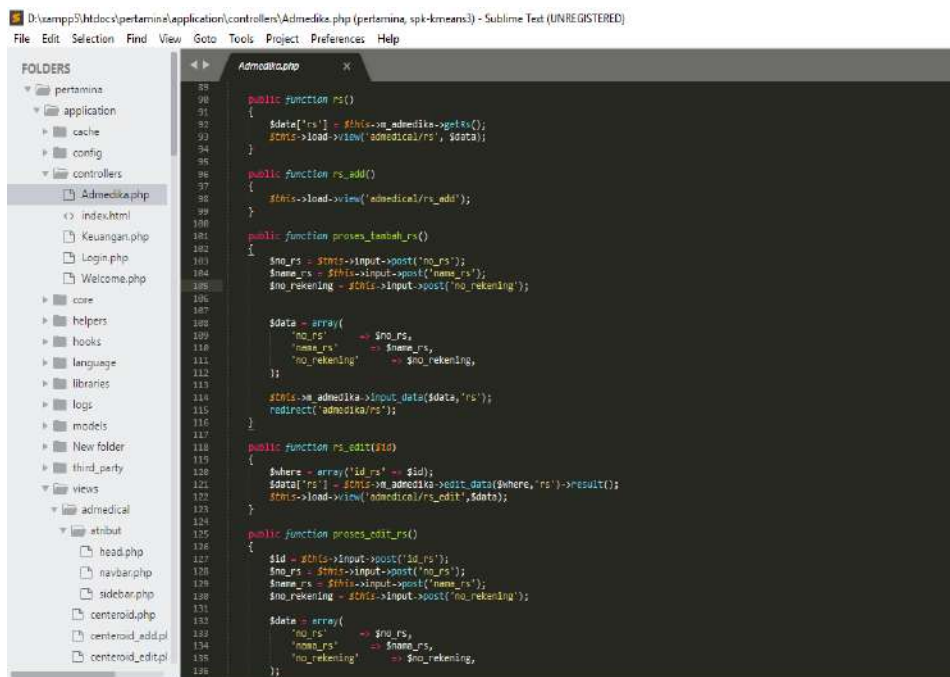
2. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, tetapi tidak dapat berhubungan langsung dengan bagian view.
3. Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.



Gambar 3. 5 Perbandingan PHP Biasa dan *Codeigniter*

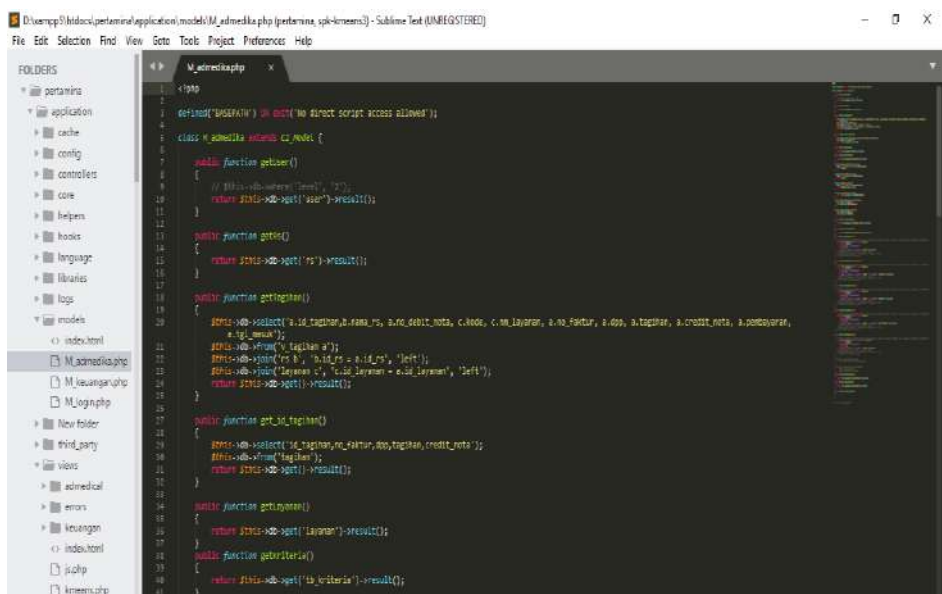
Berikut contoh file untuk Model, View dan Controller

1. *Controller* / Admedika



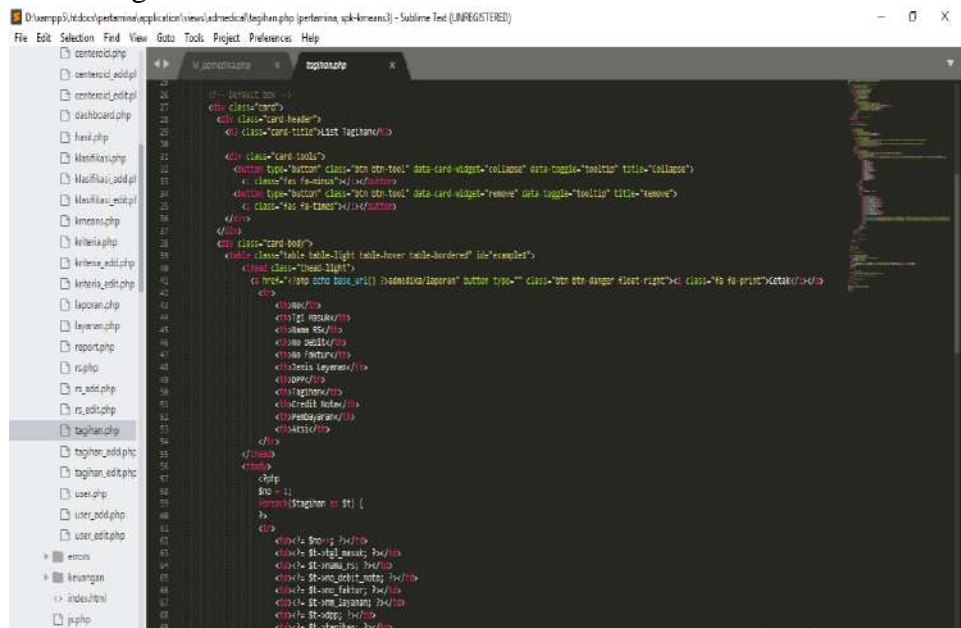
Gambar 3. 6 Tampilan Controller Admedika

2. Model / Admedika



Gambar 3. 7 Tampilan Model Admedika

3. View / Tagihan



Gambar 3. 8 Tampilan View Tagihan

3.1.3.1 Struktur File Framework Codeigniter

Tampilan hasil ekstrak file codeigniter seperti ini :

Nome	Date modified	Type	Size
application	24-Jan-20 1:08 AM	File Folder	
system	24-Jan-20 1:08 AM	File Folder	
user_guide	24-Jan-20 1:08 AM	File Folder	
editorconfig	19-Sep-19 5:08 AM	EDITORCONFIG File	1 KB
.gitignore	19-Sep-19 5:08 AM	Text Document	1 KB
composer.json	19-Sep-19 5:08 AM	JSON File	1 KB
contributing.md	19-Sep-19 5:08 AM	MD File	7 KB
index.php	19-Sep-19 5:08 AM	PHP File	11 KB
license.txt	19-Sep-19 5:08 AM	Text Document	2 KB
readme.rst	19-Sep-19 5:08 AM	RST File	3 KB

Gambar 3. 9 File Hasil Ekstrak

Kegunaan folder :

1. Folder Application : Di folder ini tempat menyimpan file dan konfigurasi dari aplikasi yang kita buat menggunakan framework codeigniter.

Name	Date modified	Type	Size
cache	24-Jan-20 1:08 AM	File folder	
config	24-Jan-20 1:08 AM	File folder	
controllers	24-Jan-20 1:08 AM	File folder	
core	24-Jan-20 1:08 AM	File folder	
helpers	24-Jan-20 1:08 AM	File folder	
hooks	24-Jan-20 1:08 AM	File folder	
language	24-Jan-20 1:08 AM	File folder	
libraries	24-Jan-20 1:08 AM	File folder	
logs	24-Jan-20 1:08 AM	File folder	
models	24-Jan-20 1:08 AM	File folder	
third_party	24-Jan-20 1:08 AM	File folder	
views	24-Jan-20 1:08 AM	File folder	
.htaccess	19-Sep-19 5:08 AM	HTACCESS File	1 KB
index.html	19-Sep-19 5:08 AM	HTML File	1 KB

Gambar 3. 10 Folder Application

a. Application/Cache

Jika menggunakan fitur web page caching dari codeigniter maka file cachanya akan tersimpan di folder ini.

b. Application/Config

Di dalam folder ini tersimpan file untuk konfigurasi antara lain :

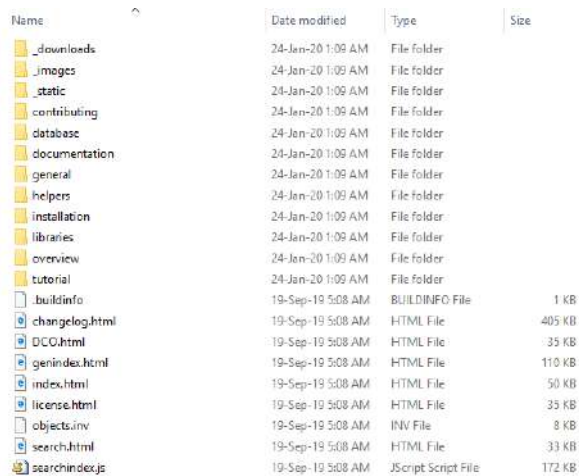
- Autoload.php : File ini digunakan untuk meload config, libraries, helper secara langsung tanpa perlu me-load melalui model atau controller.
- Config.php : File ini digunakan untuk menyimpan konfigurasi dari aplikasi yang dibuat seperti base_url, charset dan lain-lain. Data konfigurasi disimpan dalam arrat \$config.
- Routes.php : File ini digunakan untuk menyimpan setting atau konfigurasi untuk mengatur alur request dari URL, konfigurasi halaman index atau default_controller berada disini.

- Database.php : konfigurasi koneksi ke database seperti hostname, username, password dan dbdriver disimpan dalam file ini.
- c. Application/Controllers
Folder ini digunakan untuk menyimpan file-file controller dari aplikasi yang akan dibuat.
 - d. Application/Models
Folder ini digunakan untuk menyimpan file-file model dari aplikasi yang dibuat.
 - e. Application/Views
Folder ini digunakan untuk tempat menyimpan file-file tampilan atau user interface dari aplikasi yang akan dibuat.
 - f. Application/Libraries
Folder ini digunakan untuk menyimpan file libraries yang dibuat atau libraries tambahan.
2. Folder System : pada folder ini tersimpan file core dari framework codeigniter, sebaiknya folder ini di biarkan jangan diubah-ubah kecuali anda sudah paham apa yang akan anda lakukan.

Name	Date modified	Type
core	24-Jan-20 1:08 AM	File folder
database	24-Jan-20 1:08 AM	File folder
fonts	24-Jan-20 1:08 AM	File folder
helpers	24-Jan-20 1:08 AM	File folder
language	24-Jan-20 1:08 AM	File folder
libraries	24-Jan-20 1:09 AM	File folder
.htaccess	19-Sep-19 5:08 AM	HTACCESS File
index.html	19-Sep-19 5:08 AM	HTML File

Gambar 3. 11 Folder System

3. Folder `User_guide` : berisikan panduan penggunaan framework codeigniter dalam bentuk file html yang dapat anda buka di browser.



Name	Date modified	Type	Size
_downloads	24-Jan-20 1:09 AM	File folder	
_images	24-Jan-20 1:09 AM	File folder	
_static	24-Jan-20 1:09 AM	File folder	
contributing	24-Jan-20 1:09 AM	File folder	
database	24-Jan-20 1:09 AM	File folder	
documentation	24-Jan-20 1:09 AM	File folder	
general	24-Jan-20 1:09 AM	File folder	
helpers	24-Jan-20 1:09 AM	File folder	
installation	24-Jan-20 1:09 AM	File folder	
libraries	24-Jan-20 1:09 AM	File folder	
overview	24-Jan-20 1:09 AM	File folder	
tutorial	24-Jan-20 1:09 AM	File folder	
.buildinfo	19-Sep-19 5:08 AM	BUILDINFO File	1 KB
changelog.html	19-Sep-19 5:08 AM	HTML File	405 KB
DCO.html	19-Sep-19 5:08 AM	HTML File	35 KB
genindex.html	19-Sep-19 5:08 AM	HTML File	110 KB
index.html	19-Sep-19 5:08 AM	HTML File	50 KB
license.html	19-Sep-19 5:08 AM	HTML File	35 KB
objects.inv	19-Sep-19 5:08 AM	INV File	8 KB
search.html	19-Sep-19 5:08 AM	HTML File	33 KB
searchindex.js	19-Sep-19 5:08 AM	JScript Script File	172 KB

Gambar 3. 12 Folder `User_guide`

3.1.3.2 Fitur Codeigniter Framework

A. Footprint Kecil

Seluruh kode sumber untuk kerangka kerja CodeIgniter mendekati 2MB. Ini membuatnya mudah untuk menguasai CodeIgniter dan cara kerjanya. Ini juga menyederhanakan penyebaran dan pembaruannya.

B. Sangat Cepat

Pengguna cenderung menyukai aplikasi yang memuat sangat cepat. Jika Anda telah bekerja dengan beberapa kerangka kerja modern, maka Anda akan menyadari bahwa mereka membutuhkan waktu kurang dari satu detik untuk memuat setelah instalasi. CodeIgniter, Anda dapat memuat rata-rata sekitar kurang dari 50 ms. Waktu ekstra yang dihabiskan untuk mengoptimalkan seperti halnya dalam kerangka kerja lain dibebaskan ketika Anda bekerja dengan CodeIgniter.

C. Loosely Coupled

Fitur bawaan dirancang untuk bekerja secara independen tanpa terlalu bergantung pada komponen lain. Ini membuatnya mudah untuk mempertahankan dan melakukan peningkatan

D. Arsitektur MVC

Kerangka kerja ini menggunakan desain arsitektur Model-View-Controller. Ini adalah praktik standar industri ketika bekerja dengan aplikasi web. MVC memisahkan data, logika bisnis, dan presentasi.

E. Dokumentasi Yang Baik

Kerangka ini didokumentasikan dengan baik, dan ada buku-buku bagus, tutorial dan menjawab pertanyaan forum tentang CodeIgniter. Ini berarti tantangan apa pun yang Anda miliki, kemungkinan seseorang telah mengalami masalah, menyelesaikannya dan solusinya ada di luar sana untuk Anda.

F. Komponen Bawaan Khusus Aplikasi

CodeIgniter memiliki komponen untuk mengirim email, manajemen basis data, manajemen sesi, dan banyak lagi yang akan Anda temukan saat kami melanjutkan tutorialnya.

G. Dapat Diperpanjang

CodeIgniter dilengkapi dengan beberapa perpustakaan, dan pembantu di luar kotak. Jika apa yang Anda inginkan tidak ada di sana atau Anda ingin menerapkan fitur yang ada dengan cara Anda. Kemudian Anda dapat melakukannya dengan mudah dengan membuat perpustakaan, pembantu, paket, dll.

H. Kurva Belajar Yang Pendek CodeIgniter mudah dikuasai bagi siapa saja yang sudah terbiasa dengan PHP. Dalam waktu yang sangat singkat, siswa dapat mulai mengembangkan aplikasi profesional menggunakan CodeIgniter.

3.1.4 Bootstrap



Gambar 3. 13 Logo Bootstrap

Bootstrap adalah toolkit sumber terbuka untuk dikembangkan dengan HTML, CSS, dan JS. Prototype ideide Anda dengan cepat atau bangun seluruh aplikasi Anda dengan variabel dan mixin Sass kami, sistem grid responsif, komponen prebuilt yang luas, dan plugin tangguh yang dibangun di jQuery. 72 Bootstrap terdiri dari templat desain berbasis HTML dan CSS untuk berbagai komponen antarmuka dan ditujukan untuk memudahkan pengembangan web. Dengan memperbarui CSS, Anda dapat beradaptasi dengan tren modern dengan cepat. Pengembang harus lebih berkonsentrasi pada komponen interaksi karena bootstrap itu sendiri akan menangani tampilan data standar, yang dapat diubah nanti jika Anda mau. Bootstrap kompatibel dengan hampir semua browser versi terbaru seperti Internet Explorer, Google Chrome, Opera, Firefox, dan Safari. Ini mendukung desain web responsif dan secara dinamis menyesuaikan tata letak halaman web dengan mempertimbangkan karakteristik perangkat yang digunakan.

3.1.4.1 Kelebihan Dan Kekurangan Bootstrap

A. Kelebihan Bootstrap

1. Mudah Digunakan

Ini adalah prosedur yang sangat mudah dan cepat untuk memulai dengan Bootstrap. Bootstrap juga sangat mudah beradaptasi. Anda dapat menggunakan Bootstrap bersama dengan CSS, atau LESS, atau juga dengan Sass [setelah Anda mengunduh versi Sass].

2. Responsiveness

Setiap tahun perangkat seluler terus tumbuh sangat populer, dan persyaratan untuk memiliki situs web yang responsif telah menjadi keharusan dan juga penting. Karena tata letak jaringan fluida berubah dengan kuat ke resolusi layar yang sesuai, maka membuat situs yang siap-pakai seluler adalah tugas yang mulus dan mudah bersama dengan Bootstrap. Dengan menggunakan kelas-kelas yang sudah jadi dari Bootstrap, Anda dapat mengenali jumlah tempat di sistem kisi yang Anda inginkan untuk digunakan oleh setiap kolom. Maka hanya Anda yang bisa mengidentifikasi pada titik mana saja Anda ingin kolom Anda memuat dalam posisi horizontal. , bukannya vertikal untuk dipamerkan secara akurat di perangkat seluler. Menurut prediksi CISCO, lalu lintas data seluler global akan meningkat sekitar 11 kali lipat antara 2013 dan 2018. Statistik ini menunjukkan perlunya situs web responsif dalam berbagai jenis perangkat seluler. Bootstrap dilengkapi dengan tata letak responsif dan sistem grid 12 kolom yang membantu menyesuaikan situs web secara dinamis dengan resolusi layar yang sesuai. Fitur 'kelas utilitas responsif' dari Bootstrap memungkinkan Anda untuk menyembunyikan / menampilkan bagian konten tertentu untuk ukuran layar tertentu.

3. Kecepatan Pengembangan

Salah satu manfaat utama memanfaatkan Bootstrap adalah kecepatan pengembangan. Saat mengusir situs web atau aplikasi baru yang baru dengan cepat, Anda harus mempertimbangkan menggunakan Bootstrap. Alih-alih mengkodekan dari scrape, Bootstrap memungkinkan Anda untuk menggunakan blok kode yang sudah jadi untuk membantu Anda mengatur. Anda dapat memadukannya bersama dengan fungsionalitas CSS-Kurang dan kompatibilitas lintas-peramban yang dapat memberikan cara untuk menghemat banyak pengkodean. Anda bahkan dapat membeli tema Bootstrap yang sudah jadi dan mengubahnya agar sesuai dengan kebutuhan Anda, untuk mendapatkan rute potensial tercepat.

4. Kustomisasi Bootstrap

Bootstrap dapat dikustomisasi sesuai desain proyek Anda. Pengembang web dapat membuat pilihan untuk memilih aspek-aspek yang diperlukan yang dapat diselesaikan dengan memanfaatkan halaman kustomisasi Bootstrap. Anda hanya perlu menandai semua aspek yang tidak Anda perlukan, seperti- CSS umum: tipografi, kode, sistem kisi, tabel, tombol, formulir, gaya media cetak; Komponen: typography, code, grid system, tables, buttons, forms, print media styles; Komponen JavaScript: dropdown, popover, modals, tooltips, carousels; Utilitas: Utilitas responsif, utilitas dasar. Dengan demikian, versi khusus Bootstrap Anda siap untuk proses unduhan.

5. Konsistensi

Beberapa karyawan Twitter pertama memperluas Bootstrap sebagai kerangka kerja untuk meningkatkan konsistensi di seluruh

alat interior. Namun kemudian Co-founder Mark Otto setelah memahami potensi aktual yang dirilis pada Agustus 2011 versi opensource pertama Bootstrap. Dia bahkan menggambarkan bagaimana Bootstrap diperbesar dengan penggunaan satu inti konsep-pasangan desainer bersama dengan pengembang. Dengan demikian Bootstrap menjadi populer di Twitter.

6. Support

Karena Bootstrap memiliki komunitas dukungan besar, Anda dapat diberi bantuan kapan pun ada masalah. Pembuat selalu memperbarui Bootstrap. Saat ini Bootstrap dihosting, diperluas, dan dipelihara di GitHub bersama dengan lebih dari 9.000 komitmen, serta lebih dari 500 kontributor.

7. Komponen Javascript

Pendekatan Bootstrap dengan paket komponen JavaScript untuk menyertakan fungsionalitas yang membuatnya dengan cara sederhana untuk mengoperasikan hal-hal, seperti tooltips, modal windows, alert, dll. Anda bahkan dapat meninggalkan skrip penulisan sepenuhnya.

8. Integrasi Sederhana

Bootstrap dapat dengan mudah diintegrasikan bersama dengan platform dan kerangka kerja lain yang berbeda, di situs yang ada dan yang baru juga. Anda juga dapat menggunakan elemen-elemen tertentu dari Bootstrap bersama dengan CSS Anda saat ini.

9. Grid

Bootstrap memiliki kemampuan untuk menggunakan grid 12-kolom yang responsif. Ini juga menjunjung tinggi elemen offset dan bersarang. Kotak dapat dipertahankan dalam mode responsif, atau Anda dapat memodifikasinya menjadi Layout aman.

10. Komponen PreStyled

Pendekatan Bootstrap dengan komponen pre-styled untuk alert, dropdown, nav bar, dll. Oleh karena itu, menjadi kaya fitur, Bootstrap memberikan banyak keuntungan dalam menggunakannya. Semoga Anda akan memahami alasan di atas sehingga Anda dapat dengan mudah menggunakan Bootstrap untuk membuat desain web yang luar biasa untuk situs Anda.

B. Kekurangan Bootstrap

1. Dikarenakan menggunakan CSS3, maka bootstrap minim gambar.
2. Dikarenakan terlalu fokus pada coding, sehingga membuat kita kurang begitu kreatif untuk mendesain website sesuai dengna apa yang benar-benar diinginkan.
3. Bootstrap ternyata masih belum mampu untuk bisa menampilkan tampilan website yang sama di seluruh browser. Jadi hanya beberapa browser tertentu saja yang tampikan website dari bootstrap sesuai dengan layar perangkat yang digunakan.

3.1.5 Google Chrome



Gambar 3. 14 Logo Google Chrome

Google Chrome adalah browser web lintas platform yang dikembangkan oleh Google. Ini pertama kali dirilis pada 2008 untuk Microsoft Windows, dan kemudian diangkut ke Linux, macOS, iOS, dan Android. Browser juga merupakan komponen utama Chrome OS, di mana ia berfungsi sebagai platform untuk aplikasi web. Sebagian besar kode sumber Chrome berasal dari proyek Chromium sumber terbuka Google, tetapi Chrome dilisensikan sebagai freeware eksklusif. WebKit adalah mesin render asli, tetapi Google akhirnya memotongnya untuk membuat mesin Blink; semua varian Chrome kecuali iOS sekarang menggunakan Blink.

3.1.5.1 Kelebihan Dan Kelemahan Google Chrome

1. Kelebihan Google Chrome

- a. Chrome adalah browser web yang sangat cepat; itu memuat dan menampilkan halaman dengan sangat cepat.
- b. Google Chrome memiliki desain yang sangat mendasar dan sederhana, sehingga mudah digunakan.
- c. Halaman pembuka memuat daftar halaman yang paling sering Anda kunjungi dan memungkinkan Anda mengklik dan mengaksesnya dengan mudah.
- d. Chrome secara otomatis menerjemahkan halaman ke dalam bahasa yang Anda mengerti untuk kenyamanan Anda.
- e. Jika situs mogok di salah satu tab Anda, tab terbuka lainnya tidak akan terpengaruh.
- f. Anda dapat menjelajahi internet tanpa login menggunakan fitur pribadi baru Chrome: Incognito.
- g. Tidak seperti Firefox dan Safari, Anda dapat mengubah warna dan tema browser.
- h. Chrome lebih cepat dari Explorer dan FireFox.

- i. Chrome hanya membutuhkan beberapa detik untuk menginstal.

2. Kelemahan Google Chrome

- a. Chrome menggunakan banyak memori (RAM) saat menjalankannya.

- b. Jika Anda menutup browser secara tidak sengaja, Chrome akan menutup semua tab yang dibuka tanpa peringatan.

- c. Privasi

Google menyimpan 2% interpretation pencarian pengguna, lengkap dengan alamat IP-a. Walaupun dalam beberapa waktu tertentu interpretation ini akan dianonimkan. Ini artinya google bisa saja tahu “siapa mencari apa dan dimana”.

- d. Celah Keamanan

Beberapa pakar confidence menemukan adanya lubang kecil/bugs pada chrome. Sehingga ketika membuka suatu halaman website akan membuat browser ini menjadi crash. Lalu Chrome juga memiliki underline Download Otomatis yang dikhawatirkan akan disalah gunakan oleh Hacker.

- e. Bahasa

Pada chrome ketika kita memilih untuk menggunakan dalam bahasa Indonesia maka akan terdapat beberapa kejanggalan dalam bahasanya.

- f. Extensions

Pada Chrome tidak terdapat extension/plugin/addons yang dapat ditambahkan. Tidak seperti Firefox yang memiliki banyak aplikasi2 tambahan yang dapat membuat dan meningkatkan kinerja browser.

3.2 Bahasa Pemrograman Yang Digunakan

3.2.1 PHP (*PHP Hypertext Pre-processor*)

PHP (*PHP Hypertext Pre-processor*) adalah Bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML. *PHP* (Hypertext Preprocessor) merupakan sebuah bahasa *scripting* yang terpasang pada *HTML* untuk membuat *websites*, yang dinamis. *PHP* ini bersifat *open source* sehingga *source code*-nya dapat diubah-ubah dan dapat didistribusikan dengan bebas.

Fungsi bahasa pemrograman php sendiri untuk web digunakan untuk dapat menyesuaikan tampilan konten sesuai dengan situasi. Web yang bersifat dinamis juga digunakan untuk menyimpan data ke database dengan memproses form dan juga dapat mengubah tampilan website sesuai inputan dari seorang user. PHP sendiri memiliki framework.

PHP Framework adalah sebagai sebuah kerangka kerja yang disusun oleh berbagai komunitas pengembang web diseluruh dunia. Kerangka ini bertujuan untuk mempermudah dalam membuat sebuah aplikasi web kita sering menulis script PHP secara keseluruhan (konvensional) dan itu pun kita ulang pada halaman yang lain.

- CakePHP
- *Codeigniter*
- Laravel
- PRADO
- Symfony
- Zend Framework
- Yii

- Akelos
- QPHP
- ZooP

BAB IV

INSTALASI TOOLS YANG DIGUNAKAN

4.1 Instalasi Sublime Text

1. Tahap pertama untuk menginstall sublime text adalah download file sublime. Dengan klik download lalu pilih windows 64 bit.

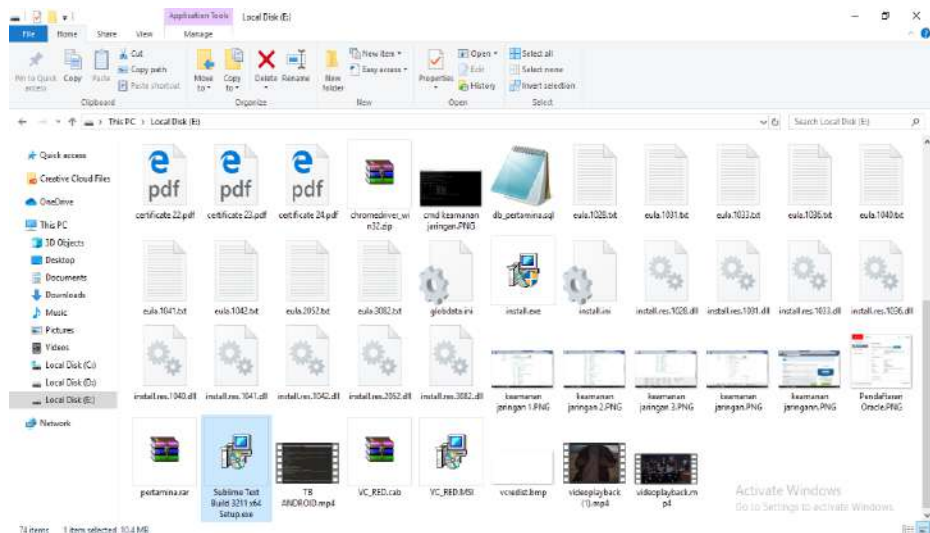


Gambar 4. 1 Download File Sublime Text



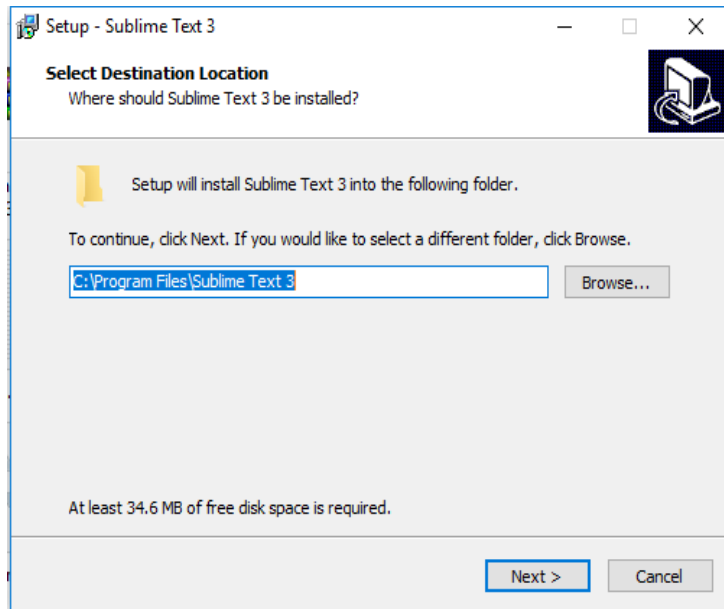
Gambar 4. 2 Download File Sublime Text

2. Tahap kedua setelah mendownload file sublime text kemudian buka direktori dimana file tersebut disimpan. Setelah itu double klik.



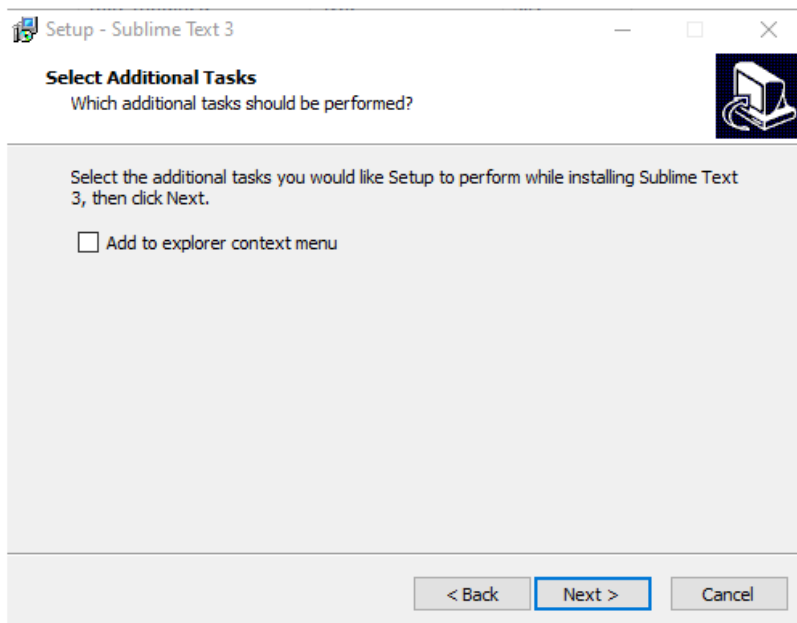
Gambar 4. 3 File Sublime Text

3. Tahap ketiga setelah di double klik, selanjutnya pilih dimana letak instalasi sublime text.



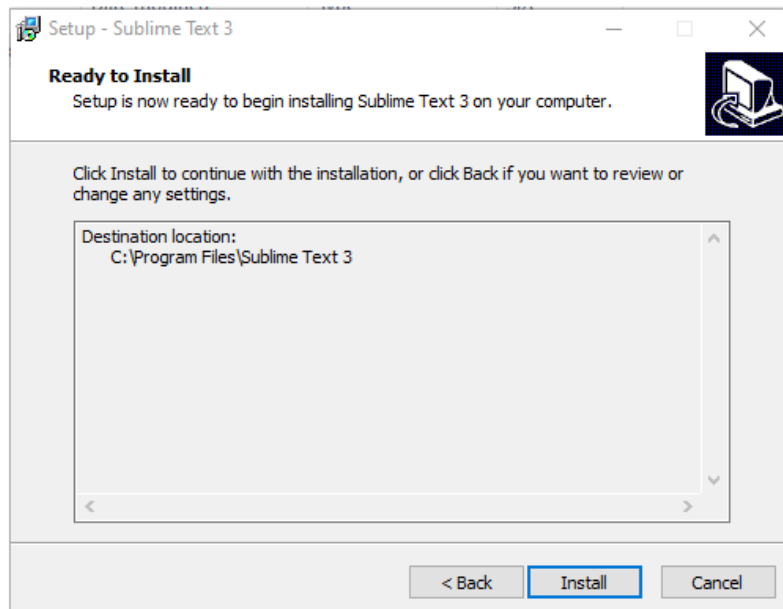
Gambar 4. 4 Proses Instalasi Sublime Text

4. Tahap keempat ceklis kotak jika menyetujui ingin menampilkan sublime text pada menu windows. Seteah itu klik “next”.



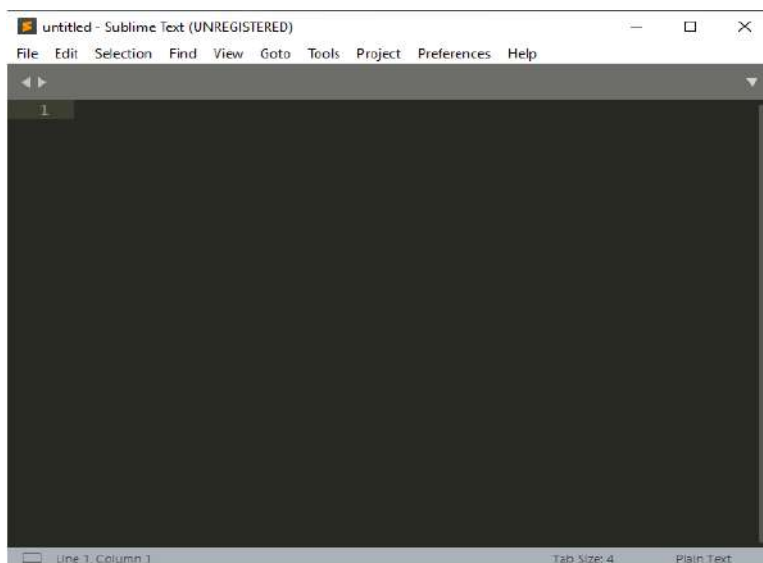
Gambar 4. 5 Proses Installasi Sublime Text

5. Tahap kelima klik “install” dan tunggu hingga instalasi selesai.



Gambar 4. 6 Proses Instalasi Sublime Text

6. Setelah selesai proses instalasi, maka tampilan awal sublime text adalah seperti dibawah ini



Gambar 4. 7 Tampilan Sublime Text

4.2 Instalasi XAMPP

1. Tahap pertama untuk menginstall XAMPP adalah mendownload dahulu file XAMPP. Pilih versi PHPnya. Terdapat 2 versi yaitu PHP 5 dan PHP 7.



