

MEMBUAT SISTEM E-BUDGETING DENGAN PHP, CODEIGNITER, DAN BOOTSTRAP

Septi Nurhidayah 1.16.4.027

KATA PENGANTAR

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PHP

1.1 Pengertian PHP

PHP merupakan singkatan dari *Hypertext Preprocessor* dengan Bahasa yang berbentuk skrip yang bersifat *server side* yang dimana proses pengerjaan kode program dilakukan di *server*, dan hasilnya akan ditampilkan di *browser*. PHP bekerja di dalam sebuah dokumen *Hypertext Markup Language* (HTML) untuk dapat menghasilkan isi dari sebuah halaman *web* sesuai permintaan. PHP bersifat *software open source* dan juga *software cross platform*, jadi bisa berjalan dengan baik pada sistem operasi Windows, Mac OS, maupun Unix (Linux). PHP merupakan Bahasa pemrograman yang cukup populer di kalangan *developer* untuk membuat *website* dinamis. PHP dapat dijalankan secara runtime melalui console dan juga dapat menjalankan perintah-perintah sistem. PHP memiliki *website* resmi, yaitu <https://www.php.net/>.

1.2 Sejarah PHP

Pada awalnya PHP adalah kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Rasmus Lerdorf sendiri merupakan seorang *programmer* yang berasal dari Denmark/Greenland. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisannya kode sumber ini menjadi sumber

terbuka (*open source*), maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI versi 2.0. Pada perilisan ini, *interpreter* PHP sudah diimplementasikan dalam program C dan disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikannya sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang *PHP: Hypertext Preprocessor*.

Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan dikenal dengan PHP 4.0. PHP 4.0 ini adalah versi PHP yang paling banyak digunakan pada awal abad ke-21. Versi ini banyak digunakan dengan alasan kemampuannya untuk membangun aplikasi *web* yang kompleks dan tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan yang cukup besar. Versi ini juga memasukkan model pemrograman berorientasi objek atau *Object Oriented Programming* (OOP) ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. *Web server* bawaan ditambahkan pada versi 5.4 untuk mempermudah pengembang dalam menjalankan kode PHP tanpa menginstall *software server* terlebih dahulu.

Pada tanggal 17 Februari 2017, versi terbaru dan stabil dari bahasa pemrograman PHP saat ini resmi dirilis, yaitu versi 7.0.16 dan 7.1.2.

1.3 Fungsi PHP

Dalam membangun sebuah *web* PHP, PHP bukan saja menjadi bahasa pemrograman yang utama. Menggunakan HTML dan CSS sudah bisa menghasilkan halaman *web* statis, yang mana kontennya bersifat tetap. Bahasa PHP bisa diartikan sebagai bahasa pemrograman pelengkap yang digunakan untuk menghasilkan sebuah *web* yang dinamis dan interaktif. Berikut fungsi dari PHP, diantaranya yaitu:

1. Mempersingkat Tatahan HTML dan CSS

Dalam membangun sebuah halaman *web* yang dinamis, PHP bisa berfungsi sebagai penyingkat penggunaan tatahan HTML dan CSS. Sebagai contoh dalam suatu sistem mahasiswa memiliki jumlah baris sebanyak 10. Jika menggunakan HTML dan CSS maka barisan tersebut akan menjadi sangat panjang. Sedangkan jika ditambah dengan menggunakan PHP, maka bisa mengatur beberapa baris yang diperlukan atau ditampilkan.

2. Input Data

Dengan menggunakan bahasa pemrograman PHP, maka bisa melakukan input data dan menyimpannya dalam sistem *database* seperti MariaDB.

3. Manajemen Cookie dan Session

Dalam PHP, *cookie* dan *session* digunakan untuk menyimpan informasi pengguna. Fungsi *session_start* untuk memulai *session* dan fungsi *setcookie* untuk *cookie*. Sebagai proses *cookie*, menyimpan *username* dan *password* pengguna pada *browser* menjadikan tidak harus mengisinya ulang pada saat membuka situs yang sama. *Session*

contohnya menyimpannya informasi *login* yang hanya berlaku dalam satu sesi saja.

4. Kompres Teks

PHP bisa melakukan kompresi teks yang panjang menjadi lebih pendek dengan fungsi `gzcompress` dan mengembalikan dengan fungsi `gzuncompress`.

1.4 Aturan Penulisan PHP

Berikut ini adalah aturan penulisan kode PHP:

1. Penamaan file

Penamaan file PHP harus diakhiri dengan ekstensi `.php`.

2. Penggunaan open tag `<?php`

Untuk memulai *script* PHP harus diawali dengan *open tag* `<?php`.

3. Penggunaan close tag `?>`

Gunakan *close tag* `?>` hanya jika *script* PHP yang dibuat bergabung dengan *script* lain seperti HTML yang jelas *close tag* tersebut sebagai akhir dari *script* PHP. Jika *script* hanya berisi file PHP, maka kita tidak perlu menutupnya dengan closing tag `?>`.

4. Akhir baris program PHP

Setiap baris intruksi program pada PHP diakhiri dengan tanda titik koma (;). Hal ini berarti walaupun menulis lebih dari 1 baris, hal

tersebut akan tetap dianggap hanya satu baris apabila belum diberikan titik koma (;).

5. Tanda kurung ()

Salah satu fungsi tanda kurung () adalah untuk memanggil sebuah fungsi. Contoh: `print()`;

6. Case sensitive

Pada PHP tidak ada perbedaan format penulisan untuk huruf besar dan kecil (*case sensitive*) untuk penulisan fungsi, class yang dideklarasikan sendiri, maupun constructor bawaan PHP seperti `print`, `echo`, `for`, `while` dan `class`.

7. Penulisan baris perintah

Statement atau baris perintah merupakan kumpulan perintah PHP yang menginstruksikan PHP untuk melakukan sesuatu, baik perintah sederhana, seperti menyimpan data ke variabel `$a = 1` atau mencetak *output* seperti `echo $a`, maupun yang rumit seperti, logika `if else` atau `for loop`. Setiap *statement* ini harus diakhiri dengan tanda *semicolon* (;) kecuali perintah yang menggunakan kurung kurawal {}, contoh seperti `statement class`, `if else`, `do while`, dll.

Contoh penggunaan *semicolon* (;) :

```
1  <?php
2      echo 'Contoh';
3      trim($variable);
4      $a = $b + 1;
5  ?>
```

Contoh penggunaan tanpa *semicolon* (;) :

```

1  <?php
2      //cek apakah kondisi true
3      if (true) {
4          //jangan lupakan titik koma
5          echo "Baris ini akan dijalankan";
6      } //setelah kurung kurawal tutup, tidak perlu titik koma
7  ?>

```

8. Penulisan komentar pada PHP

Pada PHP, penulisan komentar diawali dengan menggunakan tanda // untuk satu baris komentar, dan /* ... */ untuk komentar yang terdiri dari satu baris atau lebih.

```

87  /*
88  |-----
89  | Constants for Site
90  |-----
91  |
92  */
93
94  // Untuk menampilkan nama website
95  define('SITE_NAME', 'E-Budgeting');

```

9. Karakter spasi, tab, dan enter pada PHP

Karakter spasi, tab dan enter sering diabaikan pada saat eksekusi program dalam PHP. Hal ini akan memudahkan kita untuk menulis script dengan baik, karena dapat memecah statement menjadi beberapa baris.

1.5 Tipe Data PHP

Setiap bahasa pemrograman memiliki kebijakan sendiri terkait tipe data. Ada yang strict (ketat) ada yang longgar, strict artinya tipe data harus benar benar sesuai dengan yang disyaratkan.

Misal untuk melakukan operasi matematika, data yang terlibat harus ber tipe integer, demikian juga untuk string, jika tidak maka akan muncul pesan error atau output yang dihasilkan akan tidak sesuai dengan harapan.

PHP sendiri termasuk bahasa pemrograman yang longgar untuk masalah tipe data, ketika menggunakan PHP, kita tidak perlu dipusingkan dengan tipe data, karena ketika melakukan operasi, PHP akan otomatis mengubah nilai sesuai dengan tipe data yang diperlukan.

Misal: jika kita memiliki angka dalam bentuk string, entah angka hasil pengambilan data dari database, hasil regex/pemecahan dari string lain dan kita lakukan operasi aritmetika, maka kita tidak akan mendapati error dan akan memperoleh hasil sesuai dengan yang kita harapkan, misal:

```
<?php
$nilai = 'Nilai UAS = 9';
echo substr($nilai,-1) + 5; // 14
?>
```

Pada contoh diatas, fungsi substr(\$nilai,-1) akan menghasilkan angka 9 dengan tipe data string lalu kita jumlahkan dengan 9 yang memiliki tipe data integer, hasilnya adalah 14 dengan tipe data integer.

Pada operasi aritmetika, di belakang layar PHP akan mengubah tipe data string tadi ke integer, sehingga akan sama tipe datanya, proses pengubahan ini disebut type juggling.

Pada contoh lain, misal kita akan melakukan operasi string pada integer, maka PHP akan mengubah tipe data bilangan tersebut menjadi string, misal:

```
<?php
$angka = 12345;
echo substr(12345, 3);
?>
```

Script tersebut akan menghasilkan string 45.

Pada PHP, saat mendeklarasikan variabel, kita tidak perlu mendefinisikan tipe data (fitur default PHP), misal:

```
$harga = '1000'; // Tipe data string
```

PHP akan otomatis mendeteksi tipe data dari nilai yang kita berikan, misal ketika nilai diawali dengan tanda kutip maka otomatis dianggap string.

PHP membedakan kelompok data berdasarkan jenisnya, seperti berikut:

1. Integer. berupa angka tanpa desimal dan tanpa awalan 0.

```
$nilai = 10;
echo $nilai; // 10;

$nilai2 = 01234;
echo $nilai2; // 668 - bilangan octal
```

Note: Jika integer diawali dengan 0 maka menjadi sistem bilangan octal, seperti pada contoh \$nilai2 diatas.

2. String. berupa karakter, baik angka, huruf, karakter khusus, dll. Penulisannya diapit dengan satu tanda kutip " atau dua tanda kutip "", misal: \$nilai = '10' lebih jauh tentang string dapat dibaca disini.
3. Float. berupa angka dengan desimal (desimal separatnya berupa titik), misal: \$nilai = 7.5
4. Booleans. berupa true atau false, misal:

```
$aktif = false;
if (date('Y-m-d') > "2017-12-1") {
    $aktif = true;
}

if ($aktif) {
    echo 'aktif';
}
```

5. Array. berupa array, misal:

```
$array = array('Mobil', 'Motor');
print_r($array);
```

6. Object. berupa objek (instansi dari class), misal:

```
class User {
    ...
}
```

```
}  
  
$obj = new User; // Variabel $obj berisi data dengan tipe objek
```

Pada bagian sebelumnya telah disinggung sekilas mengenai type juggling dimana untuk kebutuhan tertentu dibelakang layar PHP dapat mengubah tipe data suatu variabel. Umumnya type juggling terjadi pada dua keadaan, yaitu (1) pada penggunaan operator aritmetika dan (2) penggunaan fungsi string. Contoh type juggling ketika menggunakan operator aritmetika adalah sebagai:

```
$harga = 1000; // integer  
$diskon = 0.5; // float  
$diskon = 1000 * 0.5; // variabel harga menjadi float  
$harga = $harga - $diskon; // float
```

Contoh type juggling ketika menggunakan fungsi string adalah sebagai berikut:

```
$harga = 1000;  
$harga = 'Total harga ' . $harga;  
echo $harga; // Total harga 1000 -> variabel $harga memiliki tipe data string
```

Selain operasi aritmetika dan string, type juggling juga terjadi ketika ada perbandingan dua variabel, misal ketika menggunakan pada operator pembandingan sebagai berikut:

```
$nilai      = '10';  
$nilai2 = '5';  
if ($nilai > $nilai2) {  
    echo 'Benar';  
}  
// Output: Benar
```

Pada contoh tersebut, sebelum PHP membandingkan kedua nilai, PHP akan mengubah keduanya menjadi integer.

Pada kondisi tertentu, kita perlu untuk menguji tipe data untuk memastikan bahwa data yang akan kita olah memiliki tipe data yang sesuai dengan yang kita harapkan. Untuk menguji tipe data, kita gunakan fungsi `is_xxx` diantaranya adalah:

- `is_string()` untuk menguji apakah nilai berupa string;
- `is_int()` atau `is_integer()` untuk menguji apakah nilai berupa integer;
- `is_array()` untuk menguji apakah nilai berupa array;
- `is_object()` untuk menguji apakah nilai berupa object;
- `is_float()` untuk menguji apakah nilai berupa float.

Misal:

```
$nilai = 10;
is_int($nilai); // true

$harga = '10000';
is_string($harga); // true

$array = array(10);
is_array($array); // true
```

Dalam prakteknya, pengujian nilai ini digunakan untuk menguji argumen yang dimasukkan kedalam fungsi, misal kita buat fungsi untuk memecah string sebagai berikut:

```
function cut_text($kata) {
    if(!is_string($kata)) {
        return false;
    }

    return substr($kata, 0, 100);
}
```



```
}
```

Atau untuk argumen array, misal:

```
function split_array ($array) {  
    if(!is_array($kata)) {  
        return false;  
    }  
  
    foreach ($array as $key => $val) {  
        ...  
    }  
}
```

1.6 Kelebihan PHP

PHP merupakan salah satu Bahasa pemrograman yang banyak digunakan, tentu dalam hal ini ada beberapa alasan yang menjadikan Bahasa pemrograman ini diminati oleh pengembang, salah satunya adalah mempunyai kelebihan dibandingkan dengan Bahasa pemrograman lain yang sejenis. Berikut ini beberapa kelebihan dari Bahasa pemrograman PHP:

1. Dapat membuat web menjadi lebih dinamis.
2. Bersifat *open source* yang artinya bisa digunakan oleh siapa saja secara gratis.
3. Bersifat multiplatform sehingga program yang dibuat dengan PHP dapat dijalankan di semua sistem operasi karena PHP berjalan secara *web base* yang berarti semua sistem operasi bahkan *Handphone* 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 MariaDB, Oracle, PostgreSQL, dan lain-lain.

6. Tidak membutuhkan kompilasi dalam penggunaannya.
7. Banyak *web server* yang mendukung PHP seperti Apache, Lighttpd, IIS dan lainnya dengan konfigurasi yang mudah dan tidak rumit.
8. Dalam pengembangan aplikasi, PHP lebih mudah karena banyak dokumentasi, referensi dan *developer* yang mendukung dalam pengembangannya.

1.7 Kekurangan PHP

Selain mempunyai kelebihan, Bahasa pemrograman PHP juga mempunyai kekurangan. Berikut ini beberapa kekurangan dari Bahasa pemrograman PHP yang bisa menjadi pertimbangan bagi pengembang dalam memilih Bahasa pemrograman ini:

1. Dalam PHP tidak mengenal *Package*.
2. Jika tidak di-*encoding*, maka kode PHP dapat dibaca oleh semua orang dan untuk meng-*encoding*-nya dibutuhkan *tool* dari Zend yang memiliki biaya yang sangat mahal.
3. PHP memiliki kelemahan keamanan tertentu. Jadi, *Programmer* harus lebih teliti dan berhati-hati dalam melakukan pemrograman serta konfigurasi PHP.

BAB II

CodeIgniter

2.1 Pengertian CodeIgniter

CodeIgniter merupakan aplikasi berbasis web yang bersifat open *source* yang berupa *framework* PHP dengan konsep Model, View, Controller (MVC) untuk membangun sebuah *website* dinamis dengan menggunakan Bahasa pemrograman PHP. CodeIgniter membantu pengembang untuk membuat aplikasi web dengan lebih mudah dan cepat dibandingkan dengan membuatnya dari awal. Selain itu, CodeIgniter juga mempunyai dokumentasi yang lengkap dan disertai dengan contoh implementasi kodenya. Dokumentasi inilah yang menjadi salah satu alasan mengapa pengembang memilih CodeIgniter sebagai *framework* pilihannya. Dibandingkan dengan *framework* PHP lainnya, CodeIgniter memiliki desain yang lebih sederhana dan fleksibel.

2.2 Sejarah CodeIgniter

CodeIgniter dikembangkan oleh Rick Ellis pada tahun 2006 dan dirilis pada 28 Februari 2006. Rick Ellis merupakan pendiri EllisLab. EllisLab adalah suatu tim kerja yang berdiri pada tahun 2002 yang bergerak dibidang pembuatan *software* dan *tool* untuk pengembang web.

Dengan logo api yang menyala, CodeIgniter dengan cepat “membakar” semangat para pengembang *web* untuk mengembangkan *web* dinamis dengan cepat dan mudah menggunakan *framework* PHP yang satu ini.

Pada 9 Juli 2013, EllisLab mengumumkan bahwa mereka mencari pemilik baru untuk CodeIgniter karena internal mereka sendiri tidak memiliki cukup fokus untuk terus mengembangkan CodeIgniter. Akhirnya pada 6 Oktober 2014, kepemilikan CodeIgniter berpindah tangan kepada British Columbia Institute of Technology (BCIT) untuk pengembangan lebih lanjut. Saat ini *website* resmi dari CodeIgniter adalah <https://codeigniter.com/> .

2.3 Konsep MVC pada CodeIgniter

CodeIgniter merupakan *framework* yang mengimplementasikan desain dengan konsep Model View Controller atau disingkat MVC. MVC merupakan metode pendekatan yang berfungsi memisahkan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi. MVC memiliki 3 komponen, yaitu:

1. Model

Model digunakan untuk mempresentasikan data yang akan digunakan oleh aplikasi, biasanya berhubungan dengan *database*. Biasanya melakukan operasi yang berupa manipulasi data seperti Insert, Update, Delete.

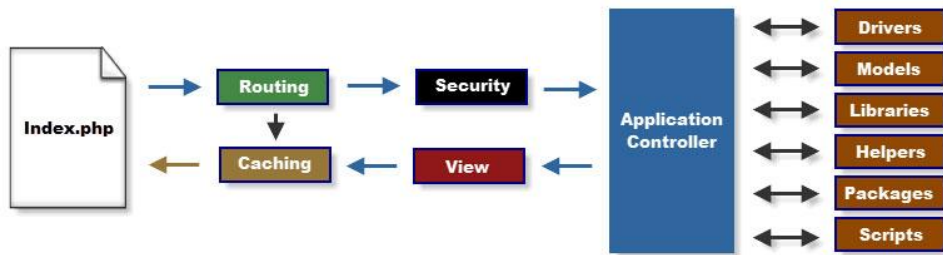
2. View

View berhubungan dengan tampilan *website* ke pengguna *website* yang umumnya berisi tampilan halaman *web* berupa HTML, PHP, dan JavaScript. Pada CodeIgniter, view bisa digunakan untuk *template* seperti *header* atau *footer*.

3. Controller

Controller berperan sebagai penghubung antara model dan view untuk memproses permintaan HTTP yang berasal dari pengguna untuk menghasilkan halaman *web*.

2.4 Alur Kerja CodeIgniter



Gambar 2. 1 Alur Kerja CodeIgniter

Keterangan:

- Index.php berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- Router akan memeriksa HTTP *request* atau permintaan untuk menentukan hal apa yang harus dilakukan selanjutnya oleh program.
- Cache file, apabila di dalam program sudah terdapat “*cache file*” maka file tersebut akan langsung dikirim ke *browser* melewati eksekusi sistem normal. File *cache* inilah yang dapat membuat sebuah *website* dapat dibuka dengan lebih cepat. *Cache file* dapat melewati proses yang sebenarnya harus dilakukan oleh program CodeIgniter.
- Security. Sebelum file controller dimuat keseluruhan, HTTP *request* atau permintaan dan data yang dikirimkan oleh pengguna akan disaring terlebih dahulu melalui fasilitas *security* yang dimiliki oleh CodeIgniter.
- Controller akan membuka file model, core libraries, helper dan semua sumber daya yang dibutuhkan dalam program tersebut.
- Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke *browser* supaya dapat dilihat. Apabila file view sudah ada yang di “*cache*” maka file

view baru yang belum ter-*cache* akan mengupdate file view yang sudah ada.

2.5 Kelebihan CodeIgniter

Beberapa kelebihan dari CodeIgniter adalah sebagai berikut:

1. Memiliki ukuran file sangat kecil. File *download*-nya hanya sekitar 2MB, itupun sudah termasuk dokumentasinya yang sangat lengkap.
2. Dokumentasi yang bagus. Pada saat mendownload file, telah disertakan juga dengan dokumentasi yang berisi pengantar, tutorial, bagaimana panduan penggunaan, serta referensi dokumentasi untuk komponen-komponennya.
3. CodeIgniter mampu berjalan dengan baik atau kompatibel pada hampir semua *platfom hosting*. CodeIgniter juga mendukung *database-database* paling umum, termasuk MariaDB.
4. Tidak ada aturan *coding* yang ketat. Terserah anda jika anda hanya ingin menggunakan Controller, tanpa View, atau tidak menggunakan Model, atau tidak salah satu keduanya. Namun dengan menggunakan ketiga komponennya adalah pilihan lebih bijak.
5. Kinerja yang baik. Codeigniter sangat cepat bahkan mungkin bisa dibilang merupakan *framework* yang paling cepat yang ada saat ini.
6. CodeIgniter sangat mengerti tentang pengembangan berbagai *library* saat ini. Karenanya CodeIgniter memberikan kemudahan untuk diintegrasikan dengan *library-library* yang tersedia saat ini.
7. CodeIgniter tidak membutuhkan konfigurasi yang banyak dan rumit, bahkan untuk mencoba menjalankannya, tanpa melakukan konfigurasi sedikitpun ia sudah bisa berjalan. Konfigurasi CodeIgniter terletak di folder **aplication/config**.
8. Mudah dipelajari. Di samping dokementasi yang lengkap, CodeIgniter juga memiliki berbagai forum diskusi.

2.6 Kekurangan CodeIgniter

Adapun kekurangan dari CodeIgniter adalah sebagai berikut:

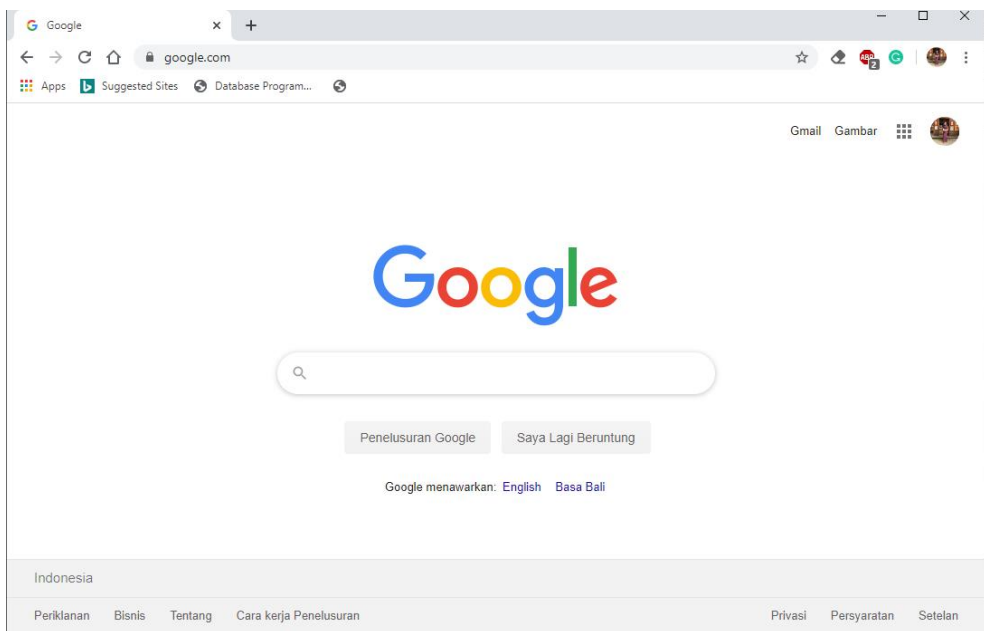
1. CodeIgniter tidak ditujukan untuk pembuatan *web* dengan skala besar.
2. Library yang masih terbatas. Hal ini dikarenakan sangat sulit mencari *plugin* tambahan yang terverifikasi secara resmi, karena pada situsnya CodeIgniter tidak menyediakan *plugin* tambahan untuk mendukung pengembangan aplikasi dengan CI.
3. Belum adanya *editor* khusus CodeIgniter, sehingga dalam melakukan *create project* dan modul-modulnya harus berpindah-pindah folder.

BAB III

Persiapan Tools

3.1 Web Browser

Web browser berfungsi untuk membuka atau menampilkan halaman atau konten aplikasi *website* yang dibuat. *Web browser* yang digunakan di sini adalah Google Chrome.



Gambar 3. 1 Google Chrome

Google Chrome merupakan web browser yang dikembangkan oleh Google. Google Chrome ini merupakan salah satu web browser yang sangat populer. Hal ini dikarenakan semua fitur dan kelebihan yang dimiliki, tak heran jika banyak orang yang suka menggunakan web browser ini.

Untuk memberikan user experience yang baik, Google selalu berusaha menghadirkan layanan serta fitur terbaik pada software ciptaannya. Begitu

juga dengan Chrome ini, ia dikembangkan dan dibekali dengan berbagai fitur yang mumpuni untuk memenuhi kebutuhan para penggunanya. Berikut ada beberapa keunggulan dari Google Chrome, yaitu:

1. Desain yang Minimalis

Jika kalian perhatikan, interface dari Google Chrome ini sangatlah sederhana. Tidak banyak hal yang neko-neko ataupun berlebihan disana. Hal ini malah menjadi kelebihan karena tentu akan lebih mudah digunakan. Karena tampilan yang minimalis maka user tidak akan kebingungan ketika mencari fitur-fitur penting yang ada disana. Tampilan menu yang dibuat sedemikian rupa dimaksudkan agar pengguna bisa menggunakannya dengan mudah. Jadi kesederhanaan bukanlah suatu kekurangan, tapi malah jadi kelebihannya.

2. Proses Loading Tab Terpisah

Peramban ini menerapkan sistem loading tab yang terpisah. Ini sangat berguna jika ada salah satu tab yang crash. Tab yang lain tetap aman dan dapat dibuka seperti biasa, hanya tab yang crash yang akan mengalami masalah.

