

Bukuisbn

kopisharif

January 2020

1 Penggunaan Aplikasi

Kali ini kami akan memberi tahu kalian nih soal aplikasi yang kami gunakan buat bikin web ini, kami menggunakan bahasa PHP, dan HTML tak lupa desainnya menggunakan CSS dan database XAMPP dan MySQL, kami juga menggunakan SUBLIME sebagai alat bantu membaca bahasa pemrograman, lalu frameworknya kami gunakan CodeIgniter versi 3.1.11 dan juga Bootstrap versi 4 sebagai Template dan bisa juga digunakan untuk mendesain. Untuk lebih jelasnya akan kami bahas disini.

2 PHP

Apakah PHP itu? PHP (akronim dari PHP: Hypertext Preprocessor) adalah bahasa pemrograman yang berfungsi dalam membuat web yang dinamis maupun aplikasi berbentuk web. Sangat berbeda dengan HTML yang hanya dapat menampilkan konten statis, PHP dapat berinteraksi atau terhubung dengan database, folder maupun file, sehingga membuat bahasa pemrograman PHP bisa menampilkan konten yang dinamis dari Blog, web, Toko Online, Forum, dan Website JRP adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP termasuk dalam bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk dalam bahasa yang cross-platform, yang artinya PHP dapat berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran “.php”.

3 Sejarah PHP

Sejarah PHP Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. 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 perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

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

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 meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessing.

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

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Server web bawaan ditanamkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstall software server.

Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7.0.16 dan 7.1.2 yang resmi dirilis pada tanggal 17 Februari 2017. Pembatas PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun di luar pembatas tidak diproses oleh PHP (meskipun teks PHP ini masih mengendalikan struktur yang dijelaskan dalam kode PHP). Pembatas yang paling umum adalah `"<?php"` untuk membuka dan `"?>"` untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, JavaScript.

4 Variabel

Variabel diawali dengan simbol dolar. Pada versi php PHP 5 diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa mereka menjadi parameter objek dari class tertentu, array, atau fungsi. Namun, jenis petunjuk tidak dapat digunakan dengan jenis skalar seperti angka atau string. Contoh variabel dapat ditulis sebagai `namaVariabel`.

Penulisan fungsi, penamaan kelas, nama variabel adalah peka akan huruf besar (Kapital) dan huruf kecil. Kedua kutip ganda `" "` dari string memberikan kemampuan untuk interpolasi nilai variabel ke dalam string PHP. PHP menerjemahkan baris sebagai spasi, dan pernyataan harus diakhiri dengan titik koma `;`. beberapa contoh aplikasi yang menggunakan php

1. Facebook



Figure 1: Caption

2. Yahoo



Figure 2: Caption

3. Wikipedia



Figure 3: Caption

5 CI

merupakan sebuah framework PHP yang bisa mempercepat pengembang untuk membuat sebuah aplikasi web. Dilengkapi banyak library dan helper yang berguna di dalamnya dan tentunya mempermudah proses development. Sedangkan Twitter Bootstrap adalah sebuah alat bantu (framework) HTML dan CSS untuk membuat sebuah tampilan halaman website yang elegan dan support segala macam device. Dengan implementasi Twitter Bootstrap pada framework CodeIgniter akan mempermudah dan mempercepat pembuatan maupun pengembangan website dinamis tanpa harus kesulitan membuat desain.

6 Sejarah CI

CodeIgniter adalah sebuah framework berbasis PHP yang kuat dengan footprint yang sangat kecil, dibangun untuk pengembang yang membutuhkan toolkit

sederhana dan elegan untuk membuat aplikasi web dengan fitur lengkap. Framework atau dalam bahasa Indonesia dapat diartikan sebagai "kerangka kerja" merupakan sebuah tool yang bekerja pada suatu konsep tertentu dan terdiri dari berbagai fungsi yang dapat dengan mudah dijabarkan atau dialih-gunakan untuk membuat fungsi-fungsi lain yang lebih kompleks. Maksudnya, jika seorang Programmer menggunakan sebuah framework sebagai lingkup kerjanya, dia harus mematuhi segala ketentuan dari framework tersebut. Lebih lanjut ia dapat membangun berbagai fungsi yang rumit dan kompleks dengan menggunakan fungsi-fungsi yang telah disediakan oleh Framework tersebut. Jadi dia tidak perlu lagi menulis ulang semua coding, ia cukup mempelajari cara menggunakan kerangka (frame) yang telah tersedia dari framework tersebut.

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website-dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.4

CodeIgniter pertama kali ditulis oleh Rick Ellis (<http://www.ellislab.com>), seorang musisi rock yang beralih profesi menjadi programmer dalam riset kecil-kecilannya dan menghasilkan suatu framework PHP yang berukuran kecil, ringan serta memenuhi fitur umum aplikasi PHP. Namun, sejak tahun 2014 CodeIgniter telah dimiliki oleh British Columbia Institute of Technology (BCIT). Segala hal mengenai CodeIgniter dapat ditemui di websiter resminya beralamatkan <http://www.codeigniter.com>. Untuk saat ini telah dirilis versi 3.0.6.

1. Download CodeIgniter: CodeIgniter dapat diunduh secara gratis melalui situs resminya di <https://codeigniter.com/> Pada artikel ini versi yang digunakan adalah 3.1.11
2. Instalasi CodeIgniter: Karena CodeIgniter terdiri dari file dan folder PHP maka hal yang perlu dilakukan dalam instalasi CodeIgniter hanyalah mengekstrak file yang di unduh dari <http://ellislab.com/codeigniter/user-guide/> dan meletakkan pada direktori web server, Misalkan untuk Windows pada : C:/xamp/htdocs/ atau /var/www pada Linux. Kemudian pada file config.php yang berada pada direktori application/config/config.php lalu set base URL anda, misalkan : <http://localhost/ci/> . Jika aplikasi website yang anda buat akan menggunakan database buatlah koneksi database dengan mengedit file database.php yang berada pada aplikasi/config/database.php. Anda juga bisa menjalankan auto-load modul yang akan anda gunakan misalkan session, libraries, dll dengan mengedit file autoload.php kemudian untuk menentukan controller pertama yang dijalankan terletak pada routes.php.
3. memahami MVC CodeIgniter sendiri dibangun menggunakan konsep Model-View-Controller development pattern. Manfaat konsep MVC ini adalah memudahkan logic dalam coding aplikasi tersebut dan memudahkan pengembangan sistem yang sudah ada.

- (a) Model berhubungan dengan data dan interaksi ke database atau web-service. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL.
- (b) View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript.
- (c) Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model.
- (d) Help dan Tutorial Pendukung CodeIgniter Jika anda masih merasa bingung dan kesulitan memahami konsep CodeIgniter, cobalah search tutorial di google yang jumlahnya tidak terhitung. Akan tetapi CodeIgniter sendiri menyediakan dokumentasi yang sangat lengkap

7 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMP merupakan singkatan dari X (empat system operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis dan berikut tampilan UI pada xampp

CodeIgniter 2.0 CodeIgniter versi 2.0 baru saja diluncurkan pada awal tahun 2011. Pada versi terbaru ini, terdapat banyak perubahan mendasar dari CodeIgniter daripada versi sebelumnya. Perubahan ini menjadikan CodeIgniter jauh lebih kaya dan matang dibandingkan framework lainnya. Perubahan itu diantaranya adalah: • Menghilangkan dukungan terhadap PHP4. PHP4 sudah tidak didukung oleh tim pengembang PHP, karena memberikan dukungan untuk PHP4 membuat CodeIgniter semakin ketinggalan dari segi fitur. • Menghilangkan fitur plugin. Plugin mirip dengan helper, bertujuan untuk menghilangkan kerancuan ini maka fitur ini dihilangkan pada CodeIgniter 2.0. • Menghilangkan fitur scaffolding. Fitur ini nyaris tidak pernah digunakan dan implementasinya masih kurang bagus. • Penambahan library driver. Ini adalah library khusus



Figure 4: Caption

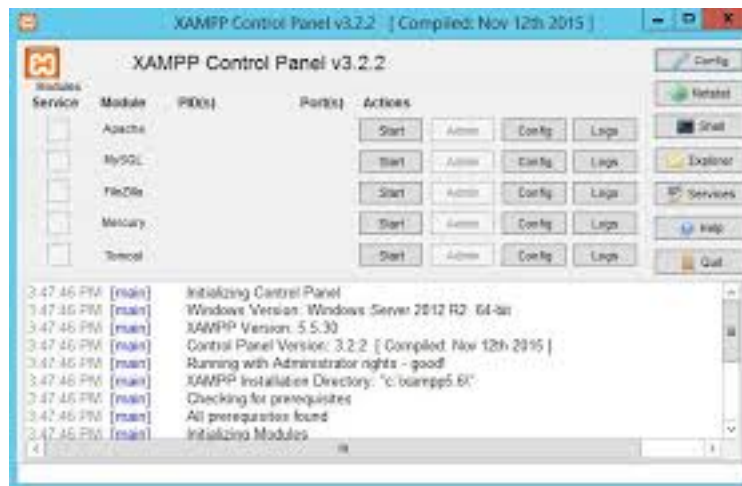


Figure 5: Caption

dimana kita dapat membuat driver dari library yang telah kita buat. • Support query string dan command-line execution. Hal ini menjawab kesulitan-kesulitan yang dialami pada versi sebelumnya. • Penambahan library cache. Untuk meningkatkan kualitas aplikasi maka library cache baik menggunakan apc, memcached maupun file base. • Penambahan fitur package. Untuk mempermudah distribusi resource dalam sebuah folder.

Apa itu MVC?

MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter . MVC adalah singkatan dari Model View Controller. MVC sebenarnya adalah sebuah pattern/teknik pemogramanan yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Adapun komponen-komponen MVC antara lain: 1. Model Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresen-

tasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL. 2. View View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller 3. Controller: Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

Gambar 2. Perbandingan PHP Biasa dan CodeIgniter

Perhatikan gambar 2. Pada eksekusi PHP, biasanya kita akan me-“load” semua library dan fungsi yang dibutuhkan kemudian digabungkan ke dalam HTML untuk di eksekusi oleh PHP. Untuk kasus sederhana cara tersebut masih baik-baik saja, tetapi ketika aplikasi tersebut menjadi kompleks/rumit maka kita akan sulit memeliharanya jika tidak didukung oleh arsitektur software yang bagus. Hal tersebut bisa terjadi disebabkan oleh code yang sama namun dibuat berulang-ulang, kode tidak konsisten dan lain-lain.

Gambar 3. Flow CodeIgniter

Jika dipetakan, alur kerja CodeIgniter akan tampak seperti gambar 3. Browser berinteraksi melalui controller. Controller-lah yang akan menerima dan membalas semua request dari browser. Untuk data maka controller akan meminta ke Model dan untuk UI/template akan meminta ke View. Jadi “Otak” dari aplikasi ada di controller, “Muka” aplikasi ada di view dan “Data” ada di model. Ketika

browser meminta sebuah halaman web maka router akan mencari controller mana yang harus menangani request tersebut. Setelah itu barulah si controller menggunakan model untuk mengakses data dan View untuk menampilkan data tersebut. Jangan Belajar CodeIgniter!!

Jangan coba-coba belajar CodeIgniter ketika Anda belum mengenal PHP. Berdasarkan pengalaman di forum dan milis, kebanyakan mereka yang belajar CodeIgniter tanpa memiliki dasar PHP yang baik akan mengalami banyak kesulitan, bukan dalam menguasai konsep CodeIgniter tetapi masih berkutat seputar PHP. Setidaknya Anda telah memahami konsep OOP pada PHP untuk

mulai belajar CodeIgniter. Sebaiknya ketika Anda mempelajari CodeIgniter Anda sudah memahami PHP dan Object Oriented Programming. Akan lebih bagus lagi jika Anda sudah familiar dengan design pattern.

Chapter 2 PHP Object Oriented Programming Syarat utama untuk menguasai CodeIgniter adalah memahami PHP dan konsep Object Oriented Programming. Pada bab ini kita akan membahas sedikit mengenai PHP dan OOP. Apasih PHP? PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman di sisi server. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server. Misalnya kita merequest sebuah file PHP `http://www.koder.web/index.php`, maka webserver akan melakukan parsing terhadap file PHP tersebut. PHP parser yang menjalankan kode-kode PHP yang terdapat pada file `index.php` lalu mengirimkan hasilnya ke web browser. Sejarah PHP Pada tahun 1994, Rasmus Lerdorf mengembangkan sebuah perangkat yang digunakan sebagai engine parsing sebagai penerjemah/interpreter beberapa macro. Pada saat itu engine digunakan untuk pembuatan buku tamu, counter dan beberapa homepage. Ia menamai engine parser tersebut dengan nama PHP/FI. Dengan semangat open source, para pengembang di dunia mencoba mengembangkan PHP/FI. Sampai pada tahun 1997, lebih dari 500.000 website di dunia menggunakan PHP/FI untuk menyelesaikan masalah seperti koneksi ke database, menampilkan content yang dinamis dan lain-lain. Pada juni 1998, PHP 3.0 dirilis. Pada saat itu PHP sudah mendukung multplatform (bukan hanya linux), webserver, sejumlah database, SNMP (Simple Network Management Protocol) and IMAP (Internet Message Access Protocol). Menurut survei yang dilakukan oleh `http://netcraft.org`, saat ini pengguna PHP sudah mencapai 9,5 juta domain. Menggunakan PHP Ketika kita ingin mempelajari PHP, hal pertama yang harus disiapkan adalah sebuah web server, seperti yang sudah dibahas sebelumnya bahwa PHP merupakan bahasa pemrograman di sisi server. Banyak sekali jenis web server yang dapat digunakan, antara lain Apache, IIS, iPlanet, Omni, Xintami, dan lain-lain. Setelah web server terinstal barulah kita install PHP sebagai modul.

PHP dapat diperoleh secara gratis dengan cara men-download dari situs resmi PHP (`http://www.php.net/downloads.php`) atau website lain yang mempunyai salinan program PHP untuk di-download. Sebelum menginstall PHP, terlebih dahulu harus menginstall web server. Setelah PHP terinstall. Anda dapat meletakkan source code PHP ke dalam folder yang akan di publish oleh web server dengan ekstensi file `.php`. Jika tidak ingin direpotkan oleh proses konfigurasi dan implementasi webserver, Anda dapat menggunakan software yang sudah terpaket menjadi satu antara PHP, APACHE, dan MySQL. Contoh Software-nya adalah XAMPP, PHP TRIAD, FOX Server, dan lain-lain.

Apa sih Object Oriented Programming (OOP)? Object Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada obyek. Semua data dan fungsi pada paradigma ini dibungkus dalam kelas-kelas atau obyek-obyek. Bandingkan dengan logika pemrograman terstruktur, setiap obyek dapat menerima pesan, memproses data, dan mengirim pesan ke obyek lainnya. OOP diciptakan untuk mengatasi keterbatasan pada bahasa pemrograman tradisional. Konsep dari OOP sendiri adalah semua pemecahan masalah

dibagi ke dalam obyek. Dalam konsep OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai obyek. Apa itu Object Sederhananya, sebuah obyek adalah kumpulan dari variabel dan fungsi yang dibungkus menjadi satu entitas. Entitas tersebut dapat berupa variabel biasa. Sebuah obyek diciptakan melalui sebuah kelas atau dengan istilah *instance of class*. Obyek memiliki 2 elemen utama: 1. *Attributes* atau *Properties*: Yaitu nilai-nilai yang tersimpan dalam objek tersebut dan secara langsung maupun tidak langsung menentukan karakteristik dari obyek tersebut. 2. *Method*: Yaitu suatu aksi yang akan dijalankan atau dikerjakan oleh obyek tersebut.

Apa itu Class? Class dapat didefinisikan sebagai struktur data atau cetak biru dari suatu obyek. Lebih jelasnya adalah sebuah bentuk dasar atau blueprint yang mendefinisikan variabel, method umum pada semua obyek. Obyek sendiri adalah kumpulan variabel dan fungsi yang dihasilkan dari template khusus atau disebut class. Obyek adalah elemen pada saat run-time yang akan diciptakan, dimanipulasi, dan dibuang/di-destroy ketika eksekusi. Adapun class merupakan definisi statik dari himpunan obyek yang mungkin diciptakan sebagai instantiasi dari class. Perhatikan contoh class di berikut ini.

```

<?php /** Contoh kelas */ class Kendaraan /** end of class */ ?>

```

Contoh diatas memperlihatkan bagaimana mendefinisikan sebuah class dan meng-create sebuah instance dari class. Pada contoh di atas kita membuat sebuah kelas bernama “Kendaraan”. Dalam pembuatan kelas, pertama kita menggunakan kata kunci class yang diikuti oleh nama kelas, kemudian diakhiri dengan kurung kurawal. Di dalam kurung kurawal kita menuliskan kode-kode (berisi property dan method) supaya kelas tersebut bekerja seperti yang diinginkan. Kode-kode di dalam sebuah kelas terbagi menjadi dua kelompok, yaitu property dan method. Property adalah suatu wadah penyimpanan di dalam kelas yang bisa menampung informasi. Sederhananya property itu bisa disebut sebagai variabel di dalam kelas. Sedangkan method adalah fungsi yang ada di dalam kelas. Perhatikan contoh berikut.

```

<?php /** contoh kelas */ class kendaraan
    /** property class */ private $warna; private $jumlah_pintu; private $jumlah_roda; public $harga;
    public $merk;
    /** method class */ public function construct() { echo 'ini adalah object kendaraan. \n';
        public function set_harga($harga) { $this->harga = $harga;
        public function show_harga() { echo 'hargakendaraan : ' . $this->harga . ' \n';
        public function jalan() { echo 'brrrroooooom!!!';

```

```

    /** end of class */
    /** contoh object */
    #mengcreate object $saya_adalah_object dari class kendaraan
    $saya_adalah_object = new kendaraan;

```

```

$saya_adalah_object->set_harga(100000);
$saya_adalah_object->show_harga();

$saya_adalah_object->jalan();

```

Setiap property dan method memiliki identifier. Identifier-lah yang mengatur bagaimana property dan method digunakan. Identifier tersebut adalah public, private dan protected. Private berarti method atau property yang ada di dalam suatu kelas hanya bisa diakses di dalam kelasnya. Sedangkan pada method atau property yang bersifat public berarti method atau property tersebut bisa diakses di dalam dan di luar kelas. Inheritance Inheritance atau dalam bahasa Indonesianya disebut sebagai pewarisan adalah suatu cara untuk membuat sebuah kelas yang baru dengan menggunakan kelas lain yang sebelumnya sudah dibuat. Pada hubungan inheritance, sebuah class turunan mewarisi kelas leluhur (parent class). Oleh karena mewarisi, maka semua atribut dan method class dari induk akan dibawa (kecuali yang bersifat private), secara intrinsik menjadi bagian dari class anak. Adapun keuntungan yang didapat dari inheritance menambah fitur baru pada kelas anak dan mengubah atau mengganti fitur yang diwarisi dari kelas parent Adapun contoh kelas yang menggunakan konsep inheritance adalah

```

<?php
/* inheritance.php */ class Bapak {
private $nama ="Bapak"; function Bapak($n) {
$this->nama = $n;
}
function Hallo() {
echo "Halo, saya $this->nama <br>";
}
}

class Anak extends Bapak { function Hai(){
Echo \hai dari kelas anak";
}
}

$test = new Anak("Anak dari Bapak");
$test->Hallo();

```

Jika kita perhatikan di kelas anak sama sekali tidak memiliki fungsi hello, tetapi karena parentnya memiliki fungsi tersebut maka si anak dapat menggunakan fungsi tersebut. Selain menggunakan fungsi bapak, anak juga dapat menambah fungsi baru yaitu fungsi Hai.

Chapter 3 Instalasi dan Konfigurasi CodeIgniter Agar dapat menggunakan CodeIgniter, Anda harus menginstall dan melakukan konfigurasi terhadap CodeIgniter

terlebih dahulu. Instalasi CodeIgniter sangatlah mudah. Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Hal-hal yang harus dipersiapkan dalam menginstall CodeIgniter adalah Mempersiapkan Web Server

Seperti yang kita tahu bahwa PHP adalah bahasa pemrograman website yang berjalan disisi server oleh karena itu untuk dapat menjalankan website yang dibuat dengan menggunakan PHP, di komputer harus terinstall aplikasi web server yang mendukung PHP. Banyak sekali aplikasi web server yang beredar, salah satu web server yang sangat terkenal dan juga bersifat bebas adalah web server Apache, sebuah web server yang digunakan pada sebagian server yang ada di internet. Untuk website yang melibatkan basis data sebagai tempat penyimpanan datanya maka tidak perlu bingung karena CodeIgniter juga mendukung berbagai jenis server basis data, yaitu MySQL, MySQLi, MS SQL, PostgreSQL, Oracle, dan SQLite. CodeIgniter juga bisa dijalankan di semua sistem operasi yang bisa menjalankan aplikasi-aplikasi di atas, baik Windows, Linux, BSD dan yang lainnya. Instalasi CodeIgniter

Instalasi CodeIgniter sangat mudah. Meskipun namanya instalasi tetapi karena CodeIgniter adalah aplikasi berbasis website maka sebenarnya yang perlu dilakukan adalah meng-copy folder aplikasi CodeIgniter ke dalam folder htdocs atau DocumentRoot dari web server yang telah diinstall sebelumnya. Berbeda dengan instalasi software pada umumnya. Sebelum melakukan instalasi yang perlu dilakukan pertama kali adalah mendapatkan kode sumber dari CodeIgniter itu sendiri yang dapat didownload di <http://www.CodeIgniter.com/> (disediakan pula di dalam CD penyerta buku). Selanjutnya letakkan folder hasil ekstrak tadi di DocumentRoot web server, yaitu folder htdocs didalam direktori C:\apache\friendsbagi yang menggunakan XAMPP di Windows atau /var/www/html bagi yang menggunakan linux (semua tergantung dimana anda menginstall dan mengkonfigurasi webserver).

Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Adapun struktur utama dari CodeIgniter terbagi menjadi dua bagian, yaitu application dan sistem/core CodeIgniter. Application adalah tempat kita meletakkan code yang akan dibuat (bewarna merah dan hijau sedangkan sistem/core CodeIgniter yang bewarna ungu) . Folder sistem berisi library-library dan helper bawaan CodeIgniter.

Gambar 4. Struktur Direktori CodeIgniter Adapun susunan folder CodeIgniter secara default adalah: • Folder application: disinilah aplikasi yang akan kita bangun diletakkan. Folder config - tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari database, router dan autoload aplikasi. Folder controllers - tempat menyimpan semua file controller. Folder errors - tempat menyimpan semua template error aplikasi. Folder helpers - tempat menyimpan helper-helper yang bukan berasal dari CI. Folder hooks - tempat menyimpan hook yang digunakan untuk mengubah alur fungsi dari core CI. Folder language - tempat menyimpan bahasa-bahasa yang akan di gunakan. Folder libraries - tempat menyimpan semua library buatan kita sendiri. Folder models - tempat menyimpan semua model. Folder views - tempat menyimpan semua file view aplikasi.

- Folder system menyimpan semua file baik itu file aplikasi yang dibuat maupun core framework-nya. Folder cache - tempat menyimpan semua cache yang dibuat caching library. Folder codeigniter - tempat menyimpan semua semua file internals CI. Folder database - tempat menyimpan semua driver database drivers dan class yang akan digunakan. Folder fonts - tempat menyimpan semua font yang digunakan image manipulation library. Folder helpers - tempat menyimpan semua helper core CI. Folder language - tempat menyimpan semua language core CI. Folder libraries - tempat menyimpan semua library core CI. Folder logs - tempat menyimpan semua logs generated oleh CI. Folder plugin - tempat menyimpan semua plugin core CI. Folder scaffolding - tempat menyimpan semua file yang berfungsi sebagai scaffolding.
- Folder *user_guide* berisi *user_guide/manual* penggunaan CI. *File index.php* file yang akan *handle* semua request yang masuk ke aplikasi.

Setelah meletakkan CodeIgniter ke dalam folder htdocs maka akan didapatkan tampilan seperti di bawah ini, artinya CodeIgniter telah sukses berjalan di aplikasi Anda.

Gambar 5. Tampilan browser ketika sukses menginstall Codeigniter

Adapun checklist yang harus dilakukan untuk menjalankan CodeIgniter secara default adalah

- Pastikan Apache dan PHP telah terinstall dan berjalan di komputer.
- Pastikan peletakkan source code CodeIgniter di folder/direktori web apache (biasanya htdocs) dan memiliki permission setidaknya-tidaknya read only atau kode 644.

Konfigurasi CodeIgniter

Walaupun CodeIgniter dapat berjalan dengan konfigurasi default, tetapi untuk sebuah aplikasi yang nyata kita harus tetap melakukan konfigurasi, setidaknya pada bagian *base_url* dan *router*. Pengaturan *base_url* dan *router* sangat berguna ketika proses pengembangan file yang terdapat dalam direktori tersebut dan sering digunakan antar lain : *Config.php*. Pada file konfigurasi lain. Adapun konfigurasi – konfigurasi yang perlu diperhatikan adalah : *o config['base_url'] – Konfigurasi ini berisi alamat url sebuah aplikasi. Jika menggunakan helper url maka konfigurasi ini harus diisi dengan benar. Contoh : aplikasi Anda akan diakses dengan menggunakan domain www.contoh.com/app_cim*

http : //www.contoh.com/app_ci/";

Tetapi jika ingin menggunakan base url yang lebih fleksibel maka dapat menggantinya dengan variabel server. Contoh: *config['base_url'] = "http : //".SERVER['HTTP_HOST'].str_replace(basename(SERVER['SCRIPT_NAME']),"",SERVER['SCRIPT_NAME'])."*

o config['index.php'] – Konfigurasi ini berisi file yang menjadi front controller. Konfigurasi ini berhubungan dengan konfigurasi ini bertujuan untuk menentukan bagaimana library URI bekerja. CodeIgniter dapat menangkap URL masing-masing cara mempunyai kelebihan dan kekurangan masing-masing, bahkan tidak semua web server mendukung konfigurasi ini bertujuan untuk menambahkan akhiran pada url. Contoh Anda mempunyai sebuah controller page //localhost/index.php/page, dengan menambahkan url_sufix berisi \html" maka url tadi dapat juga diakses melalui //localhost/index.php/page.html.

o config['language'] – Secara default CodeIgniter sudah mendukung banyak bahasa / multy language. Kita dapat pasang yang ada dalam CodeIgniter dengan bahasa yang kita kehendaki. Untuk mengubah bahasa tersebut cukup pada konfigurasi ini bertujuan mengaktifkan / menonaktifkan hook pada CodeIgniter. Hook dapat dikatakan event – event yang terjadi pada CodeIgniter, dimana kita bisa meletakkan fungsi di dalamnya. Hook akan bermanfaat ketika Anda ingin mengubah urutan loading library dimana Anda sudah meng – extend library router untuk menggunakan nya. Artinya Anda harus me load library database sebelum library loader. Hal tersebut bisa dilakukan melalui hook.

MY. Jikalau kita ingin mengubah/mengextend library CodeIgniter maka library tersebut harus memiliki prefik yang sama dengan konfigurasi pesan kesalahan yang termasuk kesalahan PHP. Jika di set 2 maka akan menampilkan semua pesan debug dan pesan logging yang berisi informasi seperti sebuah library telah di load juga akan ditampilkan. Jika di set 4 maka semuanya lain melalui konfigurasi ini. `config['sess_cookie_name'] = 'Namacookie'; config['sess_expiration'] = 7200; config['sess_encrypt_cookie'] = FALSE; config['sess_use_database'] = FALSE; config['sess_table_name'] = 'session_table'; config['sess_match_ip'] = FALSE; config['sess_match_useragent'] = TRUE; config['sess_time_to_update'] = 300;`

- Autoload.php. Konfigurasi ini bertujuan untuk menentukan sumber daya apa yang akan di load secara otomatis. Cara penggunaannya sederhana, misalnya kita ingin meload library database, pagination dan lain-lain secara otomatis maka kita tinggal mengubah `autoload['libraries']` menjadi `:autoload['libraries']=array('database','session','pagination')`
- Routes.php. Konfigurasi di file ini bertujuan untuk menentukan kemana routing oleh library route akan dilakukan. Hal paling sederhana yang harus dilakukan adalah mengubah default controller (controller yang akan dibuka ketika tidak ada uri yang diberikan oleh browser). Misalnya website kita beralamat `www.koder.web.id`. Maka ketika membuka website tersebut maka secara otomatis CodeIgniter akan mengalihkan ke controller default, karena tidak disertakan di dalam url kita. Adapun yang perlu diubah adalah `route['default_controller'] = "welcome"`;

Kesepakatan Coding (Coding Standart) CodeIgniter

Sebelum melakukan coding menggunakan codeigniter maka ada baiknya kita mengetahui apa saja kesepakatan-kesepakatan yang ada di codeigniter. Kesepakatan-kesepakatan tersebut akan membuat kode kita lebih mudah dipahami oleh developer lainnya. Adapun kesepakatan tersebut diantaranya : • PHP Closing Tag Ketika kita menulis library, helper, controller ataupun model maka sebaiknya tidak menggunakan tanda penutup pada dokumen php `?>`. Hal tersebut dilakukan untuk mencegah adanya spasi atau karakter yang tidak diinginkan pada code kita sehingga membuat aplikasi error. Kita juga disarankan untuk memberikan informasi tentang akhir dokumen dan berisi path dokumen tersebut.

Contoh salah: `if (php echo "Here's my code!"); ?>`

Contoh Benar: `if (php echo "Here's my code!");`

`/* End of file myfile.php */ /* Location: ./system/modules/mymodule/myfile.php */`

- Penamaan Class dan Method Penamaan Class harus dimulai dengan huruf besar. Jika class menggunakan beberapa kata maka kata-kata tersebut dipisahkan menggunakan underscore dan bukan camelcase.

Contoh salah: `class superclass class SuperClass` Contoh Benar: `class Superclass`

Aturan diatas juga berlaku untuk method contohnya

Contoh kurang tepat: `function fileproperties()` // Tidak deskriptif dan memiliki underscore `function fileProperties()` // Tidak deskriptif dan underscore CamelCase `function getFileproperties()` // Kurang underscore `function getFileProperties()` // menggunakan CamelCase `get_the_file_properties_from_the_file()` // terlalu panjang Contoh Tepat : `function getFileProperties()` // deskriptif, pakai underscore, dan huruf kecil

Chapter 3 Hello CodeIgniter Pada bab ini kita akan memulai coding menggunakan CodeIgniter. Sebelum memulai memulai maka perlu kita pahami kembali

apa itu Controller? Dan bagaimana hubungan controller dengan URI? Apakah Controller?

CodeIgniter adalah sebuah framework berbasis MVC. Sebuah Controller dapat dikatakan sebagai jantung dari suatu aplikasi, karena controller menentukan bagaimana permintaan HTTP yang harus ditangani. Sebuah kelas Controller adalah sebuah file yang terletak di dalam folder `application/controller` dan memiliki nama file yang sama dengan nama kelasnya dan dikaitkan dengan URL. Segmen-segmen pada URL pada codeigniter mencerminkan Controller yang dipanggil. Contoh: `http://www.domainsaya.com/class/fungsi/id` maka domain tersebut dapat dipecah menjadi bagian-bagian diantaranya:

Adapun komponen-komponen URL diatas adalah

- Konfigurasi Base Url, Bagian ini merupakan url yang kita masukkan pada konfigurasi base url yang merupakan url paling dasar untuk mengakses web atau aplikasi kita. *Segmen URI pertama*

Untuk contoh pertama, kita akan membuat sebuah aplikasi hello codeigniter yang sederhana. Aplikasi tersebut akan diletakkan di folder hello dari htdocs anda, sehingga anda dapat mengaksesnya dengan membuka url `http://localhost/hello`. Perhatikan URL Berikut ini: `http://localhost/hello/index.php/blog/` Pada contoh di atas, CodeIgniter akan berusaha untuk menemukan sebuah controller bernama `blog` pada file `blog.php`, kemudian menampilkannya ke browser. Mari kita coba membuat sebuah controller sederhana sehingga dapat melihat apa yang terjadi. Dengan menggunakan teks editor, buatlah file bernama `blog.php` di dalam folder `application/controller`. Setelah itu buatlah sebuah kelas `Blog` yang merupakan turunan dari kelas `CI_Controller`.

```
1. <?php if ( ! defined('BASEPATH')) 2. exit('No direct script access allowed'); 3. 4. class Blog extends CI_Controller { 5. 6. function __construct() { 7. parent::construct(); 8. 9. 10. 11. function index() { 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. }
```

Setelah itu apa yang terjadi? Coba buka situs Anda dengan menggunakan URL seperti ini:

`http://localhost/index.php/blog/`

Gambar 6. Tampilan Aplikasi Hello Codeigniter

Jika Anda melakukannya dengan benar, maka akan tampak tulisan Hello.. saya adalah contoh CodeIgniter pertama. Sekedar mengingatkan kembali, ketentuan penamaan class (baik controller maupun library) harus dimulai dengan huruf besar. Contoh:

```
<?php class Blog extends CI_Controller
```

Berikut ini contoh yang salah :

```
<?php class blog extends CI_Controller
```

Pada contoh di atas nama fungsi yang dipanggil adalah `index()`. Fungsi "Index" akan selalu dibaca secara default jika segmen kedua dari URI kosong. Cara lain untuk menampilkan "Hello CodeIgniter" adalah dengan mengakses url berikut: `localhost/index.php/blog/index/`

Segmen kedua dari URI yang menentukan fungsi mana yang akan dipanggil dari controller. Mari kita coba menambahkan fungsi baru ke controller:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed'); class Blog extends CI_Controller { function __construct() { parent::construct();
```

```
function index() { echo "Haloo.. saya adalah contoh codeigniter pertama";
```

```

function komentar() echo "Ini adalah fungsi komentar"; /* End of file
Blog.php */ /* Location: ./application/controllers/blog.php */
Sekarang cobalah panggil URL berikut untuk melihat fungsi komentar:
localhost/index.php/blog/komentar/
Maka hasilnya akan menampilkan pesan "Ini Fungsi Komentar". Jika ingin
mengisi variabel param maka Anda tinggal menambahkan sebuah URI lagi setelah Blog. Biasanya parameter ketik
localhost/index.php/blog/komentar/tutorial-ci/123
Fungsi post pada controller dapat memiliki dua parameter yang akan dile-
watkan pada URI segmen 3 dan 4 ( "tutorial-ci" dan "123"). Jadi dapat
disimpulkan bahwa URI yang dipanggil dapat ditunjukkan seperti ini: local-
host/index.php/[Controller]/[Fungsi]/[param]/[param]/...

```

Controller dan View

Pada contoh program Hello CodeIgniter diatas adalah cara untuk menampilkan tulisan "hello CodeIgniter" secara langsung di controller. Namun sebenarnya hal tersebut bisa dilakukan di view. Sebagai contoh: tuliskan kode program di bawah ini pada file application/controller/blog.php :

```

i?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller function construct() parent :: construct();
function index() this->load->view("hello_codeigniter");
/* End of file Blog.php */ /* Location: ./application/controllers/ Blog.php
*/

```

Selanjutnya buatlah file *hello_codeIgniter.php* di folder *application > view(application/view/hello_codeIgniter.php)* :

Maka kode diatas akan memberikan hasil yang sama dengan contoh kasus pertama (tanpa menggunakan view), yang berbeda hanya tulisannya saja. Sebuah View sebenarnya hanyalah sebuah halaman web atau bagian dari halaman web, seperti sebuah header, footer, sidebar, dan lain-lain. Bahkan, View bisa menjadi fleksibel karena view dapat dimasukkan ke dalam view yang lain jika dibutuhkan. Untuk memanggil file view dapat digunakan fungsi seperti berikut ini (perhatikan baris 12):: *this->load->view('nama_view')*;

Nama_view adalah nama file view Anda. Dan file tersebut harus diletakkan di dalam folder *application/view*.

1. Nama file view - Nama file yang hendak di-load yang terletak di dalam folder *application/view*

2. Data Parameter - Parameter ini digunakan untuk melewati data dari controller ke dalam view.

Contoh: buatlah file controller dengan nama *blog.php* dengan kode program seperti berikut ini:

```

i?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller function construct() parent :: construct(); function index() data['judul']="J
blog"; data['isi'] = "Isi blog"; this->load->view("blog_view",data);
Blog.php */

```

Selanjutnya buatlah file view bernama *blog_view.php* dengan kode program seperti berikut ini :

```

i!i?php echo judul;? >< /h1 >< p ><?php echo isi; ?i/pi i?i?br /iPage
rendered in elapsed_time seconds < /p >

```

Jika halaman *blog.php* dipanggil maka tampilannya kurang lebih seperti berikut:

CodeIgniter yang otomatis di load. Loader berfungsi sebagai pengatur dari sumberdaya-sumberdaya yang ada di dalam CodeIgniter seperti Model, View, Library, Helper, dan plugin. Cara penggunaannya adalah: *this->load->library('nama_library');*; *this->load->helper('nama_helper');*; *this->load->plugin('nama_plugin');*;

Nama library, helper dan plugin harus di isi dengan huruf kecil.

Ketika sebuah library sudah di-load maka library tersebut menjadi property pada object Controller. Adapun cara penggunaannya adalah sebagai berikut: *this->nama_library->fungsi();*

Library CodeIgniter Secara default CodeIgniter telah menyediakan library yang dapat digunakan secara langsung. Adapun library yang telah tersedia antara lain: • Benchmarking Class Library ini digunakan untuk melakukan pengukuran terhadap aplikasi yang dibuat. Seperti untuk mengetahui berapa lama waktu eksekusi dan berapa jumlah memori yang digunakan. Library ini sudah digunakan dan di-load secara otomatis oleh CodeIgniter. • Calendar Class Library ini berfungsi untuk menampilkan dan men-generate kalender. • Cart Class Library ini berfungsi untuk membuat shopping cart (keranjang belanja). Library ini memiliki ketergantungan terhadap kelas session karena item-item chart tersebut disimpan di dalam session. • Config Class Library ini berfungsi untuk mengambil data-data di dalam file konfigurasi. Library ini sudah di load secara otomatis oleh CodeIgniter. • Database Class Library database digunakan untuk memanipulasi serta mendapatkan data dari sebuah sistem database. Secara default database yang sudah didukung oleh CodeIgniter adalah mysql, mssql, oracle, postgres. Sedangkan database yang tidak didukung secara langsung oleh CodeIgniter dapat dijembatani dengan driver odbc. • Email Class Library email digunakan untuk mengirimkan email. Pengiriman email tersebut bisa dilakukan dengan menggunakan protokol mail, sendmail dan smtp.sqawd • Encryption Class Library Encryption digunakan untuk melakukan penyandian terhadap string tertentu • File Uploading Class Library Uploading digunakan untuk meng-upload file. Kelas ini sudah dilengkapi dengan pengecekan jenis file, dan ukuran file. • Form Validation Class Library form Validation digunakan untuk mengecek keabsahan form-form yang sudah di-submit oleh user.

• FTP Class Library FTP digunakan untuk meng-upload atau download file melalui ftp server. • HTML Table Class Library HTML table adalah sebuah kelas yang berfungsi untuk men-generate table dari data array. • Image Manipulation Class Library image manipulation berfungsi untuk mengolah gambar. Adapun fungsi-fungsi yang telah disediakan adalah Image Resizing, Thumbnail Creation, Image Cropping, Image Rotating dan Image Watermarking. • Input and Security Class Library Input dan security berfungsi untuk menjamin bahwa inputan dari form telah bersih dari karakter-karakter “aneh”. • Loader Class Library ini dapat disebut sebagai pengatur sumberdaya CodeIgniter. Semua sumberdaya yang ada akan dikendalikan oleh kelas ini. Library ini sudah di-load secara otomatis oleh CodeIgniter. • Language Class Library language digunakan untuk mengatur bahasa apa yang akan dipakai oleh CodeIgniter. • Output Class Library Output bertujuan untuk meng-handle output dari CodeIgniter, mulai dari cache sampai ke profiling bisa dilakukan kelas ini. • Pagination Class Untuk mem-paginate hasil database untuk performance dan usability, kita bisa

mengontrol berapa banyak record untuk ditampilkan disetiap halaman website, berapa banyak record untuk ditarik dari database dan tampilan dari bagian pagination • Session Class Library Session dapat digunakan untuk memelihara informasi status tentang user (seperti layaknya session di PHP). Tetapi Library ini tidak menggunakan session built-in dari PHP, Library Session men-generate session datanya sendiri yang disimpan di dalam Cookies. • Trackback Class Library Trackback digunakan untuk mengirim dan menerima data trackback. • Template Parser Class Library Template Parser digunakan untuk membuat template yang berisi parsable pseudo – templates. • Unit Testing Class Library Unit Testing digunakan untuk unit test function dalam aplikasi yang sedang dibuat. CodeIgniter menyediakan fungsi evaluasi dan dua fungsi hasil dalam library ini.

- URI Class Library URI digunakan untuk memarsing URL, lalu memecahnya ke dalam beberapa segmen dan kemudian di-passing ke controller atau disimpan sebagai variabel.
- User Agent Class Library User Agent digunakan untuk mengidentifikasi browser, mobile device, atau robot yang mengunjungi website. Kita juga bisa menggunakannya untuk mendeteksi dukungan bahasa, sekumpulan karakter, dan referrer.
- XML-RPC Class Library XML-RPC digunakan untuk men-setup klien XML-RPC dan server.
- Zip Encoding Class Library Zip Encoding digunakan untuk membuat file ZIP baik yang berjenis teks maupun data binary.

Helper CodeIgniter Helpers seperti namanya akan membantu Anda membangun aplikasi dengan tugas tertentu. Tidak seperti library, helper bukanlah Object Oriented tapi berupa prosedural. Setiap helper berisi satu atau lebih fungsi, masing-masing berfokus pada tugas tertentu yang tidak ada ketergantungan dengan fungsi lainnya. Helper dapat juga di-load secara otomatis dalam /system/application/config/autoload.php. Adapun helper yang ada di dalam CodeIgniter antara lain:

- Array — Helper array berisi fungsi yang membantu pekerjaan berhubungan dengan array. Sebagai contoh fungsi `random_element()` mengambil array sebagai input dan mengembalikan satu elemen acak dari array tersebut.
- Download — Helper download berisi fungsi untuk mengunduh file dari internet. Fungsi `force_download()` menghasilkan header server yang memaksa data untuk di-download.
- File — Helper file berisi fungsi yang membantu untuk membaca, menulis, dan menghapus file.
- Form — Helper form berisi fungsi yang membantu membangun form.
- HTML — Helper HTML berisi fungsi yang membantu membuat blok HTML.
- Language — Helper language berisi fungsi yang membantu untuk mengubah kata-kata menjadi bentuk plural atau singular, memberlakukan camel case atau mengubah kata-kata yang dipisahkan oleh spasi menjadi frase yang digarisbawahi, sebagai contoh fungsi `singular` bisa mengubah kata-kata yang membantu pengelolaan emoticons.
- String — Helper string berisi fungsi-fungsi yang membantu pekerjaan berhubungan dengan string, seperti fungsi `random_string()` yang akan membuat string acak.
- Text — Helper text berisi fungsi-fungsi yang membantu Anda bekerja dengan teks. Sebagai contoh: fungsi `word_limiter()` dapat membatasi string ke sejumlah kata tertentu yang berguna untuk memformat output.
- Typography — Helper typography berisi fungsi-fungsi yang membantu memformat output yang berkaitan dengan HTML.
- URL — Helper URL berisi fungsi-fungsi yang membantu pekerjaan dengan URL. Anda akan menggunakan fungsi `base_url()` dan `anchor()` dalam setiap halaman.

Membuat Library Sendiri Untuk mendapatkan kode yang bagus sebaiknya fungsionalitas yang sama itu di bungkus menjadi sebuah library. Sebuah library adalah sebuah kelas yang diletakkan pada folder application/libraries. Sebuah library yang bernama `MyClass` akan memiliki file `MyClass.php` yang akan ditempatkan di folder `application/libraries`. Untuk menggunakan library ini, kita perlu memanggilnya di controller atau model. Contoh penggunaan library ini:

```

// Di Controller
$this->load->library('MyClass');
// Di Model
$this->load->library('MyClass');

```

CodeIgniter sangat mudah untuk mengadopsi library yang bersifat external. Library yang paling mudah untuk diadopsi adalah sebuah single class tanpa parameter pada constructor. Untuk kelas yang memiliki konstruktor lebih dari satu maka sebaiknya kelas tersebut diubah terlebih dahulu atau jika kelas library tersebut terdiri atas beberapa class maka lebih mudahnya kita membuat sebuah library factory dari library tersebut. Contoh 1. Library CSV Reader

Sebagai contoh kita akan menggunakan sebuah library yang berfungsi untuk membaca file CSV kedalam memori.. Adapun isi dari librari tersebut adalah

```

i?php if(!defined('BASEPATH'))exit('No direct script access allowed'); /**
 * CSVReader Class * @author Pierre-Jean Turpeau * @link http://www.CodeIgniter.com/wiki/CSVReader
 */
class csv_eader
{
    var $fields; var $separator=';'; var $enclosure='\"'; var $max_row_size = 4096;
    /** * Parse a file containing CSV formatted data. * * @access public *
    @param string * @param boolean * @return array */
    function parse_file($filepath, $namedFields = true) {
        $content=false; $file =
        fopen($filepath, 'r'); if ($namedFields) {
            $this->fields = fgetcsv($file, $this->max_row_size, $this->
            $separator, $this->enclosure);
            while (($row = fgetcsv($file, $this->max_row_size, $this->
            $separator, $this->enclosure)) != false) {
                if (row[0] != null) // skip empty lines
                if (!$content) $content=array();
                if ($namedFields) $items=array();
                foreach ($this->fields as $id=>$field) {
                    if (isset($row[$id])) $items[$field] = $row[$id];
                }
            }
            return $content;
        }
    }
}

```

Agar dapat menggunakan library tersebut maka lakukan langkah-langkah berikut ini:

- Copy Library Ke Direktori application/libraries Untuk menggunakan library buatan kita sendiri atau library eksternal maka langkah yang pertama yang harus dilakukan adalah meletakkan library itu di direktori application/libraries.

Nama file dan nama kelas harus sama. Contoh nama file adalah `csv_eader.php` maka nama kelasnya adalah `csv_eader`.

Perhatikan controller berikut ini

```

1. i?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Welcome extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    function index() {
        $this->load->library('csv_eader');
        $data = $this->csv_eader->parse_file(FPATH.'data_nilai.csv');
        echo $pre > ";
        print_r($data);
    }
}
/* End of file welcome.php */
/* Location: ./application/controllers/welcome.php */

```

Pada baris ke-13 kita akan mencoba menggunakan library `csv_eader`, kelas tersebut akan memparsing sebuah file CSV. Lalu hasil parsisi

Chapter 6 Kasus 1. Penanganan dan Validasi Form Pada bab ini akan dibahas mengenai penanganan form dan cara melakukan validasi terhadap sebuah form inputan. Penanganan form artinya bagaimana cara kita untuk mendapatkan dan mengolah data yang dikirimkan oleh pengguna melalui form. Untuk membuat form sendiri, sebaiknya kita menggunakan helper form. Setelah kita berhasil mendapatkan data dari user, kita harus memvalidasi data-data yang telah dikirimkan tersebut agar aplikasi yang kita bangun menjadi aman. Inputan yang tidak divalidasi membuat aplikasi kita rentan terhadap serangan seperti Sql Injection, Xss dan lain-lain. Penanganan Form

Untuk menangani form kita membutuhkan library input. Library tersebut sudah dipanggil secara otomatis oleh codeigniter. Fungsi yang tersedia pada library ini selain untuk menangani form juga memiliki fungsi security/keamanan.

Contohnya untuk menghandle atau memfilter xss kita dapat menggunakan fungsi `xssfilter.nama = this->input->xssfilter($this->input->post('nama'));`

Jika kita ingin melindungi seluruh aplikasi dari dengan xss filter maka kita bisa menconfignya dengan mengubah konfigurasi di `system/application/config/config.php` dari `$config['global_xss_filtering'] = FALSE;`

Menjadi

`$config['global_xss_filtering'] = TRUE;`

Selain kedua cara diatas, kita dapat juga menggunakan parameter kedua dari fungsi `post` atau `get` maka kita akan mendapatkan hasil yang sama dengan kedua cara diatas, contohnya `$this->input->post('nama', TRUE);` `$this->input->get('nama', TRUE);` Fungsi `post` digunakan untuk menangkap inputan POST sedangkan `get` digunakan untuk menangkap inputan GET. Selain itu fungsi `input` juga dapat digunakan untuk cookie, variabel SERVER, dan user agent `$this->input->cookie('nama_cookie', TRUE);` `$this->input->ip_address();` `$this->input->server('PATH_INFO', TRUE);`

Selain library `input`, Codeigniter juga menyediakan sebuah helper untuk mempermudah penanganan form. Helper tersebut adalah helper `form`. Helper tersebut membantu kita dalam membuat form pada view. Adapun cara meload helper `form` adalah `$this->load->helper('form');`

Ada banyak fungsi yang disediakan oleh helper ini diantaranya `form_open()`, `form_close()`, `form_open_multipart()` *lain.* Untuk lebih detailnya anda dapat melihat pada dokumentasi codeigniter. Pada kasus ini, fungsi yang banyak digunakan adalah `form_open()` dan `form_close()`.

Contoh : `$attributes = array('class' => 'email', 'id' => 'myform'); echo form_open('form/send', $attributes);`

Maka akan menghasilkan

`<form method="post" accept-charset="utf-8" action="http://localhost/index.php/email/send" class="email" id="myform">`

`<input type="text" name="username" id="username" value="ibnoe">` *Carapadahtml.* Fungsi untuk menghasilkan tag input pada html. Fungsi ini dapat dipakai dengan dua cara. Cara pertama adalah dengan menggunakan `form_open()` dan `form_close()`.

Contoh : `$data = array('name' => 'username', 'id' => 'username', 'value' => 'ibnoe', 'maxlength' => '100', 'size' => '50', 'style' => 'width: 50px');`

`echo form_input($data);`

Akan menghasilkan output :

`<input type="text" name="username" id="username" value="ibnoe" maxlength="100" size="50" style="width: 50px;" />`

Untuk contoh menangani form maka ikutilah langkah-langkah berikut ini.

1. Membuat Controller Hitung

Perlu diingat hal pertama yang harus di konfigurasi pada setiap aplikasi adalah `BASE_URL`. Setelah melakukan konfigurasi `base_url` pada `file application/config/config.php` maka kita akan dapat menggunakan `load->helper(array('url', 'form'));` 9. 10. 11. `function index()` 12. 13. `$this->load->view('menu_hitung');` 14. 15. 16. `function perkalian()` 17. 18. `$data['v1'] = (int) $this->`

```

input->post('v1',true);19.data['v2']=(int)$this->input->post('v2',true);20.data['hasil']=data['v1']*data
21. $this->load->view('perkalian',data);22.23.24. function pembagian()25.26. data['v1']=(int)$this->input->post('v1',true);27. data['v2']=(int)$this->input->post('v2',true);28. if (data['v2'] > 0)29.data['hasil']=data['v1']/data['v2'];
30. else31. data['hasil']='Error,v2tidakboleh0!';32.$this->load->view('pembagian',data);33.34.

```

Pada baris ke 8, dilakukan loading terhadap helper url dan form sekaligus. Loading tersebut dilakukan dalam fungsi konstruktor karena kedua helper tersebut digunakan pada semua fungsi dan view yang ada. Pada kelas hitung tersebut, terdapat 3 fungsi yaitu index, perkalian dan pembagian. Pada fungsi index hanya berisi sebuah pemanggilan terhadap view. View

tersebut akan berisi menu-menu yang akan mengarahkan pengguna untuk melakukan perkalian atau pembagian. Pada fungsi perkalian dan pembagian akan dilakukan proses perkalian atau pembagian. Pada fungsi tersebut, variabel yang berasal dari view akan ditangkap menggunakan fungsi post (berarti dikirimkan melalui POST method). 2. Membuat View

Setelah membuat controller maka kita akan membuat view yang merupakan interface dari aplikasi tersebut. Code berikut ini adalah view menu *hitung*. *View* ini berisi navigasi ke fungsi perkalian dan pembagian.

```

!DOCTYPE html > 2. <html lang="en"> 3. <head> 4. <meta charset="utf-8"> 5. <title> Welcome to CodeIgniter </title> 6. 7. <style type="text/css"> 8. 9. body { 10. background-color: #fff; 11. margin: 40px; 12. font-family: Lucida Grande, Verdana, Arial, Helvetica, sans-serif; 13. } 14. </style> 15. </head> 16. <body> 17. 18. <h1> CodeIgniter 2.0 dan Form! </h1> 19.

```

```

20. <p>Silakan pilih menu dibawah ini.</p> 21. 22. <ul> 23. <li><a href="hitung/perkalian">Perkalian</a></li> 24. <li><a href="hitung/pembagian">Pembagian</a></li> 25. </ul> 26. <p>Page rendered in elapsed time: 0.0001 seconds </p> 27. </body> 28. </html>

```

Perhatikan baris ke-41 dan ke-42, itu adalah contoh penggunaan helper. Fungsi anchor bertujuan untuk membuat sebuah link ke page tertentu. Berikut ini adalah tampilan dari view diatas jika dipanggil.

Gambar 7. Screenshoot view menu *hitung* Setelah menu *hitung* selanjutnya membuat view untuk perkalian dan pembagian.

```

!DOCTYPE html > 2. <html lang="en"> 3. <head> 4. <meta charset="utf-8"> 5. <title> Welcome to CodeIgniter </title> 6. 7. <style type="text/css"> 8. 9. body { 10. background-color: #fff; 11. margin: 40px; 12. font-family: Lucida Grande, Verdana, Arial, Helvetica, sans-serif; 13. } 14. </style> 15. </head> 16. <body> 17. 18. <h1> Perkalian! </h1> 19. 20. <p>Silakan masukan data berikut!!</p> 21. 22. <form action="hitung/perkalian"> 23. <input type="text" value="v1"/> 24. x 25. <input type="text" value="v2"/> 26. <br> 27. <input type="submit" value="Hitung!!"/> 28. </form> 29. 30. <p>Hasil: <input type="text" value="hasil"/> </p> 31. <p>Page rendered in elapsed time: 0.0001 seconds </p> 32. </body> 33. </html>

```

Pada view ini banyak digunakan helper form (baris 40 – 45). Tag input yang ada pada view ini di-generate oleh fungsi `form_input()` dan `form_submit()`. Adapun tampilan

```

Application/views/pembagian.php 1. <!DOCTYPE html> 2. <html lang="en"> 3. <head> 4. <meta charset="utf-8"> 5. <title> Welcome to CodeIgniter</title>

```

```

6. 7. <style type="text/css"> 8. 9. body 10. background-color: #fff; 11. margin: 40px; 12. font-family: Lucida Grande, Verdana, Sans-serif; 13. font-size: 14px; 14. color: #4F5155; 15. 16. 17. a 18. color: #003399; 19. background-color: transparent; 20. font-weight: normal; 21. 22. 23. h1 24. color: #444; 25. background-color: transparent; 26. border-bottom: 1px solid #D0D0D0; 27. font-size: 16px; 28. font-weight: bold; 29. margin: 24px 0 2px 0; 30. padding: 5px 0 6px 0; 31. 32. 33. </style> 34. </head> 35. <body> 36. 37. <h1>pembagian!</h1> 38. 39. <p>Silakan masukan data berikut!!</p> 40. <?php echo form_open('hitung/pembagian');? > 41. <?php echo form_input('v1',v1);?> 42. <?php echo form_input('v2',v2);?><br> 43. 44. <?php echo form_submit('submit','Hitung!!');? > 45. <?php echo form_close();? ><br> 46. Hasil :<?php echo hasil;?> 47. 48. </br></body></html>

```

Berikut ini adalah tampilan dari form pembagian

Gambar 9. Screenshot view pembagian

Catatan :

Menggunakan Validasi Form

Sebenarnya sampai pada tahap kedua di atas kita telah mampu membuat sebuah aplikasi perkalian dan pembagian sederhana, tetapi sebuah aplikasi yang baik harus memiliki sebuah kontrol terhadap inputan user. Validation form sangat penting pada aplikasi. Sebuah aplikasi akan memiliki banyak inputan dari pengguna dan semua inputan untuk pengguna harus aman. CodeIgniter memiliki sebuah library Form Validation Library yang akan membantu kita untuk membuat sebuah validasi yang cepat, mudah dan aman. Pada contoh validasi form ini hampir sama dengan 2 langkah diatas, tetapi memiliki sedikit penambahan terutama dalam fungsi perkalian dan pembagian. Perhatikan controller hitung berikut ini (controller berikut ini berasal dari contoh sebelumnya dengan penambahan). Application/controllers/hitung.php

```

1. <?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Hitung extends CI_Controller {
4.     function __construct() {
5.         parent::__construct();
6.     }
7.     function index() {
8.         $this->load->helper(array('url','form'));
9.         10. 11.
10.    function index() {
11.        12. 13.
14.    }
15.    }
16.    }
17.    }
18.    }
19.    }
20.    }
21.    }
22.    }
23.    }
24.    }
25.    }
26.    }
27.    }
28.    }
29.    }
30.    }
31.    }
32.    }
33.    }
34.    }
35.    }
36.    }
37.    }
38.    }
39.    }
40.    }
41.    }
42.    }
43.    }
44.    }
45.    }
46.    }
47.    }
48.    }
49.    }
50.    }
51.    }
52.    }
53.    }
54.    }
55.    }
56.    }
57.    }
58.    }
59.    }
60.    }
61.    }
62.    }

```

Perhatikan baris 19-24 pada fungsi perkalian, disana kita me-load sebuah library bernama `form_validation`. Library tersebut memiliki fungsi untuk melakukan validasi terhadap inputan user.

```

1. <?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Hitung extends CI_Controller {
4.     function __construct() {
5.         parent::__construct();
6.     }
7.     function index() {
8.         $this->load->helper(array('url','form'));
9.         10. 11.
10.    function index() {
11.        12. 13.
14.    }
15.    }
16.    }
17.    }
18.    }
19.    }
20.    }
21.    }
22.    }
23.    }
24.    }
25.    }
26.    }
27.    }
28.    }
29.    }
30.    }
31.    }
32.    }
33.    }
34.    }
35.    }
36.    }
37.    }
38.    }
39.    }
40.    }
41.    }
42.    }
43.    }
44.    }
45.    }
46.    }
47.    }
48.    }
49.    }
50.    }
51.    }
52.    }
53.    }
54.    }
55.    }
56.    }
57.    }
58.    }
59.    }
60.    }
61.    }
62.    }

```

Baris di atas merupakan salah satu contoh untuk menentukan aturan inputan pada form. Parameter pertama (berisi v1) merupakan name dari inputan yang akan divalidasi (harus sama dengan atribut name pada tag input contoh: `<input name="v1">`). Parameter kedua adalah nama dari inputan tersebut. Nama tersebut dapat berbeda atau sama dengan name pada parameter pertama. Parameter ketiga merupakan aturan dari inputan tersebut. Aturan-aturan tersebut harus dipatuhi agar sebuah form dapat diproses. Aturan-aturan tersebut dipisahkan oleh tanda `—`. Pada contoh di atas berarti inputan v1 harus diisi (required) dan berisi data integer(integer). Adapun aturan-aturan yang tersedia di dalam CodeIgniter adalah:

Aturan Keterangan required Isi inputan tidak boleh kosong. matches Isi inputan harus sama dengan inputan tertentu contoh

`matches[password] minlength` Panjang inputan harus memiliki jumlah minimal karakter, contoh penggunaan `z.alpha_numeric` Inputan harus berisi karakter alfabet dan numeric. `alpha_dash` Inputan harus berisi semua huruf, `z.alpha_dash` Inputan harus berisi karakter alfabet, dash, dan underscore. `numeric` Inputan hanya boleh berisi angka/huruf numeric. `integer` Inputan hanya boleh berisi 0, 1, 2, 3, dan seterusnya. `is_natural` Inputan hanya boleh berisi bilangan natural saja kecuali nol : 1, 2, 3, dan seterusnya. `valid_email` Inputan harus berisi format email yang benar. `valid_emails` Inputan harus berisi

Pada baris-46, kita melakukan pengecekan terhadap rule-rule yang telah diset (`this->form_validation->run()`). Fungsi tersebut akan menghasilkan nilai TRUE apabila semua

rule terpenuhi dan menghasilkan nilai FALSE jika sebaliknya. Selain fungsi-fungsi diatas Anda juga dapat membuat custom validation jika aturan-aturan di atas tidak ada yang memenuhi dengan kebutuhan Anda. 1. `<!DOCTYPE html>` 2. `<html lang="en">` 3. `<head>` 4. `<meta charset="utf-8">` 5. `<title>` Perkalian menggunakan validasi!! `</title>` 6. 7. `<style type="text/css">` 8. 9. `body` 10. `background-color: fff;` 11. `margin: 40px;` 12. `font-family: Lucida Grande, Verdana, Sans-serif;` 13. `font-size: 14px;` 14. `color: 4F5155;` 15. 16. 17. `a` 18. `color: 003399;` 19. `background-color: transparent;` 20. `font-weight: normal;` 21. 22. 23. `h1` 24. `color: 444;` 25. `background-color: transparent;` 26. `border-bottom: 1px solid D0D0D0;` 27. `font-size: 16px;` 28. `font-weight: bold;` 29. `margin: 24px 0 2px 0;` 30. `padding: 5px 0 6px 0;` 31. 32. 33. `</style>` 34. `</head>` 35. `<body>` 36. 37. `<h1>` Perkalian!! `</h1>` 38. `<?php echo validation_errors();? >` 39. `< p >` Silakan masuk dan data berikut!! `</p >` 40. `<?php echo form_open('hitung/perkalian');? >` 41. `<?php echo form_input('v1',v1);? >` 42. `<?php echo form_input('v2',v2);? >` 43. 44. `<?php echo form_submit('submit','Hitung!!');? >` 45. `<?php echo form_close();? >` 46. `Hasil :<?php echo hasil;? >` 47. 48. `</body>` 49. `</html>`

Pada view, kita cukup menambahkan satu fungsi yang berguna untuk menampilkan kesalahan yang terjadi. Perhatikan baris 38. Fungsi `validation_error` adalah fungsi untuk menampilkan kesalahan.

Gambar 10. Form perkalian dengan tampilan error

Begitu juga untuk view pembagian, kita tinggal menambah fungsi `validation_error` untuk menampilkan error.

1. `<!DOCTYPE html>` 2. `<!DOCTYPE html>` 3. `<html lang="en">` 4. `<head>` 5. `<meta charset="utf-8">` 6. `<title>` Pembagian menggunakan validasi!! `</title>` 7. 8. `<style type="text/css">` 9. 10. `body` 11. `background-color: fff;` 12. `margin: 40px;` 13. `font-family: Lucida Grande, Verdana, Sans-serif;` 14. `font-size: 14px;` 15. `color: 4F5155;` 16. 17. 18. `a` 19. `color: 003399;` 20. `background-`