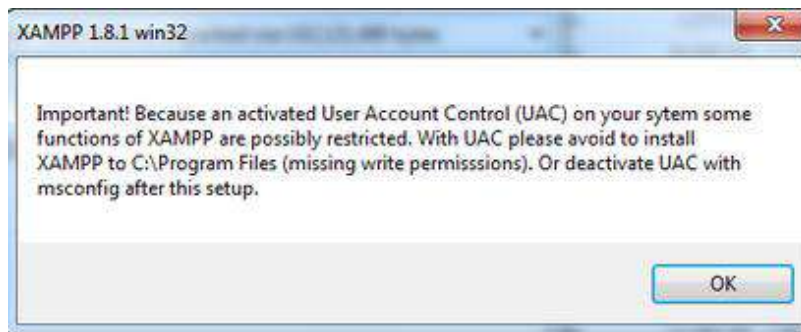
Gambar 4. 8 Download File XAMPP

2. Tahap kedua yaitu klik dua kali pada file XAMPP yang telah didownload. Maka setelah itu akan muncul jendela "installer language". Lalu pilih Bahasa yang ingin digunakan. Biasanya yang dipilih adalah Bahasa Inggris (English).

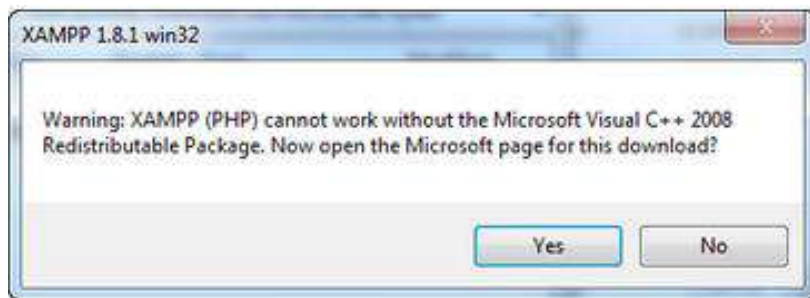


Gambar 4. 9 Tampilan Installer Language

3. Terkadang saat menginstall XAMPP akan muncul pesan error. Tetapi tenang saja, abaikan dan klik “ok” dan juga “yes”.



Gambar 4. 10 Proses Installasi XAMPP



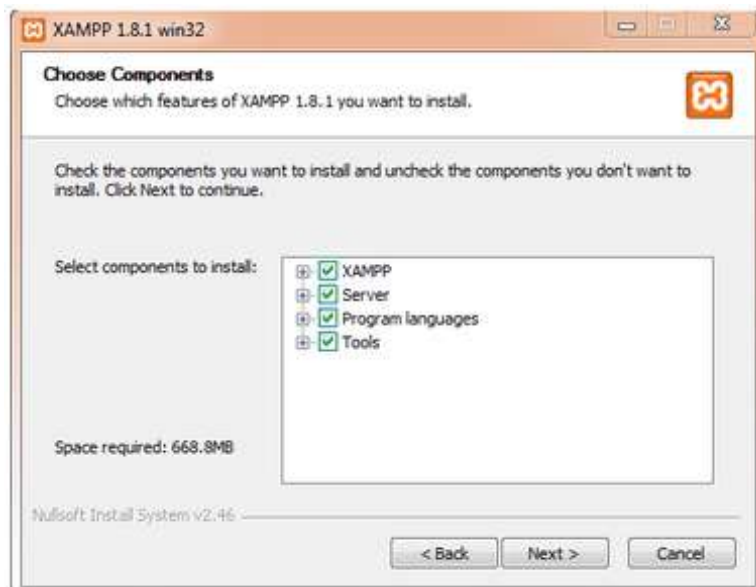
Gambar 4. 11 Proses Installasi XAMPP

4. Selanjutnya, akan muncul jendela tampilan seperti ini, tampilan ini memberitahu untuk menutup semua aplikasi yang sedang berjalan sebelum menginstall XAMPP. Setelah semua aplikasi yang sedang berjalan sudah ditutup lalu klik “next”.



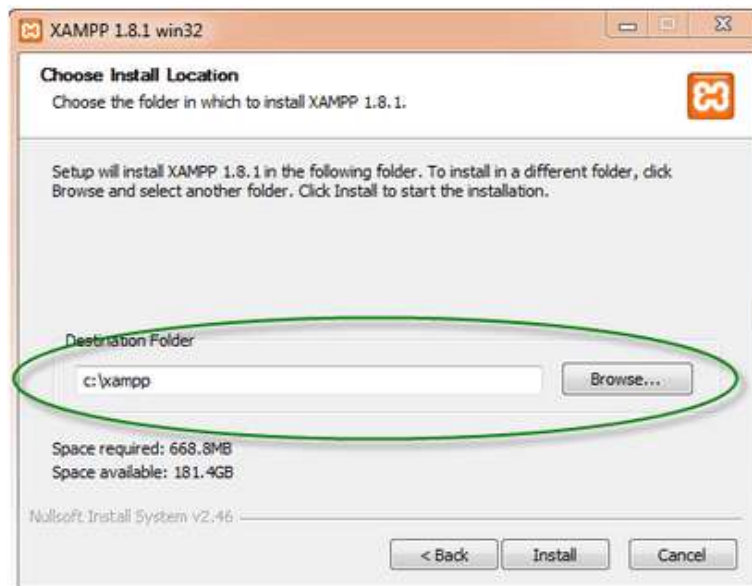
Gambar 4. 12 Proses Installasi XAMPP

5. Anda akan diminta untuk memilih aplikasi apa saja yang akan di install. Centang saja semua yang ada lalu klik tombol “Next”.



Gambar 4. 13 Proses Installasi XAMPP

6. Kemudian anda diminta untuk menentukan lokasi installer. Jika secara default akan diarahkan ke lokasi c:\xampp. Jika anda ingin menyimpan di folder yang lain bisa saja, dengan cara klik “browse” lalu tentukan secara manual folder yang ingin digunakan. Setelah sudah menentukan foldernya lanjutkan dengan klik tombol “Install”.



Gambar 4. 14 Proses Installasi XAMPP

7. Tunggu hingga proses instalasi selesai. Setelah itu akan muncul tampilan jendela seperti dibawah ini, lalu klik “finish”.



Gambar 4. 15 Proses Instalasi XAMPP

8. Langkah selanjutnya, akan muncul jendela dialog seperti gambar dibawah ini yang menanyakan kepada Anda apakah mau langsung menjalankan aplikasi XAMPP atau tidak. Jika ya, maka klik “Yes”.

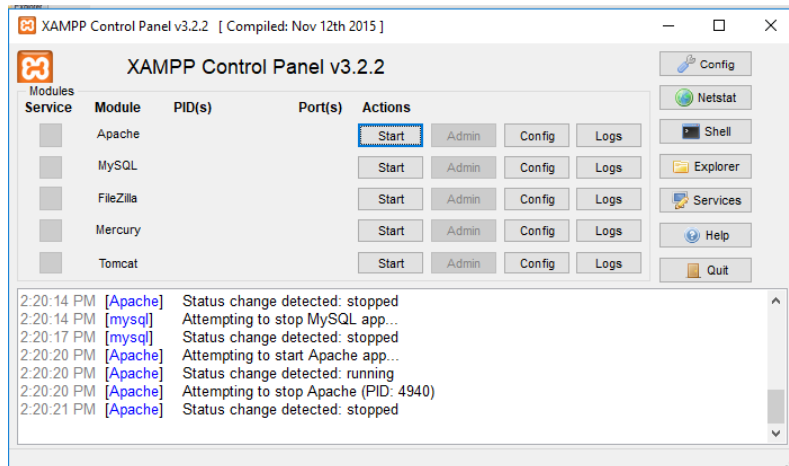


Gambar 4. 16 Proses Instalasi XAMPP

4.2.1 Cara Menjalankan XAMPP

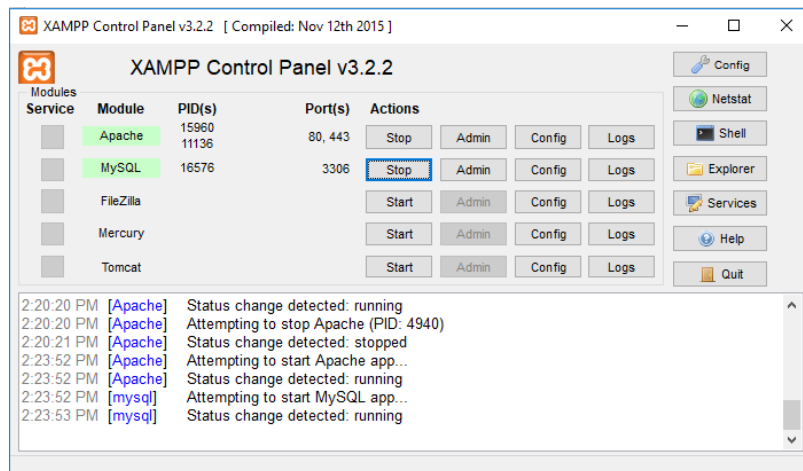
Setelah berhasil melakukan tahap instalasi XAMPP, maka kita akan mencoba untuk menjalankan XAMPP yang sudah berhasil terinstall. Cara menjalankan XAMPP adalah sebagai berikut :

1. Buka aplikasi XAMPP, bisa melalui desktop atau start menu. Setelah itu double klik pada icon XAMPP. Maka akan muncul tampilan seperti ini.



Gambar 4. 17 Proses Menjalankan XAMPP

2. Setelah terbuka klik tombol start hingga tombol tersebut berubah menjadi stop. Jika ingin menggunakan XAMPP ini biasanya yang di jalankan hanya aplikasi *Apache* dan *MySQL*.



Gambar 4. 18 Proses Menjalankan XAMPP

3. Buka browser sesuai dengan keinginan anda lalu coba ketikkan <http://localhost/xampp> pada address bar. Maka akan muncul tampilan seperti dibawah ini. Dan jika sudah muncul tampilan seperti di bawah maka tahapan instalasi sudah berhasil.



Gambar 4. 19 Tampilan XAMPP

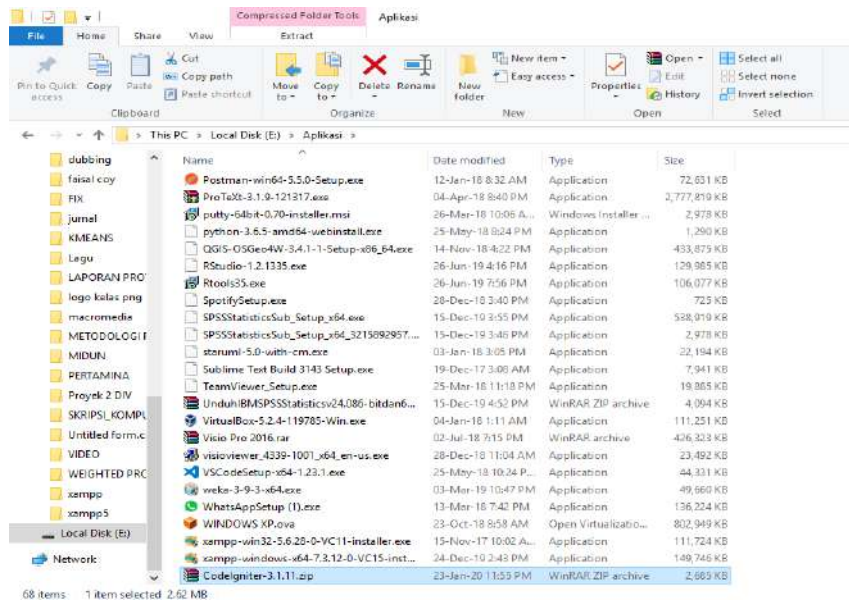
4.3 Instalasi *Codeigniter*

1. Langkah pertama yaitu download package *codeigniter* pada situs resminya <http://codeigniter.com>. Simpan file ditempat yang diinginkan.



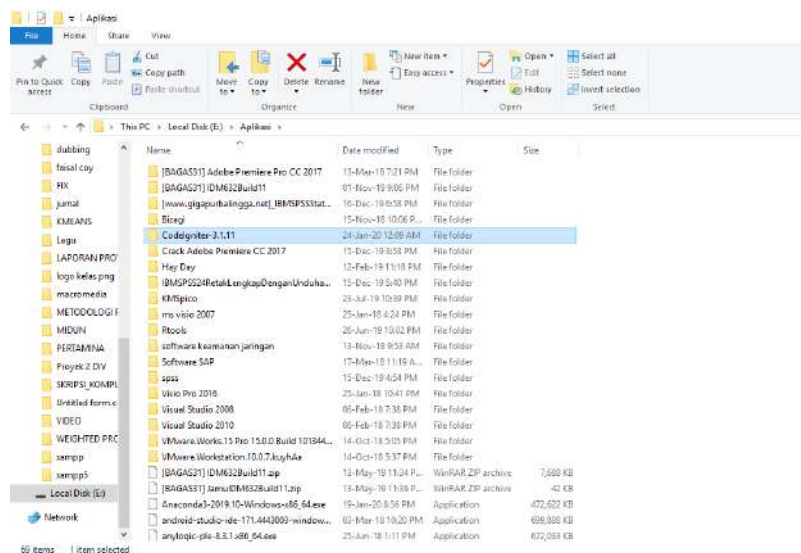
Gambar 4. 20 Proses Instalasi *Codeigniter*

2. Disini saya menyimpan di direktori E:\Aplikasi. Setelah itu ekstrak file *codeigniter* yang telah di download.



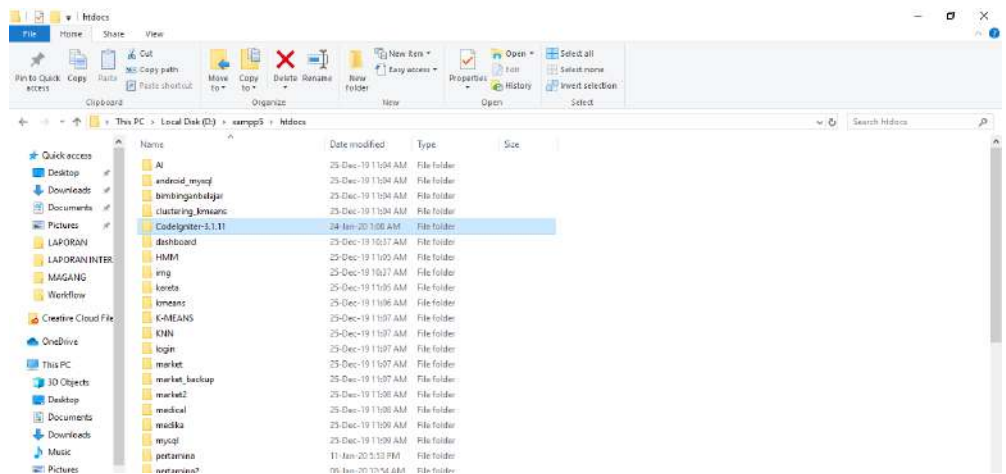
Gambar 4. 21 File Download *Codeigniter*

3. Maka hasil ekstrak dari file tersebut adalah sebagai berikut.



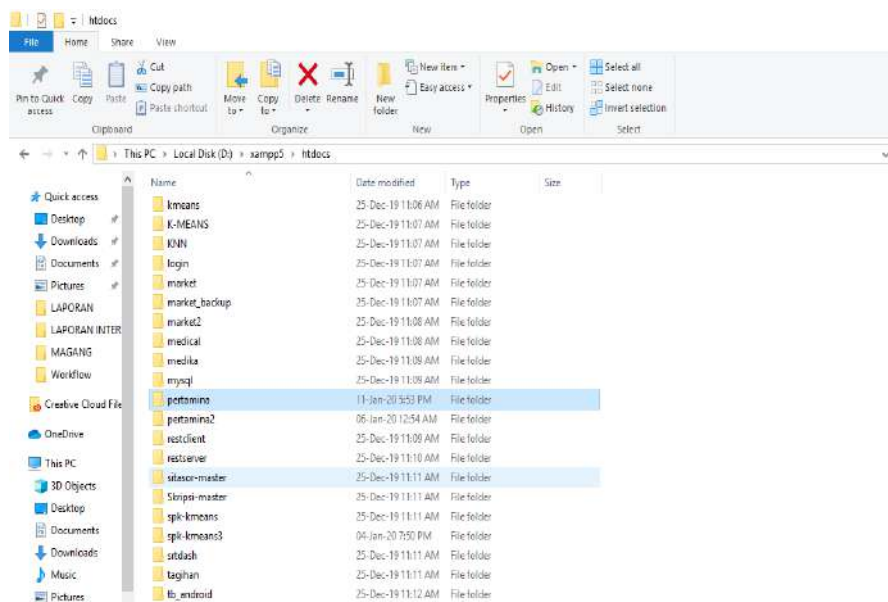
Gambar 4. 22 Hasil Ekstrak File *Codeigniter*

4. Tahap selanjutnya pindahkan file tersebut ke folder xampp yang sebelumnya sudah di install. Pindahkan ke folder D:\xampp5\htdocs.



Gambar 2. 5 File *Codeigniter*

5. Lalu rename nama folder menjadi sesuai kebutuhan. Disini saya mengubah menjadi portamina.



Gambar 4. 23 File *Codeigniter*

BAB V

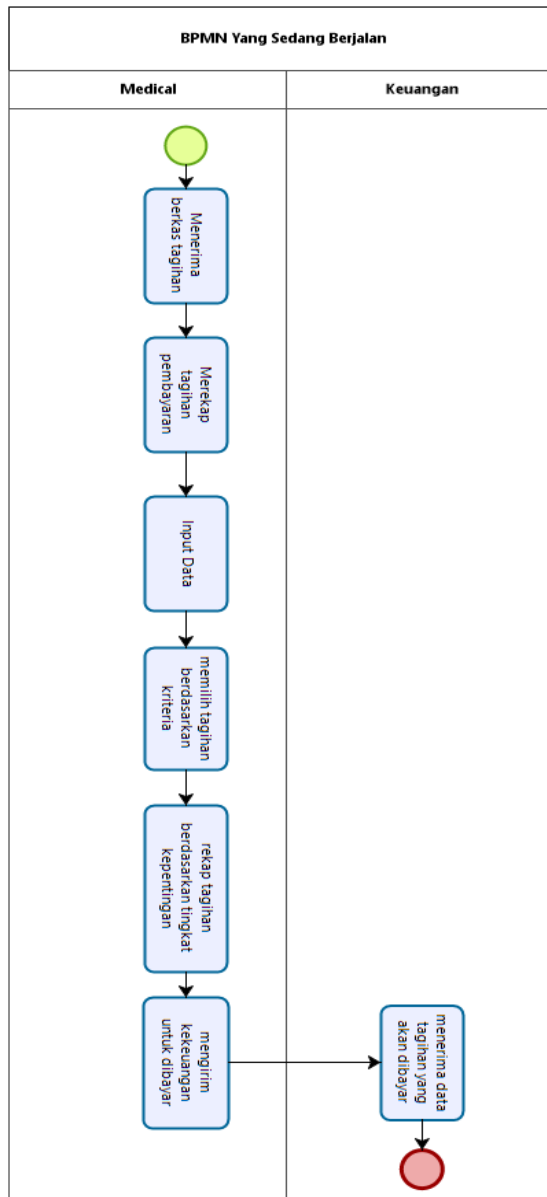
ANALISIS DAN PERANCANGAN PEMBUATAN APLIKASI

5.1 Perancangan Aplikasi

Dalam membangun suatu aplikasi atau sistem informasi sangat diperlukan suatu perancangan, supaya aplikasi atau sistem informasi yang di buat sesuai dengan yang diinginkan, maka dari itu diperlukan suatu perancangan sistem seperti berikut :

5.1.1 Analisis Sistem Yang Sedang Berjalan

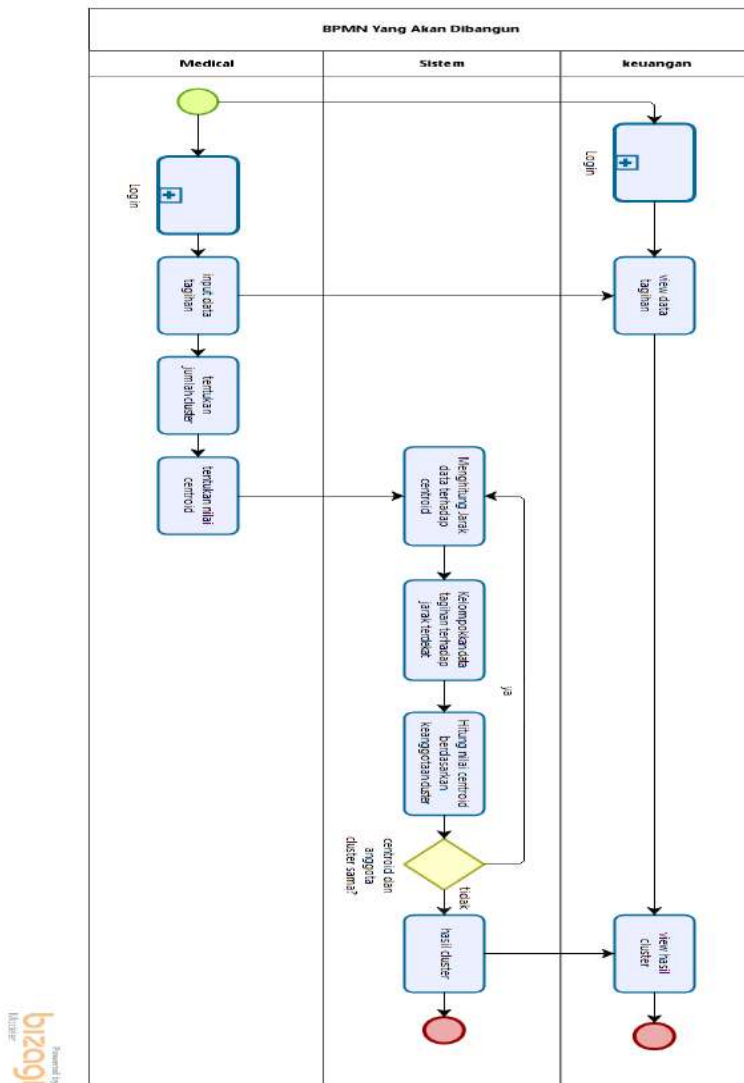
Tahapan yang diperlukan dalam pembuatan suatu program yaitu menganalisa sistem yang telah ada, dimana analisa sistem akan memberikan gambaran tentang sistem yang saat ini sedang berjalan dan bertujuan mengetahui lebih jelas bagaimana cara kerja atau rancangan sistem tersebut serta untuk mendefinisikan dan mengevaluasi permasalahan terjadi dan kebutuhan-kebutuhan yang diharapkan dapat diusulkan suatu perbaikan.



Gambar 5. 1 Analisis Prosedur Yang Sedang Berjalan

5.1.2 Analisis Sistem Yang Akan Dibangun

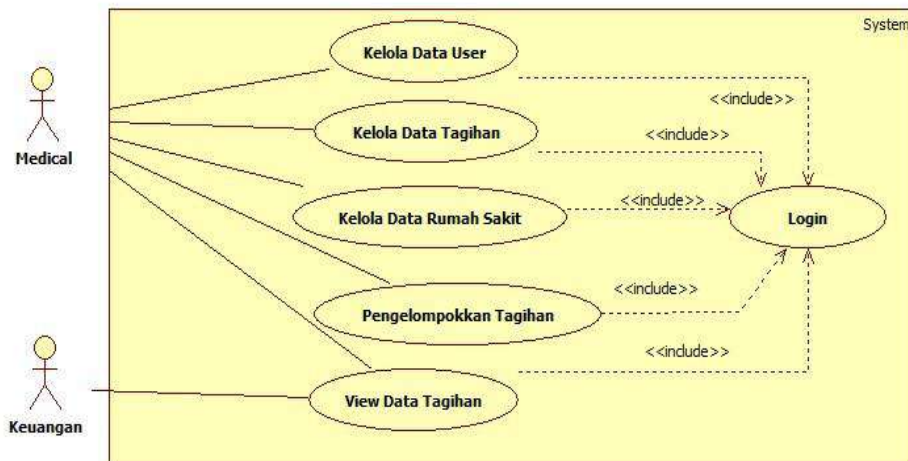
Setelah melakukan analisa dan mengetahui kelemahan kelemahan pada sistem yang sedang berjalan, maka dapat dibuat sebuah sistem baru yang dapat memperbaiki kekurangan kekurangan terhadap sistem yang lama sehingga dapat mebantu untuk memproses informasi dengan lebih cepat.



Gambar 5. 2 Analisis Prosedur Yang Akan Dibangun

5.1.3 Use Case Diagram

Use Case diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antara aktor dengan aktivitas yang terjadi pada sistem dan alat.

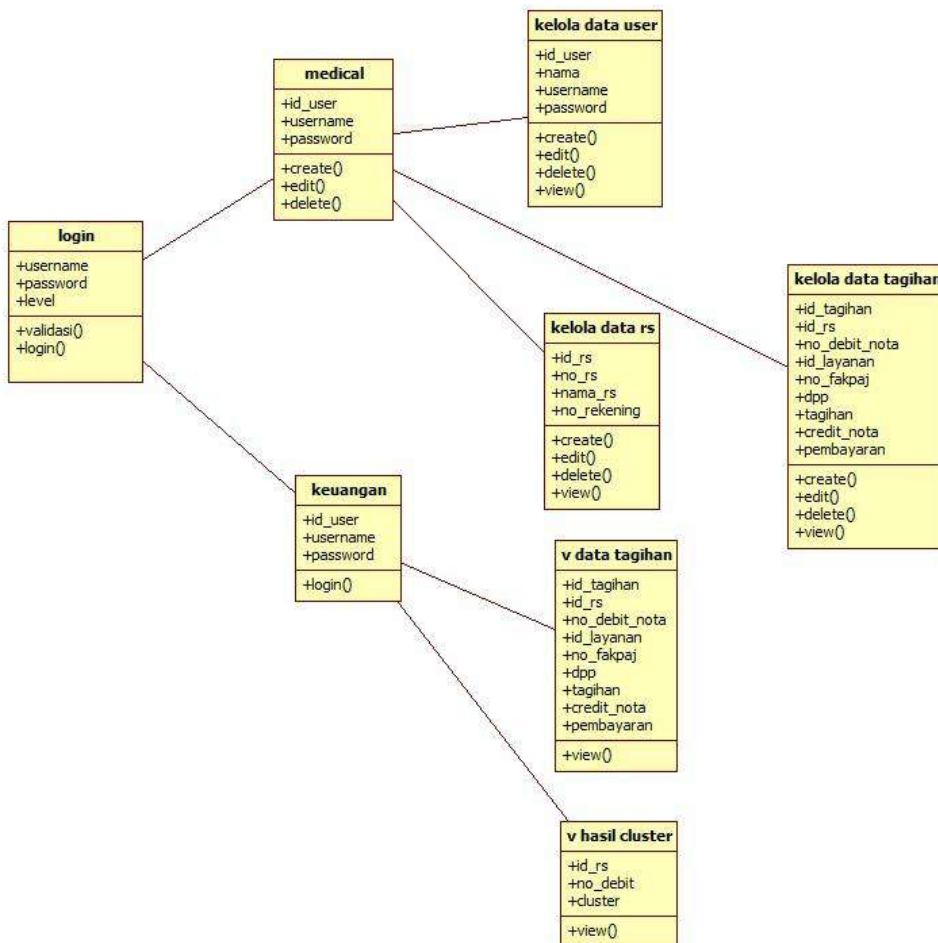


Gambar 5. 3 *Use Case Diagram*

5.1.4 Class Diagram

Class Diagram menggambarkan *Class* dan hubungan antar *Class* didalam sistem. *Class* digambarkan dengan sebuah kotak dibagi menjadi tiga bagian. Bagian paling atas diisikan nama *Class*, bagian tengah diisikan variabel yang dimiliki *Class*, dan bagian bawah diisikan *method-method* dari *Class*.

Dalam proses pembuatan aplikasi ini *Class diagram* yang akan digunakan adalah sebagai berikut :

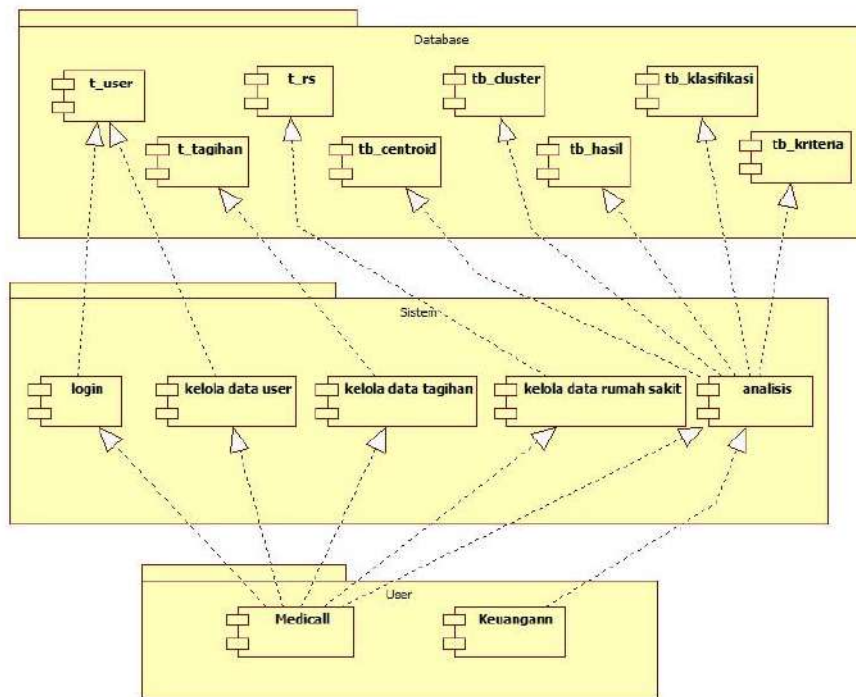


Gambar 5. 4 Class Diagram

5.1.5 Component Diagram

Component diagram yaitu salah satu jenis diagram pada UML yang menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan *software* dari satu ataupun lebih *Class*, dan biasanya berupa *file* data atau *.exe*, *source code*, *table*, dokumen, dll.

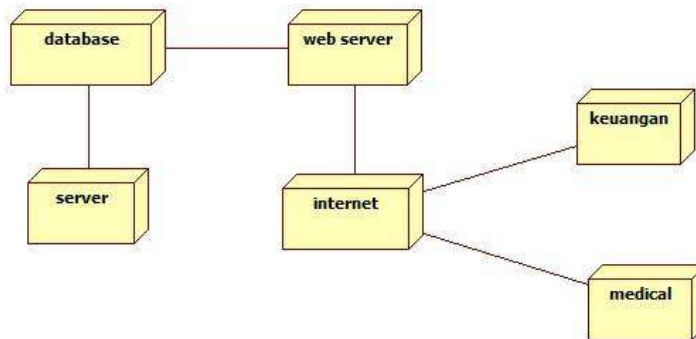
Berikut ini merupakan *component diagram* yang menjelaskan semua komponen yang ada pada sistem.



Gambar 5. 5 *Component Diagram*

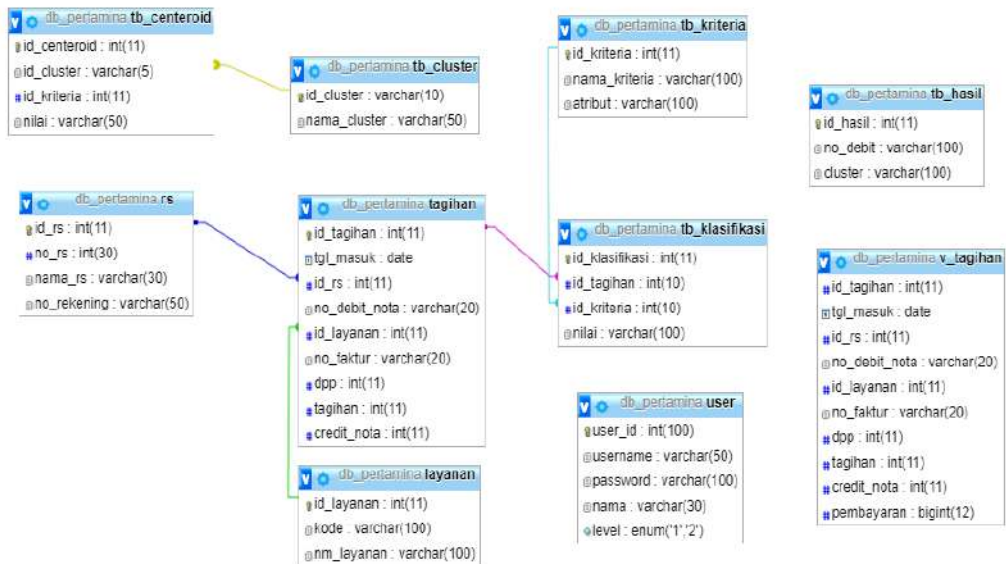
5.1.6 Deployment Diagram

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 *client/server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi.



Gambar 5. 6 *Deployment Diagram*

5.2 Perancangan Database

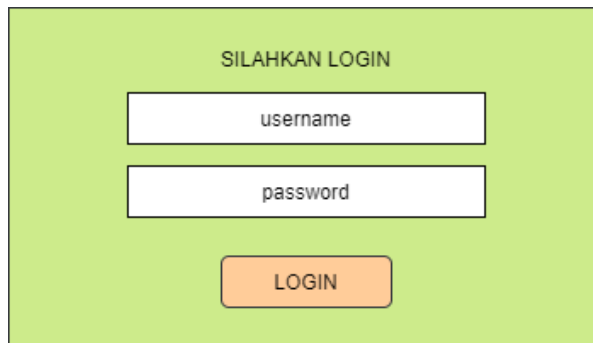


Gambar 5. 7 Perancangan Database

5.3 Perancangan User Interface

Interface adalah rancangan pembangunan dari komunikasi antar pemakai (administrator) dengan komputer. Antar muka (interface) ini terdiri dari proses pemasukan data ke sistem dan menampilkan output informasi kepada administrator.

1. Menu *Login*



A login form with a light green background. At the top, it says "SILAHKAN LOGIN". Below this are two white input fields: the first is labeled "username" and the second is labeled "password". At the bottom is an orange button labeled "LOGIN".