3. Integrasi dengan Layanan Google

Google menyematkan Chrome dengan beragam layanan dari mereka. Seperti Gmail, Google Drive, dan masih banyak lagi. Semuanya akan lebih mudah dilakukan di Chrome dibandingkan di browser lain.

4. Mempermudah Pencarian

Seperti yang kita ketahui bersama bahwa saat ini Google merupakan situs mesin pencari terpopuler di dunia. Google bahkan menduduki peringkat 1 sebagai website yang paling banyak dikunjungi. Hampir

semua orang ketika ingin mencari suatu informasi, pasti mengawalinya dari Google.

5. Memiliki Banyak Ekstensi

Google sebenarnya sudah melengkapi Chrome dengan banyak fitur bawaan yang berguna. Mulai dari download manajer bawaan, terjemahan, login otomatis, penyimpanan password dan masih banyak lagi.

6. Tersedia Banyak Tema

Google Chrome memang dirancang dengan tampilan yang simpel dan minimalis. Namun beberapa orang mungkin ingin mendapatkan tampilan yang berbeda. Kita dapat mengunjungi Chrome Web Store dan mencari tema Google Chrome disana. Ada banyak sekali tema yang bisa di download. Beberapa diantaranya dibuat langsung oleh Google, dan sisanya dibuat oleh pengembang pihak ketiga.

7. Dapat Diintegrasikan dengan Internet Download Manager

Sebagian besar pengguna Windows yang sering mengunduh file dari internet pasti menggunakan aplikasi Internet Download Manager. Google Chrome biasanya akan langsung mengenali jika di PC terinstall Google Chrome. Jadi ketika mendownload file, akan langsung diarahkan menggunakan Google Chrome.

8. Selalu di Update

Chrome merupakan salah satu cara Google agar penggunaanya tetap menggunakan produk-produknya yang lain. Setiap update, Chrome membawa perbaikan dan juga fitur-fitur baru.

9. Tersedia di Banyak Platform

Saat ini ada tiga sistem operasi yang biasa digunakan orang di PC atau laptop mereka. Mulai dari Linux, macOS hingga Windows. Dan Google Chrome dapat digunakan diketiganya. Jadi sebagai pengguna, kamu bebas menggunakan Google Chrome di sistem operasi mana saja. Tidak heran kalau Google Chrome memiliki basis pengguna yang banyak. Tidak seperti beberapa browser lainnya yang hanya dapat digunakan di Windows saja, atau macOS saja.

10. Ada Fitur Terjemahkan Halaman Web

Dengan fitur terjemahan bawaannya, Google Chrome mampu mengubah bahasa suatu website menjadi Bahasa Indonesia.

Selain keunggulan, ternyata Google Chrome juga memiliki kelemahan, seperti berikut:

1. Tingkat Privacitas

Google melakukan proses penyimpanan sekitar 2% interpretation atau pencarian pemakainya dan hal ini termasuk juga alamat IP. Meski pada saat-saat tertentu proses pencarian identitas pemakai tersebut dianonimkan, namun tetap saja ada kemungkinan bahwa Goolge tetap dapat mengetahui siapa saja yang menggunakan fasilitasnya. Bahkan termasuk juga lokasi dimana pemakai tersebut berada. Hal ini tentu saja sangat mengurangi privacitas penggunaannya.

2. Sistem Keamanan

Beberapa waktu ada beberapa ahli confidence yang berhasil menemukan dan membuktikan adanya suatu bugs atau celah kecil pada

Google Chrome sehubungan dengan sistem keamanannya. Ketika ada pengguna yang membuat suatu halaman situs kemudian terjadi crash, ternyata Google Chrome mempunyai underline download secara otomatis.

Kondisi ini memunculkan sebuah kemungkinan bagi para hacker untuk melakukan pencurian atau perusakan data. Sehingga bisa dikatakan apabila sistem pengamanan pada browser ini masih belum sempurna karena ternyata masih ada celah yang bisa dimasuki oleh hacker tersebut.

3. Terlalu Banyak merekam Data kita Pada server Google

Perlu anda ketahui, google juga punya layanan periklanan yang disebut google Ads. Jadi, terkadang browser ini akan mengumpulkan data kita saat berselancar di internet. Beberapa data tersebut diantaranya adalah umur, minat, aktifitas pencarian browsing dan masih banyak yang lainnya.

Data tersebut digunakan oleh google untuk menampilkan iklan atau memberikan penawaran yang relevan pada perangkat pengguna. Jika anda melihat iklan produk sejenis yang sering di cari tiba-tiba muncul pada perangkat anda, maka itulah hasil olah data yang dikumpulkan oleh google.

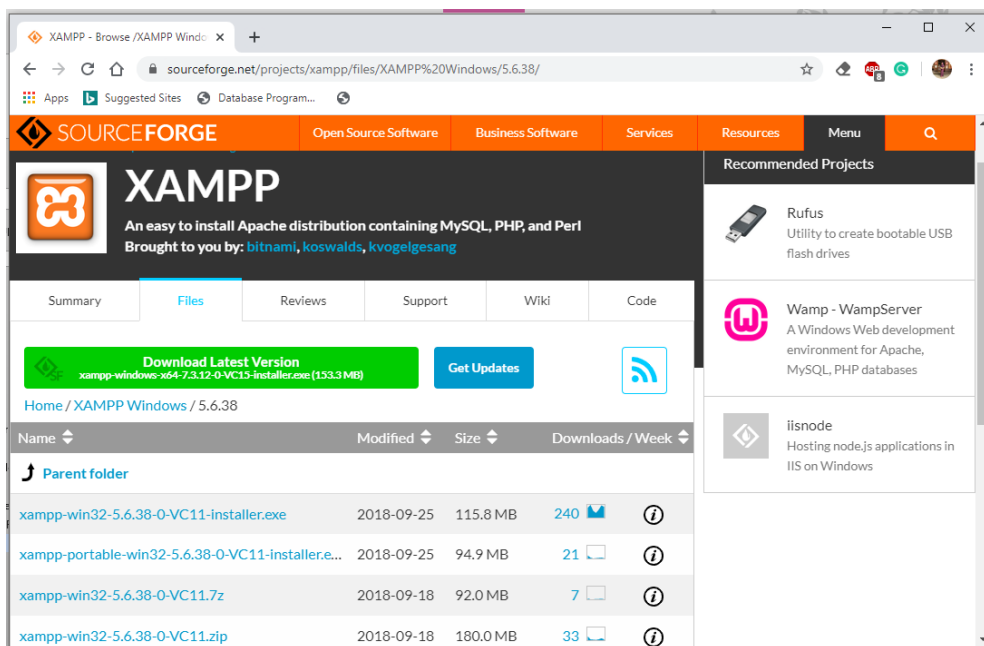
3.2 XAMPP

XAMPP adalah aplikasi yang bersifat *open source* yang bisa digunakan di banyak sistem operasi seperti Windows, Linux, dan Mac OS. XAMPP berfungsi untuk membuat *server* sendiri di PC/Laptop atau biasa disebut dengan istilah *localhost*. XAMPP merupakan kompilasi dari beberapa program. XAMPP sendiri adalah singkatan dari empat kata, yaitu:

- X : *Cross platform/multiplatform* yang bisa berjalan di banyak sistem operasi, seperti Windows, Linux, Mac OS.
- A : Apache, merupakan aplikasi *web server* yang bersifat *open source*.
- M : MariaDB, merupakan aplikasi yang bisa digunakan sebagai *database server* yang berfungsi untuk membuat dan mengolah database pada sebuah aplikasi. MariaDB merupakan pengembangan dari MySQL.
- P : PHP, merupakan bahasa pemrograman yang digunakan untuk membuat *website* dinamis.
- P : Perl, merupakan bahasa pemrograman tingkat tinggi yang cukup tua.

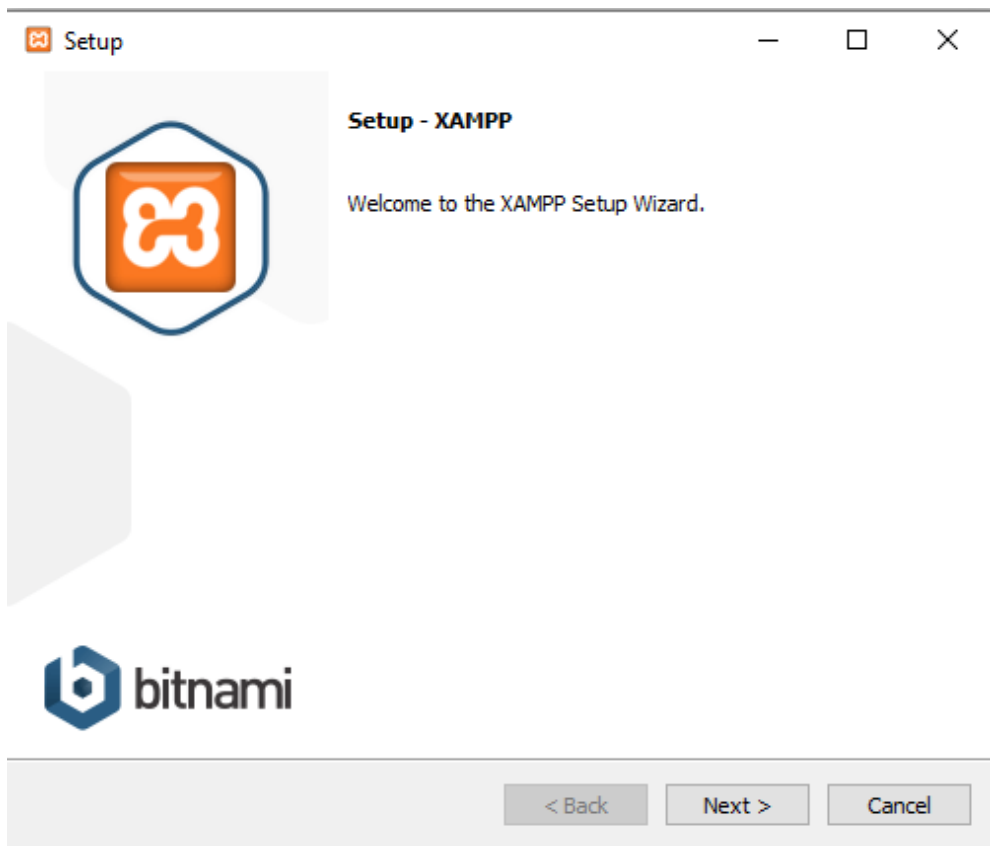
Untuk melakukan instalasi XAMPP sangat mudah, yaitu seperti langkah berikut ini:

1. Download file XAMPP untuk Windows di <https://www.apachefriends.org/index.html>. Dikarenakan di sini menggunakan XAMPP versi 5.6.38, maka nanti untuk *download*-nya akan diarahkan ke <https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/5.6.38/>, seperti pada gambar berikut:



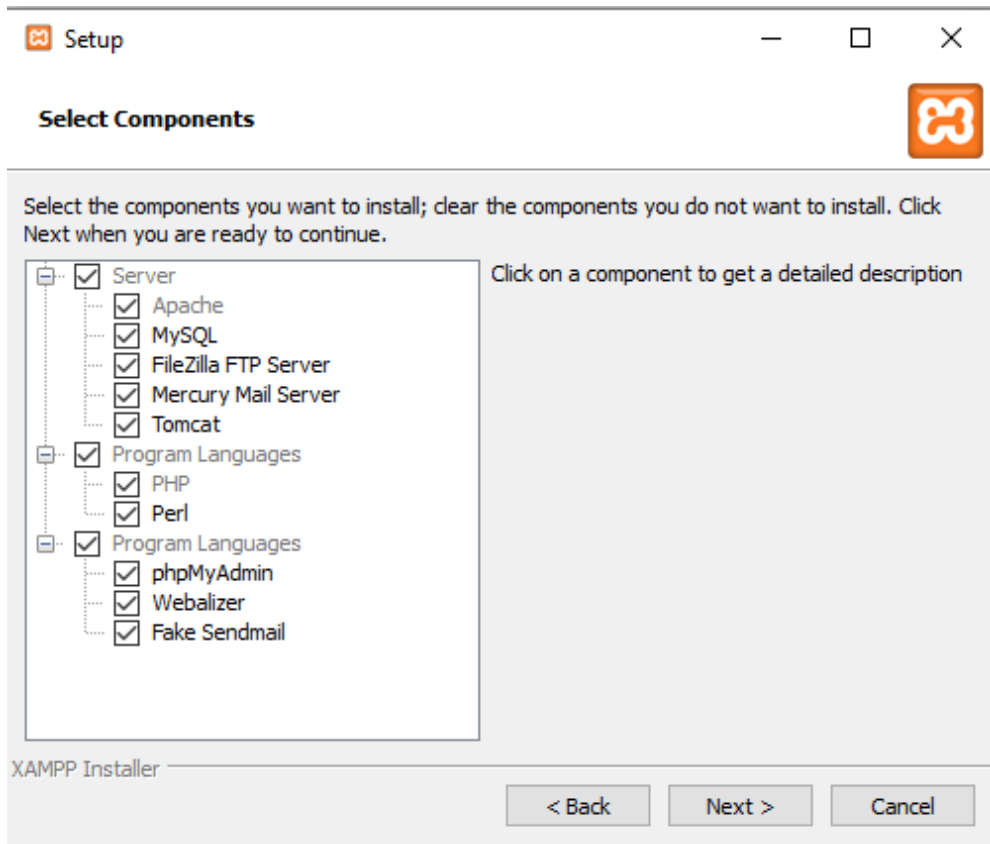
Gambar 3.2 XAMPP

2. *Double click* pada file XAMPP yang sudah terdownload tadi, sampai muncul gambar berikut ini dan klik tombol **Next**.



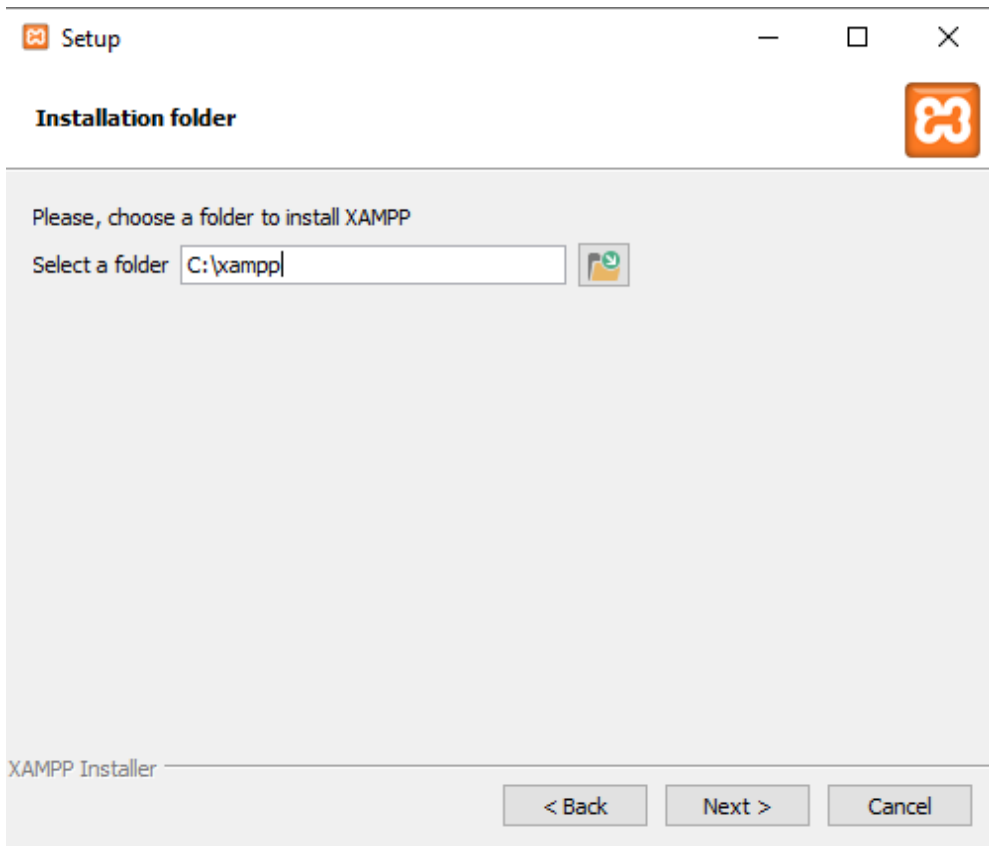
Gambar 3. 3 Setup XAMPP

3. Selanjutnya pilih komponen aplikasi yang akan diinstall. Di sini pilih semua komponen dan klik tombol **Next**.



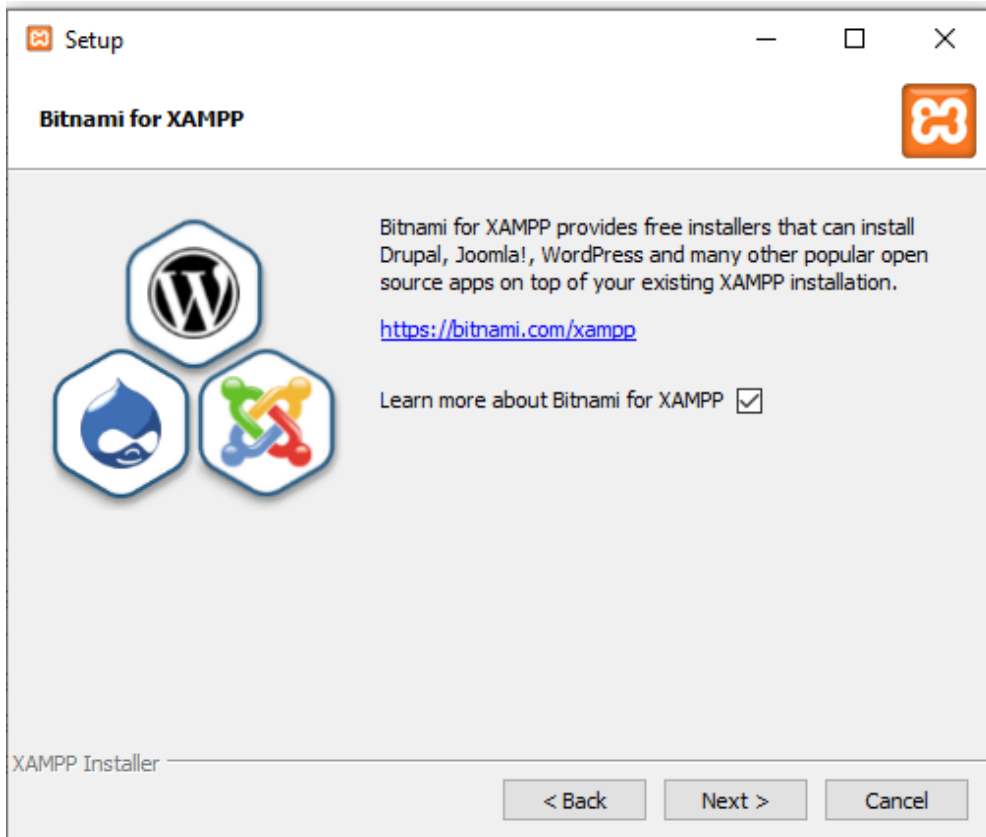
Gambar 3.4 Pilihan Komponen Aplikasi yang akan Diinstall

4. Kemudian tentukan lokasi folder penyimpanan file. Secara *default*, akan diarahkan ke **c:\xampp**. Namun jika ingin menyimpan di folder lain, bisa klik ikon folder dan tentukan folder yang akan digunakan. Jika sudah, klik tombol **Next**.



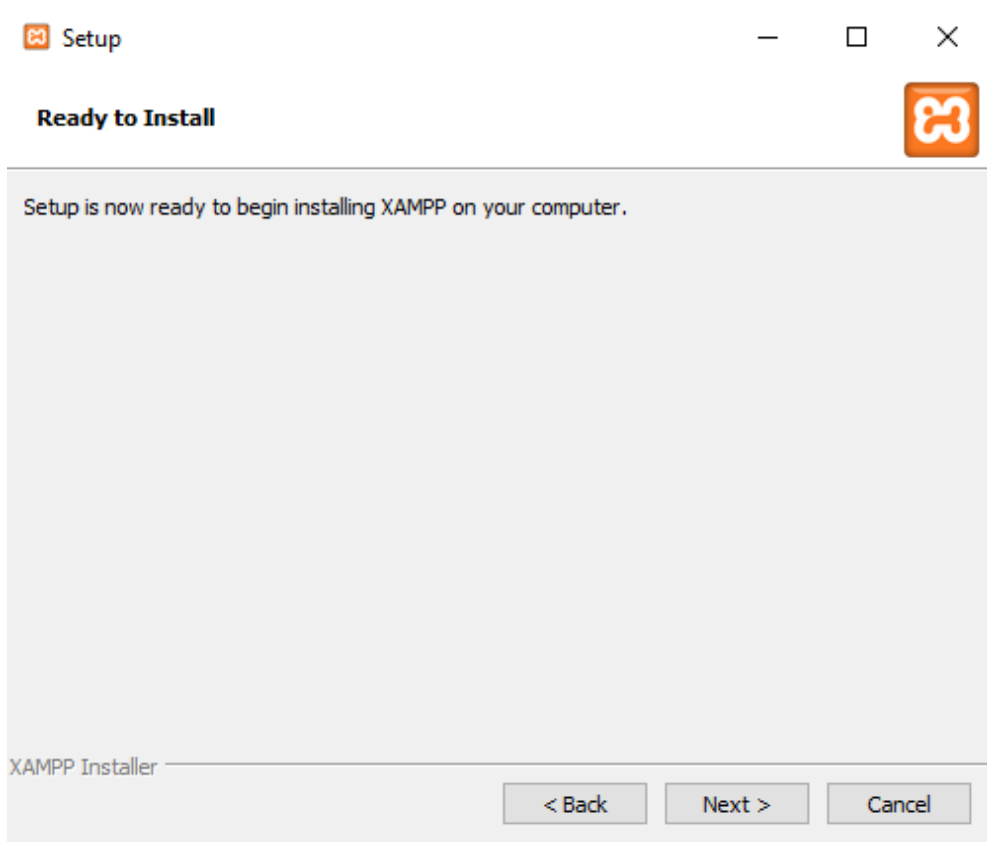
Gambar 3.5 Pemilihan Folder Penyimpanan

5. Pada halaman selanjutnya, akan ada pilihan apakah Anda ingin menginstal Bitnami untuk XAMPP, dimana nantinya dapat Anda gunakan untuk install WordPress, Drupal, dan Joomla seccara otomatis.