```

color: transparent; 21. font-weight: normal; 22. 23. 24. h1 25. color: 444;
26. background-color: transparent; 27. border-bottom: 1px solid D0D0D0; 28.
font-size: 16px; 29. font-weight: bold;
30. margin: 24px 0 2px 0; 31. padding: 5px 0 6px 0; 32. 33. 34. 35.
i/stylei 36. i/headi 37. i/bodyi 38. 39. ih1ipembagian!i/h1i 40. i?php echo
validation_errors();? > 41. < p > Silakanmasukandataberikut!! < /p > 42. <
?phpechoform_open('hitung/pembagian');? > 43. <?phpechoform_input('v1',v1);?i
/ 44. i?php echo form_input('v2',v2);?iibr 45. 46. i?php echo form_submit('submit',' Hitung!!');? >
47. <?phpechoform_close();? >< br > 48.Hasil :<?phpechohasil;?i 49. 50.
ipibr/iPage rendered in elapsedimeseconds < /p > 51.52. < /body ><
/html >

```

View diatas akan memberikan tampilan seperti gambar 11 berikut ini.

Gambar 11. Form perkalian dengan tampilan error

Chapter 7 CodeIgniter Database

CodeIgniter mendukung banyak jenis database misalnya MySQL, PostgreSQL, Oracle, dan lain- lain. Dukungan database dari CodeIgniter berupa penyediaan beberapa driver database yang sekaligus juga memiliki fungsi sekuriti, caching dan active record.

Connect ke Database

CodeIgniter memiliki sebuah file konfigurasi yang memungkinkan Anda menyimpan konfigurasi untuk melakukan koneksi ke database (username, password, nama database, dan lain-lain). File konfigurasi terletak berada di application/config/database.php. Pengaturan konfigurasi disimpan dalam bentuk array multi-dimensi dengan prototipe seperti berikut ini: `db['default']['hostname'] = "localhost";db['default']['username']="root"; db['default']['password'] = "";``db['default']['database']="databas``db['default']['dbprefix'] = "";``db['default']['pconnect']=TRUE;``db['default']['debug'] = FALSE;``db['default']['cache_on'] = FALSE;``db['default']['cachedir']="";``db['default']['charset'] = "utf8";db['default']['dbcollat']="utf8_general_ci";`

Kita dapat memiliki beberapa konfigurasi database. Alasan kita menggunakan array multi-dimensi adalah agar memungkinkan Anda secara opsional menyimpan beberapa set nilai-nilai koneksi. Jika, misalnya, Anda menjalankan aplikasi di beberapa kondisi lingkungan/environment (development, production, testing, dll) di bawah satu instalasi, maka Anda dapat mengatur kelompok konfigurasi untuk masing-masing environment. Sebagai contoh, Anda sedang menjalankan aplikasi di environment test maka Anda tinggal menambahkan konfigurasi berikut ini dan mengubah default konfigurasinya (mengeset `active_group = "test"`)`menjaditest :db['test']['hostname']="localhost"; db['test']['username'] = "root";db['test']['password']=""; db['test']['database'] = "database_name";db['test']['dbdriver']="mysql"; db['test']['dbprefix'] = "";``db['test']['pconnect']=TRUE;``db['test']['debug'] = FALSE;``db['test']['cache_on'] = FALSE;``db['test']['cachedir'] = "";``db['test']['charset'] = "utf8";db['test']['dbcollat']="utf8_general_ci";active_group = "test";`

Untuk connect ke database ada beberapa cara yang disediakan oleh CodeIgniter diantaranya :

1. Menambahkan Database Library Sebagai Autoload Library Untuk connect ke database Anda bisa menambahkan database sebagai autoload library di

file application/config/autoload.php. Cara ini sangat sederhana, cukup menambahkan kata "database" ke dalam autoload library sehingga menjadi : `autoload['libraries'] = array("database");`

Ketika menambahkan autoload pastikan bahwa database sudah terkonfigurasi dengan benar untuk menghindari kesalahan. 2. Mengaktifkan Manual Dari Library Database Jika hanya ada beberapa halaman website yang memerlukan konektivitas database, maka untuk optimalisasi lakukan koneksi ke database secara manual, cukup dengan menambahkan baris kode di bawah ini pada tiap fungsi tempat yang membutuhkan koneksi ke database atau dalam konstruktor kelas Anda untuk membuat database yang tersedia secara global di kelas. `this->load->database();`

Jika fungsi di atas tidak berisi informasi apapun di parameter pertama akan menyambung ke group konfigurasi yang aktif. Untuk memilih kelompok tertentu dari file konfigurasi, Anda dapat melakukan seperti pada contoh berikut. Hal tersebut berguna pada saat menggunakan aplikasi yang memiliki dua database. `this->load->database('group_name');`

Group_name adalah nama grup konfigurasi dari file konfigurasi Anda. Untuk menghubungkan secara manual `config['username'] = "myusername"; config['password'] = "mypassword"; config['database'] = "mydatabase"; config['dbdriver'] = "mysql"; config['dbprefix'] = ""; config['pconnect'] = FALSE; config['debug'] = TRUE; config['cache_on'] = FALSE; config['cachedir'] = ""; config['charset'] = "utf8"; config['dbcollat'] = "utf8_general_ci";`
`this->load->database(config);`

Atau Anda bisa mengirimkan nilai-nilai database Anda sebagai Data Source Name. DSN harus memiliki prototipe seperti ini: `dsn = 'dbdriver://username:password@hostname/database';` `this->load->database(dsn);`

3. Mengaktifkan manual dari model Selain kedua cara di atas Anda juga bisa mengaktifkan database pada saat loading model. Caranya adalah dengan mengeset TRUE pada parameter ketiga load model. Contoh : `this->load->model('Model_name', "", TRUE);`

Selain itu Anda juga bisa menggunakan konfigurasi tertentu untuk sebuah model. Contoh:

`manual_config['hostname'] = "localhost"; manual_config['username'] = "myusername"; manual_config['password'] = "mypassword"; manual_config['database'] = "mydatabase"; manual_config['dbdriver'] = "mysql"; manual_config['dbprefix'] = ""; manual_config['pconnect'] = FALSE; manual_config['debug'] = TRUE; this->load->model('Model_name', "", manual_config);`

CodeIgniter Model

Model pada CodeIgniter adalah sebuah kelas php yang berfungsi untuk menangani data. Ingat data bukan hanya dari database tetapi juga bisa dari File Text, Web Service atau layanan-layanan data lainnya. Contoh Model: class Blogmodel extends Model

`var $title = ""; var $content = ""; var $date = "";`
`function Blogmodel() { parent::Model(); }`
`function get_data() { $query = $this->db->get('entries', 10); return $query->result(); }`

Sebuah model sebenarnya tidak harus meng-extend class Model. Kelas model di-extend ketika hendak menggunakan fitur database pada CodeIgniter

saja. Semua Model harus diletakkan di dalam folder application/models. Agar dapat menggunakan model maka kita harus me-load model tersebut. Adapun perintah yang dapat digunakan untuk meload sebuah model adalah *this->load->model('Model_name');*this->Model_name->get_data();

Ketika sudah berhasil me-load sebuah model maka model tersebut akan menjadi sebuah property. Melalui property itulah Anda akan menggunakan semua fungsi yang ada di dalam model (perhatikan baris ke 2 pada contoh diatas *get_data* adalah fungsi di dalam kelas *Model_name*)

Melakukan Query pada Database

Untuk mendapatkan data pada database Anda harus melakukan query. Setelah query dilakukan barulah bisa mendapatkan data hasil query tersebut, baik dalam bentuk object maupun array. *query = this->db->query('QUERY SQL ANDA');*

Perlu ditekankan bahwa fungsi query di atas belum menghasilkan data apapun. Keluarannya hanya berupa Object(true) atau False. Ketika keluarannya adalah False maka query yang dilakukan berarti gagal. Tetapi jika true atau mengembalikan sebuah object maka query yang

dilakukan berarti berhasil. Dari object tersebut (variabel *query* kala umengacu contoh diatas) Anda dapat men
db->query('SELECT name, title, email FROM my_table');

foreach(query->result() as row) echo row->title; echo row->name; echo row->email;

echo 'Total Results: '.query->num_rows();

Contoh di atas adalah salah satu contoh penggunaan query (tapi ingat, pastikan Anda sudah memiliki konfigurasi database yang benar dan telah me-load library database. Perhatikan *query->result()*. Itu adalah contoh syntax untuk mendapatkan hasil query dalam bentuk array untuk mendapatkan hasil query dalam bentuk array asosiatif.

Query Return Value Seperti yang telah disebutkan di atas, setelah query dilakukan kita perlu memanggil fungsi tertentu untuk mendapatkan hasil dari query. Secara umum ada dua jenis tipe data yang dapat dihasilkan yaitu array dan object. Dari segi jumlah kita hanya dapat mengambil satu record (per record) atau keseluruhan record. • Mengembalikan Hasil Query Sebagai Kumpulan Array. Sebelum mengambil hasil query, ada baiknya mengecek terlebih dahulu apakah query tersebut memiliki hasil atau tidak. Perhatikan fungsi *result->result_array()*, itu adalah method yang digunakan untuk mengambil hasil query. Sedangkan fungsi *result->num_rows()* digunakan untuk mengetahui berapa jumlah record yang didapat. *result = this->*

*db->query('SELECT * FROM users');* *if (result->num_rows() > 0) foreach (result->result_array() as row) echo row['username']; echo row['email'];* • Mengembalikan Hasil Query Sebagai Kumpulan Object. Fungsi yang digunakan hampir sama dengan mengembalikan nilai query sebagai array. Cukup dengan memanggil method *result->result()*.

*result = this->db->query('SELECT * FROM users');* *if (result->num_rows() > 0) foreach (result->result() as row) echo row->username; echo row->email; // Mengembalikan Hasil Query Sebagai Row Object*
*db->query('SELECT * FROM users');* *if (query->num_rows() > 0) row = query->row(); echo row->username; echo row->email; // Mengembalikan Hasil Query Sebagai Row Object jadi kita hanya*
*db->query('SELECT * FROM users');* *if (query->num_rows() > 0) row = query->row(); echo row->username; echo row->email;*

Framework Codeigniter
Sebuah Panduan dan Best Practice
Framework CodeIgniter Sebuah Panduan dan Best Practice
Kata Pengantar

Codeigniter merupakan salah satu framework terbaik yang ada saat ini. Banyak website-website besar telah menggunakan codeigniter sebagai framework utama dalam membangun website maupun aplikasi mereka. Selain memudahkan dalam menggunakan, codeigniter juga cepat, mudah di kustomisasi dan integrasikan dengan library atau framework lainnya. Framework ini juga sempat menjadi perhatian pembuat php – Rasmus Lerdorf

“I like CodeIgniter because it is faster, lighter and the least like a framework.” - Rasmus Lerdorf

Dalam buku ini akan dibahas mengenai cara-cara penggunaan Codeigniter, mulai dari pengenalan tentang controller, model dan view, penggunaan codeigniter dan form, penggunaan database di codeigniter, studi-studi kasus hingga penggunaannya dengan library-library lain seperti jQuery dan jQuery UI. Akhir kata penulis menyadari bahwa penulisan dalam buku ini masih jauh dari sempurna. Oleh karena itu pertanyaan, kritik dan saran dapat di emailkan ke xibnoe@gmail.com atau dapat mengunjungi blog penulis di <http://www.koder.web.id>

Pekanbary, Juni 2011 Ibnu Daqiqil Id, M.Ti
DAFTAR ISI

Perkenalan CodeIgniter 2.0 1 Kenapa Menggunakan Framework? 2 Apa itu Codeigniter? 3 Apa sih Kelebihan CodeIgniter? 3 CodeIgniter 2.0 4 Apa itu MVC? 5 Jangan Belajar CodeIgniter!! 7 PHP Object Oriented Programming 9 Apasih PHP? 9 Sejarah PHP 9 Menggunakan PHP 9 Apa sih Object Oriented Programming (OOP)? 10 Apa itu Object 10 Apa itu Class? 10 Inheritance 12 Instalasi dan Konfigurasi CodeIgniter 13 Mempersiapkan Web Server 13 Instalasi CodeIgniter 13 Konfigurasi CodeIgniter 16 Kesepakatan Coding (Coding Standard) CodeIgniter 18 Hello CodeIgniter 21 Apaan sih Controller? 21 Controller dan View 24 Mempercantik URL Codeigniter 26 CodeIgniter Helper dan Library 29 Menggunakan Library dan Helper di CodeIgniter 29 Library CodeIgniter 30 Helper CodeIgniter 33 Membuat Library Sendiri 34 Menggunakan Library External 34 Kasus 1. Penanganan dan Validasi Form 37 Penanganan Form 37 Menggunakan Validasi Form 45 CodeIgniter Database 51 Connect ke Database 51 CodeIgniter Model 54 Melakukan Query pada Database 54 Query Return Value 55 Menggunakan Active Record 57 Kasus 2. CRUD dan Pagination Database 61 Kasus 3. Sistem Templating 72 Native CodeIgniter Tempating 72 Kasus 4. Sistem Authentikasi 78 Kasus 5. Image Gallery Sederhana 87 Kasus 6. Buku Tamu menggunakan Codeigniter 92 Membuat Table Guestbook 92 Konfigurasi Guestbook 92 Membuat Model Guestbook 93 Membuat Controller Dan View 94 Kasus 7. Membuat Shopping Cart Sederhana 100 Kasus 8. CodeIgniter dan Ajax 107 Kasus 9. Codeigniter dan jQuery AutoComplete 111 Kasus 10. Codeigniter dan Openflash Chart 116

Chapter 1 Perkenalan CodeIgniter 2.0

Akhir-akhir ini CodeIgniter menjadi sebuah framework yang hangat dibicarakan di Indonesia. Hampir semua milis dan forum PHP banyak membahas masalah

CodeIgniter. Terlebih lagi banyak perusahaan-perusahaan ternama di Indonesia (Kompas.com, okezone.com, urbanesia.com, bejubel.com, dan lain-lain) yang telah menggunakan CodeIgniter dalam produk mereka. Apa sih CodeIgniter? CodeIgniter adalah sebuah framework PHP yang dapat mempercepat pengembangan untuk membuat sebuah aplikasi web. Ada banyak library dan helper yang berguna didalamnya dan tentunya mempermudah proses development. Ibarat ingin membangun rumah maka Anda tidak perlu membuat semen, memotong kayu menjadi papan, mengubah batu menjadi porselen dan lain-lain. Anda cukup memilih komponen-komponen tersebut lalu dikombinasikan menjadi rumah yang indah.

Gambar1. Website-website indonesia yang menggunakan Codeigniter Jadi keuntungan yang didapat dalam penggunaan framework adalah : • Menghemat Waktu Pengembangan – Dengan struktur dan library yang telah disediakan oleh framework maka tidak perlu lagi memikirkan hal-hal tersebut, jadi Anda hanya fokus ke proses bisnis yang akan dikerjakan. • Reuse of code – Dengan menggunakan framework maka pekerjaan kita akan memiliki struktur yang baku, sehingga kita dapat menggunakannya kembali di proyek-proyek lainnya. • Bantuan komunitas - Ada komunitas-komunitas yang siap membantu jika ada permasalahan, selain itu juga bisa berbagi ilmu sehingga dapat meningkatkan kemampuan pemrograman kita. • Kumpulan best practice – sebuah framework merupakan kumpulan best practice yang sudah teruji. Jadi kita dapat meningkatkan kualitas kode kita.

Sebelum mendalami CodeIgniter lebih jauh, sebaiknya dipahami terlebih dahulu apa itu framework. Framework adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan, bahkan isu-isu kompleks yang ada. Sebuah framework telah berisi sekumpulan arsitektur/konsep-konsep yang dapat mempermudah dalam pemecahan sebuah permasalahan. Perlu diingat, framework bukanlah peralatan/tools untuk memecahkan sebuah masalah, tetapi sebagai ALAT BANTU. Framework hanya menjadi sebuah konstruksi dasar yang menopang sebuah konsep atau sistem yang bersifat “essential support” (penting tapi bukan komponen utama).

Kenapa Menggunakan Framework?

Salah satu alasan mengapa orang menggunakan framework terutama dalam membangun sebuah aplikasi adalah kemudahan yang ditawarkan. Didalam sebuah framework biasanya sudah tersedia struktur aplikasi yang baik, standard coding (1), best practice (2) dan design pattern (3), dan common function (4). Dengan menggunakan framework kita dapat langsung fokus kepada business process yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standar coding dan lain-lain. Dengan memanfaatkan design pattern dan common function yang telah ada di dalam framework maka hal tersebut dapat mempercepat proses pengembangan aplikasi. Kita tidak perlu membuat sesuatu fungsionalitas yang bersifat umum. Tanpa disadari ketika kita membangun sebuah aplikasi yang banyak melibatkan banyak fungsionalitas yang telah dibangun itu ternyata sama atau berulang-ulang. Dengan pengelempokkan itulah kita dapat mempercepat pengembangan aplikasi. Selain kemudahan dan kecepatan dalam membangun sistem, dengan menggunakan framework tertentu kita juga

dapat “menyeragamkan” cara kita mengimplementasikan kode program. Dengan framework kita akan “dipaksa” untuk patuh kepada sebuah kesepakatan. Selain itu juga akan memudahkan pengembang lain untuk mempelajari dan mengubah aplikasi yang telah dibuat apabila kode yang dihasilkan konsisten dan patuh pada sebuah aturan tertentu.

1 Standar Coding adalah sebuah standar yang harus diikuti oleh programmer untuk menulis code. 2 Best Practice adalah Kumpulan-kumpulan action yang telah teruji oleh para expert 3 Design Pattern adalah Teknik-teknik yang menjadi best practise 4 Common Function adalah fungsi-fungsi atau library yang telah umum digunakan dalam pengembangan sebuah sistem

Apa itu CodeIgniter?

CodeIgniter adalah sebuah web application framework yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan CodeIgniter adalah untuk membantu developer untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. CodeIgniter menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan. CodeIgniter diperkenalkan kepada public pada tanggal 28 februari 2006. CodeIgniter sendiri dibangun menggunakan konsep Model-View-Controller development pattern. CodeIgniter sendiri merupakan salah satu framework tercepat dibandingkan dengan framework lainnya. Pada acara frOSCon (August 2008), pembuat php Rasmus Lerdorf mengatakan dia menyukai codeigniter karena dia lebih ringan dan cepat dibandingkan framework lainnya ("because it is faster, lighter and the least like a framework.") Apa sih Kelebihan CodeIgniter?

CodeIgniter sangat ringan, terstruktur, mudah dipelajari, dokumentasi lengkap dan dukungan yang luar biasa dari forum CodeIgniter. Selain itu CodeIgniter juga memiliki fitur-fitur lainnya yang sangat bermanfaat, antara lain: • Menggunakan Pattern MVC. Dengan menggunakan pattern MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas. • URL Friendly. URL yang dihasilkan sangat url friendly. Pada CodeIgniter diminimalisasi penggunaan *GET* dan diganti dengan *URI*. • Kemudahan dalam mempelajari, mengintegrasikan Library dan helper. Jika kita membandingkan antara CodeIgniter dengan framework-framework lainnya maka beberapa poin yang membuat CodeIgniter unggul adalah : • Kecepatan. Berdasarkan hasil benchmark CodeIgniter merupakan salah satu framework PHP tercepat yang pernah ada. • Macamserta mudah mengadopsi library lainnya. • Dokumentasi lengkap dan jelas. Bahkan tanpa buku ini pun Code

anggota team. Jika sebuah framework sangat sulit dipelajari maka akan beresiko untuk memperlambat team development anda.

CodeIgniter 2.0 CodeIgniter versi 2.0 baru saja diluncurkan pada awal tahun 2011. Pada versi terbaru ini, terdapat banyak perubahan mendasar dari CodeIgniter daripada versi sebelumnya. Perubahan ini menjadikan CodeIgniter jauh lebih kaya dan matang dibandingkan framework lainnya. Perubahan itu diantaranya adalah: • Menghilangkan dukungan terhadap PHP4. PHP4 sudah tidak didukung oleh tim pengembang PHP, karena memberikan dukungan untuk PHP4 membuat CodeIgniter semakin ketinggalan dari segi fitur. • Menghilangkan fitur plugin. Plugin mirip dengan helper, bertujuan untuk menghilangkan kerancuan ini maka fitur ini dihilangkan pada CodeIgniter 2.0. • Menghilangkan fitur scaffolding. Fitur ini nyaris tidak pernah digunakan dan implementasinya

masih kurang bagus. • Penambahan library driver. Ini adalah library khusus dimana kita dapat membuat driver dari library yang telah kita buat. • Support query string dan command-line execution. Hal ini menjawab kesulitan-kesulitan yang dialami pada versi sebelumnya. • Penambahan library cache. Untuk meningkatkan kualitas aplikasi maka library cache baik menggunakan apc, memcached maupun file base. • Penambahan fitur package. Untuk mempermudah distribusi resource dalam sebuah folder.

Apa itu MVC?

MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter. MVC adalah singkatan dari Model View Controller. MVC sebenarnya adalah sebuah pattern/teknik pemograman yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Adapun komponen-komponen MVC antara lain: 1. Model Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL. 2. View View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller 3. Controller: Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

Gambar 2. Perbandingan PHP Biasa dan CodeIgniter

Perhatikan gambar 2. Pada eksekusi PHP, biasanya kita akan me-“load” semua library dan fungsi yang dibutuhkan kemudian digabungkan ke dalam HTML untuk di eksekusi oleh PHP. Untuk kasus sederhana cara tersebut masih baik-baik saja, tetapi ketika aplikasi tersebut menjadi kompleks/rumit maka kita akan sulit memeliharanya jika tidak didukung oleh arsitektur software yang bagus. Hal tersebut bisa terjadi disebabkan oleh code yang sama namun dibuat berulang-ulang, kode tidak konsisten dan lain-lain.

Gambar 3. Flow CodeIgniter

Jika dipetakan, alur kerja CodeIgniter akan tampak seperti gambar 3. Browser berinteraksi melalui controller. Controller-lah yang akan menerima dan membalas semua request dari browser. Untuk data maka controller akan meminta ke Model dan untuk UI/template akan meminta ke View. Jadi “Otak” dari aplikasi ada di controller, “Muka” aplikasi ada di view dan “Data” ada di model. Ketika

browser meminta sebuah halaman web maka router akan mencari controller mana yang harus menangani request tersebut. Setelah itu barulah si controller menggunakan model untuk mengakses data dan View untuk menampilkan data tersebut. Jangan Belajar CodeIgniter!!

Jangan coba-coba belajar CodeIgniter ketika Anda belum mengenal PHP. Berdasarkan pengalaman di forum dan milis, kebanyakan mereka yang belajar CodeIgniter tanpa memiliki dasar PHP yang baik akan mengalami banyak kesulitan, bukan dalam menguasai konsep CodeIgniter tetapi masih berkutat seputar PHP. Setidaknya Anda telah memahami konsep OOP pada PHP untuk mulai belajar CodeIgniter. Sebaiknya ketika Anda mempelajari CodeIgniter Anda sudah memahami PHP dan Object Oriented Programming. Akan lebih bagus lagi jika Anda sudah familiar dengan design pattern.

Chapter 2 PHP Object Oriented Programming Syarat utama untuk menguasai CodeIgniter adalah memahami PHP dan konsep Object Oriented Programming. Pada bab ini kita akan membahas sedikit mengenai PHP dan OOP. Apasih PHP? PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman di sisi server. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server. Misalnya kita merequest sebuah file PHP <http://www.koder.web/index.php>, maka webserver akan melakukan parsing terhadap file PHP tersebut. PHP parser yang menjalankan kode-kode PHP yang terdapat pada file index.php lalu mengirimkan hasilnya ke web browser. Sejarah PHP Pada tahun 1994, Rasmus Lerdorf mengembangkan sebuah perkakas yang digunakan sebagai engine parsing sebagai penerjemah/interpreter beberapa macro. Pada saat itu engine digunakan untuk pembuatan buku tamu, counter dan beberapa homepage. Ia menamai engine parser tersebut dengan nama PHP/FI. Dengan semangat opensource, para pengembang di dunia mencoba mengembangkan PHP/FI. Sampai pada tahun 1997, lebih dari 500.000 website di dunia menggunakan PHP/FI untuk menyelesaikan masalah seperti koneksi ke database, menampilkan content yang dinamis dan lain-lain. Pada juni 1998, PHP 3.0 dirilis. Pada saat itu PHP sudah mendukung multiflatform (bukan hanya linux), webserver, sejumlah database, SNMP (Simple Network Management Protocol) and IMAP (Internet Message Access Protocol). Menurut survei yang dilakukan oleh <http://netcraft.org>, saat ini pengguna PHP sudah mencapai 9,5 juta domain. Menggunakan PHP Ketika kita ingin mempelajari PHP, hal pertama yang harus disiapkan adalah sebuah web server, seperti yang sudah dibahas sebelumnya bahwa PHP merupakan bahasa pemrograman di sisi server. Banyak sekali jenis web server yang dapat digunakan, antara lain Apache, IIS, iPlanet, Omni, Xintami, dan lain-lain. Setelah web server terinstal barulah kita install PHP sebagai modul.

PHP dapat diperoleh secara gratis dengan cara men-download dari situs

resmi PHP (<http://www.php.net/downloads.php>) atau website lain yang mempunyai salinan program PHP untuk di-download. Sebelum menginstall PHP, terlebih dahulu harus meng-install web server. Setelah PHP terinstall. Anda dapat meletakkan source code PHP ke dalam folder yang akan di publish oleh web server dengan ekstensi file .php. Jika tidak ingin direpotkan oleh proses konfigurasi dan implementasi webserver, Anda dapat menggunakan software yang sudah terpaket menjadi satu antara PHP, APACHE, dan MySQL. Contoh Software-nya adalah XAMPP, PHPTRIAD, FOX Server, dan lain-lain.

Apa sih Object Oriented Programming (OOP)? Object Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada obyek. Semua data dan fungsi pada paradigma ini dibungkus dalam kelas-kelas atau obyek-obyek. Bandingkan dengan logika pemrograman terstruktur, setiap obyek dapat menerima pesan, memproses data, dan mengirim pesan ke obyek lainnya. OOP diciptakan untuk mengatasi keterbatasan pada bahasa pemrograman tradisional. Konsep dari OOP sendiri adalah semua pemecahan masalah dibagi ke dalam obyek. Dalam konsep OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai obyek. Apa itu Object Sederhananya, sebuah obyek adalah kumpulan dari variabel dan fungsi yang dibungkus menjadi satu entitas. Entitas tersebut dapat berupa variabel biasa. Sebuah obyek diciptakan melalui sebuah kelas atau dengan istilah instance of class. Obyek memiliki 2 elemen utama: 1. Attributes atau Properties: Yaitu nilai-nilai yang tersimpan dalam objek tersebut dan secara langsung maupun tidak langsung menentukan karakteristik dari obyek tersebut. 2. Method: Yaitu suatu aksi yang akan dijalankan atau dikerjakan oleh obyek tersebut.

Apa itu Class? Class dapat didefinisikan sebagai struktur data atau cetak biru dari suatu obyek. Lebih jelasnya adalah sebuah bentuk dasar atau blueprint yang mendefinisikan variabel, method umum pada semua obyek. Obyek sendiri adalah kumpulan variabel dan fungsi yang dihasilkan dari template khusus atau disebut class. Obyek adalah elemen pada saat run-time yang akan diciptakan, dimanipulasi, dan dibuang/di-destroy ketika eksekusi. Adapun class merupakan definisi statik dari himpunan obyek yang mungkin diciptakan sebagai instantiasi dari class. Perhatikan contoh class di berikut ini.

```

<code>?php /** Contoh kelas */ class Kendaraan /** end of class */ ?</code>

```

Contoh diatas memperlihatkan bagaimana mendefinisikan sebuah class dan meng-create sebuah instance dari class. Pada contoh di atas kita membuat sebuah kelas bernama “Kendaraan”. Dalam pembuatan kelas, pertama kita menggunakan kata kunci class yang diikuti oleh nama kelas, kemudian diakhiri dengan kurung kurawal. Di dalam kurung kurawal kita menuliskan kode-kode (berisi property dan method) supaya kelas tersebut bekerja seperti yang diinginkan. Kode-kode di dalam sebuah kelas terbagi menjadi dua kelompok, yaitu property dan method. Property adalah suatu wadah penyimpanan di dalam kelas yang bisa menampung informasi. Sederhananya property itu bisa disebut sebagai variabel di dalam kelas. Sedangkan method adalah fungsi yang ada di dalam kelas. Perhatikan contoh berikut. <code>?php /** contoh kelas */ class kendaraan</code>


```

    /*** property class ***/ private warna; private jumlah_pintu; private jumlah_roda; public harga;
public merk;
    /*** method class ***/ public function construct() echo 'ini adalah object
kendaraan. jbr /i';
    public function set_harga(harga) this-> harga = harga;
    public function show_harga() echo 'hargakendaraan : rp.'.this-> harga.'. jbr
/i';
    public function jalan() echo 'brrrroooooom!!!';
    /*** end of class ***/ /** contoh object **/ mengcreate object saya_a_dalah_object_dari_class_kendaraan_saya_d
new_kendaraan; saya_a_dalah_object-> set_harga(100000); saya_a_dalah_object->
show_harga();
    saya_a_dalah_object-> jalan();

```

Setiap property dan method memiliki identifier. Identifier-lah yang mengatur bagaimana property dan method digunakan. Identifier tersebut adalah public, private dan protected. Private berarti method atau property yang ada di dalam suatu kelas hanya bisa diakses di dalam kelasnya. Sedangkan pada method atau property yang bersifat public berarti method atau property tersebut bisa diakses di dalam dan di luar kelas. Inheritance Inheritance atau dalam bahasa Indonesianya disebut sebagai pewarisan adalah suatu cara untuk membuat sebuah kelas yang baru dengan menggunakan kelas lain yang sebelumnya sudah dibuat. Pada hubungan inheritance, sebuah class turunan mewarisi kelas leluhur (parent class). Oleh karena mewarisi, maka semua atribut dan method class dari induk akan dibawa (kecuali yang bersifat private), secara intrinsik menjadi bagian dari class anak. Adapun keuntungan yang didapat dari inheritance menambah fitur baru pada kelas anak dan mengubah atau mengganti fitur yang diwarisi dari kelas parent Adapun contoh kelas yang menggunakan konsep inheritance adalah

```

i?php /* inheritance.php */ class Bapak private nama = "Bapak"; function Bapak(n)
this-> nama = n; function Hallo() echo "Halo, saya this-> nama < br > ";
class Anak extends Bapak function Hai() Echo "hai dari kelas anak";
test = new Anak(" AnakdariBapak"); test-> Hallo();

```

Jika kita perhatikan di kelas anak sama sekali tidak memiliki fungsi hello, tetapi karena parentnya memiliki fungsi tersebut maka si anak dapat menggunakan fungsi tersebut. Selain menggunakan fungsi bapak, anak juga dapat menambah fungsi baru yaitu fungsi Hai.

Chapter 3 Instalasi dan Konfigurasi CodeIgniter Agar dapat menggunakan CodeIgniter, Anda harus menginstall dan melakukan konfigurasi terhadap CodeIgniter terlebih dahulu. Instalasi CodeIgniter sangatlah mudah. Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Hal-hal yang harus dipersiapkan dalam menginstall CodeIgniter adalah Mempersiapkan Web Server

Seperti yang kita tahu bahwa PHP adalah bahasa pemrograman website yang berjalan disisi server oleh karena itu untuk dapat menjalankan website yang dibuat dengan menggunakan PHP, di komputer harus terinstall aplikasi web server yang mendukung PHP. Banyak sekali aplikasi web server yang beredar, salah satu web server yang sangat terkenal dan juga bersifat bebas adalah web

server Apache, sebuah web server yang digunakan pada sebagian server yang ada di internet. Untuk website yang melibatkan basis data sebagai tempat penyimpanan datanya maka tidak perlu bingung karena CodeIgniter juga mendukung berbagai jenis server basis data, yaitu MySQL, MySQLi, MS SQL, PostgreSQL, Oracle, dan SQLite. CodeIgniter juga bisa dijalankan di semua sistem operasi yang bisa menjalankan aplikasi-aplikasi di atas, baik Windows, Linux, BSD dan yang lainnya. Instalasi CodeIgniter

Instalasi CodeIgniter sangat mudah. Meskipun namanya instalasi tetapi karena CodeIgniter adalah aplikasi berbasis website maka sebenarnya yang perlu dilakukan adalah meng-copy folder aplikasi CodeIgniter ke dalam folder htdocs atau DocumentRoot dari web server yang telah diinstall sebelumnya. Berbeda dengan instalasi software pada umumnya. Sebelum melakukan instalasi yang perlu dilakukan pertama kali adalah mendapatkan kode sumber dari CodeIgniter itu sendiri yang dapat didownload di <http://www.CodeIgniter.com/> (disediakan pula di dalam CD penyerta buku). Selanjutnya letakkan folder hasil ekstrak tadi di DocumentRoot web server, yaitu folder htdocs didalam direktori C:\apache\friendsbagi yang menggunakan XAMPP di Windows atau /var/www/html bagi yang menggunakan linux (semua tergantung dimana anda menginstall dan mengkonfigurasi webserver).

Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Adapun struktur utama dari CodeIgniter terbagi menjadi dua bagian, yaitu application dan sistem/core CodeIgniter. Application adalah tempat kita meletakkan code yang akan dibuat (bewarna merah dan hijau sedangkan sistem/core CodeIgniter yang bewarna ungu) . Folder sistem berisi library-library dan helper bawaan CodeIgniter.

Gambar 4. Struktur Direktori CodeIgniter Adapun susunan folder CodeIgniter secara default adalah:

- Folder application: disinilah aplikasi yang akan kita bangun diletakkan. Folder config - tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari database, router dan autoload aplikasi. Folder controllers - tempat menyimpan semua file controller. Folder errors - tempat menyimpan semua template error aplikasi. Folder helpers - tempat menyimpan helper-helper yang bukan berasal dari CI. Folder hooks - tempat menyimpan hook yang digunakan untuk mengubah alur fungsi dari core CI. Folder language - tempat menyimpan bahasa-bahasa yang akan di gunakan. Folder libraries - tempat menyimpan semua library buatan kita sendiri. Folder models - tempat menyimpan semua model. Folder views - tempat menyimpan semua file view aplikasi.

- Folder system menyimpan semua file baik itu file aplikasi yang dibuat maupun core framework-nya. Folder cache - tempat menyimpan semua cache yang dibuat caching library. Folder codeigniter - tempat menyimpan semua semua file internals CI. Folder database - tempat menyimpan semua driver database drivers dan class yang akan digunakan. Folder fonts - tempat menyimpan semua font yang digunakan image manipulation library. Folder helpers - tempat menyimpan semua helper core CI. Folder language - tempat menyimpan semua language core CI. Folder libraries - tempat menyimpan semua library core CI Folder logs - tempat menyimpan semua logs generated oleh

nya kita ingin meload library database, pagination dan lain-lain secara otomatis maka kita tinggal mengubah *autoload['libraries']* menjadi *:autoload['libraries']=array('database','session','pagination')*

- Routes.php. Konfigurasi di file ini bertujuan untuk menentukan kemana routing oleh library route akan dilakukan. Hal paling sederhana yang harus dilakukan adalah mengubah default controller (controller yang akan dibuka ketika tidak ada uri yang diberikan oleh browser). Misalnya website kita beralamat www.koder.web.id. Maka ketika membuka website tersebut maka secara otomatis CodeIgniter akan mengalihkan ke controller default, karena tidak disertakan di dalam url kita. Adapun yang perlu diubah adalah *route['default_controller'] = "welcome"*;

Kesepakatan Coding (Coding Standart) CodeIgniter

Sebelum melakukan coding menggunakan codeigniter maka ada baiknya kita mengetahui apasaja kesepakatan-kesepakatan yang ada di codeigniter. Kesepakatan-kesepakatan tersebut akan membuat kode kita lebih mudah dipahami oleh developer lainnya. Adapun kesepakatan tersebut diantaranya :

- PHP Closing Tag

Ketika kita menulis library, helper, controller ataupun model maka sebaiknya tidak menggunakan tanda penutup pada dokumen php *?>*. Hal tersebut dilakukan untuk mencegah adanya spasi atau karakter yang tidak diinginkan pada code kita sehingga membuat aplikasi error. Kita juga disarankan untuk memberikan informasi tentang akhir dokumen dan berisi path dokumen tersebut.

Contoh salah: *?>php echo "Here's my code!"; ?>*

Contoh Benar: *?php echo "Here's my code!";*

/ End of file myfile.php */ /* Location: ./system/modules/mymodule/myfile.php */*

- Penamaan Class dan Method Penamaan Class harus dimulai dengan huruf besar. Jika class menggunakan beberapa kata maka kata-kata tersebut dipisahkan menggunakan underscore dan bukan camelcase.

Contoh salah: *class superclass class SuperClass* Contoh Benar: *class Super_class*

Aturan diatas juga berlaku untuk method contohnya

Contoh kurang tepat: *function fileproperties()* //Tidak deskriptif dan memiliki underscore *function fileProperties()* // Tidak deskriptif dan underscore CamelCase *function getFileproperties()* // Kurang underscore *function getFileProperties()* // menggunakan CamelCase *get_the_file_properties_from_the_file()* //terlalu panjang Contoh Tepat : *function getFile_properties()* //deskriptif, pakai underscore, dan huruf kecil

Chapter 3 Hello CodeIgniter Pada bab ini kita akan memulai coding menggunakan CodeIgniter. Sebelum memulai memulai maka perlu kita pahami kembali apa itu Controller? Dan bagaimana hubungan controller dengan URI? Apakah Controller?

CodeIgniter adalah sebuah framework berbasis MVC. Sebuah Controller dapat dikatakan sebagai jantung dari suatu aplikasi, karena controller menentukan bagaimana permintaan HTTP yang harus ditangani. Sebuah kelas Controller adalah sebuah file yang terletak di dalam folder application/controller dan memiliki nama file yang sama dengan nama kelasnya dan dikaitkan dengan URL. Segmen-segmen pada URL pada codeigniter mencerminkan Controller yang dipanggil. Contoh: <http://www.domainsaya.com/class/fungsi/id> maka domain tersebut dapat dipecah menjadi bagian-bagian diantaranya:

Adapun komponen-komponen URL diatas adalah

- Konfigurasi Base Url, Bagian ini merupakan url yang kita masukkan pada konfigurasi base url yang merupakan url paling dasar untuk mengakses web atau aplikasi kita. Bagian ini merupakan Segmen URI pertama.

Untuk contoh pertama, kita akan membuat sebuah aplikasi hello codeigniter yang sederhana. Aplikasi tersebut akan diletakkan di folder hello dari htdocs anda, sehingga anda dapat mengaksesnya dengan membuka url `http://localhost/hello`. Perhatikan URL Berikut ini: `http://localhost/hello/index.php/blog/` Pada contoh di atas, CodeIgniter akan berusaha untuk menemukan sebuah controller bernama blog pada file `blog.php`, kemudian menampilkannya ke browser. Mari kita coba membuat sebuah controller sederhana sehingga dapat melihat apa yang terjadi. Dengan menggunakan teks editor, buatlah file bernama `blog.php` di dalam folder `application/controllers`. Setelah itu buatlah sebuah kelas `Blog` yang merupakan turunan dari kelas `CI_Controller`.

```
1. <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2. class Blog extends CI_Controller {
3.     function __construct() {
4.         parent::__construct();
5.     }
6.     function index() {
7.         // Your code here
8.     }
9.     function komentar() {
10.        // Your code here
11.    }
12. }
13. End of file Blog.php
14. Location: ./application/controllers/blog.php
```

Setelah itu apa yang terjadi? Coba buka situs Anda dengan menggunakan URL seperti ini:

`http://localhost/index.php/blog/`

Gambar 6. Tampilan Aplikasi Hello Codeigniter

Jika Anda melakukannya dengan benar, maka akan tampak tulisan Hello.. saya adalah contoh CodeIgniter pertama. Sekedar mengingatkan kembali, ketentuan penamaan class (baik controller maupun library) harus dimulai dengan huruf besar. Contoh:

```
<?php class Blog extends CI_Controller
```

Berikut ini contoh yang salah :

```
<?php class blog extends CI_Controller
```

Pada contoh di atas nama fungsi yang dipanggil adalah `index()`. Fungsi "Index" akan selalu dibaca secara default jika segmen kedua dari URI kosong. Cara lain untuk menampilkan "Hello CodeIgniter" adalah dengan mengakses url berikut: `localhost/index.php/blog/index/`

Segmen kedua dari URI yang menentukan fungsi mana yang akan dipanggil dari controller. Mari kita coba menambahkan fungsi baru ke controller:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    function index() {
        echo "Haloo.. saya adalah contoh codeigniter pertama";
    }
    function komentar() {
        echo "Ini adalah fungsi komentar";
    }
}
End of file Blog.php
Location: ./application/controllers/blog.php
```

Sekarang cobalah panggil URL berikut untuk melihat fungsi komentar: `localhost/index.php/blog/komentar/`

Maka hasilnya akan menampilkan pesan "Ini Fungsi Komentar". Jika ingin mengisi variabel `$param` maka Anda tinggal menambahkan sebuah URI lagi setelah `Blog`. Biasanya parameter ketik `localhost/index.php/blog/komentar/tutorial-ci/123`

Fungsi `post` pada controller dapat memiliki dua parameter yang akan dilewatkan pada URI segmen 3 dan 4 ("tutorial-ci" dan "123"). Jadi dapat

disimpulkan bahwa URI yang dipanggil dapat ditunjukkan seperti ini: local-host/index.php/[Controller]/[Fungsi]/[param]/[param]/...

Controller dan View

Pada contoh program Hello CodeIgniter diatas adalah cara untuk menampilkan tulisan "hello CodeIgniter" secara langsung di controller. Namun sebenarnya hal tersebut bisa dilakukan di view. Sebagai contoh: tuliskan kode program di bawah ini pada file application/controller/blog.php :

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller{functionconstruct()parent :: construct();
function index() this->load->view("hello_codeigniter");
/* End of file Blog.php */ /* Location: ./application/controllers/ Blog.php
*/

```

Selanjutnya buatlah file hello_codeIgniter.php di folder application > view(application/view/hello_codeIgniter.F
h1 > Hellosayaadalahview < /h1 >

Maka kode diatas akan memberikan hasil yang sama dengan contoh kasus pertama (tanpa menggunakan view), yang berbeda hanya tulisannya saja. Sebuah View sebenarnya hanyalah sebuah halaman web atau bagian dari halaman web, seperti sebuah header, footer, sidebar, dan lain-lain. Bahkan, View bisa menjadi fleksibel karena view dapat dimasukkan ke dalam view yang lain jika dibutuhkan. Untuk memanggil file view dapat digunakan fungsi seperti berikut ini (perhatikan baris 12):: this->load->view('nama_view');

Nama_view adalah nama file view Anda. Dan file tersebut harus diletakkan di dalam folder application/view.

1. Nama file view - Nama file yang hendak di-load yang terletak di dalam folder application/view

2. Data Parameter - Parameter ini digunakan untuk melewati data dari controller ke dalam view.

Contoh: buatlah file controller dengan nama blog.php dengan kode program seperti berikut ini:

```

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller{functionconstruct()parent :: construct();functionindex()data['judul']="J
blog"; data['isi'] = "Isi blog";this->load->view("blog_view",data);
Blog.php */

```

Selanjutnya buatlah file view bernama blog_view.php dengan kode program seperti berikut ini :

```

<h1><?php echo $judul;? >< /h1 ><p ><?php echo $isi; ?></p> <p><br /><Page
rendered in elapsed_time seconds < /p >

```

Jika halaman blog.php dipanggil maka tampilannya kurang lebih seperti berikut:

3. Output Parameter - Parameter ini akan di set true jika kita ingin menyimpan hasil view ke dalam sebuah variabel. Jika kita mengambil contoh code controller blog sebelumnya maka kita tinggal mengubah cara pemanggilan view menjadi

```

out =this->load->view("blog_view",data,true);

```

Code diatas berarti kita akan menyimpan hasil view kedalam sebuah variabel. Contohnya :

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller{functionconstruct()parent :: construct();

```

```
function index() this->load->view(\hello_codeigniter");
function view() data['judul'] = \Judulblog";data['isi'] = "Isi blog"; out =this->load->view("blog_view",data,true); echo out;
```

Mempercantik URL Codeigniter

Jika kita memperhatikan url yang ada di website kompas.com, okezone.com atau bejube.com maka kita akan menemukan url yang unik. Contohnya <http://bejube.com/jual/pakaian-wanita>, seakan-akan url tersebut bukan lah berasal dari script php biasa. Untuk membuat url seperti diatas

kita bisa menggunakan settingan codeigniter dan file .htaccess. adapun langkah yang dapat dilakukan adalah : 1. Membuat file .htaccess di folder root aplikasi anda. Adapun isi dari difile .htaccess adalah

```
RewriteEngine on RewriteCond !(index.php|images|robot.txt) RewriteRule (.*)
/index.php/1[L]2.Mengubahkonfigurasiapplication/configs/config.php.Membuang"index.php" padaconfig[
"index.php";menjadiconfig['index_page'] = """;di fileapplication/config/config.phpDenganmenyelesaikanta
/localhost/hello/index.php/blog/sudahdapatdiaksesdenganurlhttp : //localhost/hello/blog/tanpaindex.php
```

Dengan sampai pada tahap kedua sebenarnya sudah cukup untuk membuat url yang menarik. Selain itu untuk lebih advance lagi kita dapat menggunakan konfigurasi application/config/router.php.

Chapter 5 CodeIgniter Helper dan Library

CodeIgniter menyediakan dua jenis sarana yang dapat digunakan untuk membantu proses pengembangan aplikasi, antara lain: • Library Library dapat dikatakan sebagai kumpulan tools yang dapat digunakan untuk membantu sebuah proses. CodeIgniter telah menyediakan banyak library yang dapat digunakan secara langsung. Library pada dasarnya adalah sebuah kelas yang diletakkan di dalam folder system/libraries atau application/libraries. Library yang terletak di dalam folder system merupakan library bawaan dari CodeIgniter yang secara default di beri awalan CI. *Untuklibrarybuatansendiriharusdiletakkandidalamfolderapplication/libraries* proses yang berulang, seperti generatehtml, url, security, dan lain – lain.

Menggunakan Library dan Helper di CodeIgniter Agar dapat menggunakan library, helper dan plugin, maka ketiganya harus di load terlebih dahulu. Ada dua cara yang dapat dilakukan untuk men-load sebuah library dan helper antara lain: 1. Menambahkan Pada Konfigurasi Autoload Menambahkan sebuah library di autoload berarti seluruh aplikasi Anda akan dapat menggunakan library tersebut secara langsung. Sebaiknya library yang di load dengan cara ini adalah jenis library yang dipakai di seluruh aplikasi seperti login, template, dan lain-lain. 2. Menggunakan Perintah Loader Library Kita dapat juga menggunakan library loader untuk men-load library. Library loader adalah sebuah library CodeIgniter yang otomatis di load. Loader berfungsi sebagai pengatur dari sumberdaya-sumberdaya yang ada di dalam CodeIgniter seperti Model, View, Library, Helper, dan plugin. Cara penggunaannya adalah: *this->load->*library('nama_library');*this->load->*helper('nama_helper');*this->load->*plugin('nama_plugin');

Nama library, helper dan plugin harus di isi dengan huruf kecil.

Ketika sebuah library sudah di-load maka library tersebut menjadi property pada object Controller. Adapun cara penggunaannya adalah sebagai berikut: *this->nama_library->fungsi()*;

Library CodeIgniter Secara default CodeIgniter telah menyediakan library yang dapat digunakan secara langsung. Adapun library yang telah tersedia antara lain:

- Benchmarking Class Library ini digunakan untuk melakukan pengukuran terhadap aplikasi yang dibuat. Seperti untuk mengetahui berapa lama waktu eksekusi dan berapa jumlah memori yang digunakan. Library ini sudah digunakan dan di-load secara otomatis oleh CodeIgniter.
- Calendar Class Library ini berfungsi untuk menampilkan dan men-generate kalender.
- Cart Class Library ini berfungsi untuk membuat shopping cart (keranjang belanja). Library ini memiliki ketergantungan terhadap kelas session karena item-item chart tersebut disimpan di dalam session.
- Config Class Library ini berfungsi untuk mengambil data-data di dalam file konfigurasi. Library ini sudah di load secara otomatis oleh CodeIgniter.
- Database Class Library database digunakan untuk memanipulasi serta mendapatkan data dari sebuah sistem database. Secara default database yang sudah didukung oleh CodeIgniter adalah mysql, mssql, oracle, postgres. Sedangkan database yang tidak didukung secara langsung oleh CodeIgniter dapat di-jembatani dengan driver odbc.
- Email Class Library email digunakan untuk mengirimkan email. Pengiriman email tersebut bisa dilakukan dengan menggunakan protokol mail, sendmail dan smtp.sqawd
- Encryption Class Library Encryption digunakan untuk melakukan penyandian terhadap string tertentu
- File Uploading Class Library Uploading digunakan untuk meng-upload file. Kelas ini sudah dilengkapi dengan pengecekan jenis file, dan ukuran file.
- Form Validation Class Library form Validation digunakan untuk mengecek keabsahan form-form yang sudah di-submit oleh user.

- FTP Class Library FTP digunakan untuk meng-upload atau download file melalui ftp server.
- HTML Table Class Library HTML table adalah sebuah kelas yang berfungsi untuk men-generate table dari data array.
- Image Manipulation Class Library image manipulation berfungsi untuk mengolah gambar. Adapun fungsi-fungsi yang telah disediakan adalah Image Resizing, Thumbnail Creation, Image Cropping, Image Rotating dan Image Watermarking.
- Input and Security Class Library Input dan security berfungsi untuk menjamin bahwa inputan dari form telah bersih dari karakter-karakter “aneh”.
- Loader Class Library ini dapat disebut sebagai pengatur sumberdaya CodeIgniter. Semua sumberdaya yang ada akan dikendalikan oleh kelas ini. Library ini sudah di-load secara otomatis oleh CodeIgniter.
- Language Class Library language digunakan untuk mengatur bahasa apa yang akan dipakai oleh CodeIgniter.
- Output Class Library Output bertujuan untuk meng-handle output dari CodeIgniter, mulai dari cache sampai ke profiling bisa dilakukan kelas ini.
- Pagination Class Untuk mem-paginate hasil database untuk performance dan usability, kita bisa mengontrol berapa banyak record untuk ditampilkan disetiap halaman website, berapa banyak record untuk ditarik dari database dan tampilan dari bagian pagination
- Session Class Library Session dapat digunakan untuk memelihara informasi status tentang user (seperti layaknya session di PHP). Tetapi Library ini tidak menggunakan session built-in dari PHP, Library Session men-generate session datanya sendiri yang disimpan di dalam Cookies.
- Trackback Class Library Trackback digunakan untuk mengirim dan menerima data trackback.
- Template Parser Class Library Template Parser digunakan untuk membuat

template yang berisi parsable pseudo – templates. • Unit Testing Class Library Unit Testing digunakan untuk unit test function dalam aplikasi yang sedang dibuat. CodeIgniter menyediakan fungsi evaluasi dan dua fungsi hasil dalam library ini.

- URI Class Library URI digunakan untuk memarsing URL, lalu memecahnya ke dalam beberapa segmen dan kemudian di-passing ke controller atau disimpan sebagai variabel.
- User Agent Class Library User Agent digunakan untuk mengidentifikasi browser, mobile device, atau robot yang mengunjungi website. Kita juga bisa menggunakannya untuk mendeteksi dukungan bahasa, sekumpulan karakter, dan referrer.
- XML-RPC Class Library XML-RPC digunakan untuk men-setup klien XML-RPC dan server.
- Zip Encoding Class Library Zip Encoding digunakan untuk membuat file ZIP baik yang berjenis teks maupun data binary.

Helper CodeIgniter Helpers seperti namanya akan membantu Anda membangun aplikasi dengan tugas tertentu. Tidak seperti library, helper bukanlah Object Oriented tapi berupa prosedural. Setiap helper berisi satu atau lebih fungsi, masing-masing berfokus pada tugas tertentu yang tidak ada ketergantungan dengan fungsi lainnya. Helper dapat juga di-load secara otomatis dalam /system/application/config/autoload.php. Adapun helper yang ada di dalam CodeIgniter antara lain:

- Array — Helper array berisi fungsi yang membantu pekerjaan berhubungan dengan array. Sebagai contoh fungsi `random_element()` mengambil array sebagai input dan mengembalikan satu elemen acak dari array tersebut. Fungsi `download_helper()` membantu untuk mengunduh file dari URL. Fungsi `force_download()` menghasilkan header server yang memaksa data untuk di-download. `file_helper()` membantu untuk membaca, menulis, dan menghapus file. `form_helper()` membantu untuk membangun form. `html_helper()` membantu untuk membuat blok HTML. `language_helper()` membantu untuk mengubah kata-kata menjadi bentuk plural atau singular, memberlakukan camel case atau mengubah kata-kata yang dipisahkan oleh spasi menjadi frase yang digarisbawahi, sebagai contoh fungsi `singular_bisamengubahkata()` membantu untuk mengelola emoticons. `string_helper()` membantu untuk bekerja dengan string, seperti fungsi `random_string()` yang akan membuat string acak.

- Text — Helper text berisi fungsi-fungsi yang membantu Anda bekerja dengan teks. Sebagai contoh: fungsi `word_limiter()` dapat membatasi string ke sejumlah kata tertentu yang berguna untuk membatasi jumlah kata dalam sebuah paragraf. Fungsi `auto_typography()` akan mengkonversi line break ke `
` dan mengkonversi tanda kutip, dash, dan ellipsis dengan baik. `url_helper()` membantu untuk bekerja dengan URL. Anda akan menggunakan fungsi `base_url()` dan `anchor()` dalam setiap halaman.

Membuat Library Sendiri Untuk mendapatkan kode yang bagus sebaiknya fungsionalitas yang sama itu di bungkus menjadi sebuah library. Sebuah library adalah sebuah kelas yang diletakkan pada folder application/libraries. `Get_instance()` adalah sebuah fungsi yang digunakan untuk mendapatkan instance dari library tersebut. `DataURI` dapat diakses dari library yang sudah di-load secara otomatis oleh CodeIgniter. Meng

CodeIgniter sangat mudah untuk mengadopsi library yang bersifat external. Library yang paling mudah untuk diadopsi adalah sebuah single class tanpa parameter pada constructor. Untuk kelas yang memiliki konstruktor lebih dari satu maka sebaiknya kelas tersebut diubah terlebih dahulu atau jika kelas library tersebut terdiri atas beberapa class maka lebih mudahnya kita membuat sebuah library factory dari library tersebut. Contoh 1. Library CSV Reader

Sebagai contoh kita akan menggunakan sebuah library yang berfungsi untuk membaca file CSV ke dalam memori.. Adapun isi dari librari tersebut adalah

```

i?php if(!defined('BASEPATH'))exit('No direct script access allowed'); /**
 * CSVReader Class * @author Pierre-Jean Turpeau * @link http://www.CodeIgniter.com/wiki/CSVReader
 */
class csv_reader
{
    var $fields; var $separator=';'; var $enclosure='\"'; var $max_row_size = 4096;
    /** * Parse a file containing CSV formatted data. * * @access public *
    @param string * @param boolean * @return array */
    function parse_file($filepath, $namedFields = true) {
        $content=false; $file =
        fopen($filepath, 'r'); if($namedFields) {
            $fields = fgetcsv($file, $this->max_row_size, $this->
            $separator, $this->enclosure);
            while(($row = fgetcsv($file, $this->max_row_size, $this->
            $separator, $this->enclosure)) != false) {
                if($row[0] != null) // skip empty lines
                    if(!$content) $content=array();
                if($namedFields) {
                    $items=array();
                    foreach($this->fields as $id) {
                        if(isset($row[$id]))
                            $items[$id] = $row[$id];
                    }
                }
                return $content;
            }
        }
    }
}

```

Agar dapat menggunakan library tersebut maka lakukan langkah-langkah berikut ini:

- Copy Library Ke Direktori application/libraries Untuk menggunakan library buatan kita sendiri atau library eksternal maka langkah yang pertama yang harus dilakukan adalah meletakkan library itu di direktori application/libraries.

Nama file dan nama kelas harus sama. Contoh nama file adalah `csv_reader.php` maka nama kelasnya adalah `csv_reader`.

Perhatikan controller berikut ini

```

1. i?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Welcome extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('csv_reader');
        $data = $this->csv_reader->parse_file(FCPATH.'data_nilai.csv');
        echo $pre > ";
        print_r($data);
    }
}
/* End of file welcome.php */
/* Location: ./application/controllers/welcome.php */

```

Pada baris ke-13 kita akan mencoba menggunakan library `csv_reader`, kelas tersebut akan memparsing sebuah file CSV. Lalu hasil parsingnya akan ditampilkan.

Chapter 6 Kasus 1. Penanganan dan Validasi Form Pada bab ini akan dibahas mengenai penanganan form dan cara melakukan validasi terhadap sebuah form inputan. Penanganan form artinya bagaimana cara kita untuk mendapatkan dan mengolah data yang dikirimkan oleh pengguna melalui form. Untuk membuat form sendiri, sebaiknya kita menggunakan helper form. Setelah kita berhasil mendapatkan data dari user, kita harus memvalidasi data-data yang telah dikirimkan tersebut agar aplikasi yang kita bangun menjadi aman. Inputan yang tidak divalidasi membuat aplikasi kita rentan terhadap serangan seperti Sql Injection, Xss dan lain-lain. Penanganan Form

Untuk menangani form kita membutuhkan library input. Library tersebut sudah dipanggil secara otomatis oleh codeigniter. Fungsi yang tersedia pada library ini selain untuk menangani form juga memiliki fungsi security/keamanan. Contohnya untuk menghandle atau memfilter xss kita dapat menggunakan fungsi `xssfilter.nama = $this->input->xssfilter($this->input->post('nama'))`;

Jika kita ingin melindungi seluruh aplikasi dari dengan xss filter maka kita bisa menconfignya dengan mengubah konfigurasi di `system/application/config/config.php` dari `config['global_xssfiltering'] = FALSE`;

Menjadi

`config['global_xssfiltering'] = TRUE`;

Selain kedua cara diatas, kita dapat juga menggunakan parameter kedua dari fungsi post atau get maka kita akan mendapatkan hasil yang sama dengan kedua cara diatas, contohnya `this->input->post('nama',TRUE);` `this->input->get('nama', TRUE);` Fungsi post digunakan untuk menangkap inputan POST sedangkan get digunakan untuk menangkap inputan GET. Selain itu fungsi input juga dapat digunakan untuk cookie, variabel SERVER, dan user agent `this->input->cookie('nama_cookie',TRUE);` `this->input->ip_address();` `this->input->server('PATH_INFO',TRUE);`

Selain library input, Codeigniter juga menyediakan sebuah helper untuk mempermudah penanganan form. Helper tersebut adalah helper form. Helper tersebut membantu kita dalam membuat form pada view. Adapun cara meload helper form adalah `this->load->helper('form');`

Ada banyak fungsi yang disediakan oleh helper ini diantaranya `form_open()`, `form_close()`, `form_open_multipart()` *lain.* Untuk lebih detailnya anda dapat melihat pada dokumentasi codeigniter. Pada kasus ini, fungsi yang banyak digunakan adalah `form_open()` pada html. Contoh :

```
attributes = array('class' => 'email', 'id' => 'myform'); echo form_open('form/send', attributes);
Maka akan menghasilkan
<form method="post" accept-charset="utf-8" action="http://localhost/index.php/email/send"
class="email" id="myform" />
```

Fungsi `input` berfungsi untuk menggenerasi tag input pada html. Fungsi ini dapat dipakai dengan dua cara. Cara pertama adalah dengan menggunakan `echo form_input('username', 'ibnoe');`

akan menghasilkan `<input type="text" name="username" id="username" value="ibnoe" />`
 Cara kedua adalah dengan mempassing array kedalam fungsi tersebut. Cara ini adalah cara yang advance dan sangat berguna jika kita ingin mengcustomisasi inputan. Contohnya: `data = array('name' => 'username', 'id' => 'username', 'value' => 'ibnoe', 'maxlength' => '100', 'size' => '50', 'style' => 'width: 50');`

```
echo form_input(data);
Akan menghasilkan output :
<input type="text" name="username" id="username" value="ibnoe" maxlength="100"
size="50" style="width:50
```

Untuk contoh menangani form maka ikutilah langkah-langkah berikut ini.

1. Membuat Controller Hitung

Perlu diingat hal pertama yang harus di konfigurasi pada setiap aplikasi adalah `BASE_URL`. Setelah melakukan konfigurasi `base_url` pada `file application/config/config.php` maka kita akan mendapatkan `Application/controllers/hitung.php` 1. `<?php if (!defined('BASEPATH')) exit('No direct script access allowed');` 2. `load->helper(array('url','form'));` 9. 10. 11. `function index()` 12. 13. `this->load->view('menu_hitung');` 14. 15. 16. `function perkalian()` 17. 18. `data['v1']=(int)$this->input->post('v1',true);` 19. `data['v2']=(int)$this->input->post('v2',true);` 20. `data['hasil']=data['v1']*data['v2'];` 21. `this->load->view('perkalian',data);` 22. 23. 24. `function pembagian()` 25. 26. `data['v1']=(int)$this->input->post('v1',true);` 27. `data['v2']=(int)$this->input->post('v2',true);` 28. `if (data['v2'] > 0)` 29. `data['hasil']=data['v1']/data['v2'];` 30. `else` 31. `data['hasil']='Error, v2 tidak boleh 0!';` 32. `this->load->view('pembagian',data);` 33. 34.

Pada baris ke 8, dilakukan loading terhadap helper url dan form sekaligus. Loading tersebut dilakukan dalam fungsi konstruktor karena kedua helper


```

</h1> 38. 39. <p>Silakan masukan data berikut!!</p> 40. <?php
echo form_open('hitung/pembagian');? > 41. <?php echo form_input('v1',v1);?</
/ 42. <?php echo form_input('v2',v2);?</br> 43. 44. <?php echo form_submit('submit','Hitung!!');? >
45. <?php echo form_close();? ><br> 46. Hasil :<?php echo hasil;?</ 47. 48.
</br></Page rendered in elapsed time seconds </p> 49. </body></html>

```

Berikut ini adalah tampilan dari form pembagian

Gambar 9. Screenshoot view pembagian

Catatan :

Menggunakan Validasi Form

Sebenarnya sampai pada tahap kedua di atas kita telah mampu membuat sebuah aplikasi perkalian dan pembagian sederhana, tetapi sebuah aplikasi yang baik harus memiliki sebuah kontrol terhadap inputan user. Validation form sangat penting pada aplikasi. Sebuah aplikasi akan memiliki banyak inputan dari pengguna dan semua inputan untuk pengguna harus aman. CodeIgniter memiliki sebuah library Form Validation Library yang akan membantu kita untuk membuat sebuah validasi yang cepat, mudah dan aman. Pada contoh validasi form ini hampir sama dengan 2 langkah diatas, tetapi memiliki sedikit penambahan terutama dalam fungsi perkalian dan pembagian. Perhatikan controller hitung berikut ini (controller berikut ini berasal dari contoh sebelumnya dengan penambahan).

```

Application/controllers/hitung.php 1. <?php if (! defined('BASEPATH')) exit('No direct script access allowed'); 2. 3. class Hitung
extends CI_Controller 4.5.function construct() 6.7.parent :: construct(); 8.this->load-
helper(array('url','form')); 9. 10. 11. function index() 12. 13. this->
load->view('menu_hitung'); 14.15.16.17.function perkalian() 18.19.this->load-
library('form_validation'); 20.this->form_validation->set_rules('v1','Variabel1',21.'required|integer'); 22.this-
form_validation->set_rules('v2','Variabel2',23.'required|integer'); 24.if (this-
form_validation->run()) 25.26.int) this->input->post('v1',true); 27. int) this->
input->post('v2',true); 28.data['hasil']=data['v1']*data['v2']; 29. 30. else
31. 32. 33. 34. data['hasil']=0;
35. 36. this->load->view('perkalian',data); 37. 38. 39. function
pembagian() 40. 41. this->load->library('form_validation'); 42.this-
form_validation->set_rules('v1','Variabel1',43.'required|is_natural|zero'); 44.this-
form_validation->set_rules('v2','Variabel2',45.'required|is_natural|zero'); 46.if (this-
form_validation->run()) 47.48.int) this->input->post('v1',true); 49. int) this->
input->post('v2',true); 50.data['hasil']=data['v1']/data['v2']; 51. 52. else
53. 54. 55. 56. data['hasil']=0; 57.58.59.this->load->view('pembagian',data); 60.61.62.

```

Perhatikan baris 19-24 pada fungsi perkalian, disana kita me-load sebuah library bernama form_validation. Library tersebut memiliki fungsi untuk melakukan validasi terhadap inputan user.

```

form_validation->set_rules('v1','Variabel1','required|integer');

```

Baris di atas merupakan salah satu contoh untuk menentukan aturan inputan pada form. Parameter pertama (berisi v1) merupakan name dari inputan yang akan divalidasi (harus sama dengan attribut name pada tag input contoh: <input name="v1">). Parameter kedua adalah nama dari inputan tersebut. Nama tersebut dapat berbeda atau sama dengan name pada parameter pertama. Parameter ketiga merupakan aturan dari inputan tersebut. Aturan-aturan tersebut harus dipatuhi agar sebuah form dapat diproses. Aturan-aturan

tersebut dipisahkan oleh tanda —. Pada contoh di atas berarti inputan v1 harus diisi (required) dan berisi data integer(integer). Adapun aturan-aturan yang tersedia di dalam CodeIgniter adalah:

Aturan Keterangan required Isi inputan tidak boleh kosong. matches Isi inputan harus sama dengan inputan tertentu contoh

matches[password] min_{length} Panjang inputan harus memiliki jumlah minimal karakter, contoh penggunaan `z.alpha_n` numeric Inputan harus berisi karakter alfabet dan numeric. `alpha_dash` Inputan harus berisi semua huruf `z`, `underscores` atau `dashes`. `numeric` Inputan hanya boleh berisi angka/huruf numeric. `integer` Inputan hanya boleh 0, 1, 2, 3, dan seterusnya. `is_natural` `is_natural_or_zero` Inputan hanya boleh berisi bilangan natural saja kecuali nol : 1, 2, 3, dan seterusnya. `valid_email` Inputan harus berisi format email yang benar. `valid_emails` Inputan harus berisi

Pada baris-46, kita melakukan pengecekan terhadap rule-rule yang telah diset (`this->form_validation->run()`). Fungsi tersebut akan menghasilkan nilai TRUE apabila semua

rule terpenuhi dan menghasilkan nilai FALSE jika sebaliknya. Selain fungsi-fungsi diatas Anda juga dapat membuat custom validation jika aturan-aturan di atas tidak ada yang memenuhi dengan kebutuhan Anda. 1. `!DOCTYPE html`; 2. `<html lang="en">`; 3. `<head>`; 4. `<meta charset="utf-8">`; 5. `<title>` Perkalian menggunakan validasi!!; `</title>`; 6. 7. `<style type="text/css">`; 8. 9. `body` 10. `background-color: fff`; 11. `margin: 40px`; 12. `font-family: Lucida Grande, Verdana, Sans-serif`; 13. `font-size: 14px`; 14. `color: 4F5155`; 15. 16. 17. `a` 18. `color: 003399`; 19. `background-color: transparent`; 20. `font-weight: normal`; 21. 22. 23. `h1` 24. `color: 444`; 25. `background-color: transparent`; 26. `border-bottom: 1px solid D0D0D0`; 27. `font-size: 16px`; 28. `font-weight: bold`; 29. `margin: 24px 0 2px 0`; 30. `padding: 5px 0 6px 0`; 31. 32. 33. `</style>`; 34. `</head>`; 35. `<body>`; 36. 37. `<h1>` Perkalian!; `</h1>`; 38. `<?php echo validation_errors();? >` 39. `< p > Silakan masuk dan data berikut!! </p>` 40. `<?php echo form_open('hitung/perkalian');? >` 41. `<?php echo form_input('v1',v1);?>` 42. `<?php echo form_input('v2',v2);?>` 43. `<?php echo form_submit('submit','Hitung!!');? >` 44. `<?php echo form_close();? >` 45. `
` 46. `Hasil :<?php echo hasil;?>` 47. 48. `</br>`; 49. Page rendered in elapsed `time` seconds `</p>` 49.50. `</body>` `></html>`

Pada view, kita cukup menambahkan satu fungsi yang berguna untuk menampilkan kesalahan yang terjadi. Perhatikan baris 38. Fungsi `validation_errors` adalah fungsi untuk menampilkan kesalahan

Gambar 10. Form perkalian dengan tampilan error

Begitu juga untuk view pembagian, kita tinggal menambah fungsi `validation_errors` untuk menampilkan error

1. `!DOCTYPE html`; 2. `!DOCTYPE html`; 3. `<html lang="en">`; 4. `<head>`; 5. `<meta charset="utf-8">`; 6. `<title>` Pembagian menggunakan validasi!!; `</title>`; 7. 8. `<style type="text/css">`; 9. 10. `body` 11. `background-color: fff`; 12. `margin: 40px`; 13. `font-family: Lucida Grande, Verdana, Sans-serif`; 14. `font-size: 14px`; 15. `color: 4F5155`; 16. 17. 18. `a` 19. `color: 003399`; 20. `background-color: transparent`; 21. `font-weight: normal`; 22. 23. 24. `h1` 25. `color: 444`; 26. `background-color: transparent`; 27. `border-bottom: 1px solid D0D0D0`; 28. `font-size: 16px`; 29. `font-weight: bold`; 30. `margin: 24px 0 2px 0`; 31. `padding: 5px 0 6px 0`; 32. 33. 34. 35. `</style>`; 36. `</head>`; 37. `<body>`; 38. 39. `<h1>` pembagian!; `</h1>`; 40. `<?php echo validation_errors();? >` 41. `< p > Silakan masuk dan data berikut!! </p>` 42. `<?php echo form_open('hitung/pembagian');? >` 43. `<?php echo form_input('v1',v1);?>`

```

/ 44. <?php echo form_input('v2',v2);?><br> 45. 46. <?php echo form_submit('submit','Hitung!!');? >
47. <?php echo form_close();? ><br> 48. Hasil :<?php echo hasil;?> 49. 50.
<?php echo $Page rendered in elapsed_time seconds < /p > 51.52. < /body ><
/html >

```

View diatas akan memberikan tampilan seperti gambar 11 berikut ini.

Gambar 11. Form perkalian dengan tampilan error

Chapter 7 CodeIgniter Database

CodeIgniter mendukung banyak jenis database misalnya MySQL, PostgreSQL, Oracle, dan lain-lain. Dukungan database dari CodeIgniter berupa penyediaan beberapa driver database yang sekaligus juga memiliki fungsi sekuriti, caching dan active record.

Connect ke Database

CodeIgniter memiliki sebuah file konfigurasi yang memungkinkan Anda menyimpan konfigurasi untuk melakukan koneksi ke database (username, password, nama database, dan lain-lain). File konfigurasi terletak berada di application/config/database.php. Pengaturan konfigurasi disimpan dalam bentuk array multi-dimensi dengan prototipe seperti berikut ini: `db['default']['hostname'] = "localhost"; db['default']['username'] = "root"; db['default']['password'] = ""; db['default']['database'] = "database"; db['default']['dbprefix'] = ""; db['default']['pconnect'] = TRUE; db['default']['dbdebug'] = FALSE; db['default']['cache_on'] = FALSE; db['default']['cachedir'] = ""; db['default']['charset'] = "utf8"; db['default']['dbcollat'] = "utf8_general_ci";`

Kita dapat memiliki beberapa konfigurasi database. Alasan kita menggunakan array multi-dimensi adalah agar memungkinkan Anda secara opsional menyimpan beberapa set nilai-nilai koneksi. Jika, misalnya, Anda menjalankan aplikasi di beberapa kondisi lingkungan/environment (development, production, testing, dll) di bawah satu instalasi, maka Anda dapat mengatur kelompok konfigurasi untuk masing-masing environment. Sebagai contoh, Anda sedang menjalankan aplikasi di environment test maka Anda tinggal menambahkan konfigurasi berikut ini dan mengubah default konfigurasinya (mengeset `active_group = "test"`) menjadi test: `db['test']['hostname'] = "localhost"; db['test']['username'] = "root"; db['test']['password'] = ""; db['test']['database'] = "database_name"; db['test']['dbdriver'] = "mysql"; db['test']['dbprefix'] = ""; db['test']['pconnect'] = TRUE; db['test']['dbdebug'] = FALSE; db['test']['cache_on'] = FALSE; db['test']['cachedir'] = ""; db['test']['charset'] = "utf8"; db['test']['dbcollat'] = "utf8_general_ci"; active_group = "test";`

Untuk connect ke database ada beberapa cara yang disediakan oleh CodeIgniter diantaranya :

1. Menambahkan Database Library Sebagai Autoload Library Untuk connect ke database Anda bisa menambahkan database sebagai autoload library di file application/config/autoload.php. Cara ini sangat sederhana, cukup menambahkan kata "database" ke dalam autoload library sehingga menjadi : `autoload['libraries'] = array("database");`

Ketika menambahkan autoload pastikan bahwa database sudah terkonfigurasi dengan benar untuk menghindari kesalahan.

2. Mengaktifkan Manual Dari Library Database Jika hanya ada beberapa halaman website yang memerlukan konektivitas database, maka untuk optimalisasi lakukan koneksi

ke database secara manual, cukup dengan menambahkan baris kode di bawah ini pada tiap fungsi tempat yang membutuhkan koneksi ke database atau dalam konstruktor kelas Anda untuk membuat database yang tersedia secara global di kelas. *this->load->database()*;

Jika fungsi di atas tidak berisi informasi apapun di parameter pertama akan menyambung ke group konfigurasi yang aktif. Untuk memilih kelompok tertentu dari file konfigurasi, Anda dapat melakukan seperti pada contoh berikut. Hal tersebut berguna pada saat menggunakan aplikasi yang memiliki dua database. *this->load->database('group_name')*;

Group_name adalah nama grup konfigurasi dari file konfigurasi Anda. Untuk menghubungkan secara manual *config['username'] = "myusername"; config['password'] = "mypassword"; config['database'] = "mydatabase"; config['dbdriver'] = "mysql"; config['dbprefix'] = ""; config['pconnect'] = FALSE; config['debug'] = TRUE; config['cache_on'] = FALSE; config['cachedir'] = ""; config['charset'] = "utf8"; config['dbcollat'] = "utf8_general_ci";*
this->load->database(config);

Atau Anda bisa mengirimkan nilai-nilai database Anda sebagai Data Source Name. DSN harus memiliki prototipe seperti ini: *dsn = 'dbdriver://username:password@hostname/database'; this->load->database(dsn)*;

3. Mengaktifkan manual dari model Selain kedua cara di atas Anda juga bisa mengaktifkan database pada saat loading model. Caranya adalah dengan mengeset TRUE pada parameter ketiga load model. Contoh : *this->load->model('Model_name', "", TRUE)*;

Selain itu Anda juga bisa menggunakan konfigurasi tertentu untuk sebuah model. Contoh:

manual_config['hostname'] = "localhost"; manual_config['username'] = "myusername"; manual_config['password'] = "mypassword"; manual_config['database'] = "mydatabase"; manual_config['dbdriver'] = "mysql"; manual_config['dbprefix'] = ""; manual_config['pconnect'] = FALSE; manual_config['debug'] = TRUE; this->load->model('Model_name', "", manual_config);

CodeIgniter Model

Model pada CodeIgniter adalah sebuah kelas php yang berfungsi untuk menangani data. Ingat data bukan hanya dari database tetapi juga bisa dari File Text, Web Service atau layanan-layanan data lainnya. Contoh Model: *class Blogmodel extends Model*

var \$title = ""; var \$content = ""; var \$date = "";
function Blogmodel() { parent::__construct(); }
function get_data() { \$query = \$this->db->get('entries', 10); return \$query->result(); }

Sebuah model sebenarnya tidak harus meng-extend class Model. Kelas model di-extend ketika hendak menggunakan fitur database pada CodeIgniter saja. Semua Model harus diletakkan di dalam folder application/models. Agar dapat menggunakan model maka kita harus me-load model tersebut. Adapun perintah yang dapat digunakan untuk memload sebuah model adalah *this->load->model('Model_name')*; *this->Model_name->get_data()*;

Ketika sudah berhasil me-load sebuah model maka model tersebut akan menjadi sebuah property. Melalui property itulah Anda akan menggunakan semua

fungsi yang ada di dalam model (perhatikan baris ke 2 pada contoh diatas *get_data* adalah fungsi di dalam kelas *Model_name*)

Melakukan Query pada Database

Untuk mendapatkan data pada database Anda harus melakukan query. Setelah query dilakukan barulah bisa mendapatkan data hasil query tersebut, baik dalam bentuk object maupun array. *query = this->db->query('QUERY SQL ANDA');*

Perlu ditekankan bahwa fungsi query di atas belum menghasilkan data apapun. Keluarannya hanya berupa Object(true) atau False. Ketika keluarannya adalah False maka query yang dilakukan berarti gagal. Tetapi jika true atau mengembalikan sebuah object maka query yang

dilakukan berarti berhasil. Dari object tersebut (variabel *query* kalau mengacu contoh diatas) Anda dapat men
db-> query('SELECT name, title, email FROM my_table');

foreach(query-> result() as row) echo row-> title; echo row-> name; echo row-> email;

echo 'Total Results: ' . query-> num_rows();

Contoh di atas adalah salah satu contoh penggunaan query (tapi ingat, pastikan Anda sudah memiliki konfigurasi database yang benar dan telah me-load library database. Perhatikan *query-> result()*. Itu adalah contoh syntax untuk mendapatkan hasil query dan *result_array()* untuk mendapatkan hasil query dalam bentuk array asosiatif.

Query Return Value Seperti yang telah disebutkan di atas, setelah query dilakukan kita perlu memanggil fungsi tertentu untuk mendapatkan hasil dari query. Secara umum ada dua jenis tipe data yang dapat dihasilkan yaitu array dan object. Dari segi jumlah kita hanya dapat mengambil satu record (per record) atau keseluruhan record. • Mengembalikan Hasil Query Sebagai Kumpulan Array. Sebelum mengambil hasil query, ada baiknya mengecek terlebih dahulu apakah query tersebut memiliki hasil atau tidak. Perhatikan fungsi *result-> result_array()*, itu adalah method yang digunakan untuk mengambil hasil query. Sedangkan fungsi *result-> num_rows()* digunakan untuk mengetahui berapa jumlah record yang didapat. *result = this-> db-> query('SELECT * FROM users');* *if (result-> num_rows() > 0) foreach (result-> result_array() as row) echo row['username']; echo row['email'];* • Mengembalikan Hasil Query Sebagai Kumpulan Object. Fungsi yang digunakan hampir sama dengan mengembalikan nilai query sebagai array. Cukup dengan memanggil method *result-> result()*.

*result = this->db->query('SELECT * FROM users');* *if (result-> num_rows() > 0) foreach (result-> result() as row) echo row-> username; echo row-> email; // Mengembalikan Hasil Query Sebagai Array*
*db-> query('SELECT * FROM users');* *if (query-> num_rows() > 0) row = query-> row(); echo row['username']; echo row['email']; // Mengembalikan Hasil Query Sebagai Row Object jadi kita hanya*
*db-> query('SELECT * FROM users');* *if (query-> num_rows() > 0) row = query-> row(); echo row-> username; echo row-> email;*

Menggunakan Active Record

Active Record (AR) adalah sebuah pattern / pendekatan untuk membaca data dari sebuah table atau view dengan cara membungkusnya dalam sebuah kelas. Sehingga tidak dibutuhkan SQL jika kita menggunakan active record. Keuntungan yang didapat adalah kesederhanaan dalam pengkodean program dan fleksibilitasnya ketika terjadi pergantian skema atau pun jenis database

(bebas dari sintaks-sintaks SQL yang bersifat khusus ke database tertentu). CodeIgniter tidak sepenuhnya meniru pattern Active Record, tetapi ia melakukan sedikit modifikasi terhadap pattern tersebut. Dengan menggunakan Active Record CI maka proses pengambilan data, insert, update dan delete menjadi lebih sederhana. *this->db->get('users')*

Query di atas akan menghasilkan query “SELECT * FROM users” dan setara dengan

```
this->db->query('SELECT * FROM users');
```

Ketika kita tidak menggunakan fungsi apapun maka fungsi get akan menghasilkan seluruh record yang berada di dalam tabel tersebut, tetapi jika diberi fungsi select(), where(), dan lain-lain maka hasilnya akan berbeda.

Selecting Data

Untuk mengambil data dari database dengan field-field tertentu kita dapat menggunakan perintah select(). Contoh: *this->db->select('username,password,email');*
this->db->get('users');

Query di atas setara dengan *this->db->query('SELECT username,password,email FROM users');*

Selain method select, terdapat juga method-method lainnya seperti from(), where(), where_in(), group_by, order_by, like, distinct() dan lain-lain. Semua method tersebut jika dipanggil akan menghasilkan query yang berbeda. Contoh: *this->db->select('sum(budget) as sum_budget, site_id mapping.*')*

```
->where('site_id mapping.site_id','site_id')->where('ad_status','approved')->where('request_status','active')
```

```
->where('ad_operational_status','active')
```

```
->join('ads','ads.id_ad = site_id mapping.ad_id')->join('advertisers','id_advertiser = advertiser_id')->join('site_counter','site_counter.site_id = site_id')->join('sites','site_counter.site_id = sites.id_site')->join('publishers','sites.publisher_id = id_publisher')
```

```
->from('site_id mapping')
```

```
->order_by('ad_priority_level','desc')->order_by('point','desc')
```

```
->group_by('ad_id')->limit(0,10)->get()->result();
```

Insert Data

Selain pengambilan data, Active record juga menyediakan fungsionalitas untuk insert data. Untuk menginsert sebuah data kita hanya memerlukan dua parameter yaitu nama table dan data yang akan dimasukkan. Data yang akan dimasukkan harus memiliki format berupa array asosiatif ataupun object. Adapun keuntungan menggunakan active record untuk menginsert data selain sederhana untuk digunakan, adalah method ini secara build-in disertai fungsi untuk penanganan escaping, jadi query kita sudah bisa dikatakan aman dari kesalahan, contoh: *data = array('username' => 'Ibnoe','email' => 'xibnoe@gmail.com');*
this->db->insert('users',data);

Selain dengan cara di atas, kita juga bisa memasukkan data yang diinginkan satu persatu. Contohnya: *data = array('username' => 'Ibnoe','email' => 'xibnoe@gmail.com');*
this->db->set(data);this->db->insert('users');

Atau

```
this->db->set('username','ibnoe');  
this->db->set('email','xibnoe@gmail.com');  
this->db->insert('users');
```

Kedua Contoh di atas menghasilkan query yang sama.