Gambar 5. 8 *User Interface Login*

Keterangan Gambar :

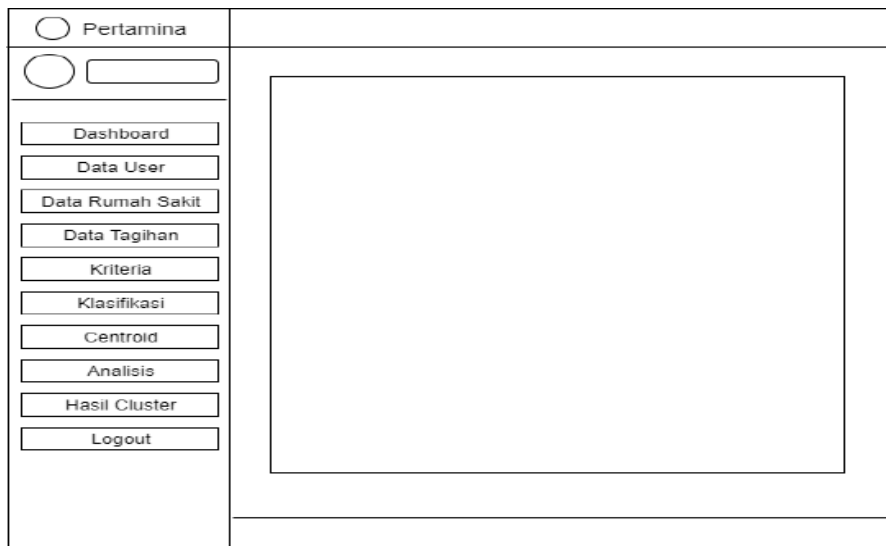
Textbox 1 : *Input Username*

Textbox 2 : *Input Password*

Button 1 : *Tombol Login*

Label 1 : *Silahkan Login*

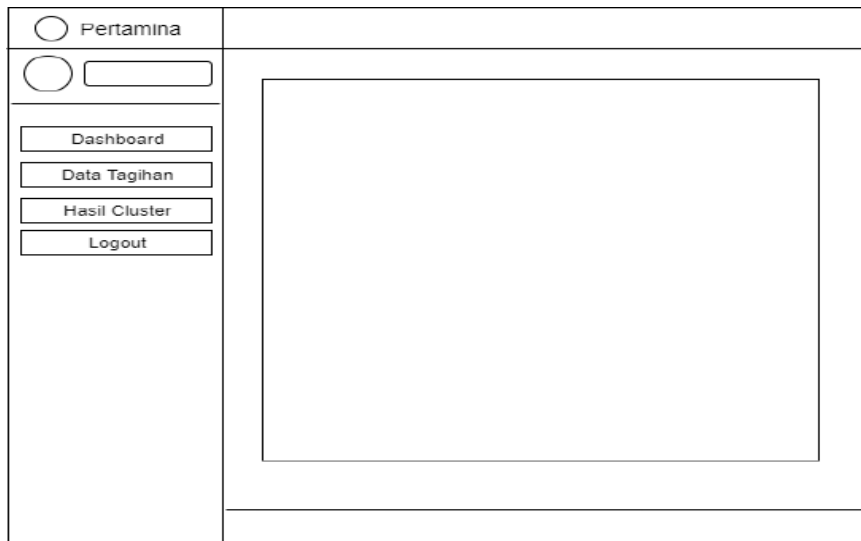
2. *User Interface Halaman Utama Medical*



A dashboard interface for a medical system. On the left is a sidebar with a header containing a radio button and the text "Pertamina". Below the header is a search bar with a magnifying glass icon. The sidebar contains a list of menu items: "Dashboard", "Data User", "Data Rumah Sakit", "Data Tagihan", "Kriteria", "Klasifikasi", "Centroid", "Analisis", "Hasil Cluster", and "Logout". The main area on the right is a large empty rectangle, likely for displaying data or charts.

Gambar 5. 9 *User Interface Halaman Utama Medical*

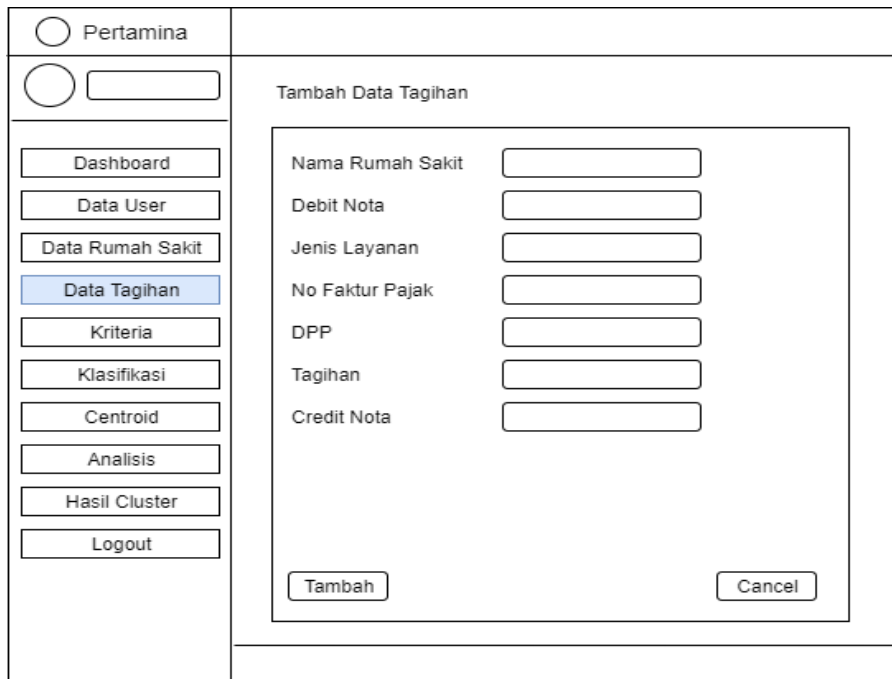
3. *User Interface* Halaman Utama Keuangan



The screenshot shows the main financial page interface. On the left is a sidebar with a logo and a search bar at the top. Below these are five menu items: 'Dashboard', 'Data Tagihan', 'Hasil Cluster', and 'Logout'. The main content area on the right is a large, empty rectangular box, indicating where data or charts would be displayed.

Gambar 5. 10 *User Interface* Halaman Utama Keuangan

4. *User Interface* Tambah Data Tagihan



The screenshot displays the 'Tambah Data Tagihan' (Add Bill Data) form. The sidebar on the left is identical to the previous page, but the 'Data Tagihan' menu item is highlighted in blue. The main content area contains the form title 'Tambah Data Tagihan' and a set of seven input fields with corresponding labels: 'Nama Rumah Sakit', 'Debit Nota', 'Jenis Layanan', 'No Faktur Pajak', 'DPP', 'Tagihan', and 'Credit Nota'. At the bottom of the form are two buttons: 'Tambah' (Add) and 'Cancel'.

Gambar 5. 11 *User Interface* Tambah Data Tagihan

5. *User Interface* Tambah Data User

<div>Pertamina</div> <div></div> <div>Dashboard</div> <div>Data User</div> <div>Data Rumah Sakit</div> <div>Data Tagihan</div> <div>Kriteria</div> <div>Klasifikasi</div> <div>Centroid</div> <div>Analisis</div> <div>Hasil Cluster</div> <div>Logout</div>	<div>Tambah Data User</div> <div><div>Username</div><div>Password</div><div>Nama User</div><div>Level</div></div> <div><div>Tambah</div><div>Cancel</div></div>
---	---

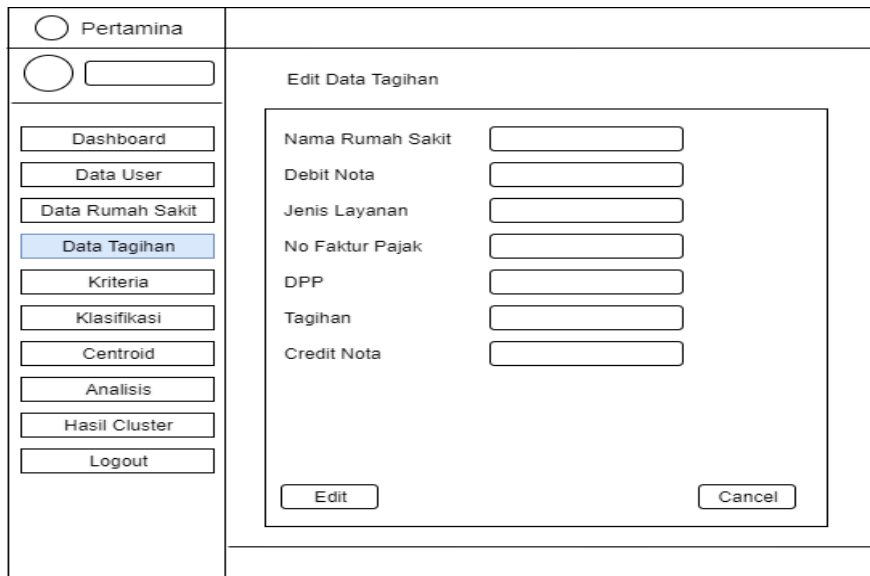
Gambar 5. 12 *User Interface* Tambah Data User

6. *User Interface* Halaman Tambah Data Rumah Sakit

<div>Pertamina</div> <div></div> <div>Dashboard</div> <div>Data User</div> <div>Data Rumah Sakit</div> <div>Data Tagihan</div> <div>Kriteria</div> <div>Klasifikasi</div> <div>Centroid</div> <div>Analisis</div> <div>Hasil Cluster</div> <div>Logout</div>	<div>Tambah Data Rumah Sakit</div> <div><div>Kode Rumah Sakit</div><div>Nama Rumah Sakit</div><div>Nomor Rekening</div></div> <div><div>Tambah</div><div>Cancel</div></div>
---	---

Gambar 5. 13 *User Interface* Halaman Tambah Data Rumah Sakit

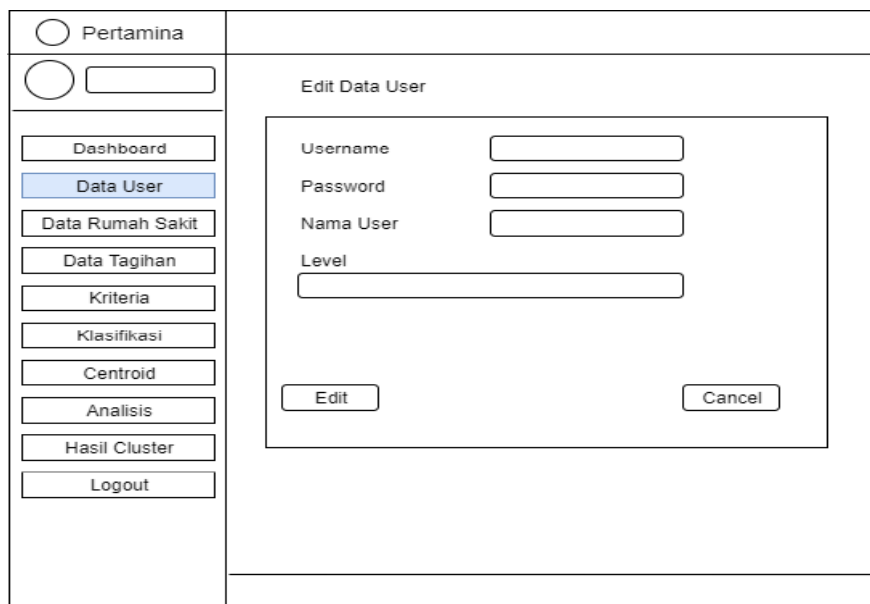
7. *User Interface Edit Data Tagihan*



The image shows a web application interface for editing bill data. On the left is a sidebar menu with a 'Pertamina' logo at the top, followed by a search bar and a list of menu items: Dashboard, Data User, Data Rumah Sakit, Data Tagihan (highlighted in blue), Kriteria, Klasifikasi, Centroid, Analisis, Hasil Cluster, and Logout. The main content area is titled 'Edit Data Tagihan' and contains a form with the following fields: Nama Rumah Sakit, Debit Nota, Jenis Layanan, No Faktur Pajak, DPP, Tagihan, and Credit Nota. Each field has a corresponding text input box. At the bottom of the form are two buttons: 'Edit' and 'Cancel'.

Gambar 5. 14 *User Interface Edit Data Tagihan*

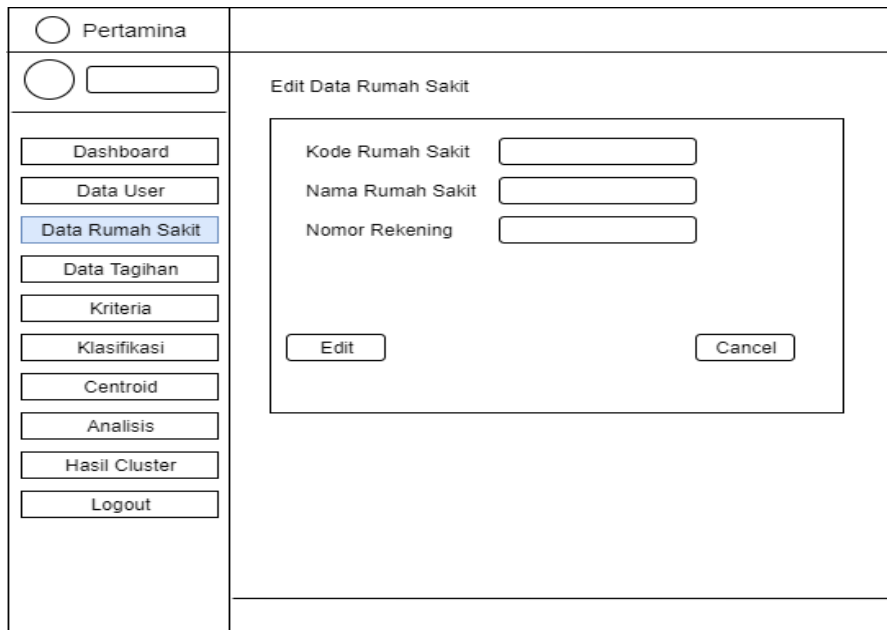
8. *User Interface Edit Data User*



The image shows a web application interface for editing user data. On the left is a sidebar menu with a 'Pertamina' logo at the top, followed by a search bar and a list of menu items: Dashboard, Data User (highlighted in blue), Data Rumah Sakit, Data Tagihan, Kriteria, Klasifikasi, Centroid, Analisis, Hasil Cluster, and Logout. The main content area is titled 'Edit Data User' and contains a form with the following fields: Username, Password, Nama User, and Level. Each field has a corresponding text input box. At the bottom of the form are two buttons: 'Edit' and 'Cancel'.

Gambar 5. 15 *User Interface Edit Data User*

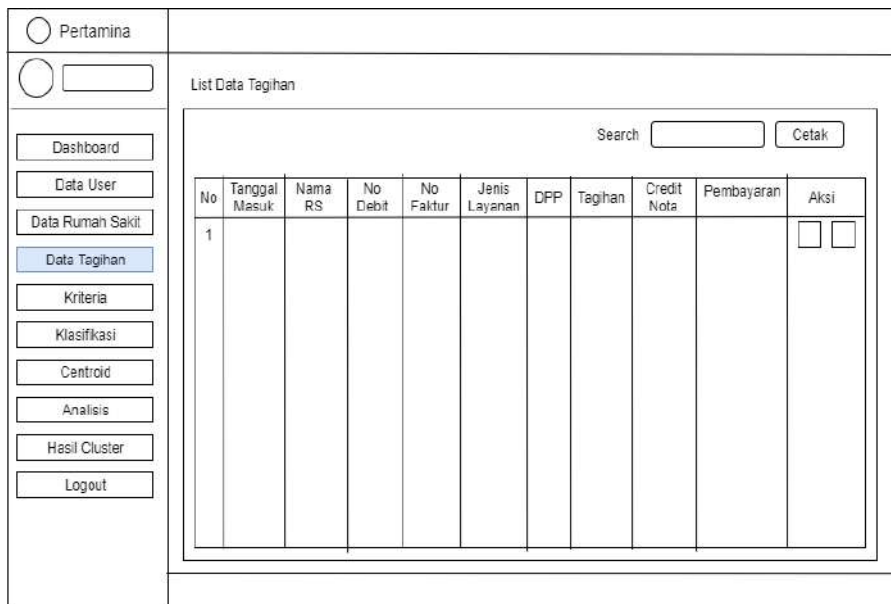
9. User Interface Edit Data Rumah Sakit



The interface for editing hospital data features a sidebar menu on the left with options: Pertamina, a search bar, Dashboard, Data User, Data Rumah Sakit (highlighted), Data Tagihan, Kriteria, Klasifikasi, Centroid, Analisis, Hasil Cluster, and Logout. The main content area is titled 'Edit Data Rumah Sakit' and contains three input fields: Kode Rumah Sakit, Nama Rumah Sakit, and Nomor Rekening. Below these fields are 'Edit' and 'Cancel' buttons.

Gambar 5. 16 User Interface Edit Data Rumah Sakit

10. User Interface Kelola Data Tagihan



The interface for managing billing data features a sidebar menu on the left with options: Pertamina, a search bar, Dashboard, Data User, Data Rumah Sakit, Data Tagihan (highlighted), Kriteria, Klasifikasi, Centroid, Analisis, Hasil Cluster, and Logout. The main content area is titled 'List Data Tagihan' and includes a search bar and a 'Cetak' button. Below these is a table with the following columns: No, Tanggal Masuk, Nama RS, No Debit, No Faktur, Jenis Layanan, DFP, Tagihan, Credit Nota, Pembayaran, and Aksi. The table contains one row with the number '1' in the 'No' column and checkboxes in the 'Aksi' column.

No	Tanggal Masuk	Nama RS	No Debit	No Faktur	Jenis Layanan	DFP	Tagihan	Credit Nota	Pembayaran	Aksi
1										<input type="checkbox"/> <input type="checkbox"/>

Gambar 5. 17 User Interface Kelola Data Tagihan

11. User Interface Kelola Data User

☐ Pertamina
☐

Dashboard
Data User
Data Rumah Sakit
Data Tagihan
Kriteria
Klasifikasi
Centroid
Analisis
Hasil Cluster
Logout

List Data User

Search

No	Username	Password	Nama	Level	Aksi
1					<input type="checkbox"/> <input type="checkbox"/>

Gambar 5. 18 User Interface Kelola Data User

12. User Interface Kelola Data Rumah Sakit

☐ Pertamina
☐

Dashboard
Data User
Data Rumah Sakit
Data Tagihan
Kriteria
Klasifikasi
Centroid
Analisis
Hasil Cluster
Logout

List Data Rumah Sakit

Search

No	No Rumah Sakit	Nama Rumah Sakit	No Rekening	Aksi
1				<input type="checkbox"/> <input type="checkbox"/>

Gambar 5. 19 User Interface Kelola Data Rumah Sakit

13. User Interface Perhitungan

The interface for the K-MEANS calculation. On the left is a sidebar menu with buttons: Dashboard, Data User, Data Rumah Sakit, Data Tagihan, Kriteria, Klasifikasi, Centroid, Analisis (highlighted in blue), Hasil Cluster, and Logout. Above the buttons are two radio buttons, the first labeled 'Pertamina'. The main area is titled 'K-MEANS' and contains a large empty rectangular box for displaying the calculation results.

Gambar 5. 20 User Interface Perhitungan

14. User Interface Hasil Cluster

The interface for displaying the cluster results. The sidebar menu is identical to the previous screen, but 'Hasil Cluster' is now highlighted in blue. The main area is titled 'List Data Tagihan' and contains a search bar with a 'Search' button and three filter buttons: 'Mendesak', 'Tdk Mendesak', and 'Celah'. Below the search bar is a table with the following columns: No, Tanggal Masuk, Tanggal Bayar, Nama RS, No Debit, DPP, Tagihan, Credit Nota, Pembayaran, and Keterangan. The table has one row with the number '1' in the 'No' column.

No	Tanggal Masuk	Tanggal Bayar	Nama RS	No Debit	DPP	Tagihan	Credit Nota	Pembayaran	Keterangan
1									

Gambar 5. 21 User Interface Hasil Cluster

5.4 Perancangan Arsitektur Sistem

5.4.1 Perancangan Arsitektur Perangkat Lunak Sistem

Perangkat Lunak yang digunakan dalam Implementasi Algoritma *K-means* Untuk *Clustering* Tingkat Kepentingan Tagihan Rumah Sakit Pada Sistem Tagihan Pembayaran Studi Kasus : PT. Pertamina (Persero) adalah sebagai berikut:

Tabel 5. 1 Arsitektur Perangkat Lunak Sistem

No.	Tools/Software	Fungsi	Keterangan
1.	<i>Windows 8 (min)</i>	Sistem Operasi	-
2.	<i>Bizagi Modeller</i>	Pembuatan Flow Map	-
3.	<i>Google Chrome</i>	Aplikasi <i>web</i> browser	-
4.	<i>Star UML</i>	Pembuatan <i>Diagram</i> UML	-
5.	<i>MySQL</i>	DBMS	-
6.	<i>Microsoft Word</i>	<i>Document</i>	-
7.	<i>Visual Studio Code</i>	<i>Text Editor</i>	-

5.4.2 Perancangan Arsitektur Perangkat Keras Sistem

Perangkat keras yang digunakan dalam Implementasi Algoritma *K-Means* Untuk *Clustering* Tingkat Kepentingan Tagihan Rumah Sakit Studi Kasus : PT. Pertamina (Persero) ini adalah sebagai berikut :

Tabel 5. 2 Arsitektur Perangkat Keras Sistem

No	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Memory</i>	4 GB	<i>Memory System</i> yang digunakan

2	<i>Processor</i>	<i>Intel® core™ CPU</i>	Untuk kecepatan <i>transfer</i> data dari sistem yang sangat bergantung pada kecepatan <i>processor</i> komputer
3	<i>System Type</i>	64-bit <i>Operating System</i> , x64- <i>Based Processor</i>	

5.5 Perhitungan K-means

Berikut *dataset* yang akan diuji menggunakan algoritma *k-means*.

Tabel 5. 3 Data Tagihan

No	Nama Rumah Sakit	No Debit Nota	DPP	Tagihan	CN	Pembayaran
1	RS Pertamina Pusat	148801/1903028	68650	2044315	201609	1842706
2	RS Pertamina Pusat	148801/1903027	1312230	12227959	201609	12026350
3	RS Pertamina Pusat	148801/1903029	6275270	31858226	682253	31175973
4	RS Pertamina Pusat	148.801/1904020	9500000	34312824	192000	34120824
5	RS Pertamina Pusat	148.801/1907016	4890768	28396469	201609	28194860
6	RS Pertamina Jaya	N02DNJ1904478	2587650	13184268	168512	13015756
7	RS Pertamina Jaya	N02DNJ1904464	4122010	39710365	710143	39000222
8	RS Pertamina Balikpapan	N03DN19051197	4733100	9137531	177350	8960181
9	RS Pertamina Balikpapan	N03DN19060092	8256265	19009751	95625	18914126
10	RS Pertamina Plaju	N10DN19040133	7670860	35804857	1160986	34643871

Setelah menentukan dataset, akan membentuk 2 *cluster* yaitu :

C1 = Mendesak

C2 = Tidak mendesak

1. Langkah awal dalam perhitungan *K-means* adalah menentukan jumlah *cluster* dan juga menentukan *centroid*. Disini peneliti menentukan jumlah *cluster* dan *centroid* berdasarkan data pada table diatas secara *random*. Disini menggunakan jumlah *cluster* 2 dan nilai *centroid* sebagai berikut :

Tabel 5. 4 Centroid Awal

C1	9500000	34312824	192000	34120824
C2	2587650	13184268	168512	13015756

2. Setelah menentukan nilai *centroid* hitung jarak dari setiap data dengan menggunakan rumus :

$$D(i, j) = \sqrt{(X_{1i} - X_{1j})^2 + (X_{2i} - X_{2j})^2 + \dots + (X_{ki} - X_{kj})^2}$$

Keterangan :

$D(i,j)$ = Jarak data ke i ke pusat *cluster* j

X_{li} = Data ke i pada atribut data ke k

X_{lj} = Titik pusat ke j pada atribut ke k

Berikut perhitungan jarak dari setiap data terhadap *cluster* pada Iterasi I:

1. Perhitungan jarak data no 1 terhadap *cluster* 1 dan *cluster* 2

$D(1,1)$

$$= \sqrt{(68650 - 9500000)^2 + (2044315 - 34312824)^2 + (201609 - 192000)^2 + (1842706 - 34120824)^2} = 46605622$$

$D(1,2)$

$$= \sqrt{(68650 - 2587650)^2 + (2044315 - 13184268)^2 + (201609 - 168512)^2 + (1842706 - 13015756)^2} = 15977548$$

2. Perhitungan jarak data no 2 terhadap *cluster* 1 dan *cluster* 2

$$D(2,1)$$

$$= \sqrt{(1312230 - 9500000)^2 + (12227959 - 34312824)^2 + (201609 - 192000)^2 + (12026350 - 34120824)^2} = 32294686$$

$$D(2,2)$$

$$= \sqrt{(1312230 - 2587650)^2 + (12227959 - 13184268)^2 + (201609 - 168512)^2 + (12026350 - 13015756)^2} = 1876497$$

3. Perhitungan jarak data no 3 terhadap *cluster* 1 dan *cluster* 2

$$D(3,1)$$

$$= \sqrt{(6275270 - 9500000)^2 + (31858226 - 34312824)^2 + (682253 - 192000)^2 + (31175973 - 34120824)^2} = 5033531$$

$$D(3,2)$$

$$= \sqrt{(6275270 - 2587650)^2 + (31858226 - 13184268)^2 + (682253 - 168512)^2 + (31175973 - 13015756)^2} = 26312975$$

4. Perhitungan jarak data no 4 terhadap *cluster* 1 dan *cluster* 2

$$D(4,1)$$

$$= \sqrt{(9500000 - 9500000)^2 + (34312824 - 34312824)^2 + (192000 - 192000)^2 + (34120824 - 34120824)^2} = 0$$

$$D(4,2)$$

$$= \sqrt{(9500000 - 2587650)^2 + (34312824 - 13184268)^2 + (192000 - 168512)^2 + (34120824 - 13015756)^2} = 30653237$$

5. Perhitungan jarak data no 5 terhadap *cluster* 1 dan *cluster* 2

$$D(5,1)$$

$$= \sqrt{(4890768 - 9500000)^2 + (28396469 - 34312824)^2 + (201609 - 192000)^2 + (28194860 - 34120824)^2} = 9558526$$

$$D(5,2)$$

$$= \sqrt{(4890768 - 2587650)^2 + (28396469 - 13184268)^2 + (201609 - 168512)^2 + (28194860 - 13015756)^2} = 21612999$$

6. Perhitungan jarak data no 6 terhadap *cluster* 1 dan *cluster* 2

$$D(6,1)$$

$$= \sqrt{(2587650 - 9500000)^2 + (13184268 - 34312824)^2 + (168512 - 192000)^2 + (13015756 - 34120824)^2} = 30653237$$

$$D(6,2)$$

$$= \sqrt{(2587650 - 2587650)^2 + (13184268 - 13184268)^2 + (168512 - 168512)^2 + (13015756 - 13015756)^2} = 0$$

7. Perhitungan jarak data no 7 terhadap *cluster* 1 dan *cluster* 2

$$D(7,1)$$

$$= \sqrt{(4122010 - 9500000)^2 + (39710365 - 34312824)^2 + (710143 - 192000)^2 + (39000222 - 34120824)^2} = 9062738$$

$$D(7,2)$$

$$= \sqrt{(4122010 - 2587650)^2 + (39710365 - 13184268)^2 + (710143 - 168512)^2 + (39000222 - 13015756)^2} = 37168184$$

8. Perhitungan jarak data no 8 terhadap *cluster* 1 dan *cluster* 2

$$D(8,1)$$

$$= \sqrt{(4733100 - 9500000)^2 + (9137531 - 34312824)^2 + (177350 - 192000)^2 + (8960181 - 34120824)^2} = 35910679$$

$$D(8,2)$$

$$= \sqrt{(4733100 - 2587650)^2 + (9137531 - 13184268)^2 + (177350 - 168512)^2 + (8960181 - 13015756)^2} = 6117745$$

9. Perhitungan jarak data no 9 terhadap *cluster* 1 dan *cluster* 2

$$D(9,1)$$

$$= \sqrt{(8256265 - 9500000)^2 + (19009751 - 34312824)^2 + (95625 - 192000)^2 + (18914126 - 34120824)^2} = 21609810$$

$$D(9,2)$$

$$= \sqrt{(8256265 - 2587650)^2 + (19009751 - 13184268)^2 + (95625 - 168512)^2 + (18914126 - 13015756)^2} = 10043183$$

10. Perhitungan jarak data no 10 terhadap *cluster* 1 dan *cluster* 2

$$D(10,1)$$

$$= \sqrt{(7670860 - 9500000)^2 + (35804857 - 34312824)^2 + (1160986 - 192000)^2 + (34643871 - 34120824)^2} = 2604693$$

$$D(10,2)$$

$$= \sqrt{(7670860 - 2587650)^2 + (35804857 - 13184268)^2 + (1160986 - 168512)^2 + (34643871 - 13015756)^2} = 31722081$$

Setelah dilakukan perhitungan jarak data terhadap *cluster* didapatkan hasil sebagai berikut :

Tabel 5. 5 Hasil Jarak Data Terhadap Cluster Iterasi I

NO	C1	C2	CLUSTER
1	46605622	15977548	2
2	32294686	1876497	2
3	5033531	26312975	1
4	0	30653237	1
5	9558526	21612999	1
6	30653237	0	2
7	9062738	37168184	1
8	35910679	6117745	2
9	21609810	10043183	2
10	2604693	31722081	1

Untuk menentukan *cluster* setiap data ditentukan berdasarkan jarak minimum. Jika C1 lebih kecil dari C2 maka masuk ke *cluster* 1 sedangkan jika C2 lebih kecil dari C1 maka masuk ke *cluster* 2.

3. Selanjutnya menghitung untuk titik pusat *cluster* baru berdasarkan rata-rata tiap *cluster* dengan rumus :

$$C = \frac{\sum m}{n}$$

Dimana :

C : *centroid* data

m : anggota data yang termasuk ke dalam jarak *cluster* terdekat

n : jumlah data yang menjadi anggota *cluster* tertentu.

Hasil *centroid* ini akan dipakai untuk perhitungan Iterasi II. Berikut cara menghitung *centroid* baru:

$$C1 \text{ DPP} = (6275270 + 9500000 + 4890768 + 4122010 + 7670860)/5 = 6491782$$

$$C1 \text{ Tagihan} = (31858226 + 34312824 + 28396469 + 39710365 + 35804857)/5 = 34016548$$

$$C1 \text{ CN} = (682253 + 192000 + 201609 + 710143 + 1160986) = 589398$$

$$C1 \text{ Pembayaran} = (31175973 + 34120824 + 28194860 + 39000222 + 34643871)/5 = 33427150$$

Lakukan perhitungan tersebut untuk *cluster* 2 sehingga didapat nilai *cluster* baru sebagai berikut :

Tabel 5. 6 Centroid Baru Untuk Iterasi II

C1	6491782	34016548	589398	33427150
C2	3391579	11120765	168941	10951824

4. Lakukan kembali langkah ke 2. Perhitungan akan berhenti jika antara iterasi terakhir dan iterasi sebelumnya tidak ada lagi perpindahan data. Berikut hasil yang didapat sesuai langkah ke 2 :

Tabel 5. 7 Hasil Jarak Data Terhadap Cluster Pada Iterasi II

NO	C1	C2	CLUSTER
1	45400526	13281600	2
2	30979267	2589449	2
3	3127563	29114218	1
4	3126702	33346477	1
5	7853393	24454466	1
6	29428359	3027237	2
7	8313187	40061257	1
8	34940819	3114419	2
9	20956838	12219039	2
10	2528912	34495129	1

5. Selanjutnya ulangi langkah ke 4 sehingga menghasilkan *centroid* baru. Hasil *centroid* ini akan dipakai untuk perhitungan Iterasi III. Berikut hasil dari perhitungan:

Tabel 5. 8 Centroid Baru Untuk Iterasi III

C1	6491782	34016548	589398	33427150
C2	3391579	11120765	168941	10951824

6. Lakukan kembali tahap ke 4. Setelah dilakukan perhitungan pada tahap 4 hasil dari Iterasi II dan Iterasi III pada *centroid* dan hasil *cluster* tidak berubah maka proses *clustering* selesai. Berikut hasil *clustering* akhir :

Tabel 5. 9 Hasil Clustering Akhir

NO	NAMA RUMAH SAKIT	NO DEBIT NOTA	DPP	TAGIHAN	CN	PEMBAYARAN	CLUSTER
1	RS Pertamina Pusat	148801/1903028	68650	2044315	201609	1842706	2
2	RS Pertamina Pusat	148801/1903027	1312230	12227959	201609	12026350	2
3	RS Pertamina Pusat	148801/1903029	6275270	31858226	682253	31175973	1
4	RS Pertamina Pusat	148.801/1904020	9500000	34312824	192000	34120824	1
5	RS Pertamina Pusat	148.801/1907016	4890768	28396469	201609	28194860	1
6	RS Pertamina Jaya	N02DNJ1904478	2587650	13184268	168512	13015756	2
7	RS Pertamina Jaya	N02DNJ1904464	4122010	39710365	710143	39000222	1
8	RS Pertamina Balikpapan	N03DN19051197	4733100	9137531	177350	8960181	2
9	RS Pertamina Balikpapan	N03DN19060092	8256265	19009751	95625	18914126	2
10	RS Pertamina Plaju	N10DN19040133	7670860	35804857	1160986	34643871	1

BAB VI

TUTORIAL PEMBUATAN APLIKASI

Sebelum memaparkan tutorial dalam pembuatan sistem *clustering* tagihan ini penulis akan menjelaskan sedikit pemaparan latar belakang mengapa “Sistem *Clustering* Tagihan” ini dibuat dan hasil yang diperoleh dari aplikasi sudah berhasil dibuat dan diuji. Adapun permasalahan yang terjadi pada department *Health and Medical Management* ini, dimana dalam menentukan tagihan mana yang harus dibayar terlebih dahulu masih manual sehingga masih kurang tepat dalam menentukan mana tagihan pembayaran yang seharusnya didahulukan namun telat dalam hal pembayaran ke rumah sakit terkait. Untuk mengatasi permasalahan tersebut maka perlu dibuatkan sistem yang memudahkan untuk pengelompokkan tagihan berdasarkan kepentingannya serta dalam mengatasi permasalahan tersebut menggunakan teknik *data mining* yaitu metode *K-means clustering*.