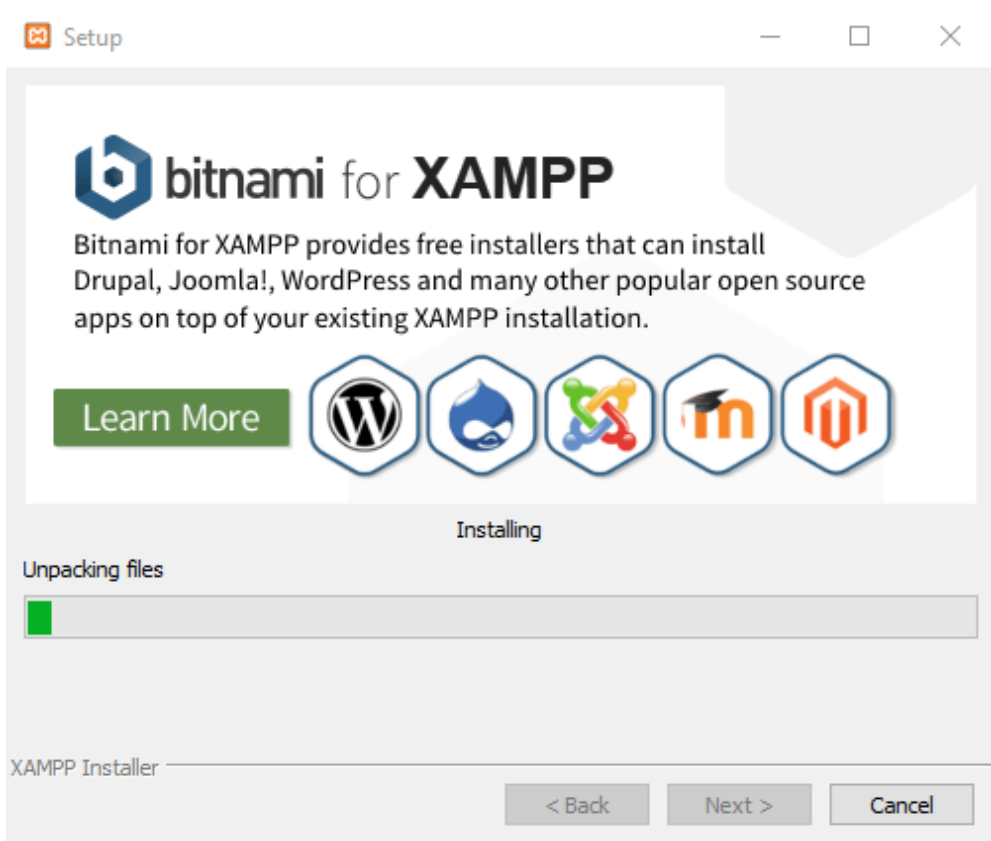
Gambar 3. 6 Bitnami for XAMPP

6. Jika muncul tampilan seperti berikut bisa langsung klik tombol **Next** untuk melanjutkan proses instalasi.



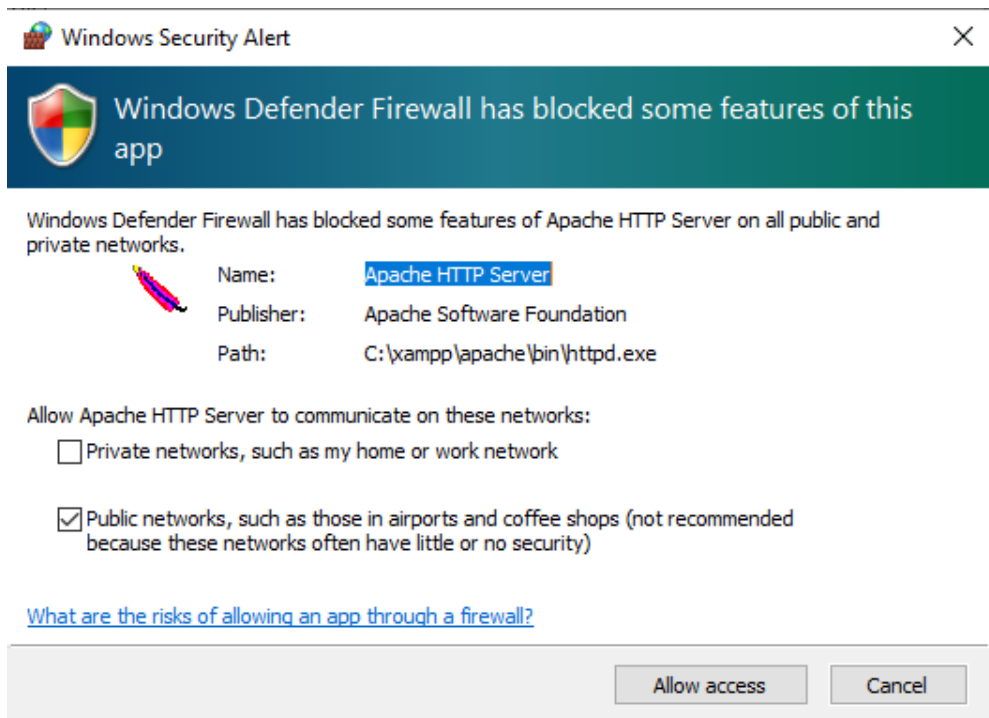
Gambar 3.7 Persiapan Instalasi

7. Pada langkah ini proses instalasi XAMPP dimulai dan tunggu beberapa saat hingga proses instalasi selesai.



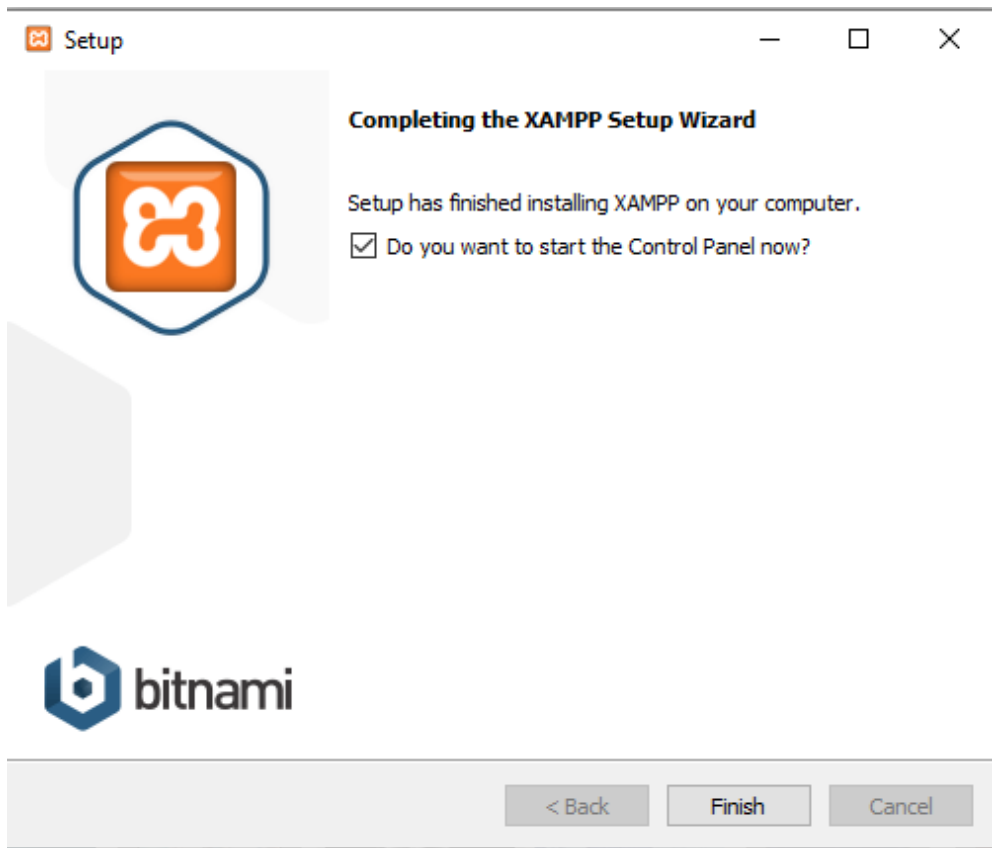
Gambar 3. 8 Proses Instalasi XAMPP

8. Jika muncul tampilan seperti berikut, pilih **Public Network** dan klik tombol **Allow access**.



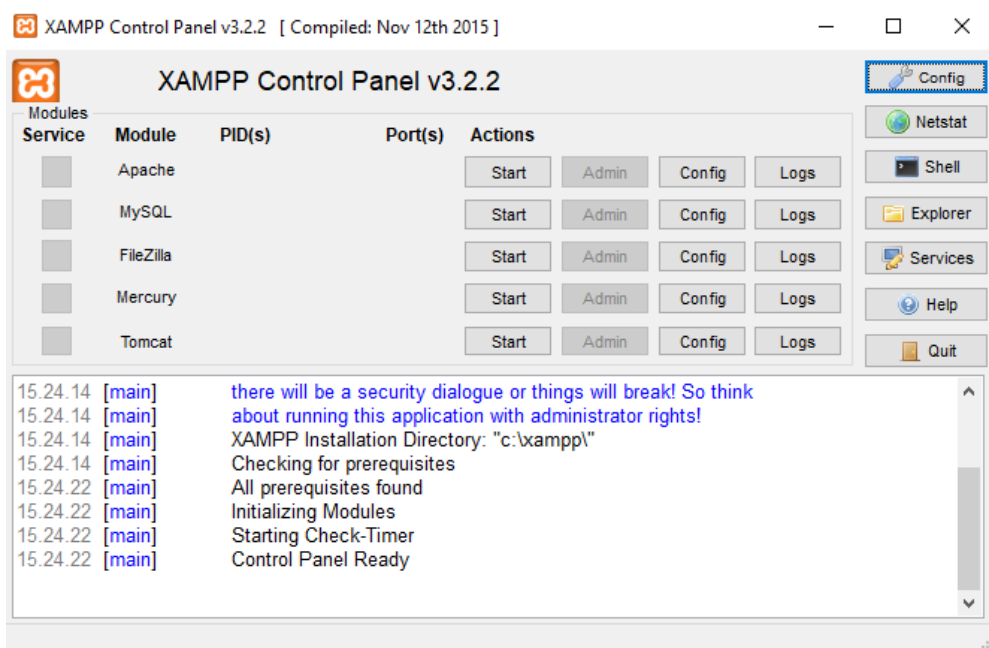
Gambar 3. 9 Windows Defender Firewall

9. Jika sudah muncul seperti gambar berikut, klik tombol **Finish** untuk menyelesaikan instalasi.



Gambar 3.10 Instalasi Selesai

10. Berikut tampilan control panel XAMPP.



Gambar 3. 11 XAMPP Control Panel

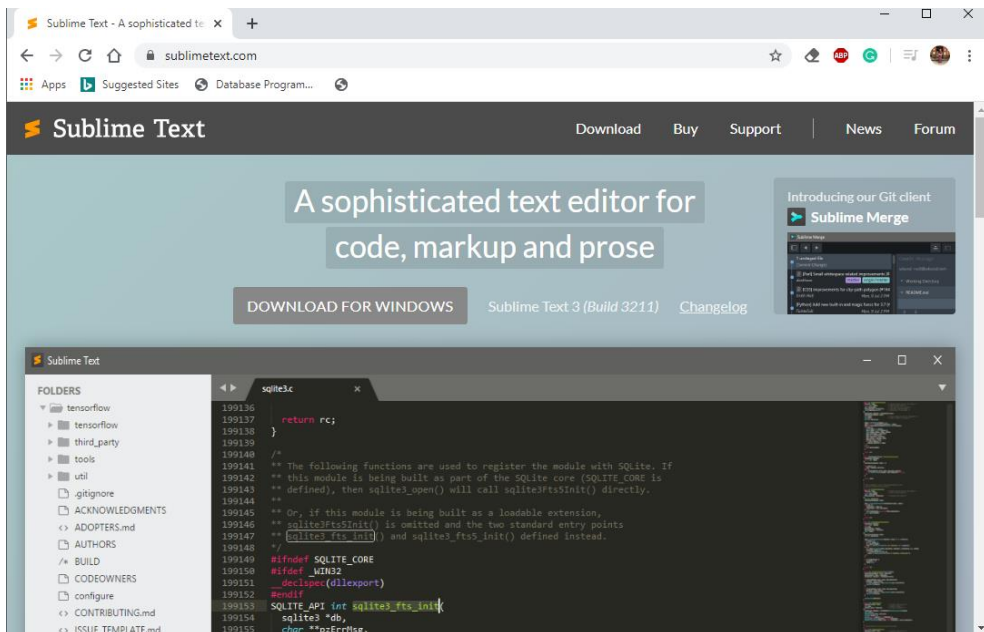
Dengan melakukan instalasi XAMPP maka tidak perlukan lagi untuk instalasi dan konfigurasi *web server* Apache, PHP dan MariaDB secara manual. XAMPP sendiri akan menginstalasi dan mengkonfigurasi secara otomatis.

3.3 Sublime Text 3

Sublime Text 3 adalah aplikasi *text editor* untuk berbagai Bahasa pemrograman, salah satunya adalah PHP. *Text editor* aplikasi yang digunakan *programmer* untuk membuat program komputer dan mengedit *source code* dari Bahasa pemrograman. *Sublime Text 3* bisa berjalan di berbagai sistem operasi, seperti Windows, Linux, dan Mac OS.

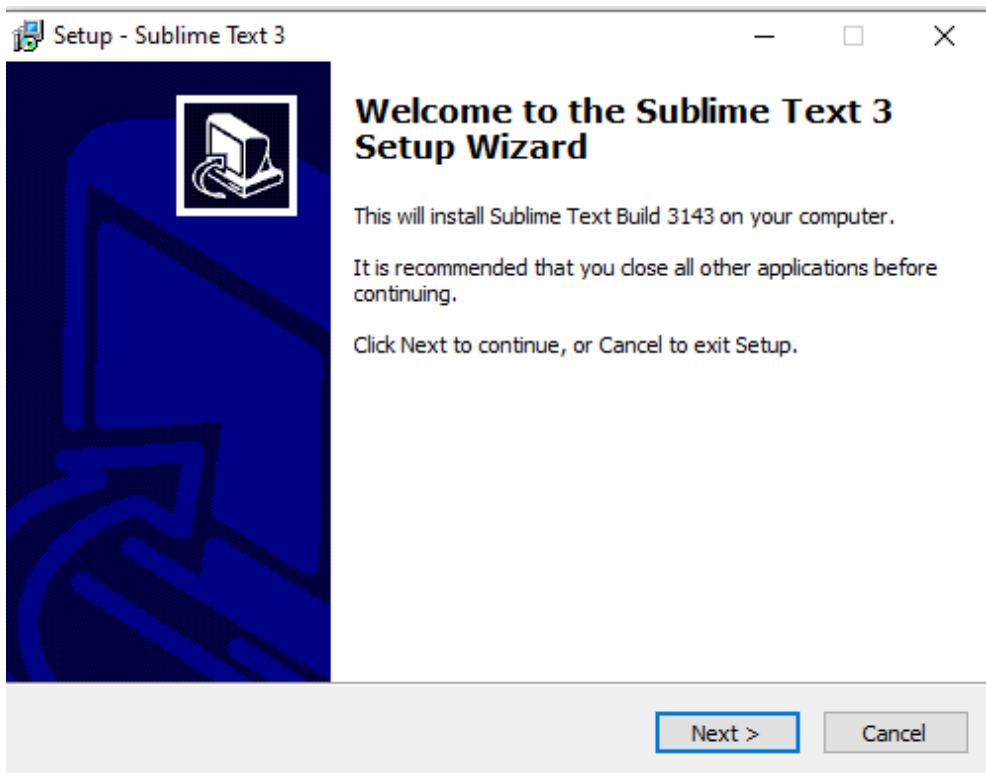
Untuk melakukan instalasi *Sublime Text 3* bisa ikuti langkah seperti di berikut ini:

1. Download file Sublime Text 3 di <https://www.sublimetext.com/>



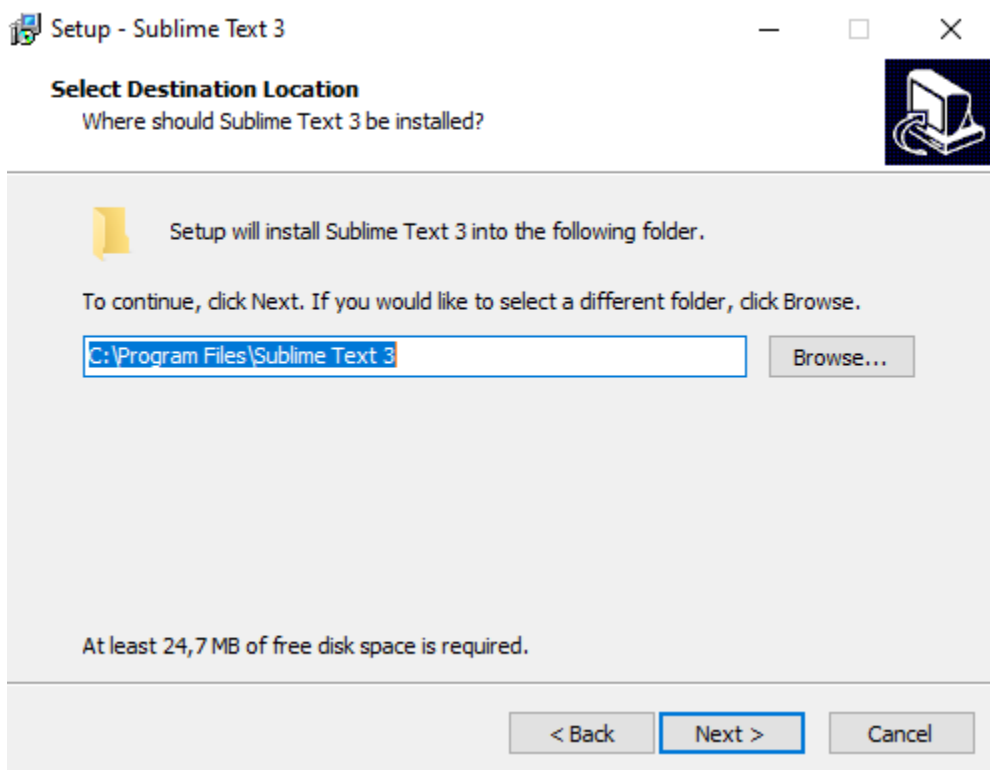
Gambar 3.12 Sublime Text

2. Double click pada file yang sudah didownload tadi. Jika sudah muncul seperti gambar berikut, pilih tombol **Next**.



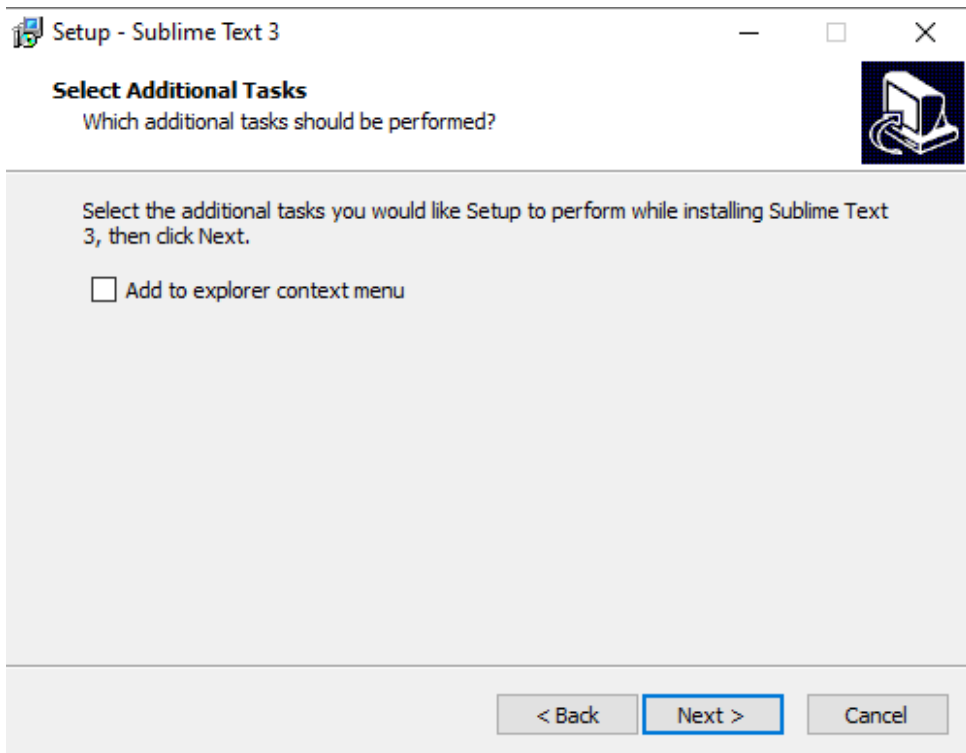
Gambar 3.13 Setup Sublime Text 3

3. Kemudian tentukan lokasi folder penyimpanan file. Secara *default*, akan diarahkan ke **c:\Program Files\Sublime Text 3**. Namun jika ingin menyimpan di folder lain, bisa klik tombol **Browse** dan tentukan folder yang akan digunakan. Jika sudah, klik tombol **Next**.



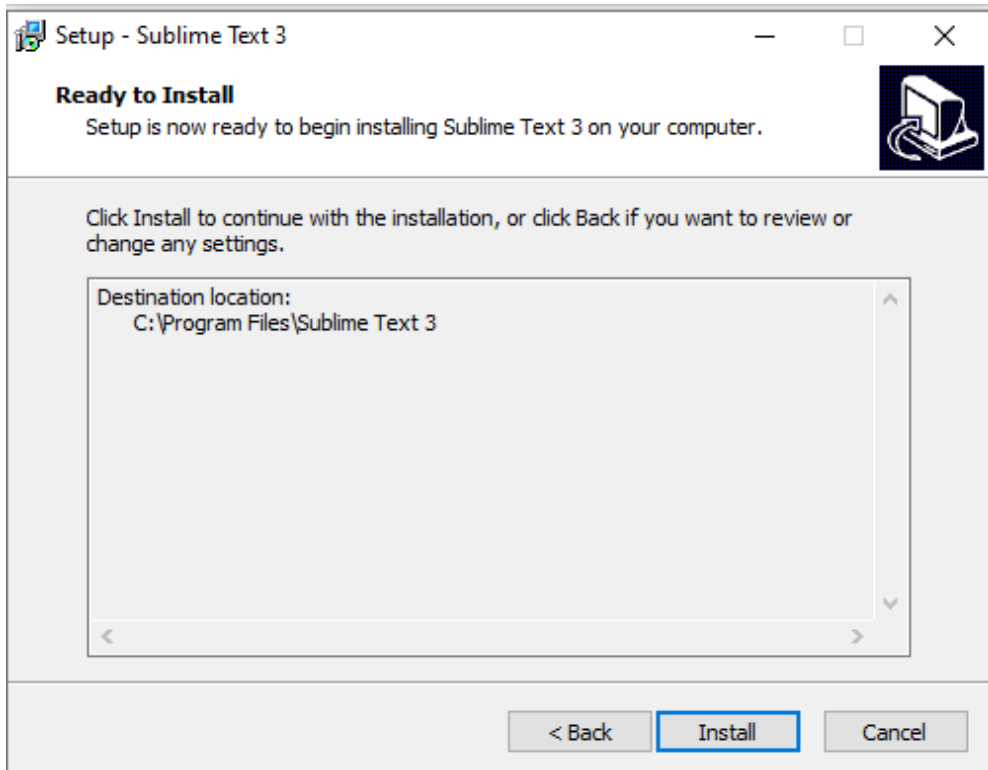
Gambar 3.14 Pemilihan Folder Penyimpanan

4. Pada langkah ini boleh langsung klik tombol **Next**.



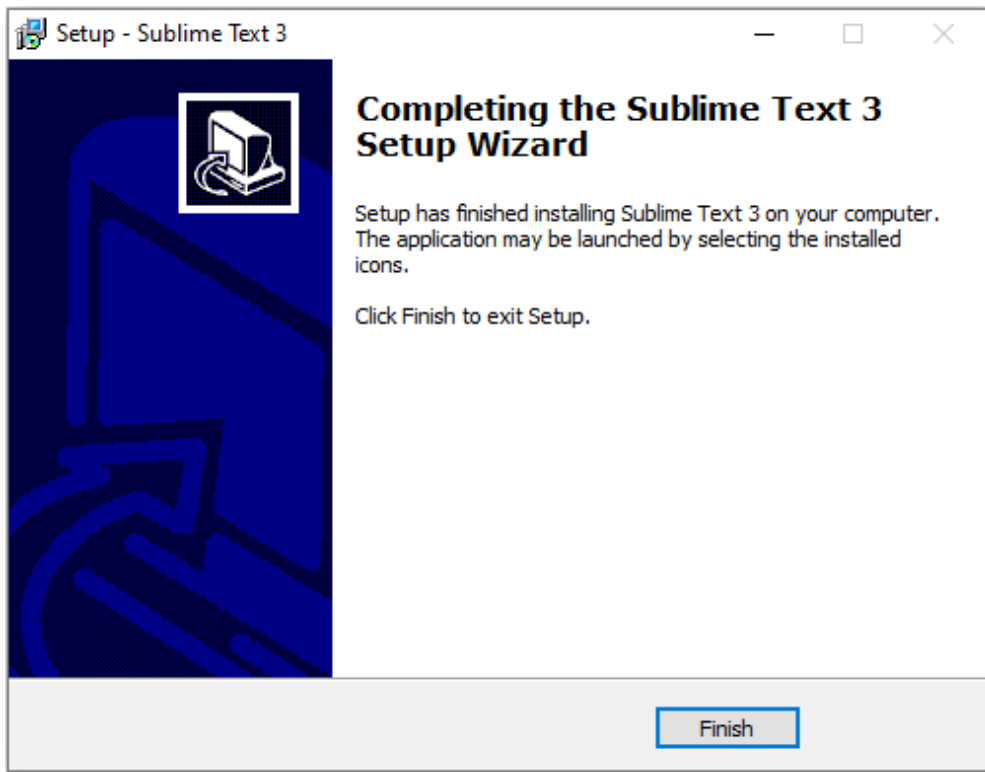
Gambar 3. 15 Pilihan Tugas Tambahan

5. Pada langkah ini memverifikasi bahwa folder tujuan yang dipilih sudah benar dan sesuai dengan keinginan Anda, klik tombol **Install** untuk melanjutkan proses instalasi.



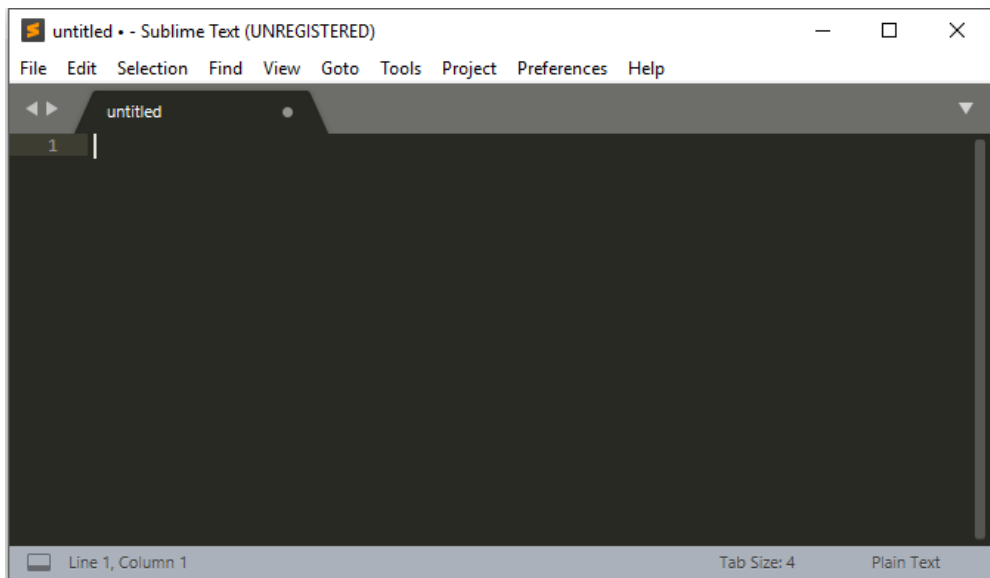
Gambar 3.16 Instalasi Sublime Text 3

6. Jika instalasi sudah selesai dan muncul halaman seperti gambar berikut, maka klik tombol **Finish** untuk menyelesaikan semua proses.



Gambar 3.17 Instalasi Selesai

7. Tampilan Sublime Text 3 yang siap digunakan untuk membuat program.



Gambar 3. 18 Tampilan Sublime Text 3

Terdapat beberapa shortcut yang bisa digunakan di Sublime Text 3 untuk mempermudah dalam proses pengkodean di antaranya:

Tabel 3. 1 Shortcut di Sublime Text

Shortcut	Fungsi
CTRL+C	Untuk menyalin
CTRL+X	Untuk memotong
CTRL+L	Untuk memilih baris (ulangi untuk memilih baris berikutnya)
CTRL+D	Untuk memilih kata
CTRL+M	Untuk menuju ke penutupan braket code atau ke pembukaan braket code
CTRL+KK	Untuk menghapus ke akhir baris
CTRL+U	Untuk membatalkan perintah sebelumnya
CTRL+J	Untuk menggabungkan baris ke akhir baris saat ini
CTRL+/ CTRL+Y	Untuk komentar atau membatalkan komentar pada baris saat ini
CTRL+Space	Untuk memilih saran auto-complete berikutnya
CTRL+V	Untuk paste
CTRL+A	Untuk pilih semua
CTRL+Z	Untuk membatalkan
CTRL+KL	Untuk mengubah pilihan menjadi huruf kecil
CTRL+KU	Untuk mengubah pilihan menjadi huruf besar
CTRL+F	Untuk mencari data
CTRL+H	Untuk mengganti data

CTRL+P	Untuk membuka file dengan cepat sesuai dengan nama di project
CTRL+O	Untuk membuka file

Sublime Text 3 juga memiliki keunggulan-keunggulan di fiturnya yang dapat membantu dan mempermudah pengembang web menggunakan Sublime Text 3. Berikut adalah keunggulan-keunggulan dari fitur yang ada di Sublime Text 3:

- **Settings**

Pengaturan dilakukan dengan membuka file `.sublime-settings` dan mengedit isinya. Perubahan langsung terjadi pada saat file tersebut disave. Untuk yang bukan programmer, mungkin akan kesulitan karena harus mengedit data berformat JSON.