Update Data

Selain insert dan select data, CodeIgniter juga menyediakan fungsi untuk update. Adapun penggunaannya mirip seperti penggunaan insert, contoh:

```
this->db->set('username','ibnoe');this->db->set('password','123456');  
this->db->update('users');
```

Query di atas akan menghasilkan query “update users set username=ibnoe, password=123456”. Cara yang lebih sederhana dengan menggunakan array asosiatif *data = array('username' => 'ibnoe','password' => '123456');**this->db->where('id',5); this->db->update('users',data);*

Delete Data

Untuk melakukan delete pada CodeIgniter, kita dapat menggunakan perintah berikut ini:

```
this->db->where('id',5);this->db->delete('table_name');
```

Chapter 8 Kasus 2. CRUD dan Pagination Database Setelah mengetahui cara menggunakan database maka kita akan mencoba melakukan perintah dasar pada sistem informasi CRUD (create, read, update dan delete) data menggunakan CodeIgniter. Selain CRUD, pada contoh ini juga akan dilakukan proses sorting dan pagination. Studi kasus yang akan diangkat adalah form data siswa. Pada bab ini aplikasi CRUD yang akan dibuat adalah aplikasi yang memiliki fungsi untuk menampilkan seluruh data siswa. Data yang ditampilkan memiliki pagination (pembagian jumlah record yang ditampilkan pada sebuah halaman website). Pagination dibutuhkan karena kita tidak mungkin menampilkan semua data dalam satu layar. Selain itu daftar siswa tadi akan memiliki fitur pengurutan berdasarkan field dan memiliki menu action per daftar siswa. Untuk melakukan pagination di Codeigniter telah disediakan sebuah kelas yaitu kelas pagination. Untuk menggunakan kelas ini minimal harus memiliki tiga data yang selanjutnya akan menjadi konfigurasi pada library ini. Ketiga data tersebut adalah pagination *base_url, total_row(jumlah total baris data record yang kita punya) dan per_page(jumlah baris yang akan ditampilkan per halaman)*.
\$load->library('pagination');
\$this->load->helper('url');
\$config['base_url'] = site_url('results/page/');
\$config['total_rows'] = 200;
\$config['per_page'] = 20;
\$pagination->initialize(\$config);
\$echothis->pagination->create_links();

Contoh diatas menunjukkan cara penggunaan library pagination. Output dari library ini adalah link- link yang terdiri atas halaman dan link ke halaman berikut (next) atau sebelumnya (prev). *Base_url harus berisi sebuah fungsi dari controller. Fungsi tersebut akan menerima parameter di atas masi banyak parameter lainnya yang dapat digunakan untuk mengkonfigurasi library pagination*
3; //mengganti tulisan next menjadi Berikutnya > \$config['next_link'] = 'Berikutnya'; //mengganti tulisan prev menjadi Sebelumnya < \$config['prev_link'] = 'Sebelumnya';

Adapun langkah-langkah yang harus dilakukan untuk membuat CRUD dan pagination adalah

1. Membuat Database Dan Table Data Siswa

Untuk memulai, Anda harus memiliki sebuah database. Selain database, Anda harus membuat sebuah tabel siswa. Adapun table yang harus di buat adalah CREATE TABLE IF NOT EXISTS 'siswa' ('id' bigint(20) NOT NULL AUTO_INCREMENT, 'nama' varchar(50) DEFAULT NULL, 'alamat' varchar(200) NOT NULL, 'jenis_kelamin' enum('laki-laki','perempuan') NOT NULL, 'tanggal_lahir' date NOT NULL, PRIMARY KEY (id)) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT = 1;

Tabel siswa di atas memiliki lima field yaitu id siswa, nama siswa, alamat siswa, jenis kelamin, dan tanggal lahir. Tabel ini digunakan untuk menyimpan

data siswa. 2. Konfigurasi CodeIgniter

Sebelum memulai pengkodean, sebaiknya kita melakukan konfigurasi terhadap tools yang akan digunakan. Ada tiga konfigurasi yang harus disesuaikan, antara lain: config.php, database.php dan router.php. Adapun yang harus dilakukan adalah: • config.php - pada file konfigurasi ini yang perlu diubah adalah $base_url$. Konfigurasi ini berfungsi untuk menentukan path dasar dari aplikasi. Konfigurasi ini akan digunakan

```
config['base_url'] = 'http://localhost/crud'; // ganti sesuai path di server Anda
// database.php
$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'crud';
$db['default']['dbdriver'] = 'mysql';
// router.php
// beri konfigurasi mengenai default controller dan routing rule.
// Pada konfigurasi ini yang diubah hanya default controller
"siswa";
```

3. Membuat Model Siswa

Model siswa ini adalah model yang bertugas dan bertanggung jawab untuk melakukan proses-proses yang berhubungan dengan database. Misalnya menyimpan, mengubah, menghapus serta mengambil data dari database. Biasanya untuk memudahkan kita, pembuatan model mengikuti desain database atau dapat diartikan satu tabel dapat diwakili oleh satu model. Model tersebutlah yang bertanggung jawab pada semua operasi pada tabel tersebut. Pada aplikasi CRUD ini kita akan membuat sebuah model siswa. Adapun isi dari model siswa adalah 1. `index.php` 2. 3. `class Siswa_m extends CI_Model` 4. 5. `private $primary_key = 'id'` 6. 7. `private $table_name = 'siswa'` 7. 8. `function construct()` 9. `parent::construct()` 10. 11. 12. `function get_paged($limit, $offset)` 13. `order_column = ''`, `order_type = 'asc'` 14. 15. `if (empty($order_column) || empty($order_type))` 16. `$this->db->order_by($this->primary_key, 'asc')` 17. `else` 18. `$this->db->order_by($order_column, $order_type)` 19. `return $this->db->get($this->table_name, $limit, $offset)` 20. 21. 22. `function count_all()` 23. `return $this->db->count_all($this->table_name)` 24. 25. 26. `function get_by_id($id)` 27. `$this->db->where($this->primary_key, $id)` 28. `return $this->db->get($this->table_name)` 29. 30. 31. `function save($person)` 32. `$this->db->insert($this->table_name, $person)` 33. `return $this->db->insert_id()` 34. 35. 36. `function update($id, $person)` 37. `$this->db->where($this->primary_key, $id)` 38. `$this->db->update($this->table_name, $person)` 39. 40.

```
41. function delete($id)
42. $this->db->where($this->primary_key, $id)
43. $this->db->delete($this->table_name)
44. 45.
```

4. Membuat Controller Dan View Data Siswa

Controller berisi logika untuk melakukan proses-proses yang ada. Adapun proses yang akan ditangani oleh controller ini adalah menampilkan daftar siswa, menambah siswa, mengubah data siswa, menghapus data siswa dan melihat detail siswa.

```
1. index.php 2. class Siswa extends CI_Controller
3. 4. private $limit = 10;
5. function construct()
6. 7. parent::construct()
8. 9. load library dan helper yang dibutuhkan
10. $this->load->library(array('table', 'form_validation'));
11. $this->load->helper(array('form', 'url'));
12. $this->load->model('siswa_model', TRUE);
13. 14. 15. function index()
16. 17. 18. if (empty($offset)) $offset = 0;
19. if (empty($order_column)) $order_column = 'id';
20. if (empty($order_type)) $order_type = 'asc';
21. // TODO : check for valid column
22. 23. // load data siswa
24. $siswa_model->get_paged($this->limit, $this->offset, $order_column, $order_type)
25. $result = $siswa_model->result();
26. 27. // generate pagination
28. $this->load->library('pagination');
29. config['base_url'] = site_url('siswa/index/');
30. config['total_rows'] = $siswa_model->count_all();
31. config['per_page'] = $this->limit;
32. config['uri_segment'] = 3;
33. $this->pagination->initialize(config);
34. data['pagination'] = $this->pagination->render();
```

```

pagination->create_links();35.36.//generatetabledata37.load->library('table');38.table->
set_empty("nbsp;");39.40.this->table->set_heading(41.'No',42.anchor('siswa/index/'.offset.'/nama/'.new_order
44. anchor('siswa/index/'.offset.'/jenis_kelamin/'.new_order,45.'JenisKelamin'),46.anchor('siswa/index/
mm-yyyy)'),48.'Actions'49.);50.i=0+offset;51.foreach(siswas as siswa)52.this-
->table->add_row(++i,53. siswa->nama,54.siswa->alamat,55. strtoupper(siswa->
jenis_kelamin) == 'M'?56.57.date('d-m-Y',strtotime(58.siswa->tanggal_lahir)),59.anchor('siswa/view/'.sis
->id,60. 'view',array('class'=>'view')).' '.61. anchor('siswa/update/'.siswa->
id,62.'update',array('class'=>'update')).'".63.anchor('siswa/delete/'.siswa-
->id,64. 'delete',array('class'=>'delete',65. 'onclick'=>'return confirm(66.
'Apakah Anda yakin ingin menghapus67. data siswa?'))68. );69.70.
data['table']=this->table->generate();71.72. if (this->uri->segment(3) == '
delete_success')73.data['message']='Data berhasil dihapus';74. else if (this->
uri->segment(3) == 'add_success')75.data['message']='Data berhasil ditam-
bah';76. else77. data['message']='';78.//loadview79.this->load->view('siswaList',data);80.81.82.83.function
siswa_baru';86. data['action'] = site_url('siswa/add');87.data['link_back'] =
anchor('siswa/index/',88.'Backto list of siswas',array('class'=>'back'));89.90.this-
->set_rules();91.92.//runvalidation93.if (this->form_validation->run() ==
FALSE)94.95.96.data['title']='Add new siswa';97.98.99.100. data['siswa']['alamat']='
';
101.102.103. anchor('siswa/index/',104. 'Lihat Daftar Siswa',array('class'=>'back'));
105.106. this->load->view('siswaEdit',data);107.108. else109. //
save data110. ma=>this->input->post('nama'),111.input->post('alamat'),112.'jenis_kelamin'=>this-
->input->post('jenis_kelamin'),113.date('Y-m-d',114.strtotime(this->input-
->post('tanggal_lahir'))));115.id=this->siswa_model->save(siswa);116.117.
// set form input nama="id"118. this->validation->id=id;119.120.
redirect('siswa/index/add_success');121.122.123.124.functionview(id)125. //
set common properties126. data['title'] = 'siswaDetails';127.data['link_back'] =
anchor('siswa/index/',128.'Lihatdaftar siswas',array('class'=>'back'));129.130.
131.132.133.
// load view
siswa_model->get_by_id(id)->row();
134. this->load->view('siswaView',data);135.136.137. function
update(id)138.//setcommonproperties139.data['title']='Update siswa';140. this->
load->library('form_validation');141.//setvalidationproperties142.this->set_rules();143.data['action']=('sis
->form_validation->run() == FALSE)147.148.data['message']='';149. data['siswa']=this-
->siswa_model->get_by_id(id)->row_array();150.POST['jenis_kelamin'] = 151.strtoupper(data['siswa']['jenis_kelam
date('d-m-Y',153.strtotime(data['siswa']['tanggal_lahir']));154.155.//setcommonproperties156.data['title']='
siswa';157. data['message']='';
158.159. else160. // save data161. id =this->input->post('id');162.
siswa = array('nama'=>this->input->post('nama'),163. 'alamat'=>this->
input->post('alamat'),164.'jenis_kelamin'=>this->input->post('jenis_kelamin'),165.'tanggal_lahir'=>
date('Y-m-d',166.strtotime(this->input->post('tanggal_lahir'))));167.this->siswa_model->
update(id,siswa);168.data['siswa']=this->siswa_model->get_by_id(id)->row_array();169.//setusermessage1
siswa success';171.172. data['link_back'] = anchor('siswa/index/',173.'Lihatdaftar siswa',array('class'=>
back'));174.//loadview175.this->load->view('siswaEdit',data);176.177.178.functiondelete(id)
179. // delete siswa180. this->siswa_model->delete(id);181. // redirect

```



```

"alamat" class = "text" value = " <?php echo form_error('alamat');? >" />
35. <td align="center">?php echo form_error('alamat');? > </td> 36. </tr> 37. <tr>
38. <td align="top"> jenis_kelamin <span style="color: red;"> * <
/td> </td> 39. <td align="center"> input type="radio" name="jenis_kelamin" value =
"M" <?php echo form_radio('jenis_kelamin', 'M', TRUE);? > /> Laki-Laki 41. <td align="center">
input type="radio" name="jenis_kelamin" value="F" <?php echo form_radio('jenis_kelamin', 'F');? >
/ > Perempuan 43. <td align="center"> form_error('jenis_kelamin');? > </td> 44. <tr>
45. <td align="center"> Date of Birth (dd - mm - yyyy) <span style="color: red;"> * </span> </td> 48. <td align="center"> input type =
"text" name = "tanggal_lahir" class = "text" 49. value = " <?php echo form_error('tanggal_lahir');? >" 50. </td>
" />
51. <td align="center"> form_error('tanggal_lahir');? > </td> 52. </tr> 53. <tr>
54. <td align="center"> <input type="submit" value = "Save" /> </td> 56. </tr> 57. </table> 58. </div> 59. </form>
60. <br /> 61. <?php echo form_error('tanggal_lahir');? > 62. </div> 63. </body> 64. </html>

```

View di atas juga digunakan oleh fungsi edit data siswa. Untuk melakukan penyimpanan maka perlu dipanggil fungsi *this->siswa_model->update(id, siswa);* pada fungsi tersebut perlu

Selain tambah dan ubah, ada juga fungsi hapus dan melihat detail siswa. Untuk menghapus siswa hanya dibutuhkan id siswa yang ingin dihapus. Sedangkan untuk melihat detail data siswa hal yang perlu dilakukan adalah memanggil data siswa berdasarkan id siswa kemudian ditampilkan ke dalam view berikut ini: 1. `<?php echo form_error('tanggal_lahir');? >` 2. `<?php echo form_error('tanggal_lahir');? >` 3. `<?php echo form_error('tanggal_lahir');? >` 4. `<?php echo form_error('tanggal_lahir');? >` 5. `<?php echo form_error('tanggal_lahir');? >` 6. `<?php echo form_error('tanggal_lahir');? >` 7. `<?php echo form_error('tanggal_lahir');? >` 8. `<?php echo form_error('tanggal_lahir');? >` 9. `<?php echo form_error('tanggal_lahir');? >` 10. `<?php echo form_error('tanggal_lahir');? >` 11. `<?php echo form_error('tanggal_lahir');? >` 12. `<?php echo form_error('tanggal_lahir');? >` 13. `<?php echo form_error('tanggal_lahir');? >` 14. `<?php echo form_error('tanggal_lahir');? >` 15. `<?php echo form_error('tanggal_lahir');? >` 16. `<?php echo form_error('tanggal_lahir');? >` 17. `<?php echo form_error('tanggal_lahir');? >` 18. `<?php echo form_error('tanggal_lahir');? >` 19. `<?php echo form_error('tanggal_lahir');? >` 20. `<?php echo form_error('tanggal_lahir');? >` 21. `<?php echo form_error('tanggal_lahir');? >` 22. `<?php echo form_error('tanggal_lahir');? >` 23. `<?php echo form_error('tanggal_lahir');? >` 24. `<?php echo form_error('tanggal_lahir');? >` 25. `<?php echo form_error('tanggal_lahir');? >` 26. `<?php echo form_error('tanggal_lahir');? >` 27. `<?php echo form_error('tanggal_lahir');? >` 28. `<?php echo form_error('tanggal_lahir');? >` 29. `<?php echo form_error('tanggal_lahir');? >` 30. `<?php echo form_error('tanggal_lahir');? >` 31. `<?php echo form_error('tanggal_lahir');? >` 32. `<?php echo form_error('tanggal_lahir');? >` 33. `<?php echo form_error('tanggal_lahir');? >` 34. `<?php echo form_error('tanggal_lahir');? >` 35. `<?php echo form_error('tanggal_lahir');? >` 36. `<?php echo form_error('tanggal_lahir');? >` 37. `<?php echo form_error('tanggal_lahir');? >` 38. `<?php echo form_error('tanggal_lahir');? >` 39. `<?php echo form_error('tanggal_lahir');? >` 40. `<?php echo form_error('tanggal_lahir');? >` 41. `<?php echo form_error('tanggal_lahir');? >` 42. `<?php echo form_error('tanggal_lahir');? >` 43. `<?php echo form_error('tanggal_lahir');? >` 44. `<?php echo form_error('tanggal_lahir');? >` 45. `<?php echo form_error('tanggal_lahir');? >` 46. `<?php echo form_error('tanggal_lahir');? >` 47. `<?php echo form_error('tanggal_lahir');? >` 48. `<?php echo form_error('tanggal_lahir');? >` 49. `<?php echo form_error('tanggal_lahir');? >` 50. `<?php echo form_error('tanggal_lahir');? >` 51. `<?php echo form_error('tanggal_lahir');? >` 52. `<?php echo form_error('tanggal_lahir');? >` 53. `<?php echo form_error('tanggal_lahir');? >` 54. `<?php echo form_error('tanggal_lahir');? >` 55. `<?php echo form_error('tanggal_lahir');? >` 56. `<?php echo form_error('tanggal_lahir');? >` 57. `<?php echo form_error('tanggal_lahir');? >` 58. `<?php echo form_error('tanggal_lahir');? >` 59. `<?php echo form_error('tanggal_lahir');? >` 60. `<?php echo form_error('tanggal_lahir');? >` 61. `<?php echo form_error('tanggal_lahir');? >` 62. `<?php echo form_error('tanggal_lahir');? >` 63. `<?php echo form_error('tanggal_lahir');? >` 64. `<?php echo form_error('tanggal_lahir');? >` 65. `<?php echo form_error('tanggal_lahir');? >` 66. `<?php echo form_error('tanggal_lahir');? >` 67. `<?php echo form_error('tanggal_lahir');? >` 68. `<?php echo form_error('tanggal_lahir');? >` 69. `<?php echo form_error('tanggal_lahir');? >` 70. `<?php echo form_error('tanggal_lahir');? >` 71. `<?php echo form_error('tanggal_lahir');? >` 72. `<?php echo form_error('tanggal_lahir');? >` 73. `<?php echo form_error('tanggal_lahir');? >` 74. `<?php echo form_error('tanggal_lahir');? >` 75. `<?php echo form_error('tanggal_lahir');? >` 76. `<?php echo form_error('tanggal_lahir');? >` 77. `<?php echo form_error('tanggal_lahir');? >` 78. `<?php echo form_error('tanggal_lahir');? >` 79. `<?php echo form_error('tanggal_lahir');? >` 80. `<?php echo form_error('tanggal_lahir');? >` 81. `<?php echo form_error('tanggal_lahir');? >` 82. `<?php echo form_error('tanggal_lahir');? >` 83. `<?php echo form_error('tanggal_lahir');? >` 84. `<?php echo form_error('tanggal_lahir');? >` 85. `<?php echo form_error('tanggal_lahir');? >` 86. `<?php echo form_error('tanggal_lahir');? >` 87. `<?php echo form_error('tanggal_lahir');? >` 88. `<?php echo form_error('tanggal_lahir');? >` 89. `<?php echo form_error('tanggal_lahir');? >` 90. `<?php echo form_error('tanggal_lahir');? >` 91. `<?php echo form_error('tanggal_lahir');? >` 92. `<?php echo form_error('tanggal_lahir');? >` 93. `<?php echo form_error('tanggal_lahir');? >` 94. `<?php echo form_error('tanggal_lahir');? >` 95. `<?php echo form_error('tanggal_lahir');? >` 96. `<?php echo form_error('tanggal_lahir');? >` 97. `<?php echo form_error('tanggal_lahir');? >` 98. `<?php echo form_error('tanggal_lahir');? >` 99. `<?php echo form_error('tanggal_lahir');? >` 100. `<?php echo form_error('tanggal_lahir');? >`

Sebelum melakukan delete perlu dilakukan konfirmasi untuk mencegah kesalahan user dalam menekan link action.

Chapter 10 Kasus 3. Sistem Templat Sistem templat adalah salah satu fitur atau library yang harus dipikirkan di dalam CodeIgniter, karena CodeIgniter memiliki metode yang sangat sederhana. Tidak memiliki sistem layout, widget dan lain-lain. Untuk membuat sistem templat di CodeIgniter

kita dapat membuat sendiri dengan menggunakan fungsi-fungsi yang telah disediakan oleh CodeIgniter atau kita juga dapat menggunakan sistem templating yang sudah ada dan populer. Kegunaan sistem templating adalah : • Kerja Sama Tim Yang Lebih Baik - Dengan pemisahan menggunakan templating ini, maka rekan bagian pengerjaan desain tidak lagi terganggu oleh kode program yang Anda buat. Begitupun Anda, tidak khawatir lagi script/kode program yang telah dibuat akan terganggu. • Skrip/code Yang Bersih - Situs dengan aplikasi kompleks, misalnya sebuah portal, pasti memerlukan skrip yang kompleks yang menghabiskan sampai ribuan baris. Tentunya akan sangat mengganggu sekali kalau skrip yang sudah memusingkan itu ditambah lagi dengan tag-tag HTML di dalamnya. • Perubahan Tampilan Lebih Cepat Dan Mudah - Dengan pemisahan melalui template, hal tersebut dapat dilakukan dengan mudah, bahkan tanpa harus merombak skrip PHP sedikit pun.

Native CodeIgniter Templating

Dengan menyusun view-view yang ada, kita sebenarnya dapat membuat sebuah template library yang cukup powerful. Idennya sederhana saja, cukup membagi sebuah halaman menjadi beberapa area. Sebagai contoh kita akan menggunakan template dari opendesign.org. Dari desain tersebut dapat kita bagi menjadi area header, top menu, right menu dan content. Masing-masing area tersebut akan digabungkan dalam satu halaman utuh yang dinamakan template. Masing-masing area tersebut akan ditangani oleh sebuah view agar tidak terjadi duplikasi dan akan mempermudah penggunaan kembali area tersebut. Adapun area-area tersebut dapat dilihat pada gambar berikut.

Template yang akan dibuat terbagi menjadi empat bagian yaitu Header (berfungsi sebagai tempat logo dan slogan aplikasi), Top menu (bagian menu utama disebelah atas), Right Menu (menu navigasi tambahan disebelah kanan) dan sebuah Content. Area content ini lah yang seringkali berubah pada setiap page. Untuk mengimplementasikan sistem templating tersebut sebaiknya kita buat sebuah library template. Adapun langkah-langkah yang harus dilakukan adalah: 1. Membuat Library Template

Fungsi library tersebut hanya mengatur view mana yang akan dipanggil dan meletakkannya di dalam template. Kenapa membuatnya dalam sebuah library? Karena dengan cara ini memberikan fleksibilitas terhadap sistem template. Cara pembuatan library ini diawali dengan membuat sebuah file bernama template.php pada folder application/libraries. Adapun isi file template.php adalah: 1. `if(!defined('BASEPATH')){exit('No direct script access allowed');}` 2. `class Template` 3. `protected $ci;` 4. `function __construct()` 5. `6. $this->ci = get_instance();` 7. `8. 9. function display($template, $data = null)` 10. `11. $data['_content'] = $this->ci->load->view(12. $template, $data, true);` 13. `$data['_header'] = $this->ci->load->view(14. 'template/header', $data, true);`

15. `$data['_top_menu'] = $this->ci->load->view(16. 'template/menu', $data, true);` 17. `$data['_right_menu'] = $this->ci->load->view(18. 'template/sidebar', $data, true);` 19. `$this->ci->load->view('template.php', $data);` 20. 21.

Perhatikan fungsi display, disana ada dua parameter yaitu template dan data. Parameter data berfungsi sebagai data yang akan dikirimkan ke controller. Sedangkan template adalah view yang akan dipanggil untuk ditampilkan sebagai content utama. Pada template ini akan dibagi menjadi empat area yaitu

content, header, top menu, dan right menu. Masing-masing area diisi oleh sebuah view (perhatikan baris 11,13,15,17). Masing-masing view tadi di-load dan disimpan dalam memori. Lalu digabungkan kedalam sebuah template (baris 19). Fungsi yang ada pada library ini sebenarnya dapat kita tambahkan lagi, misalnya untuk keperluan seo, kita ingin memanipulasi title dan meta tag dll. Library diatas merupakan contoh paling sederhana ketika kita ingin membuat sebuah sistem template sendiri.

2. Membuat Template Layout View
Sebuah template layout view adalah sebuah view yang akan menggabungkan masing-masing view menjadi suatu bentuk kesatuan. Adapun isi dari template layout adalah: Application/view/template.php

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" 2.
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> 3. <html xmlns="http://www.w3.org/1999/xhtml"
4. <head> 5. <meta http-equiv="content-type" content="text/html; charset=utf-
8"> 6. <link rel="stylesheet" type="text/css" href="http://www.w3.org/1999/xhtml"
public/style.css"/> 7. <title> SistemTemplateCodeIgniterTemplate <
/title> 8. </head> 9. <body> 10. <div id="wrap"> 11. <div id="
header"> 12. <!-- 13.AreaHeader 14. --> 15. <?php echo header();?>
16. </div> 17. <div id="menu"> 18. <!-- 19.AreaTopMenu 20. -->
21. <?php echo top_menu();?> 22. </div> 23. <div id="contentwrap">
24. <div id="content"> 25. <!-- 26. Area content 27. --> 28. <?php echo
content();?> 29.30. </div> 31.32. <div id="sidebar"> 33. <!-- 34.AreaRightMenu 35.-->
36. <?php echo right_menu();?> 37. </div> 38. <div style="clear :
both;"> 39. </div> 40.41. <div id="footer"> 42. <p>
Copyrightcopy; <a href="http://www.readcrazyreviews.com"> 43. <a href="
http://www.readcrazyreviews.com"> 44.ReadCrazyReviews </a> <
/p> 45. </div> 46.47. </div> 48.49. </body> 50. </html>
```

File diatas merupakan penggabungan dari semua bagian yang telah dibahas. Perhatikan baris 15, 21, 28, dan 36. Keempat variabel tersebut akan diisi view dari masing-masing bagian. Berikut ini adalah view-view yang menjadi part/area dari template

a. View header

Application/view/template/header.php 1. <h1>Red Wall Tem-
plate 2. <h2>Your web site slogan goes here 3.

b. View Top menu

Application/view/template/menu.php 1. 2. Home 3. Articles 4. About Us 5. Resources 6. Contact Us 7.

c. View sidebar menu

Application/view/template/sidebar.php 1. <h3>Main Content 2. 3. Conflantur 4. Externarum 5. Essem paulo 6. Aeque fecto 7. Quo locis 8. Related Web Sites 9. Plus 10. Agi praecise 11. Infinitum 12. Corporea ac perpauca 13. Aeque fecto 14. Perpauca 15. Agi praecise 16. Corporea ac perpauca 17.

Setelah memiliki view ketiga area tersebut (header, top menu dan sidebar) maka kita siap untuk menggunakan sistem template ini. Selanjutnya adalah membuat sebuah controller. Agar lebih mudah pembuatan controller-nya maka gunakan saja controller default dari CodeIgniter, yaitu dengan sedikit melakukan perubahan kode program pada controller-nya. `Application/controllers/welcome.php`

```
1. if (! defined('BASEPATH')) exit('No direct script access allowed'); 2.
3. class Welcome extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('template');
        $this->load->helper('url');
        $this->functionindex();
        $this->template->display('welcome_message');
        $this->template->display('view_parameter', 20, array('judul' => 'judulView'));
        $this->template->display('view_parameter', 20, array('judul' => 'judulView'));
    }
}
26. Location : ./application/controllers/welcome.php
/
```

Perhatikan pada baris 8, di situ kita me-load library yang telah dibuat tadi. Pada baris 14 dan 19 kita menggunakan fungsi display untuk menampilkan template. Kita harus menyiapkan sebuah view yang bernama `welcome_message.View` tersebut akan diletakkan pada

```
1. <h1>Welcome to CodeIgniter!</h1> 2. 3. <p>The page you are looking at is being generated dynamically by CodeIgniter.</p> 4. 5. <p>If you would like to edit this page you'll find it located at:</p> 6. <code>application/views/welcome_message.php </code> 7.8. <p>The corresponding controller for this page is found at:</p> 9. <code>application/controllers/welcome.php </code> 10.11. <p>If you are exploring CodeIgniter for the very first time, you should start by reading the <a href = "user_guide/">User Guide</a>.</p> 14.15.16. <p><br />The page rendered in elapsed time seconds</p>
```

Adapun tampilannya adalah seperti berikut ini:

Chapter 10 Kasus 4. Sistem Authentikasi Sistem autentikasi atau sistem login merupakan salah satu bagian dari aplikasi yang sering kita kerjakan. Sistem ini juga menjadi bagian yang vital pada aplikasi. Bagian ini lah yang menjamin keamanan data dari aplikasi yang sedang dikerjakan. Pada kasus ini library yang paling penting untuk di ketahui adalah library session. Perlu diingat bahwa library session codeigniter disimpan di sebuah cookie. Cookie tersebut dapat kita enkripsi. Selain itu kita juga dapat menyimpan session tersebut di database. Yang artinya user cookie harus cocok dengan cookie yang ada di database. Secara default hanya cookie yang digunakan dan walaupun anda tidak menggunakan enkripsi cookie anda harus tetap mengkonfigurasi enkripti key. Untuk menggunakan library session sama seperti penggunaan library pada biasanya. Kita bisa mengkonfigurasi file autoload atau memanggilnya secara manual `$this->load->library('session');`

Ketika library sudah di load kelas session akan mengecek apakah data session yang diinginkan berada di cookie. Jika data tidak ada di cookie maka akan dibuatkan sebuah session baru dan disimpan didalam cookie. Jika data cookie ditemukan maka data tersebut akan diupdate terutama untuk `last_activity` dan `session_id`. Untuk penggunaan library session `$session->userdata('item');`

Sedangkan untuk menyimpan session dapat dilakukan dengan cara

```
newdata = array(
);
'username' => 'ibnoe', 'email' => 'ibnoe@gmail.com', 'logged_in' => TRUE
```

```
this -> session -> set_userdata(newdata);
```

Catatan: Data session codeigniter secara default disimpan dalam cookie. Cookie memiliki batasan sebesar 4Kb data. Dengan menggunakan enkripsi maka data yang disimpan akan menjadi lebih panjang. Jadi harap berhati-hati ada kemungkinan data tidak tersimpan. Anda bisa menggunakan alternatif database atau mengextend session ke native session

Untuk membuat sebuah sistem autentikasi maka hal pertama yang perlu dilakukan adalah

1. Membuat Desain Aplikasi Desain aplikasi sangat berpengaruh pada code yang akan kita buat. Karena sistem autentikasi ini merupakan salah satu bagian yang kritikal maka penulis akan mencoba untuk membuatnya aman bukan hanya dari segi kode, tetapi juga design. Perhatikan struktur website dibawah ini.

Pada gambar diatas ada dua tipe page yaitu page yang bisa diakses oleh semua orang dan page yang hanya boleh diakses oleh member. Hal tersebut sederhana jika kita hanya handle dua page tetapi jika pagennya ada banyak maka kita akan mengecek satu-satu informasi user. Hal tersebut kurang aman karena kode kita terduplikasi keseluruh aplikasi. Kita akan mencoba untuk meng-extend controller membuat kasus ini menjadi lebih simple. Untuk controller untuk user yang login harus menggunakan *member_controller*.

Jika kita perhatikan class diagram diatas maka semua *member_controller* sudah memiliki fungsi untuk pengecekan

2. Membuat Tabel User Untuk sistem autentikasi kita akan membuat tiga tabel. Tabel pertama yaitu tabel user berisi data informasi login user, tabel *user_group* berisi data pengelompokan user menjadi *group*, dan tabel *tracker* berfungsi untuk menyimpan data info *MYISAM*;

3. Membuat Library Access Setelah tabel terbentuk barulah kita membuat library untuk mengakses dan mengverifikasi data user. Library tersebut kita sebut Access. Adapun kode programnya adalah sebagai berikut *application/libraries/access.php*

```
1. if (!defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Access
4.     5. public user;
6. 7. /**
8.  *Constructor
9.  */
10. function construct()
11. 12. {
13.     $CI = get_instance();
14.     $auth = this -> $CI -> config -> item('auth');
15.     $CI -> load -> helper('cookie');
16.     this -> $CI -> load -> model('users_model');
17. 18. $users_model = this -> $CI -> $users_model;
19. 20. 21. /**
22.  *Cek login user
23.  */
24. 25. function login(username,
26. 27. 28. password)
29. 26. 27. 28. result = this -> $users_model -> get_login_info(username);
29. 30. if (result) // Result Found
31.     32. password = md5(password);
33.     if (password === result ->
34.     35. // Start session
36.     this -> $CI -> session ->
37.     set_userdata('user_id', result -> $user_id);
38.     return TRUE;
39. 40. 41. return FALSE;
42. 43. 44. 45. /**
46.  *cek apakah sudah login
47.  */
48. function is_login()
49. 50. {
51.     return (this -> $CI -> session ->
52.     userdata('user_id')) ? TRUE : FALSE;
53. 51. 52. 53. /**
54.  *Logout
55.  */
56. 57. function logout()
58. 59. {
60.     this -> $CI -> session -> unset_userdata('user_id');
61. 62.
```

Library access di atas membutuhkan library session dan database, jadi sebelum menggunakan library tersebut pastikan Anda telah setting konfigurasi CodeIgniter dengan benar. Perhatikan fungsi login, fungsi itu bertugas untuk melakukan pengecekan terhadap data username dan password yang

diberikan oleh user. Fungsi login akan mengambil semua informasi tentang user berdasarkan username (baris 28). Setelah data didapatkan maka dilakukan pencocokan username dan password (baris 33). Jika passwordnya cocok maka session user tersebut disimpan sebagai penanda bahwa user telah login. 4. Membuat Library Access Untuk mendapatkan data user tersebut maka kita membutuhkan sebuah model. Model ini sangat sederhana hanya terdiri satu fungsi yaitu `get_login_info`. Fungsi tersebut akan mengembalikan object data user apabila username yang dimasukkan ada di data

```
application/models/users_model.php
1. if (!defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Users_model extends CI_Model { public $table = 'users';
7. public $primary_key = 'user_id';
8. 9. function construct() { parent::construct();
12. 13. 14. function get_login_info($username) {
15. 16. $this->db->where('username', $username);
17. $this->db->limit(1);
18. $query = $this->db->get($this->table);
19. return ($query->num_rows() > 0) ? $query->row() : FALSE;
20. 21. 22.
```

5. Membuat Controller Member

Pertama-tama kita akan membuat controller untuk login. Controller tersebut kita beri nama member. Controller member ini mempunyai dua fungsi utama yaitu login dan logout. 1. `if (!defined('BASEPATH')) exit('No direct script access allowed');` 2. 3. `class Member extends CI_Controller {`
6. 7. `function construct() { parent::construct();`
8. `load->library('access');`
9. 10. 11. 12. `function index() {`
13. 14. 15. `$this->access->logout();`
16. `$this->login();`
17. 18. 19. 20. `function login() {`
21. 22. 23.

```
24. $this->load->library('form_validation');
25. $this->load->helper('form');
26. 27. $this->form_validation->set_rules('username', 'Username', 'trim|required|strip_tags');
28. $this->form_validation->set_rules('password', 'Password', 'trim|required');
29. $this->form_validation->set_rules('token', 'token', 'callback_check_login');
30. 30. 31. // $this->output->enable_profiler(1);
32. if ($this->form_validation->run() == FALSE) {
33. 34. 35. 36. $this->template->display('member/login');
37. 38. 39. 40. redirect('dashboard');
41. 42. 43. 44. function logout() {
45. 46. $this->access->logout();
47. redirect('member/login');
48. 49. 50. function check_login() {
51. 52. 53. $username = $this->input->post('username', TRUE);
54. $password = $this->input->post('password', TRUE);
55. 56. $login = $this->access->login($username, $password);
57. if ($login) {
58. 59. return TRUE;
60. 61. 62. 63. $this->form_validation->set_message('check_login', 'Username atau password anda salah.');
64. return FALSE;
65. 66.
```

Pada controller member kita akan melakukan pengecekan terhadap inputan yang telah di masukkan user. Oleh karena itu kita tetap menggunakan library validation. Kita akan menggunakan custom validation. Perhatikan baris 29 dan baris 50. Fungsi `check_login` akan

menjadi custom validator pada controller ini. Pada fungsi itu juga library access dipanggil untuk melakukan pengecekan login. Baris 63 digunakan untuk mengeset pesan error pada validasi. 6. Membuat view controller member

Setelah memiliki model dan controller maka kita tinggal membuat view

```
1. if ($this->form_validation->run() == FALSE) {
2. 3. < h1 > Sign On < /h1 >
4. 5. < p > Welcome. Please fill your username and password. < /p >
6. < ?php if (validation_errors()) {
7. < div class = "fail" >
8. < ?php echo validation_errors();
9. < /div >
10. 11. < form name = "login_form" method = "post" target = "< ?php site_url('member/login')? >
12. 13. < formname = "login_form" method = "post" target = "< ?php site_url('member/login')? >
```

```
"style = "margin : 0px;" > 14.15.16. < p >< label > Username < /label >
17. < inputname = "username" id = "username" value = " <?php echo @v-
username?>" class="input large" type="text" /> 18. < /p> 19. < inputlabel Password /label>
20. < input name="password" id="password" width="100px" type="password"
class="input large" /> 21. < /p> 22. < input input name="submit" type="submit"
value="Submit " class="button" /> 23. < /form>
```

Adapun tampilan dari view diatas adalah

7. Membuat Mengextend library Controller

Controller member hanya digunakan memverifikasi user yang login. Kita akan membuat sebuah controller yang hanya bisa diakses oleh user yang sudah login. Controller tersebut akan diberi nama `MemberController`. Untuk membuat `MemberController` kita harus melakukan beberapa langkah berikut ini:

```
?php if (!defined('BASEPATH')) exit('No direct script access allowed');
2.3.4. class MemberController extends CI_Controller {
    public function __construct() {
        parent::__construct();
    }
    public function login() {
        // ...
    }
}
```

Class `MemberController` merupakan turunan dari kelas `CI_Controller` dengan penambahan fungsi pengecekan apakah script dijalankan dari browser atau tidak.

```
1. <?php if (!defined('BASEPATH')) exit('No direct script access allowed');
2.
3. class Dashboard extends MemberController {
    public function __construct() {
        parent::__construct();
    }
    public function index() {
        // ...
    }
}
```

Chapter 11 Kasus 5. Image Gallery Sederhana Sebuah image gallery merupakan aplikasi yang menarik untuk dikerjakan. Dengan menggunakan php biasa, image gallery tidak mudah untuk dibuat. Terutama bagian upload dan meresize gambar yang telah di upload. Dengan codeigniter pembuatan gallery menjadi lebih mudah dan cepat. Dengan memanfaatkan library upload, resize serta beberapa helper, kita dapat membuat sebuah gallery yang menarik. Library upload codeigniter mudah untuk digunakan. Cara pemanggilannya juga sama dengan library lainnya `$this->load->library('upload');`

Untuk dapat menggunakan library ini kita harus mengkonfigurasi beberapa hal diantaranya dimana kita akan mengupload file tersebut, tipe dan ukuran file, dan lain-lain. Contohnya: `config['upload_path'] = APPPATH.'uploads/'; config['allowed_types'] = 'jpg|jpeg|gif|png'; config['max_size'] = 1024;`

```
$this->upload->initialize(config);
```

Setelah mensetting library upload kita tinggal memanggil fungsi `do_upload()` untuk mengupload file.

```
else
    //gagal mengupload file error menyimpan error message dalam //variabel
    $error = array('error' => $this->upload->display_errors());
    //sukses mengupload file informasi file disimpan dalam // variabel data
    $data = array('upload_data' => $this->upload->data());
```

Image gallery yang akan kita buat memiliki fitur untuk mengupload file gambar, lalu merisize gambar tersebut. Setelah semua proses berhasil maka gallery akan menampilkan thumbnail dari image tersebut. Jika thumbnail tersebut di klik maka barulah gambar yang sebenarnya muncul dalam bentuk popup.

Adapun langkah-langkah yang dilakukan untuk membuat image gallery adalah:

- Membuat folder `upload` di root aplikasi dan jangan lupa untuk mengubah permisidari file

Struktur direktori gallery

- Mengubah settingan Codeigniter. Settingan yang harus di ubah adalah `config['base_url']` di `application/config/config.php`, lalu sesuaikan dengan aplikasi. `function __construct() { parent::__construct(); } 2. class Gallery extends CI_Controller { 3. public function index() { 4. if (!defined('BASEPATH')) exit('No direct script access allowed'); 5. $this->load->model('gallery_model'); 6. $this->load->helper(array('form', 'url')); 7. $data['message'] = ''; 8. if (!$this->upload->do_upload()) { 9. if (isset($_POST['submit'])) { 10. $data['message'] = $this->upload->display_errors(); 11. } else { 12. $data['upload_data'] = $this->upload->data(); 13. $data['message'] = 'Anda telah sukses mengupload gambar!!!'; 14. $config['resize'] = array(15. 'source_image' => $data['upload_data']['full_path'], 16. 'new_image' => 'thumb/', 17. 'maintain_ratio' => true, 18. 'width' => 160, 19. 'height' => 120 20.); 21. $this->load->library('image_lib', $config['resize']); 22. if (!$this->image_lib->resize()) { 23. $data['message'] = $this->image_lib->display_errors(); 24. } 25. $data['images'] = $this->gallery_model->fetch_image(FPATH.'upload_image/'); 26. $this->template->display('gallery', $data); 27. } 28. } 29. } 30. } 31. } 32. } 33. } 34. } 35. } 36. } 37. } 38. } 39. } 40. } 41. } 42. } 43. } 44. } 45. } 46. } 47. } 48. } 49. } 50. } 51. } 52. } 53. } 54. } 55. } 56. } 57. } 58. } 59. } 60. } 61. } 62. } 63. } 64. } 65. } 66. } 67. } 68. } 69. } 70. } 71. } 72. } 73. } 74. } 75. } 76. } 77. } 78. } 79. } 80. } 81. } 82. } 83. } 84. } 85. } 86. } 87. } 88. } 89. } 90. } 91. } 92. } 93. } 94. } 95. } 96. } 97. } 98. } 99. } 100. }`

Perhatikan baris 18-21, disana kita meload semua library dan helper yang dibutuhkan. Khusus untuk library upload kita menggunakan konfigurasi pada saat pemanggilan library `$this->load->library('upload', $config)`; Perhatikan parameter kedua (variabel `$config`). Variable tersebut berisi settingan dimana file tersebut akan diupload, apa saja yang akan dilakukan. Pada baris 23, kita memanggil fungsi untuk mengupload jika gambar sukses diupload maka kita akan melakukan `44. $this->image_lib->resize($data['upload_data']['full_path'], $data['upload_data']['thumb'], $config['resize']);` Untuk meresize gambar kita juga perlu memberikan parameter khusus pada saat loading library `$this->load->library('image_lib', $config['resize']);` pada konfigurasi itu lah kita menentukan ukuran thumbnail yang akan diupload. `2. class Gallery extends CI_Model { 3. function __construct() { 4. parent::__construct(); 5. } 6. function fetch_image($path) { 7. $this->load->helper('file'); 8. return get_filenames($path); 9. } 10. } 11. }`

Perhatikan baris 10 dan 11 baris tersebut akan mengembalikan daftar nama file yang berada di sebuah folder • Membuat view gallery. View akan menampilkan dua komponen utama yaitu daftar gambar dan form yang akan digunakan untuk mengupload image. `1. <?php echo $message; 2. <div id="gallery"> 3. <?php $atts = array(4. 'width' => '800', 5. 'height' => '600', 6. 'scrollbars' => 'yes', 7. 'status' => 'yes', 8. 'resizable' => 'yes', 9. 'screenx' => '0', 10. 'screeny' => '0' 11.); 12. foreach($images as $image): 13. ?> 14. <div class="thumb"> 15. 16. <img src="<?php echo anchor_popup(base_url().$upload_image.'/'.$image, $image, $atts); 17. 18. </div> 19. </div> 20. </div> 21. </div> 22. </div> 23. </div> 24. </div> 25. </div> 26. </div> 27. </div> 28. </div> 29. </div> 30. </div> 31. </div> 32. </div> 33. </div> 34. </div> 35. </div> 36. </div> 37. </div> 38. </div> 39. </div> 40. </div> 41. </div> 42. </div> 43. </div> 44. </div> 45. </div> 46. </div> 47. </div> 48. </div> 49. </div> 50. </div> 51. </div> 52. </div> 53. </div> 54. </div> 55. </div> 56. </div> 57. </div> 58. </div> 59. </div> 60. </div> 61. </div> 62. </div> 63. </div> 64. </div> 65. </div> 66. </div> 67. </div> 68. </div> 69. </div> 70. </div> 71. </div> 72. </div> 73. </div> 74. </div> 75. </div> 76. </div> 77. </div> 78. </div> 79. </div> 80. </div> 81. </div> 82. </div> 83. </div> 84. </div> 85. </div> 86. </div> 87. </div> 88. </div> 89. </div> 90. </div> 91. </div> 92. </div> 93. </div> 94. </div> 95. </div> 96. </div> 97. </div> 98. </div> 99. </div> 100. </div></code`

Perhatikan baris ke empat. Disana kita men-setting `anchor_popup($link, $image, $atts)` yang akan digunakan untuk membuka popup. `34. <div class="thumb"> 35. 36. <img src="<?php echo anchor_popup(base_url().$upload_image.'/'.$image, $image, $atts); 37. 38. </div> 39. </div> 40. </div> 41. </div> 42. </div> 43. </div> 44. </div> 45. </div> 46. </div> 47. </div> 48. </div> 49. </div> 50. </div> 51. </div> 52. </div> 53. </div> 54. </div> 55. </div> 56. </div> 57. </div> 58. </div> 59. </div> 60. </div> 61. </div> 62. </div> 63. </div> 64. </div> 65. </div> 66. </div> 67. </div> 68. </div> 69. </div> 70. </div> 71. </div> 72. </div> 73. </div> 74. </div> 75. </div> 76. </div> 77. </div> 78. </div> 79. </div> 80. </div> 81. </div> 82. </div> 83. </div> 84. </div> 85. </div> 86. </div> 87. </div> 88. </div> 89. </div> 90. </div> 91. </div> 92. </div> 93. </div> 94. </div> 95. </div> 96. </div> 97. </div> 98. </div> 99. </div> 100. </div></code`

Untuk yang lebih advancenya anda dapat menggabungkan image berikut ini dengan menggunakan script gallery javascript lainnya atau mengkombinasikan

nya menggunakan database

Chapter 11 Kasus 6. Buku Tamu menggunakan CodeIgniter

Setelah mengetahui konsep dasar CodeIgniter mari kita mulai sebuah contoh penggunaan CodeIgniter. Pada contoh pertama ini akan ditunjukan bagaimana menggunakan query-query dasar pada CodeIgniter. Kita akan membuat Buku Tamu. Selanjutnya ikutilah langkah-langkah berikut ini. Membuat Table Guestbook Pada contoh ini kita akan membuat sebuah aplikasi Buku Tamu. Untuk itu kita membutuhkan sebuah tabel dengan schema sebagai berikut CREATE TABLE guestbook (id bigint auto_increment PRIMARY KEY, namavarchar(50), emailvarchar(50), tanggaldate

Konfigurasi Guestbook

Buka file application/config/database.php. Setting sesuai dengan konfigurasi mysql Anda. Isikan username, password dan nama database yang digunakan. Setelah itu buka juga file application/config/autoload.php ubahlah variabel *autoload['libraries']* (kira – kirabaris41) menjadi

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

Kemudian buka melalui browser. Jika tidak terjadi kesalahan apapun berarti Anda sudah berhasil menyeting database dengan benar

Membuat Model Guestbook

Untuk mendapatkan data dari database maka kita harus memiliki model yang dapat mengambil data tersebut. Oleh karena itu Anda harus menambahkan model berikut di Application/models/guestbook_model.php. Adapun isi dari file tersebut adalah

```
if php class Guestbook_model extends CI_Model
public $table, $record_count;
function Guestbook_model($parent :: Model());
function get_data($start=NULL, $count = NULL) {
    $results=array();
    $this->db->from($this->table, $record_count)
    => $this->db->count_all_results();
    if ($start) {
        if ($count) {
            $this->db->limit($start, $count);
        }
        $query=$this->db->get($this->table);
        return $query->result_array();
    }
    return FALSE;
function add($data) {
    $this->db->insert($this->table, $data);
function update($keyvalue, $data) {
    $this->db->update($this->table, $data);
function delete($idField) {
    $this->db->where($idField);
```

Model diatas terdiri atas lima fungsi yaitu konstruktor, *get_data*, *add*, *update* dan *delete*. Fungsi *get_data* digunakan untuk mengambil data dari database.

Membuat Controller Dan View Pada controller ini kita akan memanggil library, helper dan model yang akan digunakan. Adapun library yang akan digunakan adalah *form_validation* dan *table*. Sedangkan *helper* yang harus digunakan adalah *url* dan *smiley*. Adapun

Selanjutnya adalah pembuatan controller

```
if php if (!defined('BASEPATH')) exit('No direct script access allowed');
class Guestbook extends CI_Controller {
function Guestbook($parent :: CI_Controller());
$this->load->library('pagination');
$this->load->library('form_validation');
$this->load->helper('smiley');
$this->load->helper('url');
$this->load->library('table');
```

```

function show() if(this->validate_data())data['nama']=this->input->
post('nama',TRUE);data['email']=this->input->post('email',TRUE);data['komentar']=this->
input->post('komentar',TRUE);
data['status']='Guestbooksuksesditambahkan';
data['status']='Guestbookgagalditambahkan';paging_uri = 2;if(this->uri-
->segment(paging_uri))start=this->uri->segment(paging_uri);elsestart=0;
limit_per_page = 10;data['tguestbook_list']=this->guestbook->get_data(limit_per_page,start);
config['base_url']=site_url('guestbook');config['total_rows']=this->guestbook-
->table->record_count;config['per_page']=limit_per_page;config['uri_segment']=paging_uri;
this->pagination->initialize(config); data['page_links']=this->pagination-
->create_links();image_array = get_clickable_smileys(base_url().'/smileys/');col_array =this-
->table->make_columns(image_array, 20);data['smiley_table']=this->table->generate(col_array);this-
->load->view('guestbook',data);
function index() this->show();
function validate_data()this->form_validation->set_rules('nama','Nama','required|min_length[5]|max_length[50]|
form_validation->set_rules('email','Email','required|valid_email|htmlspecialchars');this-
->form_validation->set_rules('komentar','Komentar','required|htmlspecialchars');
return(this->form_validation->run() == FALSE)?FALSE : TRUE;
Pada controller ini ada 3 fungsi utama yaitu:

```

- Konstruktor – fungsi ini akan kita gunakan sebagai tempat untuk me-load sumberdaya yang dibutuhkan. Seperti library dan helper. Adapun library yang akan digunakan adalah pagination, table, input (sudah autoload). Sedangkan helper yang akan digunakan adalah url dan smiley.

Berikut ini adalah kode yang digunakan untuk memanggil library dan model tersebut.

```

this->load->library('pagination');this->load->library('form_validation');this-
->load->model('guestbook_model','guestbook');

```

Pada fungsi load model, parameter kedua itu digunakan untuk alias, jadi jika tanpa alias maka kita harus memanggil fungsi model dengan cara *this->guestbook_model*, *sedangkandenganaliascukupthis->guestbook*. • Show – fungsi ini akan menampilkan dan memasukkan data buku tamu. Secara logis isi fungsi ini terbagi dua. Yang pertama adalah bagian untuk memasukkan data buku-tamu.

```

if(this->validate_data())data['nama']=this->input->post('nama',TRUE);data['email']=this->
input->post('email',TRUE);data['komentar']=this->input->post('komentar',TRUE);data['tanggal']=
date('Y-m-d H:m:s');data['status'] = 0;if(this->guestbook->add(data))data['status']='Guestbook
sukses ditambahkan'; else data['status']='Guestbookgagalditambahkan';

```

Jika data sudah tervalidasi dengan benar maka artinya data sudah siap untuk dimasukkan. Maka kita akan menggunakan library input untuk mengambil data form lalu memanggil fungsi *this->guestbook->add(data)* untuk memasukkan data tersebut ke dalam database. Bagian kedua adalah bagian untuk menampilkan data buku tamu. *paging_uri = 2;*

```

if (this->uri->segment(paging_uri))start=this->uri->segment(paging_uri);elsestart=0;
limit_per_page = 10;
data['tguestbook_list']=this->guestbook->get_data(limit_per_page,start);

```



```

    config['base_url'] = site_url('guestbook'); config['total_rows'] = this->guestbook-
    >table->record_count; config['per_page'] = limit_per_page; config['uri_segment'] = paging_uri;
    this->pagination->initialize(config);
    data['page_links'] = this->pagination->create_links();
    image_array = get_clickable_smileys(base_url(). 'smileys/');
    col_array = this->table->make_columns(image_array, 20);
    data['smiley_table'] = this->table->generate(col_array);
    this->load->view('guestbook', data);

```

Jika kita akan menampilkan data dalam pagination perlu diingat dua hal, pertama kita membutuhkan inputan berupa berapa jumlah data yang akan ditampilkan dan data dimulai dari halaman berapa dan data yang dihasilkan dari model harus berisi jumlah total data yang kita punya. Untuk mendapatkan data yang akan diambil mulai dari data ke berapa, library pagination meletakkannya informasi tersebut di uri tertentu. Pada kasus ini kita meletakkannya di uri ke 2. Maka untuk mendapatkannya kita perlu memanggil fungsi `this->uri->segment(2)`; Sedangkan untuk jumlah data sudah tersimpan di `property model(table->record_count)`.
`ffValidate_data`

```

    this->form_validation->set_rules('nama', 'Nama', 'required|min_length[5]|max_length[12]'); this-
    >form_validation->set_rules('email', 'Email', 'required|valid_email|htmlspecialchars'); this-
    >form_validation->set_rules('komentar', 'Komentar', 'required|htmlspecialchars'); return( this-
    >form_validation->run() == FALSE)?FALSE:TRUE;

```

Selanjutnya adalah membuat view yang akan digunakan untuk menampilkan data buku tamu tersebut

```

<html>
<head>
<title>Buku Tamu</title>
<style type="text/css">
body {background-color: fff; margin: 40px; font-family: Lucida Grande, Verdana, Sans-serif; font-size: 14px; color: 4F5155;}
</head>
<body>
<h2>Guest Book</h2>

```

```

<?php echo validation_errors();? ><?php if(isset(status)):?>
<div class="success">
<span class="message_content"> Data Sukses Disimpan </span>
</div>
</div>
<?php unset(v);endif;?>
<?if (tguestbook_list) foreach(tguestbook_list as $value) echo"<li>
<strong>{$value->nama}</strong> <br>
<u>{$value->tanggal}</u>": " . nl2br(parse_smileys($value->komentar, base_url(). 'smileys/')) .
?>
<?php echo page_links;? ><?php echo js_insert_smiley('bukutamu', 'komentar');? ><
<br>
<h4>Isi Buku Tamu</h4>
<form name="bukutamu" method="post">
<label for="nama"> Nama :</label>
<input type="text" name="nama" value=" " >
<?php echo set_value('nama');? >
<br>
<label for="email"> Email :</label>
<input type="text" name="email" value=" " >
<?php echo set_value('email');? >
<br>
<label for="komentar"> Komentar :</label>
<input type="text" name="komentar" value=" " >
<?php echo set_value('komentar');? >
</div>
<input type="Submit" />
</form>
</body>
</html>

```

Pada view tersebut akan di tampilkan data isi buku tamu dalam list beserta form yang akan digunakan untuk menginputkan data. Kita dapat menggunakan validation helper (fungsi `validation_errors()`) untuk menampilkan error yang terjadi.

Chapter 13 Kasus 7. Membuat Shopping Cart Sederhana Jika anda ingin membangun toko online, salah satu fitur yang hampir pasti ada yaitu shop-

ping cart (keranjang belanja). Membuat shopping cart tidaklah begitu sulit, terlebih jika anda menggunakan framework CodeIgniter. CodeIgniter telah menyediakan suatu library/pustaka untuk mempermudah anda membuat shopping cart. Library cart membutuhkan library session, tetapi kita tidak perlu meload library tersebut secara manual. Untuk menggunakannya anda bisa memanggil dengan library loader atau menambahkannya di konfigurasi autoload *this->load->library('cart');*

Penggunaannya juga tidak sulit, untuk menambah item di cart kita bisa menggunakan fungsi inser seperti berikut ini

```
data = array(
);
'id' => 'sku123ABC', 'qty' => 1, 'price' => 39.95, 'name' => 'T-Shirt', 'options' =>
array('Size' => 'L', 'Color' => 'Red')
$this->cart->insert(data);
```

Untuk studi kasus ini, kita akan mengkombinasikannya dengan pemilihan produk dimana produk- produk tersebut telah disimpan didalam database. Adapun langkah-langkah yang dilakukan adalah 1. Membuat database produk

Ketika kita menggunakan shopping chart tentunya akan ada produk yang akan kita jual. Kita akan menggunakan produk yang sangat sederhana pada contoh kasus ini. Sebuah produk hanya memiliki Id, nama produk dan harga. CREATE TABLE 'products' ('id' int NOT NULL AUTO_INCREMENT, 'name' varchar(100) NOT NULL, 'price' int NOT NULL);

```
INSERT INTO 'products' ('id', 'name', 'price') VALUES (1, 'Baju Batman', 30000), (2, 'Mouse Wireless', 434444), (3, 'Tas Ransel', 2344), (4, 'Kasur Busa', 3453);
```

2. Mengkonfigurasi Codeigniter

Kita harus mengkonfigurasi codeigniter terutama di bagian database karena kita akan menggunakan database pada aplikasi ini. Oleh karena itu kita harus mengubah file application /configs/database.php

```
db['default']['hostname'] = "localhost"; db['default']['username'] = "root";
db['default']['password'] = "root"; db['default']['database'] = "shop"; db['default']['dbdriver'] = "mysql";
```

Selain konfigurasi database kita juga mengset konfigurasi base_url di application/configs/config.php config['base_url'] = "http://localhost/shopping_cart/";

3. Membuat Model Produk

Model product mempunyai 2 fungsi yaitu untuk mengambil seluruh data barang, dan fungsi untuk mengambil data barang tertentu sesuai dengan id yang diinginkan. 1. `index.php` 2. `class Product_model extends CI_Model` 3. `function construct()` 4. `parent::construct()` 5. `$db = NULL, $offset = NULL` 6. `$this->db->get('products', limit, offset);` 7. `$result = $this->db->query($sql);` 8. `$result->result();` 9. `$result->result_array();` 10. `$result->result_object();` 11. `$result->result_array();` 12. `$result->result_object();` 13. `$result->result_array();` 14. `$result->result_object();` 15. `$result->result_array();` 16. `$result->result_object();` 17. `$result->result_array();` 18. `$result->result_object();` 19. `$result->result_array();` 20. `$result->result_object();`

Model ini hanya digunakan untuk menampilkan informasi produk yang dijual.

4. Membuat Controller Produk dan Cart

Setelah membuat model maka kita akan membuat dua buah controller yang akan handle penampilan produk dan keranjang belanja. Adapun controller


```

"1" style = "width : 1006. < tr > 7. < th > QTY < /th > 8. < th >
ItemDescription < /th > 9. < thstyle = "text-align : right" > ItemPrice <
/th > 10. < thstyle = "text-align : right" > Sub - Total < /th >
11. < /tr > 12.13. <?php i = 1; ?> 14. 15. i?php foreach($this->cart->
contents() as $items): ?> 16. 17. i?php echo form_hidden($i['rowid'], $items['rowid']); ?>
18.19. < tr > 20. < td > <?php echo form_input(array('name' => $i['qty'],
'value' => $i['qty'], 'maxlength' => '3', 'size' => '5')); ?> < /td >
22. < td > 23. <?php echo $items['name']; ?> 24. 25. i?php if ($this->cart->
has_options($items['rowid']) == TRUE): ?> 26. 27. i?php
28. i?php foreach ($this->cart->product_options($items['rowid']) as $option) {
$option_name => $option_value; ?> 30.31. < strong > <?php echo $option_name; ?> : <
/strong > 32. <?php echo $option_value; ?> < br / > 33.34. <?php endforeach; ?>
35. < /p > 36.37. <?php endif; ?> 38.39. < /td > 40. < tdstyle = "text-align :
right" > 41. <?php echo $this->cart->format_number($items['price']); ?> i?td i
42. i?td style="text-align:right" i43. <?php echo $this->cart->format_number($items['subtotal']);
?> i?td i44. i?tr i45. 46. i?php i ++; ?> 47.48. <?php endforeach; ?>
49.50. < tr > 51. < td colspan = "2" > < /td > 52. < tdclass = "right" > <
strong > Total < /strong > < /td > 53. < tdclass = "right" > 54. i?php echo
$this->cart->format_number($this->cart->total()); ?> i?td i55. i?tr i56. 57.
i?table i58. 59. i?i?php echo form_submit('', 'Update your Cart'); ?> < /p >
60.61. < a href = " <?php echo site_url('produk')? > " > Kembali < /a >

```

Perhatikan baris 15, fungsi `$this->cart->contents()` digunakan untuk mendapatkan seluruh data cart yang ada di `$cart->total()`;

Adapun tampilan dari view diatas adalah gambar berikut ini.

Pada gambar diatas kita dapat mengubah jumlah barang yang dipesan. Subtotal dan total akan otomatis ditambahkan oleh library cart.

Chapter 12 Kasus 8. CodeIgniter dan Ajax

AJAX yang dimaksud disini bukanlah nama club sepakbola yang berasal dari Amsterdam, Anda atau pun nama pahlawan dalam sejarah perang Trojan, tetapi AJAX di sini adalah singkatan dari Asynchronous JavaScript and XML. Pada intinya ajax itu merupakan gabungan beberapa teknologi yang bertujuan untuk menghindari page reload. Dengan menghindari page reload, kita dapat menghindari paradigma click-and-wait serta memberikan sebuah fitur yang cukup kompleks pada website seperti validasi data secara realtime, drag n drop dan fitur-fitur lain yang belum dimiliki web biasa. Dengan AJAX, suatu aplikasi web dapat mengambil data kemudian diolah di client melalui request asynchronous HTTP yang diinisialisasi oleh Javascript, sehingga dapat mengupdate bagian-bagian tertentu dari web tanpa harus memanggil keseluruhan halaman web. Request ini dapat dieksekusi dalam beberapa cara dan beberapa format transmisi data. Dikombinasikannya cara pengambilan data remote dengan interaktivitas dari Document Object Model (DOM) telah menghasilkan generasi terbaru dari aplikasi web yang menggebrak aturan-aturan tradisional tentang apa yang dapat terjadi di dalam web. Keuntungan dari aplikasi web berbasis AJAX adalah memungkinkan untuk membuat website dan aplikasi web yang lebih baik dan lebih responsif. Sehingga meningkatkan kemudahan pengguna. Codeigniter sebenarnya tidak terpengaruh dengan teknik ajax ini karena ajax bekerja di sisi

clinet sedangkan CI bekerja disisi server. Yang perlu dipersiapkan hanya di sisi template dan view. Untuk bagian library template kita melakukan perubahan dengan menambahkan pengecekan apakah sebuah request tersebut merupakan ajax request apa bukan. application/libraries/template.php 1. `if (isset($_POST['ajax']))` 2. `class Template` 3. `protected $ci`; 4.5. `function construct()` 6.7. `this->ci = get_instance();` 8.9.10. `function display($template, $data)` 11.12. `if (!$this->is_ajax())` 13.14. `$data['content'] = $this->load->view($template, $data, true);` 15. `$data['header'] = $this->load->view('template/header', $data, true);`

18. `$data['opmenu'] = $this->load->view('template/menu', $data, true);` 20. `$data['rightmenu'] = $this->load->view('template/sidebar', $data, true);` 22. `$this->load->view('template.php', $data);` 23. 24. `else` 25. 26. `$this->load->view($template, $data);` 27.28.29.30. `function is_ajax()` 31.32. `return ($this->input->server('HTTP_X_REQUESTED_WITH') === 'XMLHttpRequest');` 33.37.

Perhatikan fungsi `is_ajax`, fungsitersebutuntukmengecekapakahrequesttersebutmerupakansebuahrequest

1. `!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"` 2. `http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd` 3. `htmlxmlns="http://www.w3.org/1999/xhtml"` 4. `head` 5. `meta http-equiv="content-type" content="text/html; charset=utf-8"` 6. `link rel="stylesheet" type="text/css" href="public/style.css"` 7. `if (isset($_POST['ajax']))` 8. `<script src="public/js/jquery-1.4.2.min.js">` 9. `</script>` 10. `<script>` 11. `(document).ready(function()` 12. `$('.ajax').click(function(e)` 13. `e.preventDefault();` 14. `.get(this.attr('href'), function(Res)` 15. `$('#content')` 16. `);` 17. `)` 18. `)` 19. `if (isset($_POST['ajax']))` 20. `title` 21. `System Template CodeIgniter` 22. `body` 23. 24. `div id="wrap"` 25. `div id="header"` 26. `if (isset($_POST['ajax']))` 27. `Area Header`

28. `<div>` 29. `if (isset($_POST['ajax']))` 30. `</div>` 31.32. `<div id="menu">` 33. `<!--` 34. `AreaMenu` 35. `-->` 36. `<?php echo $opmenu;` 37. `</div>` 38.39. `<div id="contentwrap">` 40. `<div id="content">` 41. `<!--` 42. `Areacontent` 43. `-->` 44. `<?php echo $content;` 45. `</div>` 46. `<div id="sidebar">` 47. `<!--` 48. `AreaRightMenu` 49. `-->` 50. `<?php echo $rightmenu;` 51. `</div>` 52. `<div style="clear: both;">` 53. `</div>` 54. `<div id="footer">` 55. `<p> Copyrightcopy;` 56. `` 57. `</div>` 58.59. `</div>` 60.61. `</body>` 62. `</html>`

Perhatikan baris 8-18, pada kode tersbut kita menggunakan jquery untuk mempermudah melakukan request ajax. Pada kode di atas kita akan menyari semua link yang memiliki kelas `ajax` lalu meloadnya melalui ajax (lihat view `template/menu.php`). `application/view/template/menu.php`

1. `if (isset($_POST['ajax']))` 2. `if (isset($_POST['ajax']))` 3. `` 4. `PageAjax1` 5. `` 6. `PageAjax2` 7. `` 8. `PageAjax3` 9. `` 10. `PageAjax4`

11. `< /ul >`

Untuk bagian controller tidak melakukan perubahan apa-apa. kita hanya menambahkan page yang dapat dipanggil

```
1. if (! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Welcome extends CI_Controller {
4.5. function __construct()
6.7. parent::__construct();
8. this->load->library('template');
9. this->load->helper('url');
10.11.12. function index()
13.14. this->template->display('welcome_message');
15.16.17. function page1()
18.19. this->template->display('page1');
20. 21.22. function page2()
23. 24. this->template->display('page2');
25.26.27. function page3()
28.29. this->template->display('page3');
30. 31.32. function page4()
33. 34. this->template->display('page4');
35.36.37.38. /* End of file welcome.php */
39. /* Location: ./application/controllers/welcome.php */
/
```

Ketika kita mengakses page 1 secara langsung maka library template akan menampilkan page secara utuh tetapi melalui ajax hanya akan memberikan area content.

Chapter 13 Kasus 9. Codeigniter dan jQuery AutoComplete AutoComplete adalah sebuah fitur dimana kita memberikan saran kepada pengguna mengenai apa yang telah mereka ketikkan di textbox. Jadi user tidak perlu mengetik secara keseluruhan mengenai hal yang ingin dicari. Untuk mendapatkan fitur autocomplete kita dapat menggunakan bantuan javascript terutama jquery. Fitur autocomplete itu sendiri sudah dimiliki oleh plugin jqueryui secara default, sehingga tidak perlu plugin tambahan lagi. Adapun kasus yang cocok untuk menggunakan fitur autocomplete adalah pencarian nama kota. Misalkan kita memiliki database kota dan kita akan menggunakan database tersebut untuk mempermudah pengguna dalam mengisi field kota. Adapun hal-hal yang harus di persiapkan adalah

1. Membuat Tabel Kota dan Konfigurasi Database CodeIgniter

Buatlah tabel dan isi nama kota seperti contoh berikut

```
CREATE TABLE IF NOT EXISTS 'kota' (
'id_kota' int(11) NOT NULL AUTO_INCREMENT,
'nama_kota' varchar(100) NOT NULL,
PRIMARY KEY ('id_kota')
) ENGINE=InnoDB;

-- Data for table 'kota' --
```

```
INSERT INTO 'kota' ('id_kota', 'nama_kota') VALUES (1, 'Aceh'), (2, 'Jakarta'), (3, 'Bandung'), (4, 'Cirebon');
```

Tabel diatas cukup sederhana, kita akan menyimpan id_kota dan nama kota sebagai data utama pada tabel tersebut

Codeigniter. Konfigurasi tersebut berada di file application/config/database.php (perhatikan bab sebelumnya jika Anda bermasalah dalam mengkoneksikan database.

2. Membuat Model Tabel Kota

Setelah membuat table, hal berikutnya adalah membuat sebuah model yang digunakan untuk mengakses data dari tabel tersebut. Perhatikan model berikut

```
1. if (! defined('BASEPATH')) exit('No direct script access allowed');
2. class Kota_model extends CI_Model {
3.4. function __construct()
5. parent::__construct();
6.7.8. function index()
9. this->db->like('nama_kota', keyword, 'after');
10. $query = $this->db->get('kota');
11. return $query->result_array();
12.13.14. /* End of file kota_model.php */
15. /* Location: ./application/model/kota_model.php */
```

Model tersebut hanya memiliki sebuah fungsi find yang digunakan untuk mencari nama kota yang termirip berdasarkan apa yang di ketikkan pengguna. Misalnya ketika pengguna mengetikkan hurup “m” maka akan dicarikan semua

data json seperti di atas.

Jika kita memasukkan huruf maka akan keluar kata-kata yang direkomendasikan oleh script autocomplete. Hal tersebut terjadi karena kita telah memanggil fungsi autocomplete dengan menggunakan id inputan tertentu (*"id_kota"*).*autocomplete*).*id_kotamerupa*

Chapter 14 Kasus 10. Codeigniter dan Openflash Chart

Open Flash Chart adalah perangkat pembangkit grafik berbasis swf. Open flash chart adalah proyek open source. Perangkat ini dapat menampilkan data secara dinamis dan menarik dalam berbagai bentuk animasi grafik, namun demikian grafik dapat juga disimpan dalam bentuk gambar. Keuntungan yang didapatkan ketika menggunakan flash sebagai media grafik adalah selain tampilannya yang menarik ia juga dapat dikendalikan melalui javascript sehingga untuk proses reload, ganti tipe chart dan lain-lain dapat menggunakan library. Open flash chart menerima input berupa data dengan format JSON. Untungnya, ada API dari berbagai jenis bahasa pemrograman yang dapat digunakan untuk meng-generate data ini. Codeigniter sebagai salah satu framework php juga mampu diintegrasikan dengan library lain meskipun berbeda bahasa. Penulis akan menggunakan library yang telah penulis tulis untuk mengintegrasikan open flash chart dengan codeigniter. Pada bab ini penulis tidak akan membahas proses pembuatan library ini tetapi lebih kearah penggunaannya. Adapun class diagram untuk library yang telah penulis buat adalah seperti gambar berikut ini.

Library diatas dapat diperoleh di http://www.koder.com/download/id_openflashchart.zip atau di CD buku. *Library* *KeApplication/Library* Setelah Andamen—download library copy—kan library—library tersebut lalu copy—kan ke application/library. Selain itu kita juga membutuhkan library java. *generate flash object* 2. *Buat Controller Chart* Buatlah sebuah controller yang akan memanggil library *id_chart*. *Ad*

```
1. if (! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Chart extends CI_Controller {
4.5. function construct() {
6.7. parent::construct();
8.9.10. function index() {
11. load—helper('url');
13. this—>load—>library('id_chart/id_chart');
14. chart['c1']=this—>id_chart—>chart_embed('test', 15.800, 250, site_url('chart/example1'), base_url());
16. chart['c2']=this—>id_chart—>chart_embed('test2', 17.800, 250, site_url('chart/example2'), base_url());
18. chart['c3']=this—>id_chart—>chart_embed('test3', 19.800, 250, site_url('chart/example3'), base_url());
20. chart['c4']=this—>id_chart—>chart_embed('test4', 21.300, 300, site_url('chart/example4'), base_url());
22.23. this—load—view('chart', chart);
24.25.26. function example1() {
27.28. this—load—helper('url');
29. this—>load—>library('id_chart/id_chart');
30. for(i=1; i < 30; i++) {
31. data[] = array('label' => 'data'.i,
32. 'value' => rand(1,300));
33. echo this—>id_chart—>set_chart('line');
34.—>set_data(data);
35. —set_vertical();
36.—>render();
37.38.
```

```
39. function example2() {
40. 41. this—>load—>helper('url');
42. this—load—library('id_chart/id_chart');
43. for(i=1; i < 30; i++) {
44. array('label' => 'data'.i,
45. 'value' => rand(1,300));
46.47. echo this—id_chart—>set_chart('bar');
48.—>set_data(data);
49. —set_vertical();
50.—>render();
51.52.53. function example3() {
54.55. this—load—helper('url');
56. this—>load—>library('id_chart/id_chart');
57. for(i=1; i < 30; i++) {
58. data[] = array('label' => 'data'.i,
59. 'value' => rand(1,300));
60. 61. echo this—>id_chart—>set_chart('area');
62.—>set_data(data);
63. —set_vertical();
64.—>render();
65.66.67. function example4() {
68.69. this—load—helper('url');
70. this—>load—>library('id_chart/id_chart');
71. for(i=1; i <
```



```
6; i++) 72. array('label'=> $i, 'data' => rand(20, 300)); 74. 75. echo this-
<id_chart -> set_chart('pie') 76. -> set_data($data) 77. // -> set_radius(20) 78. ->
render(); 79. 80. 81. // * End of file chart.php */ 82. // * Location : ./application/controllers/chart.php */
/
```

Perhatikan fungsi index, fungsi tersebut akan men-generate script javascript yang akan me-load open flash chart. Perhatikan function `chart_embed`, *fungsi tersebut berisi empat parameter* `di` (nama pengenalan script), `width` (lebar chart), `height` (panjang chart), `url` (url yang berisi data json), `base` (letak flash script berada). Perhatikan kembali fungsi example1-4 itu adalah contoh fungsi untuk men-generate data json yang akan dipakai oleh flash chart

3. Buat View Chart

View yang akan kita buat mirip seperti view pada chapter jquery auto-complete, hanya sebagai view yang berisi script javascript yang akan memanggil open flash chart dan menampilkan chart. Adapun view tersebut adalah :

```
1. <html> 2. <head> 3. <title> Welcome to CodeIgniter </title> 4. <script type="text/javascript"
src="http://www.dreamwaver.com/swfobject.js"> 5. <style type="
text/css"> 6. 7. body { background-color: #fff; 9. margin: 40px; 10. font-family: LucidaGrande, Verdana,
Arial, sans-serif; 11. } 12. 13. 14. 15. 16. 17. </head> 18. <body> 19. 20. <h1> Line Chart </h1> 21. <
?php echo $c1; 22. <h1> Bar Chart </h1> 23. <?php echo $c2; 24. 25. <h1>
Area Chart </h1> 26. <?php echo $c3; 27. <h1> Pie Chart </h1> 29. <?php
echo $c4; 30. </body> 31. </html>
```

Perhatikan baris ke empat, disana kita mencoba me-load swfobject.js yang bertujuan me-load script open flash chart. Jika semua terinstall dengan benar maka akan keluar tampilan seperti di bawah ini

Variabel `c1`, `c2`, `c3` dan `c4` merupakan variable yang menyimpan script-script tersebut berasal dari fungsi `example1`, `example2`, `example3` dan `example4` lah yang menentukan tipe beserta data dari masing-masing chart.

Menggunakan Active Record

Active Record (AR) adalah sebuah pattern / pendekatan untuk membaca data dari sebuah table atau view dengan cara membungkusnya dalam sebuah kelas. Sehingga tidak dibutuhkan SQL jika kita menggunakan active record. Keuntungan yang didapat adalah kesederhanaan dalam pengkodean program dan fleksibilitasnya ketika terjadi pergantian skema atau pun jenis database (bebas dari sintaks-sintaks SQL yang bersifat khusus ke database tertentu). CodeIgniter tidak sepenuhnya meniru pattern Active Record, tetapi ia melakukan sedikit modifikasi terhadap pattern tersebut. Dengan menggunakan Active Record CI maka proses pengambilan data, insert, update dan delete menjadi lebih sederhana. `$this->db->get('users')`

Query di atas akan menghasilkan query "SELECT * FROM users" dan setara dengan

```
$this->db->query('SELECT * FROM users');
```

Ketika kita tidak menggunakan fungsi apapun maka fungsi `get` akan menghasilkan seluruh record yang berada di dalam tabel tersebut, tetapi jika diberi fungsi `select()`, `where()`, dan lain-lain maka hasilnya akan berbeda.

Selecting Data

Untuk mengambil data dari database dengan field-field tertentu kita dapat menggunakan perintah `select()`. Contoh: `$this->db->select('username, password, email');`

```
$db->get('users');
```

Query di atas setara dengan *this->db->query('SELECT username,password,email FROM users');*

Selain method select, terdapat juga method-method lainnya seperti from(), where(), where_in(), group_by, order_by, like, distinct() dan lain-lain. Semua method tersebut jika dipanggil akan menghasilkan query yang sesuai.