6.1 Mempersiapkan Tools

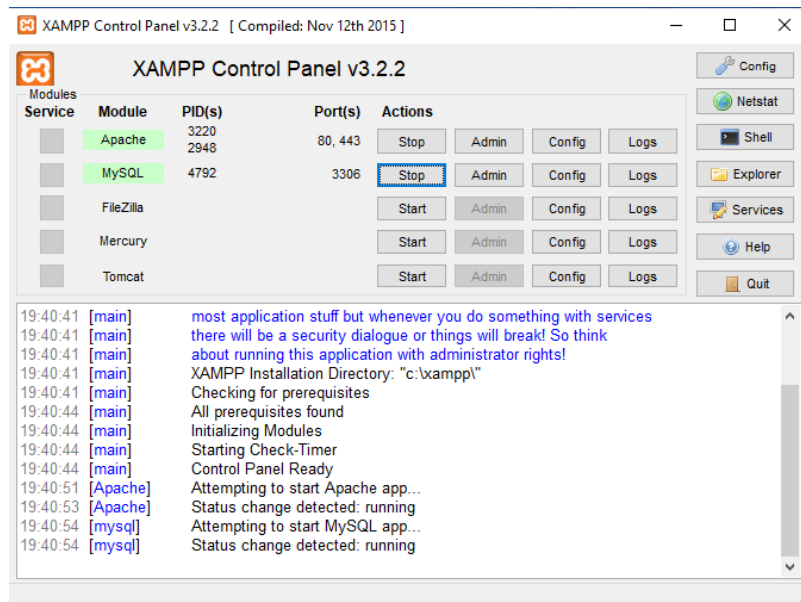
Untuk membuat sistem *clustering* tagihan maka diperlukan beberapa *tools* yang dapat mendukung proses pembangunan sistem ini, di mana pada bab sebelumnya telah dijelaskan *tools* apa saja yang digunakan. Berikut beberapa *tools* yang digunakan:

1. *Sublime Text*

2. *XAMPP*
3. *Framework Codeigniter*

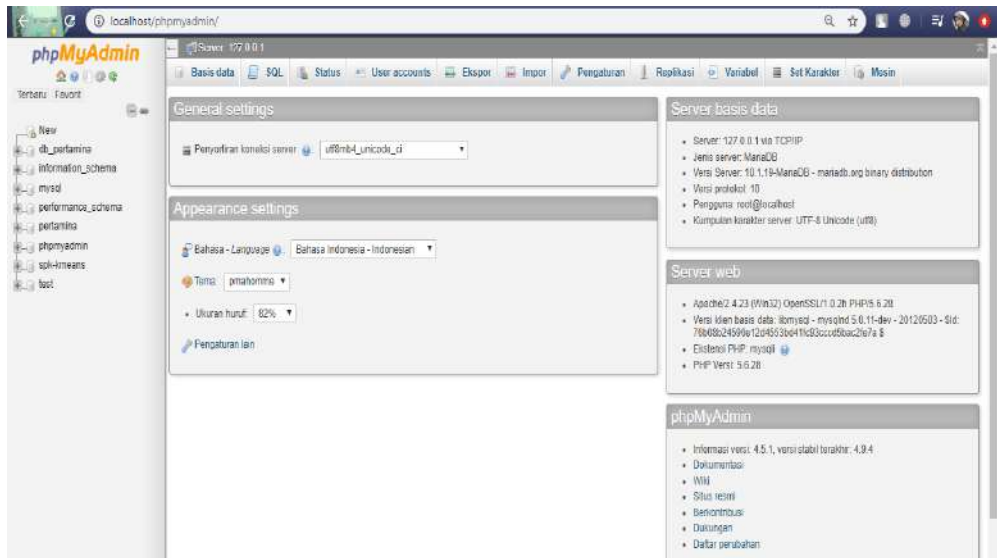
6.2 Membuat Database

1. Langkah awal yang harus dilakukan adalah membuka *XAMPP* di mana *XAMPP* akan memberikan koneksi kepada kita agar bisa mengakses *MySQL* sebagai penyimpanan *database*. Jalankan *XAMPP* dengan klik *start* pada *APACHE* dan *MySQL*.



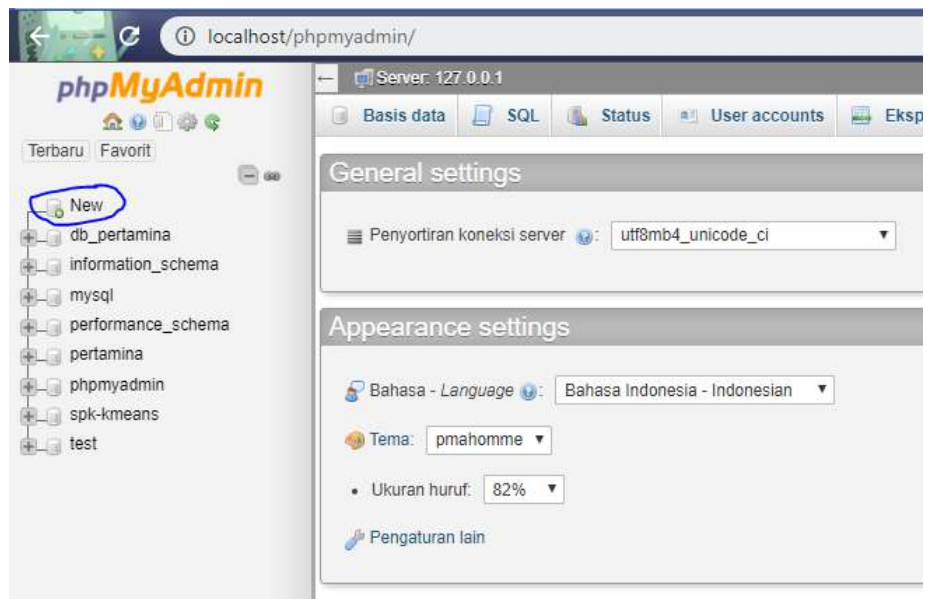
Gambar 6. 1 Menjalankan XAMPP

2. Setelah menjalankan aplikasi *XAMPP*, buka browser yang ada di perangkat anda, kemudian ketikkan <http://localhost/phpmyadmin/> pada *address bar*.



Gambar 6. 2 Tampilan *Localhost*

3. Setelah itu klik *new* untuk membuat *database* baru.



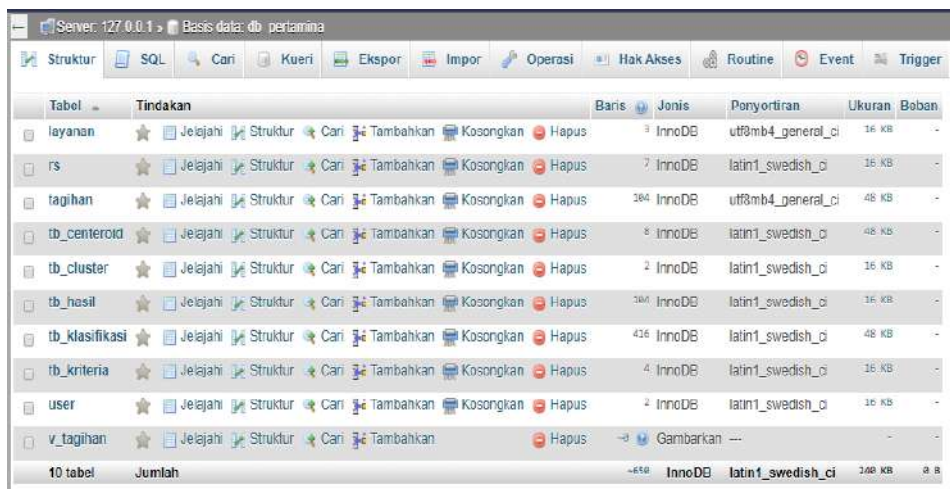
Gambar 6. 3 Membuat *Database* Baru

4. Pada pembuatan sistem ini saya membuat database yang saya butuhkan dengan nama db_pertamina. selanjutnya klik buat untuk membuat *database* baru.



Gambar 6. 4 Membuat *Database* Baru

5. Setelah membuat nama *database* yang diinginkan selanjutnya buat table sesuai dengan kebutuhan.



Gambar 6. 5 Membuat Table Database

Pada table db_pertamina ini terdapat 9 tabel antara lain :

1. Layanan



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_layanan	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	kode	varchar(100)			Tidak	Tidak ada	
3	nm_layanan	varchar(100)			Tidak	Tidak ada	

Gambar 6. 6 Struktur Tabel Layanan

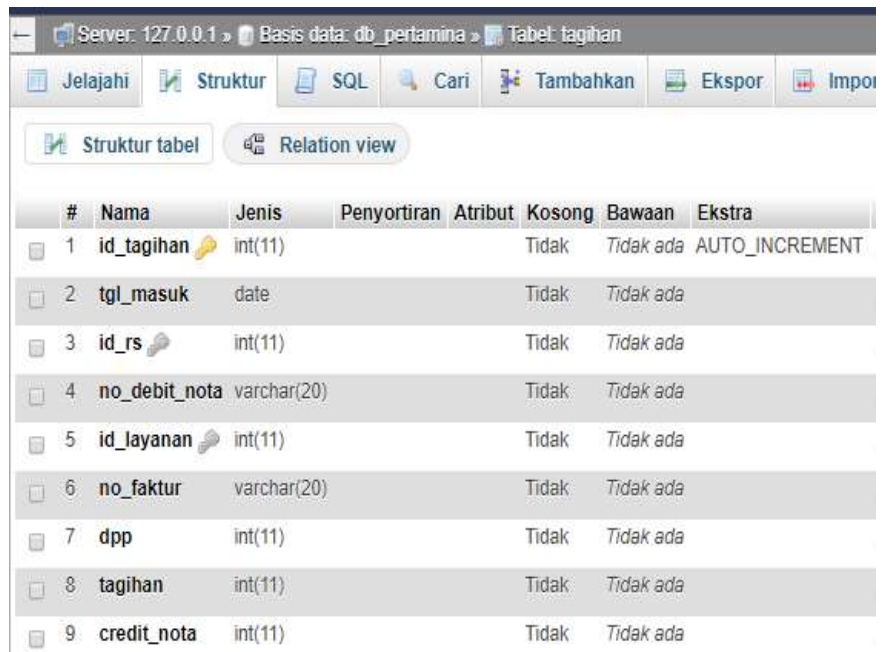
2. RS (Rumah Sakit)



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_rs	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	no_rs	int(30)			Tidak	Tidak ada	
3	nama_rs	varchar(30)			Tidak	Tidak ada	
4	no_rekening	varchar(50)			Tidak	Tidak ada	

Gambar 6. 7 Tabel RS

3. Tagihan



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_tagihan	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	tgl_masuk	date			Tidak	Tidak ada	
3	id_rs	int(11)			Tidak	Tidak ada	
4	no_debit_nota	varchar(20)			Tidak	Tidak ada	
5	id_layanan	int(11)			Tidak	Tidak ada	
6	no_faktur	varchar(20)			Tidak	Tidak ada	
7	dpp	int(11)			Tidak	Tidak ada	
8	tagihan	int(11)			Tidak	Tidak ada	
9	credit_nota	int(11)			Tidak	Tidak ada	

Gambar 6. 8 Struktur Tabel Tagihan

4. Tb_centroid



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_centeroid	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	id_cluster	varchar(5)			Tidak	Tidak ada	
3	id_kriteria	int(11)			Tidak	Tidak ada	
4	nilai	varchar(50)			Tidak	Tidak ada	

Gambar 6. 9 Struktur Tb_centroid

5. Tb_Cluster



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan
1	id_cluster	varchar(10)			Tidak	Tidak ada
2	nama_cluster	varchar(50)			Tidak	Tidak ada

Gambar 6. 10 Struktur Tb_cluster

6. Tb_klasifikasi



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_klasifikasi	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	id_tagihan	int(10)			Tidak	Tidak ada	
3	id_kriteria	int(10)			Tidak	Tidak ada	
4	nilai	varchar(100)			Tidak	Tidak ada	

Gambar 6. 11 Struktur Tb_klasifikasi

7. Tb_kriteria



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_kriteria	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	nama_kriteria	varchar(100)			Tidak	Tidak ada	
3	atribut	varchar(100)			Tidak	Tidak ada	

Gambar 6. 12 Struktur Tb_kriteria

8. User



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	user_id	int(100)			Tidak	Tidak ada	AUTO_INCREMENT
2	username	varchar(50)			Tidak	Tidak ada	
3	password	varchar(100)			Tidak	Tidak ada	
4	nama	varchar(30)			Tidak	Tidak ada	
5	level	enum('1', '2')			Tidak	Tidak ada	

Gambar 6. 13 Struktur Tabel *User*

9. Tb_hasil



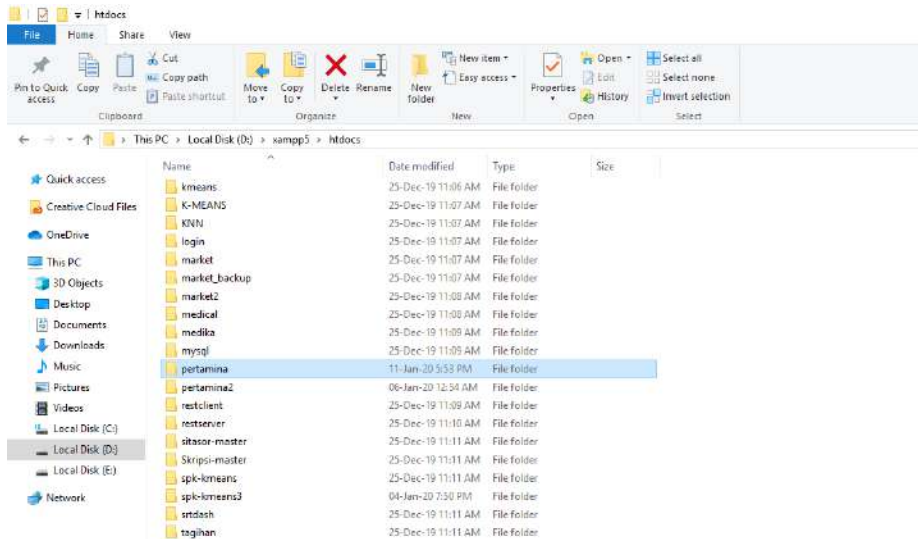
#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_hasil	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	no_debit	varchar(100)			Tidak	Tidak ada	
3	cluster	varchar(100)			Tidak	Tidak ada	

Gambar 6. 14 Struktur Tb_hasil

6.3 Konfigurasi Database

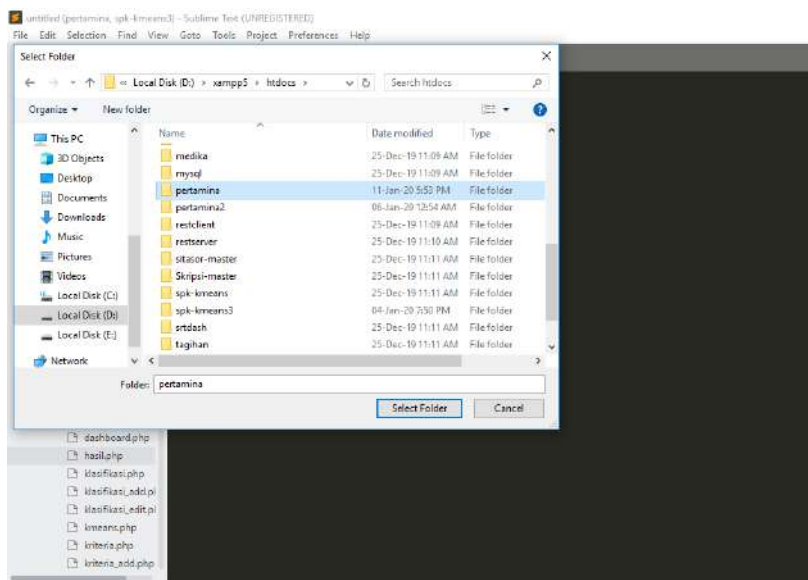
Pada setiap pembuatan aplikasi, sistem informasi ataupun program harus melakukan konfigurasi terlebih dahulu agar program yang dibuat dapat terhubung dengan *database* yang telah dibuat sebelumnya. Berikut merupakan langkah-langkah dari konfigurasi *CodeIgniter*:

1. Pindahkan folder *CodeIgniter* yang telah di *download* kedalam folder D:\xampp5\htdocs, kemudian *rename* sesuai dengan kebutuhan anda masing-masing. Pada kasus ini saya memberi nama sistem informasi saya dengan *pertamina*.



Gambar 6. 15 Rename Nama Folder

2. Buka *tools sublime text*, kemudian buka folder *codeigniter* yang telah di *rename* menjadi *pertamina*.



Gambar 6. 16 Open Folder

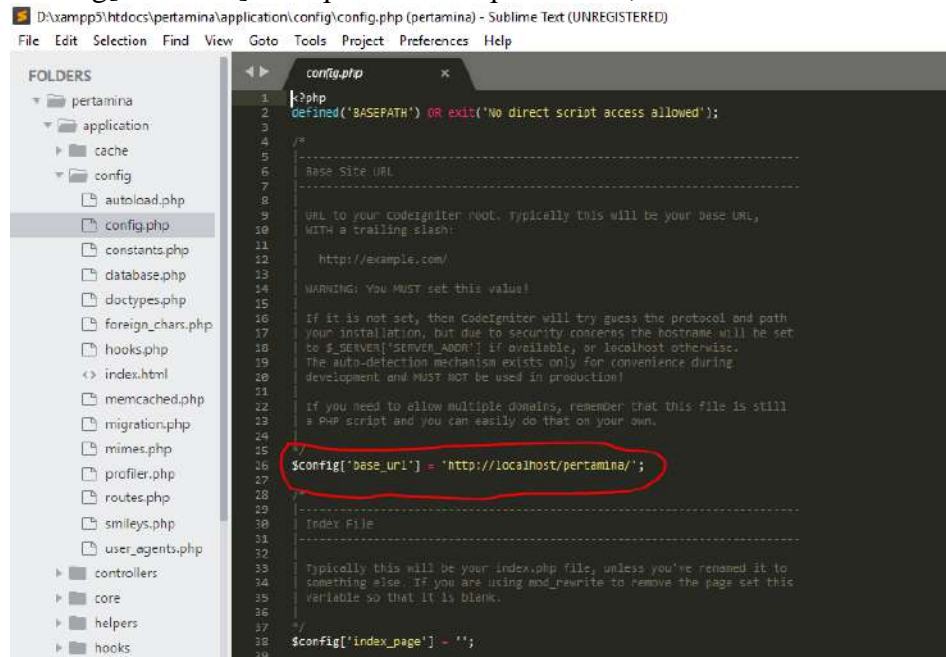
3. Kemudian untuk konfigurasi buka folder *config>config.php*.
Kemudian pada bagian:

`$config['base_url'] =`

isikan menjadi sesuai nama folder codeigniter

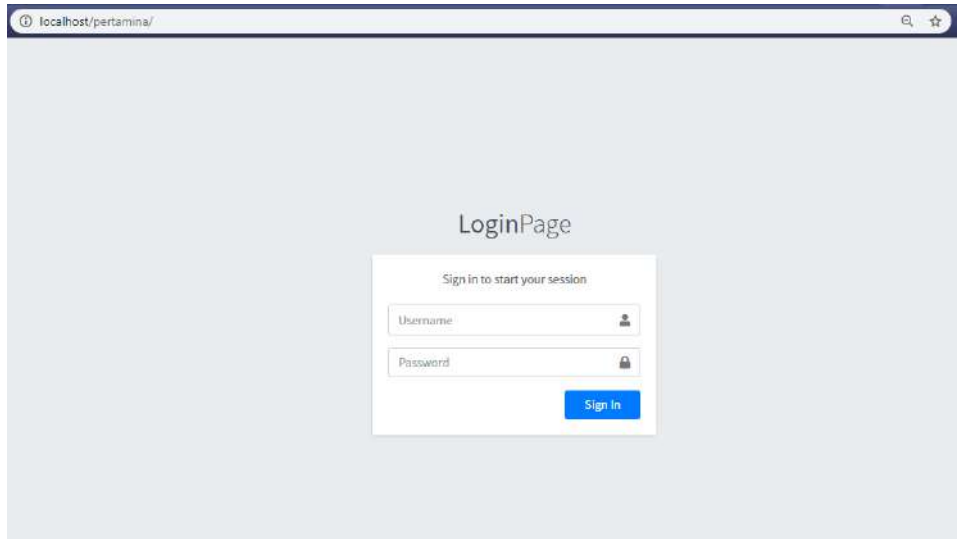
menjadi:

`$config['base_url'] = 'http://localhost/pertamina';`



Gambar 6. 17 Konfigurasi Config

4. Dan hasilnya jika kita memanggil localhost/pertamina/ pada aplikasi web browser akan muncul tampilan program tersebut.



Gambar 6. 18 Tampilan *Login*

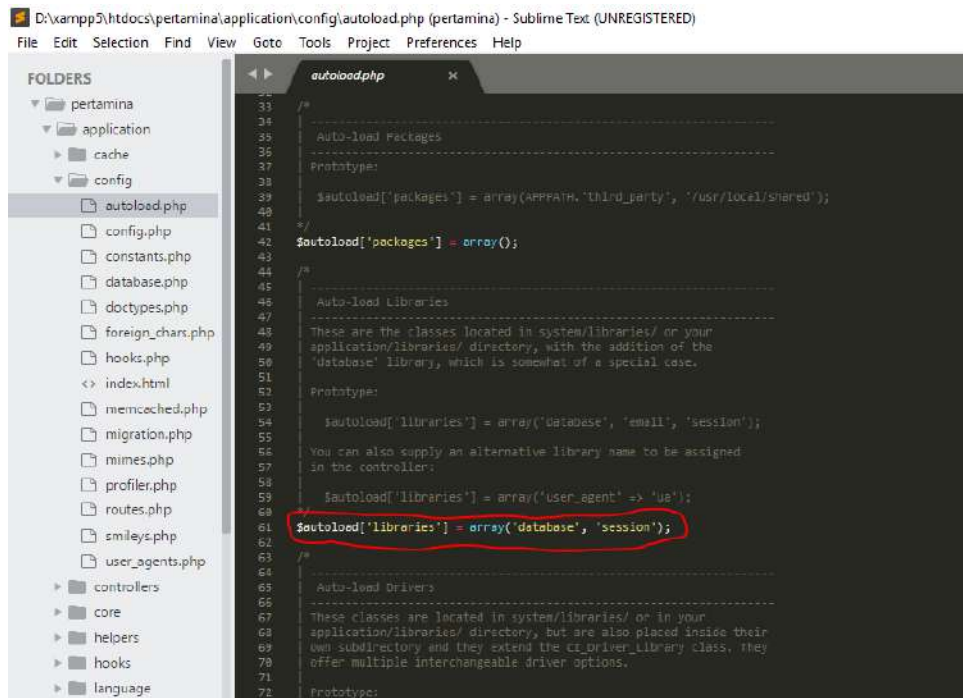
5. Kemudian langkah selanjutnya adalah buka folder *config>autoload.php* pada bagian:

```
$autoload['libraries'] =
```

isi menjadi:

```
$autoload['libraries'] = array('database', 'session');
```

Konfigurasi pada file ini bertujuan untuk menentukan sumber daya apa yang akan di *load* secara otomatis nantinya.



Gambar 6. 19 Konfigurasi Autoload (Libraries)

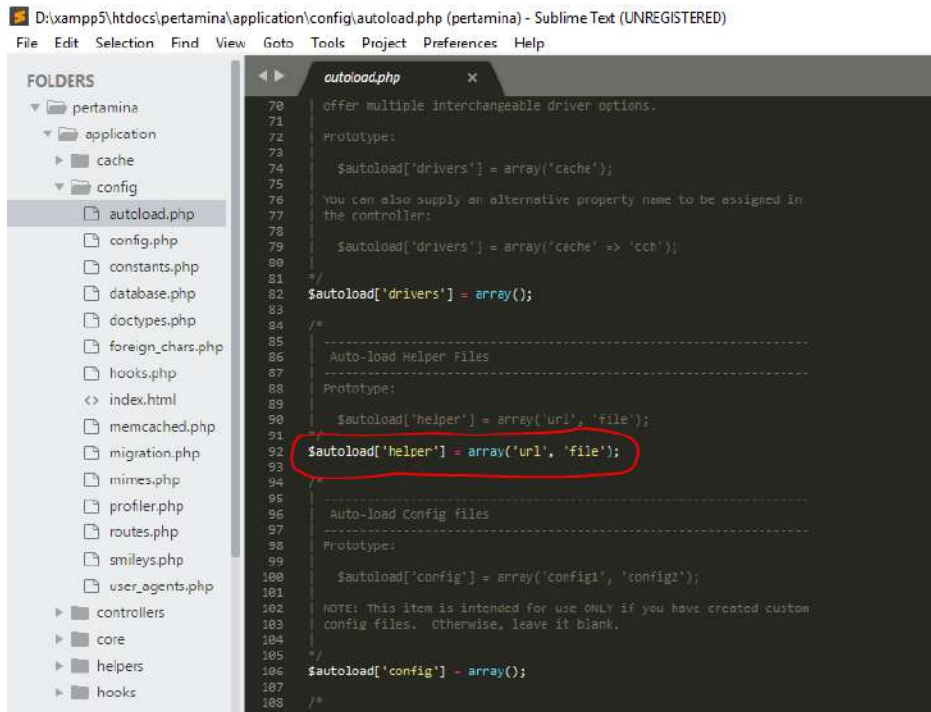
6. Kemudian buka folder *config>autoload.php* pada bagian:

`$autoload['helper'] =`

isi menjadi:

`$autoload['helper'] = array('url','file');`

Konfigurasi pada file ini bertujuan untuk menentukan sumber daya apa yang akan di *load* secara otomatis nantinya.



Gambar 6. 20 Konfigurasi Autoload (Helper)

7. Setelah melakukan langkah konfigurasi di atas lanjut pada langkah selanjutnya yaitu pada bagian *config>database.php* kemudian pada bagian:

```
$db['default'] = array(
    'dsn' => "",
    'hostname' => 'localhost',
    'username' => "",
    'password' => "",
    'database' => "",
    'dbdriver' => 'mysqli',
```

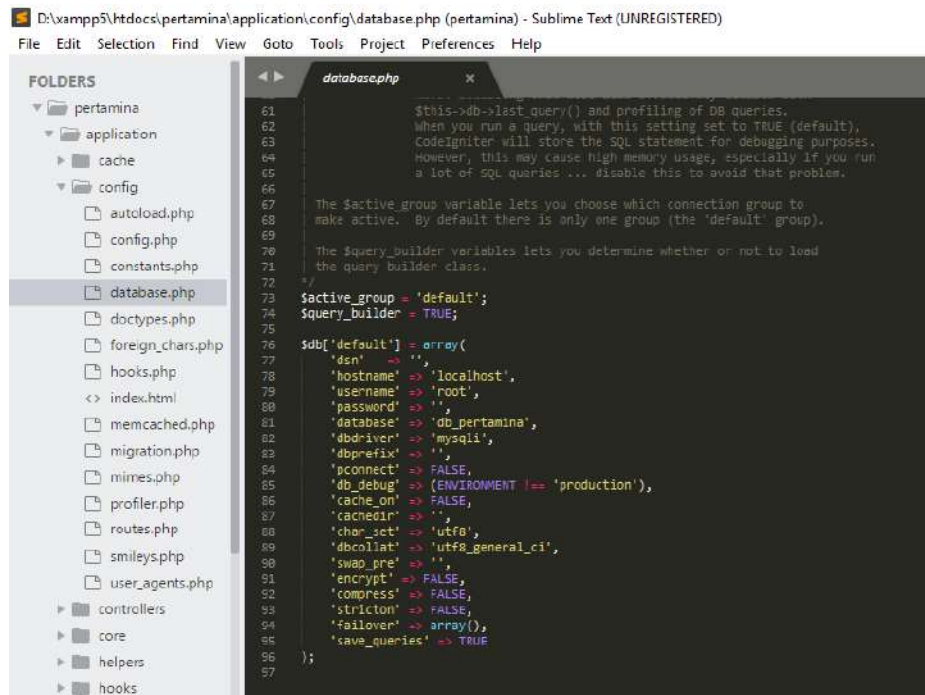
Ubah menjadi:

```
$db['default'] = array(
    'dsn' => "",
```



```
'hostname' => 'localhost',  
'username' => 'root',  
'password' => '',  
'database' => 'db_pertamina',  
'dbdriver' => 'mysqli',
```

Konfigurasi ini berfungsi untuk menghubungkan koneksi *database* yang telah dibuat sebelumnya ke dalam aplikasi *web* yang akan dibuat.



Gambar 6. 21 Konfigurasi *Database*

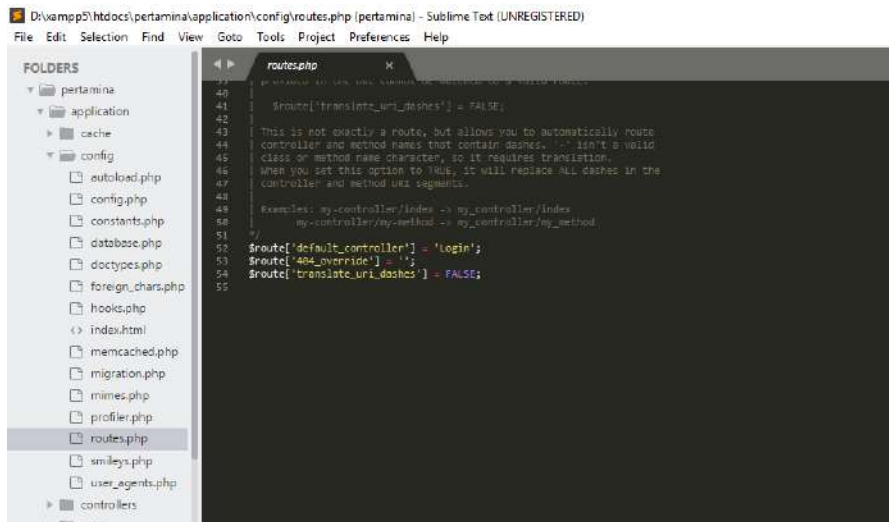
8. Kemudian buka *config>routes.php*, pada bagian:

```
$route['default_controller'] = 'welcome';
```

Ubah menjadi:

```
$route['default_controller'] = 'login';
```

Konfigurasi pada file ini bertujuan untuk menentukan *default controller* yang nantinya akan dipanggil.

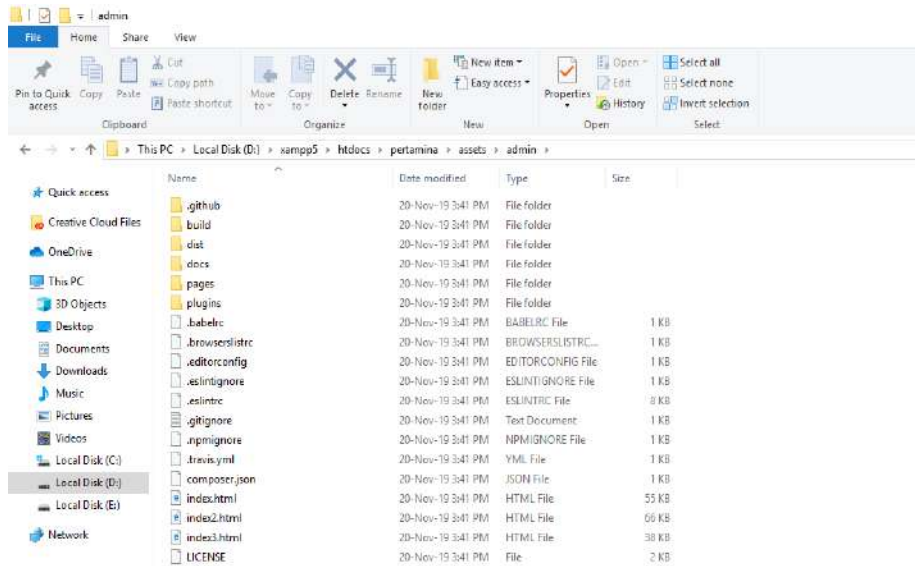


Gambar 6. 22 Konfigurasi *Routes*

6.4 Konfigurasi *Template CSS Bootstrap*

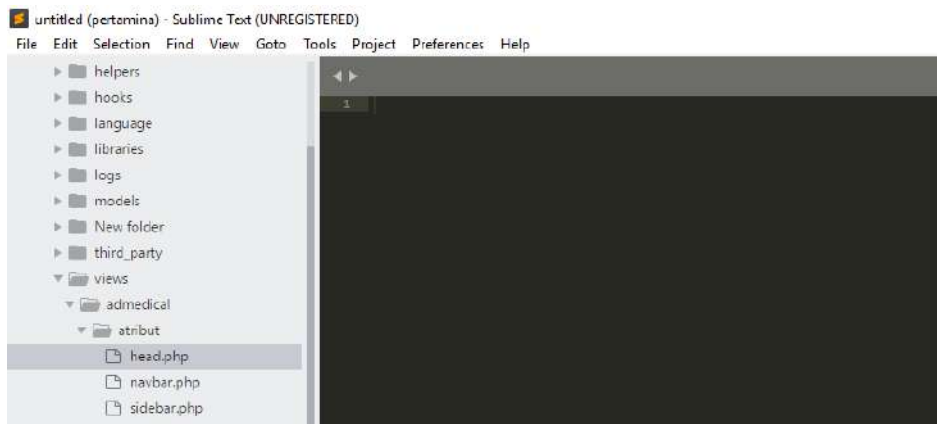
Pada pembuatan sistem informasi, aplikasi ataupun program tentunya harus memiliki tampilan yang menarik. Tujuan dari dibuatnya *template* berfungsi untuk pemanggilan tampilan (*view*) menjadi lebih mudah. Maka dari itu diwajibkan untuk melakukan pengaturan (*setting*) *template* pada *codeigniter* untuk mempermudah pemanggilan *view*. Berikut langkah-langkahnya:

1. Pilih *template* yang anda inginkan atau sesuaikan dengan kebutuhan sistem informasi atau aplikasi yang akan anda bangun. *Template* dapat anda cari dan download di *web browser* yang ada di perangkat anda. Setelah didownload di ekstrak lalu simpan di folder *assets*.