- **Color Scheme/Theme**

Sublime Text secara default menggunakan warna background gelap, sedangkan Notepad++ menggunakan background terang. Tapi tentu saja kita bisa merubahnya dengan mudah. Gambar di atas adalah screenshot Sublime Text menggunakan warna terang dan Notepad++ menggunakan warna gelap.

- **Goto Anything**

Dengan Goto Anything, kita bisa membuka file di dalam project dengan cepat, tinggal tekan `Ctrl + P` kemudian ketik nama filenya. Untuk mencari nama file tidak harus mengetik secara tepat, karena Sublime Text menggunakan algoritma fuzzy untuk searching nya.

- **Command Palette**

Sublime Text memiliki tampilan yang lebih simple dan sangat minim menu, bahkan tidak ada toolbar sama sekali. Kebanyakan perintah-perintahnya bisa kita akses menggunakan Shortcut, atau kalau belum hafal shortcut kita bisa mengakses Command Palette (tekan `Ctrl +`

Shift + P) kemudian cari perintah yang kita inginkan (menggunakan fuzzy search). Sebagai contoh, saya ingin membuat teks menjadi uppercase, maka teks tersebut saya blok dulu, kemudian masuk Command Palette dan ketik “case” atau “upc” maka akan muncul perintah untuk uppercase beserta keterangan Shortcut nya.

- **Package/Plugin**

Sublime Text juga memiliki sangat banyak package. Untuk menginstall package kita bisa menjalankan Package Manager kemudian akan muncul list package, kita tinggal mencari (tentu saja menggunakan fuzzy search juga) package yang diinginkan.

Kelebihan menggunakan package manager adalah package akan otomatis terupdate jika ada versi baru. Sublime Text juga menyediakan plugin API untuk kita yang ingin membuat package sendiri menggunakan bahasa python.

- **Multiple Selections/Multi-editing**

Ini adalah salah satu fitur unggulan di Sublime Text, di mana kita bisa meletakkan cursor di beberapa tempat (menggunakan Ctrl + click), kemudian mengedit secara bersamaan.

Di Notepad++ juga ada fitur ini, tapi kita harus mengaktifkan dulu melalui Settings -> Preferences -> Editing -> Multi-Editing Settings.

Fitur Multiple Selections di Sublime Text lebih baik karena bisa untuk mempercepat editing teks yang sama. Misal ingin mengedit sebuah kata, letakkan kursor pada kata tersebut kemudian tekan Ctrl + D untuk menyeleksi satu persatu semua kata yang sama, atau tekan Alt + F3 untuk langsung menyeleksi semua kata tersebut.

- **Split Editing**

Jika anda memiliki monitor yang lebar maka kedua editor ini sangat cocok karena sama-sama memiliki fitur untuk Split Editing (membuka 2 atau lebih file secara berdampingan).

- **Auto-completion**

Sublime Text memiliki auto complete untuk beberapa bahasa yang saya pakai seperti PHP, CSS, Javascript. Fitur ini juga mendukung fuzzy search sehingga tidak harus mengetik secara tepat.

Notepad++ juga memiliki auto complete, namun tidak senyaman Sublime Text dan harus diaktifkan dulu melalui Settings -> Preferences -> Backup/Auto-Completion.

- **Minimap/Document Map**

Sublime Text memiliki Minimap, semacam versi mini dari file untuk mempermudah melihat file secara keseluruhan.

Di Notepad++ juga ada fitur seperti ini dengan nama Document Map. Untuk mengaktifkannya dari View -> Document Map.

- **Goto Definition**

Ini adalah fitur baru Sublime Text 3 (di ST2 blm ada). Fitur ini sangat membantu menemukan function/class di dalam project, caranya dengan meletakkan cursor di nama function/class kemudian tekan F12, maka file yang berisi definisi function/class tersebut akan terbuka. Untuk melihat list semua function/class bisa menggunakan Goto Symbol in Project (tekan Ctrl + Shift + R).

- **Instant Project Switch**

Untuk mempermudah bekerja di Sublime Text, sebaiknya kita membuat project terlebih dahulu. Caranya, buka jendela Sublime Text baru, kemudian folder tempat file-file yang akan kita edit kita drag ke dalam jendela Sublime Text, kemudian simpan project tersebut melalui Project -> Save Project As. Sublime Text akan mengindex semua file di dalam folder tersebut supaya fitur Goto bisa bekerja. Proses indeks pun berjalan cepat, sebagai contoh, script-script WordPress bisa diindex semua dalam waktu sekitar 20 detik saja.

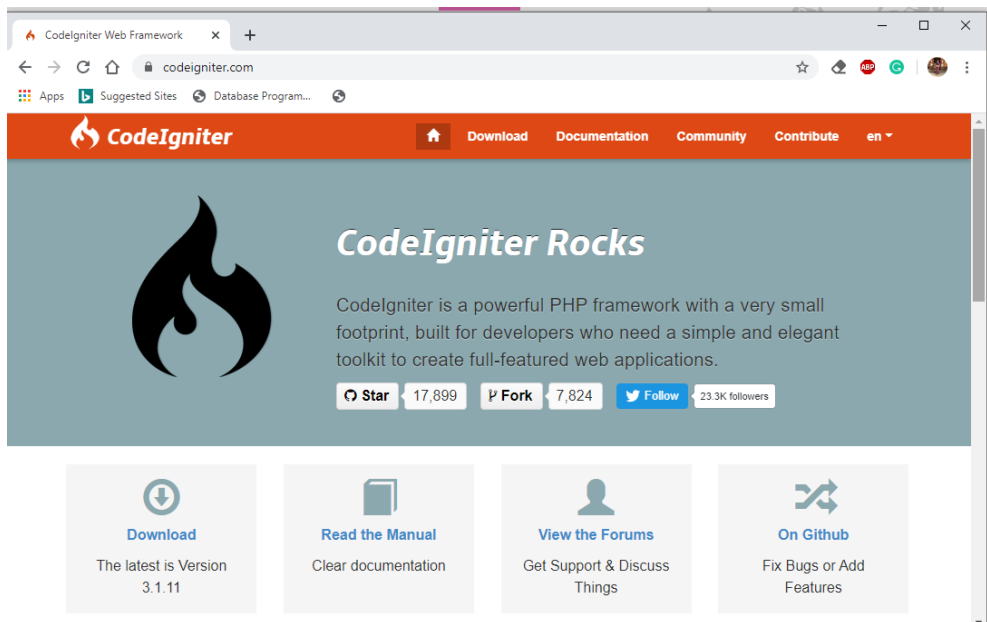
Dengan membuat project, maka kita bisa berpindah-pindah project dengan cepat, dengan menekan Ctrl + Alt + P, lalu pilih project dari list project.

Saat kita berpindah project atau saat Sublime Text ditutup, kita tidak akan diganggu dengan dialog konfirmasi untuk menyimpan file. Semua file yang belum disave akan kembali lagi (tidak hilang) saat project dibuka lagi.

3.4 CodeIgniter

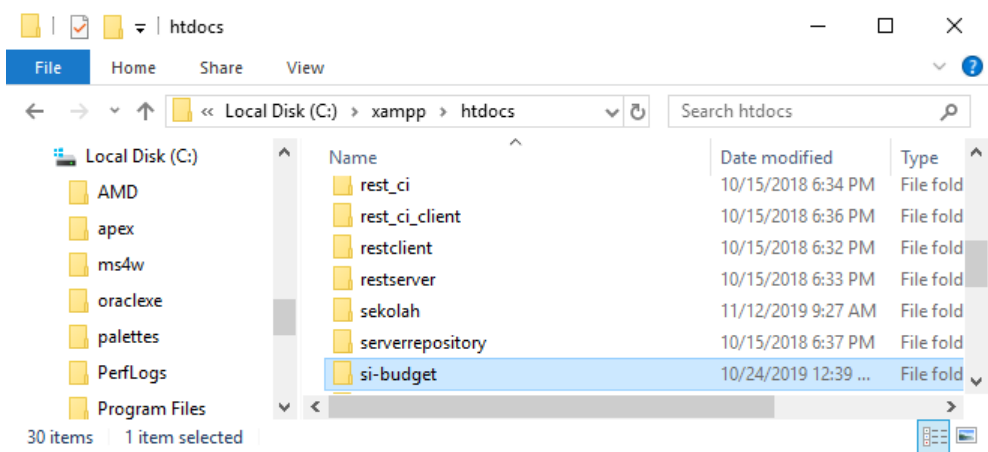
Untuk melakukan instalasi CodeIgniter sangat mudah, seperti langkah berikut ini:

1. Download file CodeIgniter di <https://codeigniter.com/>.



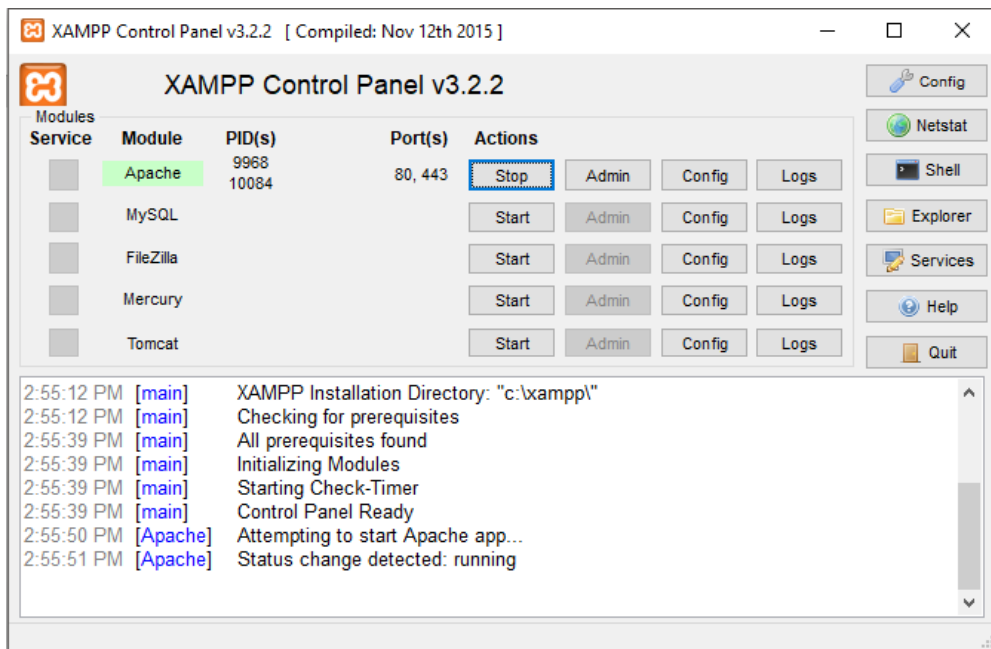
Gambar 3.19 CodeIgniter

2. Ekstrak file zip yang sudah didownload ke dalam folder **C:\xampp\htdocs** dan ubah namanya menjadi **si-budget**.



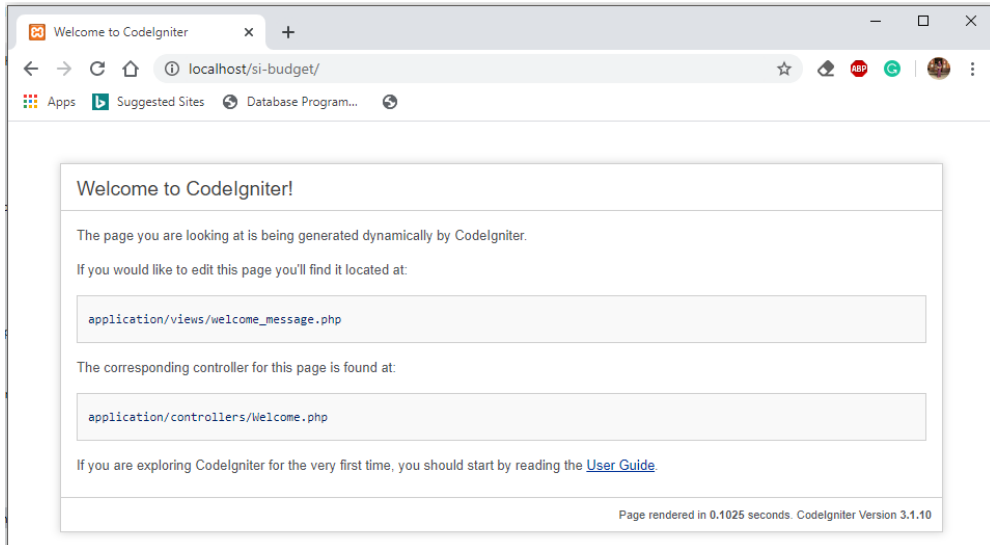
Gambar 3.20 Direktori Penyimpanan Folder si-budget

3. Akifikan web server yang ada di XAMPP dengan cara klik Start yang ada di sebelah apache, seperti gambar berikut.



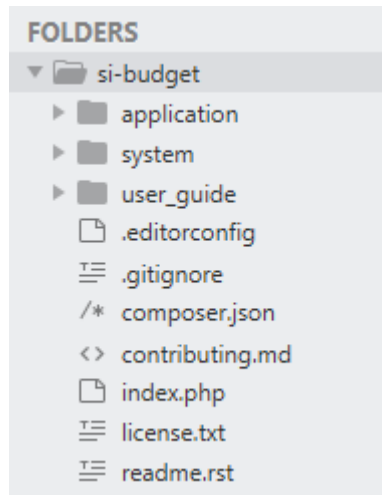
Gambar 3.21 Start Apache

4. Buka web browser dan ketikkan **localhost/si-budget/** pada URL, jika instalasi CodeIgniter berhasil maka akan muncul halaman “Welcome to CodeIgniter!”, seperti gambar berikut.



Gambar 3.22 Tampilan Awal Setelah Instalasi CodeIgniter

Untuk mengetahui struktur direktori dari CodeIgniter, silahkan buka folder **si-budget** melalui sublime text. Di dalam CodeIgniter terdapat dua direktori yang penting, yaitu **application** dan **system**.



Gambar 3.23 Direktori CodeIgniter

Berikut penjelasan mengenai direktori yang ada di CodeIgniter:

1. **application** berisi semua kode program aplikasi. Di dalam direktori inilah kita akan menulis semua kode aplikasi kita.
 - a. **cache** berisi cache dari aplikasi.
 - b. **config** berisi konfigurasi aplikasi.
 - **autoload.php** tempat kita mendefinisikan autoload;
 - **config.php** konfigurasi aplikasi;
 - **constants.php** berisi konstanta;
 - **database.php** konfigurasi database aplikasi;
 - **doctype.php** berisi definisi untuk doctype HTML;
 - **foreign_chars.php** berisi karakter dan simbol;
 - **hooks.php** berisi konfigurasi hooks;
 - **index.html** untuk mencegah direct access;

- **memcached.php** untuk berisi konfigurasi untuk memcached;
 - **migration.php** konfigurasi untuk migrasi;
 - **mimes.php** berisi definisi tipe file;
 - **profiler.php** konfigurasi untuk profiler;
 - **routers.php** tempat kita menulis route aplikasi;
 - **smileys.php** berisi kode untuk emoji;
 - **user_agents.php** berisi definisi untuk user agents.
- c. **controller** berisi kode untuk controller.
 - d. **core** berisi kode untuk custom core.
 - e. **helpers** berisi fungsi-fungsi helper.
 - f. **hooks** berisi kode untuk script hook.
 - g. **language** berisi string untuk bahasa, apabila web mendukung multibahasa.
 - h. **libraries** berisi library.
 - i. **logs** berisi logs dari aplikasi.
 - j. **models** berisi kode untuk model.
 - k. **thrid_party** berisi library dari pihak ketiga.
 - l. **views** berisi kode untuk view.
 - m. **index.html** file html untuk mencegah direct access.
2. **system** berisi kode-kode inti dari Codeiniter. Jangan mengubah apapun di dalam direktori ini. Jika kita ingin upgrade versi, kita cukup me-replace direktori ini dengan yang baru.
 3. **tests** berisi kode untuk melakukan unit testing.
 4. **user_guide** berisi dokumentasi codeigniter. Kita bisa menghapus direktori ini saat web sudah jadi.
 5. **.editor_config** berisi konfigurasi untuk teks editor.
 6. **.gitignore** berisi daftar file dan folder yang akan diabaikan oleh Git.

7. **comspoer.json** adalah file yang berisi keterangan project dan keterangan library yang digunakan. File ini dibutuhkan oleh composer.
8. **contributing.md** adalah file yang berisi penjelasan cara berkontribusi di proyek CI. Kita bisa menghapus file ini, apabila web sudah jadi.
9. **license.txt** adalah file yang berisi keterangan lisensi dari CI.
10. **readme.rst** sama seperti file contributing.md file ini berisi penjelasan dan informasi tentang project CI. Kita juga bisa menghapus file ini saat web sudah selesai.
11. **index.php** adalah file utama dari CI. File yang akan dibuka pertamakali saat kita mengakses web.

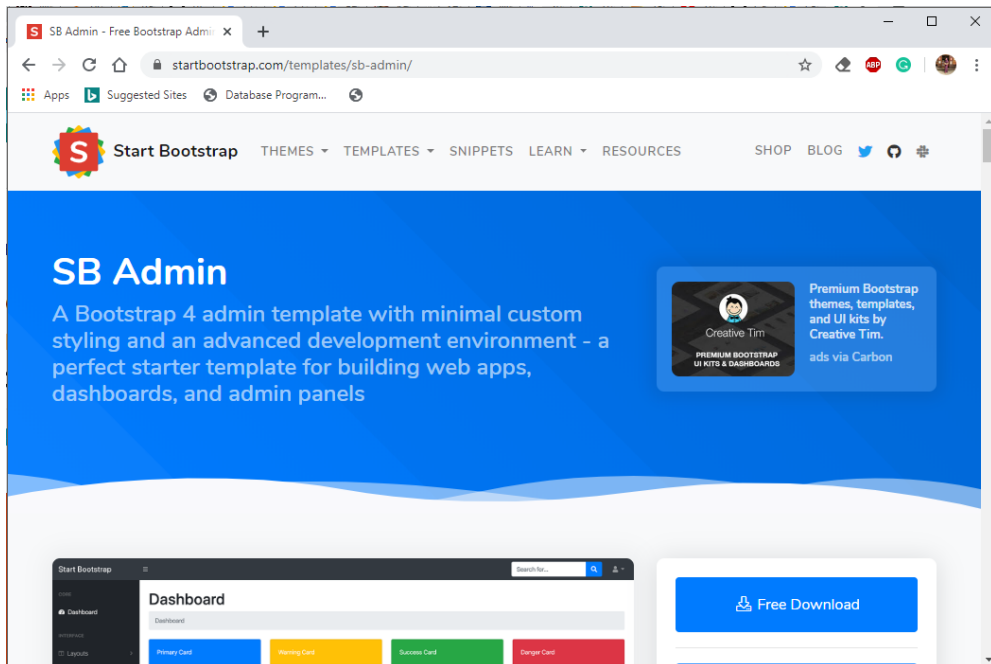
3.5 Bootstrap

Bootstrap merupakan software yang bersifat open source yang digunakan untuk mempermudah membangun tampilan web yang responsive dengan HTML, CSS, dan JS. Di dalam bootstrap terdapat kumpulan class yang siap pakai.

Bootstrap sangat populer di kalangan pengembang web. Bootstrap, awalnya bernama Twitter Blueprint, yang dikembangkan oleh Nirav Panchal dan Jacob Thornton pada pertengahan 2010 di Twitter sebagai *framework* untuk mendorong konsistensi di perangkat internal yang sesuai.

Berikut langkah-langkah menggunakan template web dari bootstrap:

1. Download template untuk tampilan web admin di <https://startbootstrap.com/templates/sb-admin/>.



Gambar 3.24 Bootstrap

2. Ekstrak file yang sudah didownload tadi dan pindahkan ke dalam folder project.
3. Memanggil file CSS dan JS yang ada di direktori dengan menggunakan PHP.

```
<link href="php echo<br/base_url('nama_direktori/css/bootstrap.min.css') ?>"  
rel="stylesheet">  
  
<script src="php echo<br/base_url('nama_direktori/js/bootstrap.bundle.min.js')  
?>"></script>
```

Berikut merupakan cara menggunakan Bootstrap:

1. Membuat Tabel Dengan Bootstrap

Bootstrap memberikan Anda kemudahan untuk membuat tabel tanpa harus mengetikkan berbaris-baris kode program. Anda hanya perlu memanggil class-class yang sudah ada di dalam file Bootstrap.

Untuk menggunakan class dengan kombinasi class lainnya, gunakan class table ditambah dengan class lain. Contoh penggunaannya seperti di bawah ini:

```
# untuk membuat tabel striped  
<table class="table table-striped">
```

Atau

```
# untuk membuat tabel hover<table class="table table-hover">
```

Di dalam file Bootstrap sudah terdapat beberapa class yang didesain untuk kebutuhan pembuatan tabel, antara lain “.table”, “.table-striped”, “.table-bordered”, dan “.table-hover”.

- “.table”: class yang digunakan untuk membuat dan mendefinisikan tabel biasa/standar. Class ini hanya menambahkan lapisan yang kecil dan pembatas secara horizontal. Class ini menggunakan baris kode `<table class="table">`.
- “.table-striped”: class yang digunakan untuk membuat table dengan gaya yang zebra. Jadi warna setiap baris berselang seling. Class ini menggunakan baris kode `<table class="table table-striped">`.

- “.table-bordered”: merupakan class yang digunakan untuk menampilkan border di dalam table. Jadi keseluruhan table akan dipasang sebuah border. Class ini menggunakan baris kode `<table class="table table-bordered">`.
- “.table-hover”: dapat Anda manfaatkan untuk memberikan efek hover pada sebuah row. Efek ini akan berjalan ketika Anda meletakkan mouse di row yang diberikan class ini. Class ini menggunakan baris kode `<table class="table table-hover">`.

Selain kumpulan class untuk mendesain table, terdapat beberapa class pewarnaan row tabel dan data menggunakan Bootstrap, yaitu “.success”, “.danger”, “.info”. “.warning”, dan “.active”.

Untuk menggunakan class ini, selipkan di antara tag tr, seperti ini:

```
... <tr class="info">    <td>Info</td>    <td>Khafid</td>
<td>july@example.com</td>    </tr>...
```

Menggunakan kumpulan class ini, Anda dapat mendesain tabel yang menarik dan mudah dibaca oleh pengunjung website.

2. Mengatur Tampilan Gambar Dengan Bootstrap

Anda juga dapat mendesain gambar menggunakan Bootstrap. Membuat gambar yang responsive, berbentuk round, lingkaran, dan juga thumbnail sangat mudah menggunakan framework ini.

Class “.img-responsive”, “.img-rounded”, “.img-circle”, dan “.img-thumbnail” adalah kumpulan class yang digunakan untuk memanipulasi gambar di Bootstrap. Kembangkan desain yang maksimal dan hadirkan gambar menarik di website Anda.

- “.img-responsive”: class ini memungkinkan Anda membuat gambar yang dapat dibuka di berbagai macam resolusi layar. Class .img-responsive membuka ukuran gambar dapat menyesuaikan dengan ukuran elemen inti. Anda dapat melihat efeknya ketika melakukan resize pada browser.

```

```

- “.img-rounded”: ketika Anda ingin membuat gambar dengan sisi melengkung, gunakan class ini, maka Anda akan memiliki gambar yang melengkung di bagian sisinya.

```

```

- “.img-circle”: terkadang ingin gambar yang berbentuk lingkaran, gunakan class ini dan gambar akan tampil dengan bentuk yang melingkar.

```

```

- ".img-thumbnail": class ini digunakan untuk membuat gambar thumbnail dengan Bootstrap.

```

```

3. Membuat Pesan Peringatan (Alert) Dengan Bootstrap

Bootstrap menyediakan class khusus untuk menampilkan alert dengan sangat sederhana dan mudah. Bagaimana tidak, Anda tinggal memanggil sebuah class di Bootstrap menggunakan satu perintah saja. Supaya tampilan alert dapat muncul, cukup panggil class “alert” untuk mendefinisikan tampilan alert di Bootstrap. Supaya tampilannya lebih bervariasi, tambahkan class lain.

Untuk membuat class ini caranya cukup mudah. Cukup buat tag <div> dengan isi peringatan yang ingin dibuat. Kemudian di tag pembuka <div> buat class alert untuk mendeklarasikan alert, berikut contohnya:

```
...<div class="container"> <h2>Alerts</h2> <div class="alert alert-  
success"> <strong>Success!</strong> Alert box ini dapat digunakan  
untuk mengindikasikan aksi sukses atau positif. </div> <div  
class="alert alert-info"> <strong>Info!</strong> Alrt box ini dapat  
mengindikasikan informasi yang netral. </div> <div class="alert  
alert-warning"> <strong>Warning!</strong> Alert box ini dapat  
untuk mengindikasikan pesan warning dan mungkin butuh perbaikan.  
</div> <div class="alert alert-danger"> <strong>Danger!</strong>  
Alert box ini dapat mengindikasikan pesan penting dan kemungkinan  
efek buruk. </div></div>...
```

Terdapat beberapa pilihan class untuk menampilkan pesan peringatan di Bootstrap, seperti “.alert-success”, “.alert-info”, “.alert-warning”, atau “.alert-danger”. Kumpulan class ini akan membuat tampilan alert Anda lebih bervariasi.

- .alert-success: Class alert box ini dapat digunakan untuk mengindikasikan aksi sukses atau positif.

Anda dapat memanfaatkan class ini ketika ada input yang berhasil. Misalnya ketika pengguna selesai mengisi data dan mengirimkannya.

Untuk menggunakan class ini cukup masukkan kode `<div class="alert alert-success">`.

- `.alert-info`: Class alert box ini dapat mengindikasikan informasi yang netral.