```
$db->select("sum(budget) as sum_budget, site_id_mapping.*")
->where('site_id_mapping.site_id', $site_id)
->where('ads.status', 'approved')
->where('request.status', 'active')
->where('ad_operational.status', 'active')
->join('ads', 'ads.id_ad = site_id_mapping.ad_id')
->join('advertiser', 'advertiser.id_advertiser = ads.advertiser_id')
->join('site_counter', 'site_counter.site_id = site_id')
->join('sites', 'site_counter.site_id = sites.id_site')
->join('publishers', 'sites.publisher_id = id_publisher')
->from('site_id_mapping')
->order_by('ad_priority_level', 'desc')
->order_by('point', 'desc')
->group_by('ad_id')
->limit(0, 10)
->get()
->result();
```

Insert Data

Selain pengambilan data, Active record juga menyediakan fungsionalitas untuk insert data. Untuk menginsert sebuah data kita hanya memerlukan dua parameter yaitu nama table dan data yang akan dimasukkan. Data yang akan dimasukkan harus memiliki format berupa array asosiatif ataupun object. Adapun keuntungan menggunakan active record untuk menginsert data selain sederhana untuk digunakan, adalah method ini secara build-in disertai fungsi untuk penanganan escaping, jadi query kita sudah bisa dikatakan aman dari kesalahan, contoh: *data = array('username' => 'Ibnoe', 'email' => 'xibnoe@gmail.com');* *this->db->insert('users', \$data);*

Selain dengan cara di atas, kita juga bisa memasukkan data yang diinginkan satu persatu. Contohnya: *data = array('username' => 'Ibnoe', 'email' => 'xibnoe@gmail.com');* *this->db->set(\$data);* *this->db->insert('users');*

Atau

```
$this->db->set('username', 'ibnoe');
$this->db->set('email', 'xibnoe@gmail.com');
```

```
$this->db->insert('users');
```

Kedua Contoh di atas menghasilkan query yang sama.

Update Data

Selain insert dan select data, CodeIgniter juga menyediakan fungsi untuk update. Adapun penggunaannya mirip seperti penggunaan insert, contoh:

```
$this->db->set('username', 'ibnoe');
$this->db->set('password', '123456');
```

```
$this->db->update('users');
```

Query di atas akan menghasilkan query "update users set username=ibnoe, password=123456". Cara yang lebih sederhana dengan menggunakan array asosiatif *data = array('username' => 'ibnoe', 'password' => '123456');* *this->db->where('id', 5);* *this->db->update('users', \$data);*

Delete Data

Untuk melakukan delete pada CodeIgniter, kita dapat menggunakan perintah berikut ini:

```
$this->db->where('id', 5);
$this->db->delete('table_name');
```

Chapter 8 Kasus 2. CRUD dan Pagination Database Setelah mengetahui cara menggunakan database maka kita akan mencoba melakukan perintah dasar

pada sistem informasi CRUD (create, read, update dan delete) data menggunakan CodeIgniter. Selain CRUD, pada contoh ini juga akan dilakukan proses sorting dan pagination. Studi kasus yang akan diangkat adalah form data siswa. Pada bab ini aplikasi CRUD yang akan dibuat adalah aplikasi yang memiliki fungsi untuk menampilkan seluruh data siswa. Data yang ditampilkan memiliki pagination (pembagian jumlah record yang ditampilkan pada sebuah halaman website). Pagination dibutuhkan karena kita tidak mungkin menampilkan semua data dalam satu layar. Selain itu daftar siswa tadi akan memiliki fitur pengurutan berdasarkan field dan memiliki menu action per daftar siswa. Untuk melakukan pagination di CodeIgniter telah disediakan sebuah kelas yaitu kelas pagination. Untuk menggunakan kelas ini minimal harus memiliki tiga data yang selanjutnya akan menjadi konfigurasi pada library ini. Ketiga data tersebut adalah pagination $base_url$, $total_row$ ($jumlah\ total\ baris\ data\ record\ yang\ kita\ punya$) dan per_page ($jumlah\ baris\ yang\ akan\ ditampilkan\ per\ halaman$).

```

<?php
load->library('pagination');
$this->load->helper('url');
$config['base_url'] = site_url('results/page/');
$config['total_rows'] = 200;
$config['per_page'] = 20;
$this->pagination->initialize($config);
echo $this->pagination->create_links();

```

Contoh diatas menunjukkan cara penggunaan library pagination. Output dari library ini adalah link- link yang terdiri atas halaman dan link ke halaman berikut (next) atau sebelumnya (prev). $base_url$ harus berisi sebuah fungsi dari controller. Fungsi tersebut akan menerima parameter di atas masi banyak parameter lainnya yang dapat digunakan untuk mengcustomisasi library pagination.

```

3; //mengganti tulisan next menjadi Berikutnya
> $config['next_link'] = 'Berikutnya'; //mengganti tulisan prev
sebelumnya dengan config['prev_link'] = '<';

```

Adapun langkah-langkah yang harus dilakukan untuk membuat CRUD dan pagination adalah

1. Membuat Database Dan Table Data Siswa

Untuk memulai, Anda harus memiliki sebuah database. Selain database, Anda harus membuat sebuah tabel siswa. Adapun table yang harus di buat adalah CREATE TABLE IF NOT EXISTS 'siswa' ('id' bigint(20) NOT NULL AUTO_INCREMENT, 'nama' varchar(50) DEFAULT NULL, 'alamat' varchar(200) NOT NULL, 'jenis_kelamin' enum('laki-laki','perempuan') NOT NULL, 'tanggal_lahir' date NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;

Tabel siswa di atas memiliki lima field yaitu id siswa, nama siswa, alamat siswa, jenis kelamin, dan tanggal lahir. Tabel ini digunakan untuk menyimpan data siswa.

2. Konfigurasi CodeIgniter

Sebelum memulai pengkodean, sebaiknya kita melakukan konfigurasi terhadap tools yang akan digunakan. Ada tiga konfigurasi yang harus disesuaikan, antara lain: config.php, database.php dan router.php. Adapun yang harus dilakukan adalah:

- config.php - pada file konfigurasi ini yang perlu diubah adalah $base_url$. Konfigurasi ini berfungsi untuk menentukan path dasar dari aplikasi. Konfigurasi ini akan digunakan untuk menentukan path dari database.

```

$config['base_url'] = 'http://localhost/crud'; //ganti sesuai path di server Anda

```

```

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'crud';
$db['default']['dbdriver'] = 'mysql';

```

router.php - pada file konfigurasi ini yang perlu diubah adalah $routing_rule$. Pada konfigurasi ini yang diubah adalah $routing_rule$.

```

$route['default_controller'] = 'siswa';

```

3. Membuat Model Siswa

Model siswa ini adalah model yang bertugas dan bertanggung jawab untuk

melakukan proses- proses yang berhubungan dengan database. Misalnya menyimpan, mengubah, menghapus serta mengambil data dari database. Biasanya untuk memudahkan kita, pembuatan model mengikuti desain database atau dapat diartikan satu tabel dapat diwakili oleh satu model. Model tersebutlah yang bertanggung jawab pada semua operasi pada tabel tersebut. Pada aplikasi CRUD ini kita akan membuat sebuah model siswa. Adapun isi dari model siswa adalah 1. `1. ?php` 2. 3. `class SiswamodelextendsCIModel4.5.privateprimarykey =`

```
id';6.privatetablename = 'siswa';7.8.functionconstruct()9.parent :: construct();10.11.12.functiongetpagedi
0,13.ordercolumn = "",ordertype = 'asc'14.15.if(empty(ordercolumn)||empty(ordertype))16.this-
db-iorderby(this-iprimarykey,'asc');17.else18.this-db-iorderby(ordercolumn,ordertype);19.returnthis-
db-iget(this->tablename,limit,offset);20.21.22.functioncountall()23.returnthis-
db-icountall(this->tablename);24.25.26.functiongetbyid(id)27. this->db->
where(this->primarykey,id);28. return this->db->get(this->tablename);29.30.31.functionsave(person)
32. this->db->insert(this->tablename,person);33. return this->db->
insertid();34.35.36.functionupdate(id,person)37.this->db->where(this->primarykey,id);
38. this->db->update(this->tablename,person);39. 40.
```

```
41. function delete(id)42.this->db->where(this->primarykey,id);43. this->
db->delete(this->tablename);44.45.
```

4. Membuat Controller Dan View Data Siswa

Controller berisi logika untuk melakukan proses-proses yang ada. Adapun proses yang akan ditangani oleh controller ini adalah menampilkan daftar siswa, menambah siswa, mengubah data siswa, menghapus data siswa dan melihat detail siswa. 1. `1. ?php` 2. `class Siswa extends CIController3.4.privatelimit=10; 5.`

```
6. function construct() 7. 8. parent:: construct(); 9. load library dan helper
yang dibutuhkan10. this->load->library(array('table','form_validation'));11.this-
iload-⊂>ihelper(array('form','url'));12. this->load->model('siswamodel','TRUE');13.14.15.functioninde
id',16.ordertype = 'asc'17.18.if(empty(offset)) offset = 0;19.if(empty(ordercolumn))ordercolumn = '
id';20.if(empty(ordertype))ordertype = 'asc';21.//TODO : checkforvalidcolumn22.23.//loadaddasiswa24.sis
siswamodel->getpagedist(this-⊂>ilimit,25. offset,ordercolumn,ordertype)->
result();26.27.//generatepagination28.this-⊂>iload-ilibrary('pagination');29. config['baseurl'] =
siteurl('siswa/index/');30.config['totalrows'] =this-⊂>isiswamodel->countall();31.config['perpage'] =this-
ilimit;32. config['urisegment'] = 3;33.this-⊂>ipagination-iinitialize(config);34.data['pagination'] =this->
pagination->createiinks();35.36.//generatetabledata37.load->library('table');38.table->
seteempty("nbsp;");39.40.this-⊂>itable-isetheading(41.'No',42.anchor('siswa/index/'.offset.'/nama/'.new_order
44. anchor('siswa/index/'.offset.'/jeniskelamin/'.new_order,45.'JenisKelamin'),46.anchor('siswa/index/
mm-yyyy)'),48.'Actions'49.);50.i=0+offset;51.foreach(siswas as siswa)52.this-
itable-⊂>iaddow(++i,53. siswa->nama,54.siswa-⊂>ialamat,55. strtoupper(siswa->
jeniskelamin) == 'M'?56.57.date('d-m-Y',strtotime(58.siswa-⊂>itanggalahir)),59.anchor('siswa/view/'.sis
iid,60. 'view',array('class'=⊂>i'view')).' '. 61. anchor('siswa/update/'.siswa->
id,62.'update',array('class'=>'update')).'63.anchor('siswa/delete/'.siswa-
iid,64. 'delete',array('class'=⊂>i'delete',65. 'onclick'=⊂>i"return confirm(66.
'Apakah Anda yakin ingin menghapus data siswa?')"))68. );69. 70.
```

```
data['table'] =this-⊂>itable-igenerate();71. 72. if(this->uri->segment(3) == '
deletesuccess')73.data['message'] = 'Data berhasil dihapus';74. else if(this->
uri->segment(3) == 'addsuccess')75.data['message'] = 'Data berhasil ditam-
bah';76. else77. data['message'] = "";78.//loadview79.this-⊂>iload-iview('siswaList',data);80.81.82.83.function
```

```

siswa baru'; 86. data['action'] = site_url('siswa/add'); 87.data['link_back'] =
anchor('siswa/index/', 88.'Back to list of siswas', array('class' => 'back')); 89.90.this-
->isset($_rules()); 91.92.//run validation 93.if (this->form_validation->run() ==
FALSE) 94.95.96.data['title'] = 'Add new siswa'; 97. 98. 99. 100. data['siswa']['alamat'] =''
;
101. 102. 103. anchor('siswa/index/', 104. 'Lihat Daftar Siswa', array('class' => 'back'));
105. 106. this->load->view('siswaEdit', data); 107. 108. else 109. //
save data 110. ma'=$_this->input->post('nama'), 111.input->post('alamat'), 112.'jenis_kelamin' => this-
->input->post('jenis_kelamin'), 113.date('Y - m - d', 114.strtotime(this->input-
->post('tanggal_lahir')))); 115.id=$_this->siswa_model->save(siswa); 116. 117.
// set form input nama="id" 118. this->validation->id = id; 119. 120.
redirect('siswa/index/add_success'); 121.122.123.124.function view(id) 125. //
set common properties 126. data['title'] = 'siswaDetails'; 127.data['link_back'] =
anchor('siswa/index/', 128.'Lihat daftar siswas', array('class' => 'back')); 129.130.
131. 132. 133.
// load view
siswa_model->get_by_id(id)->row();
134. this->load->view('siswaView', data); 135. 136. 137. function
update(id) 138.//set common properties 139.data['title'] = 'Update siswa'; 140. this->
load->library('form_validation'); 141.//set validation properties 142.this->isset($_rules()); 143.data['action'] = ('sis-
->form_validation->run() == FALSE) 147.148.data['message'] =''; 149. data['siswa'] = this-
->siswa_model->get_by_id(id)->row_array(); 150.$POST['jenis_kelamin'] = 151.strtoupper(data['siswa']['jenis_kelam
in']); 152.date('d-m-Y', 153.strtotime(data['siswa']['tanggal_lahir'])); 154.155.//set common properties 156.data['title'] =
siswa'; 157. data['message'] ='';
158. 159. else 160. // save data 161. id = this->input->post('id'); 162.
siswa = array('nama' => this->input->post('nama'), 163. 'alamat' => this->
input->post('alamat'), 164.'jenis_kelamin' => this->input->post('jenis_kelamin'), 165.'tanggal_lahir' =>
date('Y - m - d', 166.strtotime(this->input->post('tanggal_lahir')))); 167.this->siswa_model->
update(id, siswa); 168.data['siswa'] = this->siswa_model->get_by_id(id)->row_array(); 169.//set user message 1
siswa success'; 171. 172. data['link_back'] = anchor('siswa/index/', 173.'Lihat daftar siswas', array('class' =>
'back')); 174.//load view 175.this->load->view('siswaEdit', data); 176.177.178.function delete(id)
179. // delete siswa 180. this->siswa_model->delete(id); 181. // redirect
to siswa list page 182. redirect('siswa/index/delete_success', 'refresh'); 183.184.185.//validation rules 186.function
->isset($_rules)->set_rules('nama', 'Nama', 189.'required|trim'); 190.this->form_validation->
set_rules('jenis_kelamin', 'Password', 191.'required'); 192.this->form_validation->
set_rules('alamat', 'Alamat', 193.'required|callback_validate'); 194.this->form_validation->
set_rules('tanggal_lahir', 'Tanggal Lahir', 195.'required'); 196.197.198.199.//date validation callback 200.function
201. 202. if(!preg_match('/[0 - 9]4 - [0 - 9]2 - [0 - 9]2/', $str)) 203.204.this-
->form_validation->set_message('validate', 205.'date format is not valid yyyy -
mm - dd'); 206.return false; 207.208.else 209.210.return true; 211.212.213.
Perhatikan baris ke-15 pada fungsi index($offset = 0, $order_column = 'id', $order_type =
'asc'). Pada fungsi tersebut ada 3 parameter yang bersi fat optional yang dapat diberikan melalui URI. Parameter of
berfungsi untuk menentukan record pertama yang akan ditampilkan, $order_column ber fungsi untuk menentukan
70).

```

Untuk mendapatkan tampilan seperti diatas maka perlu dibuat sebuah view seperti berikut ini. View ini disimpan dalam sebuah file bernama siswaL-

```

ist.php 1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" 2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/1999/xhtml"> 4. <head> 5. <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"> 6. <title>SIMPLE
CRUD APPLICATION</title> 7. <link href="i?php echo base_url(); ? > style/style.css"8.rel =
"stylesheet" type = "text/css" / > 9. </head> 10. <body> 11. <div class =
"content"> 12. <h1>ContohInsertUpdatedanddelete </h1> 13. <div class = "paging"><?php echo pagination;?></div> 14. <div class="data"><?php
echo table;?></div> 15. <div class = "paging"><?php echo pagination;?></div><br/>
16. <?php echo anchor('siswa/add/', 17. 'Tambah Siswa baru', array('class'='add'))></div> 19. </body> 20. </html>

```

Selain untuk menampilkan data, controller tadi memiliki fungsi ubah dan tambah data siswa. Kedua fungsi tersebut pada intinya sama. Perbedaannya adalah action atau fungsi model yang dipanggil dan pemanggilan record yang akan di edit. Perhatikan fungsi add pada baris 82. Pada fungsi add tersebut dilakukan pemanggilan terhadap fungsi *this->et_rules(); fungsitersebutdigunakanuntukmengesetvalidation* *is_siswa_model->saveuntukmenyimpandatatatersebut.Adapunviewyangdigunakanadalah*

```

1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> 3. <html
xmlns="http://www.w3.org/1999/xhtml"> 4. <head> 5. <meta http-equiv="Content-
Type" content="text/html; charset=iso-8859-1"> 6. <title>SIMPLE CRUD AP-
PLICATION</title> 7. <link href="i?php echo base_url(); ? > style/style.css"8.rel =
"stylesheet" type = "text/css" / > 9. </head> 10. <body> 11. <div class =
"content"> 12. <h1><?php echo title;?></h1> 13. <?php echo message;? >
14. <?php echo validation_errors();? > 15. <?php echo form_open(action);?> 16.
<div class="data"> 17. <table> 18. <tr> 19. <td width="30"> 20. <td <input type="text" name="id" disabled="disabled"
21. value="i?php echo (isset($siswa['id']))? $siswa['id'] : ''> /> <td> 22. <input
type="hidden" name="id" value="i?php echo 23. (isset($siswa['id']))? $siswa['id'] : ''> />
24. </tr> 25. <tr> 26. <td align="top"> <span style="color:red;"> * /> <span> /> <td>
27. <td <input type="text" name="nama" class="text" value="i?php echo 28. (isset($nama))? $nama : ''>
29. <?php echo form_error('nama');? > </td> 30. </tr> 31. <tr> 32. <td align = "top"> Alamat </td> 33. <td <input type = "text" name =
"alamat" class = "text" value = "<?php echo 34. (isset($alamat))? $alamat : ''> :siswa[alamat];?> />
35. <?php echo form_error('alamat');? > </td> 36. </tr> 37. <tr> 38. <td align = "top"> jenis_kelamin <span style = "color : red;"> * <
/ > <span> <td> <input type = "radio" name = "jenis_kelamin" value =
"M" <?php echo 40. (isset($jenis_kelamin))? $jenis_kelamin : 'M'> TRUE);? > / > Laki-Laki 41. <input type = "radio" name = "jenis_kelamin" value = "F" <?php echo 42. (isset($jenis_kelamin))? $jenis_kelamin : 'F'> FALSE);? > / > Perempuan 43. <?php echo form_error('jenis_kelamin');? > </td> 44. </tr> 45. <tr> 46. <td align = "top"> Date of birth (dd - mm - yyyy) <span style = "color : red;"> * < / > <span> <td> <input type = "text" name = "tanggalahir" class = "text" 49. value = "<?php echo (isset($tanggalahir))? $tanggalahir : ''>
" / >
51. <?php echo form_error('tanggalahir');? > </td> 52. </tr> 53. <tr> 54. <td> <td> <input type = "submit" value = "Save" / > </td> 56. </tr> 57. </table> 58. </div> 59. </form>

```


Dengan menyusun view-view yang ada, kita sebenarnya dapat membuat sebuah template library yang cukup powerful. Idenya sederhana saja, cukup membagi sebuah halaman menjadi beberapa area. Sebagai contoh kita akan menggunakan template dari opendesign.org. Dari desain tersebut dapat kita bagi menjadi area header, top menu, right menu dan content. Masing-masing area tersebut akan digabungkan dalam satu halaman utuh yang dinamakan template. Masing-masing area tersebut akan ditangani oleh sebuah view agar tidak terjadi duplikasi dan akan mempermudah penggunaan kembali area tersebut. Adapun area-area tersebut dapat dilihat pada gambar berikut.

Template yang akan dibuat terbagi menjadi empat bagian yaitu Header (berfungsi sebagai tempat logo dan slogan aplikasi), Top menu (bagian menu utama disebelah atas), Right Menu (menu navigasi tambahan disebelah kanan) dan sebuah Content. Area content ini lah yang seringkali berubah pada setiap page. Untuk mengimplementasikan sistem templating tersebut sebaiknya kita buat sebuah library template. Adapun langkah-langkah yang harus dilakukan adalah: 1. Membuat Library Template

Fungsi library tersebut hanya mengatur view mana yang akan dipanggil dan meletakkannya di dalam template. Kenapa membuatnya dalam sebuah library? Karena dengan cara ini memberikan fleksibilitas terhadap sistem template. Cara pembuatan library ini diawali dengan membuat sebuah file bernama template.php pada folder application/libraries. Adapun isi file template.php adalah: 1. `!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"` 2. `http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd` 3. `xmlns="http://www.w3.org/1999/xhtml"` 4. `function construct()` 5. `6.this->load->view(12,'template',data,true);` 10. `11.data['content']=this->load->view(14,'template/header',data,true);` 13. `14.data['header']=this->load->view(14,'template/header',data,true);`

15. `15.data['topmenu']=this->load->view(16,'template/menu',data,true);` 17. `17.data['rightmenu']=this->load->view(18,'template/sidebar',data,true);` 19. `19.this->load->view('template.php',data);` 20. 21.

Perhatikan fungsi display, disana ada dua parameter yaitu template dan data. Parameter data berfungsi sebagai data yang akan dikirimkan ke controller. Sedangkan template adalah view yang akan dipanggil untuk ditampilkan sebagai content utama. Pada template ini akan dibagi menjadi empat area yaitu content, header, top menu, dan right menu. Masing-masing area diisi oleh sebuah view (perhatikan baris 11,13,15,17). Masing-masing view tadi di-load dan disimpan dalam memori. Lalu digabungkan kedalam sebuah template (baris 19). Fungsi yang ada pada library ini sebenarnya dapat kita tambahkan lagi, misalnya untuk keperluan seo, kita ingin memanipulasi title dan meta tag dll. Library diatas merupakan contoh paling sederhana ketika kita ingin membuat sebuah sistem template sendiri. 2. Membuat Template Layout View

Sebuah template layout view adalah sebuah view yang akan menggabungkan masing-masing view menjadi suatu bentuk kesatuan. Adapun isi dari template layout adalah: Application/view/template.php

1. `!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"` 2. `http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd` 3. `xmlns="http://www.w3.org/1999/xhtml"` 4. `function construct()` 5. `6.this->load->view(12,'template',data,true);` 10. `11.data['content']=this->load->view(14,'template/header',data,true);` 13. `14.data['header']=this->load->view(14,'template/header',data,true);` 15. `15.data['topmenu']=this->load->view(16,'template/menu',data,true);` 17. `17.data['rightmenu']=this->load->view(18,'template/sidebar',data,true);` 19. `19.this->load->view('template.php',data);` 20. 21.


```

public/style.css"/ > 7. < title > SistemTemplateCodeIgniterTemplate <
/title > 8. < /head > 9. < body > 10. < divid = "wrap" > 11. < divid =
"header" > 12. <!--13.AreaHeader14. -- > 15. <?php echo header;? >
16. < /div > 17. < divid = "menu" > 18. <!--19.AreaTopMenu20. -- >
21. <?php echo topmenu;? > 22. < /div > 23. < divid = "contentwrap" >
24. <div id="content"> 25. <!-- 26. Area content 27. <?php echo
content;? > 29.30. < /div > 31.32. < divid = "sidebar" > 33. <!--34.AreaRightMenu35. --
> 36. <?php echo rightmenu;? > 37. < /div > 38. < divstyle = "clear :
both;" >< /div > 39. < /div > 40.41. < divid = "footer" > 42. < p >
Copyrightcopy;< ahref = "" > Ibnoe < /a > |Designby43. < ahref =
"http : //www.readcrazyreviews.com" > 44.ReadCrazyReviews < /a ><
/p > 45. < /div > 46.47. < /div > 48.49. < /body > 50. < /html >

```

File diatas merupakan penggabung dari semua bagian yang telah dibahas. Perhatikan baris 15, 21, 28, dan 36. Keempat variabel tersebut akan diisi view dari masing-masing bagian. Berikut ini adalah view-view yang menjadi part/area dari template

a. View header

```

Application/view/template/header.php 1. <h1><a href=""><Red Wall Tem-
plate</a></h1> 2. <h2>Your web site slogan goes here</h2>

```

b. View Top menu

```

Application/view/template/menu.php 1. <ul> 2. <li><a href=""><Home</a></li>
3. <li><a href=""><Articles</a></li> 4. <li><a href=""><About Us</a></li> 5. <li><a
href=""><Resources</a></li> 6. <li><a href=""><Contact Us</a></li> 7. </ul>

```

c. View sidebar menu

```

Application/view/template/sidebar.php 1. <h3>Main Content</h3> 2. <ul>
3. <li><a href=""><Conflantur</a></li>
4. <li><a href=""><Externarum vi</a></li> 5. <li><a href=""><Essem pauloi</a></li>
6. <li><a href=""><Aeque fecto i</a></li> 7. <li><a href=""><Quo locis utensi</a></li>
8. </ul> 9. 10. <h3>Related Web Sites</h3> 11. <ul> 12. <li><a href=""><Plus
vi</a></li> 13. <li><a href=""><Agi praecise</a></li> 14. <li><a href=""><Infinitum
veritates</a></li> 15. <li><a href=""><Corporea ac perpauca</a></li> 16. <li><a
href=""><Aeque fecto</a></li> 17. </ul>

```

```

Setelah memiliki view ketiga area tersebut (header, top menu dan side-
bar) maka kita siap untuk menggunakan sistem template ini. Selanjutnya
adalah membuat sebuah controller. Agar lebih mudah pembuatan controller-
nya maka gunakan saja controller default dari CodeIgniter, yaitu dengan sedikit
melakukan perubahan kode program pada controller-nya. Application/controllers/welcome.php
1. <?php if (! defined('BASEPATH')) exit('No direct script access allowed'); 2.
3. class Welcome extends CI_Controller{4.5.functionconstruct()6.7.parent :: construct();8.this-
load->library('template');9. this->load->helper('url');10.11.12.functionindex()13.14.this-
load->template->display('welcome_message');15.16.17.functioncontohparameter()18.19.this-
load->template->display('view_parameter',20.array('judul' => 'judulView'));21.22.23.24.25./*
Endoffilewelcome.php*/26./*Location : ./application/controllers/welcome.php*/
/

```

Perhatikan pada baris 8, di situ kita me-load library yang telah dibuat tadi. Pada baris 14 dan 19 kita menggunakan fungsi display untuk menampilkan tem-

Pada gambar diatas ada dua tipe page yaitu page yang bisa diakses oleh semua orang dan page yang hanya boleh diakses oleh member. Hal tersebut sederhana jika kita hanya menghandle dua page tetapi jika pagennya ada banyak maka kita akan mengecek satu-satu informasi user. Hal tersebut kurang aman karena kode kita terduplikasi keseluruh aplikasi. Kita akan mencoba untuk meng-extend controller membuat kasus ini menjadi lebih simple. Untuk controller untuk user yang login harus menggunakan member_controller.

Jika kita perhatikan class diagram diatas maka semua member_controller sudah memiliki fungsi untuk pengecekan

2. Membuat Tabel User Untuk sistem autentikasi kita akan membuat tiga tabel. Tabel pertama yaitu tabel user berisi data informasi login user, tabel user_group berisi data pengelompokan user menjadi group, dan tabel tracker berisi fungsi untuk menyimpan data dalam MYISAM;

3. Membuat Library Access Setelah tabel terbentuk barulah kita membuat library untuk mengakses dan mengverifikasi data user. Library tersebut kita sebut Access. Adapun kode programnya adalah sebagai berikut application/libraries/access.php

```
1. i?php if (!defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Access 4. 5. public user; 6. 7. /** 8. *Constructor 9. */ 10. function construct() 11. 12. this->CI = get_instance(); 13. auth = this->CI->config->item('auth'); 14. 15. this->CI->load->helper('cookie'); 16. this->CI->load->model('users_model'); 17. 18. this->users_model = this->CI->users_model; 19. 20. 21. /** 22. *Cek login user 23. */ 24. 25. function login(username, password) 26. 27. 28. result = this->users_model->get_login_info(username);
29. 30. if (result) // Result Found
31. 32. password = md5(password); 33. if (password === result->password) 34. 35. // Start session 36. this->CI->session->set_userdata('user_id', 37. result->user_id); 38. return TRUE; 39. 40. 41. return FALSE; 42. 43. 44. 45. /**
*46. *cek apakah sudah login 47. */ 48. function is_login() 49. 50. return ((this->CI->session->userdata('user_id')) ? TRUE : FALSE); 51. 52. 53. /** 54. *Logout 55. */ 56. */ 57. function logout() 58. 59. this->CI->session->unset_userdata('user_id'); 60. 61. 62.
```

Library access di atas membutuhkan library session dan database, jadi sebelum menggunakan library tersebut pastikan Anda telah menyetting konfigurasi CodeIgniter dengan benar. Perhatikan fungsi login, fungsi itu bertugas untuk melakukan pengecekan terhadap data username dan password yang diberikan oleh user. Fungsi login akan mengambil semua informasi tentang user berdasarkan username (baris 28). Setelah data didapatkan maka dilakukan pencocokan username dan password (baris 33). Jika passwordnya cocok maka session user tersebut disimpan sebagai penanda bahwa user telah login. 4. Membuat Library Access Untuk mendapatkan data user tersebut maka kita membutuhkan sebuah model. Model ini sangat sederhana hanya terdiri satu fungsi yaitu get_login_info. Fungsi tersebut akan mengembalikan object data user apabila username yang dimasukkan ada di database.

```
application/models/users_model.php
1. i?php if (!defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Users_model extends CI_Model 4. 5. 6. public table = 'users'; 7. public primary_key = 'user_id'; 8. 9. function construct() 10. 11. parent :: construct(); 12. 13. 14. function get_login_info(username, password) 15. 16. this->db->where('username', username); 17. this->db->limit(1); 18. query = this->db->get(this->table); 19. return (query->
```

`num_rows() > 0)?query->row() : FALSE; 20. 21. 22.`

5. Membuat Controller Member

Pertama-tama kita akan membuat controller untuk login. Controller tersebut kita beri nama member. Controller member ini mempunyai dua fungsi utama yaitu login dan logout 1. `if (!defined('BASEPATH')) exit('No direct script access allowed');` 2. 3. `class Member extends CI_Controller{` 4. `function __construct(){` 5. `parent::__construct();` 6. `load->library('access');` 7. 8. 9. 10. 11. 12. `function index()` 13. 14. 15. `if ($this->access->logout())` 16. `if ($this->login())` 17. 18. 19. 20. `function login()` 21. 22. 23.

24. `$this->load->library('form_validation');` 25. `$this->load->helper('form');` 26. 27. `$this->form_validation->set_rules('username','Username','trim|required|strip_tags');` 28. `$this->form_validation->set_rules('password','Password','trim|required');` 29. `$this->form_validation->set_rules('token','token','callback_check_login');` 30. 31. `// $this->output->enable_profiler(1);` 32. `if ($this->form_validation->run() == FALSE)` 33. 34. 35. 36. `$this->template->display('member/login');` 37. 38. `else` 39. 40. `redirect('dashboard');` 41. 42. 43. 44. `function logout()` 45. 46. `$this->access->logout();` 47. `redirect('member/login');` 48. 49. 50. `function check_login()` 51. 52. 53. `$username = $this->input->post('username', TRUE);` 54. `$password = $this->input->post('password', TRUE);` 55. 56. `$login = $this->access->login($username, $password);` 57. `if ($login)` 58. 59. `return TRUE;` 60. 61. `else` 62. 63. `$this->form_validation->set_message('check_login','Username atau password anda salah.');` 64. `return FALSE;` 65. 66.

Pada controller member kita akan melakukan pengecekan terhadap inputan yang telah di masukkan user. Oleh karena itu kita tetap menggunakan library validation. Kita akan menggunakan custom validation. Perhatikan baris 29 dan baris 50. Fungsi `check_login`

menjadi custom validator pada controller ini. Pada fungsi itu juga library access dipanggil untuk melakukan pengecekan login. Baris 63 digunakan untuk mengeset pesan error pada validasi. 6. Membuat view controller member

Setelah memiliki model dan controller maka kita tinggal membuat view

1. `if ($this->form_validation->run() == FALSE)` 2. 3. `< h1 > Sign On < /h1 >` 4. 5. `< p > Welcome. Please fill your username and password. < /p >` 6. `< ?php if ($validation->errors())` 7. `< div class="fail">` 8. `< ?php echo $validation->errors();` 9. `< /div >` 10. 11. 12. 13. `< form name="login" method="post" target="" < ?php site_url('member/login')? >` 14. `< p >` 15. 16. `< label > Username < /label >` 17. `< input name="username" id="username" value="" < ?php echo $v->username? >` 18. `< input type="text" />` 19. `< input type="password" id="password" width="100px" type="password" class="input large" />` 20. `< input type="password" class="input large" />` 21. `< input type="submit" value="Submit" class="button" />` 22. `< /form >` 23. `< /form >`

Adapun tampilan dari view diatas adalah

7. Membuat Mengextend library Controller

Controller member hanya digunakan memverifikasi user yang login. Kita akan membuat sebuah controller yang hanya bisa diakses oleh user yang sudah login. Controller tersebut akan diberi nama `Member_Controller`. Untuk membuat member controller maka kita har...

```

        $access->is_login(); 19.20.21.22.23.class MY_Controller extends CI_Controller 24.function construct() 25.26.parent

```

Class Member_Controller merupakan turunan dari kelas CI_Controller dengan penambahan fungsi pengecekan *extend* kelas bawaan Codeigniter kita harus membuat sebuah kelas MY_Controller. Kita dapat merubah prefix MY_ menjadi MY_

```

1. if(!defined('BASEPATH')) exit('No direct script access allowed');
2.

```

```

3. class Dashboard extends Member_Controller 4.5.function construct() 6.7.parent :: construct(); 8.9.10.function
$template->display('dashboard'); 13. 14. 15. 16.

```

Chapter 11 Kasus 5. Image Gallery Sederhana Sebuah image gallery merupakan aplikasi yang menarik untuk dikerjakan. Dengan menggunakan php biasa, image gallery tidak mudah untuk dibuat. Terutama bagian upload dan meresize gambar yang telah di upload. Dengan codeigniter pembuatan gallery menjadi lebih mudah dan cepat. Dengan memanfaatkan library upload, resize serta beberapa helper, kita dapat membuat sebuah gallery yang menarik. Library upload codeigniter mudah untuk digunakan. Cara pemanggilannya juga sama dengan library lainnya *this->load->library('upload');*

Untuk dapat menggunakan library ini kita harus mengkonfigurasi beberapa hal diantaranya dimana kita akan mengupload file tersebut, tipe dan ukuran file, dan lain-lain. Contohnya: *config['upload_path'] = APPPATH.'uploads/'; config['allowed_types'] = 'jpeg|jpg|gif|png'; config['max_size'] = '1024';*
this->upload->initialize(config);

Setelah men-setting library upload kita tinggal memanggil fungsi *do_upload()* untuk mengupload file. *file yang telah dimasukkan oleh user. if(!this->upload->do_upload())*
 else

```

//gagal mengupload file error menyimpan error message dalam //variabel
$error = array('error' => this->upload->display_errors());
//sukses mengupload file informasi file disimpan dalam // variabel data
$data = array('upload_data' => this->upload->data());

```

Image gallery yang akan kita buat memiliki fitur untuk mengupload file gambar, lalu meresize gambar tersebut. Setelah semua proses berhasil maka gallery akan menampilkan thumbnail dari image tersebut. Jika thumbnail tersebut di klik maka barulah gambar yang sebenarnya muncul dalam bentuk popup.

Adapun langkah-langkah yang dilakukan untuk membuat image gallery adalah:

- Membuat folder *upload_image* dan *thumb* di root aplikasi dan jangan lupa untuk mengubah permisidari folder Struktur direktori gallery

```

• Mengubah settingan Codeigniter. Settingan yang harus di ubah adalah
config['base_url'] di application/config/config.php, lalu sesuaikan dengan aplikasi.
Membuat controller gallery
?php if(!defined('BASEPATH')) exit('No direct script access allowed');
2.3.class Gallery extends CI_Controller
./upload_image/'; 13.config['allowed_types'] = 'gif|jpg|png'; 14.config['max_size'] =
1000'; 15.config['max_width'] = '2024'; 16.config['max_height'] = '1468'; 17.18.this-
->load->library('upload', config); 19.this->load->library('template'); 20. this->
load->model('gallery_model'); 21.this->load->helper(array('form','url')); 22. $data['message'] = ''
; 23.if(!this->upload->do_upload()) 24.if(isset($_POST['submit'])) 25.$data['message']
= this->upload->display_errors(); 26.27.else 28.29.$data['upload_data'] = this-
->upload->data(); 30. $data['message'] = 'Anda telah sukses mengupload gambar!!'; 31.32.config['resize'] =
array(33.'source_image' => $data['upload_data']['full_path'], 34.'new_image' =>
./thumb/, 35.'maintain_ratio' => true, 36.'width' => 160,

```

```

37. 'height' = $120 $38. ); 39. 40. $this->load->library('image_lib',config,$size); 41. if(!this-
$image_lib->resize()) 42. 43. $data['message'] = $this->$image_lib->display_errors(); 44. 45. 46. 47. $data['images
'] = $this->$gallery_model->fetch_image($FCPATH.'upload_image'); 49. $this-
$template->display('gallery',$data); 50.; 51. 52. 53. 54. / * End of file Gallery.php *
/55. / * Location : ./application/controllers/Gallery.php * /

```

Perhatikan baris 18-21, disana kita meload semua library dan helper yang dibutuhkan. Khusus untuk library upload kita menggunakan konfigurasi pada saat pemanggilan library `$this->load->library('upload',config)`; Perhatikan parameter kedua (variabel `config`). Variable tersebut berisi setting dimana file tersebut akan diupload, apa saja lain. Pada baris 23, kita memanggil fungsi untuk mengupload jika image sukses diupload maka kita akan melakukan 44). Untuk meresize image kita juga perlu memberikan parameter khusus pada saat loading library - `$this->load->library('image_lib',config,$size)`; pada konfigurasi itulah kita menentukan ukuran thumbnail yang akan diupload. `?php2.class Gallery_model extends CI_Model3.function construct()4.5.parent :: construct();6.7.8.function fetch,` 9. 10. `$this->load->helper('file');` 11. `return get_filenames($path);` 12. 13.

Perhatikan baris 10 dan 11 baris tersebut akan mengembalikan daftar nama file yang berada di sebuah folder • Membuat view gallery. View akan menampilkan dua komponen utama yaitu daftar gambar dan form yang akan digunakan untuk mengupload image. 1. `if($php echo $message;? > 2. < $divid = "gallery" >` 3. `<?php $atts = array(5. 'width' = $800', 6. 'height' = $600', 7. 'scrollbars' = $yes', 8. 'status' = $yes', 9. 'resizable' = $yes', 10. 'screenx' = $0', 11. 12.); 'screeny' = $0' 13. 14. 15. foreach($images as $image): 16. ? $17. 18. <div class="thumb" $19. <a href=$if($php echo $base_url().$upload_image/'.'image; ? $20. <?php echo $anchor_popup($base_url().$upload_image/'.'image 21. , 'img src=$base_url().$thumb/'.'image.'" alt=$" / $i', $atts); 22. ? > 23. < /a > < /div >` 24. `<?php 25.endforeach; 26. ? > 27. < /div > 28. < $divid = "upload" >` Silakan Memilih file yang akan diupload : 29. `<?php 30.echo $form_open_multipart('gallery');` 31. `echo $form_upload` 35. `< /div >`

Perhatikan baris ke empat. Disana kita men-setting `anchor_popup($link_popup)` yang akan digunakan untuk mengupload gambar tersebut. Jika semua instalasi yang telah anda Screenshoot Gallery Image

Untuk yang lebih avancenya anda dapat menggabungkan image berikut ini dengan menggunakan script gallery javascript lainnya atau mengkombinasikannya menggunakan database

Chapter 11 Kasus 6. Buku Tamu menggunakan CodeIgniter

Setelah mengetahui konsep dasar CodeIgniter mari kita mulai sebuah contoh penggunaan CodeIgniter. Pada contoh pertama ini akan ditunjukkan bagaimana menggunakan query-query dasar pada CodeIgniter. Kita akan membuat Buku Tamu. Selanjutnya ikutilah langkah-langkah berikut ini. Membuat Table Guestbook Pada contoh ini kita akan membuat sebuah aplikasi Buku Tamu. Untuk itu kita membutuhkan sebuah tabel dengan schema sebagai berikut CREATE TABLE guestbook (id bigint auto_increment PRIMARY KEY, nama varchar(50), email varchar(50), tanggal date

Konfigurasi Guestbook

Buka file `application/config/database.php`. Setting sesuai dengan konfigurasi mysql Anda. Isikan username, password dan nama database yang digunakan. Setelah itu buka juga file `application/config/autoload.php` ubahlah

variabel `autoload['libraries']` (kira – kirabaris41) menjadi

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

Kemudian buka melalui browser. Jika tidak terjadi kesalahan apapun berarti Anda sudah berhasil menyinget database dengan benar

Membuat Model Guestbook

Untuk mendapatkan data dari database maka kita harus memiliki model yang dapat mengambil data tersebut. Oleh karena itu Anda harus menam-
bahkan model berikut di `Application/models/guestbook_model.php`. Adapun isi dari file tersebut adalah

```
if (php) class Guestbook_model extends CI_Model
{
    public $table = 'record_count';
    function __construct() { parent::__construct(); }
    function get_data($start=NULL, $count=NULL) {
        $results=array();
        $this->db->from('guestbook');
        $this->db->select($table);
        $this->db->limit($count, $start);
        $this->db->limit($start);
        $query=$this->db->get($table);
        return $query->result_array();
        return FALSE;
    }
    function add($data) {
        $this->db->insert($table, $data);
    }
    function update($key, $value, $data) {
        $this->db->update($table, $data);
    }
    function delete($id) {
        $this->db->where('id', $id);
    }
}
```

Model diatas terdiri atas lima fungsi yaitu konstruktor, `get_data`, `add`, `update` dan `delete`. Fungsi `get_data` digunakan untuk mengambil data dari database.

Membuat Controller Dan View Pada controller ini kita akan memanggil library, helper dan model yang akan digunakan. Adapun library yang akan digunakan adalah `form_validation` dan `table`. Sedangkan helper yang akan digunakan adalah `url` dan `smiley`. Adapun

Selanjutnya adalah pembuatan controller

```
if (php) if (!defined('BASEPATH')) exit('No direct script access allowed');
class Guestbook extends CI_Controller {
    function __construct() { parent::__construct(); }
    $load->library('pagination');
    $this->load->library('form_validation');
    $this->load->helper('smiley');
    $this->load->model('guestbook_model');
    $this->load->model('guestbook_model');
    function show() {
        if ($this->input->post('nama')) {
            $data['nama'] = $this->input->post('nama', TRUE);
            $data['email'] = $this->input->post('email', TRUE);
            $data['komentar'] = $this->input->post('komentar', TRUE);
            $data['status'] = 'Guestbook sukses ditambahkan';
            $data['status'] = 'Guestbook gagal ditambahkan';
            $paging_uri = 2;
            if ($this->uri->segment($paging_uri)) {
                $start = $this->uri->segment($paging_uri);
            } else {
                $start = 0;
            }
            $limit_per_page = 10;
            $data['total_records'] = $this->guestbook->get_data($limit_per_page, $start);
            $config['base_url'] = site_url('guestbook');
            $config['total_rows'] = $this->guestbook->table->record_count;
            $config['per_page'] = $limit_per_page;
            $config['uri_segment'] = $paging_uri;
            $this->pagination->initialize($config);
            $data['page_links'] = $this->pagination->create_links();
            $image_array = get_lickable_smileys(base_url().'smileys/');
            $col_array = $this->table->make_columns($image_array, 20);
            $data['smiley_table'] = $this->table->generate($col_array);
            $this->load->view('guestbook', $data);
        }
    }
}
```

```

function index() this->show();
function validate_data()this->form_validation->set_rules('nama','Nama','required|min_length[5]|max_length[50]');
this->form_validation->set_rules('email','Email','required|valid_email|htmlspecialchars');this->form_validation->set_rules('komentar','Komentar','required|htmlspecialchars');
return(this->form_validation->run() == FALSE)?FALSE:TRUE;

```

Pada controller ini ada 3 fungsi utama yaitu:

- Kontruktor – fungsi ini akan kita gunakan sebagai tempat untuk me-load sumberdaya yang dibutuhkan. Seperti library dan helper. Adapun library yang akan digunakan adalah pagination, table, input (sudah autoloan). Sedangkan helper yang akan digunakan adalah url dan smiley.

Berikut ini adalah kode yang digunakan untuk memanggil library dan model tersebut.

```

this->load->library('pagination');this->load->library('form_validation');this->load->model('guestbook_model','guestbook');

```

Pada fungsi load model, parameter kedua itu digunakan untuk alias, jadi jika tanpa alias maka kita harus memanggil fungsi model dengan cara `this->guestbook_model`, *sedangkandenganalias* cukup `this->guestbook`. • Show – fungsi ini akan menampilkan dan memasukkan data buku tamu. Secara logis isi fungsi ini terbagi dua. Yang pertama adalah bagian untuk memasukkan data buku-tamu.

```

if($this->validate_data())$data['nama']=$this->input->post('nama',TRUE);$data['email']=$this->input->post('email',TRUE);$data['komentar']=$this->input->post('komentar',TRUE);$data['tanggal']=date('Y-m-d H:m:s'); $data['status']=0;if($this->guestbook->add($data))$data['status']='Guestbook sukses ditambahkan'; else $data['status']='Guestbookgagal ditambahkan';

```

Jika data sudah tervalidasi dengan benar maka artinya data sudah siap untuk dimasukkan. Maka kita akan menggunakan library input untuk mengambil data form lalu memanggil fungsi `this->guestbook->add($data)` untuk memasukkan data tersebut ke dalam database. Bagian kedua adalah bagian untuk menampilkan data buku tamu. `pagination->start=0;`

```

if($this->uri->segment($pagination->start=$this->uri->segment($pagination->start=0;
limit_per_page=10;
$data['total_rows']=$this->guestbook->get_data($limit_per_page,$start);
$config['base_url']=$site_url('guestbook');$config['total_rows']=$this->guestbook->table_row_count;$config['per_page']=$limit_per_page;$config['uri_segment']=$pagination->start;
$this->pagination->initialize($config);
$data['page_links']=$this->pagination->create_links();
$image_array=get_clickable_smileys($base_url().$smileys/);
$col_array=$this->table->make_columns($image_array,20);
$data['smiley_table']=$this->table->generate($col_array);
$this->load->view('guestbook',$data);

```

Jika kita akan menampilkan data dalam pagination perlu diingat dua hal, pertama kita membutuhkan inputan berupa berapa jumlah data yang akan ditampilkan dan data dimulai dari halaman berapa dan data yang dihasilkan dari model harus berisi jumlah total data yang kita punya. Untuk mendapatkan data yang akan diambil mulai dari data ke berapa, library pagination meletakkannya informasi tersebut di uri tertentu. Pada kasus ini kita meletakkannya di uri ke


```

2. Maka untuk mendapatkannya kita perlu memanggil fungsi this->uri->
segment(2); Sedangkan untuk jumlah data sudah tersimpan di property model (table_record_count). ffValidate_data
    this->form_validation->set_rules('nama','Nama','required|min_length[5]|max_length[12]'); this-
form_validation->set_rules('email','Email','required|valid_email|htmlspecialchars'); this-
form_validation->set_rules('komentar','Komentar','required|htmlspecialchars'); return(this-
form_validation->run() == FALSE)? FALSE : TRUE;

```

Selanjutnya adalah membuat view yang akan digunakan untuk menampilkan data buku tamu tersebut `<html> <head> <title>Buku Tamu</title> <style type="text/css">
body background-color: fff; margin: 40px; font-family: Lucida Grande, Verdana, Sans-serif; font-size: 14px; color: 4F5155; </style> </head> <body> <h2>Guest
Book </h2> <hr>`

```

    <?php echo validation_errors();? ><?php if(isset(status)):? <div class="success">
<span class="message_content"> Data Sukses Disimpan </span> </div> <
?php unset(v);endif;? <
    <? if(isset($tguestbook_list)) foreach($tguestbook_list as $value) echo "<div> <strong> <u>". $value['nama']. " <
/u> </strong> < (" . $value['tanggal']. " ) : " . nl2br(parse_url($value['komentar'],base_url()). "smileys/").
    ? <
    <?php echo page_links;? ><?php echo $insert_smiley('buku tamu','komentar');? ><
    <br> <h4> Isi Buku Tamu </h4> <form name="buku tamu" method="
    post"> <label for="nama"> Nama : </label> <input type="text" name="
    nama" value="" <?php echo set_value('nama');? > "/> <br> <label for="
    email"> Email : </label> <input type="text" name="email" value=""
    <?php echo set_value('email');? > "/> <br> <textarea name="komentar" cols="
    40" rows="4"> <?php echo set_value('komentar');? > </textarea> <div class="
    no-border">
    <?php echo $smiley_table;? > </div> <input type="Submit"/> </form> <
    </body> </html>

```

Pada view tersebut akan di tampilkan data isi buku tamu dalam list beserta form yang akan digunakan untuk menginputkan data. Kita dapat menggunakan validation helper (fungsi *validation_error*) untuk menampilkan error yang terjadi.

Chapter 13 Kasus 7. Membuat Shopping Cart Sederhana Jika anda ingin membangun toko online, salah satu fitur yang hampir pasti ada yaitu shopping cart (keranjang belanja). Membuat shopping cart tidaklah begitu sulit, terlebih jika anda menggunakan framework CodeIgniter. CodeIgniter telah menyediakan suatu library/pustaka untuk mempermudah anda membuat shopping cart. Library cart membutuhkan library session, tetapi kita tidak perlu meload library tersebut secara manual. Untuk menggunakannya anda bisa memanggil dengan library loader atau menambahkannya di konfigurasi autoload

```

this->load->library('cart');

Penggunaannya juga tidak sulit, untuk menambah item di cart kita bisa menggunakan fungsi inser seperti berikut ini
    data = array(
        );
    'id' => 'sku_123ABC', 'qty' => 1, 'price' => 39.95, 'name' => 'T-Shirt', 'options' =>
array('Size' => 'L', 'Color' => 'Red')
    this->cart->insert(data);

```

Untuk studi kasus ini, kita akan mengkombinasikannya dengan pemilihan produk dimana produk- produk tersebut telah disimpan didalam database. Adapun langkah-langkah yang dilakukan adalah 1. Membuat database produk

Ketika kita menggunakan shopping chart tentunya akan ada produk yang akan kita jual. Kita akan menggunakan produk yang sangat sederhana pada contoh kasus ini. Sebuah produk hanya memiliki Id, nama produk dan harga. CREATE TABLE 'products' ('id' int NOT NULL AUTO_INCREMENT, 'name' varchar(100) NOT NULL, 'price' int

```
INSERT INTO 'products' ('id', 'name', 'price') VALUES (1, 'Baju Batman', 30000), (2, 'Mouse Wireless', 434444), (3, 'Tas Ransel', 2344), (4, 'Kasur Busa', 3453);
```

2. Mengkonfigurasi Codeigniter

Kita harus mengkonfigurasi codeigniter terutama di bagian database karena kita akan menggunakan database pada aplikasi ini. Oleh karena itu kita harus mengubah file application /configs/database.php

```
db['default']['hostname'] = "localhost"; db['default']['username'] = "root"; db['default']['password'] = "root"; db['default']['database'] = "shop"; db['default']['dbdriver'] = "mysql";
```

Selain konfigurasi database kita juga mengset konfigurasi base_ *application/configs/config.php* config['base_url'] = "http://localhost/shopping_cart/";

3. Membuat Model Produk

Model product mempunyai 2 fungsi yaitu untuk mengambil seluruh data barang, dan fungsi untuk mengambil data barang tertentu sesuai dengan id yang diinginkan. 1. *index.php* 2. 3. class Product *model extends CI_Model* 4.5. *function construct()* 6.7. *parent :: construct()* 8. *if (\$this->db->conn_id == NULL, \$offset = NULL)* 11. *query = \$this->db->get('products', limit, offset);* 12. *return query->result();* 13. 14. 15. *function get(\$id)* 16. *query = \$this->db->get_where('products', array('id' => \$id));* 17. *return query->result();* 18. 19. 20.

Model ini hanya digunakan untuk menampilkan informasi produk yang dijual.

4. Membuat Controller Produk dan Cart

Setelah membuat model maka kita akan membuat dua buah controller yang akan handle penampilan produk dan keranjang belanja. Adapun controller pertama adalah controller produk. Controller ini akan menampilkan semua produk yang ada didalam database 1. *index.php* if (! defined('BASEPATH')) exit('No direct script access allowed'); 2. 3. class Produk *extends CI_Controller* 4.5. *function construct()* 6.7. *parent :: construct()* 8. *\$this->load->library('template');* 13. *\$this->load->model('product_model', 'product', true);* 14. *\$data['product_list'] = \$this->product->get_all();* 15. *\$this->template->display('product', \$data);* 16. 17. 18. 19. */* End of file Product.php */* 20. */* Location : ./application/controllers/Product.php */*

Kita dapat memilih produk-produk tadi, lalu memasukkannya kedalam keranjang belanja. Setelah itu maka kita harus mempunyai controller lain untuk handle keranjang belanja. Ada tiga fungsi utama pada kelas ini yaitu menampilkan, menambah serta merubah keranjang belanja. Untuk menghapus tinggal mengisikan jumlah barang yang dibeli sebanyak 0. Adapun isi dari controller tersebut adalah : 1. *index.php* if (! defined('BASEPATH')) exit('No direct script access allowed'); 2. 3. class Cart *extends CI_Controller* 4.5. *function construct()* 6.7. *parent :: construct()* 8. *\$this->load->model('product_model', 'product', true);* 9. *\$this->load->library('cart');* 10. *\$this->*

```

load- > library('template'); 11.12.13.functionadd(id) 14. product =this-
product->get(id); 15.16.data = array( 17. 'id' = product->id, 18.'qty' =>
1, 19.'price' =>product->price,
20. 'name' = product->name, 21.); 22.23.this->cart->insert(data); 24.redirect("cart"); 25.26.27.function
cart->update(P$OST); 30.redirect("cart"); 31.32.33.functionindex() 34.data['cart_list'] =this-
cart->contents(); 35. this->template->display('cart',data); 36. 37. 38.
39. 40. /* End of file Cart.php */ 41. /* Location: ./application/controllers/Cart.php
*/

```

Perhatikan baris 16-23, disana kita akan memasukkan data produk. Adapun field-field data product yang dapat dimasukan kedalam library cart adalah: •
id – Setiap produk harus memiliki Id yang unik antara satu dan yang lain.

- qty – Jumlah barang yang akan dibeli
- price – Harga dari produk
- name – Nama produk
- options – informasi tambahan mengenai produk yang ingin dibeli

Untuk memasukkan data cart anda dapat menggunakan fungsi insert - *this->cart->insert(data)*; data cart tersebut akan disimpan didalam session codeigniter. Jadi library cart ini membutuhkan depedensi library session. 5. Membuat View shopping cart

Part terakhir adalah membuat view. Anda harus menyediakan dua buah view yaitu view list barang dan view daftar keranjang belanja. Adapun ini view dari daftar produk adalah 1. `<h1>Daftar barang</h1>` 2. `<?php if (!empty($product_list)) :>` 3. `` 4. `<?php foreach($product_list as $product):>` 5. `` 6. `<?php echo $product->name?>` (`<?php echo $product->price?>`)-
7. `` `<?php echo $product->name?>` 8. `` 9. `<?php endforeach?>` 10. `` 11. `<?php else :>` 12. `<p>` Produk kosong. `</p>` 13. `<?php endif?>`

Adapun tampilan dari view diatas adalah sebagai berikut

Ketika user mengklik link beli pada list produk maka user akan di redirect ke halaman add item ke shopping cart. Adapun view yang digunakan untuk melihat daftar shopping cart adalah 1. `<h1>Shopping cart anda</h1>` 2. `<?php echo form_open('cart/update');>` 3.4. `<table cellpadding = "6" cellspacing = "1" style = "width : 100%;>` 5. `<tr>` 6. `<th>` QTY 7. `</th>` 8. `<th>` ItemDescription 9. `</th>` 10. `<th style = "text-align : right">` ItemPrice 11. `</th>` 12.13. `<?php $i = 1;>` 14. 15. `<?php foreach($this->cart->contents() as $items):>` 16. 17. `<?php echo form_hidden($i['rowid'], $items['rowid']);>` 18.19. `<tr>` 20. `<td>` `<?php echo form_input(array('name' =>$i['qty'], 'value' =>$items['qty'], 'maxlength' =>'3', 'size' =>'5'));>` 21. `</td>` 22. `<td>` 23. `<?php echo $items['name'];>` 24. 25. `<?php if ($this->cart->has_options($items['rowid']) == TRUE):>` 26. 27. `<p>`

28. `<?php foreach ($this->cart->product_options($items['rowid']) as $option_name =>$option_value):>` 30.31. `` `<?php echo $option_name;>` 32. `<?php echo $option_value;>` 33.34. `<?php endforeach;>` 35. `</p>` 36.37. `<?php endif;>` 38.39. `</td>` 40. `<td style = "text-align : right">` 41. `<?php echo $this->cart->format_number($items['price']);>` `</td>`

```

42. <td style="text-align:right">43. <?php echo this->cart->format_number(items['subtotal']);
?></td> 44. </tr> 45. 46. <?php $i++;? > 47.48. <?php endforeach;? >
49.50. <tr> 51. <td colspan="2"></td> 52. <td class="right"><
strong>Total </strong></td> 53. <td class="right"> 54. <?php echo
this->cart->format_number(this->cart->total());?></td> 55. </tr> 56. 57.
</table> 58. 59. <?php echo form_submit('','Update your Cart');? ></p>
60.61. <a href=" <?php echo site_url('produk')? >" > Kembali </a>

```

Perhatikan baris 15, fungsi `this->cart->contents()` digunakan untuk mendapatkan seluruh data cart yang `cart->total()`;

Adapun tampilan dari view diatas adalah gambar berikut ini.

Pada gambar diatas kita dapat mengubah jumlah barang yang dipesan. Subtotal dan total akan otomatis ditambahkan oleh library cart.

Chapter 12 Kasus 8. CodeIgniter dan Ajax

AJAX yang dimaksud disini bukanlah nama club sepakbola yang berasal dari Amsterdam, Anda atau pun nama pahlawan dalam sejarah perang Trojan, tetapi AJAX di sini adalah singkatan dari Asynchronous JavaScript and XML. Pada intinya ajax itu merupakan gabungan beberapa teknologi yang bertujuan untuk menghindari page reload. Dengan menghindari page reload, kita dapat menghindari paradigma click-and-wait serta memberikan sebuah fitur yang cukup kompleks pada website seperti validasi data secara realtime, drag n drop dan fitur-fitur lain yang belum dimiliki web biasa. Dengan AJAX, suatu aplikasi web dapat mengambil data kemudian diolah di client melalui request asynchronous HTTP yang diinisialisasi oleh Javascript, sehingga dapat mengupdate bagian-bagian tertentu dari web tanpa harus memanggil keseluruhan halaman web. Request ini dapat dieksekusi dalam beberapa cara dan beberapa format transmisi data. Dikombinasikannya cara pengambilan data remote dengan interaktivitas dari Document Object Model (DOM) telah menghasilkan generasi terbaru dari aplikasi web yang menggebrak aturan-aturan tradisional tentang apa yang dapat terjadi di dalam web. Keuntungan dari aplikasi web berbasis AJAX adalah memungkinkan untuk membuat website dan aplikasi web yang lebih baik dan lebih responsif. Sehingga meningkatkan kemudahan pengguna. CodeIgniter sebenarnya tidak terpengaruh dengan teknik ajax ini karena ajax bekerja di sisi client sedangkan CI bekerja di sisi server. Yang perlu dipersiapkan hanya di sisi template dan view. Untuk bagian library template kita melakukan perubahan dengan menambahkan pengecekan apakah sebuah request tersebut merupakan ajax request apa bukan. `application/libraries/template.php` 1. `if` 2. `class` `Template` 3. `protected` `$i`; 4.5.`function construct()` 6.7.`this->$i = get_instance();` 8.9.10.`function display(template, null)` 11.12.`if (!this->is_ajax())` 13.14.`$data['content'] = this->$i->load->view(template,` 15. `$data, true)`; 16.`$data['header'] = this->$i->load->view('template/header',` 17.`$data,` `true)`;

```

18. $data['op_menu'] = this->$i->load->view('template/menu', 19.$data, true); 20. $data['right_menu'] = this->$i->load->view(21.'template/sidebar', $data, true); 22. this->$i->load->view('/template.php', $data); 23. 24. else 25.
26. this->$i->load->view(template, $data); 27.28.29.30.function is_ajax() 31.32.return (33.this->$i->input->server('HTTP_X_REQUESTED_WITH') 34. (this->$i->input->server('HTTP_X_REQUESTED_WITH') == 35.'XMLHttpRequest'); 36.37.

```

Perhatikan fungsi `is_ajax`, fungsi tersebut untuk mengecek apakah request tersebut merupakan sebuah request

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" 2.
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> 3. <html xmlns="http://www.w3.org/1999/xhtml"
4. <head> 5. <meta http-equiv="content-type" content="text/html; charset=utf-
8"> 6. <link rel="stylesheet" type="text/css" href=" 7. <?php echo base_url();? >
public/style.css"/> 8. <script src=" <?php echo base_url();? > public/js/jquery-
1.4.2.min.js"> 9. </script> 10. <script> 11.(document).ready(function()
12. (<'.ajax').click(function(e)13.e.preventDefault(); 14..get((this).attr('href'), function(Res)15.('content').
16. ); 17. ) 18. ) 19. </script> 20. <title>Sistem Template CodeIgniter
Template</title> 21. </head> 22. <body> 23. 24. <div id="wrap"> 25. <div
id="header"> 26. <!-- 27. Area Header
28. --> 29. <?php echo header;? > 30. </div> 31.32. <div id="
menu"> 33. <!-- 34.Area Menu35. --> 36. <?php echo top_menu;? >
37. </div> 38.39. <div id="contentwrap"> 40. <div id="content">
41. <!-- 42.Area content43. --> 44. <?php echo content;? > 45. </div>
46. <div id="sidebar"> 47. <!-- 48.Area Right Menu49. --> 50. <
?php echo right_menu;? > 51. </div> 52. <div style="clear: both;">
</div> 53. </div> 54. <div id="footer"> 55. <p> Copyrightcopy; <
ahref=""> You </a> | Designby56. <ahref="http://www.readcrazyreviews.com">
ReadCrazyReviews </a> </p> 57. </div> 58.59. </div> 60.61. <
/body> 62. </html>
```

Perhatikan baris 8-18, pada kode tersebut kita menggunakan jquery untuk mempermudah melakukan request ajax. Pada kode di atas kita akan menyari semua link yang memiliki kelas `ajax` lalu meloadnya melalui ajax (lihat view `template/menu.php`). `application/view/template/menu.php`

```
1. <ul> 2. <li class="ajax" href="<?php echo base_url();? >"> Home <
/a> </li> 3. <li class="ajax" href=" <?php echo site_url('welcome/page1');? >
"> 4.PageAjax1 </a> </li> 5. <li class="ajax" href=" <
?php echo site_url('welcome/page2');? > "> 6.PageAjax2 </a> </li> 7. <
li class="ajax" href=" <?php echo site_url('welcome/page3');? > ">
8.PageAjax3 </a> </li> 9. <li class="ajax" href=" <
?php echo site_url('welcome/page4');? > "> 10.PageAjax4 </a> </li>
11. </ul>
```

Untuk bagian controller tidak melakukan perubahan apa-apa. kita hanya menambahkan page yang dapat dipanggil

```
1. <?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Welcome extends CI_Controller4.5.functionconstruct()6.7.parent::construct();8.this->load->library('template');9. this->load->helper('url');10.11.12.functionindex()13.14.this->template->display('welcome_message');15.16.17.functionpage1()18.19.this->template->display('page1');20. 21.22.functionpage2()23. 24. this->template->display('page2');25.26.27.functionpage3()28.29.this->template->display('page3');30. 31. 32.functionpage4()33. 34. this->template->display('page4');35.36.37.38./* End of file welcome.php */39./* Location : ./application/controllers/welcome.php */
```

Ketika kita mengakses page 1 secara langsung maka library template akan menampilkan page secara utuh tetapi melalui ajax hanya akan memberikan

area content.

8 Sublime text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai platform operating system dengan menggunakan teknologi Python API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerful. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages. sejarah sublime text 3 Sublime Text merupakan text editor berbagai bahasa pemrograman mulai dari C hingga Java, menurut penulis Sublime Text merupakan text editor yang sangat baik dan nyaman untuk digunakan sebagai alat berkoding ria. Karena tampilannya yang minimalis namun elegant, bawaannya yang ringan namun diboboti dengan keakuratan dalam penulisan yang mengikuti algoritma dan aturan setiap bahasa pemrograman yang digunakan. Intinya penulis sangat merekomendasikan untuk menjadikan Sublime Text sebagai text editor yang meski digunakan.

Kini pada versi terbaru Sublime Text dengan paket dependasi *.deb sudah dapat dijalankan pada distro linux Debian dan turunannya seperti Ubuntu.

Pada update kali ini, Sublime Text mencegah plugin merusak Sublime Text, dan mempercepat pembukaan aplikasi di awal. Selain peningkatan kecepatan, sekarang Sublime Text juga menggunakan Python 3.3 untuk setiap plugin, serta memperkenalkan symbol indexing yang membaca semua file dalam sebuah proyek untuk dibuat indeks simbol pada saat pencarian.

Pengembang Sublime Text, Jon Skinner menjelaskan bahwa lisensi Sublime Text 3 akan dijual dengan harga 70, *bertambah11* dari harga sebelumnya yang hanya 59, *walaupun dapat menggunakan setiap fitur Sublime Text secara lengkap tanpa memerlukan lisensi*

Simbol Indexing. Teks Sublime sekarang memindai file dalam proyek Anda, dan membangun sebuah indeks yang berisi file yang simbol. Ini fitur baru Goto Definisi dan Goto Simbol di Proyek, yang keduanya tersedia dari menu Goto. Goto Definition membawa Anda ke definisi simbol di bawah tanda sisipan, sementara Goto Simbol di Proyek meminta Anda untuk memilih simbol melalui pencocokan fuzzy, dan kemudian membawa Anda ke definisi simbol itu.

Manajemen Pane. Bekerja dengan beberapa panel sekarang lebih efisien, dengan perintah untuk menciptakan dan menghancurkan panel, dan cepat memindahkan file antara panel. Anda dapat melihat pilihan baru di bawah View / Grup, View / Focus Group dan Lihat / Move file Group.

Kecepatan. Teks Sublime selalu memiliki kecepatan sebagai fitur, namun versi 3 alamat beberapa titik lemah. Waktu startup sekarang hampir segera, dan plugin tidak lagi memiliki kesempatan untuk membawa ini ke bawah. Ganti Semua kinerja juga secara signifikan lebih cepat.