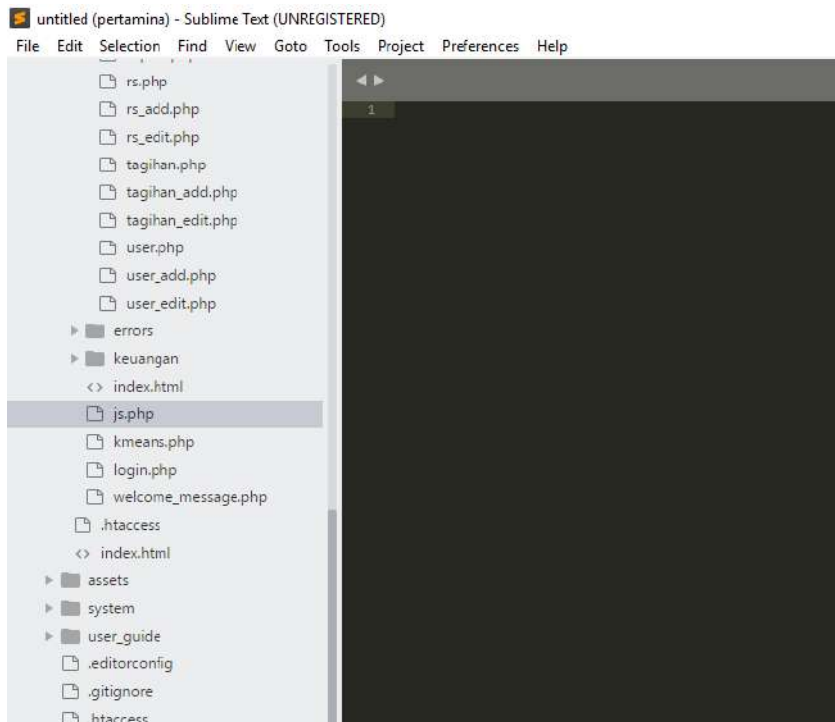


Gambar 6. 23 Folder Template

2. Pisahkan template menjadi 4 bagian, yaitu header, side bar, navbar dan js.



Gambar 6. 24 File Template



Gambar 6. 25 File *Template*

Navbar.php
<pre> <nav class="main-header navbar navbar-expand navbar-white navbar-light"> <!-- Left navbar links --> <ul class="navbar-nav"> <li class="nav-item"> <i class="fas fa- bars"></i> <!-- Right navbar links --> <ul class="navbar-nav ml-auto"> </nav> </pre>

Head.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Pertamina</title>
  <!-- Tell the browser to be responsive to screen width -->
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Font Awesome -->
  <link rel="stylesheet" href="<?=
base_url();?>assets/admin/plugins/fontawesome-free/css/all.min.css">
  <!-- Ionicons -->
  <link rel="stylesheet"
href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
  <!-- overlayScrollbars -->
  <link rel="stylesheet" href="<?=
base_url();?>assets/admin/dist/css/adminlte.min.css">
  <!-- Google Font: Source Sans Pro -->
  <link
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,7
00" rel="stylesheet">
  <!-- DataTables -->
  <link rel="stylesheet" href="<?= base_url();?>assets/admin/plugins/datatables-
bs4/css/dataTables.bootstrap4.css">
</head>
<body class="hold-transition sidebar-mini">
<!-- Site wrapper -->
<div class="wrapper">
  <!-- Navbar -->
  <?php $this->load->view('admedical/atribut/navbar'); ?>
  <!-- /.navbar -->

  <!-- Main Sidebar Container -->
  <?php $this->load->view('admedical/atribut/sidebar'); ?>
```

Sidebar.php

```
<aside class="main-sidebar sidebar-dark-primary elevation-4">
  <!-- Brand Logo -->
  <a href="<?= base_url();?>assets/admin/index3.html" class="brand-link">
    
    <span class="brand-text font-weight-light">Pertamina</span>
  </a>

  <!-- Sidebar -->
  <div class="sidebar">
    <!-- Sidebar user (optional) -->
    <div class="user-panel mt-3 pb-3 mb-3 d-flex">
      <div class="image">
        
      </div>
      <div class="info">
        <a href="#" class="d-block"><?php echo $this->session->userdata('nama');
?></a>
      </div>
    </div>

    <!-- Sidebar Menu -->
    <nav class="mt-2">
      <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview"
role="menu" data-accordion="false">
        <!-- Add icons to the links using the .nav-icon class
        with font-awesome or any other icon font library -->
        <li class="nav-item has-treeview">
          <a href="<?= base_url();?>admedika" class="nav-link">
            <i class="nav-icon fas fa-tachometer-alt"></i>
          <p>
            Dashboard
          </p>
          </a>
        </li>

        <li class="nav-item has-treeview">
          <a href="#" class="nav-link">
            <i class="nav-icon fas fa-user-alt"></i>
          <p>
            Data User
```

```

        <i class="right fas fa-angle-left"></i>
    </p>
</a>
<ul class="nav nav-treeview">
    <li class="nav-item">
        <a href="<?= base_url();?>admedika/user" class="nav-link">
            <i class="far fa-user nav-icon"></i>
            <p>Lihat Data</p>
        </a>
    </li>

    <li class="nav-item">
        <a href="<?= base_url();?>admedika/user_add" class="nav-link">
            <i class="far fa-user nav-icon"></i>
            <p>Tambah Data</p>
        </a>
    </li>
</ul>
</li>

<li class="nav-item has-treeview">
    <a href="#" class="nav-link">
        <i class="nav-icon fa fa-medkit"></i>
        <p>
            Data Rumah Sakit
            <i class="right fas fa-angle-left"></i>
        </p>
    </a>
    <ul class="nav nav-treeview">
        <li class="nav-item">
            <a href="<?= base_url();?>admedika/rs" class="nav-link">
                <i class="far fa-user nav-icon"></i>
                <p>Lihat Data</p>
            </a>
        </li>

        <li class="nav-item">
            <a href="<?= base_url();?>admedika/rs_add" class="nav-link">
                <i class="far fa-user nav-icon"></i>
                <p>Tambah Data</p>
            </a>
        </li>
    </ul>
</li>

<li class="nav-item has-treeview">

```

```

<a href="#" class="nav-link">
  <i class="nav-icon fas fa-book"></i>
  <p>
    Data Tagihan
    <i class="right fas fa-angle-left"></i>
  </p>
</a>
<ul class="nav nav-treeview">
  <li class="nav-item">
    <a href="<?= base_url();?>admedika/tagihan" class="nav-link">
      <i class="far fa-user nav-icon"></i>
      <p>Lihat Data</p>
    </a>
  </li>

  <li class="nav-item">
    <a href="<?= base_url();?>admedika/tagihan_add" class="nav-link">
      <i class="far fa-user nav-icon"></i>
      <p>Tambah Data</p>
    </a>
  </li>
</ul>
</li>
<li class="nav-item has-treeview">
  <a href="#" class="nav-link">
    <i class="nav-icon fas fa-circle"></i>
    <p>
      Kriteria
      <i class="right fas fa-angle-left"></i>
    </p>
  </a>
  <ul class="nav nav-treeview">
    <li class="nav-item">
      <a href="<?= base_url();?>admedika/kriteria" class="nav-link">
        <i class="far fa-user nav-icon"></i>
        <p>Lihat Data</p>
      </a>
    </li>

    <li class="nav-item">
      <a href="<?= base_url();?>admedika/kriteria_add" class="nav-link">
        <i class="far fa-user nav-icon"></i>
        <p>Tambah Data</p>
      </a>
    </li>
  </ul>
</li>
</ul>

```



```

</li>
<li class="nav-item has-treeview">
<a href="#" class="nav-link">
<i class="nav-icon fas fa-list-ul"></i>
<p>
Klasifikasi
<i class="right fas fa-angle-left"></i>
</p>
</a>
<ul class="nav nav-treeview">
<li class="nav-item">
<a href="<?= base_url();?>admedika/Klasifikasi" class="nav-link">
<i class="far fa-user nav-icon"></i>
<p>Lihat Data</p>
</a>
</li>

<li class="nav-item">
<a href="<?= base_url();?>admedika/klasifikasi_add" class="nav-
link">
<i class="far fa-user nav-icon"></i>
<p>Tambah Data</p>
</a>
</li>
</ul>
</li>
<li class="nav-item has-treeview">
<a href="#" class="nav-link">
<i class="nav-icon fas fa-circle"></i>
<p>
Centroid
<i class="right fas fa-angle-left"></i>
</p>
</a>
<ul class="nav nav-treeview">
<li class="nav-item">
<a href="<?= base_url();?>admedika/centeroid" class="nav-link">
<i class="far fa-user nav-icon"></i>
<p>Lihat Data</p>
</a>
</li>

<li class="nav-item">
<a href="<?= base_url();?>admedika/centeroid_add" class="nav-link">
<i class="far fa-user nav-icon"></i>
<p>Tambah Data</p>

```

```

        </a>
    </li>
</ul>
</li>
<li class="nav-item has-treeview">
    <a href="<?= base_url();?>admedika/kmeans" class="nav-link">
        <i class="nav-icon fa fa-search"></i>
        <p>
            Analisis
            <!-- <i class="right fas fa-angle-left"></i> -->
        </p>
    </a>

    <!-- <ul class="nav nav-treeview">
        <li class="nav-item">
            <a href="<?= base_url();?>assets/admin/index.html" class="nav-link">
                <i class="far fa-user nav-icon"></i>
                <p>Lihat Perhitungan</p>
            </a>
        </li>
    </ul> -->

    <li class="nav-item has-treeview">
    <a href="#" class="nav-link">
        <i class="nav-icon fas fa-list-ul"></i>
        <p>
            Hasil Cluster
            <i class="right fas fa-angle-left"></i>
        </p>
    </a>
    <?php
    $sql = "SELECT id_rs, nama_rs FROM rs";
    $query = $this->db->query($sql);
    foreach ($query->result() as $key) { ?>
    <ul class="nav nav-treeview">
        <li class="nav-item">
            <a href="<?= base_url('admedika/view_hasil/'.$key->id_rs);?>"
class="nav-link">
                <i class="far fa-user nav-icon"></i>
                <p><?= $key->nama_rs;?></p>
            </a>
        </li>

        <!-- <li class="nav-item">
            <a href="<?= base_url();?>admedika/centeroid_add" class="nav-link">
                <i class="far fa-user nav-icon"></i>
                <p>Tambah Data</p>

```

```

        </a>
      </li> -->
    </ul>
    <?php } ?>
  </li>
</li>
<!-- <li class="nav-item has-treeview">
  <a href="<?= base_url();?>admedika/view_hasil" class="nav-link">
    <i class="nav-icon fa fa-list-ul"></i>
    <p>
      Hasil Cluster
    </p>
  </a>
</li> -->

<!-- <li class="nav-item has-treeview">
  <a href="<?= base_url();?>admedika/report" class="nav-link">
    <i class="nav-icon fas fa-file"></i>
    <p>
      Laporan
    </p>
  </a>
</li> -->

<li class="nav-item has-treeview">
  <a href="<?= base_url();?>login/logout" class="nav-link">
    <i class="nav-icon fas fa-sign-out-alt"></i>
    <p>
      Logout
    </p>
  </a>
</li>

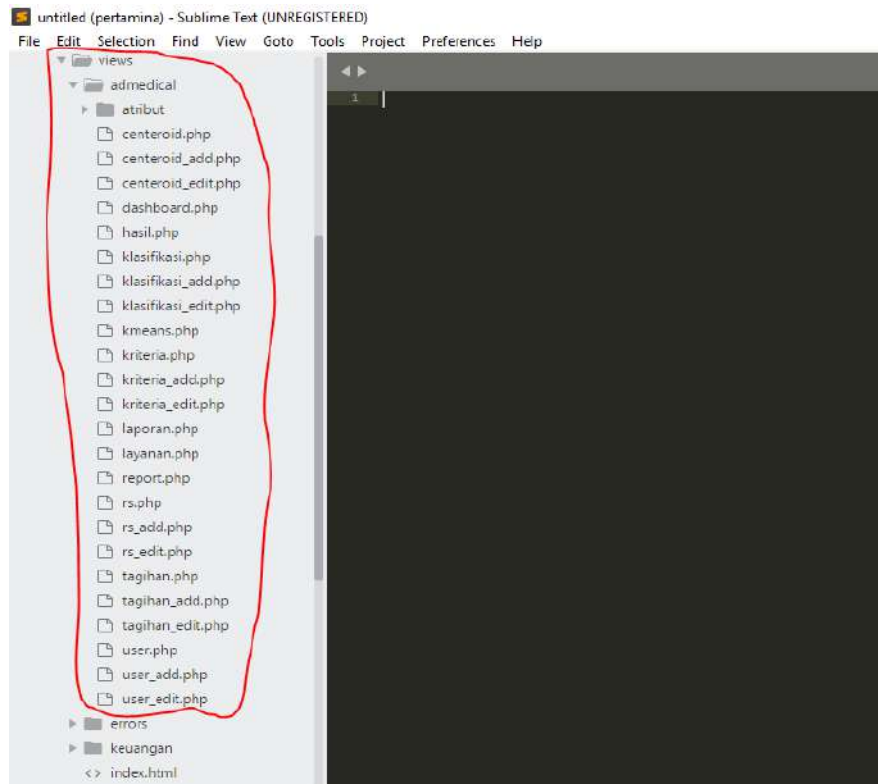
</ul>
</nav>
<!-- /.sidebar-menu -->
</div>
<!-- /.sidebar -->
</aside>

```

Js.php

```
<!-- jQuery -->
<script src="<?=
base_url();?>assets/admin/plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="<?=
base_url();?>assets/admin/plugins/bootstrap/js/bootstrap.bundle.min.js"></script
>
<!-- AdminLTE App -->
<script src="<?= base_url();?>assets/admin/dist/js/adminlte.min.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="<?= base_url();?>assets/admin/dist/js/demo.js"></script>
<!-- DataTables -->
<script src="<?=
base_url();?>assets/admin/plugins/datatables/jquery.dataTables.js"></script>
<script src="<?= base_url();?>assets/admin/plugins/datatables-
bs4/js/dataTables.bootstrap4.js"></script>
<script>
$(function () {
    $('#example1').DataTable();
    $('#example2').DataTable({
        "paging": true,
        "lengthChange": false,
        "searching": false,
        "ordering": true,
        "info": true,
        "autoWidth": false,
    });
});
</script>
```

3. Langkah selanjutnya adalah pada *sublime text* buat folder *view* yang dibutuhkan.

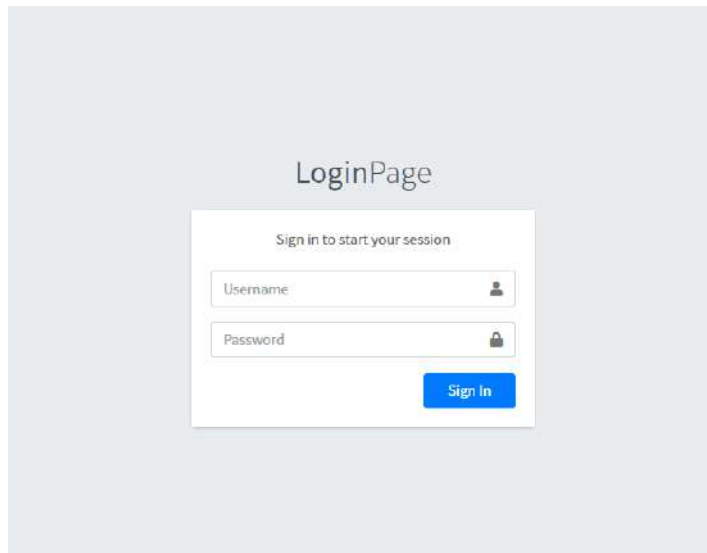


Gambar 6. 26 Tampilan View Yang Dibutuhkan

4. Ada beberapa *view* dalam pembuatan sistem ini:
 - a. *Login*
 - b. *Data User*
 - c. *Data RS (Rumah Sakit)*
 - d. *Data Tagihan*
 - e. *Kriteria*
 - f. *Klasifikasi*
 - g. *Centroid*
 - h. *Analisis*
 - i. *Hasil*
 - j. *Report*

6.5 Tahap Pengkodean

6.5.1 Login



Gambar 6. 27 Tampilan *Login*

1. Controller

Login.php
<pre><?php defined('BASEPATH') OR exit('No direct script access allowed'); class Login extends CI_Controller { public function __construct() { parent::__construct(); \$this->load->model('m_login'); } public function index() { \$this->load->view('login'); } public function login() { </pre>

```

$username = $this->input->post('username');
$password = $this->input->post('password');
$cek = $this->m_login->cek_akun($username,$password);

if ($cek->num_rows() > 0)
{
    $data = $cek->row_array();
    $username = $data['username'];
    $level = $data['level'];
    $iduser = $data['user_id'];
    $nama = $data['nama'];

    $session = array(
        'username' => $username,
        'nama'     => $nama,
        'user_id'  => $iduser,
        'level'    => $level,
        'status'   => "login"
    );

    if ($level === '1') {
        $this->session->set_userdata($session);
        redirect('admedika');
    } elseif ($level === '2') {
        $this->session->set_userdata($session);
        redirect('keuangan');
    }
    } else {
        echo $this->session->set_flashdata('msg','Username or Password is
Wrong');
        redirect('login');
    }
}

public function logout()
{
    $this->session->sess_destroy();
    redirect('login');
}

}
/* End of file Login.php */

```

2. Model

M_login.php
<pre><?php defined('BASEPATH') OR exit('No direct script access allowed'); class M_login extends CI_Model { public function cek_akun(\$username,\$password) { \$this->db->where('username', \$username); \$this->db->where('password', \$password); return \$this->db->get('user'); } } /* End of file M_login.php */</pre>

3. View

Login.php
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <title>PERTAMINA</title> <!-- Tell the browser to be responsive to screen width --> <meta name="viewport" content="width=device-width, initial-scale=1"> <!-- Font Awesome --> <link rel="stylesheet" href="<?= base_url();?>assets/admin/plugins/fontawesome-free/css/all.min.css"> <!-- Icons --> <link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css"> <!-- icheck bootstrap --> <link rel="stylesheet" href="<?= base_url();?>assets/admin/plugins/icheck- bootstrap/icheck-bootstrap.min.css"> <!-- Theme style --> <link rel="stylesheet" href="<?= base_url();?>assets/admin/dist/css/adminlte.min.css"> <!-- Google Font: Source Sans Pro --></pre>


```

<link
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400
i,700" rel="stylesheet">
</head>
<body class="hold-transition login-page">
<div class="login-box">
  <div class="login-logo">
    <a href="<?= base_url();?>"><b>Login</b>Page</a>
  </div>
  <!-- /.login-logo -->
  <div class="card">
    <div class="card-body login-card-body">
      <p class="login-box-msg">Sign in to start your session</p>

      <form action="<?= base_url();?>login/login" method="post">
        <div class="input-group mb-3">
          <input type="text" class="form-control" placeholder="Username"
name="username" required>
          <div class="input-group-append">
            <div class="input-group-text">
              <span class="fas fa-user"></span>
            </div>
          </div>
        </div>
        <div class="input-group mb-3">
          <input type="password" class="form-control" placeholder="Password"
name="password" required>
          <div class="input-group-append">
            <div class="input-group-text">
              <span class="fas fa-lock"></span>
            </div>
          </div>
        </div>
        <div class="row">
          <div class="col-8">

          </div>
          <!-- /.col -->
          <div class="col-4">
            <button type="submit" class="btn btn-primary btn-block">Sign
In</button>
          </div>
          <!-- /.col -->
        </div>
      </form>

```

```

    </div>
    <!-- /.login-card-body -->
  </div>
</div>
<!-- /.login-box -->

<!-- jQuery -->
<script src="<?=
base_url();?>assets/admin/plugins/jquery/jquery.min.js"></script>
    <!-- Bootstrap 4 -->
<script src="<?=
base_url();?>assets/admin/plugins/bootstrap/js/bootstrap.bundle.min.js"></scr
ipt>
<!-- AdminLTE App -->
<script src="<?= base_url();?>assets/admin/dist/js/adminlte.min.js"></script>

</body>
</html>

```

6.5.2 Data User

1. Controller

Admedika.php

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Admedika extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_admedika');
    }
    public function user()
    {
        $data['user'] = $this->m_admedika->getUser();
        $this->load->view('admedical/user', $data);
    }

    public function user_add()
    {
        $this->load->view('admedical/user_add');
    }
}

```

```

public function proses_tambah_user()
{
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $nama = $this->input->post('nama');
    $level = $this->input->post('level');

    $data = array(
        'username' => $username,
        'password' => $password,
        'nama' => $nama,
        'level' => $level
    );

    $this->m_admedika->input_data($data,'user');
    redirect('admedika/user');
}

public function user_edit($id)
{
    $where = array('user_id' => $id);
    $data['user'] = $this->m_admedika->edit_data($where,'user')-
>result();
    $this->load->view('admedical/user_edit',$data);
}

public function proses_edit_user()
{
    $user_id = $this->input->post('user_id');
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $nama = $this->input->post('nama');

    $data = array(
        'username' => $username,
        'password' => $password,
        'nama' => $nama,
        'level' => '2'
    );

    $where = array(
        'user_id' => $user_id
    );

    $this->m_admedika->update_data($where,$data,'user');
}

```

```

        redirect('admedika/user');
    }

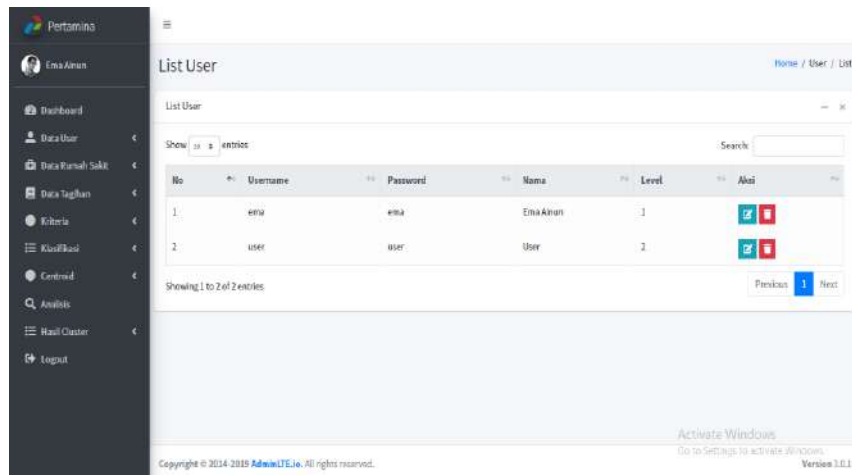
    public function user_delete($id)
    {
        $where = array('user_id' => $id);
        $this->m_admedika->hapus_data($where,'user');
        redirect('admedika/user');    }
    }

```

2. Model

M_admedika.php
<pre> <?php defined('BASEPATH') OR exit('No direct script access allowed'); class M_admedika extends CI_Model { public function getUser() { // \$this->db->where('level', '2'); return \$this->db->get('user')->result(); } } function input_data(\$data,\$table) { \$hasil = \$this->db->insert(\$table,\$data); return \$hasil; } function edit_data(\$where,\$table) { return \$this->db->get_where(\$table,\$where); } function update_data(\$where,\$data,\$table) { \$this->db->where(\$where); \$this->db->update(\$table,\$data); } function hapus_data(\$where,\$table) { \$this->db->where(\$where); \$this->db->delete(\$table); } } </pre>

3. View



Gambar 6. 28 Tampilan Data *User*

```

user.php

<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>List User</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item">User</li>
            <li class="breadcrumb-item">List</li>
          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">

    <!-- Default box -->

```

```

<div class="card">
  <div class="card-header">
    <h3 class="card-title">List User</h3>

    <div class="card-tools">
      <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
        <i class="fas fa-minus"></i></button>
      <button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">
        <i class="fas fa-times"></i></button>
    </div>
  </div>
  <div class="card-body">
    <table class="table table-light table-hover table-bordered"
id="example1">
      <thead class="thead-light">
        <tr>
          <th>No</th>
          <th>Username</th>
          <th>Password</th>
          <th>Nama</th>
          <th>Level</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody>
        <?php
$no = 1;
foreach($user as $u) {
  ?>
  <tr>
    <td><?= $no++; ?></td>
    <td><?= $u->username; ?></td>
    <td><?= $u->password; ?></td>
    <td><?= $u->nama; ?></td>
    <td><?= $u->level; ?></td>
    <td>
      <a href="<?= base_url('admedika/user_edit/'.$u->user_id);?>"
class="btn btn-flat btn-sm btn-info"><i class="fas fa-edit"></i> </a>
      <a href="<?= base_url('admedika/user_delete/'.$u->user_id);?>"
class="btn btn-flat btn-sm btn-danger"><i class="fas fa-trash"></i> </a>
    </td>
  </tr>
  <?php } ?>
    </tbody>
  </div>

```

```

        </table>
    </div>
    <!-- /.card-body -->

    <!-- /.card-footer-->
</div>
<!-- /.card -->

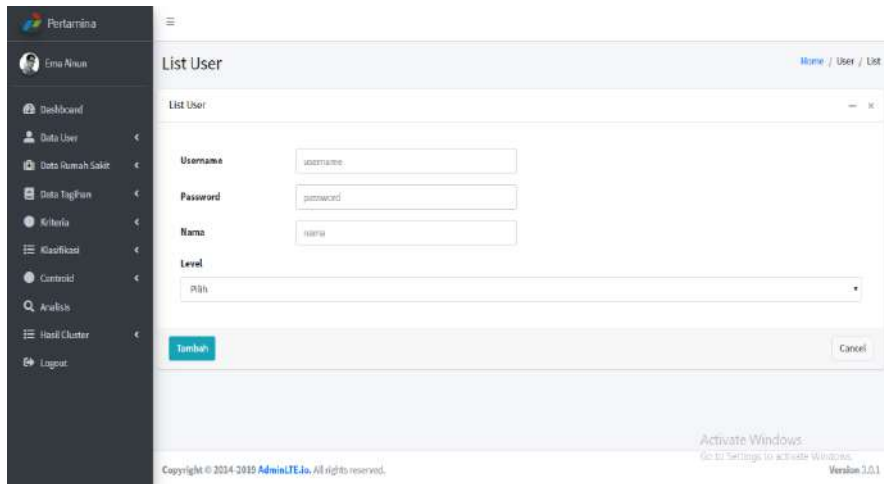
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
        reserved.
    </footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>
</html>

```



Gambar 6. 29 Tampilan Tambah *User*

```

user_add.php
<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>List User</h1>
                </div>
                <div class="col-sm-6">
                    <ol class="breadcrumb float-sm-right">
                        <li class="breadcrumb-item"><a href="#">Home</a></li>
                        <li class="breadcrumb-item">User</li>
                        <li class="breadcrumb-item">List</li>
                    </ol>
                </div>
            </div>
        </div>
    </div><!-- /.container-fluid -->
</section>

<!-- Main content -->
<section class="content">

<!-- Default box -->
<div class="card">

```



```

<div class="card-header">
<h3 class="card-title">List User</h3>

<div class="card-tools">
<button type="button" class="btn btn-tool" data-card-widget="collapse" data-
toggle="tooltip" title="Collapse">
<i class="fas fa-minus"></i></button>
<button type="button" class="btn btn-tool" data-card-widget="remove" data-
toggle="tooltip" title="Remove">
<i class="fas fa-times"></i></button>
</div>
</div>
<div class="card-body">
    <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_tambah_user" method="POST">
        <div class="card-body">

            <div class="form-group row">
                <label for="inputPassword3" class="col-sm-2 col-form-
label">Username</label>
                <div class="col-sm-4">
                    <input type="text" class="form-control" id="inputPassword3"
name="username" placeholder="username">
                </div>
            </div>
            <div class="form-group row">
                <label for="inputPassword3" class="col-sm-2 col-form-
label">Password</label>
                <div class="col-sm-4">
                    <input type="text" class="form-control" id="inputPassword3"
name="password" placeholder="password">
                </div>
            </div>
            <div class="form-group row">
                <label for="inputPassword3" class="col-sm-2 col-form-
label">Nama</label>
                <div class="col-sm-4">
                    <input type="text" class="form-control" id="inputPassword3"
name="nama" placeholder="nama">
                </div>
            </div>
            <label for="level">Level</label>
            <select id="level" name="level" class="form-control select2"
style="width: 100%;">
                <option disabled selected>Pilih</option>
                <option value="1">nama level 1</option>

```

```

        <option value="2">nama level 2</option>
    </select>
</div>
</div>
<!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Tambah</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
<!-- /.card-footer -->
</form>
<!-- </div> -->
<!-- </div> -->
<!-- /.card-body -->

<!-- /.card-footer-->
<!-- </div> -->
<!-- /.card -->

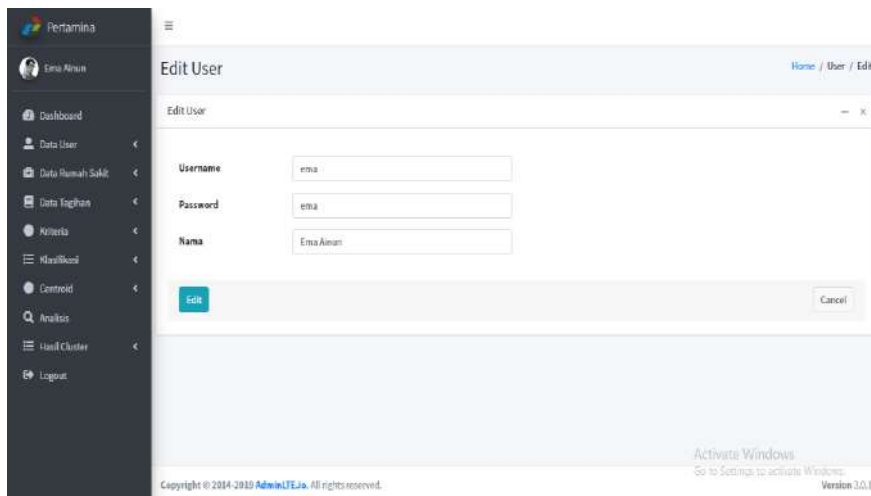
    </section>
    <!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>
</html>

```



Gambar 6. 30 Tampilan *Edit User*

user_edit.php
<pre> <?php \$this->load->view('admedical/atribut/head'); ?> <!-- Content Wrapper. Contains page content --> <div class="content-wrapper"> <!-- Content Header (Page header) --> <section class="content-header"> <div class="container-fluid"> <div class="row mb-2"> <div class="col-sm-6"> <h1>Edit User</h1> </div> <div class="col-sm-6"> <ol class="breadcrumb float-sm-right"> <li class="breadcrumb-item">Home <li class="breadcrumb-item">User <li class="breadcrumb-item">Edit </div> </div> </div> </div><!-- /.container-fluid --> </section> <!-- Main content --> <section class="content"> <!-- Default box --> <div class="card"> </pre>

```

<div class="card-header">
<h3 class="card-title">Edit User</h3>

<div class="card-tools">
<button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
<i class="fas fa-minus"></i></button>
<button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">
<i class="fas fa-times"></i></button>
</div>
</div>

<div class="card-body">
<?php foreach($user as $u) { ?>
<form class="form-horizontal" action="<?=
base_url();?>admedika/proses_edit_user" method="POST">
<div class="card-body">

<div class="form-group row">
<label for="inputPassword3" class="col-sm-2 col-form-
label">Username</label>
<div class="col-sm-4">
<input type="hidden" class="form-control" id="inputPassword3"
name="user_id" placeholder="username" value="<?= $u->user_id; ?>">
<input type="text" class="form-control" id="inputPassword3"
name="username" placeholder="username" value="<?= $u->username;
?>">
</div>
</div>

<div class="form-group row">
<label for="inputPassword3" class="col-sm-2 col-form-
label">Password</label>
<div class="col-sm-4">
<input type="text" class="form-control"
id="inputPassword3" name="password" placeholder="password"
value="<?= $u->password; ?>">
</div>
</div>

<div class="form-group row">
<label for="inputPassword3" class="col-sm-2 col-form-
label">Nama</label>
<div class="col-sm-4">
<input type="text" class="form-control" id="inputPassword3"
name="nama" placeholder="nama" value="<?= $u->nama; ?>">
</div>
</div>

```

```

</div>
<!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Edit</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
<!-- /.card-footer -->
</form>
<?php } ?>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
    </div>
<!-- /.card -->

    </section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- /.wrapper -->

<?php $this->load->view('js');?>
</body>

</html>

```

6.5.3 Data Rumah Sakit

1. Controller

Admedika.php
<pre><?php defined('BASEPATH') OR exit('No direct script access allowed'); class Admedika extends CI_Controller { public function __construct() { parent::__construct(); \$this->load->model('m_admedika'); } public function rs() { \$data['rs'] = \$this->m_admedika->getRs(); \$this->load->view('admedical/rs', \$data); } public function rs_add() { \$this->load->view('admedical/rs_add'); } public function proses_tambah_rs() { \$no_rs = \$this->input->post('no_rs'); \$nama_rs = \$this->input->post('nama_rs'); \$no_rekening = \$this->input->post('no_rekening'); \$data = array('no_rs' => \$no_rs, 'nama_rs' => \$nama_rs, 'no_rekening' => \$no_rekening,); \$this->m_admedika->input_data(\$data,'rs'); redirect('admedika/rs'); } public function rs_edit(\$id) { \$where = array('id_rs' => \$id); \$data['rs'] = \$this->m_admedika->edit_data(\$where,'rs')->result(); } }</pre>

```

        $this->load->view('admedical/rs_edit',$data);
    }

    public function proses_edit_rs()
    {
        $id = $this->input->post('id_rs');
        $no_rs = $this->input->post('no_rs');
        $nama_rs = $this->input->post('nama_rs');
        $no_rekening = $this->input->post('no_rekening');

        $data = array(
            'no_rs' => $no_rs,
            'nama_rs' => $nama_rs,
            'no_rekening' => $no_rekening,
        );

        $where = array(
            'id_rs' => $id
        );

        $this->m_admedika->update_data($where,$data,'rs');
        redirect('admedika/rs');
    }

    public function rs_delete($id)
    {
        $where = array('id_rs' => $id);
        $this->m_admedika->hapus_data($where,'rs');
        redirect('admedika/rs');
    }
}

```

2. Model

M_Admedika.php

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class M_admedika extends CI_Model {

    public function getRs()
    {
        return $this->db->get('rs')->result();
    }
}

```

```

function input_data($data,$table){
    $hasil = $this->db->insert($table,$data);
    return $hasil;
}
function edit_data($where,$table){
    return $this->db->get_where($table,$where);
}
function update_data($where,$data,$table){
    $this->db->where($where);
    $this->db->update($table,$data);
}
function hapus_data($where,$table){
    $this->db->where($where);
    $this->db->delete($table);
}
}

```

3. View

No	No Rumah Sakit	Nama Rumah Sakit	No Rekening	Aksi
1	1014	RS Pertamina Pusat	18090009798	[Edit] [Delete]
2	1015	RS Pertamina Jaya	777150969	[Edit] [Delete]
3	1016	RS Pertamina Plaju	12002323880	[Edit] [Delete]
4	1017	RS Pertamina Cirebon	0446300528	[Edit] [Delete]
5	1018	RS Pertamina Balikpapan	0287271011	[Edit] [Delete]
6	1019	RS Pertamina Balikpapan	000501001641200	[Edit] [Delete]
7	1020	RS Pertamina Cilacap	1300000009473	[Edit] [Delete]

Gambar 6. 31 Tampilan Data Rumah Sakit

```

rs.php
<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">

```



```

<div class="row mb-2">
  <div class="col-sm-6">
    <h1>List Rumah Sakit</h1>
  </div>
  <div class="col-sm-6">
    <ol class="breadcrumb float-sm-right">
      <li class="breadcrumb-item"><a href="#">Home</a></li>
      <li class="breadcrumb-item">Rumah Sakit</li>
      <li class="breadcrumb-item">List</li>
    </ol>
  </div>
</div>
</div><!-- /.container-fluid -->
</section>

<!-- Main content -->
<section class="content">

  <!-- Default box -->
  <div class="card">
    <div class="card-header">
      <h3 class="card-title">List Rumah Sakit</h3>

      <div class="card-tools">
        <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
          <i class="fas fa-minus"></i></button>
        <button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">
          <i class="fas fa-times"></i></button>
      </div>
    </div>
    <div class="card-body">
      <table class="table table-light table-hover table-bordered"
id="example1">
        <thead class="thead-light">
          <tr>
            <th>No</th>
            <th>No Rumah Sakit</th>
            <th>Nama Rumah Sakit</th>
            <th>No Rekening</th>
            <th>Aksi</th>
          </tr>
        </thead>
        <tbody>
          <?php

```