Class ini dapat Anda gunakan ketika ingin memberikan informasi tambahan kepada pengguna. Misalnya memberikan informasi kegunaan tombol atau fungsi tertentu dan cara penggunaannya.

Untuk menggunakan class ini cukup masukkan kode `<div class="alert alert-info">`.

- `.alert-warning`: Class alert box ini dapat untuk mengindikasikan pesan warning dan mungkin butuh perbaikan. Class ini dapat Anda gunakan ketika ingin memberikan informasi yang sedikit mendesak atau penting. Misalnya ketika terjadi error pada suatu konfigurasi atau kesalahan pada saat pengisian data.

Untuk menggunakan class ini cukup masukkan kode `<div class="alert alert-warning">`.

- `.alert-danger`: Class alert box ini dapat mengindikasikan pesan penting dan kemungkinan efek buruk.

Class ini dapat Anda gunakan ketika terjadi suatu yang sangat serius. Misalnya ketika terjadi permasalahan pada server atau

konfigurasi keamanan yang gagal. Penggunaan alert yang sesuai akan memberikan user experience yang menarik.

Untuk menggunakan class ini cukup masukkan kode `<div class="alert alert-danger">`.

Pengguna tidak akan kesulitan menemukan error yang terjadi sehingga problem solving pun dapat dilakukan dengan cepat.

4. Membuat Tombol Dengan Bootstrap

Tombol adalah komponen yang umum dan sering digunakan untuk membuat website. Inilah yang membuat Bootstrap juga menyediakan class untuk membuat tombol yang menarik. Anda tinggal memanggil salah satu class tombol dan tombol yang indah dapat Anda tampilkan di halaman website.

Supaya Anda bisa menggunakan tombol di dalam website, cukup panggil class “btn”. Selain itu, Anda bisa menambahkan class lain untuk memberikan variasi tombol di dalamnya.

Terdapat banyak variasi tombol yang disediakan oleh Bootstrap, seperti “.btn”, “.btn-default”, “.btn-primary”, “.btn-success”, “.btn-info”, “.btn-warning”, “.btn-danger”, dan “.btn-link”.

Contoh penggunaan skrip seperti di bawah ini:

```
<div class="container">    <h2>Button Styles</h2>    <button type="button" class="btn">Basic</button>    <button type="button" class="btn btn-default">Default</button>    <button type="button" class="btn btn-primary">Primary</button>    <button type="button" class="btn btn-success">Success</button>    <button type="button" class="btn btn-info">Info</button>    <button type="button" class="btn btn-warning">Warning</button>    <button type="button" class="btn btn-danger">Danger</button>    <button type="button" class="btn btn-link">Link</button>    </div>
```

- .btn: Class ini untuk membuat tombol standar warna abu kehitaman.

- `.btn-default`: Class ini untuk membuat tombol default berwarna abu.
- `.btn-primary`: Class ini digunakan untuk membuat tombol warna biru. Biasanya digunakan untuk tombol login dan masuk ke dalam menu tertentu.
- `.btn-success`: Class ini digunakan untuk membuat tombol ketika berhasil menginputkan sesuatu atau memproses suatu program.
- `.btn-info`: Class ini digunakan untuk membuat tombol warna biru muda.
- `.btn-warning`: Class ini digunakan untuk membuat tombol warning dengan warna kuning.
- `.btn-danger`: Class ini digunakan untuk membuat tombol warning warna merah tua.
- `.btn-link`: Selain tombol persegi, Bootstrap juga menyediakan tombol untuk link. Anda bisa menggunakan class ini.

5. Membuat Panel Dengan Bootstrap

Contoh terakhir penggunaan Bootstrap adalah membuat panel. Panel adalah sebuah garis yang terbentuk dari garis dan padding.

Di Bootstrap, sebuah panel terdiri dari tiga bagian; header, body, dan footer. Header berada di bagian atas. Body di bagian tengah atau inti. Sedangkan footer di bagian bawah.

Untuk membuat panel dengan header, body, dan footer harus dideklarasikan terlebih dahulu menggunakan kode seperti di bawah ini.

```
<div class="panel panel-default">      <div class="panel-  
heading">Panel Heading</div>      <div class="panel-body">Panel  
Content</div>      <div class="panel-footer">Panel Footer</div>  
</div>
```

Terdapat banyak variasi panel yang dapat dibuat menggunakan Bootstrap, seperti “.panel-default”, “.panel-primary”, “.panel-success”, “.panel-info”, “.panel-warning”, dan “.panel-danger”

- .panel-default: Class ini digunakan untuk membuat sebuah panel dengan warna biasa (abu).

```
<div class="panel panel-default">      <div class="panel-  
heading">Panel dengan class panel-default</div>      <div  
class="panel-body">Isi Konten</div>  </div>
```

- .panel-primary: Class ini digunakan untuk membuat panel dengan warna biru tua.


```
<div class="panel panel-primary">      <div class="panel-
heading">Panel dengan class panel-primary</div>      <div
class="panel-body">Isi Konten</div>  </div>
```

- .panel-success: Class ini digunakan untuk membuat panel dengan dengan warna atas hijau.

```
<div class="panel panel-success">      <div class="panel-
heading">Panel dengan class panel-success</div>      <div
class="panel-body">Isi Konten</div>  </div>
```

- .panel-info: Class ini digunakan untuk membuat panel dengan warna header biru muda.

```
<div class="panel panel-info">      <div class="panel-
heading">Panel dengan class panel-info class</div>      <div
class="panel-body">Isi Konten</div>  </div>
```

- .panel-warning: Class ini digunakan untuk membuat panel dengan warna header kuning muda.

```
<div class="panel panel-info">      <div class="panel-
heading">Panel dengan class panel-info class</div>      <div
class="panel-body">Isi Konten</div>  </div>
```

- .panel-danger: Class ini digunakan untuk membuat panel dengan warna merah muda,

```
<div class="panel panel-danger">      <div class="panel-
heading">Panel dengan class panel-danger class</div>      <div
class="panel-body">Isi Konten</div>  </div>
```

Seperti framework lainnya, Bootstrap dibuat dengan tujuan untuk mempercepat proses pengembangan suatu web. Jika ada yang tertarik ingin menggunakan Bootstrap, maka wajib mengetahui apa saja kelebihanannya terlebih dahulu.

Berikut ini adalah beberapa kelebihan Bootstrap:

- Mempercepat proses pengembangan website
- Bootstrap memiliki tampilan yang modern
- Bootstrap mengedepankan tampilan mobile friendly, yang artinya website buatan kamu akan nyaman untuk diakses dari ponsel dan perangkat lainnya
- Ringan karena struktur kodenya “bersih”, tidak ada kode yang tidak perlu
- Dikembangkan dengan sangat aktif
- Banyak tutorial yang tersedia di internet untuk mempelajari framework ini
- Sudah banyak website besar yang menggunakan Bootstrap sebagai frameworknya

BAB IV

Memahami Cara Kerja Php Dan Codeigniter

4.1 Menjalankan Program Sederhana PHP

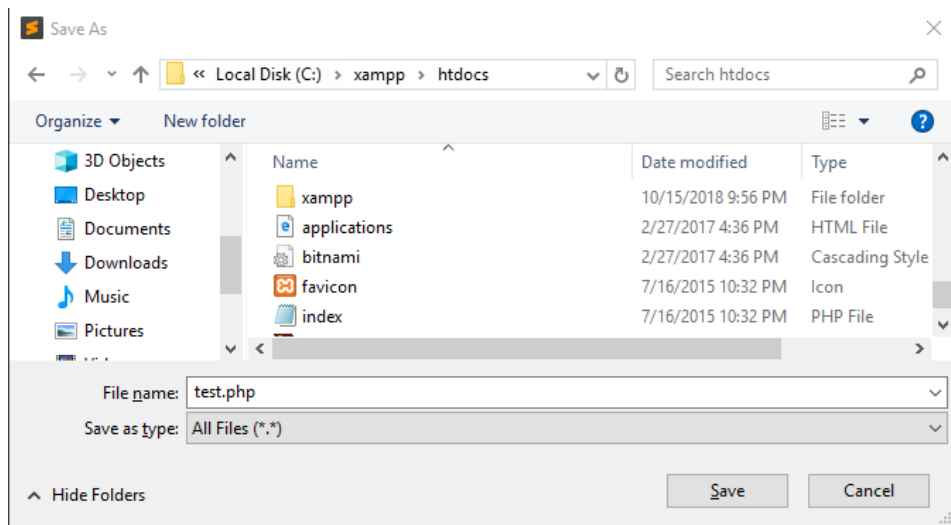
Untuk memulai membuat program sederhana PHP, terlebih dahulu kita buka sublime text dan ketikkan kode berikut.

```
<?php
    echo "Hello World !";
?>
```

Keterangan:

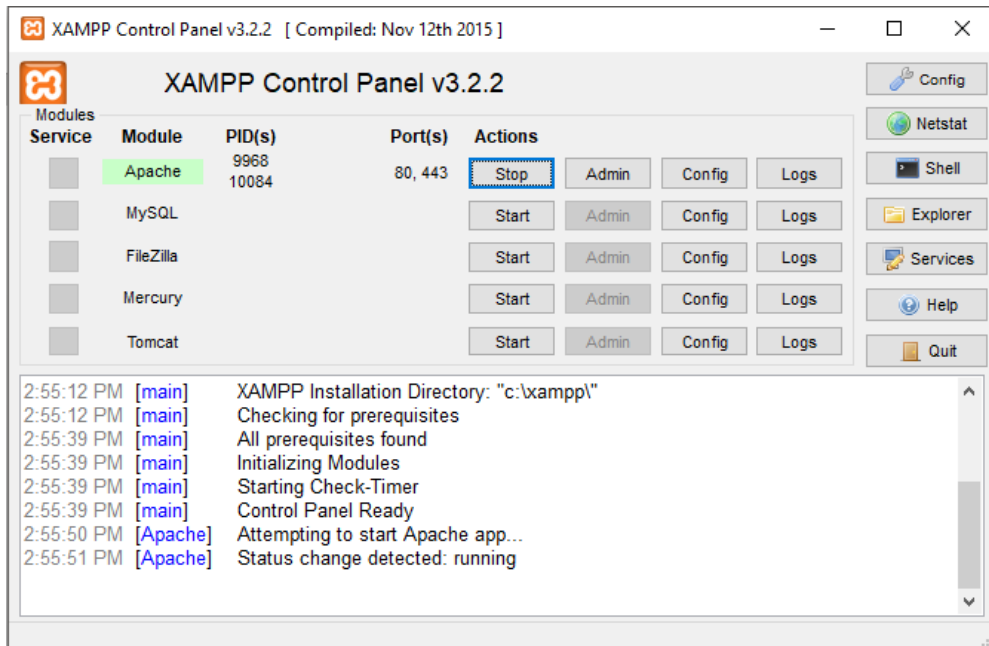
- <?php, merupakan pembuka kode program PHP yang menandakan dimulainya kode program PHP.
- echo, merupakan baris perintah untuk menampilkan “Hello World !” di halaman web.
- “Hello World !”, string yang akan ditampilkan di halaman web.
- ; , merupakan akhir yang menandakan berakhirnya satu baris kode PHP.
- ?>, merupakan penutup kode program PHP.

Kemudian simpan file dengan nama test.php dan letakkan di dalam folder **htdocs** yang ada di drive C folder XAMPP, seperti gambar berikut.



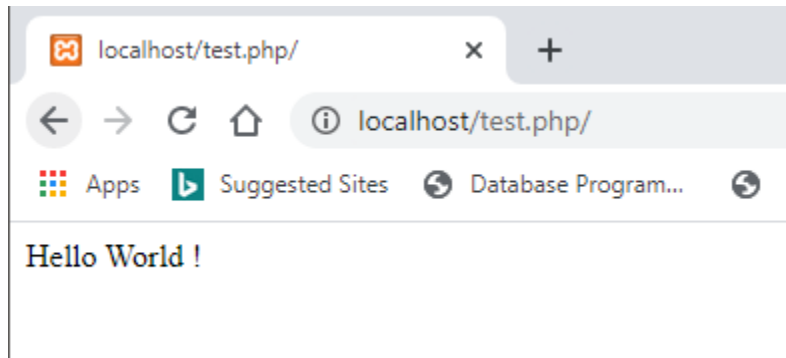
Gambar 4.1 Direktori Penyimpanan File test.php

Setelah tersimpan maka selanjutnya aktifkan web server apache yang ada di XAMPP dengan cara klik Start yang ada di sebelah apache, seperti gambar berikut.



Gambar 4.2 Start Apache

Selanjutnya buka web browser dan ketikkan **localhost/test.php** pada URL dan lihat gambar berikut.



Gambar 4.3 Hasil Tampilan Program Sederhana PHP

4.2 Konfigurasi CodeIgniter

Sebelum menggunakan CodeIgniter untuk membuat aplikasi yang nyata, diperlukan konfigurasi. File konfigurasi terletak di dalam folder **application/config**. File yang terdapat di dalam folder tersebut yang sering digunakan adalah **config.php**, **autoload.php**, **database.php**, dan **routes.php**.

1. Config.php

Pada file ini berisikan konfigurasi CodeIgniter secara umum.

Pengaturan konfigurasi dapat dilakukan pada file **config.php** yang terletak dalam folder **C:\xampp\htdocs\si-budget\application\config**. Temukan potongan script berikut:

```
$config['base_url'] = '';  
$config['index_page'] = 'index.php';  
$config['encryption_key'] = '';
```

Ubah menjadi:

```
//Mengatur alamat URL aplikasi yang dibuat  
$config['base_url'] = 'http://localhost/si-budget';  
  
//Menghilangkan index.php pada URL  
$config['index_page'] = '';  
  
//Mengamankan session  
$config['encryption_key'] = 'kuncienkripsi';
```

2. Autoload.php

Konfigurasi pada file ini bertujuan menentukan sumber daya apa saja yang akan dimuat secara otomatis. Pengaturan konfigurasi dapat dilakukan pada file **autoload.php** yang terletak dalam folder **C:\xampp\htdocs\si-budget\application\config**. Temukan potongan script berikut:

```
$autoload['libraries'] = array();  
$autoload['helper'] = array();
```

Ubah menjadi:

```
// Melakukan proses autoloading terhadap library yang  
digunakan  
$autoload['libraries'] = array('database', 'session',  
'email', 'form_validation');  
  
// Melakukan proses autoloading terhadap helper yang  
digunakan  
$autoload['helper'] = array('url', 'file', 'string',  
'form');
```


3. Database.php

Konfigurasi pada file ini bertujuan menghubungkan ke database. Pengaturan konfigurasi dapat dilakukan pada file **database.php** yang terletak dalam folder **C:\xampp\htdocs\si-budget\application\config**. Isi sesuai dengan database yang digunakan aplikasi, seperti berikut ini.

```
$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'si_budget',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

4. Routes.php

Konfigurasi di file ini bertujuan memetakan permintaan ke controller di dalam aplikasi yang dibuat. Pengaturan konfigurasi dapat dilakukan pada file **routes.php** yang terletak dalam folder **C:\xampp\htdocs\si-budget\application\config**.

```
$route['default_controller'] = 'login';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
$route['dashboard'] = 'overview';
$route['checkpoint'] = "login/validasi";
$route['logout'] = "login/logout";
$route['add_user'] = 'admin/input_user';
$route['save_user'] = 'admin/save_input_user';
$route['table_user'] = 'admin/show_user_table';
$route['forgotpassword'] = 'login/forgot_password';
$route['budgetvp'] = 'overview/input_budget';
$route['table_budget'] = 'overview/show_budget_table';
$route['departemen'] = 'overview/departemen';
$route['pengalihan'] = 'overview/tf_budget';
$route['setting'] = 'overview/setting';
$route['changepassword'] = 'overview/changepassword';
$route['pelatihan'] = 'controllertep';
$route['formpelatihan'] =
'controllertep/form_pelatihan';
$route['pesertapelatihan'] =
'controllertep/inputpeserta';
$route['updateharga'] = 'overview/update_harga';
$route['form2'] = 'controllertep/form1';
$route['totalcost'] = 'controllertep/total';
$route['add_pelatihan'] = 'admin/form_add_pelatihan';
$route['add_pekerja'] = 'admin/form_add_pekerja';
$route['add_provider'] = 'admin/form_add_provider';
```

```
$route['add_hargapatra'] =  
'admin/form_add_harga_patra';  
$route['editpeserta'] =  
'controllertep/form_edit_peserta';  
$route['details'] =  
'controllertep/detail_batch_pelatihan';  
$route['peserta'] = 'controllertep/peserta';  
$route['data_pekerja'] = 'admin/data_pekerja';  
$route['data_pelatihan'] = 'admin/data_pelatihan';  
$route['data_harga_patra'] = 'admin/data_harga_patra';  
$route['data_provider'] = 'admin/data_provider';
```

Semua route tersebut merupakan route yang digunakan dalam aplikasi di buku ini.

4.3 Contoh Menggunakan Model, View, dan Controller

1. Model

Model merupakan komponen yang berhubungan dengan database yang melakukan operasi perintah seperti memasukkan data dari form ke database, mengedit, menghapus, dan perintah lainnya. Untuk penamaan kelas model, pada huruf pertama harus menggunakan huruf besar dan harus sama dengan nama file dari model itu sendiri.

Untuk memahami penggunaan model, silahkan buat file dengan nama **Testing_model.php** di folder **application/models** dengan isi script seperti berikut ini:

```
<?php

class Testing_model extends CI_Model {
}
```

Dalam penggunaannya, model bisa dipanggil melalui controller. Contoh pemanggilannya menggunakan fungsi seperti berikut:

```
$this->load->model('nama_model');
```

2. Controller

Controller merupakan komponen yang menghubungkan antara model dan view. Dalam penamaan kelas di controller, di huruf pertama harus menggunakan huruf besar dan harus sama dengan nama filenya. Untuk memahami penggunaan controller, silahkan buat file dengan nama **Testing.php** di folder **application/controllers** dengan isi script seperti berikut ini:

```
<?php
class Testing extends CI_Controller{
    public function index(){
        $this->load->view("view_testing");
    }
}
```

Pada contoh tersebut menunjukkan bahwa pada saat aplikasi dijalankan, controller akan memuat file `view_testing.php` yang ada di dalam folder `view`. Dalam satu controller bisa terdiri dari banyak function atau method.

3. View

View merupakan bagian yang mengatur tampilan halaman web ke pengguna. Buat file dengan nama **view_testing.php** di folder **application/views**.

```
<!DOCTYPE html>
<html>
<head>
    <title>Contoh View</title>
</head>
<body>
    <p>Ini adalah contoh dari View</p>
</body>
</html>
```

Untuk menjalankan view, view harus dimuat terlebih dahulu di controller. Untuk memuat view bisa menggunakan fungsi yang diletakkan di controller, seperti berikut ini:

```
$this->load->view("nama_view");
```

BAB V

Menggunakan Bootstrap pada CodeIgniter

5.1

5.2

5.3

BAB VI

Pembuatan Aplikasi

6.1 Pembagian Elemen

6.1.1 Header

Buat view dengan nama **header.php** dan simpan di folder **si-budget/application/views/_partials**.

```
1.      <link rel="icon" type="image/ico" href="php echo
      base_url('images/favicon.ico')?&gt;"/&gt;
2.      &lt;meta charset="utf-8"&gt;
3.      &lt;meta http-equiv="X-UA-Compatible" content="IE=edge"&gt;
4.      &lt;meta name="viewport" content="width=device-width,
      initial-scale=1, shrink-to-fit=no"&gt;
5.      &lt;meta name="description" content=""&gt;
6.      &lt;meta name="author" content=""&gt;
7.
8.      &lt;title&gt;&lt;?php echo SITE_NAME .": ". ucfirst($this-&gt;uri-
      &gt;segment(1)) ." - ". ucfirst($this-&gt;uri-&gt;segment(2))
      ?&gt;&lt;/title&gt;
9.
10.     &lt;!-- Bootstrap core CSS--&gt;
11.     &lt;link href="<?php echo
      base_url('assets/bootstrap/css/bootstrap.min.css') ?&gt;"
      rel="stylesheet"&gt;
12.     &lt;link href="<?php echo
      base_url('assets/bootstrap/css/bootstrap.css') ?&gt;"
      rel="stylesheet"&gt;
13.     &lt;!-- Custom fonts for this template--&gt;
14.     &lt;link href="<?php echo base_url('assets/fontawesome-
      free/css/all.min.css') ?&gt;" rel="stylesheet"
      type="text/css"&gt;
15.
16.     &lt;!-- Page level plugin CSS--&gt;</pre
```



```

17.     <link href="<?php echo
      base_url('assets/datatables/dataTables.bootstrap4.css')
      ?>" rel="stylesheet">
18.
19.     <!-- Custom styles for this template-->
20.     <link href="<?php echo base_url('css/sb-admin.css')
      ?>" rel="stylesheet">

```

6.1.2 Navigasi Bar

Buat view dengan nama **navbar.php** dan simpan di folder **si-budget/application/views/_partials**.

```

1.  <nav class="navbar navbar-expand navbar-dark bg-dark
    static-top">
2.
3.      <a class="navbar-item"></a>
4.
5.      <button class="btn btn-link btn-sm text-white order-1
    order-sm-0" id="sidebarToggle" href="#">
6.          <i class="fas fa-bars"></i>
7.      </button>
8.
9.      <form class="d-none d-md-inline-block form-inline ml-
    auto mr-0 mr-md-3 my-2 my-md-0">
10.         <div class="input-group">
11.             <div class="input-group-append">
12.                 </div>
13.             </div>
14.         </form>
15.
16.         <!-- Navbar -->
17.         <ul class="navbar-nav ml-auto ml-md-0">
18.             <li class="nav-item no-arrow mx-1">
19.                 </li>
20.             <li class="nav-item no-arrow mx-1">
21.                 </li>
22.             <li class="nav-item no-arrow">

```

```

23.         <a class="nav-link dropdown-toggle" href="#"
    id="userDropdown" role="button" data-toggle="dropdown"
    aria-haspopup="true" aria-expanded="false">
24.             <i class="fas fa-user-circle fa-fw"></i>
25.         </a>
26.         <div class="dropdown-menu dropdown-menu-right"
    aria-labelledby="userDropdown">
27.             <a class="dropdown-item" href="<?php echo
    base_url('setting');?>">Settings</a>
28.             <div class="dropdown-divider"></div>
29.             <a class="dropdown-item" href="#" data-
    toggle="modal" data-target="#logoutModal">Logout</a>
30.         </div>
31.     </li>
32. </ul>
33.
34. </nav>

```

6.1.3 Side Bar

Buat view dengan nama **sidebar.php** dan simpan di folder **si-budget/application/views/_partials**.

```

1. <!-- Sidebar -->
2. <ul class="sidebar navbar-nav">
3.     <li class="mt-2 nav-item <?php if($active ==
    'dashboard'){echo 'active';} ?>">
4.         <a class="nav-link" href="<?php echo
    site_url('dashboard') ?>">
5.             <i class="fas fa-fw fa-tachometer-alt"></i>
6.             <span>Dashboard</span>
7.         </a>
8.     </li>
9.
10.    <?php
11.        $grup = $this->session->userdata('grup')->grup;
12.        if($grup == 'External'){
13.            ?>
14.
15.        <?php

```

```

16.         } elseif($grup == 'Admin') {
17.             ?>
18.
19.             <li class='nav-item <?php if($active ==
'<?php echo
20.             <a class='nav-link' href='<?php echo
site_url('table_user') ?>'>
21.                 <i class='fas fa-fw fa-users'></i>
22.                 <span>User data</span>
23.             </a>
24.         </li>
25.
26.         <li class="nav-item dropdown <?php if($active ==
'<?php echo
27.         <a class="nav-link dropdown-toggle" href=""
id='pagesDropdown' role='button' data-toggle='dropdown'
aria-haspopup='true' aria-expanded='false'>
28.             <i class='far fa-fw fa-folder'></i>
29.             <span>Master data</span>
30.         </a>
31.         <div class='dropdown-menu' aria-
labelledby='pagesDropdown'>
32.             <a class='dropdown-item' href='<?php echo
site_url('data_pekerja') ?>'>Data Pekerja</a>
33.             <a class='dropdown-item' href='<?php echo
site_url('data_pelatihan') ?>'>Data Pelatihan</a>
34.             <a class='dropdown-item' href='<?php echo
site_url('data_provider') ?>'>Data Provider</a>
35.             <a class='dropdown-item' href='<?php echo
site_url('data_harga_patra') ?>'>Data Harga Patra</a>
36.         </div>
37.         </li>
38.
39.         <?php
40.         } elseif($grup == 'TEP') {
41.             ?>
42.
43.             <li class="nav-item <?php if($active ==
'<?php echo
44.             <a class='nav-link' href='<?php echo
site_url('pelatihan') ?>' >

```

```

45.         <i class="fas fa-fw fa-certificate"></i>
46.         <span>Pelatihan</span>
47.     </a>
48. </li>
49. <?php
50.     } elseif($grup == 'LS') {
51.     ?>
52.
53.     <li class='nav-item dropdown <?php if($active ==
54. 'budget'){echo 'active';} ?>'>
55.         <a class='nav-link dropdown-toggle' href=''
56. id='pagesDropdown' role='button' data-toggle='dropdown'
57. aria-haspopup='true'
58.         aria-expanded='false'>
59.             <i class='fas fa-fw fa-wallet'></i>
60.             <span>Budget pelatihan</span>
61.         </a>
62.         <div class='dropdown-menu' aria-
63. labelledby='pagesDropdown'>
64.             <a class='dropdown-item' href='<?php echo
65. site_url('table_budget') ?>'>Budget table</a>
66.             <a class='dropdown-item' href='<?php echo
67. site_url('budgetvp') ?>'>Input budget</a>
68.             <a class='dropdown-item' href='<?php echo
69. site_url('pengalihan') ?>'>Transfer budget</a>
70.         </div>
71.     </li>
72.
73.     <li class="nav-item <?php if($active ==
74. 'updateharga'){echo 'active';} ?>">
75.         <a class="nav-link" href="<?php echo
76. site_url('updateharga') ?>">
77.             <i class="fas fa-fw fa-edit"></i>
78.             <span>Update harga</span>
79.         </a>
80.     </li>
81.
82. <?php
83.     } else { echo "Error";}
84.     ?>
85.

```

```

77.     <li class="nav-item <?php if($active ==
      'setting'){echo 'active';} ?>">
78.         <a class="nav-link" href="<?php echo
      site_url('setting') ?>">
79.             <i class="fas fa-fw fa-cog"></i>
80.             <span>Settings</span>
81.         </a>
82.     </li>
83. </ul>

```

6.1.4 Scroll Top

Buat view dengan nama **scrolltop.php** dan simpan di folder **si-budget/application/views/_partials**.