Teks Sublime sekarang menggunakan Python 3.3 untuk plugin, dan berjalan mereka keluar dari proses, sehingga setiap plugin yang memuat kode asli ada risiko lagi menerjang proses Teks Sublime utama. API ini juga sepenuhnya benang-aman, dan menyediakan beberapa callback yang berjalan asyn-

chronously (misalnya, *on_modified_async*). Ada juga fungsi API baru, termasuk akses penuh ke data proyek. Teks

Perubahan yang dipilih: Ditambahkan Goto Definisi, dan Goto Simbol di Proyek Secara signifikan meningkatkan waktu startup Secara signifikan meningkatkan kinerja semua Peningkatan algoritma pencocokan digunakan untuk Goto Apa saja dan auto-complete menerima karakter dialihkan UI: Peningkatan manajemen pane UI: Meninjau file dari sidebar menciptakan tab pratinjau UI: Peningkatan animasi di side bar Proyek: Beberapa ruang kerja dapat dibuat untuk satu proyek Proyek: Ketika menambahkan folder ke sidebar, symlink tidak diikuti secara default. Hal ini dapat diubah dengan memungkinkan *follow_symlinks* dalam proyek *Membangun Sistem*:
Ditambahkan 'shell_cmd', yang menggantikan 'cmd', dengan sintaks yang lebih intuitif *Membangun Sistem*:
PATH penanganan perilaku yang lebih baik pada OSX saat menggunakan shell_cmd *Membangun Sistem*:
'Membuat' membangun sistem memiliki perbaikan pesan error regex *Membangun Sistem*:
Sintaks file dapat ditentukan untuk output *Membangun Sistem*: *Wordwrap diaktifkan secara default* *Temukan*
Peningkatan penanganan file biner *Temukan di File*: *nomor baris yang tersembunyi dalam output* *Cari*:
Caridi *Seleksi tidak akan lagi secara otomatis dipilih* *OSX*: *Peningkatan kinerja pada Retina display* *OSX*:
10.7 atau yang lebih baru diperlukan *OSX*: *64 bit hanya* *OSX*: *versi Sistem Python tidak lagi ketergantungan* *OSX*:
Italic font disintesis ketika tidak tersedia di tipografi Linux: *File Deb disediakan* *Linux*:
Mula dari baris perintah daemonize proses secara default *API*: *Upgraded dari Python 2.6 ke Python 3.3* *API*:
Plugin menjalankan out-of-process *API*: *Plugin API sekarang thread aman* *API*:
Beberapa peristiwa API sekarang menjalankan asynchronous *API*: *begin_edit()/end_edit()* *tidak lagi dapat dial*
Proyek yang terkena API *API*: *Ditambahkan window.settings() dan window.template_settings()* *API*:
show_quick_panel() *menerima sebuah panggilan balik* *highlighted*

Sublime Text 3 harus bekerja pada semua sistem yang saat ini dijalankan Sublime Text 2, dengan pengecualian OS X 10.6. Sayangnya, hal itu belum mungkin untuk mempertahankan 10,6 kompatibilitas, karena tidak ada C++ 11 toolchain untuk OS X 10.6, dan Sublime Text 3 membuat ekstensif menggunakan fungsi di C++ 11.

Harga untuk kunci lisensi Teks Sublime telah meningkat sebesar 11, dari 59 sampai 70, *kenaikan harga pertama dalam sejarah Sublime Text lima tahun. Semua lisensi dibelikan dengan harga baru* 30, atau 15 untuk pengguna yang telah membeli baru-baru ini. *Memotong Tanggal untuk harga upgrade berkurang*

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 2 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.

Kelebihan Sublime Text 3 :

Multi Platform Kelebihan pertama dari Sublime Text adalah software ini tersedia dalam berbagai platform sistem operasi, antara lain Windows, Linux dan MacOS. Plugin Plugin-nya sangat beragam, sehingga bisa memudahkan

programmer dalam mengembangkan software-nya. 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.

Tema dan color scheme yang bervariasi Anda bahkan bisa membuatnya sendiri atau mendownloadnya secara terpisah. Di Sublime Text, kita bisa mengganti Color Scheme (untuk area editor) maupun Theme (untuk interface sidebar, tab, console, search dll). Untuk mengedit color scheme, kita harus membuka file xml berekstensi .tmTheme dan mengedit isinya, atau bisa juga menggunakan tmTheme editor. *new_theme_largeGoToAnythingDenganGotoAnything, kitabisa membuka file dalam project dan kemudian ketikkan nama filenya. Untuk mencari nama file tidak harus mengetik secara tepat, karena Sublime Text perintahnya bisa kita akses menggunakan Shortcut, atau kalau belum hafal shortcut kita bisa mengakses Command Palette (Ctrl+P) kemudian cari perintah yang kita inginkan (menggunakan fuzzy search). Sebagai contoh, saya ingin menambahkan package baru.*

Baik Notepad++ maupun Sublime Text memiliki fitur ini. Di Notepad++ tekan Alt kemudian mouse didrag ke bagian yang ingin diedit. Di Sublime Text menggunakan drag tombol mouse tengah, atau bisa juga dengan Shift + drag tombol mouse kanan.

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. Mini map / document map Sublime Text memiliki Minimap, semacam versi mini dari file untuk mempermudah melihat file secara keseluruhan. Go To 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). Notepad++ belum memiliki fitur ini. Fitur semacam ini sangat jarang di text editor biasa, kebanyakan ada di IDE kelas berat seperti NetBeans, Zend Studio, dll. Kelemahan :

Ada beberapa plugin yang terdapat pada notepad ++ yang belum ada di sublime text. Sekian postingan saya kali ini, semoga bermanfaat. Bagikan ini: Twitter Facebook

9 Sejarah xampp

Sejarah XAMPP XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan project non-profit yang

dikembangkan oleh Apache Friends yang didirikan Kai ‘Oswalad’ Seidler dan Kay Vogelgesang pada tahun 2002, project mereka ini bertujuan mempromosikan penggunaan Apache web server.

1. Fungsi XAMPP Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl.
2. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Untuk mendapatkannya XAMPP anda dapat mendownload langsung dari web resminya. Dan berikut beberapa definisi program lainnya yang terdapat dalam XAMPP.
3. Fitur-Fitur XAMPP Berikut ini terdapat beberapa fitur-fitur xampp, terdiri atas:
4. Apache Apache adalah perangkat lunak sumber terbuka yang menjadi alternatif dari server web Netscape. Server HTTP Apache atau Server Web/WWW Apache merupakan server web yang dapat dijalankan di banyak sistem operasi yang berguna untuk melayani dan memfungsikan situs web. Apache dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.
Baca Juga Artikel yang Mungkin Terkait : Uji Validitas adalah
5. MySQL MySQL adalah singkatan “My Structured Query Language”. Program ini berjalan sebagai server menyediakan multi-user mengakses ke sejumlah database. MySQL umumnya digunakan oleh perangkat lunak bebas yang memerlukan fitur penuh sistem manajemen database, seperti WordPress, phpBB dan perangkat lunak lain yang dibangun pada perangkat lunak LAMP. Ia juga digunakan dalam skala sangat tinggi World Wide Web, termasuk produk-produk Google dan Facebook.
6. PHP PHP adalah bahasa pemrograman script yang banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (software di belakang Wikipedia). Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (software di belakang Wikipedia).
7. Bagian-Bagian XAMPP Berikut ini terdapat beberapa bagian-bagian XAMPP, terdiri atas:
Htdocs Htdocs

8. Htdocs adalah sebuah folder yang digunakan sebagai tempat penyimpanan berkas seperti PHP, HTML, dan script lain yang digunakan dalam sebuah halaman website. Secara kapasitas penyimpanan, XAMPP tergantung dari seberapa besar kapasitas hardisk di laptop atau komputer anda. Sedangkan bila menggunakan hosting online, maka tergantung pilihan waktu membeli sebuah hosting.

phpMyadmin phpMyadmin

9. phpMyadmin adalah sebuah tempat yang digunakan untuk mengelola database MySQL yang berada di komputer atau laptop. Untuk mengakses phpMyadmin yakni dengan membuka browser internet (Mozilla atau chrome) lalu ketikkan alamat <http://localhost/phpMyadmin> maka akan muncul tampilannya.

Baca Juga Artikel yang Mungkin Terkait : “Open Source” Pengertian Contoh (Kelebihan – Kekurangan)

Control Panel control-panel

Control Panel adalah sebuah layanan untuk mengelola XAMPP baik itu mengontrol (start atau stop XAMPP) serta layanan service lainnya. Secara online di dalam hosting atau VPS dikenal CPanel

Komponen XAMPP Berikut ini terdapat beberapa komponen xampp, terdiri atas:

XAMPP 1.8.3 untuk Windows, Termasuk : Apache 2.4.4 MySQL 6.5.11 PHP 5.5.0 phpMyAdmin 4.0.4 FileZilla FTP Server 0.9.41 Tomcat 7.0.41 (with modproxyajp as connector) Strawberry Perl 5.16.3.1 Portabel XAMPP Control Panel 3.2.1 (dari hackattack142) XAMPP 1.8.3 untuk Linux, Termasuk : Apache 2.4.4 MySQL 6.5.11 PHP 5.5.0 phpMyAdmin 4.0.4 OpenSSL 1.0.1e XAMPP untuk Solaris XAMPP untuk Mac OS X Kelebihan dan Kekurangan XAMPP Berikut ini terdapat beberapa kelebihan dan kekurangan XAMPP, terdiri atas:

Kelebihan : Database Storage Engine ini banyak digunakan oleh programmer apalagi oleh web developer karena sifatnya yang free. Untuk yang expert sudah ada yang bayar. Kemampuannya sudah bisa diandalkan, mempunyai kapasitas yang cukup mumpuni sekitar 60.000 tabel dengan jumlah record mencapai 5.000.000.000 bahkan untuk yang terbaru sudah lebih. Keamanan datanya cukup aman walaupun tidak sekuat PostgreSQL apalagi Oracle. Engine ini multiplatform sehingga mampu diaplikasikan di berbagai sistem operasi. MySQL cocok diaplikasikan diaplikasi kelas kecil dan menengah. Kelebihan paling utama engine ini adalah kecepatannya. Kekurangan : Tidak cocok untuk menangani data dengan jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data. Memiliki keterbatasan kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung server karena tidak menerapkan konsep Technology Clusters. Baca Juga Artikel yang Mungkin Terkait : “PHP” Pengertian – Jenis (Kelebihan – Kekurangan)

XAMPP adalah singkatan yang setiap huruf adalah :

X : Program ini dapat dijalankan di banyak sistem operasi, seperti Windows, Linux, Mac OS, dan Solaris. A : Apache, server aplikasi Web. Apache tugas utama adalah untuk menghasilkan halaman web yang benar kepada pengguna terhadap kode PHP yang sudah dituliskan oleh pembuat halaman web. jika perlu kode PHP juga berdasarkan yang tertulis, dapat database diakses dulu (misalnya MySQL) untuk mendukung halaman web yang dihasilkan. M : MySQL, server aplikasi database. Pertumbuhannya disebut SQL singkatan dari Structured Query Language. SQL merupakan bahasa terstruktur yang difungsikan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database dan isinya. Bisa juga memanfaatkan MySQL guna untuk menambahkan, mengubah, dan menghapus data dalam database. P : PHP, bahasa pemrograman web. Bahasa pemrograman PHP adalah bahasa pemrograman untuk membuat web yang server-side scripting. PHP digunakan untuk membuat halaman web dinamis. Sistem manajemen database yang sering digunakan dengan PHP adalah MySQL. namun PHP juga mendukung Pengelolaan sistem database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya. P : Perl, bahasa pemrograman untuk semua tujuan, pertama kali dikembangkan oleh Larry Wall, mesin Unix. Perl dirilis pertama kali tanggal 18 Desember 1987 yang ditandai dengan keluarnya Perl 1. Pada versi-versi selanjutnya, Perl juga tersedia untuk berbagai sistem operasi Unix (SunOS, Linux, BSD, HP-UX), juga tersedia untuk sistem operasi seperti DOS, Windows, PowerPC, BeOS, VMS, EBCDIC, dan PocketPC.

Cara install Xampp Cara Instal Xampp Di Komputer Windows OS DIPERBARUI: 4 OKTOBER 2019 · BY AAN ROFIK

cara install xampp, cara install xampp di windows

Bagi Anda yang ingin membuat website namun belum memiliki domain dan hosting sendiri, jangan berkecil hati dulu. Karena masih ada alternatif cara lain.

Apa itu?

Ya, kita bisa tetap membuat website di PC komputer / laptop, dengan menjadikan PC kita seolah-olah seperti server hosting.

Langkah ini juga sering digunakan bagi oleh para web developer yang sudah mahir sekalipun, untuk membuat website di komputer kerja mereka, sebelum mereka mengupload hasil kerjanya ke server hosting.

Bagaimana caranya?

Caranya adalah dengan menginstal aplikasi server localhost di PC kita.

Salah satu aplikasi server localhost serta yang paling banyak digunakan dan cukup familiar di kalangan web developer saat ini adalah XAMPP.

Aplikasi XAMPP ini dibuat oleh Apache Friends dan installer-nya bisa langsung diunduh dari situs mereka. Isi aplikasinya juga sudah sangat komplit, antara lain:

Apache MySQL PHP phpMyAdmin FileZilla FTP Server Tomcat XAMPP Control Panel Untuk itu, kali ini saya akan menuliskan tentang panduan cara instal XAMPP di komputer.

Berhubung saya pakainya Windows, maka panduan ini juga diperuntukkan bagi pengguna PC Windows ya. Namun kurang lebih untuk sistem operasi lain,

caranya hampir sama lah.

Oke, let's go...

Cara Instal XAMPP Di Windows Download aplikasi XAMPP terbaru, klik disini. Pilih salah satu (tips: jika Anda bingung, bisa pilih yang versi 5.6 dulu). cara install xampp Dobel klik file XAMPP yang baru saja Anda download, nanti selanjutnya akan muncul jendela "installer language" seperti di bawah ini: instal xampp, xampp localhost Selanjutnya pilih bahasa. Pilih yang Bahasa Inggris (English). Klik OK. Kadang pada proses ini muncul pesan error. Jika ada, abaikan saja dan lanjutkan dengan klik OK dan YES. 2-instal-xampp 3-instal-xamp Berikutnya akan muncul jendela yang isinya meminta Anda menutup semua aplikasi yang sedang berjalan. Jika semua aplikasi sudah ditutup, maka klik tombol Next. 4-instal-xampp Selanjutnya Anda akan diminta untuk memilih aplikasi yang mau diinstal. Centang saja semua pilihan dan klik tombol Next. 5-instal-xampp Kemudian Anda akan diminta untuk menentukan lokasi folder penyimpanan file-file dan folder XAMPP. Secara default akan diarahkan ke lokasi c:. Namun jika Anda ingin menyimpannya di folder lain bisa klik browse dan tentukan secara manual folder yang ingin digunakan. Jika sudah selesai, lanjutkan dan klik tombol Install. 6-instal-xampp Tunggu beberapa menit hingga proses instalasi selesai. Jika sudah muncul jendela seperti di bawah ini, klik tombol Finish untuk menyelesaikannya. 7-instal-xampp Berikutnya, akan muncul jendela dialog seperti gambar di bawah ini yang menanyakan Anda apakah mau langsung menjalankan aplikasi XAMPP atau tidak. Jika ya, maka klik YES. 8-instal-xampp Cara Menjalankan Aplikasi XAMPP Bukalah aplikasi XAMPP, bisa melalui Start Menu atau Desktop, dan klik icon XAMPP. Atau, jika Anda membukanya begitu proses instalasi selesai maka klik Yes seperti yang terlihat pada gambar di atas. Setelah terbuka, silahkan klik tombol Start pada kolom Action sehingga tombol tersebut berubah menjadi Stop. Dengan mengklik tombol tersebut, artinya itulah aplikasi yang dijalankan. Biasanya jika saya menggunakan XAMPP, yang saya start hanyalah aplikasi Apache dan MySQL, karena saya tidak memerlukan aplikasi seperti Filezilla, dan lain-lain. 9-instal-xampp 10-instal-xampp Sekarang bukalah browser kesukaan Anda, dan coba ketikkan <http://localhost/xampp> di address bar. Jika muncul tampilan seperti gambar di bawah ini, instalasi telah berhasil. 11-instal-xampp Selamat, akhirnya Anda berhasil menginstal aplikasi XAMPP. Sekarang komputer Anda sudah berfungsi seperti server dan bisa menjalankan aplikasi-aplikasi berbasis web.

Windows Pengertian Windows Beserta Fungsi dan Sejarah Windows (Mulai dari Tahun 1985) Zakaria September 5, 2019 Saat menggunakan komputer, pastilah terlebih dahulu kita harus menginstall operating system untuk komputer tersebut.

Operating System (OS) atau sistem operasi merupakan perangkat lunak yang dibutuhkan agar sebuah komputer dapat berfungsi dengan baik. OS menghubungkan antara perangkat keras dengan berbagai perangkat yang terdapat pada komputer sehingga dapat saling terintegrasi.

Salah satu OS berbayar yang telah digunakan secara luas adalah Windows. Dari masa ke masa, Windows terus mengalami perkembangan dan masih meru-

pakan salah satu OS yang paling banyak digunakan. Dalam artikel ini, akan dijelaskan mengenai pengertian Windows, sejarah windows, dan fungsinya.

Daftar Isi Pengertian Windows Pengertian Windows

Windows merupakan sebuah sistem operasi yang diciptakan oleh Microsoft, dimana sistem operasi ini menyediakan antarmuka grafis (GUI / Graphical User Interface) agar lebih mudah dioperasikan.

Dengan adanya Windows, pengguna tidak perlu lagi mengetikkan perintah melalui command line layaknya pada MS-DOS. Cukup dengan menggunakan mouse atau keyboard, pengguna dapat memberikan perintah untuk membuka menu, kotak dialog, menjalankan aplikasi, menghapus file dan lain sebagainya.

Fungsi Windows Berikut merupakan beberapa fungsi dari sistem operasi Windows:

Menghubungkan antara aplikasi dan perangkat keras, sehingga dapat terintegrasi bekerja secara konsisten dan stabil. Mengendalikan dan mengelola sumber daya yang sedang dijalankan, termasuk perangkat lunak dan perangkat keras pada komputer. Mengelola proses yang terdiri dari persiapan, penjadwalan, serta pemantauan program yang sedang dijalankan. Mengelola data input dan output serta mengendalikannya. indows 1.0 Versi pertama Microsoft Windows, yang disebut dengan Windows 1.0, dirilis pada tanggal 20 November 1985. Versi ini memiliki banyak kekurangan dalam beberapa fungsionalitas, sehingga kurang populer di pasaran. Pada awalnya Windows versi 1.0 ini hendak dinamakan dengan Interface Manager, akan tetapi Rowland Hanson, kepala bagian pemasaran di Microsoft Corporation, meyakinkan para petinggi Microsoft bahwa nama "Windows" akan lebih "memikat" konsumen. Windows 1.0 bukanlah sebuah sistem operasi yang lengkap, tetapi hanya memperluas kemampuan MS-DOS dengan tambahan antarmuka grafis. Selain itu, Windows 1.0 juga memiliki masalah dan kelemahan yang sama yang dimiliki oleh MS-DOS.

Lebih jauh lagi, Apple yang menuntut Microsoft membuat Microsoft membatasi kemampuannya. Sebagai contoh, jendela-jendela di dalam Windows 1.0 hanya dapat ditampilkan di layar secara "tile" saja, sehingga jendela tersebut tidak dapat saling menimpa satu sama lainnya. Selain itu, tidak ada semacam tempat yang digunakan untuk menyimpan berkas sebelum dihapus (Recycle Bin, pada MS-DOS ada perintah UNDELETE yang dapat memulihkan file yang terhapus), karena memang Apple berkeyakinan bahwa mereka memiliki hak terhadap paradigma tersebut. Microsoft pun kemudian membuang limitasi tersebut dari Windows dengan menandatangani perjanjian lisensi dengan Apple. itulah windows sekarang yang selalu ada upgrade untuk meningkatkan versi

Windows 2.x Windows versi 2 pun muncul kemudian pada tanggal 9 Desember 1987, dan menjadi sedikit lebih populer dibandingkan dengan pendahulunya. Sebagian besar popularitasnya didapat karena kedekatannya dengan aplikasi grafis buatan Microsoft, Microsoft Excel for Windows dan Microsoft Word for Windows. Aplikasi-aplikasi Windows dapat dijalankan dari MS-DOS, untuk kemudian memasuki Windows untuk melakukan operasinya, dan akan keluar dengan sendirinya saat aplikasi tersebut ditutup.

Microsoft Windows akhirnya memperoleh peningkatan signifikan saat Aldus

PageMaker muncul dalam versi untuk Windows, yang sebelumnya hanya dapat berjalan di atas Macintosh. Beberapa ahli sejarahwan komputer mencatat ini sebagai kemunculan sebuah aplikasi yang laku secara signifikan selain buatan Microsoft sebagai awal kesuksesan Microsoft Windows.

Windows versi 2.0x menggunakan model memori modus real, yang hanya mampu mengakses memori hingga 1 megabita saja. Dalam konfigurasi seperti itu, Windows dapat menjalankan aplikasi multitasking lainnya, semacam DESQview, yang berjalan dalam modus terproteksi yang ditawarkan oleh Intel 80286.

Windows 2.1x Selanjutnya, dua versi yang baru dirilis, yakni Windows/286 2.1 dan Windows/386 2.1. Seperti halnya versi Windows sebelumnya, Windows/286 menggunakan model memori modus real, tetapi merupakan versi yang pertama yang mendukung High Memory Area (HMA). Windows/386 2.1 bahkan memiliki kernel yang berjalan dalam modus terproteksi dengan emulasi Expanded Memory Specification (EMS) standar Lotus-Intel-Microsoft (LIM), pendahulu spesifikasi Extended Memory Specification (XMS) yang kemudian pada akhirnya mengubah topologi komputasi di dalam IBM PC. Semua aplikasi Windows dan berbasis DOS saat itu memang berjalan dalam modus real, yang berjalan di atas kernel modus terproteksi dengan menggunakan modus Virtual 8086, yang merupakan fitur baru yang dimiliki oleh Intel 80386.

Versi 2.03 dan kemudian versi 3.0 mendapatkan tuntutan dari Apple karena memang versi 2.1 ini memiliki modus penampilan jendela secara cascade (bertumpuk), selain beberapa fitur sistem operasi Apple Macintosh yang "ditiru" oleh Windows, utamanya adalah masalah tampilan/look and feel. Hakim William Schwarzer akhirnya membatalkan semua 189 tuntutan tersebut, kecuali 9 tuntutan yang diajukan oleh Apple terhadap Microsoft pada tanggal 5 Januari 1989

Kesuksesan dengan Windows 3.0 Microsoft Windows akhirnya mencapai kesuksesan yang sangat signifikan saat menginjak versi 3.0 yang dirilis pada tahun 1990. Selain menawarkan peningkatan kemampuan terhadap aplikasi Windows, Windows 3.0 juga mampu mengizinkan pengguna untuk menjalankan beberapa aplikasi MS-DOS secara serentak (multitasking), karena memang pada versi ini telah diperkenalkan memori virtual. Versi ini pulalah yang menjadikan IBM PC dan kompatibelnya penantang serius terhadap Apple Macintosh. Hal ini disebabkan dari peningkatan performa pemrosesan grafik pada waktu itu (dengan adanya kartu grafis Video Graphics Array (VGA)), dan juga modus terproteksi/modus 386 Enhanced yang mengizinkan aplikasi Windows untuk memakai memori lebih banyak dengan cara yang lebih mudah dibandingkan dengan apa yang ditawarkan oleh MS-DOS.

Windows 3.0 dapat berjalan di dalam tiga modus, yakni modus real, modus standar, dan modus 386 Enhanced, dan kompatibel dengan prosesor-prosesor keluarga Intel dari Intel 8086/8088, 80286, hingga 80386. Windows 3.0 akan mencoba untuk mendeteksi modus mana yang akan digunakan, meski pengguna dapat memaksa agar Windows bekerja dalam modus tertentu saja dengan menggunakan switch-switch tertentu saat menjalankannya

win /r: memaksa Windows untuk berjalan di dalam modus real win /s: memaksa Windows untuk berjalan di dalam modus standar win /3: memaksa Windows untuk berjalan di dalam modus 386 Enhanced. Versi 3.0 juga meru-

pakan versi pertama Windows yang berjalan di dalam modus terproteksi, meskipun kernel 386 enhanced mode merupakan versi kernel yang ditingkatkan dari kernel modus terproteksi di dalam Windows/386.

Karena adanya fitur kompatibilitas ke belakang, aplikasi Windows 3.0 harus dikompilasi dengan menggunakan lingkungan 16-bit, sehingga sama sekali tidak menggunakan kemampuan mikroprosesor Intel 80386, yang notabene adalah prosesor 32-bit.

Windows 3.0 juga hadir dalam versi "multimedia", yang disebut dengan Windows 3.0 with Multimedia Extensions 1.0, yang dirilis beberapa bulan kemudian. Versi ini dibundel dengan keberadaan "multimedia upgrade kit", yang terdiri atas drive CD-ROM dan sebuah sound card, seperti halnya Creative Labs Sound Blaster Pro. Versi ini merupakan perintis semua fitur multimedia yang terdapat di dalam versi-versi Windows setelahnya, seperti halnya Windows 3.1 dan Windows for Workgroups, dan menjadi bagian dari spesifikasi Microsoft Multimedia PC.

Fitur-fitur yang disebutkan di atas dan dukungan pasar perangkat lunak aplikasi yang semakin berkembang menjadikan Windows 3.0 sangat sukses di pasaran. Tercatat, dalam dua tahun sebelum dirilisnya versi Windows 3.1, Windows 3.0 terjual sebanyak 10 juta salinan. Akhirnya, Windows 3.0 pun menjadi sumber utama pemasukan Microsoft, dan membuat Microsoft melakukan revisi terhadap beberapa rencana awalnya.

Beralih sementara ke IBM OS/2 Selama pertengahan hingga akhir 1980an, Microsoft dan IBM bekerja sama dalam mengembangkan sebuah sistem operasi penerus DOS, yang disebut sebagai IBM OS/2. OS/2 dapat menggunakan semua kemampuan yang ditawarkan oleh mikroprosesor Intel 80286 dan mampu mengakses memori hingga 16 Megabyte. OS/2 1.0 dirilis pada tahun 1987, yang memiliki fitur swapping dan multitasking, selain tentunya mengizinkan aplikasi MS-DOS untuk berjalan di atasnya.

OS/2 versi 1.0 hanyalah sebuah sistem operasi yang berbasis modus teks/command line saja. OS/2 versi 1.1 yang dirilis pada tahun 1988 menawarkan antarmuka grafis, yang disebut dengan Presentation Manager (PM). Presentation Manager ini menggunakan sistem koordinat yang sama dengan koordinat Cartesius, berbeda dengan sistem operasi Windows dan beberapa sistem GUI lainnya. Penggunaan sistem koordinat tersebut menyebabkan titik x,y 0,0 pada OS/2 diletakkan pada pojok kiri bawah layar, sementara pada Windows, peletakannya pada pojok kiri atas. OS/2 versi 1.2, yang dirilis pada tahun 1989, memperkenalkan sebuah sistem berkas baru, yang disebut dengan High Performance File System (HPFS), yang ditujukan untuk menggantikan sistem berkas File Allocation Table (FAT).

Pada awal-awal tahun 1990an, hubungan antara Microsoft dan IBM pun meregang akibat munculnya sebuah konflik. Hal ini dikarenakan mereka saling bekerja sama dalam mengembangkan sistem operasi komputer pribadi masing-masing (IBM dengan OS/2 dan Microsoft dengan Windows-nya), keduanya memiliki akses terhadap kode masing-masing sistem operasi. Microsoft menghendaki pengembangan lebih lanjut dari sistem operasi Windows buatannya, sementara IBM memiliki hasrat bahwa semua pekerjaan masa depannya harus-

lah dibuat berdasarkan sistem operasi OS/2. Dalam sebuah percobaan untuk mengakhiri konflik ini, IBM dan Microsoft akhirnya setuju bahwa IBM akan mengembangkan IBM OS/2 versi 2.0, untuk menggantikan OS/2 versi 1.3 dan Windows 3.0, sementara Microsoft harus mengembangkan sebuah sistem operasi baru, OS/2 versi 3.0, yang akan kemudian menggantikan OS/2 versi 2.0.

Persetujuan ini pun tidak berlangsung lama, sehingga hubungan IBM dan Microsoft pun dihentikan. IBM akhirnya melanjutkan pengembangan OS/2, sementara Microsoft mengganti nama sistem operasi OS/2 versi 3.0 (yang belum dirilis) menjadi Windows NT. Keduanya masih memiliki hak untuk menggunakan teknologi OS/2 dan Windows yang sudah dibentuk sampai pemutusan persetujuan; akan tetapi, Windows NT benar-benar ditulis sebagai sebuah sistem operasi yang baru dan sebagian besar kode bebas dari kode IBM OS/2.

Setelah versi 1.3 dirilis untuk membenarkan beberapa masalah dalam OS/2 versi 1.x, IBM akhirnya merilis OS/2 versi 2.0 pada tahun 1992. Versi 2.0 ini menawarkan peningkatan yang signifikan, yakni sebuah GUI berorientasi objek, yang disebut dengan Workplace Shell (WPS), yang mencakup di dalamnya sebuah dekstop dan dianggap oleh banyak orang merupakan fitur terbaik di dalam OS/2. Microsoft pun akhirnya "menjiplak" beberapa elemen dari Workplace Shell pada sistem operasi Windows 95 yang dirilis tiga tahun kemudian. Versi 2.0 juga menawarkan API yang mendukung penuh instruksi 32-bit milik Intel 80386, sehingga menawarkan fitur multitasking yang bagus dan mampu mengalokasikan memori hingga 4 gigabyte. Meskipun demikian, banyak hal di dalam internal sistem masih menggunakan kode 16-bit, yang mengharuskan device driver juga harus ditulis dengan menggunakan kode 16-bit juga, selain tentunya beberapa hal internal lainnya. Hal ini merupakan salah satu alasan mengapa OS/2 kekurangan driver perangkat keras. Versi 2.0 juga mampu menjalankan aplikasi DOS dan Windows 3.0, karena memang IBM juga masih memiliki hak untuk menggunakan kode DOS dan Windows setelah "perceraian" hubungan antara mereka.

Pada saat itu, memang tidak jelas siapa yang menjadi pemenang dalam perlombaan yang disebut "Desktop Wars", akan tetapi pada akhirnya OS/2 tidak mampu mendapatkan pangsa pasar yang cukup meskipun IBM pada akhirnya merilis beberapa versi OS/2 yang jauh lebih hebat lagi setelah versi 2.0 ini. Windows 3.1

Tampilan ruangan kerja Windows 3.x Sebagai respons dari dirilisnya IBM OS/2 versi 2.0 ke pasaran, Microsoft mengembangkan Windows 3.1, yang menawarkan beberapa peningkatan minor terhadap Windows 3.0 (seperti halnya kemampuan untuk menampilkan font TrueType Fonts, yang dikembangkan secara bersama-sama dengan Apple), dan juga terdapat di dalamnya banyak sekali perbaikan terhadap bug dan dukungan terhadap multimedia. Versi 3.1 juga menghilangkan dukungan untuk modus real, sehingga hanya berjalan pada modus terproteksi yang hanya dimiliki oleh mikroprosesor Intel 80286 atau yang lebih tinggi lagi. Microsoft pun pada akhirnya merilis Windows 3.11, yang merupakan versi Windows 3.1 yang mencakup semua tambalan dan perbaikan yang dirilis setelah Windows 3.1 diluncurkan pada tahun 1992.

/p;

Windows for Workgroups Kira-kira pada waktu yang sama, Microsoft merilis Microsoft Windows for Workgroups, yang tersedia baik sebagai tambahan untuk Windows 3.1 dan sebagai sebuah versi yang tercakup di dalamnya lingkungan dasar Windows dan tambahan kemampuan jaringan di dalam sebuah paket. Windows for Workgroups mencakup driver jaringan komputer dan stack protokol yang lebih baik, dan juga mendukung jaringan secara peer-to-peer. Salah satu download opsional untuk Windows for Workgroups adalah stack protokol TCP/IP dengan nama kode "Wolverine", yang mengizinkan akses ke Internet melalui jaringan korporat. Windows for Workgroups dirilis dalam dua versi, Windows for Workgroups 3.1 dan Windows for Workgroups 3.11. Tidak seperti versi-versi Windows sebelumnya, Windows for Workgroups 3.11 hanya berjalan di dalam modus 386 Enhanced, dan membutuhkan setidaknya mesin dengan prosesor Intel 80386SX.

Semua versi tersebut terus menerus meningkatkan laju penjualan Windows versi 3.x. Meskipun Windows 3.1x masih memiliki banyak kekurangan, yang sebelumnya telah dikoreksi oleh OS/2, seperti nama berkas yang panjang (melebihi 11 karakter, dalam format 8.3), desktop, atau proteksi sistem terhadap kelakuan aplikasi yang tidak diinginkan, Microsoft secara cepat mengambil alih pasar GUI di pangsa pasar desktop untuk IBM PC dan kompatibelnya. Windows API pun menjadi standar de-facto untuk perangkat lunak konsumen.

Windows NT Selama waktu itu, Microsoft terus melanjutkan pengembangan sistem operasi yang barunya, yang disebut dengan Windows NT. Arsitek utama dari Windows NT adalah Dave Cutler, yang merupakan salah satu dari pemimpin arsitek sistem operasi VMS di perusahaan Digital Equipment Corporation (DEC), yang kemudian dibeli oleh Compaq yang sekarang bagian dari Hewlett-Packard. Microsoft merekrut Cutler pada tahun 1988 untuk membuat sebuah versi OS/2 yang bersifat portabel, tetapi akhirnya Cutler malahan membuat sistem operasi baru.

Sebelum pindah ke Microsoft, Cutler sebenarnya sedang membuat sebuah sistem operasi penerus VMS di DEC, yang disebut dengan Mica. Dan pada saat petinggi DEC menggagalkan proyek tersebut, ia keluar dari DEC dan membawa banyak pekerja ahli bersamanya ke Microsoft. DEC memiliki keyakinan bahwa Cutler membawa kode Mica untuk digunakan oleh Microsoft dan menuntutnya. Akhirnya, Microsoft pun kalah dan pada akhirnya disuruh membayar 150 juta dolar Amerika dan memiliki kesepakatan untuk mendukung chip CPU baru buatan DEC, DEC Alpha, yang kala itu tercatat sebagai sebuah chip tercepat, agar dapat berjalan di dalam Windows NT.

Windows NT 3.1 (bidang pemasaran Microsoft menghendaki Windows NT agar terlihat sebagai kelanjutan dari Windows 3.1) akhirnya muncul pertama kali dalam bentuk Beta bagi para pengembang perangkat lunak pada bulan Juli 1992 dalam sebuah perhelatan Professional Developers Conference (PDC) yang dilangsungkan di San Fransisco, California, Amerika Serikat. Microsoft juga mengumumkan keinginannya untuk mengembangkan sebuah sistem operasi penerus bagi Windows NT sekaligus juga pengganti Windows 3.1 pada konferensi tersebut (yang diberi nama kode Chicago), yang kemudian akan menyatukan keduanya ke dalam sebuah sistem operasi yang padu. Sistem operasi

tersebut diberi nama Cairo.

Ternyata Cairo merupakan sebuah proyek yang lebih rumit dibandingkan apa yang telah diantisipasi oleh Microsoft, dan hasilnya NT dan Chicago tidak "bersatu" sampai Windows XP diluncurkan. Selain itu, bagian-bagian Cairo belum muncul di dalam sistem operasi Microsoft Windows hingga saat ini. Contohnya adalah subsistem WinFS, yang merupakan implementasi dari Object File System di dalam Cairo, memang sempat dikerjakan oleh Microsoft dalam beberapa waktu, tetapi pada akhirnya Microsoft mengumumkan bahwa mereka menghentikan pengembangan WinFS dan akan menggabungkan teknologi yang dikembangkan untuk WinFS di dalam produk dan teknologi Microsoft yang lainnya, khususnya adalah Microsoft SQL Server.

Dukungan device driver untuk Windows NT juga kurang begitu banyak karena memang mengembangkan driver untuk Windows NT dianggap rumit oleh beberapa pengembang, selain tentunya Windows NT juga memiliki superioritas dalam model abstraksi perangkat kerasnya. Masalah ini telah menghantui semua versi Windows NT hingga Windows NT 5.0 (Windows 2000) keluar ke pasaran. Para programmer pun mengeluh bahwa mengembangkan device driver untuk Windows NT adalah sesuatu hal yang rumit, dan para pengembang perangkat keras juga tidak mau mengambil risiko untuk mengembangkan device driver untuk sebuah sistem operasi yang memiliki pangsa pasar terbatas. Selain itu, meskipun Windows NT menawarkan performa yang baik dan mampu mengeksploitasi sumber daya sistem secara lebih efisien, dalam beberapa sistem dengan perangkat keras terbatas, Windows NT dianggap sebagai sistem yang boros sumber daya. Hal ini mengakibatkan munculnya opini publik bahwa Windows NT hanya cocok untuk mesin-mesin yang besar dan juga jauh lebih mahal (seperti halnya workstation dengan DEC Alpha atau Intel Pentium yang kala itu memang masih baru). Windows NT juga tidak dapat bekerja untuk pengguna pribadi karena kebutuhan sumber dayanya yang tinggi. Selain itu, GUI yang digunakannya hanyalah salinan dari GUI Windows 3.1, yang masih kalah jauh jika dibandingkan dengan Workplace Shell milik OS/2, sehingga alasan Windows NT merupakan pengganti bagi Windows 3.1 tidaklah masuk akal.

Akan tetapi, fitur-fitur tersebutlah yang membuat Windows NT pilihan yang sempurna untuk pangsa pasar server jaringan lokal (LAN), yang pada tahun 1993 sedang mengalami booming besar-besaran, seiring dengan komoditas jaringan di dalam kantor telah meningkat secara drastis. Fitur-fitur jaringan dalam Windows NT menawarkan beberapa pilihan konektivitas jaringan yang luas dan juga tentunya sistem berkas NTFS yang efisien. Windows NT 3.51 merupakan primadona Microsoft saat terjun ke pasar ini, yang kemudian mengambil alih sebagian besar pangsa pasar yang sebelumnya dimiliki oleh Novell Netware beberapa tahun ke depan.

Salah satu peningkatan terbesar dari Windows NT adalah Application Programming Interface (API) 32-bit yang baru, yang dibuat untuk menggantikan Windows API 16-bit yang sudah lama. API 32-bit ini dinamakan dengan Win32 API, dan dari sanalah Microsoft menyebut API 16-bit yang lama sebagai Win16. Win32 API memiliki tiga buah implementasi utama: satu untuk Windows NT (yang merupakan Win32 API terlengkap dengan dukungan ANSI/ASCII

ataupun Unicode), satu untuk Win32s (yang merupakan bagian dari Win32 yang dapat digunakan di atas sistem Windows 3.1), dan juga satu lagi untuk Chicago (yang hanya mendukung ANSI). Hal ini menyebabkan kompatibilitas yang tinggi antara Chicago dan Windows NT, meskipun pada dasarnya kedua sistem tersebut adalah sangat jauh berbeda jika dilihat dari arsitektur dasarnya.

Windows NT merupakan sistem operasi Windows pertama yang dibuat dengan menggunakan kernel hibrida, setelah pada versi-versi sebelumnya hanya menggunakan kernel monolithic saja.

Windows 95

Tampilan layar Microsoft Windows 95. Setelah Windows 3.11, Microsoft mulai memulai pengembangan sebuah versi Windows yang berorientasi kepada pengguna yang diberi nama kode Chicago. Chicago didesain agar memiliki dukungan terhadap multitasking secara pre-emptive 32-bit seperti halnya yang terdapat di dalam OS/2 dan Windows NT, meskipun kernel 16-bit masih terdapat di dalamnya demi alasan kompatibilitas ke belakang. Win32 API yang pertama kali diperkenalkan pada Windows NT pun diadopsi sebagai sebuah standar antarmuka pemrograman baru yang berbasis 32-bit, dengan kompatibilitas Win16 juga dipertahankan dengan menggunakan sebuah teknik yang dinamakan dengan "thunking". GUI yang baru juga dimiliki oleh sistem operasi, meskipun pada awalnya Microsoft tidak merencanakannya sebagai bagian dari sistem operasi saat dirilis. Memang beberapa elemen antarmuka pengguna yang dimiliki oleh Cairo dipinjam dan ditambahkan kepada sistem operasi tersebut sebagai aspek lainnya dari versi itu (khususnya fitur Plug and Play).

Microsoft tidak mengganti semua kode Windows menjadi 32-bit; banyak bagian di dalamnya masih berupa 16-bit (meskipun tidak menggunakan modus real secara langsung), demi alasan kompatibilitas ke belakang, kinerja, dan waktu pengembangan. Hal ini dan fakta bahwa banyaknya kelemahan dalam desain versi-versi Windows sebelumnya, membuat sistem operasi yang baru ini terganggu efisiensi dan stabilitasnya.

Akhirnya, bagian marketing Microsoft menggunakan nama Windows 95 sebagai nama produk bagi Chicago, saat dirilis pada tanggal 24 Agustus 1995. Microsoft memiliki dua keuntungan dari peluncuran ini: 1) adalah mustahil bagi para konsumen untuk menjalankan Windows 95 pada sistem operasi DOS bukan buatan Microsoft yang jauh lebih murah. 2) meskipun jejak-jejak DOS tidaklah pernah dicabut dari sistem operasi tersebut, malahan versi tersebut menancapkan sebuah versi DOS untuk dimuat sebagai bagian dari proses booting, Windows 95 berjalan dengan sendirinya di dalam modus 386 Enhanced, dengan menggunakan memori virtual dan model pengalamatan memori flat 32-bit. Fitur-fitur itu menjadikan aplikasi Win32 untuk mengalamatkan RAM virtual sebanyak maksimal 2 gigabyte (dengan 2 gigabyte sisanya dicadangkan untuk sistem operasi), dan dalam teorinya mencegah aplikasi tersebut untuk mengganggu ruangan memori yang dimiliki oleh aplikasi Win32 lainnya tanpa adanya persetujuan dari sistem operasi. Dalam hal ini, memang fungsionalitas Windows 95 mendekati apa yang dimiliki oleh Windows NT, meskipun Windows 95/98/Me tidak mendukung memori melebihi 512 megabyte tanpa menyunting konfigurasi sistem yang merepotkan.

Di balik kesuksesan Microsoft, IBM terus melanjutkan pasar OS/2, dengan memproduksi OS/2 versi 3.0 dan OS/2 Warp (versi 4.0). IBM merespons keluhan yang diajukan oleh para konsumen mengenai kebutuhan perangkat keras komputer yang tinggi yang diminta oleh OS/2 versi 2.0, OS/2 versi 3.0 pun dibuat jauh lebih ramping dengan melakukan optimalisasi pada ukuran dan kecepatan. Sebelum Windows 95 dirilis, OS/2 Warp 3.0 bahkan telah dibundel dalam beberapa penyedia perangkat keras mayor di Jerman. Akan tetapi, dengan dirilisnya Windows 95, OS/2 lagi-lagi kehilangan pangsa pasarnya kembali, diambil alih oleh Windows 95.

Mungkin tidak mungkin untuk mencari tahu apa alasan mengapa OS/2 gagal dalam mendapatkan pangsa pasar yang besar. Walaupun OS/2 terus dapat menjalankan aplikasi Windows 3.1, sebenarnya sudah tidak ada yang kekurangan lagi, kecuali pada sebagian kecil dari Windows API yang disebut Win32s. Tidak seperti Windows 3.1, IBM tidak memiliki akses terhadap kode sumber Windows 95 dan tidak mau menggunakan waktu dan sumber daya yang ada untuk melakukan emulasi terhadap Win32 API. IBM juga memperkenalkan OS/2 pada kasus Amerika Serikat versus Microsoft, dengan menyalahkan taktik marketing pada bagian Microsoft, tetapi banyak orang mungkin setuju bahwa masalah di dalam bagian marketing IBM lah dan dukungannya yang sangat kurang dari para pengembang perangkat lunak yang menyebabkan kegagalan OS/2.

Sebelum menggantinya dengan versi Windows yang baru, Microsoft merilis Windows 95 dalam lima versi berbeda, yakni sebagai berikut:

Windows 95 - rilis yang sebenarnya dari Windows 95 Windows 95 A - mencakup pembaruan Windows 95 Original Service Release 1 (OSR1) yang dimasukkan secara langsung terhadap instalasi. Windows 95 B - mencakup beberapa pembaruan mayor lainnya, seperti halnya sistem berkas FAT32, dan Internet Explorer 3.0. Versi ini juga dikenal dengan Windows 95 OSR2, atau banyak orang di Indonesia menyebutnya sebagai Windows 97. Windows 95 B USB - atau Windows 95 OSR2.1 merupakan versi Windows 95 yang menawarkan dukungan terhadap perangkat keras berbasis bus Universal Serial Bus/USB. Windows 95 C - atau Windows 95 OSR2.5 mencakup semua fitur di atas, ditambah Internet Explorer 4.0. Versi ini merupakan versi yang paling terakhir dirilis dari seri Windows 95. Windows 95 OSR2, OSR2.1 dan OSR2.5 tidaklah dirilis untuk publik, akan tetapi hanya kepada OEM saja yang mau menggunakan sistem operasi tersebut ke dalam komputer buatannya. Beberapa perusahaan OEM bahkan menjual hard disk baru dengan sistem operasi Windows 95 OSR2 di dalamnya.

Selain fitur yang terkandung di dalam Windows 95, Microsoft juga memperkenalkan Microsoft Plus! for Windows 95 yang mencakup beberapa fitur tambahan yang tidak dimiliki oleh Windows 95.

Windows NT 4.0

Tampilan layar Windows NT 4.0 Server. Microsoft merilis Windows NT 4.0, sebagai penerus Windows NT 3.x yang sukses mengancam dominasi Novell Netware dan UNIX di pasar korporat. Windows NT 4.0 ini pada awalnya dikembangkan sebagai sebuah bagian dari usaha untuk memperkenalkan Windows NT

kepada pasar workstation. NT 4.0 memiliki antarmuka yang sama dengan Windows 95, tetapi menggunakan kernel yang sama dengan Windows NT, sehingga lebih stabil. Memang, ada sebuah patch tambahan yang tersedia untuk Windows NT 3.51 yang mampu membuat NT 3.51 agar mirip seperti NT 4.0, tetapi sangat tidak stabil dan memiliki banyak bug. Antarmuka pengguna tersebut pada awalnya memang dikembangkan di atas Windows NT, tetapi karena Windows 95 dirilis terlebih dahulu sebelum NT 4.0, maka orang cenderung berpikir "Windows NT 4.0 adalah jiplakan dari Windows 95."

Windows NT 4.0 datang dalam empat versi:

Windows NT 4.0 Workstation Windows NT 4.0 Server Windows NT 4.0 Server, Enterprise Edition (yang mencakup dukungan terhadap clustering dan SMP hingga 8-way) Windows NT 4.0 Terminal Server Windows 98

Tampilan layar Windows 98. Pada 25 Juni 1998, Microsoft merilis sebuah sistem operasi Windows baru, yang dikenal sebagai Windows 98. Windows 98 dianggap sebagai revisi minor terhadap Windows 95, tetapi secara umum dilihat jauh lebih stabil dan dapat diandalkan dibandingkan dengan pendahulunya, Windows 95. Windows 98 mencakup banyak driver perangkat keras baru dan dukungan sistem berkas FAT32 yang lebih baik yang mengizinkan partisi untuk memiliki kapasitas lebih besar dari 2 gigabyte, sebuah batasan yang terdapat di dalam Windows 95. Dukungan USB di dalam Windows 98 pun juga jauh lebih baik dibandingkan dengan pendahulunya.

Windows 98 mengundang kontroversi saat Microsoft memasukkan penjelajah web Microsoft Internet Explorer ke dalam sistem operasi dan tidak dapat dicabut, sehingga menjadikan Windows Explorer dan GUI Windows mampu menampilkan direktori seolah-olah halaman web. Hal ini membuka kasus baru, yang disebut sebagai Amerika Serikat versus Microsoft, yang menanyakan mengapa Microsoft repot-repot menjaga dominasinya di dalam sistem operasi komputer pribadi untuk berkompetisi dengan para pesaingnya seperti Netscape dan IBM dengan cara yang tidak jujur.

Pada tahun 1999, Microsoft merilis Windows 98 Second Edition, sebuah rilis yang menawarkan banyak peningkatan dibandingkan versi sebelumnya. Di sini, Microsoft menambahkan Internet Explorer 5.0 yang ada di Windows 2000. Selain itu, Internet Connection Sharing (ICS), yang merupakan sebuah bentuk dari Network Address Translation (NAT), yang mengizinkan beberapa mesin di dalam sebuah jaringan lokal agar dapat menggunakan satu buah jalur koneksi Internet bersama-sama pun diperkenalkan pada versi ini. Banyak masalah minor di dalam Windows yang lama telah dikoreksi, yang menjadikan Windows 98 menurut banyak orang sebagai sebuah versi Windows 9x yang paling stabil di antara semua versi Windows 9x lainnya.

Windows 2000

Logo yang digunakan oleh Microsoft Windows selama tahun 2000

Tampilan desktop Windows 2000 Microsoft merilis Windows 2000 pada 17 Februari 2000, sebuah versi yang sebelumnya dikenal dengan sebutan Windows NT 5.0 atau "NT 5.0". Versi Windows 2000 ditujukan untuk dua pangsa pasar, yakni pangsa pasar workstation dan juga pangsa pasar server.

Di antara fitur-fitur Windows 2000 yang paling signifikan adalah Active Di-

rectory, sebuah model jaringan pengganti model jaringan NT domain, yang menggunakan teknologi yang merupakan standar industri, seperti Domain Name System (DNS), Lightweight Directory Access Protocol (LDAP), dan Kerberos untuk menghubungkan antara sebuah mesin ke mesin lainnya. Windows Terminal Services juga, yang pada Windows NT 4.0 hanya terdapat di dalam satu produk saja, pada Windows 2000 terdapat dalam semua versi server. Fitur-fitur baru yang diadopsi dari Windows 98 juga ditanamkan di dalamnya, seperti Device Manager yang telah ditingkatkan (dengan menggunakan Microsoft Management Console), Windows Media Player, dan DirectX 6.1 (yang memungkinkan sistem operasi berbasis kernel Windows NT untuk menjalankan game). Windows 2000 juga merupakan versi Windows berbasis kernel NT terakhir yang tidak mengharuskan penggunaannya untuk melakukan aktivasi terhadapnya.

Meskipun Windows 2000 dapat memperbarui komputer yang sebelumnya menjalankan Windows 98, Windows 2000 tidaklah dianggap sebagai produk yang cocok untuk pengguna rumahan. Alasannya banyak, di antaranya adalah kurangnya device driver untuk banyak perangkat pengguna seperti pemindai (scanner) dan juga pencetak (printer), pada saat dirilis. Situasi tersebut akhirnya berbalik pada saat Windows XP dirilis oleh Microsoft.

Windows 2000 tersedia dalam enam edisi, yakni:

Windows 2000 Professional Windows 2000 Server Windows 2000 Advanced Server Windows 2000 Datacenter Server Windows 2000 Advanced Server Limited Edition Windows 2000 Datacenter Server Limited Edition Windows Me

Tampilan Windows Millennium Edition screenshot

Logo Windows Me Pada bulan September 2000, Microsoft memperkenalkan Windows Millennium Edition (dikenal juga dengan sebutan Windows Me atau Windows ME). Versi ini memperbarui Windows 98 dengan dukungan multimedia dan Internet yang lebih baik. Versi ini juga memasukkan fitur "System Restore," yang mengizinkan para penggunaannya untuk mengembalikan keadaan sistem ke sebuah titik yang dikenal baik-baik saja, pada saat sistem operasi mengalami kegagalan. System Restore menjadi fitur yang masih dipertahankan pada Windows XP. Versi ini juga memperkenalkan Windows Movie Maker versi pertama.

Windows Me dibuat dalam waktu yang singkat, kira-kira hanya satu tahun, yang ditujukan hanya untuk mengisi kekosongan rilis antara Windows 98 dan Windows XP sebagai sistem operasi untuk kelas rumahan. Fitur-fitur yang terdapat di dalam Windows Me (seperti Internet Explorer 5.5, Windows Media Player 7.0, dan Microsoft DirectX 7.1) bahkan bisa diperoleh secara gratis dari situs Windows Update, kecuali System Restore. Hasilnya, Windows Me pun tidak dianggap sebagai sebuah sistem operasi yang unik di antara saudara-saudaranya dari keluarga Windows 9x, Windows 95 dan Windows 98. Windows Me juga dikritik karena munculnya masalah kestabilan, dan juga dukungan terhadap MS-DOS yang berjalan di dalam modus real. Orang-orang bahkan menyebut Windows Me sebagai Windows Mistake Edition.

Windows Me merupakan sistem operasi terakhir yang dibuat berdasarkan kernel monolithic Windows 9x dan MS-DOS. Versi ini pun menjadi versi terakhir sistem operasi Windows yang tidak memiliki Windows Product Activation

(WPA).

Windows XP: menyatukan kedua jajaran produk

Logo Microsoft Windows yang didesain ulang oleh Microsoft saat merilis Windows XP, yang digunakan hingga 4 Desember 2006

Tampilan Windows XP menampilkan folder Sample Pictures, Sample Music, dan Control Panel, dengan tema Luna dan desktop Bliss. Pada tahun 2001, Microsoft memperkenalkan Windows XP (yang memiliki nama kode "Whistler" selama pengembangan). Akhirnya, setelah merilis beberapa versi Windows berbasis Windows 9x dan NT, Microsoft berhasil menyatukan kedua jajaran produk tersebut. Windows XP menggunakan kernel Windows NT 5.1, sehingga menjadikan kernel Windows NT yang terkenal dengan kestabilannya memasuki pasar konsumen rumahan, untuk menggantikan produk Windows 9x yang berbasis 16/32-bit yang sudah menua.

Windows XP merupakan versi sistem operasi Windows yang paling lama (paling tidak hingga saat ini), karena memang berkisar dari tahun 2001 hingga tahun 2007, saat Windows Vista dirilis ke konsumen. Jajaran sistem operasi Windows XP akhirnya diteruskan oleh Windows Vista pada 30 Januari 2007.

Windows XP tersedia dalam beberapa versi:

Windows XP Home Edition, yang ditujukan untuk pasar desktop dan laptop rumahan. Windows XP Home Edition N, sama seperti Home Edition yang biasa, tetapi tidak memiliki Windows Media Player, karena memang peraturan Uni Eropa tidak memperbolehkannya. Windows XP Professional, yang ditujukan bagi para power user dan pebisnis. Windows XP Professional N, sama seperti Professional Edition, tetapi tidak memiliki Windows Media Player, karena peraturan Uni Eropa tidak mengizinkannya. Windows XP Media Center Edition (MCE), dirilis pada bulan November 2002, merupakan Windows XP Home Edition yang ditujukan untuk dekstop dan laptop dengan penekanan pada hiburan rumahan. Windows XP Media Center Edition 2003 Windows XP Media Center Edition 2004 Windows XP Media Center Edition 2005, yang dirilis pada 12 Oktober 2004. Windows XP Tablet PC Edition, yang ditujukan untuk PC Tablet (PC/netbook dengan layar sentuh) Windows XP Tablet PC Edition 2005 Windows XP Embedded, yang ditujukan untuk sistem benam (embedded system) Windows XP Starter Edition, yang ditujukan untuk para pengguna komputer di beberapa negara berkembang. Windows XP Professional x64 Edition, yang dirilis pada 25 April 2005 untuk sistem-sistem rumahan dan workstation yang menggunakan prosesor 64-bit yang berbasiskan set instruksi x86-64 (AMD64 atau Intel EM64T). Windows XP 64-bit Edition, merupakan sebuah versi Windows XP yang ditujukan untuk jajaran prosesor Intel Itanium, yang mempertahankan kompatibilitas dengan aplikasi 32-bit dengan menggunakan emulator perangkat lunak. Versi ini sama saja dengan Windows XP Professional, dari segi fitur-fiturnya. Produk ini dihentikan pada bulan September 2005 saat vendor terakhir workstation berbasis prosesor Itanium menghentikan produk-produknya sebagai "workstation", karena memang mereka cenderung memfokuskan Itanium sebagai basis komputer server. Windows XP 64-bit Edition 2003, dibuat berbasiskan basis kode Windows NT 5.2 (sama seperti Windows Server 2003). Windows Server 2003

Desktop dan menu Start Windows Server 2003. Pada tanggal 24 April 2003, Microsoft meluncurkan Windows Server 2003, sebuah pembaruan untuk sistem operasi Windows 2000 Server, yang menawarkan banyak fitur-fitur keamanan yang baru, pemandu "Manage Your Server wizard" yang menyederhanakan peranan sebuah mesin yang menjalankannya, dan juga peningkatan kinerja. Windows Server 2003 menggunakan kernel Windows NT versi 5.2.

Di dalam Windows Server 2003, beberapa layanan yang tidak terlalu dibutuhkan di dalam lingkungan server dinonaktifkan secara default, terutama "Windows Audio" dan "Themes" demi alasan kestabilan; Agar dapat menggunakan suara dan tampilan yang sama dengan Windows XP, pengguna harus mengaktifkannya secara manual, melalui snap-in Microsoft Management Console Services.msc Selain itu, akselerasi perangkat keras untuk kartu grafis juga dimatikan; lagi-lagi pengguna harus mengaktifkannya secara manual, tentu saja jika device driver yang digunakan "bisa dipercayai".

Pada bulan Desember 2005, Microsoft merilis Windows Server 2003 R2, yang merupakan Windows Server 2003 Service Pack 1 ditambah dengan beberapa paket tambahan. Di antara semua fitur-fitur barunya adalah fitur-fitur manajemen untuk kantor-kantor cabang, dan integrasi identitas yang luas.

Windows Server 2003 tersedia dalam lima buah edisi:

1. Windows Server 2003, Web Edition
2. Windows Server 2003, Standard Edition
3. Windows Server 2003, Enterprise Edition (32-bit dan 64-bit)
4. Windows Server 2003, Datacenter Edition
5. Windows Server 2003, Small Business Server
6. Windows Fundamentals for Legacy PCs sebagai platform thin client

Tampilan desktop Windows Fundamentals for Legacy PCs Pada bulan Juli 2006, Microsoft merilis sebuah versi Windows XP Service Pack 2, yang ditujukan untuk pasar thin-client, yang disebut sebagai Windows Fundamentals for Legacy PCs (WinFLP). WinFLP hanya tersedia bagi para pelanggan Microsoft Software Assurance. Tujuan dibuatnya WinFLP adalah untuk memberikan pilihan upgrade kepada para pelanggannya yang masih menggunakan Windows 95, Windows 98, Windows Me, dan Windows NT Workstation. Sebagian besar aplikasi pengguna dijalankan di atas mesin jarak jauh dengan menggunakan Terminal Services atau Citrix. Pada tahun 2009, Microsoft merilis Windows Thin PC yang merupakan penerus dari WinFLP.

Windows Vista Setelah meraih kesuksesan besar dengan Windows XP, Microsoft tidak lantas berhenti begitu saja mengembangkan Windows. Versi terbaru dari Windows, disebut dengan Windows Vista, dirilis pada tanggal 30 November 2006 [1] bagi kalangan bisnis sementara untuk kalangan pengguna rumahan dirilis pada tanggal 30 Januari 2007. Windows Vista memang dicanangkan agar memiliki keamanan yang lebih tangguh dibandingkan dengan

versi-versi sebelumnya, dengan memperkenalkan sebuah modus pengguna yang terbatas, yang disebut sebagai User Account Control (UAC), untuk menggantikan filosofi "administrator-by-default" yang diberlakukan pada Windows XP. Windows Vista juga memperkenalkan fitur grafik yang jauh lebih "memikat", yang disebut dengan Windows Aero GUI, aplikasi yang baru (seperti halnya Windows Calendar, Windows DVD Maker dan beberapa game baru termasuk Chess Titans, Mahjong, dan Purple Place). Selain itu, Windows Vista juga menawarkan versi Microsoft Internet Explorer yang lebih aman, serta Windows Media Player versi baru (versi 11).

Windows Vista, menampilkan antarmuka grafis Aero-nya yang memikat, Welcome screen dan menu Start. Windows Vista menggunakan nomor versi 6.0, sehingga memang terdapat perbedaan versi yang signifikan jika dibandingkan dengan Windows XP yang menggunakan nomor versi 5.1 atau Windows Server 2003 (5.2). Karena, memang Windows Vista memiliki banyak perbedaan yang mendasar, khususnya pada bagian arsitektur dasar sistem operasi.

Windows Vista dijual dalam beberapa edisi:

Windows Vista Starter Windows Vista Home Basic Windows Vista Home Premium Windows Vista Business Windows Vista Enterprise Windows Vista Ultimate Windows Home Server

Windows Home Server Console Windows Home Server (sebelumnya memiliki nama kode Q singkatan dari Quattro) merupakan sebuah produk server yang diturunkan dari Windows Server 2003, yang didesain khusus untuk digunakan oleh para konsumen dari pengguna rumahan. Sistem operasi ini diperkenalkan pada tanggal 7 Januari 2007 oleh Bill Gates. Windows Home Server dapat dikonfigurasi dan dipantau dengan menggunakan program console yang dapat diinstalasi pada sebuah PC klien. Windows ini memiliki fitur Media Sharing, backup terhadap drive lokal dan drive jarak jauh, dan duplikasi berkas.

Windows Server 2008 Windows Server 2008, adalah sebuah versi baru Windows Server, yang dijadwalkan untuk dirilis pada tanggal 27 Februari 2008. Pada saat pengembangannya, Windows Server memiliki nama kode "Windows Server Codenamed Longhorn." Windows Server 2008 dibangun di atas beberapa keunggulan teknologi dan keamanan yang pada awalnya diperkenalkan dengan Windows Vista, dan ditujukan agar bisa lebih modular secara signifikan, ketimbang pendahulunya, Windows Server 2003.

Windows 7 Rilis selanjutnya setelah Windows Vista adalah Windows 7, yang sebelumnya dikenal dengan sebutan Blackcomb dan Vienna. Saat pertama kali dirilis, Windows ini memiliki kernel NT versi 6.1 build 7600, yaitu perbaikan dari Windows Vista dimana saat rilis pertama memiliki kernel NT 6.0 build 6000. Windows 7 yang dirilis pada tanggal 22 Oktober 2009 ini memiliki keamanan dan fitur yang baru, diantaranya adalah: Jump List, Taskbar yang membuka program dengan tampilan kecil, Windows Media Player 12, Internet Explorer 8, dan lain-lain. Beberapa fitur yang unik adalah Sidebar yang berganti nama menjadi Gadget dan bebas ditaruh ke mana-mana pada desktop (tidak seperti Sidebar yang hanya bisa diletakkan di tempat tertentu). Fitur itu membuat Windows 7 menjadi menarik. Spesifikasi Windows 7 lebih ringan dan harganya juga lebih murah daripada Windows Vista.

Sama seperti Windows Vista, Windows 7 juga tersedia dalam 6 edisi yaitu: Windows 7 Starter Windows 7 Home Basic Windows 7 Home Premium Windows 7 Professional Windows 7 Ultimate Windows 7 Enterprise Windows 8
Artikel utama: Windows 8

Layar mulai Windows 8 Pengembangan Windows 8 dimulai sebelum Windows 7 diluncurkan pada tahun 2009. Tanggal 13 September 2011, build 8102 (Windows 8 Developer Preview) dirilis ke publik. Build ini terbuka penuh untuk pertama kalinya dan dilengkapi Start Screen baru, antarmuka pengguna Metro dan sejumlah sampel aplikasi buatan para karyawan magang di Microsoft. Pada tanggal 29 Februari 2012, Microsoft meluncurkan Windows 8 Consumer Preview, versi beta dari Windows 8, build 8250. Untuk pertama kalinya sejak Windows 95, tombol Start tidak lagi muncul di taskbar, meski layar Start-nya masih harus dibuka dengan mengklik sudut kiri bawah layar dan tombol Start di kotak Charm. Pada tanggal 1 Agustus 2012, Windows 8 (build 9200) dirilis dengan nomor build 6.2.9200.16384. Microsoft mengadakan acara peluncuran pada 25 Oktober 2012 dan meluncurkan Windows 8 untuk publik keesokan harinya.

Fitur:

Desain aplikasi Windows 8 UI, yang bisa dibuat dari bahasa pemrograman apa saja, seperti misalnya bahasa pemrograman web seperti HTML, CSS, dan JavaScript, dan bahasa pemrograman aplikasi desktop seperti C, C++, C, dll. Hilangnya tombol Mulai (digantikan oleh Layar Mulai dan Charms). Hilangnya tombol Mulai pada desktop Kompatibel dengan arsitektur Intel, AMD, dan ARM (khususnya tablet). Kompatibel dengan perangkat berspesifikasi rendah seperti Tablet, Laptop, Netbook, bahkan Smartphone Portabel di USB (Windows To Go). Windows Explorer menggunakan ribbon seperti Office 2010 2007, juga berganti nama menjadi 'File Explorer. Hybrid Boot, fitur untuk mempercepat waktu start-up dengan menyimpan memori inti Windows ke dalam hard disk dan me-load-nya tiap booting. Fitur pengembalian (recovery) baru, Refresh dan Reset. Refresh akan mengembalikan semua file Windows ke kondisi awal tanpa mengubah pengaturan, berkas, atau aplikasi Windows 8 UI. Reset akan mengembalikan komputer ke kondisi standar pabrikan. Rancangan baru dari Windows Task Manager. Proses aktivasi yang mudah. Diperkenalkannya Layar Kunci (Lock Screen). Bursa Windows, sebagai tempat membeli atau menginstal aplikasi Windows 8 UI. Aplikasi media penyimpanan awan baru yang disebut Skydrive. Antivirus yang sudah terintegrasi dengan Windows Defender. Pilihan jenis log-in yang beragam, yaitu sandi akun Microsoft/lokal, sandi gambar, dan PIN. Edisi:

Windows 8 Windows 8 adalah edisi standar Windows untuk arsitektur IA-32 dan x64. Dokumentasi yang didapat dari ImageX tool dan buku Paul Thurrott di Windows 8 juga menyebutkan versi ini sebagai "Core". Edisi ini membawa fitur-fitur yang ditunjukkan untuk pengguna rumahan dan memberikan semua fitur standar Windows 8 seperti Layar Mulai, ubin langsung, Bursa Windows, Internet Explorer 10, dan lain-lain.

Windows 8 Pro Windows 8 Pro sebanding dengan Windows 7 Professional dan Ultimate dan ditunjukkan pada pengguna tingkat lanjut atau pengguna rumahan tingkat lanjut (dengan Windows Media Center meskipun harus mem-

belinya lagi). Fitur tambahan meliputi kemampuan untuk menerima koneksi Remote Desktop, ikut serta dalam Domain Windows Server, Enkripsi Berkas Sistem, Hyper-V, pemuatan (booting) Virtual Hard Disk, Group Policy, BitLocker dan BitLocker To Go.

Windows 8 Enterprise Windows 8 Enterprise menyediakan semua fitur di Windows 8 Pro (kecuali kemampuan untuk memasang Windows Media Center), dengan fitur tambahan untuk membantu organisasi TI (lihat tabel dibawah). Edisi ini tersedia untuk pengguna Software Assurance atau pelanggan MSDN dan TechNet Professional, dan dirilis tanggal 16 Agustus 2012.

Windows RT Windows RT hanya akan tersedia terpasang di perangkat berbasis ARM seperti tablet PC. Windows RT akan terpaket dengan versi yang teroptimalkan untuk sentuh dari versi desktop Microsoft Office 2013 Home and Student (Microsoft Office RT 2013) berisi Word, Excel, PowerPoint, dan OneNote, dan mendukung kemampuan enkripsi perangkat. Beberapa fitur kantor seperti Group Policy dan dukungan domain tidak didukung.

Windows 8.1 Artikel utama: Windows 8.1

Layar Mulai Windows 8.1 Update 1 Rilis selanjutnya setelah Windows 8 adalah Windows 8.1, yang sebelumnya dikenal dengan sebutan Blue. Tanggal 26 Juni 2013, build 9431 (Windows 8.1 Free Preview) dirilis ke publik. Build ini terbuka penuh untuk pertama kalinya dan dilengkapi dengan tombol Mulai baru, booting langsung ke desktop, layar kunci yang disempurnakan dan sejumlah aplikasi Metro yang baru. Pada tanggal 27 Agustus 2013, Windows 8.1 (build 9600) dirilis dengan nomor build 6.3.9600.16384. Microsoft mengadakan acara peluncuran pada 17 Oktober 2013 dan meluncurkan Windows 8.1 untuk publik keesokan harinya.

Fitur:

Aplikasi standar baru: Calculator (Modern UI), Alarm, Sound Recorder, Reading List, Food Drink, Help Tips, dan sebuah File Manager berbasis Modern UI (menyatu bersama aplikasi SkyDrive Modern UI). Kembalinya Tombol start pada windows dan Hilangnya aplikasi Messaging. Terintegrasi dengan UWP Internet Explorer 11 (Preview).

UWP Internet Explorer 11 Preview di Windows 8.1 Preview PC Settings, merangkum pengaturan-pengaturan Control Panel (berbasis Modern UI). Windows PowerShell v40: sejumlah perintah baru untuk mengelola Start Screen, Windows Defender, dll. Bursa Windows 2.0: Tampilan Baru dan lebih fleksibel, memperbarui aplikasi secara otomatis. Boot ke desktop: Windows 8.1 bisa booting langsung ke desktop tanpa harus ke Layar Mulai. Latar Layar Mulai: Warna dan Pola latar belakang di Layar Mulai Windows 8 bisa diatur sesuka hati, bahkan bisa disamakan dengan desktop. Search Heroes: Di Windows 8, kita hanya bisa mencari aplikasi, pengaturan, dan files. Di Windows 8.1, selain mencari aplikasi, pengaturan, dan files, kita juga bisa mencari orang-orang terkenal, artis, kota, atau apa saja; karena pencarian Windows 8.1 sudah terintegrasi dengan internet lewat Bing. Multitugas fleksibel: Di Windows 8, kita hanya bisa menjalankan 2 aplikasi secara bersamaan. Di Windows 8.1, kita bisa menjalankan 3 atau 4 aplikasi secara bersamaan, tergantung ukuran layar. Layar Mulai fleksibel: Kustomisasi lebih fleksibel, menggrup,,menempel, melepas,

dan mengubah ukuran ubin aplikasi menjadi lebih mudah. Tampilan Baru File Explorer: Pustaka (Library) disembunyikan secara default (namun menjadi satu dibawah This PC) dan Computer (Komputer) berubah nama menjadi This PC (PC ini). Sama seperti Windows 8, Windows 8.1 juga tersedia dalam 4 edisi yaitu:

Windows 8.1 Windows 8.1 Pro Windows 8.1 Enterprise Windows RT 8.1 Windows 10 Artikel utama: Windows 10

Desktop Windows 10 Windows 10 merupakan rilis mendatang dari sistem operasi Microsoft Windows. Diperkenalkan pada tanggal 30 September 2014, dirilis pada tanggal 29 Juli 2015. Chrome Google Chrome adalah sebuah penjelajah web sumber terbuka yang dikembangkan oleh Google dengan menggunakan mesin rendering WebKit. WebKit adalah sebuah Mesin Layout yang didesain agar penjelajah web dapat merender halaman web. Fungsi dan kegunaan google chrome sama dengan mozilla firefox dan aplikasi web browser lainnya yang berfungsi untuk menampilkan halaman-halaman web di internet

Sejarah google google os dan google chrome Google lahir dari sebuah pertemuan tanpa disengaja dari Larry Page dan Sergey Brin di Universitas Stanford . Pada Januari 1996 Lary dan Sergey mulai melakukan kolaborasi dalam pembuatan Search Engine yang diberi nama BackRub. 1998 Teknologi Search Engine itu terus di sempurnakan, keduanya mulai mencari Investor untuk mengembangkan kecanggihan teknologi mesin pencari mereka.

Sehingga, mereka mendapatkan suntikan dana dari Andy Bechtolsheim teman kampus sekaligus pendiri Sun Microsystem. Sebuah cek senilai 100 ribu dolar AS, yang uniknya di cek itu tertulis atas nama Google yang bahkan pada saat itu belum didirikan oleh Sergey dan Larry.