```

        $no = 1;
        foreach($rs as $r) {
            ?>
            <tr>
                <td><?= $no++; ?></td>
                <td><?= $r->no_rs; ?></td>
                <td><?= $r->nama_rs; ?></td>
                <td><?= $r->no_rekening; ?></td>
                <td>
                    <a href="<?= base_url('admedika/rs_edit/'.$r->id_rs);?>"
class="btn btn-flat btn-sm btn-info"><i class="fas fa-edit"></i> </a>
                    <a href="<?= base_url('admedika/rs_delete/'.$r->id_rs);?>"
class="btn btn-flat btn-sm btn-danger"><i class="fas fa-trash"></i> </a>
                </td>
            </tr>
            <?php } ?>
        </tbody>
    </table>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
    reserved.
</footer>

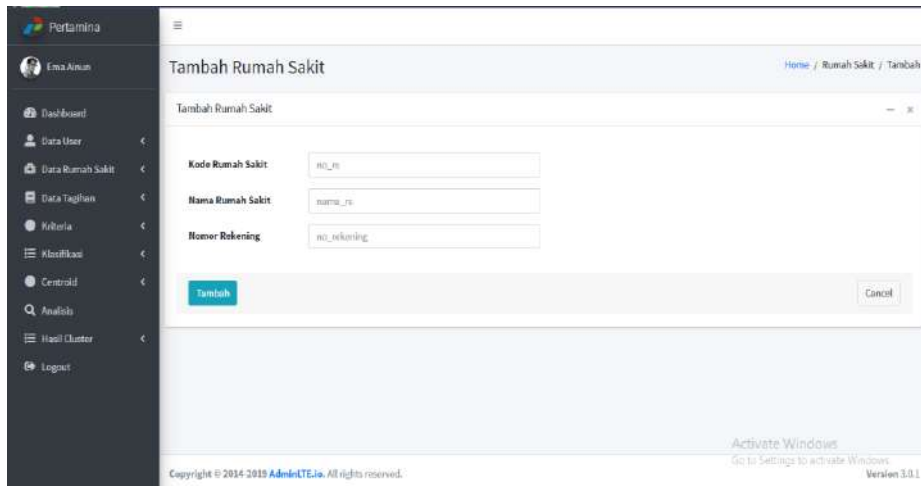
<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- /.wrapper -->

```

```

<?php $this->load->view('js');?>
</body>
</html>

```



Gambar 6. 32 Tampilan Tambah Rumah Sakit

rs_add.php
<pre> <?php \$this->load->view('admedical/atribut/head'); ?> <!-- Content Wrapper. Contains page content --> <div class="content-wrapper"> <!-- Content Header (Page header) --> <section class="content-header"> <div class="container-fluid"> <div class="row mb-2"> <div class="col-sm-6"> <h1>Tambah Rumah Sakit</h1> </div> <div class="col-sm-6"> <ol class="breadcrumb float-sm-right"> <li class="breadcrumb-item">Home <li class="breadcrumb-item">Rumah Sakit <li class="breadcrumb-item">Tambah </div> </div> </div> </div> <!-- /.container-fluid --> </section> </pre>

```

<!-- Main content -->
<section class="content">

<!-- Default box -->
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">Tambah Rumah Sakit</h3>

            <div class="card-tools">
                <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
                    <i class="fas fa-minus"></i></button>
                <button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">
                    <i class="fas fa-times"></i></button>
            </div>
        </div>
        <div class="card-body">
            <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_tambah_rs" method="POST">
                <div class="card-body">

                    <div class="form-group row">
                        <label for="inputPassword3" class="col-sm-2 col-form-label">Kode
Rumah Sakit</label>
                        <div class="col-sm-4">
                            <input type="text" class="form-control" id="inputPassword3"
name="no_rs" placeholder="no_rs">
                        </div>
                    </div>

                    <div class="form-group row">
                        <label for="inputPassword3" class="col-sm-2 col-form-label">Nama
Rumah Sakit</label>
                        <div class="col-sm-4">
                            <input type="text" class="form-control" id="inputPassword3"
name="nama_rs" placeholder="nama_rs">
                        </div>
                    </div>

                    <div class="form-group row">
                        <label for="inputPassword3" class="col-sm-2 col-form-
label">Nomor Rekening</label>
                        <div class="col-sm-4">
                            <input type="text" class="form-control" id="inputPassword3"
name="no_rekening" placeholder="no_rekening">
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>

```

```

        </div>
    </div>
    <!-- /.card-body -->
        <div class="card-footer">
            <button type="submit" class="btn btn-info">Tambah</button>
            <button type="submit" class="btn btn-default float-
right">Cancel</button>
        </div>
    <!-- /.card-footer -->
</form>
    </div>
<!-- /.card-body -->

<!-- /.card-footer-->
    </div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

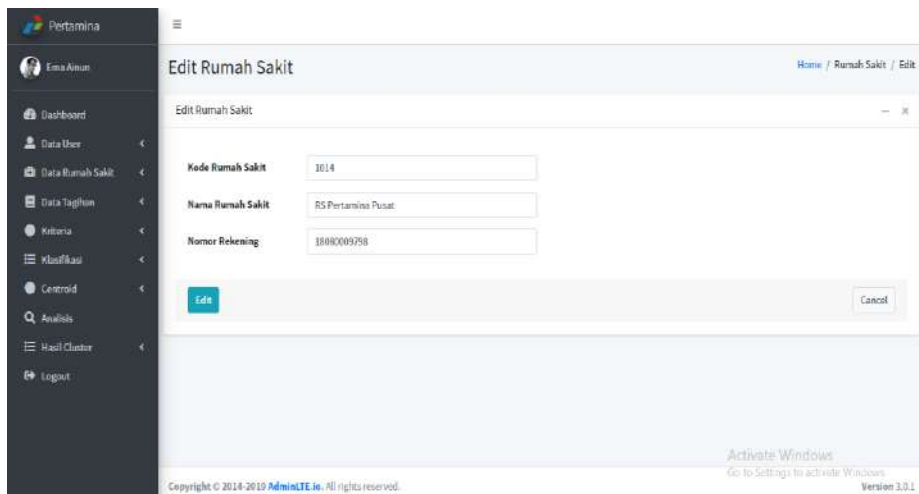
<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>

</html>

```



Gambar 6. 33 Tampilan *Edit* Rumah Sakit

rs_edit.php
<pre> <?php \$this->load->view('admedical/atribut/head'); ?> <!-- Content Wrapper. Contains page content --> <div class="content-wrapper"> <!-- Content Header (Page header) --> <section class="content-header"> <div class="container-fluid"> <div class="row mb-2"> <div class="col-sm-6"> <h1>Edit Rumah Sakit</h1> </div> <div class="col-sm-6"> <ol class="breadcrumb float-sm-right"> <li class="breadcrumb-item">Home <li class="breadcrumb-item">Rumah Sakit <li class="breadcrumb-item">Edit </div> </div> </div><!-- /.container-fluid --> </section> <!-- Main content --> <section class="content"> <!-- Default box --> <div class="card"> </pre>

```

<div class="card-header">
  <h3 class="card-title">Edit Rumah Sakit</h3>

  <div class="card-tools">
    <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
      <i class="fas fa-minus"></i></button>
    <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
      <i class="fas fa-times"></i></button>
    </div>
  </div>
  <div class="card-body">
    <?php foreach($rs as $r) { ?>
      <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_edit_rs" method="POST">
        <div class="card-body">

          <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-label">Kode
Rumah Sakit</label>
            <div class="col-sm-4">
              <input type="hidden" class="form-control" id="inputPassword3"
name="id_rs" placeholder="id_rs" value="<?= $r->id_rs; ?>">
              <input type="text" class="form-control" id="inputPassword3"
name="no_rs" placeholder="no_rs" value="<?= $r->no_rs; ?>">
            </div>
          </div>

          <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-label">Nama
Rumah Sakit</label>
            <div class="col-sm-4">
              <input type="text" class="form-control" id="inputPassword3"
name="nama_rs" placeholder="nama_rs" value="<?= $r->nama_rs; ?>">
            </div>
          </div>

          <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-label">Nomor
Rekening</label>
            <div class="col-sm-4">
              <input type="text" class="form-control" id="inputPassword3"
name="no_rekening" placeholder="no_rekening" value="<?= $r->no_rekening;
?>">
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

</div>
<!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Edit</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
<!-- /.card-footer -->
    </form>
    <?php } ?>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
    </div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>

</html>

```


6.5.4 Data Tagihan

1. Controller

Admedika.php

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Admedika extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_admedika');
    }

    public function tagihan()
    {
        $data['tagihan'] = $this->m_admedika->getTagihan();
        $this->load->view('admedical/tagihan', $data);
    }

    public function tagihan_add()
    {
        $data['rs'] = $this->m_admedika->getRs();
        $data['layanan'] = $this->m_admedika->getLayanan();
        $this->load->view('admedical/tagihan_add', $data);
    }

    public function proses_tambah_tagihan()
    {
        $id_rs      = $this->input->post('id_rs');
        $tgl_masuk   = $this->input->post('tgl_masuk');
        $no_debit_nota = $this->input->post('no_debit_nota');
        $id_layanan  = $this->input->post('id_layanan');
        $no_faktur   = $this->input->post('no_faktur');
        $dpp         = $this->input->post('dpp');
        $tagihan     = $this->input->post('tagihan');
        $scn         = $this->input->post('credit_nota');

        $data = array(
            'id_rs'      => $id_rs,
            'tgl_masuk'   => date("Y-m-d"),
            'no_debit_nota' => $no_debit_nota,
            'id_layanan'  => $id_layanan,
            'no_faktur'   => $no_faktur,
```

```

        'dpp'          => $dpp,
        'tagihan'      => $tagihan,
        'credit_nota'  => $cn
    );
    // print_r($data); die;

    $this->m_admedika->input_data($data,'tagihan');
    redirect('admedika/tagihan');
}

public function tagihan_edit($id)
{
    $where = array('id_tagihan' => $id);
    $data['tagihan'] = $this->m_admedika-
>edit_data($where,'tagihan')->result();
    $data['rs'] = $this->m_admedika->getRs();
    $data['layanan'] = $this->m_admedika->getLayanan();
    $this->load->view('admedical/tagihan_edit',$data);
}

public function proses_edit_tagihan()
{
    $id_tagihan = $this->input->post('id_tagihan');
    $tgl_masuk = $this->input->post('tgl_masuk');
    $id_rs = $this->input->post('id_rs');
    $no_debit_nota = $this->input->post('no_debit_nota');
    $id_layanan = $this->input->post('id_layanan');
    $no_faktur = $this->input->post('no_faktur');
    $dpp = $this->input->post('dpp');
    $tagihan = $this->input->post('tagihan');
    $cn = $this->input->post('credit_nota');

    $data = array(
        'tgl_masuk'      => $tgl_masuk,
        'id_rs'          => $id_rs,
        'no_debit_nota'  => $no_debit_nota,
        'id_layanan'     => $id_layanan,
        'no_faktur'      => $no_faktur,
        'dpp'            => $dpp,
        'tagihan'        => $tagihan,
        'credit_nota'    => $cn
    );

    $where = array(
        'id_tagihan'     => $id_tagihan
    );

```

```

        $this->m_admedika->update_data($where,$data,'tagihan');
        redirect('admedika/tagihan');
    }

    public function tagihan_delete($id)
    {
        $where = array('id_tagihan' => $id);
        $this->m_admedika->hapus_data($where,'tagihan');
        redirect('admedika/tagihan');
    }

```

2. Model

M.Admedika.php

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class M_admedika extends CI_Model {
    public function getTagihan()
    {
        $this->db->select('a.id_tagihan,b.nama_rs, a.no_debit_nota,
        c.kode, c.nm_layanan, a.no_faktur, a.dpp, a.tagihan, a.credit_nota,
        a.pembayaran, a.tgl_masuk');
        $this->db->from('v_tagihan a');
        $this->db->join('rs b', 'b.id_rs = a.id_rs', 'left');
        $this->db->join('layanan c', 'c.id_layanan = a.id_layanan', 'left');
        return $this->db->get()->result();
    }

    public function getLayanan()
    {
        return $this->db->get('layanan')->result();
    }

    public function getRs()
    {
        return $this->db->get('rs')->result();
    }

    function input_data($data,$table){
        $hasil = $this->db->insert($table,$data);
        return $hasil;
    }
}

```

```

function edit_data($where,$table){
    return $this->db->get_where($table,$where);
}

function update_data($where,$data,$table){
    $this->db->where($where);
    $this->db->update($table,$data);
}

function hapus_data($where,$table){
    $this->db->where($where);
    $this->db->delete($table);
}

```

3. View

No	Tgl Masuk	Nama RS	No Debit	No Faktur	Jenis Layanan	DPP	Tagihan	Credit	Pembayaran	Aksi
1	2020-01-15	RS Pertamina Pusat	148803/1903006		Rawat Inap	4890768	9012917	1409	9011508	[Edit] [Hapus]
2	2020-01-15	RS Pertamina Pusat	148803/1903006		Rawat Inap	4890768	4036370	163928	3872458	[Edit] [Hapus]
3	2020-01-15	RS Pertamina Pusat	148803/1903009		Rawat Inap	4890768	21279521	201609	21068962	[Edit] [Hapus]
4	2020-01-15	RS Pertamina Pusat	148806/1903013		Rawat Inap	4890768	5532136	201609	5335517	[Edit] [Hapus]
5	2020-01-15	RS Pertamina Pusat	148806/1903005		Rawat Inap	4890768	29308717	296020	29113167	[Edit] [Hapus]
6	2020-01-15	RS Pertamina Pusat	148806/1903008		Rawat Inap	4890768	87903187	141051	8564336	[Edit] [Hapus]
7	2020-01-15	RS Pertamina Pusat	148801/1903020	010.003-19-012545174	Rawat Jalan	68650	2004515	201609	1842706	[Edit] [Hapus]
8	2020-01-15	RS Pertamina Pusat	148801/1903027	010.003-19-012545173	Rawat Jalan	1812230	13227959	201609	12068590	[Edit] [Hapus]
9	2020-01-15	RS Pertamina Pusat	148801/1903029	010.003-19-012545175	Rawat Jalan	6275270	31858326	68229	3179971	[Edit] [Hapus]
10	2020-01-15	RS Pertamina Pusat	140.001/1904020	010.003-19-01257504	Rawat Jalan	9500900	343321624	192000	34126024	[Edit] [Hapus]

Gambar 6. 34 Tampilan Data Tagihan

tagihan.php

```

<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">

```

```

        <h1>List Tagihan</h1>
    </div>
    <div class="col-sm-6">
        <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item">Tagihan</li>
            <li class="breadcrumb-item">List</li>
        </ol>
    </div>
</div>
</div><!-- /.container-fluid -->
</section>

<!-- Main content -->
<section class="content">
<!-- Default box -->
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">List Tagihan</h3>

            <div class="card-tools">
                <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
                    <i class="fas fa-minus"></i></button>
                <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
                    <i class="fas fa-times"></i></button>
            </div>
        </div>
        <div class="card-body">
            <table class="table table-light table-hover table-bordered" id="example1">
                <thead class="thead-light">
                    <a href="<?php echo base_url() ?>admedika/laporan" button type=""
class="btn btn-danger float-right"><i class="fa fa-print">Cetak</i></a>
                <tr>
                    <th>No</th>
                    <th>Tgl Masuk</th>
                    <th>Nama RS</th>
                    <th>No Debit</th>
                    <th>No Faktur</th>
                    <th>Jenis Layanan</th>
                    <th>DPP</th>
                    <th>Tagihan</th>
                    <th>Credit Nota</th>
                    <th>Pembayaran</th>
                    <th>Aksi</th>
                </tr>
            </table>
        </div>
    </div>

```

```

        </tr>
    </thead>
    <tbody>
        <?php
            $no = 1;
            foreach($tagihan as $t) {
                ?>
                <tr>
                    <td><?= $no++; ?></td>
                    <td><?= $t->tgl_masuk; ?></td>
                    <td><?= $t->nama_rs; ?></td>
                    <td><?= $t->no_debit_nota; ?></td>
                    <td><?= $t->no_faktur; ?></td>
                    <td><?= $t->nm_layanan; ?></td>
                    <td><?= $t->dpp; ?></td>
                    <td><?= $t->tagihan; ?></td>
                    <td><?= $t->credit_nota; ?></td>
                    <td><?= $t->pembayaran; ?></td>
                    <td>
                        <a href="<?= base_url('admedika/tagihan_edit/'.$t-
>id_tagihan);?>" class="btn btn-flat btn-sm btn-info"><i class="fas fa-edit"></i>
</a>
                        <a href="<?= base_url('admedika/tagihan_delete/'.$t-
>id_tagihan);?>" class="btn btn-flat btn-sm btn-danger"><i class="fas fa-
trash"></i> </a>
                    </td>
                </tr>
            <?php } ?>
        </tbody>
    </table>
</div>
<!-- /.card-body -->
<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>

```

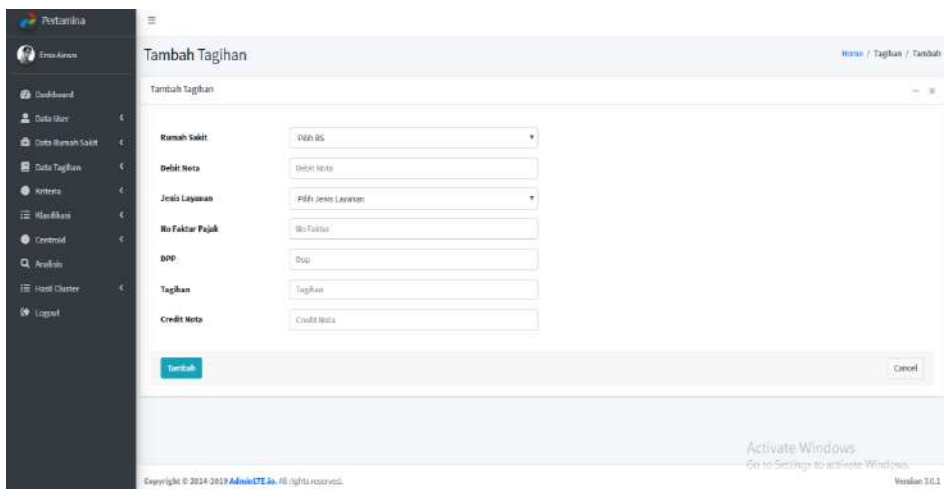
```

<strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
  <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- /.wrapper -->

<?php $this->load->view('js');?>
</body>
</html>

```



Gambar 6. 35 Tampilan Tambah Tagihan

tagihan_add.php

```

<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">

```

```

        <h1>Tambah Tagihan</h1>
    </div>
    <div class="col-sm-6">
        <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item">Tagihan</li>
            <li class="breadcrumb-item">Tambah</li>
        </ol>
    </div>
</div>
</div>
</div><!-- /.container-fluid -->
</section>

<!-- Main content -->
    <section class="content">

        <!-- Default box -->
        <div class="card">
            <div class="card-header">
                <h3 class="card-title">Tambah Tagihan</h3>

                <div class="card-tools">
                    <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
                        <i class="fas fa-minus"></i></button>
                    <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
                        <i class="fas fa-times"></i></button>
                </div>
            </div>
            <div class="card-body">
                <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_tambah_tagihan" method="POST">
                    <div class="card-body">
                        <div class="form-group row">
                            <label for="inputEmail3" class="col-sm-2 col-form-label">Rumah
Sakit</label>
                            <div class="col-sm-4">
                                <select name="id_rs" id="id_rs" class="form-control">
                                    <option value="#">Pilih RS</option>
                                    <?php foreach($rs as $l) { ?>
                                        <option value="<?= $l->id_rs; ?>"><?= $l->nama_rs; ?></option>
                                    <?php } ?>
                                </select>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```



```

        <!-- <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">Tanggal
Masuk</label>
        <div class="col-sm-4">
        <input type="text" class="form-control" id="inputPassword3"
name="tgl_masuk" placeholder="Debit Nota">
        </div>
</div> -->
        <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">Debit
Nota</label>
        <div class="col-sm-4">
        <input type="text" class="form-control" id="inputPassword3"
name="no_debit_nota" placeholder="Debit Nota">
        </div>
</div>
        <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">Jenis
Layanan</label>
        <div class="col-sm-4">
        <select name="id_layanan" id="id_layanan" class="form-control">
        <option value="#">Pilih Jenis Layanan</option>
        <?php foreach($layanan as $l) { ?>
        <option value="<?=$l->id_layanan; ?>"><?=$l->nm_layanan;
?></option>
        <?php } ?>
</select>
</div>
</div>
        <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">No Faktur
Pajak</label>
        <div class="col-sm-4">
        <input type="text" class="form-control" id="inputPassword3"
name="no_faktur" placeholder="No Faktur">
        </div>
</div>
        <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-
label">DPP</label>
        <div class="col-sm-4">
        <input type="number" class="form-control" id="inputPassword3"
name="dpp" placeholder="Dpp">
        </div>
</div>
        <div class="form-group row">

```

```

        <label for="inputPassword3" class="col-sm-2 col-form-
label">Tagihan</label>
        <div class="col-sm-4">
            <input type="number" class="form-control" id="inputPassword3"
name="tagihan" placeholder="Tagihan">
        </div>
        </div>
        <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-label">Credit
Nota</label>
            <div class="col-sm-4">
                <input type="number" class="form-control" id="inputPassword3"
name="credit_notas" placeholder="Credit Nota">
            </div>
        </div>
    </div>
    <!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Tambah</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
    <!-- /.card-footer -->
</form>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->

```

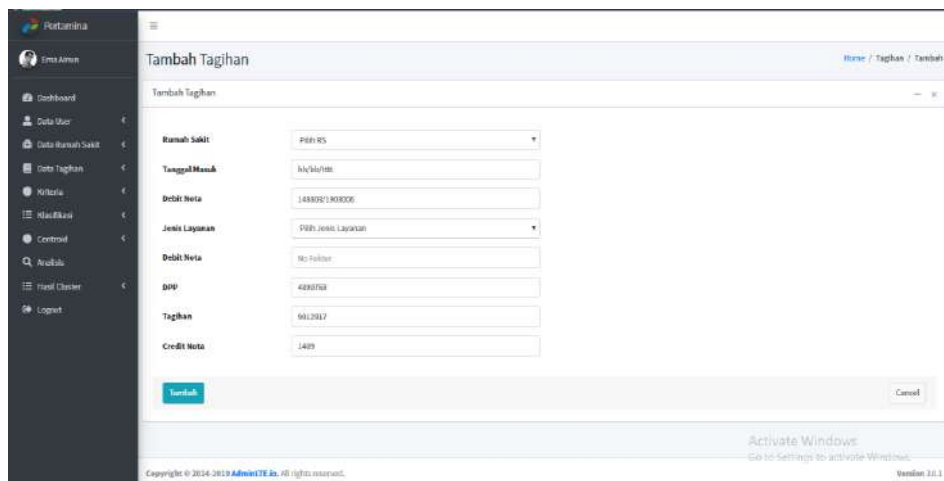
```

<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>

</html>

```



Gambar 6. 36 Tampilan *Edit Tagihan*

```

tagihan_edit.php

<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Tambah Tagihan</h1>
                </div>
                <div class="col-sm-6">
                    <ol class="breadcrumb float-sm-right">
                        <li class="breadcrumb-
item"><a href="#">Home</a></li>

```

```

item">Tagihan</li>
item">Tambah</li>
</ol>
</div>
</div>
</div><!-- /.container-fluid -->
</section>

<!-- Main content -->
<section class="content">

    <!-- Default box -->
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">Tambah Tagihan</h3>

            <div class="card-tools">
                <button type="button" class="btn btn-
tool" data-card-widget="collapse" data-toggle="tooltip"
                    title="Collapse">
                    <i class="fas fa-
minus"></i></button>

                <button type="button" class="btn btn-
tool" data-card-widget="remove" data-toggle="tooltip"
                    title="Remove">
                    <i class="fas fa-
times"></i></button>
            </div>
        </div>
        <div class="card-body">
            <?php foreach($tagihan as $t) { ?>
                <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_edit_tagihan" method="POST">
                    <div class="card-body">
                        <div class="form-group row">
                            <label
for="inputEmail3" class="col-sm-2 col-form-label">Rumah Sakit</label>
                            <div class="col-sm-4">
                                <input type="hidden" class="form-control"
id="inputPassword3" name="id_tagihan" placeholder="Debit Nota" value="<?=
$t->id_tagihan; ?>">
                                <select name="id_rs" id="id_rs" class="form-control" value="<?= $t-
>id_rs; ?>">
                                    <option value="#">Pilih RS</option>

```

```

        <?php foreach($rs as $l) { ?>
        <option value="<?= $l->id_rs; ?>"><?= $l->nama_rs; ?></option>
<?php } ?>
</select>
    </div>
</div>
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">Tanggal
Masuk</label>
        <div class="col-sm-4">
            <input type="date" class="form-control" id="inputPassword3"
name="tgl_masuk" placeholder="tgl masuk value="<?= $t->tgl_masuk; ?>">
</div>
</div>
        <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-label">Debit
Nota</label>
            <div class="col-sm-4">
                <input type="text" class="form-control" id="inputPassword3"
name="no_debit_nota" placeholder="Debit Nota" value="<?= $t-
>no_debit_nota; ?>">
            </div>
        </div>
        <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-label">Jenis
Layanan</label>
            <div class="col-sm-4">
                <select name="id_layanan" id="id_layanan" class="form-control">
                <option value="#">Pilih Jenis Layanan</option>
                <?php foreach($layanan as $l) { ?>
                <option value="<?= $l->id_layanan; ?>"><?= $l->nm_layanan;
?></option>
                <?php } ?>
            </select>
            </div>
        </div>
        <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-label">Debit
Nota</label>
            <div class="col-sm-4">
                <input type="text" class="form-control" id="inputPassword3"
name="no_faktur" placeholder="No Faktur" value="<?= $t->no_faktur; ?>">
            </div>
        </div>
    </div>
</div>

```

```

        <label for="inputPassword3" class="col-sm-2 col-form-
label">DPP</label>
        <div class="col-sm-4">
            <input type="number" class="form-control" id="inputPassword3"
name="dpp" placeholder="Dpp" value="<?= $t->dpp; ?>">
        </div>
    </div>
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-
label">Tagihan</label>
        <div class="col-sm-4">
            <input type="number" class="form-control" id="inputPassword3"
name="tagihan" placeholder="Tagihan" value="<?= $t->tagihan; ?>">
        </div>
    </div>
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">Credit
Nota</label>
        <div class="col-sm-4">
            <input type="number" class="form-control" id="inputPassword3"
name="credit_nota" placeholder="Credit Nota" value="<?= $t->credit_nota;
?>">
        </div>
    </div>
</div>
<!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Tambah</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
<!-- /.card-footer -->
</form>
    <?php } ?>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

```

```

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
    reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>

</html>

```

6.5.5 Kriteria

1. Controller

Admedika.php
<pre> <?php defined('BASEPATH') OR exit('No direct script access allowed'); class Admedika extends CI_Controller { public function __construct() { parent::__construct(); \$this->load->model('m_admedika'); } public function kriteria() { \$data['kriteria'] = \$this->m_admedika->getKriteria(); \$this->load->view('admedical/kriteria', \$data); } </pre>

```

public function kriteria_add()
{
    $this->load->view('admedical/kriteria_add');
}

public function proses_tambah_kriteria()
{
    $nama_kriteria = $this->input->post('nama_kriteria');
    $atribut = $this->input->post('atribut');

    $data = array(
        'nama_kriteria' => $nama_kriteria,
        'atribut' => $atribut,
    );

    $this->m_admedika->input_data($data,'tb_kriteria');
    redirect('admedika/kriteria');
}

public function kriteria_edit($id)
{
    $where = array('id_kriteria' => $id);
    $data['kriteria'] = $this->m_admedika->edit_data($where,'tb_kriteria')-
>result();
    $this->load->view('admedical/kriteria_edit',$data);
}

public function proses_edit_kriteria()
{
    $id = $this->input->post('id_kriteria');
    $nama_kriteria = $this->input->post('nama_kriteria');
    $atribut = $this->input->post('atribut');

    $data = array(
        'nama_kriteria' => $nama_kriteria,
        'atribut' => $atribut,
    );

    $where = array(
        'id_kriteria' => $id
    );

    $this->m_admedika->update_data($where,$data,'tb_kriteria');
}

```



```

        redirect('admedika/kriteria');
    }

    public function kriteria_delete($id)
    {
        $where = array('id_kriteria' => $id);
        $this->m_admedika->hapus_data($where,'tb_kriteria');
        redirect('admedika/kriteria');
    }

```

2. Model

M.Admedika.php

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class M_admedika extends CI_Model {

    public function getKriteria()
    {
        return $this->db->get('tb_kriteria')->result();
    }

    function input_data($data,$table){
        $hasil = $this->db->insert($table,$data);
        return $hasil;
    }

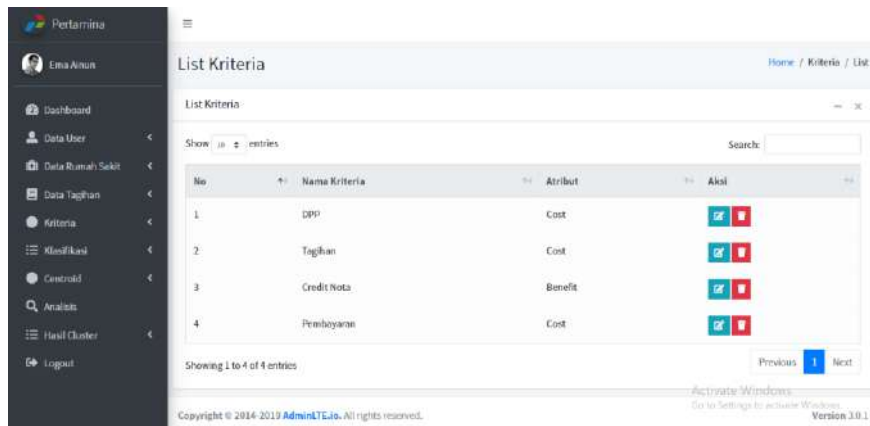
    function edit_data($where,$table){
        return $this->db->get_where($table,$where);
    }

    function update_data($where,$data,$table){
        $this->db->where($where);
        $this->db->update($table,$data);
    }

    function hapus_data($where,$table){
        $this->db->where($where);
        $this->db->delete($table);
    }
}

```

3. View



Gambar 6. 37 Tampilan Kriteria

```
kriteria.php

<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>List Kriteria</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item">Kriteria</li>
            <li class="breadcrumb-item">List</li>
          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">

    <!-- Default box -->
```

```

<div class="card">
  <div class="card-header">
    <h3 class="card-title">List Kriteria</h3>

    <div class="card-tools">
      <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
        <i class="fas fa-minus"></i></button>
      <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
        <i class="fas fa-times"></i></button>
    </div>
  </div>
  <div class="card-body">
    <table class="table table-light table-hover table-bordered" id="example1">
      <thead class="thead-light">
        <tr>
          <th>No</th>
          <th>Nama Kriteria</th>
          <th>Atribut</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody>
        <?php
        $no = 1;
        foreach($skriteria as $kr) {
          ?>
          <tr>
            <td><?= $no++; ?></td>
            <td><?= $kr->nama_kriteria; ?></td>
            <td><?= $kr->atribut; ?></td>
            <td>
              <a href="<?= base_url('admedika/kriteria_edit/'.$kr-
>id_kriteria);?>" class="btn btn-flat btn-sm btn-info"><i class="fas fa-edit"></i>
</a>
              <a href="<?= base_url('admedika/kriteria_delete/'.$kr-
>id_kriteria);?>" class="btn btn-flat btn-sm btn-danger"><i class="fas fa-
trash"></i> </a>
            </td>
          </tr>
        <?php } ?>
      </tbody>
    </table>
  </div>
<!-- /.card-body -->

```

```

        <!-- /.card-footer-->
    </div>
    <!-- /.card -->

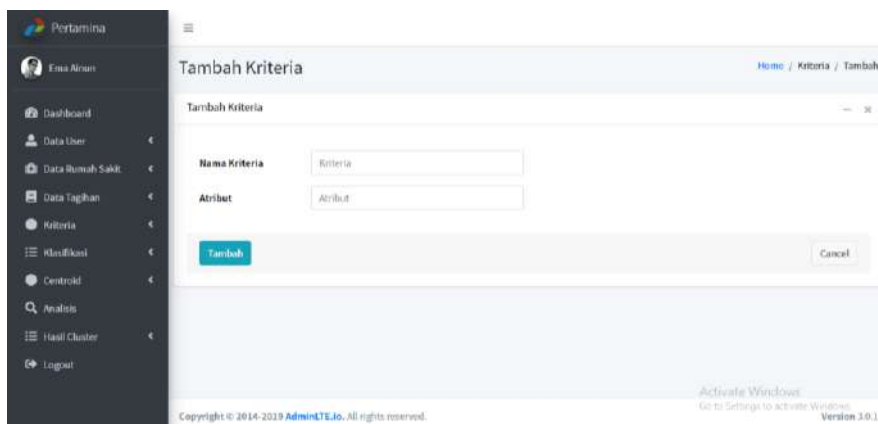
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
    reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- /.wrapper -->

<?php $this->load->view('js');?>
</body>
</html>

```



Gambar 6. 38 Tampilan Tambah Kriteria

kriteria_add.php

```

<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Tambah Kriteria</h1>
                </div>
                <div class="col-sm-6">
                    <ol class="breadcrumb float-sm-right">
                        <li class="breadcrumb-item"><a href="#">Home</a></li>
                        <li class="breadcrumb-item">Kriteria</li>
                        <li class="breadcrumb-item">Tambah</li>
                    </ol>
                </div>
            </div>
        </div>
    </div>
<!-- Main content -->
<section class="content">
    <!-- Default box -->
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">Tambah Kriteria</h3>
        </div>
        <div class="card-tools">
            <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
                <i class="fas fa-minus"></i></button>
            <button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">
                <i class="fas fa-times"></i></button>
        </div>
        <div class="card-body">
            <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_tambah_kriteria" method="POST">
                <div class="card-body">

```

```

        <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-
label">Nama Kriteria</label>
            <div class="col-sm-4">
                <input type="text" class="form-control" id="inputPassword3"
name="nama_kriteria" placeholder="Kriteria">
            </div>
        </div>
        <div class="form-group row">
            <label for="inputPassword3" class="col-sm-2 col-form-
label">Atribut</label>
            <div class="col-sm-4">
                <input type="text" class="form-control" id="inputPassword3"
name="atribut" placeholder="Atribut">
            </div>
        </div>
    </div>
    <!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Tambah</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
    <!-- /.card-footer -->
</form>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
    </div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

```

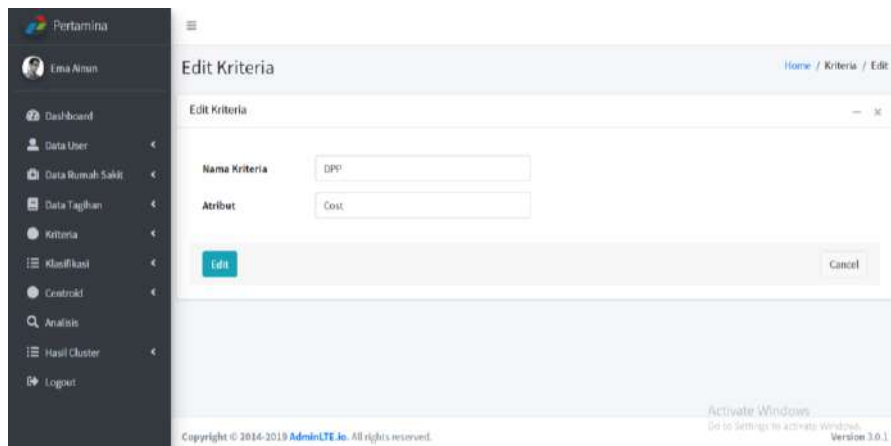
```