```

1. <!-- Scroll to Top Button-->
2. <a class="scroll-to-top rounded" href="#page-top">
3.     <i class="fas fa-angle-up"></i>
4. </a>

```

6.1.5 Modal untuk Logout

Buat view dengan nama **modal.php** dan simpan di folder **si-budget/application/views/_partials**.

```

1. <!-- Logout Modal-->
2. <div class="modal fade" id="logoutModal" tabindex="-1"
   role="dialog" aria-labelledby="exampleModalLabel" aria-
   hidden="true">
3.     <div class="modal-dialog" role="document">
4.         <div class="modal-content">
5.             <div class="modal-header">
6.                 <h5 class="modal-title"
   id="exampleModalLabel">Ready to Leave?</h5>
7.                 <button class="close" type="button" data-
   dismiss="modal" aria-label="Close">
8.                     <span aria-hidden="true">x</span>
9.                 </button>
10.            </div>

```

```

11.         <div class="modal-body">Select "Logout" below if
        you are ready to end your current session.</div>
12.         <div class="modal-footer">
13.             <button class="btn btn-secondary" type="button"
        data-dismiss="modal">Cancel</button>
14.             <a class="btn btn-primary" href="<?php echo
        base_url('logout') ?>">Logout</a>
15.         </div>
16.     </div>
17. </div>
18. </div>

```

6.2 Membuat Halaman Login

6.2.1 Controller Login

Buat controller dengan nama **login.php** dan simpan di folder **si-budget/application/controllers**.

```

1. <?php
2. class Login extends CI_Controller {
3.     // Controller untuk fitur login
4.     public function __construct()
5.     {
6.         parent::__construct();
7.         $this->load->model('system_model');
8.     }
9.
10.    public function index()
11.    {
12.        if($this->session->userdata('status') ==
        "login") {
13.            redirect(base_url("dashboard"));
14.            // jika userdata dengan nama status bernilai
        'login'
15.        }
16.        else {
17.            $this->load->view("login");

```

```

18.         // jika userdata dengan nama status bernilai
        selain 'login'
19.     }
20. }
21.
22.     function validasi()
23.     {
24.         // Fungsi ini berguna untuk menerima hasil
        input user pada form login yang berada di berkas views
        'login.php', kemudian memeriksa apakah data berupa
        username dan password yang dimasukkan oleh user valid
        (Tersedia di dalam basis data)
25.
26.         // mengambil data input
27.         $username = $this->input->post('username');
28.         $password = $this->input->post('password');
29.
30.         // menyimpan data dalam array
31.         $where = array(
32.             'username' => $username,
33.             'password' => md5($password)
34.         );
35.
36.         // memeriksa pada tabel user apakah ada baris
        yang menyimpan data sesuai dengan isi array
37.         $result = $this->system_model->check("user",
        $where, 'all')->num_rows();
38.         if($result > 0)
39.         {
40.             // Jika baris ada, maka ambil data grup dan
            email dari baris
41.             $temp = $this->system_model->check("user",
            $where, 'grup');
42.             $temp1 = $this->system_model->check("user",
            $where, 'email');
43.
44.             $data_session = array(
45.                 'username' => $username,
46.                 'email' => $temp1,
47.                 'grup' => $temp,
48.                 'status' => "login"

```

```

49.         );
50.
51.         // menetapkan userdata username, email,
    grup, status
52.         $this->session->set_userdata($data_session);
53.
54.         // masuk ke dashboard
55.         redirect(base_url("dashboard"));
56.
57.     }else{
58.
59.         // Jika baris tidak ada, maka akan
    menampilkan keterangan bahwa data invalid
60.         $data['invalid'] = "Invalid login
    credentials";
61.         $this->load->view("login", $data);
62.
63.     }
64. }
65.
66. function logout()
67. {
68.     // fungsi ini berguna untuk menghapus sesi
    seorang user yang telah melakukan logout, kemudian
    mengarahkan route ke bagian login
69.
70.     $this->session->sess_destroy();
71.     redirect(base_url('login'));
72. }
73.
74. function forgot_password()
75. {
76.     // fungsi ini berguna untuk menampilkan views
    'forgot_password.php'
77.     $this->load-
    >view('reset_password/forgot_password');
78. }
79. }

```


6.2.2 Model Login

6.2.3 View Login

Buat view dengan nama **login.php** dan simpan di folder **si-budget/application/views**.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <title>Login</title>
5.     <link rel="icon" type="image/ico" href="<?php echo
base_url('images/favicon.ico') ?>"/>
6.     <meta charset="utf-8">
7.     <meta name="viewport" content="width=device-width,
initial-scale=1">
8.     <!-- Bootstrap core CSS-->
9.     <link href="<?php echo
base_url('assets/bootstrap/css/bootstrap.css') ?>"
rel="stylesheet">
10.    <link href="<?php echo
base_url('assets/bootstrap/css/sign-in.css') ?>"
rel="stylesheet">
11.  </head>
12. <body class="text-center">
13.   <form class="form-signin" method="post" action="<?php
echo base_url('checkpoint');?>">
14.   
15.     <input type="username" name="username"
id="inputUsername" class="form-control"
placeholder="Username" required autofocus>
16.     <input type="password" name="password"
id="inputPassword" class="form-control"
placeholder="Password" required>
17.     <?php if (isset($invalid)): ?>
18.       <h6 style='color:red;'>Invalid login credentials.
Please try again</h6>
19.     <?php endif ?>
```

```

20.     <button class="form-control btn btn-lg btn-dark btn-
      block mt-4" type="submit">Login</button>
21.   <p>
22.     <?php
23.       echo anchor(site_url(). 'lupa_password', 'Forgot
      Password?');
24.     ?>
25.   </p>
26. </form>
27.
28. <script src="<?php echo
      base_url('assets/jquery/jquery.min.js') ?>"></script>
29. <script src="<?php echo
      base_url('assets/bootstrap/js/bootstrap.bundle.min.js')
      ?>"></script>
30. </body>
31. </html>

```

6.3 Membuat Halaman Dashboard

6.3.1 Controller Dashboard

Buat controller dengan nama **overview.php** dan simpan di folder **si-budget/application/controllers**.

```

1. <?php
2. class Overview extends CI_Controller {
3.     // Controller untuk menu dashboard dan fitur pada user
      group User LS, User External
4.     public function __construct()
5.     {
6.         parent::__construct();
7.         if($this->session->userdata('status') !=
      "login"){
8.             redirect(base_url("login"));
9.         }
10.    }
11.
12.    public function index()

```

```

13.     {
14.         $this->load->model('system_model');
15.         // data tabel direktorat
16.         $result['dirdata'] = $this->system_model-
>show_table('direktorat');
17.
18.         $result['jumlah'] = $this->system_model-
>total_anggaran('direktorat');
19.         $data['total'] = $this->system_model-
>hitungtotal();
20.
21.         // ambil data tabel batch_pelatihan dari db
22.         $temp = $this->system_model-
>show_table('batch_pelatihan');
23.
24.         // menetapkan array kosong
25.         $var = array();
26.         $var2 = array();
27.
28.         foreach($temp as $row){
29.             $tgl_selesai = strtotime($row-
>tgl_selesai);
30.             array_push($var, date('Y',
$ttl_selesai)); // Y untuk format tahun 4 digit
31.             array_push($var2, date('F',
$ttl_selesai)); // F untuk format nama bulan lengkap
32.         }
33.         // $var menyimpan data tahun dari tiap baris
pada kolom ttl_selesai dari tabel batch_pelatihan
34.         // $var2 menyimpan data bulan dari tiap baris
pada kolom ttl_selesai dari tabel batch_pelatihan
35.
36.         // daftar tahun yang ada di tabel db
batch_pelatihan
37.         $temp = array_unique($var); //menghapus
duplikat
38.         $result['year'] = $temp; //menetapkan variabel
year yang akan disampaikan ke view dashboard
39.
40.         // daftar bulan yang ada di tabel db
batch_pelatihan

```

```

41.         $temp2 = array_unique($var2); // menghapus
           duplikat
42.         $result['month'] = $temp2; // menetapkan
           variabel month yang akan disampaikan ke view dashboard
43.
44.         // menerima data dropdown
45.         $month = $this->input->post('month');
46.         $year = $this->input->post('year');
47.
48.         if($month == '' && $year == '') {
49.             // Jika input data kosong yang artinya
           belum ada tindakan filter dari pengguna (default)
50.             $month = date_parse(max($temp2));
           //mengambil bulan paling besar/current month
51.             // date_parse digunakan untuk mengubah
           notasi bulan dalam bentuk text ke notasi bulan dalam
           bentuk number
52.
53.             $year = max($temp); // mengambil tahun
           paling besar/current year
54.             $result['direktorat'] = $this-
           >system_model->get_realisasi_dir($month['month'],
           $year); // mengambil data realisasi direktorat sesuai
           bulan dan tahun
55.
56.             $result['selected_month'] =
           max($temp2); // menetapkan variabel selected_month untuk
           disampaikan ke view dashboard
57.         } else {
58.             // Jika input data tidak kosong yang
           artinya pengguna menekan tombol filter
59.             $result['direktorat'] = $this-
           >system_model->get_realisasi_dir($month, $year);
60.             // mengambil data realisasi direktorat
           sesuai bulan dan tahun
61.
62.             $dateObj =
           DateTime::createFromFormat('!m', $month); // konversi
           notasi bulan dalam bentuk number ke notasi bulan dalam
           bentuk text

```

```

63.             $result['selected_month'] = $dateObj-
>format('F'); // menetapkan variabel selected_month untuk
disampaikan ke view dashboard
64.         }
65.
66.         $temp3 = array();
67.         foreach($result['direktorat'] as $rows){ //
menetapkan daftar direktorat yang sudah ada
realisasi/menggunakan anggaran
68.             array_push($temp3, $rows-
>direktorat_pekerja);
69.         }
70.
71.         $result['array_dir'] = $temp3;
72.         $result['selected_year'] = $year;
73.
74.         // memuat tampilan dashboard dan menyampaikan
data-data $result ke berkas view 'overview.php'
75.         $this->load->view("overview", $result);
76.     }
77.
78.     function setting()
79.     {
80.         // menampilkan view 'setting.php'
81.         $this->load->view('setting');
82.     }
83.
84.     function changepassword()
85.     {
86.         // menerima data input form setting
87.         $oldpw = $this->input->post('oldpass');
88.         $newpw = $this->input->post('newpass');
89.
90.         // load model
91.         $this->load->model('system_model');
92.
93.         // memperbarui tabel user
94.         $this->system_model-
>update(array('password'=>$oldpw),
array('password'=>$newpw), 'user');
95.

```

```

96.         // mengarahkan ke setting
97.         redirect(base_url('setting'));
98.     }
99.     #fungsi untuk fitur pada user group LS
100.    function update_harga()
101.    {
102.        $this->load->model('system_model');
103.        $result['data'] = $this->system_model-
        >show_table('harga_patra');
104.        $this->load->view('update_harga', $result); //
        menyampaikan data tabel harga_patra ke view update_harga
105.    }
106.
107.    function updatingharga()
108.    {
109.        // mengambil data ajax
110.        $id= $this->input->post("id");
111.        $value= $this->input->post("value");
112.        $modul= $this->input->post("modul");
113.
114.        $this->load->model('system_model');
115.        $this->system_model-
        >update(array('id_harga_patra'=>$id),
        array($modul=>$value), 'harga_patra');
116.        // memperbarui tabel harga_patra
117.        echo "{}";
118.    }
119.
120.    function update_budget()
121.    {        // pengalihan anggaran
122.        $this->load->model('system_model');
123.        $date = date('Y-m-d');
124.        // get from input
125.        $vp = $this->input->post('vp');
126.        $vp2 = $this->input->post('vp2');
127.        $anggaran = $this->input->post('budget');
128.        // get old value from db
129.        $temp = $this->system_model->check('vp',
        array('nama_vp' => $vp), 'anggaran_vp');
130.        $temp2 = $this->system_model->check('vp',
        array('nama_vp' => $vp2), 'anggaran_vp');

```

```

131.         // set new value
132.         $anggaran1 = ($temp->anggaran_vp) -
(int)$anggaran;
133.         $anggaran2 = ($temp2->anggaran_vp) +
(int)$anggaran;
134.         // update db with new value
135.         $this->system_model-
>update(array('nama_vp'=>$vp),
array('update_date'=>$date, 'anggaran_vp'=>$anggaran1),
'vp');
136.         $this->system_model-
>update(array('nama_vp'=>$vp2),
array('update_date'=>$date, 'anggaran_vp'=>$anggaran2),
'vp');
137.         // back to view
138.         redirect(base_url("pengalihan"));
139.     }
140.
141.     function tf_budget()
142.     {
143.         // menampilkan view tf_budget.php
144.         $this->load->model('system_model');
145.         $result['dir'] = $this->system_model-
>show_table('direktorat');
146.         $result['svp'] = $this->system_model-
>show_table('svp');
147.         $result['vp'] = $this->system_model-
>show_table('vp');
148.         $this->load->view('tf_budget', $result);
149.     }
150.
151.     function input_budget()
152.     {
153.         // menampilkan view budget_pelatihan.php
154.         $this->load->model('system_model');
155.         $result['dir'] = $this->system_model-
>show_table('direktorat');
156.         $result['svp'] = $this->system_model-
>show_table('svp');
157.         $result['vp'] = $this->system_model-
>show_table('vp');

```

```

158.         $this->load->view("budget_pelatihan",
    $result);
159.     }
160.
161.     function save_input_budget()
162.     {
163.         // fungsi ini berguna untuk mengambil data
    input dari form input budget VP pada user group User LS
    kemudian menge
164.         $vp = $this->input->post('vp');
165.         $anggaran = $this->input->post('budget');
166.         $svp = $this->input->post('svp');
167.         $date = date("Y-m-d");
168.         $direktorat = $this->input-
    >post('direktorat');
169.
170.         $data = array(
171.             'anggaran_vp' => $anggaran,
172.             'update_date' => $date
173.         );
174.
175.         $this->load->model('system_model');
176.
177.         // update table vp
178.         $this->system_model-
    >update(array('nama_vp'=>$vp), $data, "vp");
179.         redirect(base_url("budgetvp"));
180.     }
181.
182.     function show_budget_table()
183.     {
184.         // menampilkan view table_budget.php
185.         $this->load->model('system_model');
186.         $result['data'] = $this->system_model-
    >show_table('vp');
187.         $this->load->view("table_budget", $result);
188.     }
189.
190.     function svp_detail($id_direktorat)
191.     {
192.         // menampilkan daftar svp

```



```

193.         // print_r($id_direktorat); die;
194.         $this->load->model('system_model');
195.         $result['data'] = $this->system_model-
>join_svp($id_direktorat);
196.         $this->load->view('details/detail_svp', $result);
197.     }
198.
199.     function vp_detail($id_svp)
200.     {
201.         // menampilkan daftar vp
202.         $this->load->model('system_model');
203.         $result['data'] = $this->system_model-
>join_vp($id_svp);
204.         $this->load->view('details/detail_vp', $result);
205.     }
206. }

```

6.3.2 Model Dashboard

6.3.3 View Dashboard

Buat view dengan nama **overview.php** dan simpan di folder **si-budget/application/views**.

```

1.  <!DOCTYPE html>
2.  <html lang="en">
3.
4.  <head>
5.      <?php $this->load->view("_partials/head.php") ?>
6.  </head>
7.
8.  <body id="page-top">
9.
10.     <?php $this->load->view("_partials/navbar.php") ?>
11.
12.     <div id="wrapper">
13.
14.         <?php
15.             $temp['active'] = "dashboard";
16.             $this->load->view("_partials/sidebar.php", $temp) ?>
17.
18.         <div id="content-wrapper">

```

```

19.
20.         <div class="container-fluid">
21.
22.             <div class="d-flex justify-content-between flex-
wrap flex-md-nowrap align-items-center pt-2 pb-2 mb-2">
23.                 <h1 class="h2">Dashboard</h1>
24.                 <div class="btn-toolbar mb-2 mb-md-0">
25.                     <div class="btn-group mr-2">
26.                         <a href="<?php base_url() ?>export"
type="button" class="btn btn-sm btn-outline-
dark">Export</a>
27.                     </div>
28.                     <form action="<?php echo
base_url('dashboard'); ?>" method="post">
29.                         <div class="btn-group" role="group">
30.                             <select name="month" id="month" type="select"
class="btn btn-sm btn-outline-dark">
31.                                 <?php foreach($month as $row): ?>
32.
33.                                     <?php
34.                                         $temp = date_parse($row);
35.                                         if($row == $selected_month):
36.                                             ?>
37.                                                 <option value="<?= $temp['month']
?>" selected><?= $row ?></option>
38.                                                 <?php else: ?>
39.                                                     <option value="<?= $temp['month']
?>"><?= $row ?></option>
40.                                                 <?php
41.                                                     endif;
42.                                                 endforeach;
43.                                                 ?>
44.                         </select>
45.                         <button type="submit" class="btn btn-sm btn-
outline-dark">Filter</button>
46.                         <select name="year" type="select" class="btn
btn-sm btn-outline-dark">
47.                             <!--<option>Year</option>-->
48.                             <?php foreach($year as $row): ?>
49.
50.                                     <?php if($row == $selected_year):
?>
51.                                         <option value='<?= $row ?>'
selected><?= $row ?></option>
52.                                         <?php else: ?>
53.                                             <option value= '<?= $row ?>'><?=
$row ?></option>
54.                                         <?php
55.                                             endif;
56.                                         endforeach;
57.                                         ?>

```

```

58.         </select>
59.
60.     </div>
61. </form>
62. </div>
63. </div>
64.
65.     <div class="row">
66.         <div class="col-lg-8">
67.             <div class="card mb-3">
68.                 <div class="card-header">
69.                     <i class="fas fa-chart-bar"></i>
70.                     Anggaran pelatihan Direktorat</div>
71.                 <div class="card-body">
72.                     <canvas id="myBarChart" width="100%"
height="50"></canvas>
73.                 </div>
74.                 <div class="card-footer small text-
muted"></div>
75.             </div>
76.         </div>
77.         <div class="col-lg-4">
78.             <div class="card mb-3">
79.                 <div class="card-header">
80.                     <i class="fas fa-chart-pie"></i>
81.                     Total anggaran pelatihan</div>
82.                 <div class="card-body">
83.                     <canvas id="myPieChart" width="100%"
height="111"></canvas>
84.                 </div>
85.                 <div class="card-footer small text-
muted"></div>
86.             </div>
87.         </div>
88.     </div>
89.
90.     <?php $this->load-
>view("_partials/budget_table_dir") ?>
91.
92. </div>
93. <!-- /.container-fluid -->
94.
95. <!-- Sticky Footer -->
96. <?php $this->load->view("_partials/footer.php") ?>
97.
98. </div>
99. <!-- /.content-wrapper -->
100.
101. </div>
102. <!-- /#wrapper -->
103.

```

```

104. <!-- Scroll to Top Button-->
105. <a class="scroll-to-top rounded" href="#page-top">
106.     <i class="fas fa-angle-up"></i>
107. </a>
108.
109. <?php $this->load->view("_partials/scrolltop.php") ?>
110. <?php $this->load->view("_partials/modal.php") ?>
111. <?php $this->load->view("_partials/js.php") ?>
112. <?php $this->load->view("_partials/chart.php") ?>
113. <script src="https://code.jquery.com/jquery-
    3.3.1.js"></script>
114. <script
    src="https://cdn.datatables.net/1.10.20/js/jquery.dataTab
    les.min.js"></script>
115. <script
    src="https://cdn.datatables.net/buttons/1.6.1/js/dataTabl
    es.buttons.min.js"></script>
116. <script
    src="https://cdnjs.cloudflare.com/ajax/libs/jsczip/3.1.3/j
    szip.min.js"></script>
117. <script
    src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.5
    3/pdfmake.min.js"></script>
118. <script
    src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.5
    3/vfs_fonts.js"></script>
119. <script
    src="https://cdn.datatables.net/buttons/1.6.1/js/buttons.
    html5.min.js"></script>
120. <script type="text/javascript">
121.     $(document).ready(function() {
122.         $('#example').DataTable( {
123.             dom: 'Bfrtip',
124.             buttons: [
125.                 'copyHtml5',
126.                 'excelHtml5',
127.                 'csvHtml5',
128.                 'pdfHtml5'
129.             ]
130.         } );
131.     } );
132. </script>
133. </body>
134. </html>

```

6.4 Membuat Halaman User Data di Admin

6.4.1 Controller User Data

6.4.2 Model User Data

6.4.3 View User Data

Buat view dengan nama **table_user.php** dan simpan di folder **si-budget/application/views**.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.     <?php $this->load->view("_partials/head.php") ?>
6.     <link href="<?php echo
base_url('assets/bootstrap/css/form_input_user.css') ?>"
rel="stylesheet">
7.     <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/sweetalert/sweetalert.css'); ?>">
8.     <script type="text/javascript" src="<?php echo
base_url('assets/jquery.js'); ?>"></script>
9.     <script type="text/javascript" src="<?php echo
base_url('assets/sweetalert/sweetalert.min.js');
?>"></script>
10. <style type="text/css">
11.
12.     td {
13.         cursor: pointer;
14.     }
15.
16.     .editor {
17.         display: none;
18.     }
```

```

19.
20.     </style>
21. </head>
22.
23. <body id="page-top">
24.
25.     <?php $this->load->view("_partials/navbar.php") ?>
26.
27.     <div id="wrapper">
28.
29.         <?php $temp['active'] = "data_user"; $this->load-
>view("_partials/sidebar.php", $temp) ?>
30.
31.         <div id="content-wrapper">
32.
33.             <div class="container-fluid">
34.
35.                 <!-- table -->
36.                 <div class="card mb-3">
37.                     <div class="card-header form-inline">
38.                         <i class="fas fa-table mr-1"></i>
39.                         User<a style="position: absolute; right: 0"
class="pr-4" href="<?php echo base_url('add_user');?>"><i
class="fas fa-plus"></i> Add New</a></div>
40.                     <div class="card-body">
41.                         <div class="table-responsive">
42.                             <table class="table table-bordered"
id="dataTable" width="100%" cellpadding="0">
43.                                 <thead>
44.                                     <tr>
45.                                         <th>Username</th>
46.                                         <th>Password</th>
47.                                         <th>Email</th>
48.                                         <th>Group</th>
49.                                         <th></th>
50.                                     </tr>
51.                                 </thead>
52.                                 <tbody>
53.                                     <?php
54.                                     foreach($data as $row)
55.                                     {

```

```

56.             echo "<tr data-id='". $row->id. "'>";
57.             echo "<td><span class='span-
username caption' data-id='". $row->id. "'>". $row-
>username."</span> <input type='text' class='field-
username form-control editor' value='". $row->username. "'
data-id='". $row->id. "' /></td>";
58.             echo "<td><span class='span-
password caption' data-id='". $row->id. "'>". $row-
>password."</span> <input type='text' class='field-
password form-control editor' value='". md5($row-
>password). "' data-id='". $row->id. "' /></td>";
59.             echo "<td><span class='span-email
caption' data-id='". $row->id. "'>". $row->email."</span>
<input type='text' class='field-email form-control
editor' value='". $row->email. "' data-id='". $row->id. "'
/></td>";
60.             echo "<td><span class='span-grup
caption' data-id='". $row->id. "'>". $row->grup."</span>
<input type='text' class='field-grup form-control editor'
value='". $row->grup. "' data-id='". $row->id. "' /></td>";
61.             ?>
62.             <td class="text-center"><button
class='btn btn-xs btn-danger delete-user' data-id='<?php
echo $row->id ?>'>Delete</button></td>
63.             </tr>
64.             <?php
65.             }
66.             ?>
67.             </tbody>
68.             </table>
69.             </div>
70.             </div>
71.             </div>
72.
73.             </div>
74.             <!-- Sticky Footer -->
75.             <?php $this->load->view("_partials/footer.php") ?>
76.
77.             </div>
78.             </div>
79.             <?php $this->load->view("_partials/scrolltop.php") ?>

```

```

80. <?php $this->load->view("_partials/modal.php") ?>
81. <?php $this->load->view("_partials/js.php") ?>
82. <script type="text/javascript">
83. $(function(){
84.
85. $.ajaxSetup({
86.     type:"post",
87.     cache:false,
88.     dataType: "json"
89. })
90.
91. $(document).on("click",".delete-user",function(){
92.     var id=$(this).attr("data-id");
93.     swal({
94.         title:"Delete User",
95.         text:"Are you sure?",
96.         type: "warning",
97.         showCancelButton: true,
98.         confirmButtonText: "Yes",
99.         closeOnConfirm: true,
100.     },
101.     function(){
102.         $.ajax({
103.             url:"<?php echo
base_url('index.php/overview/deleteuser'); ?>",
104.             data:{id:id},
105.             success: function(){
106.                 $("tr[data-
id='"+id+"'']").fadeOut("fast",function(){
107.                     $(this).remove();
108.                 });
109.             }
110.         });
111.     });
112. });
113.
114. $(document).on("click","td",function(){
115.     $(this).find("span[class~='caption']").hide();
116.     $(this).find("input[class~='editor']").fadeIn().focus();
117. });
118.

```



```

119. $(document).on("keydown", ".editor", function(e) {
120. if(e.keyCode==13) {
121. var target=$(e.target);
122. var value=target.val();
123. var id=target.attr("data-id");
124. var data={id:id,value:value};
125. if(target.is(".field-username")) {
126. data.modul="username";
127. }else if(target.is(".field-password")) {
128. data.modul="password";
129. }else if(target.is(".field-email")) {
130. data.modul="email";
131. }else if(target.is(".field-grup")) {
132. data.modul="grup";
133. }
134.
135. $.ajax({
136.   data:data,
137.   url:"<?php echo
      base_url('index.php/admin/update_user_table'); ?>",
138.   success: function(a) {
139.     target.hide();
140.
141.     target.siblings("span[class~='caption']").html(value).fadeIn();
142.   })
143. }
144. });
145. });
146. </script>
147. </body>
148. </html>

```

6.4.4 View Input User

Buat view dengan nama **form_input_user.php** dan simpan di folder **si-budget/application/views**.