Larry dan Sergey sebelumnya pernah menawarkan kemungkinan dengan Alta Vista, tetapi ditolak dengan alasan perusahaan induk Alta Vista yaitu Digital Equipment Corp tidak suka bergantung pada orang dari luar perusahaan.

Investor di Silicon Valley, Michael Moritz (Sequa Capital) dan Jhon Doer (Kleiner Perkins) yang saling berkompetisi akhirnya bisa diyakinkan untuk menyuntik modal senilai 12,5 juta dolar AS ke perusahaan Google yang pada 7 Desember 1998 ini didirikan.

Kantor pertamanya adalah sebuah ruang garasi rumah teman mereka di Menlo Park, California. Pada tahun 1999, Google pindah ke kantor di 165 University Ave Palo Alto California sebelum akhirnya pindah ke Googleplex pada akhir tahun tersebut.

Larry dan Sergey terus menjalankan mesin pencari Google hingga tahun 2001. Setelah itu mereka merekrut Eric E. Schmidt untuk menjadi ketua umum dan CEO Google. Google kini menjadi perusahaan yang paling berpengaruh di Internet, termasuk karena Blogger.com sudah mereka kuasai.

Googleplex menjadi tempat kerja yang sangat nyaman, karena dilengkapi sarana dan prasarana yang lengkap seperti kolam renang, bar, billiard room, futsal, voli pantai, makanan gratis, minuman kesehatan cuma-cuma, dan cemilan yang melimpah. Terlebih lagi, suasana kerja yang rileks karena karyawan google diperbolehkan untuk hanya mengenakan kaus dan celana jeans, sehingga apabila ketika kita datang ke googleplex dan melihat seseorang yang mengenakan

pakaian resmi, jas dan sebagainya, hampir bisa dipastikan dia adalah Tamu.

Satu lagi yang menarik, di Google diterapkan sistem , karyawan diberikan 20

Sebuah artikel yang ditulis oleh seorang wartawan dari Fortune, Freg Vogelstein, berjudul " Why Google Scares Bill Gates " menjelaskan persaingan antara Google dengan Microsoft. Ini membuktikan betapa hebatnya google saat ini, sehingga raksasa seperti Microsoft yang sudah berusia 25 tahun lebih, menerapkan berbagai strategi untuk bersaing dengan google. Ini terlihat nyata, dengan rencana Akuisisi Yahoo.com oleh Microsoft baru-baru ini, meskipun akhirnya kandas.

Di balik itu, Google juga melakukan kerja-kerja filantropis (sosial) dengan mendirikan Google.org. Misi dari organisasi itu adalah membantu kampanye soal perubahan iklim, pemanasan global, juga kesehatan dan kemiskinan global. Direktur dari Google.org ini adalah Dr. Larry

banyak pengguna komputer akan hadirnya operating system yang mampu bersaing dengan microsoft akhirnya terwujud. Google, perusahaan yang banyak menelurkan aplikasi menarik dan gratis, tidak lama lagi akan segera merilis mainan terbarunya, yakni Google Chrome OS pada kuartal kedua tahun 2010 dan target utamanya adalah para pengguna netbook.

Apa bedanya Google Chrome OS dan microsoft Windows? Google Chrome OS akan berjalan pada processor x86 dan ARM, jadi netbook dengan processor Intel Atom, smartbook dengan chip Nvidia Tegra, atau Qualcomm Snapdragon hingga netbook dengan platform AMD Neo dapat menggunakan Google Chrome OS. Google menambahkan bahwa Chrome OS dapat dijalankan untuk komputer dengan energi rendah seperti netbook hingga komputer desktop dengan energi berlimpah.

Teknologi HTML 5 juga akan menjadi bagian tidak terpisahkan dari Chrome OS. Banyak pengguna internet yang ragu jika menjalankan aplikasi web, tapi mereka tidak terhubung ke internet. tapi dengan HTML 5 semuanya akan berubah, pengguna dapat bekerja melalui browser walaupun tidak terhubung dengan internet. Google Chrome OS sepertinya dibuat bukan hanya untuk menegaskan bahwa Google sudah beranjak dewasa, tapi juga untuk mengusik kenyamanan Microsoft. Layaknya browser Chrome, Chrome OS akan dibuat dengan cita rasa open source. Tidak sabar mendengar apa yang akan dikatakan microsoft setelah mendengar rencana Google ini, tapi apapun komentar mereka, Chrome OS bukan hanya sesuatu yang menarik atau hebat, tapi revolusioner.

KELEBIHAN DAN KEKURANGAN GOOGLE CHROME

10 Seperti yang kita ketahui Google yang terkenal dengan Search Enginenya sudah memiliki sebuah aplikasi browser yang dinamakan CHROME, (sekarang sudah merilis vers 3 beta). Saat ini didunia sangat banyak tersedia aplikasi2 browser. seperti Firefox, Opera, IE, Safari, nestcape, dll. Pada postingan ini NEWTEKNOLOGI-G7 akan membahas kekurangan dan kelebihan yang dimiliki oleh Chrome buatan Google. Browser ini bisa dibilang heboh, karena yang mempunyai browser ini adalah si raja poke engine yaitu Google.

KELEMAHAN BROWSER CHROME : PRIVASI : Google menyimpan 2LISENSI : Google sempat mencantumkan pada Terms of Service mereka, bahwa semua

muatan dari pengguna yang hak ciptanya dimiliki oleh pengguna akan diserahkan haknya pada Google. Tapi indicate ini telah dicabut oleh pihak Google.

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

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

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

KELEBIHAN BROWSER CHROME : ..

TAMPILAN : Pada Interface dari Chrome terlihat bahwa google ingin para penggunanya lebih fokus pada web dan melupakan browser yang digunakan. Ini atinya Google Chrome memilii tampilan yang tidak mengusik dan nyaman ketika sedang digunakan

MODUS PENYAMARAN : Pada modus ini memungkinkan para penggunanya dapat mengakses website tanpa meninggalkan jejak. Sehingga cocok digunakan untuk mengkases konten yang berbau porno secara diam-diam he he he

APLIKASI WEB : Google memberikan opsi "Make Application Shortcut" . Dengan underline ini sebuah aplikasi web seperti GMAIL atau Google Teader dapat dijalankan lewat shrtcut pada Desktop atau Start Menu. Sehingga kelihatan seperti sebuah aplikasi lokal

PENGELOLAAN MEMORY : Pada setiap TAB yang dibuka di Chrome memiliki proses yang terpisah, sehingga ketika eror/crash pada salah satu ta tidak akan meyebabkan seluruh browser eror. (ini kekurangan yang terdapat pada FireFox dan browser lainnya)

PENCARIAN : Chrome memiliki fungsi pencarian yang sangat baik. Contohnya, chrome daat mendeteksi ketika pengguna pernah melakukan pencarian di suatu website dan memasukkan website tersebut dalam daftar penyedia pencarian.

ANTI PENIPUAN : Chrome menyediakan kemampuan menebalkan nama domain sebuah website. Contohnya ada sebuah website penipuan beralamatkan ibank.klikbca.d60pc.com, maka d60pc.com akan ditebalkan sehingga akan diharapkan para pengguna sadar bawah itu bukan situs/website resmi KlikBCA

Cara membuat website Jadwal menggunakan PHP Membuat Tampilan awal untuk user

```
<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
<div class="container">
  <div class="carousel-item active ">

    <div class="row pt-5 justify-content-center">
      <div class="col-4">
        <h1>Dosen</h1>
        <p>PEMETAAN NAMA DOSEN UNTUK ROADSHOW KE SMA DI BANDUNG</p>
      </div>
      <div class="col-3 offset-1">
        
      </div>
    </div>
  </div>
</div>
```

```

        </div>
    </div>
</div>

<div class="carousel-item ">

    <div class="row pt-5 justify-content-center">
        <div class="col-4">
            <h1>SMA DI BANDUNG</h1>
            <p>DAFTAR SMA DI BANDUNG</p>
        </div>
        <div class="col-3 offset-1">
            
        </div>
    </div>

</div>

</div>
</div>
<div>
<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-s
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-s
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
</a>
</div>

<!-- akhir carousel-->

<!--Jadwal-->

<section class="Jadwal">
    <div class="container">
        <div class="row p-5 text-center">
            <div class="col">
                <h1> Jadwal</h1>
            </div>
        </div>
    </div>
<!--content-->

    <div class="row p-5 text-center">
        <div class="col-5 bg-warning">

```

```

        <p> About</p>
    </div>

    <div class="col-2">
    </div>

    <div class="col-5 bg-warning">
        <p>Roni Andarsyah ST.,MT</p>
    </div>
</div>

<div class="row p-5 text-center">
    <div class="col bg-warning">
        <p> About</p>
    </div>
    <div class="col bg-warning">
        <p>Muhammad Sharif Miftahuddin</p>
    </div>
</div>

<div class="row p-5 text-center">
    <div class="col bg-warning">
        <p> About</p>
    </div>
    <div class="col bg-warning">
        <p>Fadel Asata Zohari</p>
    </div>
</div>

</div>
</section>

<!-- Akhir jadwal-->

<!-- About -->

<section class="about">
    <div class="container">
        <div class="row p-5 text-center">
            <div class="col">
                <h1> About</h1>
            </div>
        </div>
    </div>

```



```

        <div class="row p-5 text-center">
            <div class="col">
                <p> About</p>
            </div>
            <div class="col">
                <p>Pemetaan Jadwal Roadshow Untuk politeknikpos</p>
            </div>
        </div>

    </div>
</section>

<!-- Akhir About -->

<!-- Forum -->

<section class="forum">
    <div class="container">
        <div class="row p-5 text-center">
            <div class="col">
                <h1> Poltekpos </h1>
            </div>
        </div>
    </div>
</section>

<!-- Akhir Forum -->

<!--Footer-->
<footer class="footer border-top p-5">
    <div class="container">
        <div class="row">

            </div>
        </div>
    </footer>
<!-- akhir footers-->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="js/jquery-3.4.1.min.js"></script>
<script src="js/popper.min.js" ></script>
<script src="js/bootstrap.js"></script>
</body>
</html>

```

10 Issues #2

Pada *issues #2* Membuat tampilan Nama dosen Untuk user

```
<div class="content w3-display-container">

<div class="card container" style="width:300px; height:410px;">
  
  <div class="card-body">
    <h4 class="card-title">Roni Andarsyah</h4>
    <p class="card-text">Pembimbing web</p>
    <a href="#" class="btn btn-primary">See Profile</a>
  </div>
</div>

</div>
<div class="content w3-display-container">

<div class="card container" style="width:300px; height:410px;">
  
  <div class="card-body">
    <h4 class="card-title">Suparno Saputra</h4>
    <p class="card-text">Kaprod D3 MB</p>
    <a href="#" class="btn btn-primary">See Profile</a>
  </div>
</div>

</div>
<div class="content w3-display-container">
</div>
<div class="content w3-display-container">
</div>
<div class="content w3-display-container">
</div>
<div class="content w3-display-container">
</div>
</div>
```

. Membuat tampilan Jadwal untuk user

```
<div class="container">

<div class="row mt-3">
  <input type="text" class="form-control" placeholder="Cari jadwal" name="keyword">
<div class="col-md-6">
<form action="" method="post">
```

```

<div class="input-group">
  <div class="input-group-append">
    <button class="btn btn-outline-primary" type="submit">Cari</button>
  </div>
</div>
</form>
</div>
</div>

<div class="row mt-5">
<div class="col-md-9">

<table class="table table-bordered">
  <thead class="thead-dark">
    <tr>
      <th scope="col">nama</th>
      <th scope="col">tanggal</th>
      <th scope="col">jabatan</th>
      <th scope="col">tujuan</th>
    </tr>
  </thead>

  <?php foreach( $dosen as $dsn ) : ?>
    <tbody>
      <tr>
        <td><?= $dsn['Nama']; ?></td>
        <td><?= $dsn['Tanggal']; ?></td>
        <td><?= $dsn['Jabatan']; ?></td>
        <td><?= $dsn['Tujuan']; ?><a href="<?= base_url(); ?>Jadwal/ubahjadwal/<?= $dsn['Id_a

      </tr>
    </tbody>
  <?php endforeach; ?>
</table>

<div class="row mt-3">
<div class="col-md-6">
<a href="<?= base_url('jadwal/tambah'); ?>" class="btn btn-primary">Tambah Jadwal</a>
</div>
</div>

</div>
</div>

```

Membuat Tampilan Login menggunakan multi level user

```
<div class="lgn">
<center> <div class="sub">
  <div class="sub2 container" style="background-image: url('<?= base_url();?>src/img/slidesh...
  <div class="container">
    <div class="col-md-4 col-md-offset-4">
      <form class="form-signin" action="<?php echo site_url('login/auth');?>" method="pos
        <h2 class="form-signin-heading">Please sign in</h2>
        <?php echo $this->session->flashdata('msg');?>
        <label for="username" class="sr-only">Username</label>
        <input type="email" name="email" class="form-control" placeholder="Email" require
        <label for="password" class="sr-only">Password</label>
        <input type="password" name="password" class="form-control" placeholder="Passwor
        <div class="checkbox">
          <label>
            <input type="checkbox" value="remember-me"> Remember me
          </label>
        </div>
        <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
      </form>
    </div>
  </div> <!-- /container -->

  <script src="<?php echo base_url('assets/js/bootstrap.min.js');?>"></script>
</div>
</div></center>
</div>
```

11 Issues #5

Pada *issues #5* (*The Program Will Wait Until User Scan Barcode*) fungsi ini digunakan untuk agar *program* menunggu admin sampai memindai kode *barcode*, agar program tidak *error* sampai kode *barcode* muncul dan admin memindainya.

```
def waitLogin(self):
    self.target = '_3RWII'
    self.x_arg = '//div[contains(@class, ' + self.target + ')]'
    self.wait = WebDriverWait(self.driver, 600)
    self.wait.until(EC.presence_of_element_located((By.XPATH, self.x_arg)))
```

Membuat tampilan awal untuk admin

```
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<div class="content w3-display-container" style="width:100%; height:600px;">
  <div class="content1 w3-display-container" style="width:200px; height:600px; backgr
  <div class="subc w3-display-container" style="width:100%; height:100px; padding-to
```

```

<center><a class="nav-link" href="<?= base_url(); ?>userhome"><h3>Home</h3></a></center>

</div>
<div class="subc w3-display-container" style="width:100%; height:100px; margin-top:10px;">

<center><a class="nav-link" href="<?= base_url(); ?>jadwal"><h3>Jadwal</h3></a></center>

</div>
<div class="subc w3-display-container" style="width:100%; height:100px; margin-top:10px;">

<center><h3>Nama</h3></center>

</div>
<div class="subc w3-display-container" style="width:100%; height:100px; margin-top:10px;">

</div>

</div>

<div class="content2 w3-display-container" style="width:1240px; height:600px; background-color:#f0f0f0;">

    <div class="row pt-5 justify-content-center">
        <div class="col-4">
            <h1>Dosen</h1>
            <p>PEMETAAN NAMA DOSEN UNTUK ROADSHOW KE SMA DI BANDUNG</p>
        </div>
        <div class="col-3 offset-1">
            
        </div>
    </div>
</div>

```

Dalam tampilan User kita dapat menggunakan bootstrap carousel begini contoh codingannya

```

<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
    <div class="container">
        <div class="carousel-item active">

            <div class="row pt-5 justify-content-center">
                <div class="col-4">
                    <h1>Dosen</h1>
                    <p>PEMETAAN NAMA DOSEN UNTUK ROADSHOW KE SMA DI BANDUNG</p>
                </div>
            </div>

```

```

        <div class="col-3 offset-1">
            
        </div>
    </div>
</div>

<div class="carousel-item ">

    <div class="row pt-5 justify-content-center">
        <div class="col-4">
            <h1>SMA DI BANDUNG</h1>
            <p>DAFTAR SMA DI BANDUNG</p>
        </div>
        <div class="col-3 offset-1">
            
        </div>
    </div>

</div>
</div>
</div>
<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-s
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-s
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
</a>
</div>

<!-- akhir carousel-->

<!--Jadwal-->

<section class="Jadwal">
    <div class="container">
        <div class="row p-5 text-center">
            <div class="col">
                <h1>Jadwal</h1>
            </div>
        </div>
    </div>
<!--content-->

```

```

<div class="row p-5 text-center">
  <div class="col-5 bg-warning">
    <p> About</p>
  </div>

  <div class="col-2">
  </div>

  <div class="col-5 bg-warning">
    <p>Roni Andarsyah ST.,MT</p>
  </div>
</div>

<div class="row p-5 text-center">
  <div class="col bg-warning">
    <p> About</p>
  </div>
  <div class="col bg-warning">
    <p>Muhammad Sharif Miftahuddin</p>
  </div>
</div>

<div class="row p-5 text-center">
  <div class="col bg-warning">
    <p> About</p>
  </div>
  <div class="col bg-warning">
    <p>Fadel Asata Zohari</p>
  </div>
</div>

</div>
</section>

<!-- Akhir jadwal-->

<!-- About -->

<section class="about">
  <div class="container">
    <div class="row p-5 text-center">
      <div class="col">
        <h1> About</h1>
      </div>
    </div>
  </div>

```

```

    </div>

    <div class="row p-5 text-center">
        <div class="col">
            <p> About</p>
        </div>
        <div class="col">
            <p>Pemetaan Jadwal Roadshow Untuk politeknikpos</p>
        </div>
    </div>

</div>
</section>

<!-- Akhir About -->

<!-- Forum -->

<section class="forum">
    <div class="container">
        <div class="row p-5 text-center">
            <div class="col">
                <h1> Poltekpos </h1>
            </div>
        </div>
    </div>
</section>

<!-- Akhir Forum -->

<!--Footer-->
<footer class="footer border-top p-5">
    <div class="container">
        <div class="row">

            </div>
        </div>
    </footer>
<!-- akhir footers-->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="js/jquery-3.4.1.min.js"></script>
<script src="js/popper.min.js" ></script>
<script src="js/bootstrap.js"></script>

```



```

</body>
</html>

```

Membuat Foooter Untuk tampilan user

```

<footer class="footer border-top p-5">
  <div class="container">
    <div class="row">

      </div>
    </div>
  </footer>
<!-- akhir footers-->

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="js/jquery-3.4.1.min.js"></script>
  <script src="js/popper.min.js" ></script>
  <script src="js/bootstrap.js"></script>
</body>
</html>

```

Membuat tombol hapus jadwal yang berguna untuk menghapus jadwal tombol ini hanya bisa diakses oleh admin

```

?>Jadwal/Hapusjadwal/<?=$dsn['Id_acara']; ?>" class="badge badge-danger float-right" on

```

PERANCANGAN WEBSITE INFORMASI Jadwal Roadshow(JRS) BUKU LAPORAN

Program Studi DIV Teknik Informatika Oleh Woro Isti Rahayu, ST., M.T
105.79.681 Rayhan Yuda Lesmana 1.18.4.007 Nur Hanifah Amatullah 1.18.4.086
PROGRAM DIPLOMA IV TEKNIK INFORMATIKA
BAB I PENDAHULUAN

1.1 Latar Belakang Masalah Pariwisata merupakan salah satu potensi sumber daya yang dapat dikembangkan oleh daerah serta dapat meningkatkan devisa bagi negara. Banyak kota di Indonesia yang berpotensi sebagai tempat objek wisata dan dapat menarik para wisatawan, salah satu contohnya yaitu kota Bandung. Bandung terkenal dengan tempat-tempat wisata yang banyak dikunjungi oleh banyak wisatawan baik luar kota maupun luar negeri. Dengan banyaknya wisatawan yang ingin berlibur di Kota Bandung diperlukan informasi mengenai tempat-tempat wisata di Kota Bandung ini, akan tetapi dalam pencarian informasi tempat wisata di Kota Bandung ini wisatawan sulit mengetahui informasi mengenai tempat wisata yang lebih update dan terbaru. Salah satu contohnya wisatawan yang sudah pernah beberapa kali ke Kota Bandung ingin mengunjungi tempat-tempat wisata yang lain atau tempat wisata yang baru di Kota Bandung ini, sedangkan jika dicari di google website yang keluar masih itu-itu aja dan banyaknya tempat wisata yang terbaru belum dikeluarkan dalam website. Contoh lainnya yaitu ada seseorang yang ingin berkunjung ke salah

satu tempat wisata di Kota Bandung, namun tidak terdapat informasi lengkap mengenai tempat wisata tersebut. Jadi banyak keinginan wisatawan yang belum terpenuhi. Karena hal tersebut, maka dibuatlah suatu sistem informasi yang dapat merekomendasikan tempat-tempat wisata di Kota Bandung, sehingga dapat memenuhi keinginan wisatawan yang ingin berlibur di Kota Bandung. Diharapkan dengan adanya sistem informasi mengenai tempat-tempat wisata ini dapat membantu wisatawan lain untuk berkunjung ke tempat-tempat wisata yang lebih baru di Kota Bandung ini.

1.2 Identifikasi Masalah Berdasarkan latar belakang masalah diatas, dapat dituliskan identifikasi masalahnya yaitu :

1. Bagaimana pencarian informasi yang lebih update dan terbaru tentang tempat-tempat wisata di Kota Bandung ?
2. Bagaimana mendapatkan aplikasi yang dapat memberikan informasi wisata mengenai tempat-tempat wisata di Kota Bandung ?

1.3 Tujuan Tujuan dari pembuatan aplikasi ini yaitu :

1. Menyediakan informasi yang lebih update dan terbaru mengenai tempat-tempat wisata di Kota Bandung.
2. Membuat aplikasi yang mempermudah wisatawan dalam mencari lokasi tempat-tempat wisata di Kota Bandung.

1.4 Ruang Lingkup Ruang lingkup pada penulisan ini yaitu :

1. Mempermudah wisatawan dalam melakukan pencarian tempat-tempat wisata di Kota Bandung
2. Membuat suatu aplikasi berbasis website yang dapat membantu wisatawan yang berkunjung ke Bandung untuk dapat mendapatkan informasi yang lebih update dan terbaru mengenai tempat-tempat wisata di Kota Bandung.

1.5 Sistematika Penulisan Sistematika penulisan merupakan materi-materi yang akan dibahas dalam makalah. Berikut adalah susunan urutan sistematika penulisan dalam makalah ini:

BAB I PENDAHULUAN Bab ini akan membahas mengenai: Latar Belakang Masalah, yaitu menjelaskan secara umum mengenai analisis yang akan dibahas pada makalah ini serta metode yang digunakan untuk menyelesaikan masalah tersebut. Selanjutnya identifikasi Masalah, yaitu menjelaskan apa saja yang akan dibahas dalam makalah ini, Dan tujuan, yaitu menjelaskan tujuannya dibuat makalah ini. Serta ruang Lingkup, yaitu membatasi masalah yang akan dibahas agar tidak rancu. Sistematika Penulisan, untuk menjelaskan pokok-pokok pembahasan.

BAB II LANDASAN TEORI Bab ini akan menjelaskan tentang teori-teori yang mendukung makalah yang dibuat dapat berupa dari jurnal, buku dan lain-lain.

BAB III METODELOGI PENELITIAN Dalam bab ini akan menjelaskan tentang metode yang dipakai pada saat penelitian.

BAB IV ANALISIS DAN PERANCANGAN Dalam bab ini akan menjelaskan tentang metode yang terdapat pada sistem dan menganalisis proses bisnis dari sistem beserta usecase diagram, activity diagram, sequence diagram, class diagram, dan statechart diagram.

BAB V IMPLEMENTASI Dalam bab ini dapat mengimplementasikan dari bab-bab sebelumnya.

BAB VI KESIMPULAN DAN SARAN Pada bab ini berisikan kesimpulan dan saran dari penelitian yang diteliti.

BAB II LANDASAN TEORI

Adapun teori yang digunakan untuk mendukung fitur yang akan dirancang, bertujuan untuk memecahkan masalah yang dihadapi saat penelitian diantaranya :

2.1 Tinjauan Penelitian Penelitian ini tidak terlepas dari hasil penelitian-penelitian terdahulu yang pernah dilakukan sebagai tolak ukur dan

acuan. Adapun hasil penelitian-penelitian yang dijadikan tolak ukur dan acuan tidak terlepas dari topik yang dilakukan. Penelitian pertama yang berjudul “Perancangan Sistem Rekomendasi Paket Wisata dengan Metode Keyword Matching(Study kasus PT. XYZ)”. Pada penelitian ini mengangkat study kasus pada PT. XYZ, dimana PT. XYZ ini merupakan sebuah perusahaan tour yang menyediakan wisata tujuan Bandung. Tujuan diciptakan sistem ini yaitu PT. XYZ ini ingin mengembangkan layanannya dan menyediakan paket wisata yang dapat dikustomisasi sehingga dapat memenuhi permintaan wisatawan yang datang, karena selama ini PT. XYZ hanya menyediakan paket wisata yang standar yang ternyata kurang dapat memenuhi permintaan wisatawan. Oleh karena itu dipenelitian ini dibuatlah sistem yang menggunakan metode Keyword Matching untuk melakukan pemilihan tempat wisata yang sesuai dengan kategori yang diinginkan wisatawan. Dengan metode ini wisatawan dapat mengetahui tempat wisata yang popularitas, durasi perjalanan serta jam operasionalnya tempat tersebut[1]. Penelitian kedua yaitu “Perancangan Aplikasi GIS Pencarian Rute Terpendek Peta Wisata di Kota Manado Berbasis Mobile Web dengan Algoritma Dijkstra”. Tujuan dari penelitian ini yaitu membangun sebuah sistem informasi geografis, yang diharapkan membantu wisatawan dari luar kota dalam menampilkan rute perjalanan dari satu tempat wisata ke tempat wisata lainnya. Metode yang digunakan dalam menerapkan sistem informasi geografis ini adalah dengan metode Graf, dengan menerapkan algoritma terpendek yaitu menggunakan Algoritma Dijkstra, selain itu juga menggunakan Haversine Formula dalam mengakulasikan jarak[2]. Penelitian ketiga berjudul “Rancangan Bangun Sistem Informasi Geografis Parawisata dan Industri Berbasis Web”. Tujuan dari penelitian ini yaitu untuk dapat mengetahui informasi parawisata dan industri rumah tangga di sekitar wilayah tersebut, serta juga dapat mengetahui berbagai macam event-event pariwisata, kebudayaan, industri, akomodasi dan kuliner yang ada di daerah tersebut. Serta dengan adanya tambahan fitur detail informasi dan fitur search ini dapat memudahkan wisatawan dalam mencari informasi dan pencarian datanya. Dalam penelitian ini metode penelitian yang digunakan adalah Rational Unified Proses dengan menggunakan pemodelan Unified Modelling Language, serta menggunakan metode BlackBox Testing sebagai pengujian aplikasi[3]. Penelitian selanjutnya yaitu “Penerapan Model View Controller (MVC) dengan Framework Codeigniter Pada Sistem Booking Wisata Klangon”. Penelitian ini bertujuan untuk mengkonversikan sistem manajemen di klangon desa wisata dari sistem manual ke sistem terkomputerisasi. Dengan perkembangan perangkat lunak teknologi yang akan dibuat dalam bentuk kerangka PHP yaitu kerangka CI yang bertujuan untuk memudahkan dalam mengembangkan aplikasi. Struktur kerangka CI ini telah menerapkan konsep pemrograman berorientasi objek dan pendekatan (MVC), serta penelitian ini menggunakan metode waterfall dalam pengembangannya. Hasil dari penelitian ini yaitu berupa pemesanan tour klangon secara online, yang memberikan informasi tentang paket tour wisata di klangon serta harga paketnya, dimana wisatawan dapat memilih paket wisata yang tersedia secara online[4]. Penelitian berikutnya yaitu “Sistem Pendukung Keputusan Pemilihan Objek Wisata dengan Metode Analitical Hierarchie Proses (AHP) di Kabupaten Rejang Lebong

Berbasis Website Dan Virtual Reality 360 (Study Kasus Objek Wisata Rejang Labong)". Tujuan penelitian ini yaitu membangun sistem pendukung keputusan untuk pemilihan objek wisata dengan mengimplementasikan metode Analitical Hierarchie Proses dan memberikan gambar virtual reality yang bertujuan untuk dapat lebih leluasa melihat yang ada disekitar objek wisata secara virtual[5]. 2.2 Tinjauan Pustaka 2.2.1 Sistem Sistem merupakan suatu komponen-komponen yang berupa subsistem atau bagian-bagian yang membentuk suatu prosedur atau bagan pengolahan. Komponen-komponen sistem ini saling berhubungan dan bekerja sama untuk menciptakan satu kesatuan sehingga dapat menghasilkan suatu informasi[6]. 2.2.2 Informasi Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimaan dan bermanfaat dalam mengambil keputusan saat ini atau mendatang. Atau informasi juga dapat dikatakan sebagai produk dari komunikasi[7]. 2.2.3 Definisi Sitem Informasi Sistem informasi adalah kumpulan elemen-elemen atau komponen-komponen yang saling berinteraksi dan berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses, menyimpan dan mendistribusikan informasi serta dapat membentuk aliran informasi yang akan mendukung dan melakukan control terhadap jalannya suatu perusahaan[8]. 2.2.4 Website Website adalah suatu halaman web yang saling berhubungan yang umumnya berisikan kumpulan informasi berupa data teks, gambar, animasi, audio, video maupun gabungan dari semuanya[9]. 2.2.5. Database Database merupakan suatu kumpulan dari data-data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras(hardware) komputer dan digunakan perangkat lunak(software) untuk memanipulasinya. Data disimpan didalam database untuk keperluan penyediaan informasi, diorganisasikan untuk efisiensi kapasitas penyimpanan supaya informasi yang dihasilkan berkualitas. Database diakses atau dimanipulasikan dengan menggunakan software Database Management Sistem (DBMS)[10]. 2.2.6. PHP PHP adalah kependekan dari PHP Hypertext Preprocessor, bahasa interpreter yang mempunyai kemiripan dengan bahasa C dan perl yang mempunyai kesederhanaan dalam perintah, yang digunakan untuk pembuatan aplikasi web[11]. 2.3 Metodologi Pengembangan Sistem Metodologi pengembangan yang dipakai peneliti dalam penelitian ini yaitu Metode Waterfall. Metode waterfall ini merupakan pengerjaan dari suatu sistem yang dilakukan secara berurutan atau secara terstruktur[12].

Gambar Tahapan Model Waterfall

Dari metode penelitian ini meliputi beberapa proses diantaranya yaitu: a) Analisis Kebutuhan Analisis kebutuhan adalah penguraian dari suatu sistem yang utuh kedalam kegiatan-kegiatan komponennya, dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Pengumpulan data pada tahapan ini yaitu dengan study literature. b) Desain Sistem (Design) Desain Sistem merupakan tahapan perancangan dari sistem yang akan dibuat yang berupa solusi dari permasalahan yang ada dengan menggunakan pemodelan Unified Modeling Language (UML) seperti usecase diagram, activity diagram, sequence diagram, class diagram dan statechart diagram. c) Penulisan Kode Program (Coding) Pada tahapan ini

yaitu proses menerjemahkan hasil proses perancangan menjadi sebuah bentuk program komputer yang dimengerti oleh mesin komputer. d) Pengujian Program (Testing) Tahapan ini merupakan tahapan terakhir yang mana pada tahapan ini sistem yang telah dibuat akan diuji kerjanya sehingga dapat mengetahui kekurangan dari sistem dan dapat melakukan perbaikan pada sistem agar menjadi lebih baik. e) Penerapan dan Pemeliharaan Program Pada tahapan ini yaitu untuk mengimplementasikan sistem yang telah dibuat dan untuk membenarkan kesalahan atau kelemahan sistem yang tidak terdeteksi pada saat pengujian serta untuk membuat sistem up to date.

BAB III METODELOGI PENELITIAN 3.1 Diagram Alur Metodologi Penelitian Metode pengembangan sistem yang digunakan yaitu metode SDLC (Sistem Development Life Cycle) atau sering disebut sebagai pendekatan air terjun (waterfall). Metode waterfall ini merupakan pengerjaan dari suatu sistem yang dilakukan secara berurutan atau secara terstruktur. Alasan menggunakan metode waterfall ini yaitu metode pengerjaannya yang terstruktur dan tidak berfokus pada tahap tertentu. Jadi jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya. Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan [13].

Gambar Tahapan Model Waterfall

3.2 Tahapan – Tahapan Diagram Alur Metodologi Tahapan pada metodologi waterfall ini antara lain: 3.2.1 Analisis Kebutuhan Tahapan awalnya yaitu menganalisis dan mengumpulkan kebutuhan perangkat lunak sesuai dengan yang diinginkan pengguna. Disini terlebih dahulu kita mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang terjadi. Setelah dianalisis maka dapat memutuskan software yang akan digunakan dan sesuai dengan pengguna inginkan. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Dokumen ini lah yang akan menjadi acuan sistem analisis untuk menerjemahkan ke dalam bahasa pemrograman. 3.2.2 Desain Sistem (Design) Pada tahapan desain dilakukan pembuatan pemodelan sistem berdasarkan hasil analisis kebutuhan yang telah diputuskan bersama. Pemodelan sistem yang digunakan yaitu Unified Modelling Language (UML) yang terdiri dari beberapa diagram diantara lain Use-Case Diagram, Sequence Diagram, Activity Diagram, Class Diagram, dan Statechart Diagram. 3.2.3 Penulisan Program atau Pengkodean (Coding) Tahapan ini merupakan implementasi dari tahapan desain yang akan menghasilkan sebuah prototype perangkat lunak. Dalam pembangunan website ini menggunakan bahasa pemrograman PHP serta mengimplementasikan frame CI pada website ini. 3.2.4 Pengujian Program (Testing) Ujicoba software merupakan tahapan akhir dimana sistem yang dibuat diuji kemampuan dan keefektifannya sehingga dapat mengetahui kekurangan dan kelemahan dari sistem yang kemudian akan dilakukan perbaikan terhadap sistem atau aplikasi supaya menjadi lebih baik lagi. 3.2.5 Penerapan dan Pemeliharaan Sistem Tahap Penerapan (implementasi) adalah tahap dimana semua elemen dan aktivitas sistem disatukan dan dapat diimplementasikan oleh pengguna. Selanjutnya Tahapan Pemeliharaan Sistem (Maintenance) yaitu untuk membenarkan kesalahan atau kelemahan sistem yang tidak terdeteksi pada

saat pengujian serta untuk membuat sistem up to date.

BAB IV ANALISIS DAN PERANCANGAN

4.1 Analisis Tahap analisis merupakan tahap penelitian dengan melakukan suatu percobaan yang menghasilkan kesimpulan dari penguraian suatu sistem aplikasi, sehingga dapat diketahui mekanisme sistem, masalah-masalah yang terjadi. Dari proses penelitian tersebut, dapat diusulkan perbaikan-perbaikan yang dapat membangun dan mempertinggi sistem kinerja alat yang akan dibuat. Analisis sistem yang akan dibangun disesuaikan dengan kebutuhan, berdasarkan hasil evaluasi. Pada bagian ini, akan dibahas mengenai perancangan website informasi tempat wisata di Kota Bandung. 4.2 Perancangan Berikut ini adalah suatu gambar analisa data pada “Perancangan website informasi tempat wisata di Kota Bandung” menggunakan notasi UML (Unified Modeling Language). 4.2.1 Use Case Diagram Use case diagram adalah diagram yang menunjukkan suatu kelompok use case dan actors serta relationships-nya.

Gambar Usecase Diagram Scenario Use Case : Tabel 4. 1 Definisi Use Case No Use Case Deskripsi 1. Login Masuk ke dalam sistem 2. Melihat jadwal Melihat Jadwal yang ada 3. Mencari jadwal Mencari Jadwal yang ada 4. Mengedit jadwal Ketika admin menambahkan jadwal namun jadwal yang sudah ada ingin diubah bisa menggunakan fitur edit 5. Menghapus Jadwal Admin dapat menghapus Jadwal yang sudah dilaksanakan 6. Menambahkan jadwal Admin dapat menambahkan jadwal 7. Melihat/Mendapatkan info destinasi wisatanya Mendapatkan info destinasi objek wisata yang ditelusuri

Tabel 4.2 Aktor pada Model Use Case No. Aktor Peran 1. Admin Yang mengelola seluruh proses dalam sistem. 2. User(wisatawan) Aktor yang menggunakan website/sistem tersebut

Tabel 4. 3 Skenario Use Case Login No Uji : UC 01 Aksi Aktor Reaksi Sistem Skenario normal 1. Memasukkan username dan password 2. Memvalidasi username dan password user 3. Sistem menampilkan form main menu Skenario Alternatif : Validasi user name dan password gagal 1. User memasukkan username dan password yang salah, username dan password tidak lengkap 1. Tidak dapat masuk ke form main menu

Tabel 4. 4 Skenario Use Case Menambahkan Jadwal No Uji : UC 02 Aksi Aktor Reaksi Sistem Skenario normal 1 Memilih fitur Tambah jadwal 2. Menampilkan form Tambah Jadwal dan akan ditambahkan pada database Skenario alternatif 1. Salah memasukkan data Jadwal 2. Sistem akan menampilkan kesalahan pada kategori jadwal yang salah

Tabel 4. 5 Skenario Use Case Cari Jadwal No Uji : UC 03 Aksi Aktor Reaksi Sistem Skenario normal 1. Memilih fitur cari jadwal 2. Menampilkan jadwal yang dicari Skenario Alternatif 1. Memasukkan data yang salah pada pencarian 2. Sistem tidak akan memberikan data pada pencarian

Tabel 4. 6 Skenario Use Case menghapus jadwal No Uji : UC 04 Aksi Aktor Reaksi Sistem Skenario normal

1. Cara Instal Aplikasi XMPP Pada Windows 10

Untuk dapat menginstal aplikasi XAMPP harus memiliki file instalasi XAMPP terlebih dahulu. Jika belum memiliki file instalasinya maka harus mengunduhnya terlebih dahulu pada website <https://www.apachefriends.org/download.html>.

Setelah membuka website tadi maka akan muncul tampilan sebagai berikut:

Setelah muncul tampilan seperti diatas klik download seperti pada gambar berikut:

Setelah klik download tunggu hingga proses download selesai. Jika sudah selesai proses download didalam folder tempat tersimpannya file download, akan terdapat file intallsai XAMPP seperti gambar dibawah:

Sekaranglah untuk proses install XAMPP, caranya klik 2 kali pada file installasi XAMPP tersebut. Jika tampil UAC (User Account Control) seperti gambar dibawah silakan klik Yes.

Selanjutnya, jika muncul warning tentang UAC XAMPP seperti gambar dibawah silakan klik OK saja.

Maka akan muncul setup XAMPP seperti gambar dibawah, silakan klik Next.

Selanjutnya akan tampil beberapa pilihan komponen XAMPP, pada pilihan ini biarkan default saja. Yaitu pada tutorial ini semua pilihan komponen XAMPP di ceklis semua, untuk detailnya bisa dilihat contoh gambar dibawah. Dan selanjutnya silakan klik Next.

Maka akan tampil pilihan folder installasi XAMPP, pada pilihan ini juga saya biarkan default, yaitu pada directori C:. Selanjutnya klik Next.

Kemudian akan tampil bitnami for XAMPP. Pada langkah ini kamu boleh ceklis atau tidak pada Learn more about Bitnami for XAMPP. Apabila kamu ceklis maka setelah klik next akan tampil sebuah halaman web tentang Bitnami for XAMPP, dan begitu juga dengan sebaliknya. selanjutnya klik Next.

Dan akan tampil XAMPP ready to Install, selanjutnya klik Next.

Maka proses install XAMPP dimulai, silakan tunggu hingga proses installasi selesai.

Jika install XAMPP sudah selesai maka akan tampil seperti gambar dibawah. Yang memberitahu bahwa XAMPP sudah selesai di install. Silakan ceklis Do you want to start the Control Panel now?, untuk membuka XAMPP Control Panel. Dan selanjutnya klik Finish.

Pengisntalan XAMPP selesai. 2. Cara Install Codeigniter Silahkan klik link <https://codeigniter.com/> lalu download framework codeigniter terbaru, saat ini yang terbaru adalah versi 3.1.11, dan akan terus berlanjut ke versi berikutnya.

Jika proses download selesai maka akan mendapatkan sebuah file bernama CodeIgniter-3.1.11.zip, lalu klik kanan pada file ekstrak file tersebut.

setelah itu akan muncul folder bernama CodeIgniter-3.1.11.

silahkan anda rename menjadi codeigniter lalu salin folder tersebut ke folder root anda, yaitu di htdocs.

Pada tahapan ini kita sudah selesai memasang codeigniter pada xampp localhost di komputer kita. Selanjutnya kita akan mengetest apakah codeigniter sudah bisa berjalan dan digunakan atau tidak melalui browser. Buka XAMPP lalu aktifkan apache

Kemudian buka web browser pada tutorial ini menggunakan google chrome. Lalu kita panggil alamat localhost pada URL browser dengan ditambahkan nama file folder codeigniter yang sudah kita pindahkan ke dalam folder HTDOCS di XAMPP.

Cara Install Codeigniter Di Xampp

Codeigniter adalah sebuah open source framework PHP dengan basis utama MVC (model,view dan controller). Dengan codeigniter para developer website bisa membangun sebuah web dinamis dengan mudah dengan memanfaatkan fungsi,class dan method-method yang sudah disediakan oleh codeigniter.Pada pembahasan kali ini kita akan memulai dengan menginstall codeigniter di xampp yang selanjutnya kita gunakan dalam mendvelop web keinginan kita. Langkah-Langkah Memasang Codeigniter Di Xampp:

Download codeigniter terlebih dahulu versi terbaru 3.0 di www.codeigniter.com

Sebelum memasang codeigniter di xampp pastikan kalian sudah meng install xampp localhost, jika belum maka instal terlebih dahulu. Cara Menginstall XAMPP di Windows Kemudian ekstrak file codeigniter yang berupa .zip menjadi sebuah folder yang nantinya kita pindahkan ke dalam xampp. Lalu pindahkan folder yang sudah di ekstrak tadi ke dalam xampp tepatnya di folder htdocs , secara default folder xampp terinstall di drive C komputer kita (lihat gambar dibawah) Output as PDF file has been powered by [Universal Post Manager] plugin from www.ProfProjects.com — Page 1/2 — This page was exported from - UPT Sumber Daya Informatika Export date: Thu Jan 30 12:52:12 2020 / +0000 GMT Pada tahapan ini kita sudah selesai memasang codeigniter pada xampp localhost di komputer kita. Selanjutnya kita akan mengetest apakah codeigniter sudah bisa berjalan dan digunakan atau tidak melalui browser. Kita aktifkan terlebih dahulu apache pada xampp.

Kemudian buka web browser disini saya gunakan google chorme. lalu kita panggil alamat localhost pada URL browser dengan ditambahkan nama file folder codeigniter yang sudah kita pindahkan ke dalam folder HTDOCS di xampp. Jika sudah tampil seperti gambar no 7 diatas maka codeigniter sudah siap digunakan dalam localhost kita. Demikian penjelasan tentang cara pasang codeigniter pada xampp di komputer kita, selanjutnya bisa kalian gunakan secara offline. source: <http://computernet-news.blogspot.co.id/2016/05/tutorial-cara-install-codeigniter-di-localhost.html>

Gambar 2. Perbandingan PHP Biasa dan CodeIgniter

Perhatikan gambar 2. Pada eksekusi PHP, biasanya kita akan me-“load” semua library dan fungsi yang dibutuhkan kemudian digabungkan ke dalam HTML untuk di eksekusi oleh PHP. Untuk kasus sederhana cara tersebut masih baik-baik saja, tetapi ketika aplikasi tersebut menjadi kompleks/rumit maka kita akan sulit memeliharanya jika tidak didukung oleh arsitektur software yang bagus. Hal tersebut bisa terjadi disebabkan oleh code yang sama namun dibuat berulang- ulang , kode tidak konsisten dan lain-lain.

Gambar 3. Flow CodeIgniter

Jika dipetakan, alur kerja CodeIgniter akan tampak seperti gambar 3. Browser berinteraksi melalui controller. Controller-lah yang akan menerima dan membalas semua request dari browser. Untuk data maka controller akan meminta ke Model dan untuk UI/template akan meminta ke View. Jadi “Otak” dari aplikasi ada di controller, “Muka” aplikasi ada di view dan “Data” ada di model. Ketika

browser meminta sebuah halaman web maka router akan mencari con-

troller mana yang harus menangani request tersebut. Setelah itu barulah si controller menggunakan model untuk mengakses data dan View untuk menampilkan data tersebut. Jangan Belajar CodeIgniter!!

Jangan coba-coba belajar CodeIgniter ketika Anda belum mengenal PHP. Berdasarkan pengalaman di forum dan milis, kebanyakan mereka yang belajar CodeIgniter tanpa memiliki dasar PHP yang baik akan mengalami banyak kesulitan, bukan dalam menguasai konsep CodeIgniter tetapi masih berkutat seputar PHP. Setidaknya Anda telah memahami konsep OOP pada PHP untuk mulai belajar CodeIgniter. Sebaiknya ketika Anda mempelajari CodeIgniter Anda sudah memahami PHP dan Object Oriented Programming. Akan lebih bagus lagi jika Anda sudah familiar dengan design pattern.

Chapter 2 PHP Object Oriented Programming Syarat utama untuk menguasai CodeIgniter adalah memahami PHP dan konsep Object Oriented Programming. Pada bab ini kita akan membahas sedikit mengenai PHP dan OOP. Apasih PHP? PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman di sisi server. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server. Misalnya kita request sebuah file PHP <http://www.koder.web/index.php>, maka webserver akan melakukan parsing terhadap file PHP tersebut. PHP parser yang menjalankan kode-kode PHP yang terdapat pada file index.php lalu mengirimkan hasilnya ke web browser. Sejarah PHP Pada tahun 1994, Rasmus Lerdorf mengembangkan sebuah perangkat yang digunakan sebagai engine parsing sebagai penerjemah/interpreter beberapa macro. Pada saat itu engine digunakan untuk pembuatan buku tamu, counter dan beberapa homepage. Ia menamai engine parser tersebut dengan nama PHP/FI. Dengan semangat opensource, para pengembang di dunia mencoba mengembangkan PHP/FI. Sampai pada tahun 1997, lebih dari 500.000 website di dunia menggunakan PHP/FI untuk menyelesaikan masalah seperti koneksi ke database, menampilkan content yang dinamis dan lain-lain. Pada juni 1998, PHP 3.0 dirilis. Pada saat itu PHP sudah mendukung multplatform (bukan hanya linux), webserver, sejumlah database, SNMP (Simple Network Management Protocol) and IMAP (Internet Message Access Protocol). Menurut survei yang dilakukan oleh <http://netcraft.org>, saat ini pengguna PHP sudah mencapai 9,5 juta domain. Menggunakan PHP Ketika kita ingin mempelajari PHP, hal pertama yang harus disiapkan adalah sebuah web server, seperti yang sudah dibahas sebelumnya bahwa PHP merupakan bahasa pemrograman disisi server. Banyak sekali jenis web server yang dapat digunakan, antara lain Apache, IIS, iPlanet, Omni, Xintami, dan lain-lain. Setelah web server terinstal barulah kita install PHP sebagai modul.

PHP dapat diperoleh secara gratis dengan cara men-download dari situs resmi PHP (<http://www.php.net/downloads.php>) atau website lain yang mempunyai salinan program PHP untuk di-download. Sebelum menginstall PHP, terlebih dahulu harus meng-install web server. Setelah PHP terinstall. Anda dapat meletakkan source code PHP ke dalam folder yang akan di publish oleh web server dengan ekstensi file .php. Jika tidak ingin direpotkan oleh proses konfigurasi dan implementasi webserver, Anda dapat menggunakan software yang sudah terpaket menjadi satu antara PHP, APACHE, dan MySQL. Contoh

Software-nya adalah XAMPP, PHPTRIAID, FOX Server, dan lain-lain.

Apa sih Object Oriented Programming (OOP)? Object Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada obyek. Semua data dan fungsi pada paradigma ini dibungkus dalam kelas-kelas atau obyek-obyek. Bandingkan dengan logika pemrograman terstruktur, setiap obyek dapat menerima pesan, memproses data, dan mengirim pesan ke obyek lainnya. OOP diciptakan untuk mengatasi keterbatasan pada bahasa pemrograman tradisional. Konsep dari OOP sendiri adalah semua pemecahan masalah dibagi ke dalam obyek. Dalam konsep OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai obyek. Apa itu Object Sederhananya, sebuah obyek adalah kumpulan dari variabel dan fungsi yang dibungkus menjadi satu entitas. Entitas tersebut dapat berupa variabel biasa. Sebuah obyek diciptakan melalui sebuah kelas atau dengan istilah *instance of class*. Obyek memiliki 2 elemen utama: 1. *Attributes* atau *Properties*: Yaitu nilai-nilai yang tersimpan dalam objek tersebut dan secara langsung maupun tidak langsung menentukan karakteristik dari obyek tersebut. 2. *Method*: Yaitu suatu aksi yang akan dijalankan atau dikerjakan oleh obyek tersebut.

Apa itu Class? Class dapat didefinisikan sebagai struktur data atau cetak biru dari suatu obyek. Lebih jelasnya adalah sebuah bentuk dasar atau blueprint yang mendefinisikan variabel, method umum pada semua obyek. Obyek sendiri adalah kumpulan variabel dan fungsi yang dihasilkan dari template khusus atau disebut class. Obyek adalah elemen pada saat run-time yang akan diciptakan, dimanipulasi, dan dibuang/di-destroy ketika eksekusi. Adapun class merupakan definisi statik dari himpunan obyek yang mungkin diciptakan sebagai instantiasi dari class. Perhatikan contoh class di berikut ini.

```
ï¿½?php /** Contoh kelas */ class Kendaraan /** end of class */ ï¿½;
```

Contoh diatas memperlihatkan bagaimana mendefinisikan sebuah class dan meng-create sebuah instance dari class. Pada contoh di atas kita membuat sebuah kelas bernama “Kendaraan”. Dalam pembuatan kelas, pertama kita menggunakan kata kunci class yang diikuti oleh nama kelas, kemudian diakhiri dengan kurung kurawal. Di dalam kurung kurawal kita menuliskan kode-kode (berisi property dan method) supaya kelas tersebut bekerja seperti yang diinginkan. Kode-kode di dalam sebuah kelas terbagi menjadi dua kelompok, yaitu property dan method. Property adalah suatu wadah penyimpanan di dalam kelas yang bisa menampung informasi. Sederhananya property itu bisa disebut sebagai variabel di dalam kelas. Sedangkan method adalah fungsi yang ada di dalam kelas. Perhatikan contoh berikut. ï¿½?php /** contoh kelas */ class kendaraan

```
    /** property class */ private warna; private jumlah_pintu; private jumlahroda; public harga;
    public merk;
    /** method class */ public function construct() echo 'ini adalah object
    kendaraan. ï¿½br ï¿½';
        public function set_harga(harga) this-> harga =harga;
        public function show_harga() echo'hargakendaraan : rp.'.this->harga.'. ï¿½br
    ï¿½';
```

```

    public function jalan() echo 'brrroooooom!!!';
    /*** end of class ***/ /*** contoh object **/ mengcreate object saya_a dalah_o bject dariclass kendaraan saya_a d
newkendaraan; saya_a dalah_o bject - > set_harga(100000); saya_a dalah_o bject - >
show_harga();
    saya_a dalah_o bject - > jalan();

```

Setiap property dan method memiliki identifier. Identifier-lah yang mengatur bagaimana property dan method digunakan. Identifier tersebut adalah public, private dan protected. Private berarti method atau property yang ada di dalam suatu kelas hanya bisa diakses di dalam kelasnya. Sedangkan pada method atau property yang bersifat public berarti method atau property tersebut bisa diakses di dalam dan di luar kelas. Inheritance Inheritance atau dalam bahasa Indonesianya disebut sebagai pewarisan adalah suatu cara untuk membuat sebuah kelas yang baru dengan menggunakan kelas lain yang sebelumnya sudah dibuat. Pada hubungan inheritance, sebuah class turunan mewarisi kelas leluhur (parent class). Oleh karena mewarisi, maka semua atribut dan method class dari induk akan dibawa (kecuali yang bersifat private), secara intrinsik menjadi bagian dari class anak. Adapun keuntungan yang didapat dari inheritance menambah fitur baru pada kelas anak dan mengubah atau mengganti fitur yang diwarisi dari kelas parent Adapun contoh kelas yang menggunakan konsep inheritance adalah

```

i?php /* inheritance.php */ class Bapak private nama = "Bapak"; function Bapak(n)
this - > nama = n; function Hallo() echo "Halo, saya this - > nama < br > ";
class Anak extends Bapak function Hai() Echo "hai dari kelas anak";
test = new Anak("AnakdariBapak"); test -> Hallo();
Pertemuan 1

```

12 Issues #1

Pada *issues #1* Membuat Tampilan awal untuk user

```

<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
<div class="container">
    <div class="carousel-item active ">

        <div class="row pt-5 justify-content-center">
            <div class="col-4">
                <h1>Dosen</h1>
                <p>PEMETAAN NAMA DOSEN UNTUK ROADSHOW KE SMA DI BANDUNG</p>
            </div>
            <div class="col-3 offset-1">
                
            </div>
        </div>
    </div>
</div>

```

```

<div class="carousel-item ">

    <div class="row pt-5 justify-content-center">
        <div class="col-4">
            <h1>SMA DI BANDUNG</h1>
            <p>DAFTAR SMA DI BANDUNG</p>
        </div>
        <div class="col-3 offset-1">
            
        </div>
    </div>
</div>

</div>
</div>
</div>
<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-s
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-s
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
</a>
</div>

<!-- akhir carousel-->

<!--Jadwal-->

<section class="Jadwal">
    <div class="container">
        <div class="row p-5 text-center">
            <div class="col">
                <h1> Jadwal</h1>
            </div>
        </div>
    </div>
<!--content-->

    <div class="row p-5 text-center">
        <div class="col-5 bg-warning">
            <p> About</p>
        </div>

        <div class="col-2">

```

```

        </div>

        <div class="col-5 bg-warning">
            <p>Roni Andarsyah ST.,MT</p>
        </div>
    </div>

    <div class="row p-5 text-center">
        <div class="col bg-warning">
            <p> About</p>
        </div>
        <div class="col bg-warning">
            <p>Muhammad Sharif Miftahuddin</p>
        </div>
    </div>

    <div class="row p-5 text-center">
        <div class="col bg-warning">
            <p> About</p>
        </div>
        <div class="col bg-warning">
            <p>Fadel Asata Zohari</p>
        </div>
    </div>

</div>
</section>

<!-- Akhir jadwal-->

<!-- About -->

<section class="about">
    <div class="container">
        <div class="row p-5 text-center">
            <div class="col">
                <h1> About</h1>
            </div>
        </div>

        <div class="row p-5 text-center">
            <div class="col">
                <p> About</p>
            </div>
        </div>
    </div>

```

```

        <div class="col">
            <p>Pemetaan Jadwal Roadshow Untuk politeknikpos</p>
        </div>
    </div>

</div>
</section>

<!-- Akhir About -->

<!-- Forum -->

<section class="forum">
    <div class="container">
        <div class="row p-5 text-center">
            <div class="col">
                <h1> Poltekpos </h1>
            </div>
        </div>
    </div>
</section>

<!-- Akhir Forum -->

<!--Footer-->
<footer class="footer border-top p-5">
    <div class="container">
        <div class="row">

            </div>
        </div>
    </footer>
<!-- akhir footers-->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="js/jquery-3.4.1.min.js"></script>
<script src="js/popper.min.js" ></script>
<script src="js/bootstrap.js"></script>
</body>
</html>

```

13 Issues #2

Pada *issues #2* Membuat tampilan Nama dosen Untuk user

```
<div class="content w3-display-container">

<div class="card container" style="width:300px; height:410px;">
  
  <div class="card-body">
    <h4 class="card-title">Roni Andarsyah</h4>
    <p class="card-text">Pembimbing web</p>
    <a href="#" class="btn btn-primary">See Profile</a>
  </div>
</div>

</div>
<div class="content w3-display-container">

<div class="card container" style="width:300px; height:410px;">
  
  <div class="card-body">
    <h4 class="card-title">Suparno Saputra</h4>
    <p class="card-text">Kaprod D3 MB</p>
    <a href="#" class="btn btn-primary">See Profile</a>
  </div>
</div>

</div>
<div class="content w3-display-container">
</div>
<div class="content w3-display-container">
</div>
<div class="content w3-display-container">
</div>
<div class="content w3-display-container">
</div>
</div>
.
```

14 Issues #3

Pada *issues #3* Membuat tampilan Jadwal untuk user

```
<div class="container">
```

```

<div class="row mt-3">
<div class="col-md-6">
<form action="" method="post">
<div class="input-group">
    <input type="text" class="form-control" placeholder="Cari jadwal" name="keyword">
    <div class="input-group-append">
        <button class="btn btn-outline-primary" type="submit">Cari</button>
    </div>
</div>
</form>
</div>
</div>

```

```

<div class="row mt-5">
<div class="col-md-9">

```

```

<table class="table table-bordered">
    <thead class="thead-dark">
        <tr>
            <th scope="col">nama</th>
            <th scope="col">tanggal</th>
            <th scope="col">jabatan</th>
            <th scope="col">tujuan</th>
        </tr>
    </thead>

```

```

<?php foreach( $dosen as $dsn ) : ?>
    <tbody>
        <tr>
            <td><?= $dsn['Nama']; ?></td>
            <td><?= $dsn['Tanggal']; ?></td>
            <td><?= $dsn['Jabatan']; ?></td>
            <td><?= $dsn['Tujuan']; ?><a href="<?= base_url(); ?>Jadwal/ubahjadwal/<?= $dsn['Id_a

        </tr>
    </tbody>
<?php endforeach; ?>
</table>

```

```

<div class="row mt-3">
<div class="col-md-6">
<a href="<?= base_url('jadwal/tambah'); ?>" class="btn btn-primary">Tambah Jadwal</a>
</div>
</div>

```


</div>

</div>

</div>

15 Issues #4

Pada *issues #4* Membuat Tampilan Login menggunakan multi level user

```
<div class="lgn">
<center> <div class="sub">
  <div class="sub2 container" style="background-image: url('<?= base_url();?>src/img/slidesh
  <div class="container">
    <div class="col-md-4 col-md-offset-4">
      <form class="form-signin" action="<?php echo site_url('login/auth');?>" method="pos
      <h2 class="form-signin-heading">Please sign in</h2>
      <?php echo $this->session->flashdata('msg');?>
      <label for="username" class="sr-only">Username</label>
      <input type="email" name="email" class="form-control" placeholder="Email" require
      <label for="password" class="sr-only">Password</label>
      <input type="password" name="password" class="form-control" placeholder="Passwor
      <div class="checkbox">
        <label>
          <input type="checkbox" value="remember-me"> Remember me
        </label>
      </div>
      <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
    </form>
  </div>
</div> <!-- /container -->

  <script src="<?php echo base_url('assets/js/bootstrap.min.js');?>"></script>
</div>
</div></center>
</div>
```

16 Issues #5

Pada *issues #5* (*The Program Will Wait Until User Scan Barcode*) fungsi ini digunakan untuk agar *program* menunggu admin sampai memindai kode *barcode*, agar program tidak *error* sampai kode *barcode* muncul dan admin memindainya.

```
def waitLogin(self):
    self.target = '"_3RWII"'
```

```

self.x_arg = '//div[contains(@class, ' + self.target + ')]'
self.wait = WebDriverWait(self.driver, 600)
self.wait.until(EC.presence_of_element_located((By.XPATH, self.x_arg)))

```

17 Issues #6

Pada *issues #6* Membuat tampilan awal untuk admin

```

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<div class="content w3-display-container" style="width:100%; height:600px;">
  <div class="content1 w3-display-container" style="width:200px; height:600px; background-color: #f0f0f0;">
    <div class="subc w3-display-container" style="width:100%; height:100px; padding-top: 10px;">

      <center><a class="nav-link" href="<?= base_url(); ?>userhome"><h3>Home</h3></a></center>

    </div>
    <div class="subc w3-display-container" style="width:100%; height:100px; margin-top: 10px;">

      <center><a class="nav-link" href="<?= base_url(); ?>jadwal"><h3>Jadwal</h3></a></center>

    </div>
    <div class="subc w3-display-container" style="width:100%; height:100px; margin-top: 10px;">

      <center><h3>Nama</h3></center>

    </div>
    <div class="subc w3-display-container" style="width:100%; height:100px; margin-top: 10px;">

      <center><h3>Nama</h3></center>

    </div>
  </div>
</div>
<div class="content2 w3-display-container" style="width:1240px; height:600px; background-color: #f0f0f0;">

  <div class="row pt-5 justify-content-center">
    <div class="col-4">
      <h1>Dosen</h1>
      <p>PEMETAAN NAMA DOSEN UNTUK ROADSHOW KE SMA DI BANDUNG</p>
    </div>
    <div class="col-3 offset-1">
      
    </div>
  </div>
</div>

```

18 Issues #7

Pada *issues #7* Membuat Tampilan jadwal untuk Admin

```
<div class="container">

<div class="row mt-3">
<div class="col-md-6">
<form action="" method="post">
<div class="input-group">
    <input type="text" class="form-control" placeholder="Cari jadwal" name="keyword">
    <div class="input-group-append">
        <button class="btn btn-outline-primary" type="submit">Cari</button>
    </div>
</div>
</form>
</div>
</div>

<div class="row mt-5">
<div class="col-md-9">

<table class="table table-bordered">
    <thead class="thead-dark">
        <tr>
            <th scope="col">nama</th>
            <th scope="col">tanggal</th>
            <th scope="col">jabatan</th>
            <th scope="col">tujuan</th>
        </tr>
    </thead>

<?php foreach( $dosen as $dsn ) : ?>
    <tbody>
        <tr>
            <td><?= $dsn['Nama']; ?></td>
            <td><?= $dsn['Tanggal']; ?></td>
            <td><?= $dsn['Jabatan']; ?></td>
            <td><?= $dsn['Tujuan']; ?><a href="<?= base_url(); ?>Jadwal/ubahjadwal/<?= $dsn['Id_a
```

```

<div class="row mt-3">
<div class="col-md-6">
<a href="<?= base_url('jadwal/tambah'); ?>" class="btn btn-primary">Tambah Jadwal</a>
</div>
</div>
</div>

</div>

</div>

```

19 Issues #8

Pada *issues #8* Membuat Tombol Tambah data yang berguna untuk menambahkan jadwal yang hanya bisa dilakukan oleh admin

```

<div class="row mt-3">
<div class="col-md-6">
<a href="<?= base_url('jadwal/tambah'); ?>" class="btn btn-primary">Tambah Jadwal</a>
</div>
</div>

```

20 Issues #9

Pada *issues #9* Membuat controller tambah jadwal

```

    public function tambah()
    {
        //tambah jadwal
        $data['judul'] = 'Tambah Jadwal';

        $this->form_validation->set_rules('nama_dosen','Nama Dosen','required');
        $this->form_validation->set_rules('jabatan','Jabatan','required');
        $this->form_validation->set_rules('tujuan','Tujuan','required');

        if($this -> form_validation -> run() == false){
            $this->load->view('template/header', $data);
            $this->load->view('jadwal/tambah');
            $this->load->view('template/footer');
        }
        else
        {
            $this -> Dosen_model -> tambah_Data();
            $this->session->set_flashdata('data', 'Ditambah');
        }
    }

```

```

redirect('jadwal');
}
}

```

21 Issues #10

Pada *issues #10* Membuat Header untuk tampilan user

```

<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="<?= base_url(); ?>src/css/bootstrap.css">

  <!-- font awesome-->
  <link rel="stylesheet" href="<?= base_url(); ?>src/css/all.css">

  <!-- google font -->
  <link href="https://fonts.googleapis.com/css?family=Montserrat:200,400,600&display=swap"
  rel="stylesheet">

  <!-- My CSS-->
  <link rel="stylesheet" href="<?= base_url(); ?>src/css/css.css">
  <link rel="stylesheet" href="<?= base_url(); ?>src/srf/design.css">

  <title> <?= $judul ?></title>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark">
    <div class="container">
      <a class="navbar-brand" href="#">
        
      </a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav text-uppercase mx-auto">
          <li class="nav-item active">
            <a class="nav-link" href="<?= base_url(); ?>Home">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="<?= base_url(); ?>namadosen">Nama dosen</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

```

```

        <li class="nav-item">
            <a class="nav-link" href="<?= base_url(); ?>admin">Jadwal</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="<?php echo site_url('login/logout');?>">SIGN OUT
        </li>
    </ul>
</div>
</nav>

```

article [utf8]inputenc float graphicx
 Bukuisbn kopisharif January 2020

22 Penggunaan Aplikasi

Kali ini kami akan memberi tahu kalian nih soal aplikasi yang kami gunakan untuk membuat web ini, kami menggunakan bahasa PHP, dan HTML tak lupa desainnya menggunakan CSS dan database XAMPP dan MySQL, kami juga menggunakan SUBLIME sebagai alat bantu membaca bahasa pemrograman, lalu frameworknya kami gunakan CodeIgniter versi 3.1.11 dan juga Bootstrap versi 4 sebagai Template dan bisa juga digunakan untuk mendesain. Untuk lebih jelasnya akan kami bahas disini.

23 PHP

Apakah PHP itu? PHP (akronim dari PHP: Hypertext Preprocessor) adalah bahasa pemrograman yang berfungsi dalam membuat web yang dinamis maupun aplikasi berbentuk web. Sangat berbeda dengan HTML yang hanya dapat menampilkan konten statis, PHP dapat berinteraksi atau terhubung dengan database, folder maupun file, sehingga membuat bahasa pemrograman PHP bisa menampilkan konten yang dinamis dari Blog, web, Toko Online, Forum, dan Website JRP adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP termasuk dalam bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk dalam bahasa yang cross-platform, yang artinya PHP dapat berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran “.php”.

24 Sejarah PHP

sejarah php Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. 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 perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

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

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 meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessing.

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

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Server web bawaan ditingkatkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstall software server.

Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7.0.16 dan 7.1.2 yang resmi dirilis pada tanggal 17 Februari 2017. Pembatas PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun di luar pembatas tidak diproses oleh PHP (meskipun teks PHP ini masih mengendalikan struktur yang dijelaskan dalam kode PHP). Pembatas yang paling umum adalah `"<?php"` untuk membuka dan `"?>"` untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, JavaScript.

25 Variabel

Variabel diawali dengan simbol dolar. Pada versi PHP 5 diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa mereka menjadi parameter objek dari class tertentu, array, atau fungsi. Namun, jenis petunjuk tidak dapat digunakan dengan jenis skalar seperti angka atau string. Contoh variabel dapat ditulis sebagai `namaVariabel`.

Penulisan fungsi, penamaan kelas, nama variabel adalah peka akan huruf besar (Kapital) dan huruf kecil. Kedua kutip ganda `" "` dari string memberikan kemampuan untuk interpolasi nilai variabel ke dalam string PHP. PHP menerjemahkan baris sebagai spasi, dan pernyataan harus diakhiri dengan titik koma `;`. beberapa contoh aplikasi yang menggunakan php

1. Facebook



Figure 6: Caption

2. Yahoo



Figure 7: Caption

3. Wikipedia



Figure 8: Caption

26 CI

merupakan sebuah framework PHP yang bisa mempercepat pengembang untuk membuat sebuah aplikasi web. Dilengkapi banyak library dan helper yang berguna di dalamnya dan tentunya mempermudah proses development. Sedangkan Twitter Bootstrap adalah sebuah alat bantu (framework) HTML dan CSS untuk membuat sebuah tampilan halaman website yang elegan dan support segala macam device. Dengan implementasi Twitter Bootstrap pada framework CodeIgniter akan mempermudah dan mempercepat pembuatan maupun pengembangan website dinamis tanpa harus kesulitan membuat desain.

27 Sejarah CI

CodeIgniter adalah sebuah framework berbasis PHP yang kuat dengan footprint yang sangat kecil, dibangun untuk pengembang yang membutuhkan toolkit

sederhana dan elegan untuk membuat aplikasi web dengan fitur lengkap. Framework atau dalam bahasa Indonesia dapat diartikan sebagai "kerangka kerja" merupakan sebuah tool yang bekerja pada suatu konsep tertentu dan terdiri dari berbagai fungsi yang dapat dengan mudah dijabarkan atau dialih-gunakan untuk membuat fungsi-fungsi lain yang lebih kompleks. Maksudnya, jika seorang Programmer menggunakan sebuah framework sebagai lingkup kerjanya, dia harus mematuhi segala ketentuan dari framework tersebut. Lebih lanjut ia dapat membangun berbagai fungsi yang rumit dan kompleks dengan menggunakan fungsi-fungsi yang telah disediakan oleh Framework tersebut. Jadi dia tidak perlu lagi menulis ulang semua coding, ia cukup mempelajari cara menggunakan kerangka (frame) yang telah tersedia dari framework tersebut.

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website-dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.4

CodeIgniter pertama kali ditulis oleh Rick Ellis (<http://www.ellislab.com>), seorang musisi rock yang beralih profesi menjadi programmer dalam riset kecil-kecilannya dan menghasilkan suatu framework PHP yang berukuran kecil, ringan serta memenuhi fitur umum aplikasi PHP. Namun, sejak tahun 2014 CodeIgniter telah dimiliki oleh British Columbia Institute of Technology (BCIT). Segala hal mengenai CodeIgniter dapat ditemui di websiter resminya beralamatkan <http://www.codeigniter.com>. Untuk saat ini telah dirilis versi 3.0.6.

1. Download CodeIgniter: CodeIgniter dapat diunduh secara gratis melalui situs resminya di <https://codeigniter.com/> Pada artikel ini versi yang digunakan adalah 3.1.11
2. Instalasi CodeIgniter: Karena CodeIgniter terdiri dari file dan folder PHP maka hal yang perlu dilakukan dalam instalasi CodeIgniter hanyalah mengekstrak file yang di unduh dari <http://ellislab.com/codeigniter/user-guide/> dan meletakkan pada direktori web server, Misalkan untuk Windows pada : C:/xamp/htdocs/ atau /var/www pada Linux. Kemudian pada file config.php yang berada pada direktori application/config/config.php lalu set base URL anda, misalkan : <http://localhost/ci/> . Jika aplikasi website yang anda buat akan menggunakan database buatlah koneksi database dengan mengedit file database.php yang berada pada aplikasi/config/database.php. Anda juga bisa menjalankan auto-load modul yang akan anda gunakan misalkan session, libraries, dll dengan mengedit file autoload.php kemudian untuk menentukan controller pertama yang dijalankan terletak pada routes.php.
3. memahami MVC CodeIgniter sendiri dibangun menggunakan konsep Model-View-Controller development pattern. Manfaat konsep MVC ini adalah memudahkan logic dalam coding aplikasi tersebut dan memudahkan pengembangan sistem yang sudah ada.

- (a) Model berhubungan dengan data dan interaksi ke database atau web-service. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL.
- (b) View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript.
- (c) Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model.
- (d) Help dan Tutorial Pendukung CodeIgniter Jika anda masih merasa bingung dan kesulitan memahami konsep CodeIgniter, cobalah search tutorial di google yang jumlahnya tidak terhitung. Akan tetapi CodeIgniter sendiri menyediakan dokumentasi yang sangat lengkap

28 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMP merupakan singkatan dari X (empat system operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis dan berikut tampilan UI pada xampp

CodeIgniter 2.0 CodeIgniter versi 2.0 baru saja diluncurkan pada awal tahun 2011. Pada versi terbaru ini, terdapat banyak perubahan mendasar dari CodeIgniter daripada versi sebelumnya. Perubahan ini menjadikan CodeIgniter jauh lebih kaya dan matang dibandingkan framework lainnya. Perubahan itu diantaranya adalah: • Menghilangkan dukungan terhadap PHP4. PHP4 sudah tidak didukung oleh tim pengembang PHP, karena memberikan dukungan untuk PHP4 membuat CodeIgniter semakin ketinggalan dari segi fitur. • Menghilangkan fitur plugin. Plugin mirip dengan helper, bertujuan untuk menghilangkan kerancuan ini maka fitur ini dihilangkan pada CodeIgniter 2.0. • Menghilangkan fitur scaffolding. Fitur ini nyaris tidak pernah digunakan dan implementasinya masih kurang bagus. • Penambahan library driver. Ini adalah library khusus



Figure 9: Caption



Figure 10: Caption

dimana kita dapat membuat driver dari library yang telah kita buat. • Support query string dan command-line execution. Hal ini menjawab kesulitan-kesulitan yang dialami pada versi sebelumnya. • Penambahan library cache. Untuk meningkatkan kualitas aplikasi maka library cache baik menggunakan apc, memcached maupun file base. • Penambahan fitur package. Untuk mempermudah distribusi resource dalam sebuah folder.

Apa itu MVC?

MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter . MVC adalah singkatan dari Model View Controller. MVC sebenarnya adalah sebuah pattern/teknik pemogramanan yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Adapun komponen-komponen MVC antara lain: 1. Model Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresen-

tasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL. 2. View View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller 3. Controller: Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

Gambar 2. Perbandingan PHP Biasa dan CodeIgniter

Perhatikan gambar 2. Pada eksekusi PHP, biasanya kita akan me-“load” semua library dan fungsi yang dibutuhkan kemudian digabungkan ke dalam HTML untuk di eksekusi oleh PHP. Untuk kasus sederhana cara tersebut masih baik-baik saja, tetapi ketika aplikasi tersebut menjadi kompleks/rumit maka kita akan sulit memeliharanya jika tidak didukung oleh arsitektur software yang bagus. Hal tersebut bisa terjadi disebabkan oleh code yang sama namun dibuat berulang-ulang, kode tidak konsisten dan lain-lain.

Gambar 3. Flow CodeIgniter

Jika dipetakan, alur kerja CodeIgniter akan tampak seperti gambar 3. Browser berinteraksi melalui controller. Controller-lah yang akan menerima dan membalas semua request dari browser. Untuk data maka controller akan meminta ke Model dan untuk UI/template akan meminta ke View. Jadi “Otak” dari aplikasi ada di controller, “Muka” aplikasi ada di view dan “Data” ada di model. Ketika

browser meminta sebuah halaman web maka router akan mencari controller mana yang harus menangani request tersebut. Setelah itu barulah si controller menggunakan model untuk mengakses data dan View untuk menampilkan data tersebut. Jangan Belajar CodeIgniter!!

Jangan coba-coba belajar CodeIgniter ketika Anda belum mengenal PHP. Berdasarkan pengalaman di forum dan milis, kebanyakan mereka yang belajar CodeIgniter tanpa memiliki dasar PHP yang baik akan mengalami banyak kesulitan, bukan dalam menguasai konsep CodeIgniter tetapi masih berkutat seputar PHP. Setidaknya Anda telah memahami konsep OOP pada PHP untuk

mulai belajar CodeIgniter. Sebaiknya ketika Anda mempelajari CodeIgniter Anda sudah memahami PHP dan Object Oriented Programming. Akan lebih bagus lagi jika Anda sudah familiar dengan design pattern.

Chapter 2 PHP Object Oriented Programming Syarat utama untuk menguasai CodeIgniter adalah memahami PHP dan konsep Object Oriented Programming. Pada bab ini kita akan membahas sedikit mengenai PHP dan OOP. Apasih PHP? PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman di sisi server. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server. Misalnya kita merequest sebuah file PHP <http://www.koder.web/index.php>, maka webserver akan melakukan parsing terhadap file PHP tersebut. PHP parser yang menjalankan kode-kode PHP yang terdapat pada file index.php lalu mengirimkan hasilnya ke web browser. Sejarah PHP Pada tahun 1994, Rasmus Lerdorf mengembangkan sebuah perangkat yang digunakan sebagai engine parsing sebagai penerjemah/interpreter beberapa macro. Pada saat itu engine digunakan untuk pembuatan buku tamu, counter dan beberapa homepage. Ia menamai engine parser tersebut dengan nama PHP/FI. Dengan semangat opensource, para pengembang di dunia mencoba mengembangkan PHP/FI. Sampai pada tahun 1997, lebih dari 500.000 website di dunia menggunakan PHP/FI untuk menyelesaikan masalah seperti koneksi ke database, menampilkan content yang dinamis dan lain-lain. Pada juni 1998, PHP 3.0 dirilis. Pada saat itu PHP sudah mendukung multplatform (bukan hanya linux), webserver, sejumlah database, SNMP (Simple Network Management Protocol) and IMAP (Internet Message Access Protocol). Menurut survei yang dilakukan oleh <http://netcraft.org>, saat ini pengguna PHP sudah mencapai 9,5 juta domain. Menggunakan PHP Ketika kita ingin mempelajari PHP, hal pertama yang harus disiapkan adalah sebuah web server, seperti yang sudah dibahas sebelumnya bahwa PHP merupakan bahasa pemrograman di sisi server. Banyak sekali jenis web server yang dapat digunakan, antara lain Apache, IIS, iPlanet, Omni, Xintami, dan lain-lain. Setelah web server terinstal barulah kita install PHP sebagai modul.

PHP dapat diperoleh secara gratis dengan cara men-download dari situs resmi PHP (<http://www.php.net/downloads.php>) atau website lain yang mempunyai salinan program PHP untuk di-download. Sebelum menginstall PHP, terlebih dahulu harus menginstall web server. Setelah PHP terinstall. Anda dapat meletakkan source code PHP ke dalam folder yang akan di publish oleh web server dengan ekstensi file .php. Jika tidak ingin direpotkan oleh proses konfigurasi dan implementasi webserver, Anda dapat menggunakan software yang sudah terpaket menjadi satu antara PHP, APACHE, dan MySQL. Contoh Software-nya adalah XAMPP, PHPTRIAD, FOX Server, dan lain-lain.

Apa sih Object Oriented Programming (OOP)? Object Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada obyek. Semua data dan fungsi pada paradigma ini dibungkus dalam kelas-kelas atau obyek-obyek. Bandingkan dengan logika pemrograman terstruktur, setiap obyek dapat menerima pesan, memproses data, dan mengirim pesan ke obyek lainnya. OOP diciptakan untuk mengatasi keterbatasan pada bahasa pemrograman tradisional. Konsep dari OOP sendiri adalah semua pemecahan masalah

dibagi ke dalam obyek. Dalam konsep OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai obyek. Apa itu Object Sederhananya, sebuah obyek adalah kumpulan dari variabel dan fungsi yang dibungkus menjadi satu entitas. Entitas tersebut dapat berupa variabel biasa. Sebuah obyek diciptakan melalui sebuah kelas atau dengan istilah *instance of class*. Obyek memiliki 2 elemen utama: 1. *Attributes* atau *Properties*: Yaitu nilai-nilai yang tersimpan dalam objek tersebut dan secara langsung maupun tidak langsung menentukan karakteristik dari obyek tersebut. 2. *Method*: Yaitu suatu aksi yang akan dijalankan atau dikerjakan oleh obyek tersebut.

Apa itu Class? Class dapat didefinisikan sebagai struktur data atau cetak biru dari suatu obyek. Lebih jelasnya adalah sebuah bentuk dasar atau blueprint yang mendefinisikan variabel, method umum pada semua obyek. Obyek sendiri adalah kumpulan variabel dan fungsi yang dihasilkan dari template khusus atau disebut class. Obyek adalah elemen pada saat run-time yang akan diciptakan, dimanipulasi, dan dibuang/di-destroy ketika eksekusi. Adapun class merupakan definisi statik dari himpunan obyek yang mungkin diciptakan sebagai instantiasi dari class. Perhatikan contoh class di berikut ini.

```

<?php /** Contoh kelas */ class Kendaraan /** end of class */ ?>

```

Contoh diatas memperlihatkan bagaimana mendefinisikan sebuah class dan meng-create sebuah instance dari class. Pada contoh di atas kita membuat sebuah kelas bernama “Kendaraan”. Dalam pembuatan kelas, pertama kita menggunakan kata kunci class yang diikuti oleh nama kelas, kemudian diakhiri dengan kurung kurawal. Di dalam kurung kurawal kita menuliskan kode-kode (berisi property dan method) supaya kelas tersebut bekerja seperti yang diinginkan. Kode-kode di dalam sebuah kelas terbagi menjadi dua kelompok, yaitu property dan method. Property adalah suatu wadah penyimpanan di dalam kelas yang bisa menampung informasi. Sederhananya property itu bisa disebut sebagai variabel di dalam kelas. Sedangkan method adalah fungsi yang ada di dalam kelas. Perhatikan contoh berikut.

```

<?php /** contoh kelas */ class kendaraan
    /** property class */ private $warna; private $jumlah_pintu; private $jumlah_roda; public $harga;
    public $merk;
    /** method class */ public function construct() { echo 'ini adalah object kendaraan. \n';
        public function set_harga($harga) { $this-> $harga = $harga;
        public function show_harga() { echo 'hargakendaraan : Rp.'. $this-> $harga.' \n';
        public function jalan() { echo 'brrrroooooom!!!';

```

```

    /** end of class */
    /** contoh object */
    #mengcreate object $saya_adalah_object dari class kendaraan
    $saya_adalah_object= new kendaraan;

```

```

$saya_adalah_object->set_harga(100000);
$saya_adalah_object->show_harga();

$saya_adalah_object->jalan();

```

Setiap property dan method memiliki identifier. Identifier-lah yang mengatur bagaimana property dan method digunakan. Identifier tersebut adalah public, private dan protected. Private berarti method atau property yang ada di dalam suatu kelas hanya bisa diakses di dalam kelasnya. Sedangkan pada method atau property yang bersifat public berarti method atau property tersebut bisa diakses di dalam dan di luar kelas. Inheritance Inheritance atau dalam bahasa Indonesianya disebut sebagai pewarisan adalah suatu cara untuk membuat sebuah kelas yang baru dengan menggunakan kelas lain yang sebelumnya sudah dibuat. Pada hubungan inheritance, sebuah class turunan mewarisi kelas leluhur (parent class). Oleh karena mewarisi, maka semua atribut dan method class dari induk akan dibawa (kecuali yang bersifat private), secara intrinsik menjadi bagian dari class anak. Adapun keuntungan yang didapat dari inheritance menambah fitur baru pada kelas anak dan mengubah atau mengganti fitur yang diwarisi dari kelas parent Adapun contoh kelas yang menggunakan konsep inheritance adalah

```

<?php
/* inheritance.php */ class Bapak {
private $nama ="Bapak"; function Bapak($n) {
$this->nama = $n;
}
function Hallo() {
echo "Halo, saya $this->nama <br>";
}
}

class Anak extends Bapak { function Hai(){
Echo \hai dari kelas anak";
}
}

$test = new Anak("Anak dari Bapak");
$test->Hallo();

```

Jika kita perhatikan di kelas anak sama sekali tidak memiliki fungsi hello, tetapi karena parentnya memiliki fungsi tersebut maka si anak dapat menggunakan fungsi tersebut. Selain menggunakan fungsi bapak, anak juga dapat menambah fungsi baru yaitu fungsi Hai.

Chapter 3 Instalasi dan Konfigurasi CodeIgniter Agar dapat menggunakan CodeIgniter, Anda harus menginstall dan melakukan konfigurasi terhadap CodeIgniter

terlebih dahulu. Instalasi CodeIgniter sangatlah mudah. Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Hal-hal yang harus dipersiapkan dalam menginstall CodeIgniter adalah Mempersiapkan Web Server

Seperti yang kita tahu bahwa PHP adalah bahasa pemrograman website yang berjalan disisi server oleh karena itu untuk dapat menjalankan website yang dibuat dengan menggunakan PHP, di komputer harus terinstall aplikasi web server yang mendukung PHP. Banyak sekali aplikasi web server yang beredar, salah satu web server yang sangat terkenal dan juga bersifat bebas adalah web server Apache, sebuah web server yang digunakan pada sebagian server yang ada di internet. Untuk website yang melibatkan basis data sebagai tempat penyimpanan datanya maka tidak perlu bingung karena CodeIgniter juga mendukung berbagai jenis server basis data, yaitu MySQL, MySQLi, MS SQL, PostgreSQL, Oracle, dan SQLite. CodeIgniter juga bisa dijalankan di semua sistem operasi yang bisa menjalankan aplikasi-aplikasi di atas, baik Windows, Linux, BSD dan yang lainnya. Instalasi CodeIgniter

Instalasi CodeIgniter sangat mudah. Meskipun namanya instalasi tetapi karena CodeIgniter adalah aplikasi berbasis website maka sebenarnya yang perlu dilakukan adalah meng-copy folder aplikasi CodeIgniter ke dalam folder htdocs atau DocumentRoot dari web server yang telah diinstall sebelumnya. Berbeda dengan instalasi software pada umumnya. Sebelum melakukan instalasi yang perlu dilakukan pertama kali adalah mendapatkan kode sumber dari CodeIgniter itu sendiri yang dapat didownload di <http://www.CodeIgniter.com/> (disediakan pula di dalam CD penyerta buku). Selanjutnya letakkan folder hasil ekstrak tadi di DocumentRoot web server, yaitu folder htdocs didalam direktori C:\apache\friendsbagi yang menggunakan XAMPP di Windows atau /var/www/html bagi yang menggunakan linux (semua tergantung dimana anda menginstall dan mengkonfigurasi webserver).

Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Adapun struktur utama dari CodeIgniter terbagi menjadi dua bagian, yaitu application dan sistem/core CodeIgniter. Application adalah tempat kita meletakkan code yang akan dibuat (bewarna merah dan hijau sedangkan sistem/core CodeIgniter yang bewarna ungu) . Folder sistem berisi library-library dan helper bawaan CodeIgniter.

Gambar 4. Struktur Direktori CodeIgniter Adapun susunan folder CodeIgniter secara default adalah: • Folder application: disinilah aplikasi yang akan kita bangun diletakkan. Folder config - tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari database, router dan autoload aplikasi. Folder controllers - tempat menyimpan semua file controller. Folder errors - tempat menyimpan semua template error aplikasi. Folder helpers - tempat menyimpan helper-helper yang bukan berasal dari CI. Folder hooks - tempat menyimpan hook yang digunakan untuk mengubah alur fungsi dari core CI. Folder language - tempat menyimpan bahasa-bahasa yang akan di gunakan. Folder libraries - tempat menyimpan semua library buatan kita sendiri. Folder models - tempat menyimpan semua model. Folder views - tempat menyimpan semua file view aplikasi.

- Folder system menyimpan semua file baik itu file aplikasi yang dibuat maupun core framework-nya. Folder cache - tempat menyimpan semua cache yang dibuat caching library. Folder codeigniter - tempat menyimpan semua semua file internals CI. Folder database - tempat menyimpan semua driver database drivers dan class yang akan digunakan. Folder fonts - tempat menyimpan semua font yang digunakan image manipulation library. Folder helpers - tempat menyimpan semua helper core CI. Folder language - tempat menyimpan semua language core CI. Folder libraries - tempat menyimpan semua library core CI. Folder logs - tempat menyimpan semua logs generated oleh CI. Folder plugin - tempat menyimpan semua plugin core CI. Folder scaffolding - tempat menyimpan semua file yang berfungsi sebagai scaffolding.
- Folder *user_guide* berisi *user_guide/manual* penggunaan CI. *File index.php* file yang akan *handle* semua request yang masuk ke aplikasi.

Setelah meletakkan CodeIgniter ke dalam folder htdocs maka akan didapatkan tampilan seperti di bawah ini, artinya CodeIgniter telah sukses berjalan di aplikasi Anda.

Gambar 5. Tampilan browser ketika sukses menginstall Codeigniter

Adapun checklist yang harus dilakukan untuk menjalankan CodeIgniter secara default adalah

- Pastikan Apache dan PHP telah terinstall dan berjalan di komputer.
- Pastikan peletakkan source code CodeIgniter di folder/direktori web apache (biasanya htdocs) dan memiliki permission setidaknya-tidaknya read only atau kode 644.

Konfigurasi CodeIgniter

Walaupun CodeIgniter dapat berjalan dengan konfigurasi default, tetapi untuk sebuah aplikasi yang nyata kita harus tetap melakukan konfigurasi, setidaknya pada bagian *base_url* dan *router*. Pengaturan *base_url* dan *router* sangat berguna ketika proses pengembangan file yang terdapat dalam direktori tersebut dan sering digunakan antar lain : *file config.php*. Pada file konfigurasi lain. Adapun konfigurasi – konfigurasi yang perlu diperhatikan adalah : *o config['base_url'] – Konfigurasi ini berisi alamat url sebuah aplikasi. Jika menggunakan helper url maka konfigurasi ini harus diisi dengan benar. Contoh : aplikasi Anda akan diakses dengan menggunakan domain www.contoh.com/app_cim*
"http://www.contoh.com/app_cim/";

Tetapi jika ingin menggunakan base url yang lebih fleksibel maka dapat menggantinya dengan variabel server. Contoh: *config['base_url'] = "http :
 //" . \$SERVER['HTTP_HOST'] . str_replace(basename(\$SERVER['SCRIPT_NAME']), "", \$SERVER['SCRIPT_NAME']) . \$config['index.php']* – Konfigurasi ini berisi file yang menjadi front controller. Konfigurasi ini berhubungan dengan konfigurasi ini bertujuan untuk menentukan bagaimana library URI bekerja. CodeIgniter dapat menangkap URL masing-masing cara mempunyai kelebihan dan kekurangan masing-masing, bahkan tidak semua web server mendukung konfigurasi ini bertujuan untuk menambahkan akhiran pada url. Contoh Anda mempunyai sebuah controller *page* yang berada di *localhost/index.php/page*, dengan menambahkan *url_suffix* berisi *'html'* maka url tadi dapat juga diakses melalui *localhost/index.php/page.html*.

o config['language'] – Secara default CodeIgniter sudah mendukung banyak bahasa / *multylanguage*. Kita dapat pasang yang ada dalam CodeIgniter dengan bahasa yang kita kehendaki. Untuk mengubah bahasa tersebut cukup pada konfigurasi ini bertujuan mengaktifkan / menonaktifkan hook pada CodeIgniter. Hook dapat dikatakan event – event yang terjadi pada CodeIgniter, dimana kita bisa meletakkan fungsi di dalamnya. Hook akan bermanfaat ketika ingin mengubah urutan loading library dimana Anda sudah meng – *extend library* router untuk menggunakan nya. Artinya Anda harus me load library database sebelum library loader. Hal tersebut bisa dilakukan melalui hook.

MY. Jikalau kita ingin mengubah/mengextend library CodeIgniter maka library tersebut harus memiliki prefik yang sama dengan konfigurasi pesan kesalahan yang termasuk kesalahan PHP. Jika di set 2 maka akan menampilkan semua pesan debug dan pesan logging yang berisi informasi seperti sebuah library telah di load juga akan ditampilkan. Jika di set 4 maka semuanya lain melalui konfigurasi ini. `config['sess_cookie_name'] = 'Namacookie'; config['sess_expiration'] = 7200; config['sess_encrypt_cookie'] = FALSE; config['sess_use_database'] = FALSE; config['sess_table_name'] = 'session_table'; config['sess_match_ip'] = FALSE; config['sess_match_useragent'] = TRUE; config['sess_time_to_update'] = 300;`

- Autoload.php. Konfigurasi ini bertujuan untuk menentukan sumber daya apa yang akan di load secara otomatis. Cara penggunaannya sederhana, misalnya kita ingin meload library database, pagination dan lain-lain secara otomatis maka kita tinggal mengubah `autoload['libraries']` menjadi `:autoload['libraries']=array('database','session','pagination')`
- Routes.php. Konfigurasi di file ini bertujuan untuk menentukan kemana routing oleh library route akan dilakukan. Hal paling sederhana yang harus dilakukan adalah mengubah default controller (controller yang akan dibuka ketika tidak ada uri yang diberikan oleh browser). Misalnya website kita beralamat `www.koder.web.id`. Maka ketika membuka website tersebut maka secara otomatis CodeIgniter akan mengalihkan ke controller default, karena tidak disertakan di dalam url kita. Adapun yang perlu diubah adalah `route['default_controller'] = "welcome"`;

Kesepakatan Coding (Coding Standart) CodeIgniter

Sebelum melakukan coding menggunakan codeigniter maka ada baiknya kita mengetahui apa saja kesepakatan-kesepakatan yang ada di codeigniter. Kesepakatan-kesepakatan tersebut akan membuat kode kita lebih mudah dipahami oleh developer lainnya. Adapun kesepakatan tersebut diantaranya :

- PHP Closing Tag

Ketika kita menulis library, helper, controller ataupun model maka sebaiknya tidak menggunakan tanda penutup pada dokumen php `?>`. Hal tersebut dilakukan untuk mencegah adanya spasi atau karakter yang tidak diinginkan pada code kita sehingga membuat aplikasi error. Kita juga disarankan untuk memberikan informasi tentang akhir dokumen dan berisi path dokumen tersebut.

Contoh salah: `if (php echo "Here's my code!"); ?>`

Contoh Benar: `if (php echo "Here's my code!");`

`/* End of file myfile.php */ /* Location: ./system/modules/mymodule/myfile.php */`

- Penamaan Class dan Method Penamaan Class harus dimulai dengan huruf besar. Jika class menggunakan beberapa kata maka kata-kata tersebut dipisahkan menggunakan underscore dan bukan camelcase.

Contoh salah: `class superclass class SuperClass` Contoh Benar: `class Superclass`

Aturan diatas juga berlaku untuk method contohnya

Contoh kurang tepat: `function fileproperties()` // Tidak deskriptif dan memiliki underscore `function fileProperties()` // Tidak deskriptif dan underscore CamelCase `function getFileproperties()` // Kurang underscore `function getFileProperties()` // menggunakan CamelCase `get_the_file_properties_from_the_file()` // terlalu panjang Contoh Tepat : `function getFileProperties()` // deskriptif, pakai underscore, dan huruf kecil

Chapter 3 Hello CodeIgniter Pada bab ini kita akan memulai coding menggunakan CodeIgniter. Sebelum memulai memulai maka perlu kita pahami kembali

apa itu Controller? Dan bagaimana hubungan controller dengan URI? Apakah Controller?

CodeIgniter adalah sebuah framework berbasis MVC. Sebuah Controller dapat dikatakan sebagai jantung dari suatu aplikasi, karena controller menentukan bagaimana permintaan HTTP yang harus ditangani. Sebuah kelas Controller adalah sebuah file yang terletak di dalam folder `application/controller` dan memiliki nama file yang sama dengan nama kelasnya dan dikaitkan dengan URL. Segmen-segmen pada URL pada codeigniter mencerminkan Controller yang dipanggil. Contoh: `http://www.domainsaya.com/class/fungsi/id` maka domain tersebut dapat dipecah menjadi bagian-bagian diantaranya:

Adapun komponen-komponen URL diatas adalah

- Konfigurasi Base Url, Bagian ini merupakan url yang kita masukkan pada konfigurasi base url yang merupakan url paling dasar untuk mengakses web atau aplikasi kita. *Segmen URI pertama*

Untuk contoh pertama, kita akan membuat sebuah aplikasi hello codeigniter yang sederhana. Aplikasi tersebut akan diletakkan di folder hello dari htdocs anda, sehingga anda dapat mengaksesnya dengan membuka url `http://localhost/hello`. Perhatikan URL Berikut ini: `http://localhost/hello/index.php/blog/` Pada contoh di atas, CodeIgniter akan berusaha untuk menemukan sebuah controller bernama `blog` pada file `blog.php`, kemudian menampilkannya ke browser. Mari kita coba membuat sebuah controller sederhana sehingga dapat melihat apa yang terjadi. Dengan menggunakan teks editor, buatlah file bernama `blog.php` di dalam folder `application/controller`. Setelah itu buatlah sebuah kelas `Blog` yang merupakan turunan dari kelas `CI_Controller`.

```
1. <?php if ( ! defined('BASEPATH')) 2. exit('No direct script access allowed'); 3. 4. class Blog extends CI_Controller { 5. 6. function __construct() { 7. parent::__construct(); 8. 9. 10. 11. function index() { 12. 13. } 14. }
```

Setelah itu apa yang terjadi? Coba buka situs Anda dengan menggunakan URL seperti ini:

`http://localhost/index.php/blog/`

Gambar 6. Tampilan Aplikasi Hello Codeigniter

Jika Anda melakukannya dengan benar, maka akan tampak tulisan Hello.. saya adalah contoh CodeIgniter pertama. Sekedar mengingatkan kembali, ketentuan penamaan class (baik controller maupun library) harus dimulai dengan huruf besar. Contoh:

```
<?php class Blog extends CI_Controller
```

Berikut ini contoh yang salah :

```
<?php class blog extends CI_Controller
```

Pada contoh di atas nama fungsi yang dipanggil adalah `index()`. Fungsi "Index" akan selalu dibaca secara default jika segmen kedua dari URI kosong. Cara lain untuk menampilkan "Hello CodeIgniter" adalah dengan mengakses url berikut: `localhost/index.php/blog/index/`

Segmen kedua dari URI yang menentukan fungsi mana yang akan dipanggil dari controller. Mari kita coba menambahkan fungsi baru ke controller:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed'); class Blog extends CI_Controller { function __construct() { parent::__construct();
```

```
function index() { echo "Haloo.. saya adalah contoh codeigniter pertama";
```

```

function komentar() echo "Ini adalah fungsi komentar"; /* End of file
Blog.php */ /* Location: ./application/controllers/blog.php */
Sekarang cobalah panggil URL berikut untuk melihat fungsi komentar:
localhost/index.php/blog/komentar/
Maka hasilnya akan menampilkan pesan "Ini Fungsi Komentar". Jika ingin
mengisi variabel param maka Anda tinggal menambahkan sebuah URI lagi setelah Blog. Biasanya parameter ketika
localhost/index.php/blog/komentar/tutorial-ci/123
Fungsi post pada controller dapat memiliki dua parameter yang akan dile-
watkan pada URI segmen 3 dan 4 ( "tutorial-ci" dan "123"). Jadi dapat
disimpulkan bahwa URI yang dipanggil dapat ditunjukkan seperti ini: local-
host/index.php/[Controller]/[Fungsi]/[param]/[param]/...

```

Controller dan View

Pada contoh program Hello CodeIgniter diatas adalah cara untuk menampilkan tulisan "hello CodeIgniter" secara langsung di controller. Namun sebenarnya hal tersebut bisa dilakukan di view. Sebagai contoh: tuliskan kode program di bawah ini pada file application/controller/blog.php :

```

i?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller function construct() parent :: construct();
function index() this->load->view("hello_codeigniter");
/* End of file Blog.php */ /* Location: ./application/controllers/ Blog.php
*/

```

Selanjutnya buatlah file *hello_codeIgniter.php* di folder *application* > *view* (*application/view/hello_codeIgniter.php*)

```

h1 > Hellosaya adalah view < /h1 >

```

Maka kode diatas akan memberikan hasil yang sama dengan contoh kasus pertama (tanpa menggunakan view), yang berbeda hanya tulisannya saja. Sebuah View sebenarnya hanyalah sebuah halaman web atau bagian dari halaman web, seperti sebuah header, footer, sidebar, dan lain-lain. Bahkan, View bisa menjadi fleksibel karena view dapat dimasukkan ke dalam view yang lain jika dibutuhkan. Untuk memanggil file view dapat digunakan fungsi seperti berikut ini (perhatikan baris 12):: *this->load->view('nama_view')*;

Nama_view adalah nama file view Anda. Dan file tersebut harus diletakkan di dalam folder *application/view*.

1. Nama file view - Nama file yang hendak di-load yang terletak di dalam folder *application/view*

2. Data Parameter - Parameter ini digunakan untuk melewati data dari controller ke dalam view.

Contoh: buatlah file controller dengan nama *blog.php* dengan kode program seperti berikut ini:

```

i?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller function construct() parent :: construct(); function index() data['judul']="J
blog"; data['isi'] = "Isi blog"; this->load->view("blog_view",data);
Blog.php */

```

Selanjutnya buatlah file view bernama *blog_view.php* dengan kode program seperti berikut ini :

```

i h1 i i?php echo $judul; ? > < /h1 > < p > < ?php echo $isi; ? i i /p i i p i i br / i Page
rendered in elapsed $imeseconds < /p >

```

Jika halaman *blog.php* dipanggil maka tampilannya kurang lebih seperti berikut:

CodeIgniter yang otomatis di load. Loader berfungsi sebagai pengatur dari sumberdaya-sumberdaya yang ada di dalam CodeIgniter seperti Model, View, Library, Helper, dan plugin. Cara penggunaannya adalah: *this->load->library('nama_library');*; *this->load->helper('nama_helper');*; *this->load->plugin('nama_plugin');*;

Nama library, helper dan plugin harus di isi dengan huruf kecil.

Ketika sebuah library sudah di-load maka library tersebut menjadi property pada object Controller. Adapun cara penggunaannya adalah sebagai berikut: *this->nama_library->fungsi();*

Library CodeIgniter Secara default CodeIgniter telah menyediakan library yang dapat digunakan secara langsung. Adapun library yang telah tersedia antara lain: • Benchmarking Class Library ini digunakan untuk melakukan pengukuran terhadap aplikasi yang dibuat. Seperti untuk mengetahui berapa lama waktu eksekusi dan berapa jumlah memori yang digunakan. Library ini sudah digunakan dan di-load secara otomatis oleh CodeIgniter. • Calendar Class Library ini berfungsi untuk menampilkan dan men-generate kalender. • Cart Class Library ini berfungsi untuk membuat shopping cart (keranjang belanja). Library ini memiliki ketergantungan terhadap kelas session karena item-item chart tersebut disimpan di dalam session. • Config Class Library ini berfungsi untuk mengambil data-data di dalam file konfigurasi. Library ini sudah di load secara otomatis oleh CodeIgniter. • Database Class Library database digunakan untuk memanipulasi serta mendapatkan data dari sebuah sistem database. Secara default database yang sudah didukung oleh CodeIgniter adalah mysql, mssql, oracle, postgres. Sedangkan database yang tidak didukung secara langsung oleh CodeIgniter dapat dijembatani dengan driver odbc. • Email Class Library email digunakan untuk mengirimkan email. Pengiriman email tersebut bisa dilakukan dengan menggunakan protokol mail, sendmail dan smtp.sqawd • Encryption Class Library Encryption digunakan untuk melakukan penyandian terhadap string tertentu • File Uploading Class Library Uploading digunakan untuk meng-upload file. Kelas ini sudah dilengkapi dengan pengecekan jenis file, dan ukuran file. • Form Validation Class Library form Validation digunakan untuk mengecek keabsahan form-form yang sudah di-submit oleh user.

• FTP Class Library FTP digunakan untuk meng-upload atau download file melalui ftp server. • HTML Table Class Library HTML table adalah sebuah kelas yang berfungsi untuk men-generate table dari data array. • Image Manipulation Class Library image manipulation berfungsi untuk mengolah gambar. Adapun fungsi-fungsi yang telah disediakan adalah Image Resizing, Thumbnail Creation, Image Cropping, Image Rotating dan Image Watermarking. • Input and Security Class Library Input dan security berfungsi untuk menjamin bahwa inputan dari form telah bersih dari karakter-karakter “aneh”. • Loader Class Library ini dapat disebut sebagai pengatur sumberdaya CodeIgniter. Semua sumberdaya yang ada akan dikendalikan oleh kelas ini. Library ini sudah di-load secara otomatis oleh CodeIgniter. • Language Class Library language digunakan untuk mengatur bahasa apa yang akan dipakai oleh CodeIgniter. • Output Class Library Output bertujuan untuk meng-handle output dari CodeIgniter, mulai dari cache sampai ke profiling bisa dilakukan kelas ini. • Pagination Class Untuk mem-paginate hasil database untuk performance dan usability, kita bisa

mengontrol berapa banyak record untuk ditampilkan disetiap halaman website, berapa banyak record untuk ditarik dari database dan tampilan dari bagian pagination • Session Class Library Session dapat digunakan untuk memelihara informasi status tentang user (seperti layaknya session di PHP). Tetapi Library ini tidak menggunakan session built-in dari PHP, Library Session men-generate session datanya sendiri yang disimpan di dalam Cookies. • Trackback Class Library Trackback digunakan untuk mengirim dan menerima data trackback. • Template Parser Class Library Template Parser digunakan untuk membuat template yang berisi parsable pseudo – templates. • Unit Testing Class Library Unit Testing digunakan untuk unit test function dalam aplikasi yang sedang dibuat. CodeIgniter menyediakan fungsi evaluasi dan dua fungsi hasil dalam library ini.

- URI Class Library URI digunakan untuk memarsing URL, lalu memecahnya ke dalam beberapa segmen dan kemudian di-passing ke controller atau disimpan sebagai variabel.
- User Agent Class Library User Agent digunakan untuk mengidentifikasi browser, mobile device, atau robot yang mengunjungi website. Kita juga bisa menggunakannya untuk mendeteksi dukungan bahasa, sekumpulan karakter, dan referrer.
- XML-RPC Class Library XML-RPC digunakan untuk men-setup klien XML-RPC dan server.
- Zip Encoding Class Library Zip Encoding digunakan untuk membuat file ZIP baik yang berjenis teks maupun data binary.

Helper CodeIgniter Helpers seperti namanya akan membantu Anda membangun aplikasi dengan tugas tertentu. Tidak seperti library, helper bukanlah Object Oriented tapi berupa prosedural. Setiap helper berisi satu atau lebih fungsi, masing-masing berfokus pada tugas tertentu yang tidak ada ketergantungan dengan fungsi lainnya. Helper dapat juga di-load secara otomatis dalam /system/application/config/autoload.php. Adapun helper yang ada di dalam CodeIgniter antara lain:

- Array — Helper array berisi fungsi yang membantu pekerjaan berhubungan dengan array. Sebagai contoh fungsi `random_element()` mengambil array sebagai input dan mengembalikan satu elemen acak dari array tersebut.
- Download — Helper download berisi fungsi untuk mengunduh data dengan mudah. Fungsi `force_download()` menghasilkan header server yang memaksa data untuk di-download.
- File — Helper file berisi fungsi yang membantu untuk membaca, menulis, dan menghapus file.
- Form — Helper form berisi fungsi yang membantu membangun form.
- HTML — Helper HTML berisi fungsi yang membantu membuat blok HTML.
- Language — Helper language berisi fungsi yang membantu untuk mengubah kata-kata menjadi bentuk plural atau singular, memberlakukan camel case atau mengubah kata-kata yang dipisahkan oleh spasi menjadi frase yang digarisbawahi, sebagai contoh fungsi `singular` bisa mengubah kata-kata yang membantu pengelolaan emoticons.
- String — Helper string berisi fungsi-fungsi yang membantu pekerjaan berhubungan dengan string, seperti fungsi `random_string()` yang akan membuat string acak.
- Text — Helper text berisi fungsi-fungsi yang membantu Anda bekerja dengan teks. Sebagai contoh: fungsi `word_limiter()` dapat membatasi string ke sejumlah kata tertentu yang berguna untuk memformat output.
- Typography — Helper typography berisi fungsi-fungsi yang membantu memformat output dengan menggunakan tag HTML.
- URL — Helper URL berisi fungsi-fungsi yang membantu pekerjaan dengan URL. Anda akan menggunakan fungsi `base_url()` dan `anchor()` dalam setiap halaman.

Membuat Library Sendiri Untuk mendapatkan kode yang bagus sebaiknya fungsionalitas yang sama itu di bungkus menjadi sebuah library. Sebuah library adalah sebuah kelas yang diletakkan pada folder application/libraries. Sebuah library yang dibuat dengan menggunakan `get_instance()` adalah sebuah fungsi yang mengembalikan instance dari library tersebut. Untuk membuat library sendiri, pertama-tama buatlah folder library yang akan digunakan. Setelah itu, buatlah file library yang akan digunakan. Setelah itu, buatlah fungsi-fungsi yang akan digunakan. Setelah itu, buatlah dokumentasi untuk library tersebut. Setelah itu, buatlah tes untuk library tersebut. Setelah itu, buatlah library tersebut siap untuk digunakan.

CodeIgniter sangat mudah untuk mengadopsi library yang bersifat external. Library yang paling mudah untuk diadopsi adalah sebuah single class tanpa parameter pada constructor. Untuk kelas yang memiliki konstruktor lebih dari satu maka sebaiknya kelas tersebut diubah terlebih dahulu atau jika kelas library tersebut terdiri atas beberapa class maka lebih mudahnya kita membuat sebuah library factory dari library tersebut. Contoh 1. Library CSV Reader

Sebagai contoh kita akan menggunakan sebuah library yang berfungsi untuk membaca file CSV kedalam memori.. Adapun isi dari librari tersebut adalah

```
if(!defined('BASEPATH'))exit('No direct script access allowed'); /**
 * CSVReader Class * @author Pierre-Jean Turpeau * @link http://www.CodeIgniter.com/wiki/CSVReader
 */
class csv_reader
{
    var $fields; var $separator=';'; var $enclosure='\"'; var $max_row_size = 4096;
    /** * Parse a file containing CSV formatted data. * * @access public *
    @param string * @param boolean * @return array */
    function parse_file($filepath, $namedFields = true) {
        $content = false; $file = fopen($filepath, 'r');
        if($namedFields) {
            $this->fields = fgetcsv($file, $this->max_row_size, $this->separator, $this->enclosure);
            while(($row = fgetcsv($file, $this->max_row_size, $this->separator, $this->enclosure)) != false) {
                if($row[0] != null) // skip empty lines
                    if(!$content) $content = array();
                if($namedFields) {
                    $items = array();
                    foreach($this->fields as $id => $field) {
                        if(isset($row[$id])) $items[$field] = $row[$id];
                    }
                }
                return $content;
            }
        }
    }
}
```

Agar dapat menggunakan library tersebut maka lakukan langkah-langkah berikut ini:

- Copy Library Ke Direktori application/libraries Untuk menggunakan library buatan kita sendiri atau library eksternal maka langkah yang pertama yang harus dilakukan adalah meletakkan library itu di direktori application/libraries.

Nama file dan nama kelas harus sama. Contoh nama file adalah `csv_reader.php` maka nama kelasnya adalah `csv_reader`.

Perhatikan controller berikut ini

```
1. if(!defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Welcome extends CI_Controller {
    function __construct() {
        parent::__construct();
        $this->load->library('csv_reader');
        $data = $this->csv_reader->parse_file(FPATH.'data_nilai.csv');
        echo $pre > ";
        print_r($data);
    }
}
/* End of file welcome.php */
/* Location: ./application/controllers/welcome.php */
```

Pada baris ke-13 kita akan mencoba menggunakan library `csv_reader`, kelas tersebut akan memarsing sebuah file CSV. Lalu hasil parsingnya akan ditampilkan.

Chapter 6 Kasus 1. Penanganan dan Validasi Form Pada bab ini akan dibahas mengenai penanganan form dan cara melakukan validasi terhadap sebuah form inputan. Penanganan form artinya bagaimana cara kita untuk mendapatkan dan mengolah data yang dikirimkan oleh pengguna melalui form. Untuk membuat form sendiri, sebaiknya kita menggunakan helper form. Setelah kita berhasil mendapatkan data dari user, kita harus memvalidasi data-data yang telah dikirimkan tersebut agar aplikasi yang kita bangun menjadi aman. Inputan yang tidak divalidasi membuat aplikasi kita rentan terhadap serangan seperti Sql Injection, Xss dan lain-lain. Penanganan Form

Untuk menangani form kita membutuhkan library input. Library tersebut sudah dipanggil secara otomatis oleh codeigniter. Fungsi yang tersedia pada library ini selain untuk menangani form juga memiliki fungsi security/keamanan.

Contohnya untuk menghandle atau memfilter xss kita dapat menggunakan fungsi `xssfilter.nama = this->input->xssfilter(this->input->post('nama'));`

Jika kita ingin melindungi seluruh aplikasi dari dengan xss filter maka kita bisa menconfignya dengan mengubah konfigurasi di `system/application/config/config.php` dari `config['global_xssfiltering'] = FALSE;`

Menjadi

`config['global_xssfiltering'] = TRUE;`

Selain kedua cara diatas, kita dapat juga menggunakan parameter kedua dari fungsi post atau get maka kita akan mendapatkan hasil yang sama dengan kedua cara diatas, contohnya `this->input->post('nama', TRUE);` `this->input->get('nama', TRUE);` Fungsi post digunakan untuk menangkap inputan POST sedangkan get digunakan untuk menangkap inputan GET. Selain itu fungsi input juga dapat digunakan untuk cookie, variabel SERVER, dan user agent `this->input->cookie('nama_cookie', TRUE);` `this->input->ip_address();` `this->input->server('PATH_INFO', TRUE);`

Selain library input, Codeigniter juga menyediakan sebuah helper untuk mempermudah penanganan form. Helper tersebut adalah helper form. Helper tersebut membantu kita dalam membuat form pada view. Adapun cara meload helper form adalah `this->load->helper('form');`

Ada banyak fungsi yang disediakan oleh helper ini diantaranya `form_open()`, `form_close()`, `form_open_multipart()` lain. Untuk lebih detailnya anda dapat melihat pada dokumentasi codeigniter. Pada kasus ini, fungsi yang banyak digunakan adalah `form_open()` pada html. Contoh :

`attributes = array('class' => 'email', 'id' => 'myform'); echo form_open('form/send', attributes);`

Maka akan menghasilkan

`<form method="post" accept-charset="utf-8" action="http://localhost/index.php/email/send" class="email" id="myform">`

`Form input berfungsi untuk menggenerat tag input pada html. Fungsi ini dapat dipakai dengan dua cara. Cara pertama adalah dengan menggunakan fungsi form_open() dan form_close().`

akan menghasilkan

`<input type="text" name="username" id="username" value="ibnoe">`

Cara kedua adalah dengan mempassing array kedalam fungsi tersebut. Cara ini adalah cara yang advance dan sangat berguna jika kita ingin mengcustomisasi inputan. Contohnya: `data = array('name' => 'username', 'id' => 'username', 'value' => 'ibnoe', 'maxlength' => '100', 'size' => '50', 'style' => 'width: 50');`

`echo form_input(data);`

Akan menghasilkan output :

`<input type="text" name="username" id="username" value="ibnoe" maxlength="100" size="50" style="width:50">`

Untuk contoh menangani form maka ikutilah langkah-langkah berikut ini.

1. Membuat Controller Hitung

Perlu diingat hal pertama yang harus di konfigurasi pada setiap aplikasi adalah `BASE_URL`. Setelah melakukan konfigurasi `base_url` pada `file application/config/config.php` maka kita akan dapat menjalankan aplikasi. `<?php if (!defined('BASEPATH')) exit('No direct script access allowed');` `load->helper(array('url', 'form'));` 9. 10. `function index()` 12. 13. `this->load->view('menu_hitung');` 14. 15. 16. `function perkalian()` 17. 18. `data['v1']=(int)$this->`

```

input->post('v1',true);19.data['v2']=(int)$this->input->post('v2',true);20.data['hasil']=data['v1']*data['v2'];
21. $this->load->view('perkalian',data);22.23.24. function pembagian()25.26. data['v1']=(int)$this->input->post('v1',true);27. data['v2']=(int)$this->input->post('v2',true);28. if (data['v2'] > 0)29.data['hasil']=data['v1']/data['v2'];
30. else31. data['hasil']='Error,v2tidakboleh0!';32.$this->load->view('pembagian',data);33.34.

```

Pada baris ke 8, dilakukan loading terhadap helper url dan form sekaligus. Loading tersebut dilakukan dalam fungsi konstruktor karena kedua helper tersebut digunakan pada semua fungsi dan view yang ada. Pada kelas hitung tersebut, terdapat 3 fungsi yaitu index, perkalian dan pembagian. Pada fungsi index hanya berisi sebuah pemanggilan terhadap view. View

tersebut akan berisi menu-menu yang akan mengarahkan pengguna untuk melakukan perkalian atau pembagian. Pada fungsi perkalian dan pembagian akan dilakukan proses perkalian atau pembagian. Pada fungsi tersebut, variabel yang berasal dari view akan ditangkap menggunakan fungsi post (berarti dikirimkan melalui POST method). 2. Membuat View

Setelah membuat controller maka kita akan membuat view yang merupakan interface dari aplikasi tersebut. Code berikut ini adalah view menu *hitung*. *View* ini berisi navigasi ke fungsi perkalian dan pembagian.

```

!DOCTYPE html > 2. <html lang="en"> 3. <head> 4. <meta charset="utf-8"> 5. <title> Welcome to CodeIgniter </title> 6.7. <style type="text/css"> 8.9. body { background-color: #fff; 10. margin: 40px; 11. font-family: Lucida Grande, Verdana, sans-serif; } 12. </style> 13. </head> 14. <body> 15. <h1> CodeIgniter 2.0 dan Form! </h1> 16. </body> 17. </html>

```

```

39. <p>Silakan pilih menu dibawah ini.</p> 40. <ul> 41. <li><a href="hitung/perkalian">Perkalian</a></li> 42. <li><a href="hitung/pembagian">Pembagian</a></li> 43. </ul> 44. <p>Page rendered in elapsed time: seconds </p> 45.46. </body> > </html>

```

Perhatikan baris ke-41 dan ke-42, itu adalah contoh penggunaan helper. Fungsi anchor bertujuan untuk membuat sebuah link ke page tertentu. Berikut ini adalah tampilan dari view diatas jika dipanggil.

Gambar 7. Screenshoot view menu *hitung* Setelah menu *hitung* selanjutnya membuat view untuk perkalian dan pembagian.

```

!DOCTYPE html > 2. <html lang="en"> 3. <head> 4. <meta charset="utf-8"> 5. <title> Welcome to CodeIgniter </title> 6.7. <style type="text/css"> 8.9. body { background-color: #fff; 10. margin: 40px; 11. font-family: Lucida Grande, Verdana, sans-serif; } 12. </style> 13. </head> 14. <body> 15. <h1> Perkalian! </h1> 16. <p>Silakan masukan data berikut!</p> 17. <div> 18. <input type="text" value="" /> 19. <input type="text" value="" /> 20. <input type="button" value="Hitung!" /> 21. </div> 22. <div> 23. <input type="text" value="" /> 24. <input type="text" value="" /> 25. <input type="button" value="Pembagian!" /> 26. </div> 27. <p>Page rendered in elapsed time: seconds </p> 28. </body> > </html>

```

Pada view ini banyak digunakan helper form (baris 40 – 45). Tag input yang ada pada view ini di-generate oleh fungsi `form_input` dan `form_submit` oleh fungsi `form_submit`. Adapun tampilan dari view ini adalah sebagai berikut.

```

1. <!DOCTYPE html> 2. <html lang="en"> 3. <head> 4. <meta charset="utf-8"> 5. <title> Welcome to CodeIgniter </title>

```

```

6. 7. <style type="text/css"> 8. 9. body 10. background-color: #fff; 11. margin: 40px; 12. font-family: Lucida Grande, Verdana, Sans-serif; 13. font-size: 14px; 14. color: #4F5155; 15. 16. 17. a 18. color: #003399; 19. background-color: transparent; 20. font-weight: normal; 21. 22. 23. h1 24. color: #444; 25. background-color: transparent; 26. border-bottom: 1px solid #D0D0D0; 27. font-size: 16px; 28. font-weight: bold; 29. margin: 24px 0 2px 0; 30. padding: 5px 0 6px 0; 31. 32. 33. </style> 34. </head> 35. <body> 36. 37. <h1>pembagian!</h1> 38. 39. <p>Silakan masukan data berikut!!</p> 40. <?php echo form_open('hitung/pembagian'); ?> 41. <?php echo form_input('v1',v1); ?> 42. <?php echo form_input('v2',v2); ?> 43. <?php echo form_submit('submit','Hitung!!'); ?> 44. <?php echo form_close(); ?> 45. <br> 46. Hasil :<?php echo hasil; ?> 47. 48. </body> 49. </html>

```

Berikut ini adalah tampilan dari form pembagian

Gambar 9. Screenshoot view pembagian

Catatan :

Menggunakan Validasi Form

Sebenarnya sampai pada tahap kedua di atas kita telah mampu membuat sebuah aplikasi perkalian dan pembagian sederhana, tetapi sebuah aplikasi yang baik harus memiliki sebuah kontrol terhadap inputan user. Validation form sangat penting pada aplikasi. Sebuah aplikasi akan memiliki banyak inputan dari pengguna dan semua inputan untuk pengguna harus aman. CodeIgniter memiliki sebuah library Form Validation Library yang akan membantu kita untuk membuat sebuah validasi yang cepat, mudah dan aman. Pada contoh validasi form ini hampir sama dengan 2 langkah diatas, tetapi memiliki sedikit penambahan terutama dalam fungsi perkalian dan pembagian. Perhatikan controller hitung berikut ini (controller berikut ini berasal dari contoh sebelumnya dengan penambahan). Application/controllers/hitung.php

```

1. <?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Hitung extends CI_Controller {
4.     function __construct() {
5.         parent::__construct();
6.     }
7.     function index() {
8.         $this->load->helper(array('url','form'));
9.         10. 11. 12. 13. $this->load->view('menu_hitung');
14. 15. 16. 17. 18. 19. $this->load->library('form_validation');
20. $this->form_validation->set_rules('v1','Variabel1',21.'required|integer');
22. $this->form_validation->set_rules('v2','Variabel2',23.'required|integer');
24. if($this->form_validation->run()) {
25. 26. $int=$this->input->post('v1',true);
27. $int2=$this->input->post('v2',true);
28. $hasil=$int*$int2;
29. 30. else {
31. 32. 33. 34. $hasil=0;
35. 36. $this->load->view('perkalian',$data);
37. 38. 39. function pembagian() {
40. 41. $this->load->library('form_validation');
42. $this->form_validation->set_rules('v1','Variabel1',43.'required|is_natural_no_zero');
44. $this->form_validation->set_rules('v2','Variabel2',45.'required|is_natural_no_zero');
46. if($this->form_validation->run()) {
47. 48. $int=$this->input->post('v1',true);
49. $int2=$this->input->post('v2',true);
50. $hasil=$int/$int2;
51. 52. else {
53. 54. 55. $hasil=0;
56. 57. 58. 59. $this->load->view('pembagian',$data);
60. 61. 62.

```

Perhatikan baris 19-24 pada fungsi perkalian, disana kita me-load sebuah library bernama `form_validation`. Library tersebut memiliki fungsi untuk melakukan validasi terhadap inputan user.

```

$this->form_validation->set_rules('v1','Variabel1','required|integer');

```

Baris di atas merupakan salah satu contoh untuk menentukan aturan inputan pada form. Parameter pertama (berisi v1) merupakan name dari inputan yang akan divalidasi (harus sama dengan atribut name pada tag input contoh: `<input name="v1">`). Parameter kedua adalah nama dari inputan tersebut. Nama tersebut dapat berbeda atau sama dengan name pada parameter pertama. Parameter ketiga merupakan aturan dari inputan tersebut. Aturan-aturan tersebut harus dipatuhi agar sebuah form dapat diproses. Aturan-aturan tersebut dipisahkan oleh tanda `—`. Pada contoh di atas berarti inputan v1 harus diisi (required) dan berisi data integer(integer). Adapun aturan-aturan yang tersedia di dalam CodeIgniter adalah:

Aturan Keterangan required Isi inputan tidak boleh kosong. matches Isi inputan harus sama dengan inputan tertentu contoh

`matches[password] minlength` Panjang inputan harus memiliki jumlah minimal karakter, contoh penggunaan `z.alpha_numeric` Inputan harus berisi karakter alfabet dan numeric. `alpha_dash` Inputan harus berisi semua huruf, `z.alpha_dash` Inputan harus berisi karakter alfabet, dash, dan underscore. `numeric` Inputan hanya boleh berisi angka/huruf numeric. `integer` Inputan hanya boleh berisi 0, 1, 2, 3, dan seterusnya. `is_natural` Inputan hanya boleh berisi bilangan natural saja kecuali nol : 1, 2, 3, dan seterusnya. `valid_email` Inputan harus berisi format email yang benar. `valid_emails` Inputan harus berisi

Pada baris-46, kita melakukan pengecekan terhadap rule-rule yang telah diset (`this->form_validation->run()`). Fungsi tersebut akan menghasilkan nilai TRUE apabila semua

rule terpenuhi dan menghasilkan nilai FALSE jika sebaliknya. Selain fungsi-fungsi diatas Anda juga dapat membuat custom validation jika aturan-aturan di atas tidak ada yang memenuhi dengan kebutuhan Anda. 1. `<!DOCTYPE html>` 2. `<html lang="en">` 3. `<head>` 4. `<meta charset="utf-8">` 5. `<title>` Perkalian menggunakan validasi!! `</title>` 6. 7. `<style type="text/css">` 8. 9. `body` 10. `background-color: fff;` 11. `margin: 40px;` 12. `font-family: Lucida Grande, Verdana, Sans-serif;` 13. `font-size: 14px;` 14. `color: 4F5155;` 15. 16. 17. `a` 18. `color: 003399;` 19. `background-color: transparent;` 20. `font-weight: normal;` 21. 22. 23. `h1` 24. `color: 444;` 25. `background-color: transparent;` 26. `border-bottom: 1px solid D0D0D0;` 27. `font-size: 16px;` 28. `font-weight: bold;` 29. `margin: 24px 0 2px 0;` 30. `padding: 5px 0 6px 0;` 31. 32. 33. `</style>` 34. `</head>` 35. `<body>` 36. 37. `<h1>` Perkalian!! `</h1>` 38. `<?php echo validation_errors();? >` 39. `< p >` Silakan masuk dan data berikut!! `</p >` 40. `<?php echo form_open('hitung/perkalian');? >` 41. `<?php echo form_input('v1',v1);? >` 42. `<?php echo form_input('v2',v2);? >` 43. 44. `<?php echo form_submit('submit','Hitung!!');? >` 45. `<?php echo form_close();? >` 46. `Hasil :<?php echo hasil;? >` 47. 48. `</body>` 49. `</html>`

Pada view, kita cukup menambahkan satu fungsi yang berguna untuk menampilkan kesalahan yang terjadi. Perhatikan baris 38. Fungsi `validation_error` adalah fungsi untuk menampilkan kesalahan.

Gambar 10. Form perkalian dengan tampilan error

Begitu juga untuk view pembagian, kita tinggal menambah fungsi `validation_error` untuk menampilkan error.

1. `<!DOCTYPE html>` 2. `<!DOCTYPE html>` 3. `<html lang="en">` 4. `<head>` 5. `<meta charset="utf-8">` 6. `<title>` Pembagian menggunakan validasi!! `</title>` 7. 8. `<style type="text/css">` 9. 10. `body` 11. `background-color: fff;` 12. `margin: 40px;` 13. `font-family: Lucida Grande, Verdana, Sans-serif;` 14. `font-size: 14px;` 15. `color: 4F5155;` 16. 17. 18. `a` 19. `color: 003399;` 20. `background-`

```

color: transparent; 21. font-weight: normal; 22. 23. 24. h1 25. color: 444;
26. background-color: transparent; 27. border-bottom: 1px solid D0D0D0; 28.
font-size: 16px; 29. font-weight: bold;
30. margin: 24px 0 2px 0; 31. padding: 5px 0 6px 0; 32. 33. 34. 35.
i/stylei 36. i/headi 37. i/bodyi 38. 39. ih1ipembagian!i/h1i 40. i?php echo
validation_errors();? > 41. < p > Silakanmasukandataberikut!! < /p > 42. <
?phpechoform_open('hitung/pembagian');? > 43. <?phpechoform_input('v1',v1);?i
/ 44. i?php echo form_input('v2',v2);?iibr 45. 46. i?php echo form_submit('submit','Hitung!!');? >
47. <?phpechoform_close();? >< br > 48.Hasil :<?phpechohasil;?i 49. 50.
ipibr/iPage rendered in elapsedimeseconds < /p > 51.52. < /body ><
/html >

```

View diatas akan memberikan tampilan seperti gambar 11 berikut ini.

Gambar 11. Form perkalian dengan tampilan error

Chapter 7 CodeIgniter Database

CodeIgniter mendukung banyak jenis database misalnya MySQL, PostgreSQL, Oracle, dan lain-lain. Dukungan database dari CodeIgniter berupa penyediaan beberapa driver database yang sekaligus juga memiliki fungsi sekuriti, caching dan active record.

Connect ke Database

CodeIgniter memiliki sebuah file konfigurasi yang memungkinkan Anda menyimpan konfigurasi untuk melakukan koneksi ke database (username, password, nama database, dan lain-lain). File konfigurasi terletak berada di application/config/database.php. Pengaturan konfigurasi disimpan dalam bentuk array multi-dimensi dengan prototipe seperti berikut ini: `db['default']['hostname'] = "localhost";db['default']['username']="root";db['default']['password']="";db['default']['database']="databas`
`db['default']['dbprefix'] = "";db['default']['pconnect']=TRUE;db['default']['db_debug'] = FALSE;db['default']['cache_on'] = FALSE;db['default']['cachedir']="";db['default']['charset'] = "utf8";db['default']['dbcollat']="utf8_general_ci";`

Kita dapat memiliki beberapa konfigurasi database. Alasan kita menggunakan array multi-dimensi adalah agar memungkinkan Anda secara opsional menyimpan beberapa set nilai-nilai koneksi. Jika, misalnya, Anda menjalankan aplikasi di beberapa kondisi lingkungan/environment (development, production, testing, dll) di bawah satu instalasi, maka Anda dapat mengatur kelompok konfigurasi untuk masing-masing environment. Sebagai contoh, Anda sedang menjalankan aplikasi di environment test maka Anda tinggal menambahkan konfigurasi berikut ini dan mengubah default konfigurasinya (mengeset `active_group = "test"`)
`menjaditest :db['test']['hostname']="localhost";db['test']['username'] = "root";db['test']['password']="";db['test']['database'] = "database_name";db['test']['dbdriver']="mysql";db['test']['dbprefix'] = "";db['test']['pconnect']=TRUE;db['test']['db_debug'] = FALSE;db['test']['cache_on'] = FALSE;`
`db['test']['cachedir'] = "";db['test']['charset'] = "utf8";db['test']['dbcollat']="utf8_general_ci";active_group = "test";`

Untuk connect ke database ada beberapa cara yang disediakan oleh CodeIgniter diantaranya :

1. Menambahkan Database Library Sebagai Autoload Library Untuk connect ke database Anda bisa menambahkan database sebagai autoload library di

file application/config/autoload.php. Cara ini sangat sederhana, cukup menambahkan kata "database" ke dalam autoload library sehingga menjadi : `autoload['libraries'] = array("database");`

Ketika menambahkan autoload pastikan bahwa database sudah terkonfigurasi dengan benar untuk menghindari kesalahan. 2. Mengaktifkan Manual Dari Library Database Jika hanya ada beberapa halaman website yang memerlukan konektivitas database, maka untuk optimalisasi lakukan koneksi ke database secara manual, cukup dengan menambahkan baris kode di bawah ini pada tiap fungsi tempat yang membutuhkan koneksi ke database atau dalam konstruktor kelas Anda untuk membuat database yang tersedia secara global di kelas. `this->load->database();`

Jika fungsi di atas tidak berisi informasi apapun di parameter pertama akan menyambung ke group konfigurasi yang aktif. Untuk memilih kelompok tertentu dari file konfigurasi, Anda dapat melakukan seperti pada contoh berikut. Hal tersebut berguna pada saat menggunakan aplikasi yang memiliki dua database. `this->load->database('group_name');`

Group_name adalah nama grup konfigurasi dari file konfigurasi Anda. Untuk menghubungkan secara manual `config['username'] = "myusername"; config['password'] = "mypassword"; config['database'] = "mydatabase"; config['dbdriver'] = "mysql"; config['dbprefix'] = ""; config['pconnect'] = FALSE; config['debug'] = TRUE; config['cache_on'] = FALSE; config['cachedir'] = ""; config['charset'] = "utf8"; config['dbcollat'] = "utf8_general_ci";`
`this->load->database(config);`

Atau Anda bisa mengirimkan nilai-nilai database Anda sebagai Data Source Name. DSN harus memiliki prototipe seperti ini: `dsn = 'dbdriver://username:password@hostname/database';` `this->load->database(dsn);`

3. Mengaktifkan manual dari model Selain kedua cara di atas Anda juga bisa mengaktifkan database pada saat loading model. Caranya adalah dengan mengeset TRUE pada parameter ketiga load model. Contoh : `this->load->model('Model_name', "", TRUE);`

Selain itu Anda juga bisa menggunakan konfigurasi tertentu untuk sebuah model. Contoh:

`manual_config['hostname'] = "localhost"; manual_config['username'] = "myusername"; manual_config['password'] = "mypassword"; manual_config['database'] = "mydatabase"; manual_config['dbdriver'] = "mysql"; manual_config['dbprefix'] = ""; manual_config['pconnect'] = FALSE; manual_config['debug'] = TRUE; this->load->model('Model_name', "", manual_config);`

CodeIgniter Model

Model pada CodeIgniter adalah sebuah kelas php yang berfungsi untuk menangani data. Ingat data bukan hanya dari database tetapi juga bisa dari File Text, Web Service atau layanan-layanan data lainnya. Contoh Model: class Blogmodel extends Model

`var $title = ""; var $content = ""; var $date = "";`
`function Blogmodel() { parent::Model(); }`
`function get_data() { $query = $this->db->get('entries', 10); return $query->result(); }`

Sebuah model sebenarnya tidak harus meng-extend class Model. Kelas model di-extend ketika hendak menggunakan fitur database pada CodeIgniter

saja. Semua Model harus diletakkan di dalam folder application/models. Agar dapat menggunakan model maka kita harus me-load model tersebut. Adapun perintah yang dapat digunakan untuk meload sebuah model adalah *this->load->model('Model_name');*this->Model_name->get_data();

Ketika sudah berhasil me-load sebuah model maka model tersebut akan menjadi sebuah property. Melalui property itulah Anda akan menggunakan semua fungsi yang ada di dalam model (perhatikan baris ke 2 pada contoh diatas *get_data* adalah fungsi di dalam kelas *Model_name*)

Melakukan Query pada Database

Untuk mendapatkan data pada database Anda harus melakukan query. Setelah query dilakukan barulah bisa mendapatkan data hasil query tersebut, baik dalam bentuk object maupun array. *query = this->db->query('QUERY SQL ANDA');*

Perlu ditekankan bahwa fungsi query di atas belum menghasilkan data apapun. Keluarannya hanya berupa Object(true) atau False. Ketika keluarannya adalah False maka query yang dilakukan berarti gagal. Tetapi jika true atau mengembalikan sebuah object maka query yang

dilakukan berarti berhasil. Dari object tersebut (variabel *query* kala menggunakan contoh diatas) Anda dapat men
db->query('SELECT name, title, email FROM my_table');

foreach(query->result() as row) echo row->title; echo row->name; echo row->email;

echo 'Total Results: ' . query->num_rows();

Contoh di atas adalah salah satu contoh penggunaan query (tapi ingat, pastikan Anda sudah memiliki konfigurasi database yang benar dan telah me-load library database. Perhatikan *query->result()*. Itu adalah contoh syntax untuk mendapatkan hasil query dalam bentuk array untuk mendapatkan hasil query dalam bentuk array asosiatif.

Query Return Value Seperti yang telah disebutkan di atas, setelah query dilakukan kita perlu memanggil fungsi tertentu untuk mendapatkan hasil dari query. Secara umum ada dua jenis tipe data yang dapat dihasilkan yaitu array dan object. Dari segi jumlah kita hanya dapat mengambil satu record (per record) atau keseluruhan record. • Mengembalikan Hasil Query Sebagai Kumpulan Array. Sebelum mengambil hasil query, ada baiknya mengecek terlebih dahulu apakah query tersebut memiliki hasil atau tidak. Perhatikan fungsi *result->result_array()*, itu adalah method yang digunakan untuk mengambil hasil query. Sedangkan fungsi *result->num_rows()* digunakan untuk mengetahui berapa jumlah record yang didapat. *result = this->*

*db->query('SELECT * FROM users');* *if (result->num_rows() > 0) foreach (result->result_array() as row) echo row['username']; echo row['email'];* • Mengembalikan Hasil Query Sebagai Kumpulan Object. Fungsi yang digunakan hampir sama dengan mengembalikan nilai query sebagai array. Cukup dengan memanggil method *result->result()*.

*result = this->db->query('SELECT * FROM users');* *if (result->num_rows() > 0) foreach (result->result() as row) echo row->username; echo row->email; // Mengembalikan Hasil Query Sebagai Row Object*
*db->query('SELECT * FROM users');* *if (query->num_rows() > 0) row = query->row(); echo row->username; echo row->email; // Mengembalikan Hasil Query Sebagai Row Object jadi kita hanya*
*db->query('SELECT * FROM users');* *if (query->num_rows() > 0) row = query->row(); echo row->username; echo row->email;*

Framework Codeigniter
Sebuah Panduan dan Best Practice
Framework CodeIgniter Sebuah Panduan dan Best Practice
Kata Pengantar

Codeigniter merupakan salah satu framework terbaik yang ada saat ini. Banyak website-website besar telah menggunakan codeigniter sebagai framework utama dalam membangun website maupun aplikasi mereka. Selain memudahkan dalam menggunakan, codeigniter juga cepat, mudah di kustomisasi dan integrasikan dengan library atau framework lainnya. Framework ini juga sempat menjadi perhatian pembuat php – Rasmus Lerdorf

“I like CodeIgniter because it is faster, lighter and the least like a framework.” - Rasmus Lerdorf

Dalam buku ini akan dibahas mengenai cara-cara penggunaan Codeigniter, mulai dari pengenalan tentang controller, model dan view, penggunaan codeigniter dan form, penggunaan database di codeigniter, studi-studi kasus hingga penggunaannya dengan library-library lain seperti jQuery dan jQuery UI. Akhir kata penulis menyadari bahwa penulisan dalam buku ini masih jauh dari sempurna. Oleh karena itu pertanyaan, kritik dan saran dapat di emailkan ke xibnoe@gmail.com atau dapat mengunjungi blog penulis di <http://www.koder.web.id>

Pekanbary, Juni 2011 Ibnu Daqil Id, M.Ti

DAFTAR ISI

Perkenalan CodeIgniter 2.0 1 Kenapa Menggunakan Framework? 2 Apa itu Codeigniter? 3 Apa sih Kelebihan CodeIgniter? 3 CodeIgniter 2.0 4 Apa itu MVC? 5 Jangan Belajar CodeIgniter!! 7 PHP Object Oriented Programming 9 Apasih PHP? 9 Sejarah PHP 9 Menggunakan PHP 9 Apa sih Object Oriented Programming (OOP)? 10 Apa itu Object 10 Apa itu Class? 10 Inheritance 12 Instalasi dan Konfigurasi CodeIgniter 13 Mempersiapkan Web Server 13 Instalasi CodeIgniter 13 Konfigurasi CodeIgniter 16 Kesepakatan Coding (Coding Standard) CodeIgniter 18 Hello CodeIgniter 21 Apaan sih Controller? 21 Controller dan View 24 Mempercantik URL Codeigniter 26 CodeIgniter Helper dan Library 29 Menggunakan Library dan Helper di CodeIgniter 29 Library CodeIgniter 30 Helper CodeIgniter 33 Membuat Library Sendiri 34 Menggunakan Library External 34 Kasus 1. Penanganan dan Validasi Form 37 Penanganan Form 37 Menggunakan Validasi Form 45 CodeIgniter Database 51 Connect ke Database 51 CodeIgniter Model 54 Melakukan Query pada Database 54 Query Return Value 55 Menggunakan Active Record 57 Kasus 2. CRUD dan Pagination Database 61 Kasus 3. Sistem Templating 72 Native CodeIgniter Tempating 72 Kasus 4. Sistem Authentikasi 78 Kasus 5. Image Gallery Sederhana 87 Kasus 6. Buku Tamu menggunakan Codeigniter 92 Membuat Table Guestbook 92 Konfigurasi Guestbook 92 Membuat Model Guestbook 93 Membuat Controller Dan View 94 Kasus 7. Membuat Shopping Cart Sederhana 100 Kasus 8. CodeIgniter dan Ajax 107 Kasus 9. Codeigniter dan jQuery AutoComplete 111 Kasus 10. Codeigniter dan Openflash Chart 116

Chapter 1 Perkenalan CodeIgniter 2.0

Akhir-akhir ini CodeIgniter menjadi sebuah framework yang hangat dibicarakan di Indonesia. Hampir semua milis dan forum PHP banyak membahas masalah

CodeIgniter. Terlebih lagi banyak perusahaan-perusahaan ternama di Indonesia (Kompas.com, okezone.com, urbanesia.com, bejubel.com, dan lain-lain) yang telah menggunakan CodeIgniter dalam produk mereka. Apa sih CodeIgniter? CodeIgniter adalah sebuah framework PHP yang dapat mempercepat pengembangan untuk membuat sebuah aplikasi web. Ada banyak library dan helper yang berguna didalamnya dan tentunya mempermudah proses development. Ibarat ingin membangun rumah maka Anda tidak perlu membuat semen, memotong kayu menjadi papan, mengubah batu menjadi porselen dan lain-lain. Anda cukup memilih komponen-komponen tersebut lalu dikombinasikan menjadi rumah yang indah.

Gambar1. Website-website indonesia yang menggunakan Codeigniter Jadi keuntungan yang didapat dalam penggunaan framework adalah : • Menghemat Waktu Pengembangan – Dengan struktur dan library yang telah disediakan oleh framework maka tidak perlu lagi memikirkan hal-hal tersebut, jadi Anda hanya fokus ke proses bisnis yang akan dikerjakan. • Reuse of code – Dengan menggunakan framework maka pekerjaan kita akan memiliki struktur yang baku, sehingga kita dapat menggunakannya kembali di proyek-proyek lainnya. • Bantuan komunitas - Ada komunitas-komunitas yang siap membantu jika ada permasalahan, selain itu juga bisa berbagi ilmu sehingga dapat meningkatkan kemampuan pemrograman kita. • Kumpulan best practice – sebuah framework merupakan kumpulan best practice yang sudah teruji. Jadi kita dapat meningkatkan kualitas kode kita.

Sebelum mendalami CodeIgniter lebih jauh, sebaiknya dipahami terlebih dahulu apa itu framework. Framework adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan, bahkan isu-isu kompleks yang ada. Sebuah framework telah berisi sekumpulan arsitektur/konsep-konsep yang dapat mempermudah dalam pemecahan sebuah permasalahan. Perlu diingat, framework bukanlah peralatan/tools untuk memecahkan sebuah masalah, tetapi sebagai ALAT BANTU. Framework hanya menjadi sebuah konstruksi dasar yang menopang sebuah konsep atau sistem yang bersifat “essential support” (penting tapi bukan komponen utama).

Kenapa Menggunakan Framework?

Salah satu alasan mengapa orang menggunakan framework terutama dalam membangun sebuah aplikasi adalah kemudahan yang ditawarkan. Didalam sebuah framework biasanya sudah tersedia struktur aplikasi yang baik, standard coding (1), best practice (2) dan design pattern (3), dan common function (4). Dengan menggunakan framework kita dapat langsung fokus kepada business process yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standar coding dan lain-lain. Dengan memanfaatkan design pattern dan common function yang telah ada di dalam framework maka hal tersebut dapat mempercepat proses pengembangan aplikasi. Kita tidak perlu membuat sesuatu fungsionalitas yang bersifat umum. Tanpa disadari ketika kita membangun sebuah aplikasi yang banyak melibatkan banyak fungsionalitas yang telah dibangun itu ternyata sama atau berulang-ulang. Dengan pengelempokkan itulah kita dapat mempercepat pengembangan aplikasi. Selain kemudahan dan kecepatan dalam membangun sistem, dengan menggunakan framework tertentu kita juga

dapat “menyeragamkan” cara kita mengimplementasikan kode program. Dengan framework kita akan “dipaksa” untuk patuh kepada sebuah kesepakatan. Selain itu juga akan memudahkan pengembang lain untuk mempelajari dan mengubah aplikasi yang telah dibuat apabila kode yang dihasilkan konsisten dan patuh pada sebuah aturan tertentu.

1 Standar Coding adalah sebuah standar yang harus diikuti oleh programmer untuk menulis code. 2 Best Practice adalah Kumpulan-kumpulan action yang telah teruji oleh para expert 3 Design Pattern adalah Teknik-teknik yang menjadi best practise 4 Common Function adalah fungsi-fungsi atau library yang telah umum digunakan dalam pengembangan sebuah sistem

Apa itu CodeIgniter?

CodeIgniter adalah sebuah web application framework yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan CodeIgniter adalah untuk membantu developer untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. CodeIgniter menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan. CodeIgniter diperkenalkan kepada public pada tanggal 28 februari 2006. CodeIgniter sendiri dibangun menggunakan konsep Model-View-Controller development pattern. CodeIgniter sendiri merupakan salah satu framework tercepat dibandingkan dengan framework lainnya. Pada acara frOSCon (August 2008), pembuat php Rasmus Lerdorf mengatakan dia menyukai codeigniter karena dia lebih ringan dan cepat dibandingkan framework lainnya ("because it is faster, lighter and the least like a framework.") Apa sih Kelebihan CodeIgniter?

CodeIgniter sangat ringan, terstruktur, mudah dipelajari, dokumentasi lengkap dan dukungan yang luar biasa dari forum CodeIgniter. Selain itu CodeIgniter juga memiliki fitur-fitur lainnya yang sangat bermanfaat, antara lain: • Menggunakan Pattern MVC. Dengan menggunakan pattern MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas. • URL Friendly. URL yang dihasilkan sangat url friendly. Pada CodeIgniter di-minimalisasi penggunaan *GET* dan diganti dengan *URI*. • Kemudahan dalam mempelajari, mengintegrasikan Library dan helper. Jika kita membandingkan antara CodeIgniter dengan framework-framework lainnya maka beberapa poin yang membuat CodeIgniter unggul adalah : • Kecepatan. Berdasarkan hasil benchmark CodeIgniter merupakan salah satu framework PHP tercepat yang pernah ada. • Macamserta mudah mengadopsi library lainnya. • Dokumentasi lengkap dan jelas. Bahkan tanpa buku ini pun Code

anggota team. Jika sebuah framework sangat sulit dipelajari maka akan beresiko untuk memperlambat team development anda.

CodeIgniter 2.0 CodeIgniter versi 2.0 baru saja diluncurkan pada awal tahun 2011. Pada versi terbaru ini, terdapat banyak perubahan mendasar dari CodeIgniter daripada versi sebelumnya. Perubahan ini menjadikan CodeIgniter jauh lebih kaya dan matang dibandingkan framework lainnya. Perubahan itu diantaranya adalah: • Menghilangkan dukungan terhadap PHP4. PHP4 sudah tidak didukung oleh tim pengembang PHP, karena memberikan dukungan untuk PHP4 membuat CodeIgniter semakin ketinggalan dari segi fitur. • Menghilangkan fitur plugin. Plugin mirip dengan helper, bertujuan untuk menghilangkan kerancuan ini maka fitur ini dihilangkan pada CodeIgniter 2.0. • Menghilangkan fitur scaffolding. Fitur ini nyaris tidak pernah digunakan dan implementasinya

masih kurang bagus. • Penambahan library driver. Ini adalah library khusus dimana kita dapat membuat driver dari library yang telah kita buat. • Support query string dan command-line execution. Hal ini menjawab kesulitan-kesulitan yang dialami pada versi sebelumnya. • Penambahan library cache. Untuk meningkatkan kualitas aplikasi maka library cache baik menggunakan apc, memcached maupun file base. • Penambahan fitur package. Untuk mempermudah distribusi resource dalam sebuah folder.

Apa itu MVC?

MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter. MVC adalah singkatan dari Model View Controller. MVC sebenarnya adalah sebuah pattern/teknik pemograman yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Adapun komponen-komponen MVC antara lain: 1. Model Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL. 2. View View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller 3. Controller: Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

Gambar 2. Perbandingan PHP Biasa dan CodeIgniter

Perhatikan gambar 2. Pada eksekusi PHP, biasanya kita akan me-“load” semua library dan fungsi yang dibutuhkan kemudian digabungkan ke dalam HTML untuk di eksekusi oleh PHP. Untuk kasus sederhana cara tersebut masih baik-baik saja, tetapi ketika aplikasi tersebut menjadi kompleks/rumit maka kita akan sulit memeliharanya jika tidak didukung oleh arsitektur software yang bagus. Hal tersebut bisa terjadi disebabkan oleh code yang sama namun dibuat berulang-ulang, kode tidak konsisten dan lain-lain.

Gambar 3. Flow CodeIgniter

Jika dipetakan, alur kerja CodeIgniter akan tampak seperti gambar 3. Browser berinteraksi melalui controller. Controller-lah yang akan menerima dan membalas semua request dari browser. Untuk data maka controller akan meminta ke Model dan untuk UI/template akan meminta ke View. Jadi “Otak” dari aplikasi ada di controller, “Muka” aplikasi ada di view dan “Data” ada di model. Ketika

browser meminta sebuah halaman web maka router akan mencari controller mana yang harus menangani request tersebut. Setelah itu barulah si controller menggunakan model untuk mengakses data dan View untuk menampilkan data tersebut. Jangan Belajar CodeIgniter!!

Jangan coba-coba belajar CodeIgniter ketika Anda belum mengenal PHP. Berdasarkan pengalaman di forum dan milis, kebanyakan mereka yang belajar CodeIgniter tanpa memiliki dasar PHP yang baik akan mengalami banyak kesulitan, bukan dalam menguasai konsep CodeIgniter tetapi masih berkutat seputar PHP. Setidaknya Anda telah memahami konsep OOP pada PHP untuk mulai belajar CodeIgniter. Sebaiknya ketika Anda mempelajari CodeIgniter Anda sudah memahami PHP dan Object Oriented Programming. Akan lebih bagus lagi jika Anda sudah familiar dengan design pattern.

Chapter 2 PHP Object Oriented Programming Syarat utama untuk menguasai CodeIgniter adalah memahami PHP dan konsep Object Oriented Programming. Pada bab ini kita akan membahas sedikit mengenai PHP dan OOP. Apasih PHP? PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman di sisi server. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server. Misalnya kita merequest sebuah file PHP <http://www.koder.web/index.php>, maka webserver akan melakukan parsing terhadap file PHP tersebut. PHP parser yang menjalankan kode-kode PHP yang terdapat pada file index.php lalu mengirimkan hasilnya ke web browser. Sejarah PHP Pada tahun 1994, Rasmus Lerdorf mengembangkan sebuah perkakas yang digunakan sebagai engine parsing sebagai penerjemah/interpreter beberapa macro. Pada saat itu engine digunakan untuk pembuatan buku tamu, counter dan beberapa homepage. Ia menamai engine parser tersebut dengan nama PHP/FI. Dengan semangat opensource, para pengembang di dunia mencoba mengembangkan PHP/FI. Sampai pada tahun 1997, lebih dari 500.000 website di dunia menggunakan PHP/FI untuk menyelesaikan masalah seperti koneksi ke database, menampilkan content yang dinamis dan lain-lain. Pada juni 1998, PHP 3.0 dirilis. Pada saat itu PHP sudah mendukung multiflatform (bukan hanya linux), webserver, sejumlah database, SNMP (Simple Network Management Protocol) and IMAP (Internet Message Access Protocol). Menurut survei yang dilakukan oleh <http://netcraft.org>, saat ini pengguna PHP sudah mencapai 9,5 juta domain. Menggunakan PHP Ketika kita ingin mempelajari PHP, hal pertama yang harus disiapkan adalah sebuah web server, seperti yang sudah dibahas sebelumnya bahwa PHP merupakan bahasa pemrograman di sisi server. Banyak sekali jenis web server yang dapat digunakan, antara lain Apache, IIS, iPlanet, Omni, Xintami, dan lain-lain. Setelah web server terinstal barulah kita install PHP sebagai modul.

PHP dapat diperoleh secara gratis dengan cara men-download dari situs

resmi PHP (<http://www.php.net/downloads.php>) atau website lain yang mempunyai salinan program PHP untuk di-download. Sebelum menginstall PHP, terlebih dahulu harus meng-install web server. Setelah PHP terinstall. Anda dapat meletakkan source code PHP ke dalam folder yang akan di publish oleh web server dengan ekstensi file .php. Jika tidak ingin direpotkan oleh proses konfigurasi dan implementasi webserver, Anda dapat menggunakan software yang sudah terpaket menjadi satu antara PHP, APACHE, dan MySQL. Contoh Software-nya adalah XAMPP, PHPTRIAD, FOX Server, dan lain-lain.

Apa sih Object Oriented Programming (OOP)? Object Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada obyek. Semua data dan fungsi pada paradigma ini dibungkus dalam kelas-kelas atau obyek-obyek. Bandingkan dengan logika pemrograman terstruktur, setiap obyek dapat menerima pesan, memproses data, dan mengirim pesan ke obyek lainnya. OOP diciptakan untuk mengatasi keterbatasan pada bahasa pemrograman tradisional. Konsep dari OOP sendiri adalah semua pemecahan masalah dibagi ke dalam obyek. Dalam konsep OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai obyek. Apa itu Object Sederhananya, sebuah obyek adalah kumpulan dari variabel dan fungsi yang dibungkus menjadi satu entitas. Entitas tersebut dapat berupa variabel biasa. Sebuah obyek diciptakan melalui sebuah kelas atau dengan istilah instance of class. Obyek memiliki 2 elemen utama: 1. Attributes atau Properties: Yaitu nilai-nilai yang tersimpan dalam objek tersebut dan secara langsung maupun tidak langsung menentukan karakteristik dari obyek tersebut. 2. Method: Yaitu suatu aksi yang akan dijalankan atau dikerjakan oleh obyek tersebut.

Apa itu Class? Class dapat didefinisikan sebagai struktur data atau cetak biru dari suatu obyek. Lebih jelasnya adalah sebuah bentuk dasar atau blueprint yang mendefinisikan variabel, method umum pada semua obyek. Obyek sendiri adalah kumpulan variabel dan fungsi yang dihasilkan dari template khusus atau disebut class. Obyek adalah elemen pada saat run-time yang akan diciptakan, dimanipulasi, dan dibuang/di-destroy ketika eksekusi. Adapun class merupakan definisi statik dari himpunan obyek yang mungkin diciptakan sebagai instantiasi dari class. Perhatikan contoh class di berikut ini.