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>

</html>

```



Gambar 6. 39 Tampilan *Edit Kriteria*

kriteria_edit.php
<pre> <?php \$this->load->view('admedical/atribut/head'); ?> <!-- Content Wrapper. Contains page content --> <div class="content-wrapper"> <!-- Content Header (Page header) --> <section class="content-header"> <div class="container-fluid"> <div class="row mb-2"> <div class="col-sm-6"> <h1>Edit Kriteria</h1> </div> <div class="col-sm-6"> <ol class="breadcrumb float-sm-right"> <li class="breadcrumb-item">Home </pre>

```

                <li class="breadcrumb-item">Kriteria</li>
                <li class="breadcrumb-item">Edit</li>
            </ol>
        </div>
    </div>
</div>
</div><!-- /.container-fluid -->
</section>

<!-- Main content -->
<section class="content">

<!-- Default box -->
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">Edit Kriteria</h3>

            <div class="card-tools">
                <button type="button" class="btn btn-tool" data-card-
widget="collapse" data-toggle="tooltip" title="Collapse">
                    <i class="fas fa-minus"></i></button>
                <button type="button" class="btn btn-tool" data-card-
widget="remove" data-toggle="tooltip" title="Remove">
                    <i class="fas fa-times"></i></button>
            </div>
        </div>
        <div class="card-body">
            <?php foreach($kriteria as $kr) { ?>
                <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_edit_kriteria" method="POST">
                    <div class="card-body">

                        <div class="form-group row">
                            <label for="inputPassword3" class="col-sm-2 col-form-label">Nama
Kriteria</label>
                            <div class="col-sm-4">
                                <input type="hidden" class="form-control" id="inputPassword3"
name="id_kriteria" placeholder="" value="<?= $kr->id_kriteria; ?>">
                                <input type="text" class="form-control" id="inputPassword3"
name="nama_kriteria" placeholder="Kriteria" value="<?= $kr->nama_kriteria;
?>">
                            </div>
                        </div>
                    </div>
                    <div class="form-group row">
                        <label for="inputPassword3" class="col-sm-2 col-form-
label">Atribut</label>
                        <div class="col-sm-4">

```



```

        <input type="text" class="form-control" id="inputPassword3"
name="atribut" placeholder="Atribut" value="<?= $kr->atribut; ?>">
        </div>
</div>
</div>
<!-- /.card-body -->
        <div class="card-footer">
            <button type="submit" class="btn btn-info">Edit</button>
            <button type="submit" class="btn btn-default float-
right">Cancel</button>
        </div>
<!-- /.card-footer -->
</form>
<?php } ?>
        </div>
<!-- /.card-body -->

<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>
</html>

```

6.5.6 Centeroid

1. Controller

Admedika.php
<pre><?php defined('BASEPATH') OR exit('No direct script access allowed'); class Admedika extends CI_Controller { public function __construct() { parent::__construct(); \$this->load->model('m_admedika'); } public function centeroid() { \$data['centeroid'] = \$this->m_admedika->getCenteroid(); \$this->load->view('admedical/centeroid', \$data); } public function centeroid_add() { \$this->load->view('admedical/centeroid_add'); } public function proses_tambah_centeroid() { \$id_cluster = \$this->input->post('id_cluster'); \$id_kriteria = \$this->input->post('id_kriteria'); \$nilai = \$this->input->post('nilai'); \$data = array('id_cluster' => \$id_cluster, 'id_kriteria' => \$id_kriteria, 'nilai' => \$nilai,); \$this->m_admedika->input_data(\$data,'tb_centeroid'); redirect('admedika/centeroid'); } public function centeroid_edit(\$id) { </pre>

```

        $where = array('id_centeroid' => $id);
        $data['centeroid'] = $this->m_admedika->edit_data($where,'tb_centeroid')-
>result();
        $this->load->view('admedical/centeroid_edit',$data);
    }

    public function proses_edit_centeroid()
    {
        $id = $this->input->post('id_centeroid');
        $id_cluster= $this->input->post('id_cluster');
        $id_kriteria = $this->input->post('id_kriteria');
        $nilai = $this->input->post('nilai');

        $data = array(
            'id_cluster'    => $id_cluster,
            'id_kriteria'    => $id_kriteria,
            'nilai'        => $nilai,
        );

        $where = array(
            'id_centeroid'    => $id
        );

        $this->m_admedika->update_data($where,$data,'tb_centeroid');
        redirect('admedika/centeroid');
    }

    public function centeroid_delete($id)
    {
        $where = array('id_centeroid' => $id);
        $this->m_admedika->hapus_data($where,'tb_centeroid');
        redirect('admedika/centeroid');
    }
}

```

2. Model

M.Admedika.php

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class M_admedika extends CI_Model {
    public function getCenteroid()

```

```

    {
        return $this->db->get('tb_centeroid')->result();
    }
function input_data($data,$table){
    $hasil = $this->db->insert($table,$data);
    return $hasil;
}
function edit_data($where,$table){
    return $this->db->get_where($table,$where);
}

function update_data($where,$data,$table){
    $this->db->where($where);
    $this->db->update($table,$data);
}

function hapus_data($where,$table){
    $this->db->where($where);
    $this->db->delete($table);
}
}

```

3. View

No	Cluster	Kriteria	Nilai	Aksi
1	C1	1	950000	[Edit] [Delete]
2	C1	2	3412828	[Edit] [Delete]
3	C1	3	351000	[Edit] [Delete]
4	C1	4	3412828	[Edit] [Delete]
5	C2	1	2258438	[Edit] [Delete]
6	C2	2	17481817	[Edit] [Delete]
7	C2	3	225826	[Edit] [Delete]
8	C2	4	16745991	[Edit] [Delete]

Gambar 6. 40 Tampilan *Centroid*

centroid.php

```

<?php $this->load->view('admedical/atribut/head');
?>

```

```

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>List Centroid</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item">Centroid</li>
            <li class="breadcrumb-item">List</li>
          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">

    <!-- Default box -->
    <div class="card">
      <div class="card-header">
        <h3 class="card-title">List Centroid</h3>

        <div class="card-tools">
          <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
            <i class="fas fa-minus"></i></button>
          <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
            <i class="fas fa-times"></i></button>
        </div>
      </div>
      <div class="card-body">
        <table class="table table-light table-hover table-bordered" id="example1">
          <thead class="thead-light">
            <tr>
              <th>No</th>
              <th>Cluster</th>
              <th>Kriteria</th>
              <th>Nilai</th>
              <th>Aksi</th>

```

```

        </tr>
    </thead>
    <tbody>
        <?php
        $no = 1;
        foreach($scenteroid as $c) {
            ?>
            <tr>
                <td><?= $no++; ?></td>
                <td><?= $c->id_cluster; ?></td>
                <td><?= $c->id_kriteria; ?></td>
                <td><?= $c->nilai; ?></td>
                <td>
                    <a href="<?= base_url('admedika/centeroid_edit/'.$c-
>id_centeroid);?>" class="btn btn-flat btn-sm btn-info"><i class="fas fa-
edit"></i> </a>
                    <a href="<?= base_url('admedika/centeroid_delete/'.$c-
>id_centeroid);?>" class="btn btn-flat btn-sm btn-danger"><i class="fas fa-
trash"></i> </a>
                </td>
            </tr>
        <?php } ?>
    </tbody>
</table>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->

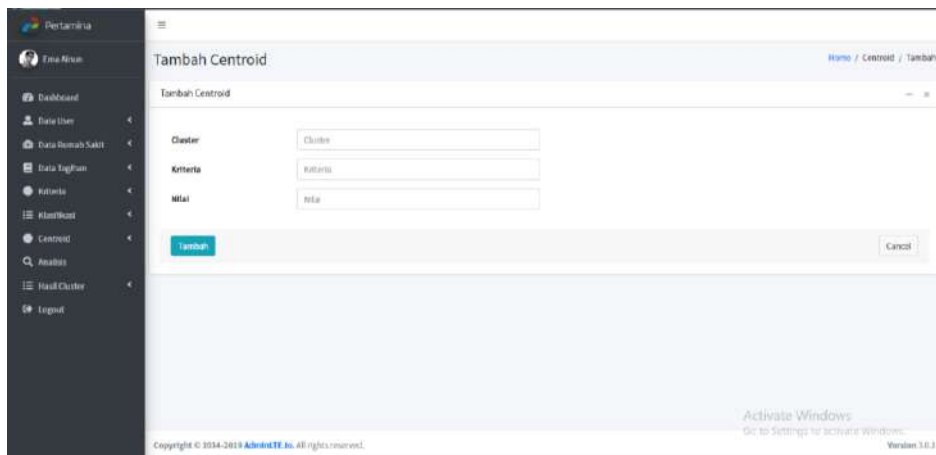
```

```

<aside class="control-sidebar control-sidebar-dark">
  <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- /.wrapper -->

<?php $this->load->view('js');?>
</body>
</html>

```



Gambar 6. 41 Tampilan Tambah *Centeroid*

centroid_add.php

```

<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>Tambah Centroid</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item">Centroid</li>
            <li class="breadcrumb-item">Tambah</li>

```

```

        </ol>
    </div>
</div>
</div><!-- /.container-fluid -->
    </section>

<!-- Main content -->
    <section class="content">

<!-- Default box -->
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">Tambah Centroid</h3>

            <div class="card-tools">
                <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
                    <i class="fas fa-minus"></i></button>
                <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
                    <i class="fas fa-times"></i></button>
            </div>
        </div>
        <div class="card-body">
            <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_tambah_centeroid" method="POST">
                <div class="card-body">

                    <div class="form-group row">
                        <label for="inputPassword3" class="col-sm-2 col-form-
label">Cluster</label>
                        <div class="col-sm-4">
                            <input type="text" class="form-control" id="inputPassword3"
name="id_cluster" placeholder="Cluster">
                        </div>
                    </div>

                    <div class="form-group row">
                        <label for="inputPassword3" class="col-sm-2 col-form-
label">Kriteria</label>
                        <div class="col-sm-4">
                            <input type="text" class="form-control" id="inputPassword3"
name="id_kriteria" placeholder="Kriteria">
                        </div>
                    </div>
                <div class="form-group row">

```



```

        <label for="inputPassword3" class="col-sm-2 col-form-
label">Nilai</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="inputPassword3"
name="nilai" placeholder="Nilai">
        </div>
    </div>

    </div>
<!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Tambah</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
<!-- /.card-footer -->
</form>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
    </div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

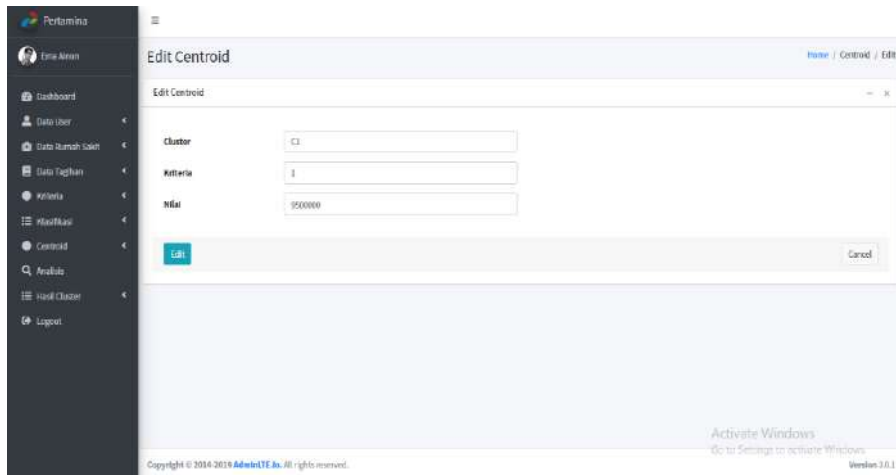
<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- /.wrapper -->

```

```
<?php $this->load->view('js');?>
</body>

</html>
```



Gambar 6. 42 Tampilan *Edit Centroid*

centroid_edit.php

```
<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
    <!-- Content Header (Page header) -->
    <section class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1>Edit Centroid</h1>
                </div>
                <div class="col-sm-6">
                    <ol class="breadcrumb float-sm-right">
                        <li class="breadcrumb-item"><a href="#">Home</a></li>
                        <li class="breadcrumb-item">Centroid</li>
                        <li class="breadcrumb-item">Edit</li>
                    </ol>
                </div>
            </div>
        </div>
    </div>
<!-- /.container-fluid -->
</section>
```

```

<!-- Main content -->
    <section class="content">

<!-- Default box -->
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">Edit Centroid</h3>

            <div class="card-tools">
                <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
                    <i class="fas fa-minus"></i></button>
                <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
                    <i class="fas fa-times"></i></button>
            </div>
        </div>
        <div class="card-body">
            <?php foreach ($centeroid as $c) { ?>
                <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_edit_centeroid" method="POST">
                    <div class="card-body">

                        <div class="form-group row">
                            <label for="inputPassword3" class="col-sm-2 col-form-
label">Cluster</label>
                            <div class="col-sm-4">
                                <input type="hidden" class="form-control" id="inputPassword3"
name="id_centeroid" placeholder="centeroid" value="<?= $c->id_centeroid;
?>">
                                <input type="text" class="form-control" id="inputPassword3"
name="id_cluster" placeholder="centeroid" value="<?= $c->id_cluster; ?>">
                            </div>
                        </div>
                        <div class="form-group row">
                            <label for="inputPassword3" class="col-sm-2 col-form-
label">Kriteria</label>
                            <div class="col-sm-4">
                                <input type="text" class="form-control" id="inputPassword3"
name="id_kriteria" placeholder="id_kriteria" value="<?= $c->id_kriteria; ?>">
                            </div>
                        </div>
                        <div class="form-group row">
                            <label for="inputPassword3" class="col-sm-2 col-form-
label">Nilai</label>

```

```

        <div class="col-sm-4">
            <input type="text" class="form-control" id="inputPassword3"
name="nilai" placeholder="nilai" value="<?= $c->nilai; ?>">
        </div>
    </div>
</div>
<!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Edit</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
<!-- /.card-footer -->
</form>
<?php } ?>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
    </div>
<!-- /.card -->

</section>
    <!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->
<?php $this->load->view('js');?>
</body>
</html>

```

6.5.7 Klasifikasi

1. Controller

Admedika.php

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Admedika extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_admedika');
    }

    public function klasifikasi()
    {
        $data['klasifikasi'] = $this->m_admedika->getKlasifikasi();
        $this->load->view('admedical/klasifikasi', $data);
    }

    public function klasifikasi_add()
    {
        $data['tagihan'] = $this->m_admedika->get_id_tagihan();
        $this->load->view('admedical/klasifikasi_add', $data);
    }

    public function proses_tambah_klasifikasi()
    {
        $id_tagihan = $this->input->post('id_tagihan');
        $id_kriteria = $this->input->post('id_kriteria');
        $nilai = $this->input->post('nilai');

        $data = array();
        $totaldata = $this->input->post('id_kriteria');
        for ($i=0; $i < count($totaldata) ; $i++)
        {
            array_push($data, array(
                'id_tagihan' => $id_tagihan[$i],
                'id_kriteria' => $id_kriteria[$i],
                'nilai' => $nilai[$i],
            ));
        }
        $this->m_admedika->input_data_batch($data,'tb_klasifikasi');
        redirect(base_url('admedika/klasifikasi'));
```

```

    }

    public function klasifikasi_edit($id)
    {
        $where = array('id_klasifikasi' => $id);
        $data['klasifikasi'] = $this->m_admedika-
>edit_data($where,'tb_klasifikasi')->result();
        $this->load->view('admedical/klasifikasi_edit',$data);
    }

    public function proses_edit_klasifikasi()
    {
        $id = $this->input->post('id_klasifikasi');
        $id_tagihan = $this->input->post('id_tagihan');
        $id_kriteria = $this->input->post('id_kriteria');
        $nilai = $this->input->post('nilai');

        $data = array(
            'id_tagihan'    => $id_tagihan,
            'id_kriteria'   => $id_kriteria,
            'nilai'         => $nilai,
        );

        $where = array(
            'id_klasifikasi' => $id
        );

        $this->m_admedika->update_data($where,$data,'tb_klasifikasi');
        redirect('admedika/klasifikasi');
    }

    public function klasifikasi_delete($id)
    {
        $where = array('id_klasifikasi' => $id);
        $this->m_admedika->hapus_data($where,'tb_klasifikasi');
        redirect('admedika/klasifikasi');
    }
}

```

2. Model

M_Admedika.php

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class M_admedika extends CI_Model {

    public function getKlasifikasi()
    {
        return $this->db->get('tb_klasifikasi')->result();
    }
    function input_data($data,$table){
        $hasil = $this->db->insert($table,$data);
        return $hasil;
    }
    function edit_data($where,$table){
        return $this->db->get_where($table,$where);
    }

    function update_data($where,$data,$table){
        $this->db->where($where);
        $this->db->update($table,$data);
    }
    function hapus_data($where,$table){
        $this->db->where($where);
        $this->db->delete($table);
    }
}
```

3. View

No	No Tagihan	Kriteria	Nilai	Aksi
1	65	1	4096768	[Edit] [Delete]
2	65	2	9012917	[Edit] [Delete]
2	65	3	1409	[Edit] [Delete]
4	65	4	9011088	[Edit] [Delete]
5	66	1	4096768	[Edit] [Delete]
6	66	2	4036176	[Edit] [Delete]
7	66	3	1637520	[Edit] [Delete]
8	66	4	1072458	[Edit] [Delete]
9	67	1	4096768	[Edit] [Delete]
10	67	2	21278571	[Edit] [Delete]

Gambar 6. 43 Tampilan Klasifikasi

```

klasifikasi.php

<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>List Klasifikasi</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item">Klasifikasi</li>
            <li class="breadcrumb-item">List</li>
          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">

    <!-- Default box -->

```



```

<div class="card">
  <div class="card-header">
    <h3 class="card-title">List Klasifikasi</h3>

    <div class="card-tools">
      <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
        <i class="fas fa-minus"></i></button>
      <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
        <i class="fas fa-times"></i></button>
    </div>
  </div>
  <div class="card-body">
    <table class="table table-light table-hover table-bordered" id="example1">
      <thead class="thead-light">
        <tr>
          <th>No</th>
          <th>No Tagihan</th>
          <th>Kriteria</th>
          <th>Nilai</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody>
        <?php
          $no = 1;
          foreach($klasifikasi as $k) {
            ?>
            <tr>
              <td><?= $no++; ?></td>
              <td><?= $k->id_tagihan; ?></td>
              <td><?= $k->id_kriteria; ?></td>
              <td><?= $k->nilai; ?></td>
              <td>
                <a href="<?= base_url('admedika/klasifikasi_edit/'.$k-
>id_klasifikasi);?" class="btn btn-flat btn-sm btn-info"><i class="fas fa-
edit"></i> </a>
                <a href="<?= base_url('admedika/klasifikasi_delete/'.$k-
>id_klasifikasi);?" class="btn btn-flat btn-sm btn-danger"><i class="fas fa-
trash"></i> </a>
              </td>
            </tr>
          <?php } ?>
        </tbody>
      </table>

```

```

        </div>
        <!-- /.card-body -->

        <!-- /.card-footer-->
    </div>
    <!-- /.card -->

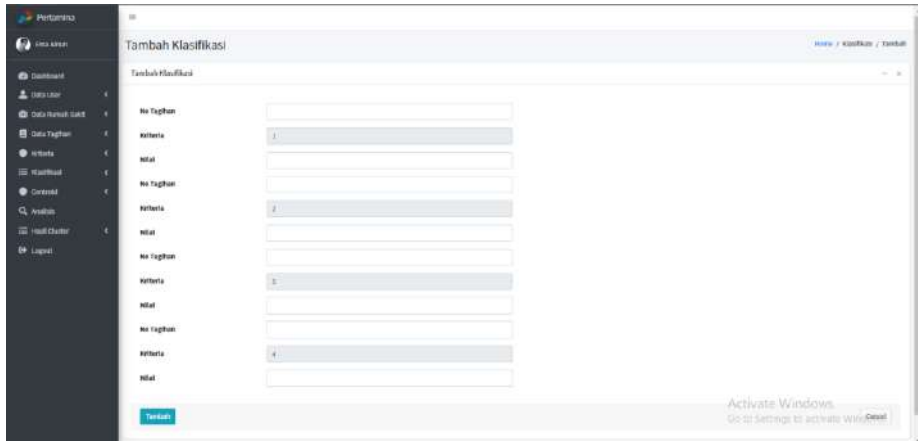
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
    reserved.
</footer>

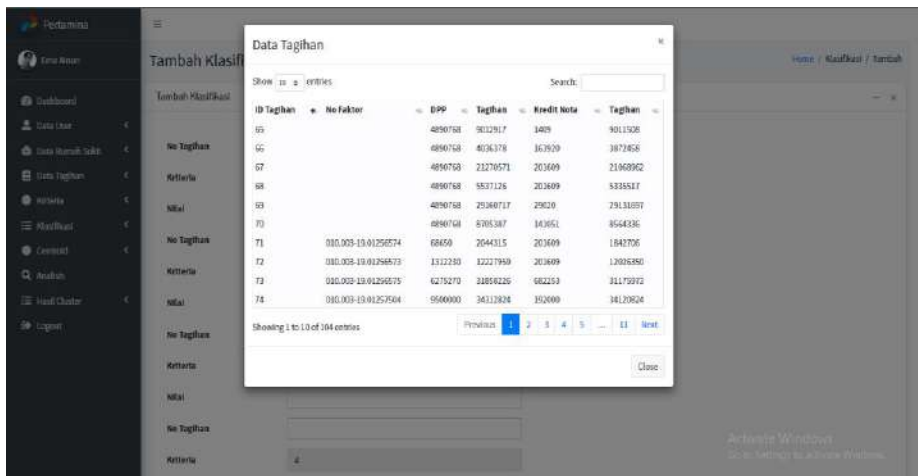
<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>
</html>

```



Gambar 6. 44 Tampilan Tambah Klasifikasi



Gambar 6. 45 Tampilan Tambah Klasifikasi

klasifikasi_add.php
<pre> <?php \$this->load->view('admedical/atribut/head'); ?> <!-- Content Wrapper. Contains page content --> <div class="content-wrapper"> <!-- Content Header (Page header) --> <section class="content-header"> <div class="container-fluid"> <div class="row mb-2"> <div class="col-sm-6"> <h1>Tambah Klasifikasi</h1> </div> <div class="col-sm-6"> </pre>

```

        <ol class="breadcrumb float-sm-right">
        <li class="breadcrumb-item"><a href="#">Home</a></li>
        <li class="breadcrumb-item">Klasifikasi</li>
        <li class="breadcrumb-item">Tambah</li>
        </ol>
    </div>
</div>
<!-- /.container-fluid -->
</section>

<!-- Main content -->
<section class="content">

<!-- Default box -->
    <div class="card">
        <div class="card-header">
            <h3 class="card-title">Tambah Klasifikasi</h3>

            <div class="card-tools">
                <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
                    <i class="fas fa-minus"></i></button>
                <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
                    <i class="fas fa-times"></i></button>
            </div>
        </div>
        <div class="card-body">
            <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_tambah_klasifikasi" method="POST">
                <div class="card-body">

                    <div class="form-group row">
                        <label for="inputPassword3" class="col-sm-2 col-form-label">No
Tagihan</label>
                        <div class="col-sm-4">
                            <input type="text" class="form-control" id="inputPassword3"
name="id_tagihan[]" placeholder="">
                        </div>
                    </div>

                    <div class="form-group row">
                        <label for="inputPassword3" class="col-sm-2 col-form-
label">Kriteria</label>
                        <div class="col-sm-4">

```

```

        <input type="text" class="form-control" readonly name="id_kriteria[]"
id="formGroupExampleInput" placeholder="" value="1">
    </div>
</div>
    <div class="form-group row">
        <label for="dpp" class="col-sm-2 col-form-label">Nilai</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="dpp" name="nilai[]"
placeholder="">
        </div>
    </div>
<!-- <hr align="right" color="black"> -->
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">No
Tagihan</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="inputPassword3"
name="id_tagihan[]" placeholder="">
        </div>
    </div>
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-
label">Kriteria</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" readonly name="id_kriteria[]"
id="formGroupExampleInput" placeholder="" value="2">
        </div>
    </div>
    <div class="form-group row">
        <label for="tagihan" class="col-sm-2 col-form-label">Nilai</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="tagihan" name="nilai[]"
placeholder="">
        </div>
    </div>
<!-- <hr align="right" color="black"> -->
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">No
Tagihan</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="inputPassword3"
name="id_tagihan[]" placeholder="">
        </div>
    </div>
    <div class="form-group row">

```

```

        <label for="inputPassword3" class="col-sm-2 col-form-
label">Kriteria</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" readonly name="id_kriteria[]"
id="formGroupExampleInput" placeholder="" value="3">
        </div>
    </div>
    <div class="form-group row">
        <label for="credit" class="col-sm-2 col-form-label">Nilai</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="credit" name="nilai[]"
placeholder="">
        </div>
    </div>
    <!-- <hr align="right" color="black"> -->
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">No
Tagihan</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="inputPassword3"
name="id_tagihan[]" placeholder="">
        </div>
    </div>
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-
label">Kriteria</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" readonly name="id_kriteria[]"
id="formGroupExampleInput" placeholder="" value="4">
        </div>
    </div>
    <div class="form-group row">
        <label for="hsl" class="col-sm-2 col-form-label">Nilai</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="hsl" name="nilai[]"
placeholder="">
        </div>
    </div>
</div>
<!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Tambah</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>

```

```

<!-- /.card-footer -->
    </form>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
    </div>
<!-- /.card -->
<!-- Large modal -->

<div class="modal fade bd-example-modal-lg" tabindex="-1" role="dialog" aria-
labelledby="myLargeModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <h4 class="modal-title">Data Tagihan</h4>
                <button type="button" class="close" data-
dismiss="modal">&times;</button>
            </div>
            <div class="modal-body">
                <table class="table table-light table-hover table-bordered table-
sm" id="example1">
                    <thead>
                        <tr>
                            <th>ID Tagihan</th>
                            <th>No Faktur</th>
                            <th>DPP</th>
                            <th>Tagihan</th>
                            <th>Kredit Nota</th>
                            <th>Tagihan</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php foreach ($tagihan as $tg) { ?>
                            <?php
                                $a = $tg->tagihan;
                                $b = $tg->credit_nota;
                                $hsl = $a - $b;
                                ?>
                            <tr id="data" onClick="masuk(this,'<?=$tg->id_tagihan ?>','<?=$tg->dpp
?>','<?=$tg->tagihan ?>','<?=$tg->credit_nota ?>','<?=$hsl ?>')">
                                <td><?=$tg->id_tagihan ?></td>
                                <td><?=$tg->no_faktur ?></td>
                                <td><?=$tg->dpp ?></td>
                                <td><?=$tg->tagihan ?></td>

```

```

<td><?= $tg->credit_nota ?></td>
<td><?= $hsl ?></td>
</tr>
    <?php } ?>
</tbody>
</table>
</div>
    <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
    </div>
</div>
</div>
</div>
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- /.wrapper -->

<?php $this->load->view('js');?>
<script>
    document.onkeydown = function(e)
    {
        switch (e.keyCode) {
            // f1
            case 113:
                $(".bd-example-modal-lg").modal('show');
                break;

```



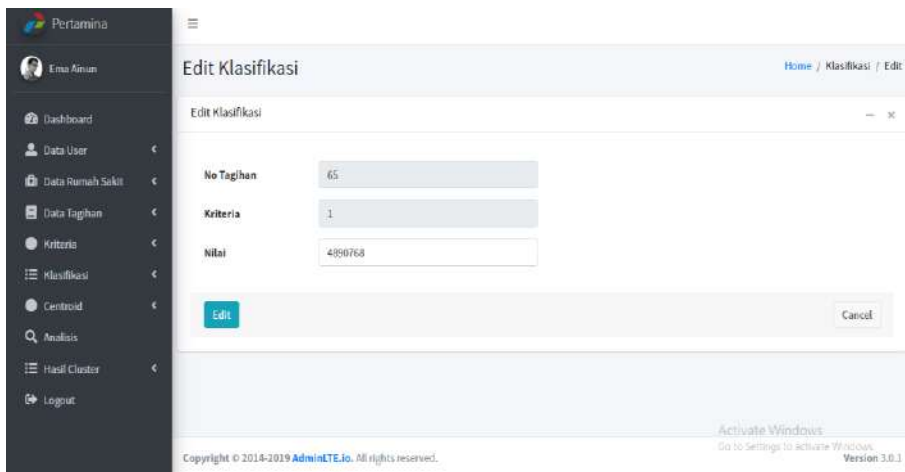
```

        // esc
        case 27:
            x
            $(''.bd-example-modal-lg').modal('hide');
            break;
        }
    };

function masuk(txt, id_tagihan,dpp,tagihan,credit_nota,hsl)
{
    $('[name="id_tagihan[]"]').val(id_tagihan);
    $('#dpp').val(dpp);
    $('#tagihan').val(tagihan);
    $('#credit').val(credit_nota);
    $('#hsl').val(hsl);
    $(''.bd-example-modal-lg').modal('hide');
}
</script>
</body>

</html>

```



Gambar 6. 46 Tampilan *Edit Klasifikasi*

Klasifikasi_edit.php
<pre> <?php \$this->load->view('admedical/atribut/head'); ?> <!-- Content Wrapper. Contains page content --> <div class="content-wrapper"> <!-- Content Header (Page header) --> </pre>

```

        <section class="content-header">
            <div class="container-fluid">
                <div class="row mb-2">
                    <div class="col-sm-6">
                        <h1>Edit Klasifikasi</h1>
                    </div>
                    <div class="col-sm-6">
                        <ol class="breadcrumb float-sm-right">
                            <li class="breadcrumb-item"><a href="#">Home</a></li>
                            <li class="breadcrumb-item">Klasifikasi</li>
                            <li class="breadcrumb-item">Edit</li>
                        </ol>
                    </div>
                </div>
            </div><!-- /.container-fluid -->
        </section>

<!-- Main content -->
        <section class="content">

<!-- Default box -->
            <div class="card">
                <div class="card-header">
                    <h3 class="card-title">Edit Klasifikasi</h3>

                    <div class="card-tools">
                        <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
                            <i class="fas fa-minus"></i></button>
                        <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
                            <i class="fas fa-times"></i></button>
                    </div>
                </div>
                <div class="card-body">
                    <?php foreach ($klasifikasi as $k) { ?>
                        <form class="form-horizontal" action="<?=
base_url();?>admedika/proses_edit_klasifikasi" method="POST">
                            <div class="card-body">

                                <div class="form-group row">
                                    <label for="inputPassword3" class="col-sm-2 col-form-label">No
Tagihan</label>
                                    <div class="col-sm-4">

```

```

        <input type="hidden" class="form-control" id="inputPassword3"
name="id_klasifikasi" placeholder="klasifikasi" value="<?=$k->id_klasifikasi;
?>">
        <input type="text" class="form-control" readonly id="inputPassword3"
name="id_tagihan" placeholder="id_tagihan" value="<?=$k->id_tagihan; ?>">
    </div>
</div>
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-
label">Kriteria</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" readonly id="inputPassword3"
name="id_kriteria" placeholder="id_kriteria" value="<?=$k->id_kriteria; ?>">
        </div>
    </div>
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-
label">Nilai</label>
        <div class="col-sm-4">
            <input type="text" class="form-control" id="inputPassword3"
name="nilai" placeholder="nilai" value="<?=$k->nilai; ?>">
        </div>
    </div>
</div>
<!-- /.card-body -->
    <div class="card-footer">
        <button type="submit" class="btn btn-info">Edit</button>
        <button type="submit" class="btn btn-default float-
right">Cancel</button>
    </div>
<!-- /.card-footer -->
</form>
    <?php } ?>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
    </div>
<!-- /.card -->

    </section>
    <!-- /.content -->
</div>
<!-- /.content-wrapper -->

```

```

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
    reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>

</html>