```

1. <!DOCTYPE html>

```

```

2.  <html lang="en">
3.
4.  <head>
5.      <?php $this->load->view("_partials/head.php") ?>
6.      <link href="<?php echo
base_url('assets/bootstrap/css/form_input_user.css') ?>"
rel="stylesheet">
7.  </head>
8.
9.  <body id="page-top">
10.     <?php $this->load->view("_partials/navbar.php") ?>
11.     <div id="wrapper">
12.
13.         <?php $temp['active'] = "data_user"; $this->load-
>view("_partials/sidebar.php", $temp) ?>
14.
15.         <div id="content-wrapper">
16.             <div class="container-fluid">
17.
18.                 <?php //$this->load-
>view("_partials/breadcrumb.php") ?>
19.
20.                 <!-- form -->
21.                 <div class="card mb-3">
22.                     <div class="card-header">
23.                         <a href="<?php echo site_url('table_user')
?>"><i class="fas fa-arrow-left"></i> Back</a>
24.                     </div>
25.                     <div class="card-body">
26.
27.                         <div class="d-flex justify-content-between flex-
wrap flex-md-nowrap align-items-center pt-2 pb-2 mb-2">
28.                             <h1 class="h2 m-auto">Input form</h1>
29.                         </div>
30.
31.                         <form class="form-signin mb-4"
onsubmit="success()" autocomplete="off" method="post"
action="<?php echo base_url('save_user');?>" >
32.                             <label for="inputEmail" class="sr-
only">Email</label>

```

```

33.         <input type="user" name="email"
           id="inputEmail" class="form-control"
           placeholder="Email" required autofocus>
34.         <label for="inputEmail" class="sr-
           only">Username</label>
35.         <input type="user" name="username"
           id="inputEmail" class="form-control"
           placeholder="Username" required>
36.
37.         <label for="inputPassword" class="sr-
           only">Password</label>
38.         <input type="password" name="password"
           id="inputPassword" class="form-control mb-10"
           placeholder="Password" required>
39.
40.         <select name="group" type="user" class="form-
           control" id="group" required>
41.             <option value="">Group</option>
42.             <option value="Admin">Admin</option>
43.             <option value="TEP">TEP</option>
44.             <option value="LS">LS</option>
45.             <option value="External">External</option>
46.         </select>
47.
48.         <button class="btn btn-lg btn-dark btn-block mt-4
           form-control" type="submit">Submit</button>
49.     </form>
50.
51. </div>
52. </div>
53. </div>
54.     <!-- Sticky Footer -->
55.     <?php $this->load->view("_partials/footer.php") ?>
56.
57. </div>
58. </div>
59. <?php $this->load->view("_partials/scrolltop.php") ?>
60. <?php $this->load->view("_partials/modal.php") ?>
61. <?php $this->load->view("_partials/js.php") ?>
62. <script type="text/javascript">
63.     function success()

```

```

64.     {
65.         alert("Input success");
66.     }
67. </script>
68. </body>
69. </html>

```

6.5 Membuat Halaman Data Pekerja di Admin

6.5.1 Contoller

6.5.2 Model

6.5.3 View Data Pekerja

Buat view dengan nama **table_pekerja.php** dan simpan di folder **si-budget/application/views**.

```

1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.     <?php $this->load->view("_partials/head.php") ?>
6.     <link href="<?php echo
base_url('assets/bootstrap/css/form_input_user.css') ?>"
rel="stylesheet">
7.     <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/sweetalert/sweetalert.css'); ?>">
8.     <script type="text/javascript" src="<?php echo
base_url('assets/jquery.js'); ?>"></script>
9.     <script type="text/javascript" src="<?php echo
base_url('assets/sweetalert/sweetalert.min.js');
?>"></script>
10. </head>

```

```

11.
12. <body id="page-top">
13.     <?php $this->load->view("_partials/navbar.php") ?>
14.
15.     <div id="wrapper">
16.         <?php $temp['active'] = "tables"; $this->load-
17.         >view("_partials/sidebar.php", $temp) ?>
18.
19.         <div id="content-wrapper">
20.             <div class="container-fluid">
21.
22.                 <!-- table -->
23.                 <div class="card mb-3">
24.                     <div class="card-header">
25.                         <i class="fas fa-table"></i>
26.                         Pekerja
27.                         <a style="position: absolute; right: 0"
28.                         class="pr-4" href="<?php echo
29.                         base_url('add_pekerja');?>"><i class="fas fa-plus"></i>
30.                         Add New</a></div>
31.
32.                         <!-- <i><a href="<?php base_url() ?>pekerja"
33.                         type="" class="btn btn-sm btn-outline-
34.                         dark">Import</a></i> -->
35.
36.                         <div class="card-body">
37.                             <div class="table-responsive">
38.                                 <table class="table table-bordered"
39.                                 id="dataTable1" width="100%" cellpadding="0">
40.                                     <thead>
41.                                         <tr>
42.                                             <th>Nomor pekerja</th>
43.                                             <th>Nama pekerja</th>
44.                                             <th>VP</th>
45.                                             <th>SVP</th>
46.                                             <th>Direktorat</th>
47.                                             <th></th>
48.                                         </tr>
49.                                     </thead>
50.                                     <tbody>
51.                                         <?php
52.                                         foreach($pekerja as $row)
53.                                         {

```

```

45.         echo "<tr data-id='". $row-
    >id_pekerja.">";
46.         echo "<td>". $row-
    >nomor_pekerja."</td>";
47.         echo "<td>". $row-
    >nama_pekerja."</td>";
48.         echo "<td>". $row-
    >vp_pekerja."</td>";
49.         echo "<td>". $row-
    >svp_pekerja."</td>";
50.         echo "<td>". $row-
    >direktorat_pekerja."</td>";
51.         ?>
52.         <td>
53.             <!-- <a href="<?php echo
site_url('Admin/update_pekerja/'. $row->id_pekerja) ?>"
class="text-center"><button class='btn btn-xs btn-info
edit-pekerja' data-id='<?php $row->id_pekerja
?>'>Edit</button> -->
54.             <a href="" class="text-
center"><button class='btn btn-xs btn-danger delete-
pekerja' data-id='<?php echo $row->id_pekerja
?>'>Delete</button>
55.             </tr>
56.         <?php
57.         }
58.         ?>
59.     </tbody>
60. </table>
61. </div>
62. </div>
63. </div>
64. </div>
65. </div>
66. </div>
67.
68. <?php $this->load->view("_partials/scrolltop.php") ?>
69. <?php $this->load->view("_partials/modal.php") ?>
70. <?php $this->load->view("_partials/js.php") ?>
71. <!-- Sticky Footer -->
72. <?php $this->load->view("_partials/footer.php") ?>

```

```

73.
74. <script type="text/javascript">
75. $(function(){
76. $.ajaxSetup({
77.   type:"post",
78.   cache:false,
79.   dataType: "json"
80. })
81.
82. $(document).on("click",".delete-pekerja",function(){
83.   var id=$(this).attr("data-id");
84.   swal({
85.     title:"Delete Pekerja",
86.     text:"Are you sure?",
87.     type: "warning",
88.     showCancelButton: true,
89.     confirmButtonText: "Yes",
90.     closeOnConfirm: true,
91.   },
92.     function(){
93.       $.ajax({
94.         url:"<?php echo
base_url('index.php/admin/delete_pekerja'); ?>",
95.         data:{id:id},
96.         success: function(){
97.           $("tr[data-
id='"+id+"'']").fadeOut("fast",function(){
98.             $(this).remove();
99.           });
100.        }
101.      });
102.    });
103. });
104. });
105. </script>
106. </body>
107. </html>

```

6.5.4 View Tambah Data Pekerja

Buat view dengan nama **add_pekerja.php** dan simpan di folder **si-budget/application/views**.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.     <?php $this->load->view("_partials/head.php") ?>
6.     <link href="<?php echo
base_url('assets/bootstrap/css/form_input_user.css') ?>"
rel="stylesheet">
7. </head>
8.
9. <body id="page-top">
10.     <?php $this->load->view("_partials/navbar.php") ?>
11.
12.     <div id="wrapper">
13.         <?php $temp['active'] = "tables"; $this->load-
>view("_partials/sidebar.php", $temp) ?>
14.
15.         <div id="content-wrapper">
16.             <div class="container-fluid">
17.
18.                 <!-- form -->
19.                 <div class="card mb-3">
20.                     <div class="card-header">
21.                         <a href="<?php echo site_url('data_pekerja')
?>"><i class="fas fa-arrow-left"></i> Back</a>
22.                         <a style="position: absolute; right: 0"
class="pr-4" href="<?php base_url() ?>members"
type="">Import File CSV</a>
23.                     </div>
24.                     <div class="card-body">
25.
26.                         <div class="d-flex justify-content-between flex-
wrap flex-md-nowrap align-items-center pt-2 pb-2 mb-2">
27.                             <h1 class="h2 m-auto">Input form</h1>
28.                         </div>
29.
```



```

30.         <form class="form-signin mb-4"
onsubmit="success()" autocomplete="off" method="post"
action="<?php echo
base_url('index.php/admin/add_pekerja');?>" >
31.             <input type="user" name="nomor_pekerja"
id="inputnopek" class="form-control" placeholder="Nomor
Pekerja" required autofocus>
32.             <input type="user" name="nama_pekerja"
id="inputnama" class="form-control" placeholder="Nama
Pekerja" required>
33.             <input type="user" name="vp" id="inputvp"
list="vp" class="form-control" placeholder="VP" required>
34.             <datalist id="vp">
35.                 <?php
36.                     foreach($vp as $row){
37.                         echo "<option value='". $row-
>nama_vp.">";
38.                     }
39.                 ?>
40.             </datalist>
41.
42.             <input type="user" name="svp" id="inputsvp"
list="svp" class="form-control" placeholder="SVP"
required>
43.             <datalist id="svp">
44.                 <?php
45.                     foreach($svp as $row){
46.                         echo "<option value='". $row-
>nama_svp.">";
47.                     }
48.                 ?>
49.             </datalist>
50.
51.             <input type="user" name="dir" id="inputdir"
list="dir" class="form-control" placeholder="Direktorat"
required>
52.             <datalist id="dir">
53.                 <?php
54.                     foreach($dir as $row){
55.                         echo "<option value='". $row-
>nama_direktorat.">";

```

```

56.         }
57.         ?>
58.     </datalist>
59.     <button class="btn btn-lg btn-dark btn-block mt-4
form-control" type="submit">Submit</button>
60.     </form>
61. </div>
62. </div>
63. </div>
64. <!-- Sticky Footer -->
65. <?php $this->load->view("_partials/footer.php") ?>
66.
67. </div>
68. </div>
69. <?php $this->load->view("_partials/scrolltop.php") ?>
70. <?php $this->load->view("_partials/modal.php") ?>
71. <?php $this->load->view("_partials/js.php") ?>
72.
73. <script type="text/javascript">
74.     function success()
75.     {
76.         alert("Input success");
77.     }
78. </script>
79. </body>
80. </html>

```

6.6 Membuat Halaman Data Pelatihan di Admin

6.6.1 Controller

6.6.2 Model

6.5.3 View Data Pelatihan

Buat view dengan nama **table_pelatihan_admin.php** dan simpan di folder **si-budget/application/views**.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.     <?php $this->load->view("_partials/head.php") ?>
6.     <link href="<?php echo
base_url('assets/bootstrap/css/form_input_user.css') ?>"
rel="stylesheet">
7.     <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/sweetalert/sweetalert.css'); ?>">
8.     <script type="text/javascript" src="<?php echo
base_url('assets/jquery.js'); ?>"></script>
9.     <script type="text/javascript" src="<?php echo
base_url('assets/sweetalert/sweetalert.min.js');
?>"></script>
10. </head>
11.
12. <body id="page-top">
13.     <?php $this->load->view("_partials/navbar.php") ?>
14.
15.     <div id="wrapper">
16.         <?php $temp['active'] = "tables"; $this->load-
>view("_partials/sidebar.php", $temp) ?>
17.
18.         <div id="content-wrapper">
19.             <div class="container-fluid">
20.
21.                 <!-- table -->
22.                 <div class="card mb-3">
23.                     <div class="card-header">
24.                         <i class="fas fa-table"></i>
25.                         Pelatihan<a style="position: absolute; right:
0" class="pr-4" href="<?php echo
base_url('add_pelatihan'); ?>"><?php echo
base_url('add_pelatihan'); ?>"><i class="fas fa-plus"></i>
Add New</a></div>
26.                 <div class="card-body">
```

```

27.         <div class="table-responsive">
28.             <table class="table table-bordered"
id="dataTable" width="100%" cellpadding="0">
29.                 <thead>
30.                     <tr>
31.                         <th>Nama pelatihan</th>
32.                         <th>Kode pelatihan</th>
33.                         <th></th>
34.                     </tr>
35.                 </thead>
36.                 <tbody>
37.                     <?php
38.                         foreach($pelatihan as $row)
39.                         {
40.                             echo "<tr data-id='". $row->
id_pelatihan.">";
41.                             echo "<td>". $row->
nama_pelatihan."</td>";
42.                             echo "<td>". $row->
kode_pelatihan."</td>";
43.                             ?>
44.                             <td class="text-center"><button
class='btn btn-xs btn-danger delete-pelatihan' data-
id='<?php echo $row->id_pelatihan
?>'>Delete</button></td>
45.                             </tr>
46.                             <?php
47.                                 }
48.                             ?>
49.                         </tbody>
50.                     </table>
51.                 </div>
52.             </div>
53.         </div>
54.     </div>
55.     <!-- Sticky Footer -->
56.     <?php $this->load->view("_partials/footer.php") ?>
57.
58. </div>
59. </div>
60. <?php $this->load->view("_partials/scrolltop.php") ?>

```

```

61. <?php $this->load->view("_partials/modal.php") ?>
62. <?php $this->load->view("_partials/js.php") ?>
63. <script type="text/javascript">
64. $(function(){
65. $.ajaxSetup({
66.     type:"post",
67.     cache:false,
68.     dataType: "json"
69. })
70.
71. $(document).on("click",".delete-pelatihan",function(){
72.     var id=$(this).attr("data-id");
73.     swal({
74.         title:"Delete Pelatihan",
75.         text:"Are you sure?",
76.         type: "warning",
77.         showCancelButton: true,
78.         confirmButtonText: "Yes",
79.         closeOnConfirm: true,
80.     },
81.     function(){
82.         $.ajax({
83.             url:"<?php echo
base_url('index.php/admin/delete_pelatihan'); ?>",
84.             data:{id:id},
85.             success: function(){
86.                 $("tr[data-
id='"+id+"'").fadeOut("fast",function(){
87.                     $(this).remove();
88.                 });
89.             }
90.         });
91.     });
92. });
93. });
94. </script>
95. </body>
96. </html>

```

6.5.4 View Tambah Data Pelatihan

Buat view dengan nama **add_pelatihan.php** dan simpan di folder **si-budget/application/views**.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.     <?php $this->load->view("_partials/head.php") ?>
6.     <link href="<?php echo
base_url('assets/bootstrap/css/form_input_user.css') ?>"
rel="stylesheet">
7. </head>
8.
9. <body id="page-top">
10.     <?php $this->load->view("_partials/navbar.php") ?>
11.
12.     <div id="wrapper">
13.         <?php $temp['active'] = "tables"; $this->load-
>view("_partials/sidebar.php", $temp) ?>
14.
15.         <div id="content-wrapper">
16.             <div class="container-fluid">
17.
18.                 <!-- form -->
19.                 <div class="card mb-3">
20.                     <div class="card-header">
21.                         <a href="<?php echo
site_url('data_pelatihan') ?>"><i class="fas fa-arrow-
left"></i> Back</a>
22.                     </div>
23.                     <div class="card-body">
24.
25.                         <div class="d-flex justify-content-between flex-
wrap flex-md-nowrap align-items-center pt-2 pb-2 mb-2">
26.                             <h1 class="h2 m-auto">Input form</h1>
27.                         </div>
28.
29.                         <form class="form-signin mb-4"
onsubmit="success()" autocomplete="off" method="post">
```

```

        action="<?php echo
        base_url('index.php/admin/add_pelatihan');?>" >
30.         <input type="user" name="kode_pelatihan"
        class="form-control" placeholder="Kode Pelatihan"
        required autofocus>
31.         <input type="user" name="nama_pelatihan"
        class="form-control" placeholder="Nama Pelatihan"
        required>
32.         <button class="btn btn-lg btn-dark btn-block
        mt-4 form-control" type="submit">Submit</button>
33.     </form>
34. </div>
35. </div>
36. </div>
37.     <!-- Sticky Footer -->
38.     <?php $this->load->view("_partials/footer.php") ?>
39.
40. </div>
41. </div>
42. <?php $this->load->view("_partials/scrolltop.php") ?>
43. <?php $this->load->view("_partials/modal.php") ?>
44. <?php $this->load->view("_partials/js.php") ?>
45. <script type="text/javascript">
46.     function success()
47.     {
48.         alert("Input success");
49.     }
50. </script>
51. </body>
52. </html>

```

6.7 Membuat Halaman Data Provider di Admin

6.7.1 Controller

6.7.2 Model

6.6.2 View Data Provider

Buat view dengan nama **table_provider.php** dan simpan di folder **si-budget/application/views**.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.     <?php $this->load->view("_partials/head.php") ?>
6.     <link href="<?php echo
base_url('assets/bootstrap/css/form_input_user.css') ?>"
rel="stylesheet">
7.     <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/sweetalert/sweetalert.css'); ?>">
8.     <script type="text/javascript" src="<?php echo
base_url('assets/jquery.js'); ?>"></script>
9.     <script type="text/javascript" src="<?php echo
base_url('assets/sweetalert/sweetalert.min.js');
?>"></script>
10. </head>
11.
12. <body id="page-top">
13.     <?php $this->load->view("_partials/navbar.php") ?>
14.
15.     <div id="wrapper">
16.         <?php $temp['active'] = "tables"; $this->load-
>view("_partials/sidebar.php", $temp) ?>
17.
18.         <div id="content-wrapper">
19.             <div class="container-fluid">
20.
21.                 <!-- table -->
22.                 <div class="card mb-3">
23.                     <div class="card-header">
24.                         <i class="fas fa-table"></i>
```



```

25.         Provider<a style="position: absolute; right:
0" class="pr-4" href="<?php echo
base_url('add_provider');?>"><i class="fas fa-plus"></i>
Add New</a></div>
26.         <div class="card-body">
27.             <div class="table-responsive">
28.                 <table class="table table-bordered"
id="dataTable2" width="100%" cellpadding="0">
29.                     <thead>
30.                         <tr>
31.                             <th>Nama provider</th>
32.                             <th></th>
33.                         </tr>
34.                     </thead>
35.                     <tbody>
36.                         <?php foreach($provider as $row): ?>
37.                             <tr data-id='<?= $row->id_provider
?>'>
38.                                 <td><?= $row->nama_provider ?></td>
39.                                 <td class="text-center"><button
class='btn btn-xs btn-danger delete-provider' data-
id='<?= $row->id_provider ?>'>Delete</button></td>
40.                             </tr>
41.                         <?php endforeach; ?>
42.                     </tbody>
43.                 </table>
44.             </div>
45.         </div>
46.     </div>
47. </div>
48. <!-- Sticky Footer -->
49. <?php $this->load->view("_partials/footer.php") ?>
50.
51. </div>
52. </div>
53. <?php $this->load->view("_partials/scrolltop.php") ?>
54. <?php $this->load->view("_partials/modal.php") ?>
55. <?php $this->load->view("_partials/js.php") ?>
56. <script type="text/javascript">
57. $(function() {
58. $.ajaxSetup({

```

```

59.     type:"post",
60.     cache:false,
61.     dataType: "json"
62. })
63.
64. $(document).on("click",".delete-provider",function(){
65.     var id=$(this).attr("data-id");
66.     swal({
67.         title:"Delete Provider",
68.         text:"Are you sure?",
69.         type: "warning",
70.         showCancelButton: true,
71.         confirmButtonText: "Yes",
72.         closeOnConfirm: true,
73.     },
74.     function(){
75.         $.ajax({
76.             url:"<?php echo
base_url('index.php/admin/delete_provider'); ?>",
77.             data:{id:id},
78.             success: function(){
79.                 $("tr[data-
id='"+id+"'").fadeOut("fast",function(){
80.                     $(this).remove();
81.                 });
82.             }
83.         });
84.     });
85. });
86. });
87. </script>
88. </body>
89. </html>

```

6.6.2 View Tambah Data Provider

Buat view dengan nama **add_provider.php** dan simpan di folder **si-budget/application/views**.

```

1. <!DOCTYPE html>

```

```

2.  <html lang="en">
3.
4.  <head>
5.      <?php $this->load->view("_partials/head.php") ?>
6.      <link href="<?php echo
base_url('assets/bootstrap/css/form_input_user.css') ?>"
rel="stylesheet">
7.  </head>
8.
9.  <body id="page-top">
10.     <?php $this->load->view("_partials/navbar.php") ?>
11.
12.     <div id="wrapper">
13.         <?php $temp['active'] = "tables"; $this->load-
>view("_partials/sidebar.php", $temp) ?>
14.
15.         <div id="content-wrapper">
16.             <div class="container-fluid">
17.
18.                 <!-- form -->
19.                 <div class="card mb-3">
20.                     <div class="card-header">
21.                         <a href="<?php echo site_url('data_provider')
?>"><i class="fas fa-arrow-left"></i> Back</a>
22.                     </div>
23.                     <div class="card-body">
24.
25.                         <div class="d-flex justify-content-between flex-
wrap flex-md-nowrap align-items-center pt-2 pb-2 mb-2">
26.                             <h1 class="h2 m-auto">Input form</h1>
27.                         </div>
28.
29.                         <form class="form-signin mb-4"
onsubmit="success()" autocomplete="off" method="post"
action="<?php echo
base_url('index.php/admin/add_provider');?>" >
30.                             <input type="user" name="provider"
id="inputprovider" class="form-control" placeholder="Nama
Provider" required autofocus>
31.

```

```

32.         <button class="btn btn-lg btn-dark btn-block mt-4
form-control" type="submit">Submit</button>
33.         </form>
34.
35.     </div>
36. </div>
37. </div>
38.     <!-- Sticky Footer -->
39.     <?php $this->load->view("_partials/footer.php") ?>
40.
41. </div>
42. </div>
43. <?php $this->load->view("_partials/scrolltop.php") ?>
44. <?php $this->load->view("_partials/modal.php") ?>
45. <?php $this->load->view("_partials/js.php") ?>
46. <script type="text/javascript">
47.     function success()
48.     {
49.         alert("Input success");
50.     }
51. </script>
52. </body>
53. </html>

```

6.8 Membuat Halaman Import Data Pekerja

6.8.1 Libraries

Buat library dengan nama **CSVReader.php** dan simpan di folder **si-budget/application/Libraries**.

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access
allowed');
3.
4. /**
5.  * CSV Reader for CodeIgniter 3.x
6.  *

```

```

7.      * Library to read the CSV file. It helps to import a CSV
      file
8.      * and convert CSV data into an associative array.
9.      *
10.     * This library treats the first row of a CSV file
11.     * as a column header row.
12.     *
13.     *
14.     * @package      CodeIgniter
15.     * @category      Libraries
16.     * @author        CodexWorld
17.     * @license        http://www.codexworld.com/license/
18.     * @link          http://www.codexworld.com
19.     * @version        3.0
20.     */
21. class CSVReader {
22.
23.     // Columns names after parsing
24.     private $fields;
25.     // Separator used to explode each line
26.     private $separator = ',';
27.     // Enclosure used to decorate each field
28.     private $enclosure = '"';
29.     // Maximum row size to be used for decoding
30.     private $max_row_size = 4096;
31.
32.     /**
33.      * Parse a CSV file and returns as an array.
34.      *
35.      * @access      public
36.      * @param      filepath      string      Location of the
CSV file
37.      *
38.      * @return mixed|boolean
39.      */
40.     function parse_csv($filepath){
41.
42.         // If file doesn't exist, return false
43.         if (!file_exists($filepath)){
44.             return FALSE;
45.         }

```

```

46.
47.         // Open uploaded CSV file with read-only mode
48.         $csvFile = fopen($filepath, 'r');
49.
50.         // Get Fields and values
51.         $this->fields = fgetcsv($csvFile, $this-
>max_row_size, $this->separator, $this->enclosure);
52.         $keys_values = explode(',', $this->fields[0]);
53.         $keys = $this->escape_string($keys_values);
54.
55.         // Store CSV data in an array
56.         $csvData = array();
57.         $i = 1;
58.         while(($row = fgetcsv($csvFile, $this-
>max_row_size, $this->separator, $this->enclosure)) !==
FALSE) {
59.             // Skip empty lines
60.             if($row != NULL) {
61.                 $values = explode(',', $row[0]);
62.                 if(count($keys) == count($values)) {
63.                     $arr          = array();
64.                     $new_values = array();
65.                     $new_values = $this-
>escape_string($values);
66.                     for($j = 0; $j < count($keys); $j++){
67.                         if($keys[$j] != ""){
68.                             $arr[$keys[$j]] =
$new_values[$j];
69.                         }
70.                     }
71.                     $csvData[$i] = $arr;
72.                     $i++;
73.                 }
74.             }
75.         }
76.         // Close opened CSV file
77.         fclose($csvFile);
78.
79.         return $csvData;
80.     }
81.