```
ï¿½?php /** Contoh kelas */ class Kendaraan /** end of class */ ï¿½
```

Contoh diatas memperlihatkan bagaimana mendefinisikan sebuah class dan meng-create sebuah instance dari class. Pada contoh di atas kita membuat sebuah kelas bernama "Kendaraan". Dalam pembuatan kelas, pertama kita menggunakan kata kunci class yang diikuti oleh nama kelas, kemudian diakhiri dengan kurung kurawal. Di dalam kurung kurawal kita menuliskan kode-kode (berisi property dan method) supaya kelas tersebut bekerja seperti yang diinginkan. Kode-kode di dalam sebuah kelas terbagi menjadi dua kelompok, yaitu property dan method. Property adalah suatu wadah penyimpanan di dalam kelas yang bisa menampung informasi. Sederhananya property itu bisa disebut sebagai variabel di dalam kelas. Sedangkan method adalah fungsi yang ada di dalam kelas. Perhatikan contoh berikut. ï¿½?php /** contoh kelas */ class kendaraan

```

    /*** property class ***/ private warna; private jumlah_pintu; private jumlah_roda; public harga;
public merk;
    /*** method class ***/ public function construct() echo 'ini adalah object
kendaraan. jbr /i';
    public function set_harga(harga) this-> harga = harga;
    public function show_harga() echo 'hargakendaraan : rp.'.this-> harga.'. jbr
/i';
    public function jalan() echo 'brrroooooom!!!';
    /*** end of class ***/ /*** contoh object ***/ mengcreate object saya_a_dalah_object_dari_class_kendaraan_saya_d
new_kendaraan; saya_a_dalah_object-> set_harga(100000); saya_a_dalah_object->
show_harga();
    saya_a_dalah_object-> jalan();

```

Setiap property dan method memiliki identifier. Identifier-lah yang mengatur bagaimana property dan method digunakan. Identifier tersebut adalah public, private dan protected. Private berarti method atau property yang ada di dalam suatu kelas hanya bisa diakses di dalam kelasnya. Sedangkan pada method atau property yang bersifat public berarti method atau property tersebut bisa diakses di dalam dan di luar kelas. Inheritance Inheritance atau dalam bahasa Indonesianya disebut sebagai pewarisan adalah suatu cara untuk membuat sebuah kelas yang baru dengan menggunakan kelas lain yang sebelumnya sudah dibuat. Pada hubungan inheritance, sebuah class turunan mewarisi kelas leluhur (parent class). Oleh karena mewarisi, maka semua atribut dan method class dari induk akan dibawa (kecuali yang bersifat private), secara intrinsik menjadi bagian dari class anak. Adapun keuntungan yang didapat dari inheritance menambah fitur baru pada kelas anak dan mengubah atau mengganti fitur yang diwarisi dari kelas parent Adapun contoh kelas yang menggunakan konsep inheritance adalah

```

i?php /* inheritance.php */ class Bapak private nama = "Bapak"; function Bapak(n)
this-> nama = n; function Hallo() echo "Halo, saya this-> nama < br > ";
class Anak extends Bapak function Hai() Echo "hai dari kelas anak";
test = new Anak(" AnakdariBapak"); test-> Hallo();

```

Jika kita perhatikan di kelas anak sama sekali tidak memiliki fungsi hello, tetapi karena parentnya memiliki fungsi tersebut maka si anak dapat menggunakan fungsi tersebut. Selain menggunakan fungsi bapak, anak juga dapat menambah fungsi baru yaitu fungsi Hai.

Chapter 3 Instalasi dan Konfigurasi CodeIgniter Agar dapat menggunakan CodeIgniter, Anda harus menginstall dan melakukan konfigurasi terhadap CodeIgniter terlebih dahulu. Instalasi CodeIgniter sangatlah mudah. Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Hal-hal yang harus dipersiapkan dalam menginstall CodeIgniter adalah Mempersiapkan Web Server

Seperti yang kita tahu bahwa PHP adalah bahasa pemrograman website yang berjalan disisi server oleh karena itu untuk dapat menjalankan website yang dibuat dengan menggunakan PHP, di komputer harus terinstall aplikasi web server yang mendukung PHP. Banyak sekali aplikasi web server yang beredar, salah satu web server yang sangat terkenal dan juga bersifat bebas adalah web

server Apache, sebuah web server yang digunakan pada sebagian server yang ada di internet. Untuk website yang melibatkan basis data sebagai tempat penyimpanan datanya maka tidak perlu bingung karena CodeIgniter juga mendukung berbagai jenis server basis data, yaitu MySQL, MySQLi, MS SQL, PostgreSQL, Oracle, dan SQLite. CodeIgniter juga bisa dijalankan di semua sistem operasi yang bisa menjalankan aplikasi-aplikasi di atas, baik Windows, Linux, BSD dan yang lainnya. Instalasi CodeIgniter

Instalasi CodeIgniter sangat mudah. Meskipun namanya instalasi tetapi karena CodeIgniter adalah aplikasi berbasis website maka sebenarnya yang perlu dilakukan adalah meng-copy folder aplikasi CodeIgniter ke dalam folder htdocs atau DocumentRoot dari web server yang telah diinstall sebelumnya. Berbeda dengan instalasi software pada umumnya. Sebelum melakukan instalasi yang perlu dilakukan pertama kali adalah mendapatkan kode sumber dari CodeIgniter itu sendiri yang dapat didownload di <http://www.CodeIgniter.com/> (disediakan pula di dalam CD penyerta buku). Selanjutnya letakkan folder hasil ekstrak tadi di DocumentRoot web server, yaitu folder htdocs didalam direktori C:\apache\friendsbagi yang menggunakan XAMPP di Windows atau /var/www/html bagi yang menggunakan linux (semua tergantung dimana anda menginstall dan mengkonfigurasi webserver).

Dengan menggunakan konfigurasi default saja maka CodeIgniter sudah dapat berjalan di web server Anda. Adapun struktur utama dari CodeIgniter terbagi menjadi dua bagian, yaitu application dan sistem/core CodeIgniter. Application adalah tempat kita meletakkan code yang akan dibuat (bwarna merah dan hijau sedangkan sistem/core CodeIgniter yang bewarna ungu) . Folder sistem berisi library-library dan helper bawaan CodeIgniter.

Gambar 4. Struktur Direktori CodeIgniter Adapun susunan folder CodeIgniter secara default adalah:

- Folder application: disinilah aplikasi yang akan kita bangun diletakkan. Folder config - tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari database, router dan autoload aplikasi. Folder controllers - tempat menyimpan semua file controller. Folder errors - tempat menyimpan semua template error aplikasi. Folder helpers - tempat menyimpan helper-helper yang bukan berasal dari CI. Folder hooks - tempat menyimpan hook yang digunakan untuk mengubah alur fungsi dari core CI. Folder language - tempat menyimpan bahasa-bahasa yang akan di gunakan. Folder libraries - tempat menyimpan semua library buatan kita sendiri. Folder models - tempat menyimpan semua model. Folder views - tempat menyimpan semua file view aplikasi.

- Folder system menyimpan semua file baik itu file aplikasi yang dibuat maupun core framework-nya. Folder cache - tempat menyimpan semua cache yang dibuat caching library. Folder codeigniter - tempat menyimpan semua semua file internals CI. Folder database - tempat menyimpan semua driver database drivers dan class yang akan digunakan. Folder fonts - tempat menyimpan semua font yang digunakan image manipulation library. Folder helpers - tempat menyimpan semua helper core CI. Folder language - tempat menyimpan semua language core CI. Folder libraries - tempat menyimpan semua library core CI Folder logs - tempat menyimpan semua logs generated oleh

CI. Folder plugin - tempat menyimpan semua plugin core CI. Folder scaffolding - tempat menyimpan semua file yang berfungsi sebagai scaffolding . • Folder *user_guide* berisi *user_guide/manual* penggunaan CI. *Fileindex.php* file yang akan *handle* semua request yang masuk ke aplikasi.

Setelah meletakkan CodeIgniter ke dalam folder *htdocs* maka akan didapatkan tampilan seperti di bawah ini, artinya CodeIgniter telah sukses berjalan di aplikasi Anda.

Gambar 5. Tampilan browser ketika sukses menginstall Codeigniter

Adapun checklist yang harus dilakukan untuk menjalankan CodeIgniter secara default adalah

- Pastikan Apache dan PHP telah terinstall dan berjalan di komputer.
- Pastikan peletakkan source code CodeIgniter di folder/direktori web apache (biasanya *htdocs*) dan memiliki permission setidaknya-tidaknya *read only* atau kode 644.

Konfigurasi CodeIgniter

Walaupun CodeIgniter dapat berjalan dengan konfigurasi default, tetapi untuk sebuah aplikasi yang nyata kita harus tetap melakukan konfigurasi, setidaknya pada bagian *base_url* dan *router*. Pengaturan *base_url* dan *router* sangat berguna ketika proses pengembangan file yang terdapat dalam direktori tersebut dan sering digunakan antar lain : *Config.php*. Pada file konfigurasi lain. Adapun konfigurasi – konfigurasi yang perlu diperhatikan adalah : *o config['base_url'] – Konfigurasi ini berisi alamat url sebuah aplikasi. Jika menggunakan helper url maka konfigurasi ini harus di – set dengan benar. Contoh : aplikasi Anda akan diakses dengan menggunakan domain *www.contoh.com/app_cim* maka http : // *www.contoh.com/app_cim* ;*

Tetapi jika ingin menggunakan base url yang lebih fleksibel maka dapat menggantinya dengan variabel server. Contoh: *config['base_url'] = "http : //".SERVER['HTTP_HOST'].str_replace(basename(SERVER['SCRIPT_NAME']),"",SERVER['SCRIPT_NAME']).config['index.php'] – Konfigurasi ini berisi file yang menjadi front controller. Konfigurasi ini berhubungan dengan konfigurasi lain untuk menentukan bagaimana library URI bekerja. CodeIgniter dapat menangkap URL masing-masing cara mempunyai kelebihan dan kekurangan masing – masing, bahkan tidak semua web server mendukung konfigurasi ini bertujuan untuk menambahkan akhiran pada url. Contoh Anda mempunyai sebuah controller *page* // *localhost/index.php/page*, dengan menambahkan *url_suffix* *berisi \html* maka url tadi dapat juga diakses melalui *localhost/index.php/page.html*.*

o config['language'] – Secara default CodeIgniter sudah mendukung banyak bahasa / multy language. Kita dapat pesannya yang ada didalam CodeIgniter dengan bahasa yang kita kehendaki. Untuk mengubah bahasa tersebut cukup di konfigurasi ini bertujuan mengaktifkan / menonaktifkan hook pada CodeIgniter. Hook dapat dikatakan event – event yang terjadi pada CodeIgniter, dimana kita bisa meletakkan fungsi didalam nya. Hook akan bermanfaat ketika Anda ingin mengubah urutan loading library dimana Anda sudah meng – extend library router untuk menggunakannya. Artinya Anda harus me load library database sebelum library loader. Hal tersebut bisa dilakukan melalui hook. o MY. Jika kita ingin mengubah / meng extend library CodeIgniter maka library tersebut harus memiliki prefix yang sama dengan konfigurasi pesannya yang terdapat dalam PHP. Jika di set 2 maka akan menampilkan semua pesannya dan bug dan pesannya yang terdapat dalam informasi seperti sebuah library telah di load juga akan ditampilkan. Jika di set 4 maka semua pesannya lain melalui konfigurasi ini. config['sess_cookie_name'] = 'Namacookie'; config['sess_expiration'] = 7200; config['sess_encrypt_cookie'] = FALSE; config['sess_database'] = FALSE; config['sess_table_name'] = 'session_table'; config['sess_match_ip'] = FALSE; config['sess_match_useragent'] = TRUE; config['sess_time_to_update'] = 300;

- Autoload.php. Konfigurasi ini bertujuan untuk menentukan sumber daya apa yang akan di load secara otomatis. Cara penggunaannya sederhana, misal-

nya kita ingin meload library database, pagination dan lain-lain secara otomatis maka kita tinggal mengubah *autoload['libraries']* menjadi *:autoload['libraries']=array('database','session','pagination')*

- Routes.php. Konfigurasi di file ini bertujuan untuk menentukan kemana routing oleh library route akan dilakukan. Hal paling sederhana yang harus dilakukan adalah mengubah default controller (controller yang akan dibuka ketika tidak ada uri yang diberikan oleh browser). Misalnya website kita beralamat www.koder.web.id. Maka ketika membuka website tersebut maka secara otomatis CodeIgniter akan mengalihkan ke controller default, karena tidak disertakan di dalam url kita. Adapun yang perlu diubah adalah *route['default_controller'] = "welcome"*;

Kesepakatan Coding (Coding Standart) CodeIgniter

Sebelum melakukan coding menggunakan codeigniter maka ada baiknya kita mengetahui apasaja kesepakatan-kesepakatan yang ada di codeigniter. Kesepakatan-kesepakatan tersebut akan membuat kode kita lebih mudah dipahami oleh developer lainnya. Adapun kesepakatan tersebut diantaranya :

- PHP Closing Tag

Ketika kita menulis library, helper, controller ataupun model maka sebaiknya tidak menggunakan tanda penutup pada dokumen php *?>*. Hal tersebut dilakukan untuk mencegah adanya spasi atau karakter yang tidak diinginkan pada code kita sehingga membuat aplikasi error. Kita juga disarankan untuk memberikan informasi tentang akhir dokumen dan berisi path dokumen tersebut.

Contoh salah: *?>php echo "Here's my code!"; ?>*

Contoh Benar: *?>php echo "Here's my code!";*

/ End of file myfile.php */ /* Location: ./system/modules/mymodule/myfile.php */*

- Penamaan Class dan Method Penamaan Class harus dimulai dengan huruf besar. Jika class menggunakan beberapa kata maka kata-kata tersebut dipisahkan menggunakan underscore dan bukan camelcase.

Contoh salah: *class superclass class SuperClass* Contoh Benar: *class Super_class*

Aturan diatas juga berlaku untuk method contohnya

Contoh kurang tepat: *function fileproperties()* //Tidak deskriptif dan memiliki underscore *function fileProperties()* // Tidak deskriptif dan underscore CamelCase *function getFileproperties()* // Kurang underscore *function getFileProperties()* // menggunakan CamelCase *get_the_file_properties_from_the_file()* //terlalu panjang Contoh Tepat : *function getFile_properties()* //deskriptif, pakai underscore, dan huruf kecil

Chapter 3 Hello CodeIgniter Pada bab ini kita akan memulai coding menggunakan CodeIgniter. Sebelum memulai memulai maka perlu kita pahami kembali apa itu Controller? Dan bagaimana hubungan controller dengan URI? Apa sih Controller?

CodeIgniter adalah sebuah framework berbasis MVC. Sebuah Controller dapat dikatakan sebagai jantung dari suatu aplikasi, karena controller menentukan bagaimana permintaan HTTP yang harus ditangani. Sebuah kelas Controller adalah sebuah file yang terletak di dalam folder application/controller dan memiliki nama file yang sama dengan nama kelasnya dan dikaitkan dengan URL. Segmen-segmen pada URL pada codeigniter mencerminkan Controller yang dipanggil. Contoh: <http://www.domainsaya.com/class/fungsi/id> maka domain tersebut dapat dipecah menjadi bagian-bagian diantaranya:

Adapun komponen-komponen URL diatas adalah

- Konfigurasi Base Url, Bagian ini merupakan url yang kita masukkan pada konfigurasi base url yang merupakan url paling dasar untuk mengakses web atau aplikasi kita. Bagian ini merupakan Segmen URI pertama.

Untuk contoh pertama, kita akan membuat sebuah aplikasi hello codeigniter yang sederhana. Aplikasi tersebut akan diletakkan di folder hello dari htdocs anda, sehingga anda dapat mengaksesnya dengan membuka url `http://localhost/hello`. Perhatikan URL Berikut ini: `http://localhost/hello/index.php/blog/` Pada contoh di atas, CodeIgniter akan berusaha untuk menemukan sebuah controller bernama blog pada file `blog.php`, kemudian menampilkannya ke browser. Mari kita coba membuat sebuah controller sederhana sehingga dapat melihat apa yang terjadi. Dengan menggunakan teks editor, buatlah file bernama `blog.php` di dalam folder `application/controllers`. Setelah itu buatlah sebuah kelas `Blog` yang merupakan turunan dari kelas `CI_Controller`.

```
1. <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2. class Blog extends CI_Controller {
3.     function __construct() {
4.         parent::__construct();
5.     }
6.     function index() {
7.         // Your code here
8.     }
9.     function komentar() {
10.        // Your code here
11.    }
12. }
13. End of file Blog.php
14. Location: ./application/controllers/blog.php
```

Setelah itu apa yang terjadi? Coba buka situs Anda dengan menggunakan URL seperti ini:

`http://localhost/index.php/blog/`

Gambar 6. Tampilan Aplikasi Hello Codeigniter

Jika Anda melakukannya dengan benar, maka akan tampak tulisan Hello.. saya adalah contoh CodeIgniter pertama. Sekedar mengingatkan kembali, ketentuan penamaan class (baik controller maupun library) harus dimulai dengan huruf besar. Contoh:

```
<?php class Blog extends CI_Controller
```

Berikut ini contoh yang salah :

```
<?php class blog extends CI_Controller
```

Pada contoh di atas nama fungsi yang dipanggil adalah `index()`. Fungsi "Index" akan selalu dibaca secara default jika segmen kedua dari URI kosong. Cara lain untuk menampilkan "Hello CodeIgniter" adalah dengan mengakses url berikut: `localhost/index.php/blog/index/`

Segmen kedua dari URI yang menentukan fungsi mana yang akan dipanggil dari controller. Mari kita coba menambahkan fungsi baru ke controller:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Blog extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    function index() {
        echo "Haloo.. saya adalah contoh codeigniter pertama";
    }
    function komentar() {
        echo "Ini adalah fungsi komentar";
    }
}
End of file Blog.php
Location: ./application/controllers/blog.php
```

Sekarang cobalah panggil URL berikut untuk melihat fungsi komentar: `localhost/index.php/blog/komentar/`

Maka hasilnya akan menampilkan pesan "Ini Fungsi Komentar". Jika ingin mengisi variabel `$param` maka Anda tinggal menambahkan sebuah URI lagi setelah `Blog`. Biasanya parameter ketik `localhost/index.php/blog/komentar/tutorial-ci/123`

Fungsi `post` pada controller dapat memiliki dua parameter yang akan dilewatkan pada URI segmen 3 dan 4 ("tutorial-ci" dan "123"). Jadi dapat

disimpulkan bahwa URI yang dipanggil dapat ditunjukkan seperti ini: local-host/index.php/[Controller]/[Fungsi]/[param]/[param]/...

Controller dan View

Pada contoh program Hello CodeIgniter diatas adalah cara untuk menampilkan tulisan "hello CodeIgniter" secara langsung di controller. Namun sebenarnya hal tersebut bisa dilakukan di view. Sebagai contoh: tuliskan kode program di bawah ini pada file application/controller/blog.php :

```

<code>
<pre>
<code>

```

Selanjutnya buatlah file helloCodeIgniter.php di folder application > view(application/view/helloCodeIgniter.F
h1 > Hellosaya adalah view < /h1 >

Maka kode diatas akan memberikan hasil yang sama dengan contoh kasus pertama (tanpa menggunakan view), yang berbeda hanya tulisannya saja. Sebuah View sebenarnya hanyalah sebuah halaman web atau bagian dari halaman web, seperti sebuah header, footer, sidebar, dan lain-lain. Bahkan, View bisa menjadi fleksibel karena view dapat dimasukkan ke dalam view yang lain jika dibutuhkan. Untuk memanggil file view dapat digunakan fungsi seperti berikut ini (perhatikan baris 12):: this->load->view('nama_view');

Nama_view adalah nama file view Anda. Dan file tersebut harus diletakkan di dalam folder application/view.F

1. Nama file view - Nama file yang hendak di-load yang terletak di dalam folder application/view

2. Data Parameter - Parameter ini digunakan untuk melewati data dari controller ke dalam view.

Contoh: buatlah file controller dengan nama blog.php dengan kode program seperti berikut ini:

```

<code>
<pre>
<code>

```

Selanjutnya buatlah file view bernama blog_view.php dengan kode program seperti berikut ini :

```

<code>
<pre>
<code>

```

Jika halaman blog.php dipanggil maka tampilannya kurang lebih seperti berikut:

3. Output Parameter - Parameter ini akan di set true jika kita ingin menyimpan hasil view ke dalam sebuah variabel. Jika kita mengambil contoh code controller blog sebelumnya maka kita tinggal mengubah cara pemanggilan view menjadi

```

<code>

```

Code diatas berarti kita akan menyimpan hasil view kedalam sebuah variabel. Contohnya :

```

<code>

```

```
function index() this->load->view(\hello_codeigniter");
function view() data['judul'] = \Judulblog";data['isi'] = "Isi blog"; out =this->load->view("blog_view",data,true); echo out;
```

Mempercantik URL Codeigniter

Jika kita memperhatikan url yang ada di website kompas.com, okezone.com atau bejubel.com maka kita akan menemukan url yang unik. Contohnya <http://bejubel.com/jual/pakaian-wanita>, seakan-akan url tersebut bukan lah berasal dari script php biasa. Untuk membuat url seperti diatas

kita bisa menggunakan settingan codeigniter dan file .htaccess. adapun langkah yang dapat dilakukan adalah : 1. Membuat file .htaccess di folder root aplikasi anda. Adapun isi dari difile .htaccess adalah

```
RewriteEngine on RewriteCond !(index.php|images|robot.txt) RewriteRule (.*)
/index.php/1[L]2.Mengubahkonfigurasiapplication/configs/config.php.Membuang"index.php" padaconfig[
"index.php"; menjadiconfig['index_page'] = """; di file application/config/config.php Dengan menyelesaikan
//localhost/hello/index.php/blog/sudahdapatdiaksesdenganurlhttp : //localhost/hello/blog/tanpaindex.php
```

Dengan sampai pada tahap kedua sebenarnya sudah cukup untuk membuat url yang menarik. Selain itu untuk lebih advance lagi kita dapat menggunakan konfigurasi `application/config/router.php`.

Chapter 5 CodeIgniter Helper dan Library

CodeIgniter menyediakan dua jenis sarana yang dapat digunakan untuk membantu proses pengembangan aplikasi, antara lain: • Library Library dapat dikatakan sebagai kumpulan tools yang dapat digunakan untuk membantu sebuah proses. CodeIgniter telah menyediakan banyak library yang dapat digunakan secara langsung. Library pada dasarnya adalah sebuah kelas yang diletakkan di dalam folder `system/libraries` atau `application/libraries`. Library yang terletak di dalam folder `system` merupakan library bawaan dari CodeIgniter yang secara default di beri awalan `CI_`. Untuk library buatan sendiri harus diletakkan di dalam folder `application/libraries`. proses yang berulang, seperti generate html, url, security, dan lain – lain.

Menggunakan Library dan Helper di CodeIgniter Agar dapat menggunakan library, helper dan plugin, maka ketiganya harus di load terlebih dahulu. Ada dua cara yang dapat dilakukan untuk men-load sebuah library dan helper antara lain: 1. Menambahkan Pada Konfigurasi Autoload Menambahkan sebuah library di autoload berarti seluruh aplikasi Anda akan dapat menggunakan library tersebut secara langsung. Sebaiknya library yang di load dengan cara ini adalah jenis library yang dipakai di seluruh aplikasi seperti login, template, dan lain-lain. 2. Menggunakan Perintah Loader Library Kita dapat juga menggunakan library loader untuk men-load library. Library loader adalah sebuah library CodeIgniter yang otomatis di load. Loader berfungsi sebagai pengatur dari sumberdaya-sumberdaya yang ada di dalam CodeIgniter seperti Model, View, Library, Helper, dan plugin. Cara penggunaannya adalah: *this->load->library('nama_library');* *this->load->helper('nama_helper');* *this->load->plugin('nama_plugin');*

Nama library, helper dan plugin harus di isi dengan huruf kecil.

Ketika sebuah library sudah di-load maka library tersebut menjadi property pada object Controller. Adapun cara penggunaannya adalah sebagai berikut: *this->nama_library->fungsi();*

Library CodeIgniter Secara default CodeIgniter telah menyediakan library yang dapat digunakan secara langsung. Adapun library yang telah tersedia antara lain:

- Benchmarking Class Library ini digunakan untuk melakukan pengukuran terhadap aplikasi yang dibuat. Seperti untuk mengetahui berapa lama waktu eksekusi dan berapa jumlah memori yang digunakan. Library ini sudah digunakan dan di-load secara otomatis oleh CodeIgniter.
- Calendar Class Library ini berfungsi untuk menampilkan dan men-generate kalender.
- Cart Class Library ini berfungsi untuk membuat shopping cart (keranjang belanja). Library ini memiliki ketergantungan terhadap kelas session karena item-item chart tersebut disimpan di dalam session.
- Config Class Library ini berfungsi untuk mengambil data-data di dalam file konfigurasi. Library ini sudah di load secara otomatis oleh CodeIgniter.
- Database Class Library database digunakan untuk memanipulasi serta mendapatkan data dari sebuah sistem database. Secara default database yang sudah didukung oleh CodeIgniter adalah mysql, mssql, oracle, postgres. Sedangkan database yang tidak didukung secara langsung oleh CodeIgniter dapat di-jembatani dengan driver odbc.
- Email Class Library email digunakan untuk mengirimkan email. Pengiriman email tersebut bisa dilakukan dengan menggunakan protokol mail, sendmail dan smtp.sqawd
- Encryption Class Library Encryption digunakan untuk melakukan penyandian terhadap string tertentu
- File Uploading Class Library Uploading digunakan untuk meng-upload file. Kelas ini sudah dilengkapi dengan pengecekan jenis file, dan ukuran file.
- Form Validation Class Library form Validation digunakan untuk mengecek keabsahan form-form yang sudah di-submit oleh user.

- FTP Class Library FTP digunakan untuk meng-upload atau download file melalui ftp server.
- HTML Table Class Library HTML table adalah sebuah kelas yang berfungsi untuk men-generate table dari data array.
- Image Manipulation Class Library image manipulation berfungsi untuk mengolah gambar. Adapun fungsi-fungsi yang telah disediakan adalah Image Resizing, Thumbnail Creation, Image Cropping, Image Rotating dan Image Watermarking.
- Input and Security Class Library Input dan security berfungsi untuk menjamin bahwa inputan dari form telah bersih dari karakter-karakter “aneh”.
- Loader Class Library ini dapat disebut sebagai pengatur sumberdaya CodeIgniter. Semua sumberdaya yang ada akan dikendalikan oleh kelas ini. Library ini sudah di-load secara otomatis oleh CodeIgniter.
- Language Class Library language digunakan untuk mengatur bahasa apa yang akan dipakai oleh CodeIgniter.
- Output Class Library Output bertujuan untuk meng-handle output dari CodeIgniter, mulai dari cache sampai ke profiling bisa dilakukan kelas ini.
- Pagination Class Untuk mem-paginate hasil database untuk performance dan usability, kita bisa mengontrol berapa banyak record untuk ditampilkan disetiap halaman website, berapa banyak record untuk ditarik dari database dan tampilan dari bagian pagination
- Session Class Library Session dapat digunakan untuk memelihara informasi status tentang user (seperti layaknya session di PHP). Tetapi Library ini tidak menggunakan session built-in dari PHP, Library Session men-generate session datanya sendiri yang disimpan di dalam Cookies.
- Trackback Class Library Trackback digunakan untuk mengirim dan menerima data trackback.
- Template Parser Class Library Template Parser digunakan untuk membuat

template yang berisi parsable pseudo – templates. • Unit Testing Class Library Unit Testing digunakan untuk unit test function dalam aplikasi yang sedang dibuat. CodeIgniter menyediakan fungsi evaluasi dan dua fungsi hasil dalam library ini.

- URI Class Library URI digunakan untuk memarsing URL, lalu memecahnya ke dalam beberapa segmen dan kemudian di-passing ke controller atau disimpan sebagai variabel.
- User Agent Class Library User Agent digunakan untuk mengidentifikasi browser, mobile device, atau robot yang mengunjungi website. Kita juga bisa menggunakannya untuk mendeteksi dukungan bahasa, sekumpulan karakter, dan referrer.
- XML-RPC Class Library XML-RPC digunakan untuk men-setup klien XML-RPC dan server.
- Zip Encoding Class Library Zip Encoding digunakan untuk membuat file ZIP baik yang berjenis teks maupun data binary.

Helper CodeIgniter Helpers seperti namanya akan membantu Anda membangun aplikasi dengan tugas tertentu. Tidak seperti library, helper bukanlah Object Oriented tapi berupa prosedural. Setiap helper berisi satu atau lebih fungsi, masing-masing berfokus pada tugas tertentu yang tidak ada ketergantungan dengan fungsi lainnya. Helper dapat juga di-load secara otomatis dalam /system/application/config/autoload.php. Adapun helper yang ada di dalam CodeIgniter antara lain:

- Array — Helper array berisi fungsi yang membantu pekerjaan berhubungan dengan array. Sebagai contoh fungsi `random_element()` mengambil array sebagai input dan mengembalikan satu elemen acak dari array tersebut. `download_helper()` membantu untuk mengunduh file dari internet. `file_helper()` membantu untuk membaca, menulis, dan menghapus file. `form_helper()` membantu untuk membangun form. `html_helper()` membantu untuk membuat blok HTML. `language_helper()` membantu untuk mengubah kata-kata dari bentuk plural ke singular, dan sebaliknya. `pagination_helper()` membantu untuk mengelola nomor halaman. `string_helper()` membantu untuk mengelola string. `url_helper()` membantu untuk mengelola URL. `xml_helper()` membantu untuk mengelola XML.

- Text — Helper text berisi fungsi-fungsi yang membantu Anda bekerja dengan teks. Sebagai contoh: fungsi `word_limiter()` dapat membatasi jumlah kata tertentu yang digunakan dalam sebuah kalimat. Fungsi `truncate()` dapat memotong string yang lebih panjang dari yang ditentukan. Fungsi `url_title()` dapat mengubah string menjadi URL yang valid. Fungsi `strip_tags()` dapat menghapus semua tag HTML dari string. Fungsi `strip_slashes()` dapat menghapus semua slash dari string. Fungsi `strip_spaces()` dapat menghapus semua spasi dari string. Fungsi `strip_whitespace()` dapat menghapus semua spasi dan karakter kontrol dari string. Fungsi `strip_html_tags()` dapat menghapus semua tag HTML dari string. Fungsi `strip_image_tags()` dapat menghapus semua tag gambar dari string. Fungsi `strip_script_tags()` dapat menghapus semua tag skrip dari string. Fungsi `strip_style_tags()` dapat menghapus semua tag gaya dari string. Fungsi `strip_all_tags()` dapat menghapus semua tag dari string.

Membuat Library Sendiri Untuk mendapatkan kode yang bagus sebaiknya fungsionalitas yang sama itu di bungkus menjadi sebuah library. Sebuah library adalah sebuah kelas yang diletakkan pada folder application/libraries. `Get_instance()` adalah sebuah fungsi yang digunakan untuk mendapatkan instance dari sebuah library. `DataURI` dapat diakses dari library yang sudah di-load secara otomatis oleh CodeIgniter. Meng

CodeIgniter sangat mudah untuk mengadopsi library yang bersifat external. Library yang paling mudah untuk diadopsi adalah sebuah single class tanpa parameter pada constructor. Untuk kelas yang memiliki konstruktor lebih dari satu maka sebaiknya kelas tersebut diubah terlebih dahulu atau jika kelas library tersebut terdiri atas beberapa class maka lebih mudahnya kita membuat sebuah library factory dari library tersebut. Contoh 1. Library CSV Reader

Sebagai contoh kita akan menggunakan sebuah library yang berfungsi untuk membaca file CSV ke dalam memori.. Adapun isi dari librari tersebut adalah

```

i?php if(!defined('BASEPATH'))exit('No direct script access allowed'); /**
 * CSVReader Class * @author Pierre-Jean Turpeau * @link http://www.CodeIgniter.com/wiki/CSVReader
 */
class csv_reader
{
    var $fields; var $separator=';'; var $enclosure='\"'; var $max_row_size = 4096;
    /** * Parse a file containing CSV formatted data. * * @access public *
    @param string * @param boolean * @return array */
    function parse_file($filepath, $namedFields = true) {
        $content = false; $file =
        fopen($filepath, 'r'); if ($namedFields) {
            $fields = fgetcsv($file, $this->max_row_size, $this->
            $separator, $this->enclosure);
            while (($row = fgetcsv($file, $this->max_row_size, $this->
            $separator, $this->enclosure)) != false) {
                if (row[0] != null) // skip empty lines
                    if (!$content) $content = array();
                if ($namedFields) {
                    $items = array();
                    foreach ($this->fields as $id => $field) {
                        if (isset($row[$id])) {
                            $items[$field] = $row[$id];
                        }
                    }
                }
                return $content;
            }
        }
    }
}

```

Agar dapat menggunakan library tersebut maka lakukan langkah-langkah berikut ini:

- Copy Library Ke Direktori application/libraries Untuk menggunakan library buatan kita sendiri atau library eksternal maka langkah yang pertama yang harus dilakukan adalah meletakkan library itu di direktori application/libraries.

Nama file dan nama kelas harus sama. Contoh nama file adalah `csv_reader.php` maka nama kelasnya adalah `csv_reader`.

Perhatikan controller berikut ini

```

1. i?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2. 3. class Welcome extends CI_Controller {
    function __construct() {
        parent::__construct();
    }
    function index() {
        $load => library('csv_reader');
        $data = $this->csv_reader->parse_file(FCPATH.'data_nilai.csv');
        echo $pre > ";
        print_r($data);
    }
}
/* End of file welcome.php */
/* Location: ./application/controllers/welcome.php */

```

Pada baris ke-13 kita akan mencoba menggunakan library `csv_reader`, kelas tersebut akan memparsing sebuah file CSV. Lalu hasil parsingnya akan ditampilkan.

Chapter 6 Kasus 1. Penanganan dan Validasi Form Pada bab ini akan dibahas mengenai penanganan form dan cara melakukan validasi terhadap sebuah form inputan. Penanganan form artinya bagaimana cara kita untuk mendapatkan dan mengolah data yang dikirimkan oleh pengguna melalui form. Untuk membuat form sendiri, sebaiknya kita menggunakan helper form. Setelah kita berhasil mendapatkan data dari user, kita harus memvalidasi data-data yang telah dikirimkan tersebut agar aplikasi yang kita bangun menjadi aman. Inputan yang tidak divalidasi membuat aplikasi kita rentan terhadap serangan seperti Sql Injection, Xss dan lain-lain. Penanganan Form

Untuk menangani form kita membutuhkan library input. Library tersebut sudah dipanggil secara otomatis oleh codeigniter. Fungsi yang tersedia pada library ini selain untuk menangani form juga memiliki fungsi security/keamanan. Contohnya untuk menghandle atau memfilter xss kita dapat menggunakan fungsi `xss_filter.nama = $this->input->xss_filter($this->input->post('nama'))`;

Jika kita ingin melindungi seluruh aplikasi dari dengan xss filter maka kita bisa menconfignya dengan mengubah konfigurasi di `system/application/config/config.php` dari `config['global_xss_filtering'] = FALSE`;

Menjadi

`config['global_xss_filtering'] = TRUE`;

Selain kedua cara diatas, kita dapat juga menggunakan parameter kedua dari fungsi post atau get maka kita akan mendapatkan hasil yang sama dengan kedua cara diatas, contohnya `this->input->post('nama',TRUE);` `this->input->get('nama', TRUE);` Fungsi post digunakan untuk menangkap inputan POST sedangkan get digunakan untuk menangkap inputan GET. Selain itu fungsi input juga dapat digunakan untuk cookie, variabel SERVER, dan user agent `this->input->cookie('nama_cookie',TRUE);` `this->input->ipaddress();` `this->input->server('PATH_INFO',TRUE);`

Selain library input, Codeigniter juga menyediakan sebuah helper untuk mempermudah penanganan form. Helper tersebut adalah helper form. Helper tersebut membantu kita dalam membuat form pada view. Adapun cara meload helper form adalah `this->load->helper('form');`

Ada banyak fungsi yang disediakan oleh helper ini diantaranya `form_open()`, `form_close()`, `form_open_multipart()` *lain.* Untuk lebih detailnya anda dapat melihat pada dokumentasi codeigniter. Pada kasus ini, fungsi yang banyak digunakan adalah `form_open()` pada html. Contoh :

```
attributes = array('class' => 'email', 'id' => 'myform'); echo form_open('form/send', attributes);
Maka akan menghasilkan
<form method="post" accept-charset="utf-8" action="http://localhost/index.php/email/send"
class="email" id="myform" />
```

Fungsi input berfungsi untuk menghasilkan tag input pada html. Fungsi ini dapat dipakai dengan dua cara. Cara pertama adalah dengan menggunakan `echo form_input('username', 'ibnoe');`

akan menghasilkan `<input type="text" name="username" id="username" value="ibnoe" />`
 Cara kedua adalah dengan mempassing array kedalam fungsi tersebut. Cara ini adalah cara yang advance dan sangat berguna jika kita ingin mengcustomisasi inputan. Contohnya: `data = array('name' => 'username', 'id' => 'username', 'value' => 'ibnoe', 'maxlength' => '100', 'size' => '50', 'style' => 'width: 50');`

```
echo form_input(data);
Akan menghasilkan output :
<input type="text" name="username" id="username" value="ibnoe" maxlength="100"
size="50" style="width:50
```

Untuk contoh menangani form maka ikutilah langkah-langkah berikut ini.

1. Membuat Controller Hitung

Perlu diingat hal pertama yang harus di konfigurasi pada setiap aplikasi adalah `BASE_URL`. Setelah melakukan konfigurasi `base_url` pada `file application/config/config.php` maka kita akan melakukan konfigurasi `hitung.php`.
`<?php if (!defined('BASEPATH')) exit('No direct script access allowed');`
`$load->helper(array('url','form'));` 9. 10. 11. `function index()` 12. 13. `this->load->view('menu_hitung');` 14. 15. 16. `function perkalian()` 17. 18. `data['v1']=(int)$this->input->post('v1',true);` 19. `data['v2']=(int)$this->input->post('v2',true);` 20. `data['hasil']=data['v1']*data['v2'];` 21. `this->load->view('perkalian',data);` 22. 23. 24. `function pembagian()` 25. 26. `data['v1']=(int)$this->input->post('v1',true);` 27. `data['v2']=(int)$this->input->post('v2',true);` 28. `if (data['v2'] > 0)` 29. `data['hasil']=data['v1']/data['v2'];` 30. `else` 31. `data['hasil']='Error, v2 tidak boleh 0!';` 32. `this->load->view('pembagian',data);` 33. 34.

Pada baris ke 8, dilakukan loading terhadap helper url dan form sekaligus. Loading tersebut dilakukan dalam fungsi konstruktor karena kedua helper


```

</h1> 38. 39. <p>Silakan masukan data berikut!!</p> 40. <?php
echo form_open('hitung/pembagian');? > 41. <?php echo form_input('v1',v1);?>
/ 42. <?php echo form_input('v2',v2);?><br> 43. 44. <?php echo form_submit('submit','Hitung!!');? >
45. <?php echo form_close();? ><br> 46. Hasil :<?php echo hasil;?> 47. 48.
</br></Page rendered in elapsed time seconds </p> 49. </body></html>

```

Berikut ini adalah tampilan dari form pembagian

Gambar 9. Screenshoot view pembagian

Catatan :

Menggunakan Validasi Form

Sebenarnya sampai pada tahap kedua di atas kita telah mampu membuat sebuah aplikasi perkalian dan pembagian sederhana, tetapi sebuah aplikasi yang baik harus memiliki sebuah kontrol terhadap inputan user. Validation form sangat penting pada aplikasi. Sebuah aplikasi akan memiliki banyak inputan dari pengguna dan semua inputan untuk pengguna harus aman. CodeIgniter memiliki sebuah library Form Validation Library yang akan membantu kita untuk membuat sebuah validasi yang cepat, mudah dan aman. Pada contoh validasi form ini hampir sama dengan 2 langkah diatas, tetapi memiliki sedikit penambahan terutama dalam fungsi perkalian dan pembagian. Perhatikan controller hitung berikut ini (controller berikut ini berasal dari contoh sebelumnya dengan penambahan).

```

Application/controllers/hitung.php 1. <?php if (! defined('BASEPATH')) exit('No direct script access allowed'); 2. 3. class Hitung
extends CI_Controller { 4. function __construct() { 5. parent::__construct(); 6. 7. 8. $this->load-
helper(array('url','form')); 9. 10. 11. function index() { 12. 13. $this->
load->view('menu_hitung'); 14. 15. 16. 17. function perkalian() { 18. 19. $this->load-
library('form_validation'); 20. $this->form_validation->set_rules('v1','Variabel1',21.'required|integer'); 22. $this->form_validation->set_rules('v2','Variabel2',23.'required|integer'); 24. if($this->form_validation->run()) { 25. 26. $int=$this->input->post('v1',true); 27. $int=$this->input->post('v2',true); 28. $data['hasil']=$int*$int; 29. 30. else { 31. 32. 33. 34. $data['hasil']=0; 35. 36. $this->load->view('perkalian',$data); 37. 38. 39. function pembagian() { 40. 41. $this->load->library('form_validation'); 42. $this->form_validation->set_rules('v1','Variabel1',43.'required|is_natural|zero'); 44. $this->form_validation->set_rules('v2','Variabel2',45.'required|is_natural|zero'); 46. if($this->form_validation->run()) { 47. 48. $int=$this->input->post('v1',true); 49. $int=$this->input->post('v2',true); 50. $data['hasil']=$int/$int; 51. 52. else { 53. 54. 55. 56. $data['hasil']=0; 57. 58. 59. $this->load->view('pembagian',$data); 60. 61. 62.

```

Perhatikan baris 19-24 pada fungsi perkalian, disana kita me-load sebuah library bernama `form_validation`. Library tersebut memiliki fungsi untuk melakukan validasi terhadap inputan user. `$this->form_validation->set_rules('v1','Variabel1','required|integer');`

Baris di atas merupakan salah satu contoh untuk menentukan aturan inputan pada form. Parameter pertama (berisi v1) merupakan name dari inputan yang akan divalidasi (harus sama dengan attribut name pada tag input contoh: `<input name="v1">`). Parameter kedua adalah nama dari inputan tersebut. Nama tersebut dapat berbeda atau sama dengan name pada parameter pertama. Parameter ketiga merupakan aturan dari inputan tersebut. Aturan-aturan tersebut harus dipatuhi agar sebuah form dapat diproses. Aturan-aturan

tersebut dipisahkan oleh tanda —. Pada contoh di atas berarti inputan v1 harus diisi (required) dan berisi data integer(integer). Adapun aturan-aturan yang tersedia di dalam CodeIgniter adalah:

Aturan Keterangan required Isi inputan tidak boleh kosong. matches Isi inputan harus sama dengan inputan tertentu contoh

matches[password] minlength Panjang inputan harus memiliki jumlah minimal karakter, contoh penggunaan z.alpha_numeric Inputan harus berisi karakter alfabet dan numeric.alpha_dash Inputan harus berisi semua huruf kapital, underscore, dan dashes.numeric Inputan hanya boleh berisi angka/huruf numeric.integer Inputan hanya boleh 0, 1, 2, 3, dan seterusnya.is_natural_or_zero Inputan hanya boleh berisi bilangan natural saja kecuali nol : 1, 2, 3, dan seterusnya.valid_email Inputan harus berisi format email yang benar.valid_emails Inputan harus berisi

Pada baris-46, kita melakukan pengecekan terhadap rule-rule yang telah diset (*this* → *form_validation* → *run()*). Fungsi tersebut akan menghasilkan nilai TRUE apabila semua

rule terpenuhi dan menghasilkan nilai FALSE jika sebaliknya. Selain fungsi-fungsi diatas Anda juga dapat membuat custom validation jika aturan-aturan di atas tidak ada yang memenuhi dengan kebutuhan Anda. 1. <!DOCTYPE html> 2. <html lang="en"> 3. <head> 4. <meta charset="utf-8"> 5. <title>Perkalian menggunakan validasi!!</title> 6. 7. <style type="text/css"> 8. 9. body 10. background-color: fff; 11. margin: 40px; 12. font-family: Lucida Grande, Verdana, Sans-serif; 13. font-size: 14px; 14. color: 4F5155; 15. 16. 17. a 18. color: 003399; 19. background-color: transparent; 20. font-weight: normal; 21. 22. 23. h1 24. color: 444; 25. background-color: transparent; 26. border-bottom: 1px solid D0D0D0; 27. font-size: 16px; 28. font-weight: bold; 29. margin: 24px 0 2px 0; 30. padding: 5px 0 6px 0; 31. 32. 33. </style> 34. </head> 35. <body> 36. 37. <h1>Perkalian!</h1> 38. <?php echo validation_errors();? > 39. < p > Silakan masuk dan data berikut!! < /p > 40. < ?php echo form_open('hitung/perkalian');? > 41. < ?php echo form_input('v1',v1);? > 42. < ?php echo form_input('v2',v2);? > 43. < ?php echo form_submit('submit','Hitung!!');? > 45. < ?php echo form_close();? > < br > 46. Hasil : < ?php echo hasil;? > 47. 48. < /br> < /body> < /html>

Pada view, kita cukup menambahkan satu fungsi yang berguna untuk menampilkan kesalahan yang terjadi. Perhatikan baris 38. Fungsi validation_errors() adalah fungsi untuk menampilkan kesalahan

Gambar 10. Form perkalian dengan tampilan error

Begitu juga untuk view pembagian, kita tinggal menambah fungsi validation_errors() untuk menampilkan error

1. <!DOCTYPE html> 2. <!DOCTYPE html> 3. <html lang="en"> 4. <head> 5. <meta charset="utf-8"> 6. <title>Pembagian menggunakan validasi!!</title> 7. 8. <style type="text/css"> 9. 10. body 11. background-color: fff; 12. margin: 40px; 13. font-family: Lucida Grande, Verdana, Sans-serif; 14. font-size: 14px; 15. color: 4F5155; 16. 17. 18. a 19. color: 003399; 20. background-color: transparent; 21. font-weight: normal; 22. 23. 24. h1 25. color: 444; 26. background-color: transparent; 27. border-bottom: 1px solid D0D0D0; 28. font-size: 16px; 29. font-weight: bold; 30. margin: 24px 0 2px 0; 31. padding: 5px 0 6px 0; 32. 33. 34. 35. </style> 36. </head> 37. <body> 38. 39. <h1>pembagian!</h1> 40. <?php echo validation_errors();? > 41. < p > Silakan masuk dan data berikut!! < /p > 42. < ?php echo form_open('hitung/pembagian');? > 43. < ?php echo form_input('v1',v1);? >

```

/ 44. <?php echo form_input('v2',v2);?><br> 45. 46. <?php echo form_submit('submit','Hitung!!');? >
47. <?php echo form_close();? ><br> 48. Hasil :<?php echo hasil;?> 49. 50.
<?php echo $Page rendered in elapsed_time seconds < /p > 51.52. < /body ><
/html >

```

View diatas akan memberikan tampilan seperti gambar 11 berikut ini.

Gambar 11. Form perkalian dengan tampilan error

Chapter 7 CodeIgniter Database

CodeIgniter mendukung banyak jenis database misalnya MySQL, PostgreSQL, Oracle, dan lain-lain. Dukungan database dari CodeIgniter berupa penyediaan beberapa driver database yang sekaligus juga memiliki fungsi sekuriti, caching dan active record.

Connect ke Database

CodeIgniter memiliki sebuah file konfigurasi yang memungkinkan Anda menyimpan konfigurasi untuk melakukan koneksi ke database (username, password, nama database, dan lain-lain). File konfigurasi terletak berada di application/config/database.php. Pengaturan konfigurasi disimpan dalam bentuk array multi-dimensi dengan prototipe seperti berikut ini: `db['default']['hostname'] = "localhost"; db['default']['username'] = "root"; db['default']['password'] = ""; db['default']['database'] = "database"; db['default']['dbprefix'] = ""; db['default']['pconnect'] = TRUE; db['default']['db_debug'] = FALSE; db['default']['cache_on'] = FALSE; db['default']['cachedir'] = ""; db['default']['charset'] = "utf8"; db['default']['dbcollat'] = "utf8_general_ci";`

Kita dapat memiliki beberapa konfigurasi database. Alasan kita menggunakan array multi-dimensi adalah agar memungkinkan Anda secara opsional menyimpan beberapa set nilai-nilai koneksi. Jika, misalnya, Anda menjalankan aplikasi di beberapa kondisi lingkungan/environment (development, production, testing, dll) di bawah satu instalasi, maka Anda dapat mengatur kelompok konfigurasi untuk masing-masing environment. Sebagai contoh, Anda sedang menjalankan aplikasi di environment test maka Anda tinggal menambahkan konfigurasi berikut ini dan mengubah default konfigurasinya (mengeset `active_group = "test"`) menjadi test: `db['test']['hostname'] = "localhost"; db['test']['username'] = "root"; db['test']['password'] = ""; db['test']['database'] = "database_name"; db['test']['dbdriver'] = "mysql"; db['test']['dbprefix'] = ""; db['test']['pconnect'] = TRUE; db['test']['db_debug'] = FALSE; db['test']['cache_on'] = FALSE; db['test']['cachedir'] = ""; db['test']['charset'] = "utf8"; db['test']['dbcollat'] = "utf8_general_ci"; active_group = "test";`

Untuk connect ke database ada beberapa cara yang disediakan oleh CodeIgniter diantaranya :

1. Menambahkan Database Library Sebagai Autoload Library Untuk connect ke database Anda bisa menambahkan database sebagai autoload library di file application/config/autoload.php. Cara ini sangat sederhana, cukup menambahkan kata "database" ke dalam autoload library sehingga menjadi : `autoload['libraries'] = array("database");`

Ketika menambahkan autoload pastikan bahwa database sudah terkonfigurasi dengan benar untuk menghindari kesalahan.

2. Mengaktifkan Manual Dari Library Database Jika hanya ada beberapa halaman website yang memerlukan konektivitas database, maka untuk optimalisasi lakukan koneksi

ke database secara manual, cukup dengan menambahkan baris kode di bawah ini pada tiap fungsi tempat yang membutuhkan koneksi ke database atau dalam konstruktor kelas Anda untuk membuat database yang tersedia secara global di kelas. *this->load->database()*;

Jika fungsi di atas tidak berisi informasi apapun di parameter pertama akan menyambung ke group konfigurasi yang aktif. Untuk memilih kelompok tertentu dari file konfigurasi, Anda dapat melakukan seperti pada contoh berikut. Hal tersebut berguna pada saat menggunakan aplikasi yang memiliki dua database. *this->load->database('group_name')*;

Group_name adalah nama grup konfigurasi dari file konfigurasi Anda. Untuk menghubungkan secara manual *config['username'] = "myusername"; config['password'] = "mypassword"; config['database'] = "mydatabase"; config['dbdriver'] = "mysql"; config['dbprefix'] = ""; config['pconnect'] = FALSE; config['debug'] = TRUE; config['cache_on'] = FALSE; config['cachedir'] = ""; config['charset'] = "utf8"; config['dbcollat'] = "utf8_general_ci";*
this->load->database(config);

Atau Anda bisa mengirimkan nilai-nilai database Anda sebagai Data Source Name. DSN harus memiliki prototipe seperti ini: *dsn = 'dbdriver://username:password@hostname/database'; this->load->database(dsn)*;

3. Mengaktifkan manual dari model Selain kedua cara di atas Anda juga bisa mengaktifkan database pada saat loading model. Caranya adalah dengan mengeset TRUE pada parameter ketiga load model. Contoh : *this->load->model('Model_name', "", TRUE)*;

Selain itu Anda juga bisa menggunakan konfigurasi tertentu untuk sebuah model. Contoh:

manual_config['hostname'] = "localhost"; manual_config['username'] = "myusername"; manual_config['password'] = "mypassword"; manual_config['database'] = "mydatabase"; manual_config['dbdriver'] = "mysql"; manual_config['dbprefix'] = ""; manual_config['pconnect'] = FALSE; manual_config['debug'] = TRUE; this->load->model('Model_name', "", manual_config);

CodeIgniter Model

Model pada CodeIgniter adalah sebuah kelas php yang berfungsi untuk menangani data. Ingat data bukan hanya dari database tetapi juga bisa dari File Text, Web Service atau layanan-layanan data lainnya. Contoh Model: class Blogmodel extends Model

var title = ""; var content = ""; var date = "";
function Blogmodel() parent::Model();
function get_data() query = this->db->get('entries', 10); Return query->result();

Sebuah model sebenarnya tidak harus meng-extend class Model. Kelas model di-extend ketika hendak menggunakan fitur database pada CodeIgniter saja. Semua Model harus diletakkan di dalam folder application/models. Agar dapat menggunakan model maka kita harus me-load model tersebut. Adapun perintah yang dapat digunakan untuk memload sebuah model adalah *this->load->model('Model_name')*; *this->Model_name->get_data()*;

Ketika sudah berhasil me-load sebuah model maka model tersebut akan menjadi sebuah property. Melalui property itulah Anda akan menggunakan semua

fungsi yang ada di dalam model (perhatikan baris ke 2 pada contoh diatas *get_data* adalah fungsi di dalam kelas *Model_name*)

Melakukan Query pada Database

Untuk mendapatkan data pada database Anda harus melakukan query. Setelah query dilakukan barulah bisa mendapatkan data hasil query tersebut, baik dalam bentuk object maupun array. *query = this->db->query('QUERY SQL ANDA');*

Perlu ditekankan bahwa fungsi query di atas belum menghasilkan data apapun. Keluarannya hanya berupa Object(true) atau False. Ketika keluarannya adalah False maka query yang dilakukan berarti gagal. Tetapi jika true atau mengembalikan sebuah object maka query yang

dilakukan berarti berhasil. Dari object tersebut (variabel *query* kalau mengacu contoh diatas) Anda dapat men
db-> query('SELECT name, title, email FROM my_table');

foreach(query-> result() as row) echo row-> title; echo row-> name; echo row-> email;

echo 'Total Results: ' . query-> num_rows();

Contoh di atas adalah salah satu contoh penggunaan query (tapi ingat, pastikan Anda sudah memiliki konfigurasi database yang benar dan telah me-load library database. Perhatikan *query-> result()*. Itu adalah contoh syntax untuk mendapatkan hasil query dan *result_array()* untuk mendapatkan hasil query dalam bentuk array asosiatif.

Query Return Value Seperti yang telah disebutkan di atas, setelah query dilakukan kita perlu memanggil fungsi tertentu untuk mendapatkan hasil dari query. Secara umum ada dua jenis tipe data yang dapat dihasilkan yaitu array dan object. Dari segi jumlah kita hanya dapat mengambil satu record (per record) atau keseluruhan record. • Mengembalikan Hasil Query Sebagai Kumpulan Array. Sebelum mengambil hasil query, ada baiknya mengecek terlebih dahulu apakah query tersebut memiliki hasil atau tidak. Perhatikan fungsi *result-> result_array()*, itu adalah method yang digunakan untuk mengambil hasil query. Sedangkan fungsi *result-> num_rows()* digunakan untuk mengetahui berapa jumlah record yang didapat. *result = this-> db-> query('SELECT * FROM users');* *if (result-> num_rows() > 0) foreach (result-> result_array() as row) echo row['username']; echo row['email'];* • Mengembalikan Hasil Query Sebagai Kumpulan Object. Fungsi yang digunakan hampir sama dengan mengembalikan nilai query sebagai array. Cukup dengan memanggil method *result-> result()*.

*result = this->db->query('SELECT * FROM users');* *if (result-> num_rows() > 0) foreach (result-> result() as row) echo row-> username; echo row-> email; // Mengembalikan Hasil Query Sebagai Array*
*db-> query('SELECT * FROM users');* *if (query-> num_rows() > 0) row = query-> row(); echo row['username']; echo row['email']; // Mengembalikan Hasil Query Sebagai Row Object jadi kita hanya*
*db-> query('SELECT * FROM users');* *if (query-> num_rows() > 0) row = query-> row(); echo row-> username; echo row-> email;*

Menggunakan Active Record

Active Record (AR) adalah sebuah pattern / pendekatan untuk membaca data dari sebuah table atau view dengan cara membungkusnya dalam sebuah kelas. Sehingga tidak dibutuhkan SQL jika kita menggunakan active record. Keuntungan yang didapat adalah kesederhanaan dalam pengkodean program dan fleksibilitasnya ketika terjadi pergantian skema atau pun jenis database

(bebas dari sintaks-sintaks SQL yang bersifat khusus ke database tertentu). CodeIgniter tidak sepenuhnya meniru pattern Active Record, tetapi ia melakukan sedikit modifikasi terhadap pattern tersebut. Dengan menggunakan Active Record CI maka proses pengambilan data, insert, update dan delete menjadi lebih sederhana. *this->db->get('users')*

Query di atas akan menghasilkan query “SELECT * FROM users” dan setara dengan

```
this->db->query('SELECT * FROM users');
```

Ketika kita tidak menggunakan fungsi apapun maka fungsi get akan menghasilkan seluruh record yang berada di dalam tabel tersebut, tetapi jika diberi fungsi select(), where(), dan lain-lain maka hasilnya akan berbeda.

Selecting Data

Untuk mengambil data dari database dengan field-field tertentu kita dapat menggunakan perintah select(). Contoh: *this->db->select('username,password,email');*
this->db->get('users');

Query di atas setara dengan *this->db->query('SELECT username,password,email FROM users');*

Selain method select, terdapat juga method-method lainnya seperti from(), where(), where_in(), group_by, order_by, like, distinct() dan lain-lain. Semua method tersebut jika dipanggil akan menghasilkan query yang berbeda. Contoh: *this->db->select('sum(budget) as sum_budget, site_id mapping.*')*

```
->where('site_id mapping.site_id','site_id')->where('ad_status','approved')->where('request_status','active')
```

```
->where('ad_operational_status','active')
```

```
->join('ads','ads.id_ad = site_id mapping.ad_id')->join('advertisers','id_advertiser = advertiser_id')->join('site_counter','site_counter.site_id = site_id')->join('sites','site_counter.site_id = sites.id_site')->join('publishers','sites.publisher_id = id_publisher')
```

```
->from('site_id mapping')
```

```
->order_by('ad_priority_level','desc')->order_by('point','desc')
```

```
->group_by('ad_id')->limit(0,10)->get()->result();
```

Insert Data

Selain pengambilan data, Active record juga menyediakan fungsionalitas untuk insert data. Untuk menginsert sebuah data kita hanya memerlukan dua parameter yaitu nama table dan data yang akan dimasukkan. Data yang akan dimasukkan harus memiliki format berupa array asosiatif ataupun object. Adapun keuntungan menggunakan active record untuk menginsert data selain sederhana untuk digunakan, adalah method ini secara build-in disertai fungsi untuk penanganan escaping, jadi query kita sudah bisa dikatakan aman dari kesalahan, contoh: *data = array('username' => 'Ibnoe','email' => 'xibnoe@gmail.com');*
this->db->insert('users',data);

Selain dengan cara di atas, kita juga bisa memasukkan data yang diinginkan satu persatu. Contohnya: *data = array('username' => 'Ibnoe','email' => 'xibnoe@gmail.com');*
this->db->set(data);this->db->insert('users');

Atau

```
this->db->set('username','ibnoe');  
this->db->set('email','xibnoe@gmail.com');  
this->db->insert('users');
```

Kedua Contoh di atas menghasilkan query yang sama.

Update Data

Selain insert dan select data, CodeIgniter juga menyediakan fungsi untuk update. Adapun penggunaannya mirip seperti penggunaan insert, contoh:

```
this->db->set('username','ibnoe');this->db->set('password','123456');  
this->db->update('users');
```

Query di atas akan menghasilkan query “update users set username=ibnoe, password=123456”. Cara yang lebih sederhana dengan menggunakan array asosiatif *data = array('username' => 'ibnoe','password' => '123456');**this->db->where('id',5); this->db->update('users',data);*

Delete Data

Untuk melakukan delete pada CodeIgniter, kita dapat menggunakan perintah berikut ini:

```
this->db->where('id',5);this->db->delete('table_name');
```

Chapter 8 Kasus 2. CRUD dan Pagination Database Setelah mengetahui cara menggunakan database maka kita akan mencoba melakukan perintah dasar pada sistem informasi CRUD (create, read, update dan delete) data menggunakan CodeIgniter. Selain CRUD, pada contoh ini juga akan dilakukan proses sorting dan pagination. Studi kasus yang akan diangkat adalah form data siswa. Pada bab ini aplikasi CRUD yang akan dibuat adalah aplikasi yang memiliki fungsi untuk menampilkan seluruh data siswa. Data yang ditampilkan memiliki pagination (pembagian jumlah record yang ditampilkan pada sebuah halaman website). Pagination dibutuhkan karena kita tidak mungkin menampilkan semua data dalam satu layar. Selain itu daftar siswa tadi akan memiliki fitur pengurutan berdasarkan field dan memiliki menu action per daftar siswa. Untuk melakukan pagination di Codeigniter telah disediakan sebuah kelas yaitu kelas pagination. Untuk menggunakan kelas ini minimal harus memiliki tiga data yang selanjutnya akan menjadi konfigurasi pada library ini. Ketiga data tersebut adalah pagination *base_url, total_row(jumlah total baris data record yang kita punya) dan per_page(jumlah baris yang akan ditampilkan per halaman)*.
\$load->library('pagination');
\$this->load->helper('url');
\$config['base_url'] = site_url('results/page/');
\$config['total_rows'] = 200;
\$config['per_page'] = 20;
\$pagination->initialize(\$config);echo \$this->pagination->create_links();

Contoh diatas menunjukkan cara penggunaan library pagination. Output dari library ini adalah link- link yang terdiri atas halaman dan link ke halaman berikut (next) atau sebelumnya (prev). *Base_url harus berisi sebuah fungsi dari controller. Fungsi tersebut akan menerima parameter di atas masi banyak parameter lainnya yang dapat digunakan untuk mengkonfigurasi library pagination*
3; //mengganti tulisan next menjadi Berikutnya > \$config['next_link'] = 'Berikutnya'; //mengganti tulisan prev menjadi Sebelumnya < \$config['prev_link'] = 'Sebelumnya';

Adapun langkah-langkah yang harus dilakukan untuk membuat CRUD dan pagination adalah

1. Membuat Database Dan Table Data Siswa

Untuk memulai, Anda harus memiliki sebuah database. Selain database, Anda harus membuat sebuah tabel siswa. Adapun table yang harus di buat adalah CREATE TABLE IF NOT EXISTS 'siswa' ('id' bigint(20) NOT NULL AUTO_INCREMENT, 'nama' varchar(50) DEFAULT NULL, 'alamat' varchar(200) NOT NULL, 'jenis_kelamin' enum('laki-laki','perempuan') NOT NULL, 'tanggal_lahir' date NOT NULL, PRIMARY KEY (id)) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT = 1;

Tabel siswa di atas memiliki lima field yaitu id siswa, nama siswa, alamat siswa, jenis kelamin, dan tanggal lahir. Tabel ini digunakan untuk menyimpan

data siswa. 2. Konfigurasi CodeIgniter

Sebelum memulai pengkodean, sebaiknya kita melakukan konfigurasi terhadap tools yang akan digunakan. Ada tiga konfigurasi yang harus disesuaikan, antara lain: config.php, database.php dan router.php. Adapun yang harus dilakukan adalah:

- config.php - pada file konfigurasi ini yang perlu diubah adalah

base_url. Konfigurasi ini berfungsi untuk menentukan path dasar dari aplikasi. Konfigurasi ini akan digunakan dalam konfigurasi database.php dan router.php.

```
config['base_url'] = 'http://localhost/crud'; // ganti sesuai path di server Anda  
// database.php  
db['default']['hostname'] = 'localhost';  
db['default']['username'] = 'root';  
db['default']['password'] = ''  
;db['default']['database'] = 'crud';  
db['default']['dbdriver'] = 'mysql';  
// router.php
```

berisi konfigurasi mengenai default controller dan routing rule. Pada konfigurasi ini yang diubah hanya default controller dan routing rule.

”siswa”;

3. Membuat Model Siswa