```

6.5.8 Perhitungan K-Means

1. Controller

Admedika.php

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Admedika extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_admedika');
    }
    public function kmeans()
    {
        $this->load->view('admedical/kmeans');
    }
}

```

2. View

Permina

Ena Aluan

K-Means

Home / K-Means / List

Analisa Menggunakan Data Mining Metode K-Means

Data, Kriteria dan Nilai Yang Akan Dihitung Keanggotaannya dalam Cluster

No Debit Nota	Kriteria			
	DPD	Tagihan	Credit Nota	Pembayaran
148803/1901006	4890768	9012917	1409	9011568
148803/1903008	4890768	4036378	163929	3872458
148803/1903009	4890768	21270571	201609	21068962
148806/1902013	4890768	5537126	201609	5335517
148806/1901005	4890768	29140717	29020	29131697
148806/1901008	4890768	8795387	141051	8564326
148801/1901028	68650	2044315	201609	1842706
148801/1903027	1312230	12227959	201609	12026350
148801/1903029	6275270	31858236	682253	31135973
148.801/1904020	9500000	34112824	192000	34120824
148.801/1907016	4890768	28196469	201609	28194860
148.862/2905003	4890768	4460102	201609	4258493
148.864/2902007	7297650	23262307	201609	33860698
NO2DA1905005	966711	10742911	170000	10572911
NO2DA1905008	984711	8119258	15248	8114010

Activate Windows
Go to Settings to activate Windows.

Gambar 6. 47 Perhitungan K-Means

Jumlah Cluster Diciari = 2

Maksimum Iterasi = 100

Iterasi Ke = 1

Pusat Cluster

No Debit Nota	Kriteria			
	DPD	Tagihan	Credit Nota	Pembayaran
C1	9560100	34112824	150000	34120824
C2	2358434	17481817	735826	16745591

Jarak Terhadap Pusat Cluster

No Debit Nota	Cluster	
	C1	C2
148803/1903006	35942236.835729	11768456.392114
148803/1903008	43045007.09928	18794878.02393
148803/1903009	19018327.138027	6304033.5048286
148806/1902013	40961911.18536	16720426.528076
148806/1903005	8525850.348877	17225408.69611
148806/1903008	36470844.849462	12277282.878231
148801/1903028	4605422.298921	21585941.496137
148801/1903027	32294685.527354	7159466.8562505
148801/1903029	8038630.6029381	20742468.288448
148.801/2904020	0	25225264.597543

Activate Windows
Go to Settings to activate Windows.

Gambar 6. 48 Perhitungan K-Means

Keanggotaan Cluster dan Jarak Minimum			
No Data Nota	Keanggotaan	Min Jarak	Waktu Min. Jarak
148803/1903006	C2	11768455.32114	1.384065793047E+14
148803/1903008	C2	18794878.92392	3.512474737624E+14
148803/1903009	C2	6304933.504286	3874083843002
148806/1902013	C2	16720426.528076	2.795726632808E+14
148806/1903005	C1	8526850.348857	72707176871802
148806/1903008	C2	12277282.878231	1.5873167487231E+14
148801/1903828	C2	21585941.498137	4.6395287036097E+14
148801/1903007	C2	7159466.8562505	5.1257965663750
148801/1903009	C1	9635830.8029383	25336430330714
148.801/1904020	C1	0	0
148.801/1907016	C1	9558525.0207697	91305417778026
148.802/1905003	C2	18226375.695296	3.3220077113187E+14
148.806/1902007	C1	2660397.8538633	7077734772146
NO20N1905065	C2	9258890.0364083	85727044706241
NO20N1905068	C2	12829242.614253	1.6458946610765E+14
NO20N1905046	C2	15036515.356605	2.2609579407321E+14
NO20N1.1904137	C2	6212132.0797754	38590584976575
NO20N1.1904139	C2	19516317.79623	3.8688744957658E+14
NO20N1.1904138	C2	6212132.0797754	38590584976575
NO20N1903378	C2	14098748.529654	2.5402062293657E+14
NO20N1.1904141	C2	7217929.4106118	52096564076575
NO20N1.1904140	C2	29696135.126307	3.8793773867737E+14
NO20N1/1905234	C2	24070692.860833	5.7839825480535E+14

Gambar 6. 49 Perhitungan K-Means

Hasil Didapatkan, Rasio Sudah Tidak Lebih Besar dari Rasio Sebelumnya	
Keanggotaan Cluster Akhir:	
Debit Nota	Keanggotaan
148803/1903006	TIDAK MENDESAR
148803/1903008	TIDAK MENDESAR
148803/1903009	TIDAK MENDESAR
148806/1902013	TIDAK MENDESAR
148806/1903005	MENDESAR
148806/1903008	TIDAK MENDESAR
148801/1903828	TIDAK MENDESAR
148801/1903007	TIDAK MENDESAR
148801/1903009	MENDESAR
148.801/1904020	MENDESAR
148.801/1907016	MENDESAR
148.802/1905003	TIDAK MENDESAR
148.806/1902007	MENDESAR
NO20N1905065	TIDAK MENDESAR
NO20N1905068	TIDAK MENDESAR
NO20N1905046	TIDAK MENDESAR
NO20N1.1904137	TIDAK MENDESAR
NO20N1.1904139	TIDAK MENDESAR
NO20N1.1904138	TIDAK MENDESAR
NO20N1903378	TIDAK MENDESAR
NO20N1.1904141	TIDAK MENDESAR
NO20N1.1904140	TIDAK MENDESAR

Gambar 6. 50 Perhitungan K-Means

```

K-Means.php
<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>K-Means</h1>
        </div>

```

```

<div class="col-sm-6">
  <ol class="breadcrumb float-sm-right">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item">K-Means</li>
    <li class="breadcrumb-item">List</li>
  </ol>
</div>
</div>
</div>
<!-- /.container-fluid -->
</section>

<!-- Main content -->
<section class="content">

  <!-- Default box -->
  <div class="card">
    <div class="card-body">
      <table width="60%" border="0" align="center" cellpadding="0"
        cellspacing="1" bgcolor="#000099">
        <tr>
          <td align="center" valign="top" bgcolor="#FFFFFF"><br>
            <strong>Analisa Menggunakan Data Mining Metode K-
Means</strong><br>
            <br>
            <?php
              $tagihan = [];
              $querytagihan = $this->db->get('tagihan')->result_array();
              $i=0;
              foreach ($querytagihan as $datatagihan)
              {
                $tagihan[$i] = $datatagihan['no_debit_nota'];
                $i++;
              }

              ///

              $kriteria = [];
              $querykriteria = $this->db->get('tb_kriteria')->result_array();
              $i=0;
              foreach ($querykriteria as $datakriteria)
              {
                $kriteria[$i] = $datakriteria['nama_kriteria'];
                $i++;
              }

```

```

// print_r($kriteria);
////

$alternatifkriteria = [];
$querytagihan = $this->db->get('tagihan')->result_array();
$i=0;
foreach ($querytagihan as $datatagihan)
{
    $querykriteria = $this->db->get('tb_kriteria')->result_array();
    $j=0;
    foreach ($querykriteria as $datakriteria)
    {
        // $queryalternatifkriteria = $this->db->get_where('klasifikasi',
['npp' => $datapensiun['npp'], 'id_kriteria' => $datakriteria['id']])->row();
        // $alternatifkriteria[$i][$j] = $queryalternatifkriteria->nilai;
        $this->db->select('*');
        $this->db->from('tb_klasifikasi');
        $this->db->where('id_tagihan', $datatagihan['id_tagihan']);
        $this->db->where('id_kriteria', $datakriteria['id_kriteria']);
        $queryalternatifkriteria = $this->db->get()->row();
        // $queryalternatifkriteria = mysql_query("SELECT * FROM
klasifikasi WHERE nis = '$dataalternatif[nis]' AND id_kriteria =
'$datakriteria[id_kriteria]'");
        $dataalternatifkriteria = $queryalternatifkriteria;

        $alternatifkriteria[$i][$j] = $dataalternatifkriteria->nilai;
        // print_r($queryalternatifkriteria);
        // $dataalternatifkriteria = $queryalternatifkriteria-
>result_array();
        $j++;
    }
    $i++;
}

echo "<br/>";
echo "Data, Kriteria dan Nilai";
echo "<br/>";
echo "Yang Akan Dihitung Keanggotaannya dalam Cluster";
echo "<br/>";
echo "<br/>";
echo "<table width=\"700\" border=\"1\" cellpadding=\"1\"
cellpadding=\"3\" >";
echo "<tr>";
echo "<td bgcolor=\"#FFFFFF\" rowspan=\"2\">No Debit
Nota</td>";

```



```

        echo "<td bgcolor=\"#FFFFFF\" colspan=\"".count($kriteria)."\"
align=\"center\">Kriteria</td>";
        echo "</tr>";
        echo "<tr>";
        for ($i=0;$i<count($kriteria);$i++) {
            echo "<td bgcolor=\"#FFFFFF\" >".$kriteria[$i]."</td>";
        }
        echo "</tr>";

        for ($i=0;$i<count($tagihan);$i++) {
            echo "<tr>";
            echo "<td bgcolor=\"#FFFFFF\" >".$tagihan[$i]."</td>";
            for ($j=0;$j<count($kriteria);$j++) {
                echo "<td bgcolor=\"#FFFFFF\"
>".$alternatifkriteria[$i][$j]."</td>";
            }
            echo "</tr>";

        }// print_r($cluster);
        echo "</table>";
    ?>

```

```

<div id="perhitungan" style="display:none;">

```

```

<?php
error_reporting(0);
$jml_cluster = $this->db->get('tb_cluster')->num_rows();

$pusat_cluster= [];
$rasio_sebelum = 0;
$ketemu = false;
$max_iterasi = 100;

echo "<br/>";
echo "Jumlah Cluster Dicari = ".$jml_cluster;
echo "<br/>";

echo "<br/>";
echo "Maksimum Iterasi = ".$max_iterasi;
echo "<br/>";

for ($z=0; $z < $max_iterasi ; $z++)
{

    echo "<br/>";
    echo "Iterasi Ke = ".$(z+1)."<br/>";
}

```

```

echo "=====<br/>";
echo "<br/>";

if ($z == 0)
{
    //ambil data centroid awal
    $i=0;
    $this->db->select('*');
    $this->db->from('tb_cluster');
    $this->db->order_by('id_cluster', 'ASC');
    $cluster = $this->db->get()->result_array();
    // mysql_query("SELECT * FROM cluster ORDER BY id_cluster
ASC");

    foreach ($cluster as $dt_cluster)
    {
        //ambil nilai dari tabel centroid
        $this->db->select('*');
        $this->db->from('tb_centeroid');
        $this->db->where('id_cluster', $dt_cluster['id_cluster']);
        $this->db->order_by('id_centeroid', 'ASC');
        $centroid = $this->db->get()->result_array();
        // mysql_query("SELECT * FROM centroid WHERE
id_cluster='$dt_cluster[id_cluster]' ORDER BY id_centeroid ASC");

        $j=0;
        foreach ($centroid as $dt_centroid)
        {
            $pusat_cluster[$i][$j] = $dt_centroid["nilai"];
            $j++;
        }
        $i++;
    }
}
else
{
    for ($i=0; $i < $jml_cluster; $i++) {
        for ($j=0; $j < count($skriteria); $j++) {
            $pusat_cluster[$i][$j] = $pusat_cluster_baru[$i][$j];
        }
    }
}
echo "<br/>";
echo "Pusat Cluster";
echo "<br/>";
echo "<br/>";

```

```

        echo "<table width=\"700\" border=\"1\" cellpadding=\"1\"
cellpadding=\"3\" >";
        echo "<tr>";
        echo "<td bgcolor=\"#FFFFFF\" rowspan=\"2\">No Debit
Nota</td>";
        echo "<td bgcolor=\"#FFFFFF\" colspan=\"".count($kriteria).\"\"
align=\"center\">Kriteria</td>";
        echo "</tr>";
        echo "<tr>";
        for ($i=0; $i < count($kriteria); $i++) {
            echo "<td bgcolor=\"#FFFFFF\" >\". $kriteria[$i].\"</td>";
        }
        echo "</tr>";

        for ($i=0; $i < $jml_cluster; $i++) {
            echo "<tr>";
            echo "<td bgcolor=\"#FFFFFF\" >\".\"C\".($i+1).\"</td>";
            for ($j=0; $j < count($kriteria); $j++) {
                echo "<td bgcolor=\"#FFFFFF\"
>\".$pusat_cluster[$i][$j].\"</td>";
            }
            echo "</tr>";
        }
        echo "</table>";

        echo "<br/>";
        echo "Jarak Terhadap Pusat Cluster";
        echo "<br/>";
        echo "<br/>";
        echo "<br/>";
        echo "<table width=\"700\" border=\"1\" cellpadding=\"1\"
cellpadding=\"3\" >";
        echo "<tr>";
        echo "<td bgcolor=\"#FFFFFF\" rowspan=\"2\">No Debit
Nota</td>";
        echo "<td bgcolor=\"#FFFFFF\" colspan=\"". $jml_cluster.\"\"
align=\"center\">Cluster</td>";
        echo "</tr>";
        echo "<tr>";
        for ($i=0; $i < $jml_cluster; $i++) {
            echo "<td bgcolor=\"#FFFFFF\" >\".\"C\".($i+1).\"</td>";
        }
        echo "</tr>";

        $jarak = array();
        $keanggotaan = array();
        $min_jarak = array();

```

```

$kuadrat_min_jarak = array();

$wcv = 0;
for ($i=0; $i < count($tagihan); $i++) {
    echo "<tr>";
    echo "<td bgcolor=\"#FFFFFF\" >".$tagihan[$i]."</td>";
    $min_jarak[$i] = 0;
    $kuadrat_min_jarak[$i] = 0;
    $keanggotaan[$i] = "";
    for ($j=0; $j < $jml_cluster; $j++) {
        $jarak[$i][$j] = 0;
        for ($k=0; $k < count($kriteria); $k++) {
            $jarak[$i][$j] = $jarak[$i][$j] + (($alternatifkriteria[$i][$k] -
$pusat_cluster[$j][$k]) * ($alternatifkriteria[$i][$k] - $pusat_cluster[$j][$k]));
        }
        $jarak[$i][$j] = sqrt($jarak[$i][$j]);
        echo "<td bgcolor=\"#FFFFFF\" >".$jarak[$i][$j]."</td>";
        if ($j==0) {
            $min_jarak[$i] = $jarak[$i][$j];
            $keanggotaan[$i] = "C".($j+1);
            $kuadrat_min_jarak[$i] = $min_jarak[$i] * $min_jarak[$i];
        }
        else if ($min_jarak[$i] >= $jarak[$i][$j]) {
            $min_jarak[$i] = $jarak[$i][$j];
            $keanggotaan[$i] = "C".($j+1);
            $kuadrat_min_jarak[$i] = $min_jarak[$i] * $min_jarak[$i];
        }
    }
    $wcv = $wcv + $kuadrat_min_jarak[$i];
    echo "</tr>";
}
echo "</table>";
echo "<br>";
echo "Keanggotaan Cluster dan Jarak Minimum";
echo "<br>";
echo "<br>";
echo "<table width=\"700\" border=\"1\" cellspacing=\"1\"
cellpadding=\"3\" >";
echo "<tr>";
echo "<td bgcolor=\"#FFFFFF\">No Debit Nota</td>";
echo "<td bgcolor=\"#FFFFFF\">Keanggotaan</td>";
echo "<td bgcolor=\"#FFFFFF\">Min Jarak</td>";
echo "<td bgcolor=\"#FFFFFF\">Kuadrat Min Jarak</td>";
echo "</tr>";

for ($i=0; $i < count($tagihan); $i++) {

```

```

        echo "<tr>";
        echo "<td bgcolor=\"#FFFFFF\" >".$tagihan[$i]."</td>";
        echo "<td bgcolor=\"#FFFFFF\" >".$keanggotaan[$i]."</td>";
        echo "<td bgcolor=\"#FFFFFF\" >".$min_jarak[$i]."</td>";
        echo "<td bgcolor=\"#FFFFFF\"
>".$kuadrat_min_jarak[$i]."</td>";
        echo "</tr>";
    }
    echo "</table>";
    echo "<br/>";
    echo "<br/>";

    $pusat_cluster_baru = array();
    $jml_keanggotaan = array();

    echo "<br/>";
    echo "Pusat Cluster Baru";
    echo "<br/>";
    echo "<br/>";
    echo "<table width=\"700\" border=\"1\" cellpadding=\"3\" cellspacing=\"1\"
cellpadding=\"3\" >";
    echo "<tr>";
    echo "<td bgcolor=\"#FFFFFF\" rowspan=\"2\">No Debit
Nota</td>";
    for ($i=0; $i < $jml_cluster; $i++) {
        echo "<td bgcolor=\"#FFFFFF\" colspan=\"".count($kriteria)."\n
align=\"center\">\"C\".($i+1)."</td>";
    }
    echo "</tr>";
    echo "<tr>";
    for ($i=0; $i < $jml_cluster; $i++) {
        for ($j=0; $j < count($kriteria); $j++) {
            $pusat_cluster_baru[$i][$j] = 0;
            $jml_keanggotaan[$i][$j] = 0;
            echo "<td bgcolor=\"#FFFFFF\" >".$kriteria[$j]."</td>";
        }
    }
    echo "</tr>";

    for ($i=0; $i < count($tagihan); $i++) {
        echo "<tr>";
        echo "<td bgcolor=\"#FFFFFF\" >".$tagihan[$i]."</td>";
        for ($j=0; $j < $jml_cluster; $j++) {
            for ($k=0; $k < count($kriteria); $k++) {
                if ($keanggotaan[$i] == "C".($j+1)) {

```

```

echo "<td bgcolor=\"#FFFFFF\"
>\".$alternatifkriteria[$i][$k].\"</td>";
        $pusat_cluster_baru[$j][$k] = $pusat_cluster_baru[$j][$k]
+ $alternatifkriteria[$i][$k];
        $jml_keanggotaan[$j][$k] = $jml_keanggotaan[$j][$k] +
1;
        } else {
            echo "<td bgcolor=\"#FFFFFF\" >&nbsp;  </td>";
        }
    }
}
echo "</tr>";
// echo "<td bgcolor=\"#FFFFFF\" >Pusat Cluster</td>";
for ($i=0; $i < $jml_cluster; $i++) {
    for ($j=0; $j < count($tagihan); $j++) {
        $pusat_cluster_baru[$i][$j] = $pusat_cluster_baru[$i][$j] /
$jml_keanggotaan[$i][$j];
        // echo "<td bgcolor=\"#FFFFFF\"
>\".$pusat_cluster_baru[$i][$j].\"</td>";
    }
}
echo "</tr>";
echo "</table>";

echo "<br/>";
echo "Jarak Antar Cluster";
echo "<br/>";
echo "<br/>";
echo "<table width=\"700\" border=\"1\" cellpadding=\"3\" >";

$bcv = 0;
for ($i=0; $i < $jml_cluster; $i++) {
    for ($j=$i+1; $j < $jml_cluster; $j++) {
        echo "<tr>";
        echo "<td bgcolor=\"#FFFFFF\" >\".C\".($i+1).\"</td>";
        echo "<td bgcolor=\"#FFFFFF\" >\".C\".($j+1).\"</td>";
        $jarak_antar_cluster = 0;
        for ($k=0; $k < count($kriteria); $k++) {
            $jarak_antar_cluster = $jarak_antar_cluster +
(($pusat_cluster[$i][$k] - $pusat_cluster[$j][$k]) * ($pusat_cluster[$i][$k] -
$pusat_cluster[$j][$k]));
        }
        $jarak_antar_cluster = sqrt($jarak_antar_cluster);
    }
}

```

```

        echo "<td bgcolor=\"#FFFFFF\"
>\".$jarak_antar_cluster.</td>";
        $bcv = $bcv + $jarak_antar_cluster;
        echo "</tr>";
    }
}
echo "</table>";

echo "<br/>";
// echo "BCV = ".$bcv;
echo "<br/>";

$rasio = $bcv / $wcv;

echo "<br/>";
// echo "Rasio = ".$rasio;
echo "<br/>";

if ($rasio <= $rasio_sebelum) {
    $rasio_sebelum = $rasio;
    $ketemu = true;
    break;
} else {
    $rasio_sebelum = $rasio;
}
}

?>

</div>
<br />
<input type="button" value="Perhitungan"
onclick="document.getElementById('perhitungan').style.display='block';"/>
<br />
<br />

<?php
if ($ketemu == true) {
    echo "<br/>";
    echo "Hasil Didapatkan, Rasio Sudah Tidak Lebih Besar dari Rasio
Sebelumnya";
    echo "<br/>";
    echo "<br/>";
    echo "Keanggotaan Cluster Akhir";
    echo "<br/>";
    echo "<br/>";

```

```

        echo "<table width=\"700\" border=\"1\" cellpadding=\"1\"
cellpadding=\"3\" >";
        echo "<tr>";
        echo "<td bgcolor=\"#FFFFFF\">Debit Nota</td>";
        echo "<td bgcolor=\"#FFFFFF\">Keanggotaan</td>";
        echo "</tr>";

        $this->db->empty_table('tb_hasil');

        for ($i=0; $i < count($tagihan); $i++) {
            $this->db->select('*');
            $this->db->from('tb_cluster');
            $this->db->order_by('id_cluster');
            $cluster = $this->db->get()->result_array();
            foreach ($cluster as $data_cluster) {
                if ($keanggotaan[$i]==$data_cluster["id_cluster"]) {
                    $c = $data_cluster["nama_cluster"];
                    $masuk = array(
                        'no_debit' => $tagihan[$i],
                        'cluster' => $c
                    );
                    $this->db->insert('tb_hasil', $masuk);
                }
            }
            echo "<tr>";
            echo "<td bgcolor=\"#FFFFFF\" >\". $tagihan[$i].\"</td>";
            echo "<td bgcolor=\"#FFFFFF\" >\". $c.\"</td>";
            echo "</tr>";
        }
        echo "</table>";
    } else {
        echo "Perhitungan Belum Mendapatkan Hasil Akhir, Rasio Masih
Lebih Besar dari Rasio Sebelumnya, Silakan Dicoba Iterasi Lebih Banyak Lagi";
    }
    ?>
</div>

</td>
</tr>
</table>

<!-- /.card-body -->

<!-- /.card-footer-->
</div>
<!-- /.card -->

```



```

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
  <div class="float-right d-none d-sm-block">
    <b>Version</b> 3.0.1
  </div>
  <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
  reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
  <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>
</html>

```

6.5.9 Pengelompokkan

1. Controller

Admedika.php

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Admedika extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_admedika');
    }
    public function view_hasil()

```

```

{
    $yomay = $this->uri->segment(3);
    $data['hasil'] = $this->m_admedika->getHasil($yomay);
    $data['yahuy'] = $yomay;
    $data['mendesak'] = $yomay;
    $data['tidak'] = $yomay;
    $data['tes'] = 3;

    $this->load->view('admedical/hasil', $data);
}

public function getMendesak($yomay)
{
    $data['hasil'] = $this->m_admedika->getMendesak($yomay);
    $data['yahuy'] = $yomay;
    $data['mendesak'] = $yomay;
    $data['tidak'] = $yomay;
    $data['tes'] = 2;

    $this->load->view('admedical/hasil', $data);
}

public function getTidak($yomay)
{
    $data['hasil'] = $this->m_admedika->getTidak($yomay);
    $data['yahuy'] = $yomay;
    $data['mendesak'] = $yomay;
    $data['tidak'] = $yomay;
    $data['tes'] = 1;
    $this->load->view('admedical/hasil', $data);
}
}

```

2. Model

M_Admedika.php

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class M_admedika extends CI_Model {
    public function getHasil($id_rs)
    {
        return $this->db->query("SELECT * FROM tb_hasil A

```

```

        LEFT JOIN tagihan B
        ON A.no_debit = B.no_debit_nota
        LEFT JOIN rs C
        ON B.id_rs = C.id_rs where B.id_rs = '$id_rs')->result();
    }

    public function getHasil2($id_rs,$cluster)
    {
        return $this->db->query("SELECT * FROM tb_hasil A
        LEFT JOIN tagihan B
        ON A.no_debit = B.no_debit_nota
        LEFT JOIN rs C
        ON B.id_rs = C.id_rs where B.id_rs = '$id_rs' AND A.cluster =
        '$cluster')->result();
    }

    public function getMendesak($id_rs)
    {
        return $this->db->query("SELECT * FROM tb_hasil A
        LEFT JOIN tagihan B
        ON A.no_debit = B.no_debit_nota
        LEFT JOIN rs C
        ON B.id_rs = C.id_rs where B.id_rs = '$id_rs' AND A.cluster
        ='MENDESAK')->result();
    }

    public function getTidak($id_rs)
    {
        return $this->db->query("SELECT * FROM tb_hasil A
        LEFT JOIN tagihan B
        ON A.no_debit = B.no_debit_nota
        LEFT JOIN rs C
        ON B.id_rs = C.id_rs where B.id_rs = '$id_rs' AND A.cluster ='TIDAK
        MENDESAK')->result();
    }

    public function getIdHasil($yaww)
    {
        return $this->db->query("SELECT c.id_rs FROM tb_hasil A
        LEFT JOIN tagihan B
        ON A.no_debit = B.no_debit_nota
        LEFT JOIN rs C
        ON B.id_rs = C.id_rs where B.id_rs = '$yaww')->result();
    }
}

```

3. View

Pertamina

Uma Ainun

Dashboard

Data User

Data Rumah Sakit

Data Tagihan

Klasifikasi

Centroid

Analisis

Hasil Cluster

Logout

List Tagihan

Home / Tagihan / List

Show 10 entries

Search:

Tidak Mendeskri Mendeskri Cetak

No	Tgl Masuk	Tgl Bayar	Nama Rumah Sakit	No Debit	DPP	Tagihan	CN	Pembayaran	Keterangan
1	2020-01-15	2020-02-05	RS Pertamina Pusat	148806/1900006	4890768	59012617	1809	5011508	TIDAK MENDESAK
2	2020-01-15	2020-02-05	RS Pertamina Pusat	148806/1900008	4890768	4936278	162501	3872458	TIDAK MENDESAK
3	2020-01-15	2020-02-05	RS Pertamina Pusat	148806/1900009	4890768	21270571	201609	23000562	TIDAK MENDESAK
4	2020-01-15	2020-02-05	RS Pertamina Pusat	148806/1900011	4890768	5537126	201609	5335317	TIDAK MENDESAK
5	2020-01-15	2020-03-29	RS Pertamina Pusat	148806/1900005	4890768	29100717	29020	20131697	MENDESAK
6	2020-01-15	2020-02-05	RS Pertamina Pusat	148806/1900008	4890768	8705387	141051	8564236	TIDAK MENDESAK
7	2020-01-15	2020-02-05	RS Pertamina Pusat	148806/1900028	69050	2944315	201609	1042706	TIDAK MENDESAK
8	2020-01-15	2020-02-05	RS Pertamina Pusat	148806/1900027	1312230	12227959	201609	12006350	TIDAK MENDESAK
9	2020-01-15	2020-03-29	RS Pertamina Pusat	148806/1900020	6275270	21088226	022253	31175973	MENDESAK
10	2020-01-15	2020-03-29	RS Pertamina Pusat	148801/1904020	9500000	34312824	192000	34120824	MENDESAK

Showing 1 to 10 of 10 entries

Activate Windows
Go to Settings to activate Windows.

Previous 1 Next

Gambar 6. 51 Tampilan Pengelompokkan

Pertamina

Uma Ainun

Dashboard

Data User

Data Rumah Sakit

Data Tagihan

Klasifikasi

Centroid

Analisis

Hasil Cluster

Logout

List Tagihan

Home / Tagihan / List

Show 5 entries

Search:

Tidak Mendeskri Mendeskri Cetak

No	Tgl Masuk	Tgl Bayar	Nama Rumah Sakit	No Debit	DPP	Tagihan	CN	Pembayaran	Keterangan
1	2020-01-15	2020-01-29	RS Pertamina Pusat	148806/1903005	4890768	29160717	29020	29131697	MENDESAK
2	2020-01-15	2020-01-29	RS Pertamina Pusat	148801/1903029	6275270	31858226	682253	31175973	MENDESAK
3	2020-01-15	2020-01-29	RS Pertamina Pusat	148801/1904020	9500000	34312824	192000	34120824	MENDESAK
4	2020-01-15	2020-01-29	RS Pertamina Pusat	148801/1907016	4890768	28395469	201609	28194860	MENDESAK
5	2020-01-15	2020-01-29	RS Pertamina Pusat	148806/1902007	7297690	33262307	201609	33060698	MENDESAK

Showing 1 to 5 of 5 entries

Activate Windows
Go to Settings to activate Windows.

Previous 1 Next

Gambar 6. 52 Tampilan Pengelompokkan

Copyright © 2014-2019 AdminLTE Inc. All rights reserved. Version 3.0.1

Gambar 6. 53 Tampilan Pengelompokkan

```

Hasil.php
<?php $this->load->view('admedical/atribut/head');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>List Tagihan</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item">Tagihan</li>
            <li class="breadcrumb-item">List</li>
          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">

    <!-- Default box -->
  
```

```

<div class="card">
  <div class="card-header">
    <h3 class="card-title">List Tagihan</h3>

    <div class="card-tools">
      <button type="button" class="btn btn-tool" data-card-widget="collapse"
data-toggle="tooltip" title="Collapse">
        <i class="fas fa-minus"></i></button>
      <button type="button" class="btn btn-tool" data-card-widget="remove"
data-toggle="tooltip" title="Remove">
        <i class="fas fa-times"></i></button>
    </div>
  </div>
  <div class="card-body">
    <table class="table table-light table-hover table-bordered" id="example1">
      <thead class="thead-light">
        <a href="#"><?php echo base_url('admedika/cetak2/'.$yahuy.'/'.$tes) ?>"
class="btn btn-danger float-right"><i class="fa fa-print">Cetak</i></a>
        <a href="#"><?php echo base_url('admedika/getMendesak/'.$mendesak)
?>" class="btn btn-primary float-right"><i class="">Mendesak</i></a>
        <a href="#"><?php echo base_url('admedika/getTidak/'.$tidak) ?>"
class="btn btn-primary float-right"><i class="">Tidak Mendesak</i></a>

      <tr>
        <th>No</th>
        <th>Tgl Masuk</th>
        <th>Tgl Bayar</th>
        <th>Nama Rumah Sakit</th>
        <th>No Debit</th>
        <th>DPP</th>
        <th>Tagihan</th>
        <th>CN</th>
        <th>Pembayaran</th>
        <th>Keterangan</th>

      </tr>
    </thead>
    <tbody>
      <?php
      $no = 1;
      foreach($hasil as $h) {
      ?>
      <tr>
        <td><?= $no++; ?></td>
        <td><?= $h->tgl_masuk; ?></td>
      </tr>
      <?php

```

```

        $start_date = $h->tgl_masuk;
        if ($h->cluster == "TIDAK MENDESAK") { ?>
        <td><?= date("Y-m-d", strtotime($start_date. "+3 week")); ?></td>
        <?php }else{ ?>
        <td><?= date("Y-m-d", strtotime($start_date. "+2 week")); ?></td>
        <?php } ?>
        <td><?= $h->nama_rs; ?></td>
        <td><?= $h->no_debit; ?></td>
        <td><?= $h->dpp; ?></td>
        <td><?= $h->tagihan; ?></td>
        <td><?= $h->credit_nota; ?></td>
        <?php
        $pembayaran = $h->tagihan-$h->credit_nota;
        ?>
        <td><?= $pembayaran; ?></td>
        <td><?= $h->cluster; ?></td>

    </tr>
    <?php } ?>
</tbody>
</table>
</div>
<!-- /.card-body -->

<!-- /.card-footer-->
</div>
<!-- /.card -->

</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.1
    </div>
    <strong>Copyright &copy; 2014-2019 <a
href="http://adminlte.io">AdminLTE.io</a>.</strong> All rights
    reserved.
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>

```

```
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

<?php $this->load->view('js');?>
</body>
</html>
```


DAFTAR PUSTAKA

- [1] R. Abdulloh, 7 In 1 Pemrograman Web Untuk Pemula, Jakarta: PT Elex Media Komputindo, 2018.
- [2] S. Galih, "Sejarah dan Perkembangan HTML," INDOSMARTDIGITAL, 27 September 2016. [Online]. Available: <https://www.indosmartdigital.com/artikel-19-sejarah-dan-perkembangan-html.html>. [Accessed 10 Februari 2020].
- [3] J. S. Putra, "Kekurangan Dan Kelebihan HTML," May 2017. [Online]. Available: <https://pengembanganwebindonesia.wordpress.com/2017/05/17/kekurangan-dan-kelebihan-html/>. [Accessed 13 Februari 2020].
- [4] A. Tsabit, M. A. Ramdani and R. Cahyana, "Pengembangan Ganesha Digital Library Untuk Membuat Situs Jurnal," *Jurnal Algoritma*, vol. 09, no. 1, 2012.
- [5] F. Hidayan, "Sejarah Pengembangan CSS Oleh W3C," 5 Februari 2017. [Online]. Available: <https://blog.sekolahprogram.com/sejarah-pengembangan-css/>. [Accessed 15 Februari 2020].
- [6] S. Manis, "Pengertian PHP, Sejarah dan Fungsi PHP (Personal Home Page) Dalam Pemograman Web Lengkap," 3 September 2018. [Online]. Available: <https://www.pelajaran.co.id/2018/03/pengertian-php-sejarah-dan-fungsi-php-personal-home-page-dalam-pemograman-web-lengkap.html>. [Accessed 15 Februari 220].
- [7] Wistau, "Type data pada PHP (Integer, Float, String, Boolean, Array)," 06 Juni 2015. [Online]. Available: <http://wistau.com/type-data-pada-php-integer-float-string-boolean-array-2/>. [Accessed 15 Februari 2020].
- [8] "Sejarah Singkan Dan Perkembangan CSS (Cascading Style-Sheet)," Ulas Informatika, [Online]. Available: <https://ulasinformatika.wordpress.com/2013/11/19/sejarah-singkat-dan-perkembangan-css-cascading-style-sheet/>. [Accessed 15 Februari 2020].
- [9] . M. Ilham, "Kelebihan dan kekurangan PHP beserta Contoh Program PHP," Materi Belajar, 14 Oktober 2019. [Online]. Available:

<https://materibelajar.co.id/kelebihan-dan-kekurangan-php/>. [Accessed 15 Februari 2020].

- [10] Wikipedia, "MySQL," Wikipedia, 6 Oktober 2019. [Online]. Available: <https://id.wikipedia.org/wiki/MySQL>. [Accessed 15 Februari 2020].
- [11] . S. F. Fadillah, "15 Kelebihan dan Kekurangan MySQL Server yang Perlu Diketahui," Nesabamedia, 20 Februari 2020. [Online]. Available: <https://www.nesabamedia.com/15-kelebihan-dan-kekurangan-mysql-server-yang-perlu-diketahui/>. [Accessed 25 Februari 2020].
- [12] A. C, "Apa Itu JavaScript? Pemahaman Dasar Mengenai JavaScript bagi Para Pemula," hostinger, 23 Januari 2019. [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-javascript/>. [Accessed 20 Februari 2020].
- [13] H. Ermawati, "Sejarah, Definisi & Komponen Sistem Basis Data," 07 Agustus 2017. [Online]. Available: <https://simpleheny.blogspot.com/2017/08/sejarah-definisi-komponen-sistem-basis.html>. [Accessed 20 Februari 2020].
- [14] I. Syafitri, "Pengertian Framework Beserta Fungsi dan Jenis-Jenis Framework, Sudah Tahu?," Nesabamedia, 03 September 2019. [Online]. Available: <https://www.nesabamedia.com/pengertian-framework/>. [Accessed 20 Februari 2020].
- [15] A. Nurzahputra, M. A. Muslim and M. Khusniati, "Penerapan Algoritma K-Means Untuk Clustering Penilaian Dosen Berdasarkan Indeks Kepuasan Mahasiswa," *Techno.COM*, vol. 16, no. 1, pp. 17-24, 2017.
- [16] T. N. Hermawan, M. Ugiarto and N. Puspitasari, "Sistem Evaluasi Kinerja Asisten Laboratorium Menggunakan Metode K-means," *Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi*, vol. 3, no. 2, 2017.
- [17] A. Maulana, M. Sadikin and A. Izzuddin, "Implementasi Sistem Informasi Manajemen Inventaris Berbasis Web Di Pusat Teknologi Informasi Dan Komunikasi – BPPT," *SETRUM*, vol. 7, no. 1, 2018.

- [18] H. Tarmizi, "Pengertian Sublime Text," 26 April 2017. [Online]. Available: <https://hasantarmizi.blogspot.com/2017/04/pengertian-sublime-text.html>. [Accessed 21 Februari 2020].
- [19] Andy, "Pengertian XAMPP Lengkap dengan Fungsi dan Cara Instalasi," qword, 09 September 2019. [Online]. Available: <https://qwords.com/blog/pengertian-xampp/>. [Accessed 21 Februari 2020].
- [20] M. Ilmi, D. L. Said and A. Hadi, "Perancangan Sistem Informasi Pernikahan Dan Status Pernikahan Kantor Urusan Agama Kecamatan Canduang Agam Berbasis Web Dengan Framework Codeigniter," *Voteknika*, vol. 4, no. 1, 2016.
- [21] W. Persolita, F. . E. and R. F. F. Afidh, "Aplikasi Web Penjualan Qubah Stainless Steel Di Mahkota Barokah Menggunakan Pemrograman PHP," *INFORMATIKA*, vol. 10, no. 1, 2018.