```

```

82.     function escape_string($data){
83.         $result = array();
84.         foreach($data as $row){
85.             $result[] = str_replace("'", '', $row);
86.         }
87.         return $result;
88.     }
89. }

```

6.8.2 Controller

Buat controller dengan nama **Import.php** dan simpan di folder **si-budget/application/controllers**.

```

1.  <?php
2.  if ( ! defined('BASEPATH')) exit('No direct script
   access allowed');
3.
4.  class Members extends CI_Controller {
5.
6.      function __construct() {
7.          parent::__construct();
8.
9.          // Load member model
10.         $this->load->model('member');
11.
12.         // Load form validation library
13.         $this->load->library('form_validation');
14.
15.         // Load file helper
16.         $this->load->helper('file');
17.     }
18.
19.     public function index(){
20.         $data = array();
21.
22.         // Get messages from the session
23.         if($this->session->userdata('success_msg')){
24.             $data['success_msg'] = $this->session-
>userdata('success_msg');

```

```

25.         $this->session-
>unset_userdata('success_msg');
26.     }
27.     if($this->session->userdata('error_msg')){
28.         $data['error_msg'] = $this->session-
>userdata('error_msg');
29.         $this->session-
>unset_userdata('error_msg');
30.     }
31.
32.     // Get rows
33.     $data['members'] = $this->member->getRows();
34.
35.     // Load the list page view
36.     $this->load->view('member/index', $data);
37. }
38.
39. public function import(){
40.     $data = array();
41.     $memData = array();
42.
43.     // If import request is submitted
44.     if($this->input->post('importSubmit')){
45.         // Form field validation rules
46.         $this->form_validation->set_rules('file',
'CSV file', 'callback_file_check');
47.
48.         // Validate submitted form data
49.         if($this->form_validation->run() == true){
50.             $insertCount = $updateCount = $rowCount
= $notAddCount = 0;
51.
52.             // If file uploaded
53.
54.             if(is_uploaded_file($_FILES['file']['tmp_name'])) {
55.                 // Load CSV reader library
56.                 $this->load->library('CSVReader');
57.
58.                 // Parse data from CSV file
59.                 $csvData = $this->csvreader-
>parse_csv($_FILES['file']['tmp_name']);

```



```

59.
60.             // Insert/update CSV data into
        database
61.             if(!empty($csvData)){
62.                 foreach($csvData as $row){
$rowCount++;
63.
64.             // Prepare data for DB
        insertion
65.                 $memData = array(
66.                     'id_pekerja' =>
$row['id_pekerja'],
67.                     'nama_pekerja' =>
$row['nama_pekerja'],
68.                     'nomor_pekerja' =>
$row['nomor_pekerja'],
69.                     'vp_pekerja' =>
$row['vp_pekerja'],
70.                     'svp_pekerja' =>
$row['svp_pekerja'],
71.                     'direktorat_pekerja' =>
$row['direktorat_pekerja'],
72.                     'used_budget' =>
$row['used_budget'],
73.                 );
74.
75.             // Check whether email
        already exists in the database
76.                 $con = array(
77.                     'where' => array(
78.                         'id_pekerja' =>
$row['id_pekerja']
79.                     ),
80.                     'returnType' => 'count'
81.                 );
82.                 $prevCount = $this->member-
>getRows($con);
83.
84.                 if($prevCount > 0){
85.                     // Update member data
86.                     $condition = array(

```

```

87.                                     'id_pekerja' =>
    $row['id_pekerja']
88.                                     );
89.                                     $update = $this-
>member->update($memData, $condition);
90.
91.                                     if($update){
92.                                         $updateCount++;
93.                                     }
94.                                     }else{
95.                                         // Insert member data
96.                                         $insert = $this-
>member->insert($memData);
97.
98.                                         if($insert){
99.                                             $insertCount++;
100.                                         }
101.                                     }
102.                                     }
103.
104.                                     // Status message with imported
data count
105.                                     $notAddCount = ($rowCount -
($insertCount + $updateCount));
106.                                     $successMsg = ' Total Rows
('.$rowCount.') | Inserted ( '.$insertCount.') | Updated
('.$updateCount.') | Not Inserted ( '.$notAddCount.')';
107.                                     $this->session-
>set_userdata('success_msg', $successMsg);
108.                                     }
109.                                     }else{
110.                                         $this->session-
>set_userdata('error_msg', 'Error on file upload,
please try again. ');
111.                                     }
112.                                     }else{
113.                                         $this->session-
>set_userdata('error_msg', 'Invalid file, please select
only CSV file. ');
114.                                     }
115.                                     }

```

```

116.         redirect('members');
117.     }
118.
119.     /*
120.      * Callback function to check file value and type
        during validation
121.      */
122.     public function file_check($str){
123.         $allowed_mime_types = array('text/x-comma-
        separated-values', 'text/comma-separated-values',
        'application/octet-stream', 'application/vnd.ms-excel',
        'application/x-csv', 'text/x-csv', 'text/csv',
        'application/csv', 'application/excel',
        'application/vnd.msexcel', 'text/plain');
124.         if(isset($_FILES['file']['name']) &&
        $_FILES['file']['name'] != ""){
125.             $mime =
        get_mime_by_extension($_FILES['file']['name']);
126.             $fileAr = explode('.',
        $_FILES['file']['name']);
127.             $ext = end($fileAr);
128.             if(($ext == 'csv') && in_array($mime,
        $allowed_mime_types)){
129.                 return true;
130.             }else{
131.                 $this->form_validation-
        >set_message('file_check', 'Please select only CSV file
        to upload.');
```

6.8.3 Model

Buat Model dengan nama **exim_model.php** dan simpan di folder **si-budget/application/models**.

```
1. <?php
2. if ( ! defined('BASEPATH')) exit('No direct script access
   allowed');
3.
4. class Member extends CI_Model{
5.
6.     function __construct() {
7.         // Set table name
8.         $this->table = 'pekerja';
9.     }
10.
11.     /*
12.      * Fetch members data from the database
13.      * @param array filter data based on the passed
   parameters
14.      */
15.     function getRows($params = array()){
16.         $this->db->select('*');
17.         $this->db->from($this->table);
18.
19.         if(array_key_exists("where", $params)){
20.             foreach($params['where'] as $key => $val){
21.                 $this->db->where($key, $val);
22.             }
23.         }
24.
25.         if(array_key_exists("returnType", $params) &&
   $params['returnType'] == 'count'){
26.             $result = $this->db->count_all_results();
27.         }else{
28.             if(array_key_exists("id_pekerja", $params)){
29.                 $this->db->where('id_pekerja',
   $params['id_pekerja']);
30.                 $query = $this->db->get();
31.                 $result = $query->row_array();
32.             }else{
```

```

33.             $this->db->order_by('id_pekerja',
'desc');
34.             if(array_key_exists("start",$params) &&
array_key_exists("limit",$params)){
35.                 $this->db-
>limit($params['limit'],$params['start']);
36.             }elseif(!array_key_exists("start",$params) &&
array_key_exists("limit",$params)){
37.                 $this->db->limit($params['limit']);
38.             }
39.
40.             $query = $this->db->get();
41.             $result = ($query->num_rows() >
0)?$query->result_array():FALSE;
42.         }
43.     }
44.
45.     // Return fetched data
46.     return $result;
47. }
48.
49. /*
50.  * Insert members data into the database
51.  * @param $data data to be insert based on the passed
parameters
52.  */
53.     public function insert($data = array()) {
54.         if(!empty($data)){
55.             // Add created and modified date if not
included
56.             if(!array_key_exists("created", $data)){
57.                 $data['created'] = date("Y-m-d H:i:s");
58.             }
59.             if(!array_key_exists("modified", $data)){
60.                 $data['modified'] = date("Y-m-d H:i:s");
61.             }
62.
63.             // Insert member data
64.             $insert = $this->db->insert($this->table,
$data);

```

```

65.
66.         // Return the status
67.         return $insert?$this->db->insert_id():false;
68.     }
69.     return false;
70. }
71.
72. /*
73.  * Update member data into the database
74.  * @param $data array to be update based on the
    passed parameters
75.  * @param $condition array filter data
76.  */
77.     public function update($data, $condition = array()) {
78.         if(!empty($data)){
79.             // Add modified date if not included
80.             if(!array_key_exists("modified", $data)){
81.                 $data['modified'] = date("Y-m-d H:i:s");
82.             }
83.
84.             // Update member data
85.             $update = $this->db->update($this->table,
    $data, $condition);
86.
87.             // Return the status
88.             return $update?true:false;
89.         }
90.         return false;
91.     }
92. }

```

6.8.4 View

Buat view dengan nama **login.php** dan simpan di folder **si-budget/application/controllers**.

```

1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>

```

```

5.      <!-- <meta charset="utf-8"> -->
6.      <?php $this->load->view("_partials/head.php") ?>
7.      <title>CodeIgniter CSV Import</title>
8.
9.      <!-- Bootstrap library -->
10.     <link rel="stylesheet" href="<?php echo
base_url('assets/bootstrap/bootstrap.min.css'); ?>">
11.
12.     <!-- Stylesheet file -->
13.     <link rel="stylesheet" href="<?php echo
base_url('assets/css/style.css'); ?>">
14.     <link href="<?php echo
base_url('assets/bootstrap/css/form_pelatihan.css') ?>"
rel="stylesheet">
15.     <link rel="stylesheet" type="text/css" href="<?php
echo base_url('assets/sweetalert/sweetalert.css'); ?>">
16.     <script type="text/javascript" src="<?php echo
base_url('assets/jquery.js'); ?>"></script>
17.     <script type="text/javascript" src="<?php echo
base_url('assets/sweetalert/sweetalert.min.js');
?>"></script>
18. </head>
19.
20. <body id="page-top">
21.     <?php $this->load->view("_partials/navbar.php") ?>
22.
23.     <div id="wrapper">
24.         <?php $temp['active'] = "tables"; $this->load-
>view("_partials/sidebar.php", $temp) ?>
25.
26.         <div class="container-wrapper">
27.             <div class="container-fluid">
28.                 <div class="card mb-3">
29.                     <div class="card-header">
30.                         <a href="<?php echo base_url('add_pekerja')
?>"><i class="fas fa-arrow-left"></i> Back</a>
31.
32.             <!-- Display status message -->
33.             <?php if(!empty($success_msg)){ ?>
34.             <div class="col-xs-12">

```

```

35.         <div class="alert alert-success"><?php echo
           $success_msg; ?></div>
36.     </div>
37.     <?php } ?>
38.     <?php if(!empty($error_msg)){ ?>
39.         <div class="col-xs-12">
40.             <div class="alert alert-danger"><?php echo
               $error_msg; ?></div>
41.         </div>
42.         <?php } ?>
43.
44.         <div class="row">
45.             <!-- Import link -->
46.             <div class="col-md-12 head">
47.                 <div class="float-right">
48.                     <a href="javascript:void(0);" class="btn
                       btn-success" onclick="formToggle('importFrm');"><i
                       class="plus"></i> Import</a>
49.                 </div>
50.             </div>
51.
52.             <!-- File upload form -->
53.             <div class="col-md-12" id="importFrm"
               style="display: none;">
54.                 <form action="<?php echo
                   base_url('members/import'); ?>" method="post"
                   enctype="multipart/form-data">
55.                     <input type="file" name="file" />
56.                     <input type="submit" class="btn btn-
                       primary" name="importSubmit" value="IMPORT">
57.                     <br>
58.                 </form>
59.             </div>
60.             <br>
61.
62.             <!-- Data list table -->
63.             <div class="card-body">
64.                 <div class="table-responsive">
65.                     <table class="table table-bordered"
                       id="dataTable1" width="100%" cellpadding="0">
66.                         <tr>

```



```

67.             <th>ID</th>
68.             <th>Nama</th>
69.             <th>Nopek</th>
70.             <th>VP</th>
71.             <th>SVP</th>
72.             <th>Direktorat</th>
73.             <th>Used Budget</th>
74.         </tr>
75.     </thead>
76.     <tbody>
77.         <?php if(!empty($members)) {
78.             <tr>
79.                 <td><?php echo $row['id_pekerja'];
80.                 ?></td>
81.                 <td><?php echo $row['nama_pekerja'];
82.                 ?></td>
83.                 <td><?php echo $row['nomor_pekerja'];
84.                 ?></td>
85.                 <td><?php echo $row['vp_pekerja'];
86.                 ?></td>
87.                 <td><?php echo $row['svp_pekerja'];
88.                 ?></td>
89.                 <td><?php echo $row['direktorat_pekerja']; ?></td>
90.                 <td><?php echo $row['used_budget'];
91.                 ?></td>
92.             </tr>
93.             <?php } }else{ ?>
94.                 <tr><td colspan="5">No member(s)
95.                 found...</td></tr>
96.             <?php } ?>
97.         </tbody>
98.     </table>
99. </div>
100. </div>
101. </div>
102. </div>
103. </div>
104. <?php $this->load->view("_partials/footer.php") ?>
105. </div>

```

```

99. </div>
100. </div>
101.
102. <?php $this->load->view("_partials/scrolltop.php") ?>
103. <?php $this->load->view("_partials/modal.php") ?>
104. <?php $this->load->view("_partials/js.php") ?>
105.
106. <script>
107. function formToggle(ID){
108.     var element = document.getElementById(ID);
109.     if(element.style.display === "none"){
110.         element.style.display = "block";
111.     }else{
112.         element.style.display = "none";
113.     }
114. }
115. </script>
116. </body>
117. </html>

```

6.9 Membuat Halaman Reset Password

6.9.1 Controller

Buat controller dengan nama **Lupa_password.php** dan simpan di folder **si-budget/application/controllers**.

```

1. <?php
2.     defined('BASEPATH') OR exit('No direct script access
   allowed');
3.
4.     class Lupa_password extends CI_Controller {
5.
6.         public function index()
7.         {
8.             $this->load->model('M_account');
9.             $this->form_validation->set_rules('email',
   'Email', 'required|valid_email');

```

```

10.
11.         if($this->form_validation->run() == FALSE) {
12.             $data['title'] = 'Halaman Reset Password';
13.             $this->load-
14. >view('reset_password/forgot_password',$data);
15.         }else{
16.             $email = $this->input->post('email');
17.             $clean = $this->security->xss_clean($email);
18.             $userInfo = $this->M_account-
19. >getUserInfoByEmail($clean);
20.
21.             if(!$userInfo){
22.                 $this->session->set_flashdata('sukses',
23. 'Incorrect email address, please try again.');
```

```

24.                 redirect(site_url('login'),'refresh');
```

```

25.             }
26.
27.             //build token
28.             $token = $this->M_account-
29. >insertToken($userInfo->id);
30.             $qstring = $this->base64url_encode($token);
31.             $url = site_url() .
32. 'lupa_password/reset_password/token/' . $qstring;
33.             $link = '<a href="' . $url . '">' . $url .
34. '</a>';
35.
36.             $message = '';
37.             $message .= '<strong>Hi, You received this
38. e-mail because there is a request to renew
39. your password.</strong><br>';
40.             $message .= '<strong>Please click on this
41. link:</strong> ' . $link;
42.
43.             echo $message; //send this through mail
44.             exit;
45.         }
46.     }
47.
48.     // menampilkan halaman reset password berdasarkan
49.     link yang dikirim melalui email, memvalidasi token dan
50.     memperbaharui password
51.     public function reset_password()

```

```

41.     {
42.         $this->load->model('M_account');
43.         $token = $this->base64url_decode($this->uri-
>segment(4));
44.         $cleanToken = $this->security->xss_clean($token);
45.
46.         $user_info = $this->M_account-
>isTokenValid($cleanToken); //either false or array();
47.
48.         if(!$user_info){
49.             $this->session->set_flashdata('sukses', 'Token
is invalid or has expired');
50.             redirect(site_url('login'),'refresh');
51.         }
52.
53.         $data = array(
54.             'title'=> 'Halaman Reset Password',
55.             'username'=> $user_info->username,
56.             'email'=>$user_info->email,
57.             'token'=>$this->base64url_encode($token)
58.         );
59.
60.         $this->form_validation->set_rules('password',
'Password', 'required|min_length[5]');
61.         $this->form_validation->set_rules('passconf',
'Password Confirmation', 'required|matches[password]');
62.
63.         if ($this->form_validation->run() == FALSE) {
64.             $this->load-
>view('reset_password/reset_password', $data);
65.         }else{
66.
67.             $post = $this->input->post(NULL, TRUE);
68.             $cleanPost = $this->security->xss_clean($post);
69.             $hashed = md5($cleanPost['password']);
70.             $cleanPost['password'] = $hashed;
71.             $cleanPost['id'] = $user_info->id;
72.             unset($cleanPost['passconf']);
73.             if(!$this->M_account-
>updatePassword($cleanPost)){

```

```

74.         $this->session->set_flashdata('sukses',
    'Password update failed.');
```

```

75.         redirect(site_url('login'),'refresh');
```

```

76.
```

```

77.     }else{
```

```

78.         $this->session->set_flashdata('sukses', 'Your
    password already updated. Please login.');
```

```

79.         redirect(site_url('login'),'refresh');
```

```

80.
```

```

81.     }
```

```

82.     redirect(site_url('login'),'refresh');
```

```

83. }
```

```

84. }
```

```

85.     // mengenkripsi token yg dikirim ke email user
```

```

86.     public function base64url_encode($data) {
```

```

87.         return rtrim(strtr(base64_encode($data), '+/', '-_'),
    '=');
```

```

88.     }
```

```

89.
```

```

90.     // mendekripsi url yang berisi token dari link
    reset password
```

```

91.     public function base64url_decode($data) {
```

```

92.         return base64_decode(str_pad(strtr($data, '-_',
    '+/'), strlen($data) % 4, '=', STR_PAD_RIGHT));
```

```

93.     }
```

```

94. }
```

6.9.2 Model

Buat model dengan nama **M_account.php** dan simpan di folder **si-budget/application/models**.

```

1. <?php
```

```

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

```

3.     class M_Account extends CI_Model{
```

```

4.
```

```

5.         function daftar($data) {
```

```

6.             $this->db->insert('user', $data);
```

```

7.         }
```

```

8.
9.      //Start: method tambahan untuk reset code
10.     public function getUserInfo($id)
11.     {
12.         $q = $this->db->get_where('user', array('id' =>
13.         $id), 1);
14.         if($this->db->affected_rows() > 0){
15.             $row = $q->row();
16.             return $row;
17.         }else{
18.             error_log('no user found getUserInfo('.$id.')');
19.             return false;
20.         }
21.     }
22.     public function getUserInfoByEmail($email){
23.         $q = $this->db->get_where('user', array('email' =>
24.         $email), 1);
25.         if($this->db->affected_rows() > 0){
26.             $row = $q->row();
27.             return $row;
28.         }
29.     }
30.     public function insertToken($user_id)
31.     {
32.         $token = substr(sha1(rand()), 0, 30);
33.         $date = date('Y-m-d');
34.
35.         $string = array(
36.             'token'=> $token,
37.             'user_id'=>$user_id,
38.             'created'=>$date
39.         );
40.         $query = $this->db->insert_string('tokens',$string);
41.         $this->db->query($query);
42.         return $token . $user_id;
43.     }
44.
45.     public function isTokenValid($token)

```

```

47.     {
48.         $tkn = substr($token,0,30);
49.         $uid = substr($token,30);
50.
51.         $q = $this->db->get_where('tokens', array(
52.             'tokens.token' => $tkn,
53.             'tokens.user_id' => $uid), 1);
54.
55.         if($this->db->affected_rows() > 0){
56.             $row = $q->row();
57.
58.             $created = $row->created;
59.             $createdTS = strtotime($created);
60.             $today = date('Y-m-d');
61.             $todayTS = strtotime($today);
62.
63.             if($createdTS != $todayTS){
64.                 return false;
65.             }
66.
67.             $user_info = $this->getUserInfo($row->user_id);
68.             return $user_info;
69.
70.         }else{
71.             return false;
72.         }
73.
74.     }
75.
76.     public function updatePassword($post)
77.     {
78.         $this->db->where('id', $post['id']);
79.         $this->db->update('user', array('password' =>
80.             $post['password']));
81.         return true;
82.     }
83.     //End: method tambahan untuk reset code

```

6.9.3 View Forgot Password

Buat view dengan nama **forgot_password.php** dan simpan di folder **si-budget/application/views/reset_passsword**.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.
6.     <meta charset="utf-8">
7.     <link rel="icon" type="image/ico" href="<?php echo
base_url('images/favicon.ico') ?>"/>
8.     <meta http-equiv="X-UA-Compatible" content="IE=edge">
9.     <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
10.    <meta name="description" content="">
11.    <meta name="author" content="">
12.
13.    <title>Forgot Password</title>
14.
15.    <!-- Custom fonts for this template-->
16.    <link href="<?php echo base_url('assets/fontawesome-
free/css/all.min.css') ?>" rel="stylesheet"
type="text/css">
17.
18.    <!-- Custom styles for this template-->
19.    <link href="<?php echo base_url('css/sb-admin.css') ?>"
rel="stylesheet">
20.
21. </head>
22.
23. <body class="bg-dark">
24.
25.     <div class="container">
26.         <div class="card card-login mx-auto mt-5">
27.             <div class="card-header">Reset Password</div>
28.             <div class="card-body">
29.                 <div class="text-center mb-4">
30.                     <h4>Forgot your password?</h4>
```



```

31.         <p>Enter your email address and we will send
you instructions on how to reset your password.</p>
32.     </div>
33.
34.     <div class="card">
35.         <div class="card-body">
36.             <?php if ($this->session-
>flashdata('sukses')) { ?>
37.                 <div class="alert alert-success">
38.                     <?= $this->session->flashdata('sukses')
?>
39.                 </div>
40.             <?php } ?>
41.             <?= form_open('Lupa_password/index') ?>
42.             <form method="post" action="<?php echo
base_url('login/email') ?>">
43.                 <div class="form-group">
44.                     <div class="form-label-group">
45.                         <input name="email" type="email"
id="inputEmail" class="form-control" placeholder="Enter
email address" required="required" autofocus="autofocus"
value="<?php echo set_value('email'); ?>">
46.                         <label for="inputEmail">Enter email
address</label>
47.                     </div>
48.                 </div>
49.                 <button class="btn btn-dark btn-block"
type="submit">Reset Password</button>
50.             </form>
51.             <div class="text-center">
52.                 <a class="d-block small" href="<?php echo
base_url('login') ?>">Login Page</a>
53.             </div>
54.             <!-- <div class="form-group">
55.                 <label for="exampleInputEmaill">Email
address</label>
56.                 <input type="email" name="email"
class="form-control" id="exampleInputEmaill" aria-
describedby="emailHelp" placeholder="Enter email"
required>

```

```

57.         <small id="emailHelp" class="form-text
text-muted">We'll never share your email with anyone
else.</small>
58.     </div> -->
59.         <!-- <button type="submit" class="btn
btn-primary" name="submit">Submit</button> -->
60.         <?= form_close() ?>
61.     </div>
62.
63.
64.         <!-- <form method="post" action="<?php echo
base_url('index.php/login/email') ?>">
65.             <div class="form-group">
66.                 <div class="form-label-group">
67.                     <input name="email" type="email"
id="inputEmail" class="form-control" placeholder="Enter
email address" required="required" autofocus="autofocus">
68.                     <label for="inputEmail">Enter email
address</label>
69.                 </div>
70.             </div>
71.             <button class="btn btn-dark btn-block"
type="submit">Reset Password</button>
72.         </form>
73.         <div class="text-center">
74.             <a class="d-block small" href="<?php echo
base_url('login') ?>">Login Page</a>
75.         </div> -->
76.     </div>
77. </div>
78. </div>
79.
80. <!-- Bootstrap core JavaScript-->
81. <script src="<?php echo
base_url('assets/jquery/jquery.min.js') ?>"></script>
82. <script src="<?php echo
base_url('assets/bootstrap/js/bootstrap.bundle.min.js')
?>"></script>
83.
84. <!-- Core plugin JavaScript-->

```

```

85.     <script src="<?php echo base_url('assets/jquery-
      easing/jquery.easing.min.js') ?>"></script>
86.
87. </body>
88. </html>

```

6.8.4 View Reset Password

Buat view dengan nama **reset_password.php** dan simpan di folder **si-budget/application/views/reset_password**.

```

1.  <!DOCTYPE html>
2.  <html lang="en">
3.
4.  <head>
5.
6.    <meta charset="utf-8">
7.    <link rel="icon" type="image/ico" href="<?php echo
      base_url('images/favicon.ico') ?>" />
8.    <meta http-equiv="X-UA-Compatible" content="IE=edge">
9.    <meta name="viewport" content="width=device-width,
      initial-scale=1, shrink-to-fit=no">
10.   <meta name="description" content="">
11.   <meta name="author" content="">
12.
13.   <!-- Custom fonts for this template-->
14.   <link href="<?php echo base_url('assets/fontawesome-
      free/css/all.min.css') ?>" rel="stylesheet"
      type="text/css">
15.
16.   <!-- Custom styles for this template-->
17.   <link href="<?php echo base_url('css/sb-admin.css') ?>"
      rel="stylesheet">
18.
19. </head>
20.
21. <body class="bg-dark">
22.   <div class="container">
23.     <div class="card card-login mx-auto mt-5">

```

```

24.         <!-- <div class="card-header">Reset Password</div>
      -->
25.         <div class="card-body">
26.             <div class="text-center mb-4">
27.                 <h4>Reset your password</h4>
28.                 <!-- <p>Enter your email address and we will
      send you instructions on how to reset your password.</p>
      -->
29.             </div>
30.             <div class="card">
31.                 <div class="card-body">
32.                     <h5>Hello <span><?php echo $username;
      ?></span>, Please create your new password.</h5>
33.                     <?php echo
      form_open('lupa_password/reset_password/token/'.$token);
      ?>
34.                     <p>New Password:</p>
35.                     <p>
36.                         <input type="password" name="password"
      value="<?php echo set_value('password'); ?>" />
37.                     </p>
38.                     <p> <?php echo form_error('password'); ?>
      </p>
39.                     <p>Confirm Password:</p>
40.                     <p>
41.                         <input type="password" name="passconf"
      value="<?php echo set_value('passconf'); ?>" />
42.                     </p>
43.                     <p> <?php echo form_error('passconf'); ?>
      </p>
44.                     <p>
45.                         <button class="btn btn-lg btn-dark" input
      type="submit" name="btnSubmit"
      value="Reset">Reset</button>
46.                     </p>
47.
48.                     <?= form_close() ?>
49.                 </div>
50.             </div>
51.         </div>
52.     </div>

```

```
53.
54.     <!-- Bootstrap core JavaScript-->
55.     <script src="<?php echo
base_url('assets/jquery/jquery.min.js') ?>"></script>
56.     <script src="<?php echo
base_url('assets/bootstrap/js/bootstrap.bundle.min.js')
?>"></script>
57.
58.     <!-- Core plugin JavaScript-->
59.     <script src="<?php echo base_url('assets/jquery-
easing/jquery.easing.min.js') ?>"></script>
60. </body>
61. </html>
```

DAFTAR PUSTAKA

TENTANG PENULIS