

Pemrograman **Web** dengan menggunakan **PHP** dan **FRAMEWORK CODEIGNITER**

Supono
Viridiandry Putratama



**PEMROGRAMAN WEB
DENGAN MENGGUNAKAN PHP
DAN FRAMEWORK CODEIGNITER**

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

PEMROGRAMAN WEB DENGAN MENGGUNAKAN PHP DAN FRAMEWORK CODEIGNITER

Supono
Vidiandry Putratama





deepublish | publisher

Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoharjo, Ngaglik, Sleman
Jl.Kaliurang Km.9,3 – Yogyakarta 55581
Telp/Faks: (0274) 4533427
Website: www.deepublish.co.id
www.penerbitdeepublish.com
E-mail: cs@deepublish.co.id

Katalog Dalam Terbitan (KDT)

SUPONO

Pemrograman Web dengan Menggunakan PHP dan Framework
Codeigniter/oleh Supono dan Vidiandry Putratama.--Ed.1, Cet. 1--
Yogyakarta: Deepublish, Mei-2018.

xi, 206 hlm.; Uk:17.5x25 cm

ISBN **978-602-475-280-4**

1. Pemrograman Web

I. Judul
005.262

Hak Cipta 2018, Pada Penulis

Desain cover : Herlambang Rahmadhani
Penata letak : Cinthia Morris Sartono

PENERBIT DEEPUBLISH
(Grup Penerbitan CV BUDI UTAMA)

Anggota IKAPI (076/DIY/2012)

Copyright © 2018 by Deepublish Publisher
All Right Reserved

Isi diluar tanggung jawab percetakan

Hak cipta dilindungi undang-undang

Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa izin tertulis dari Penerbit.

PRAKATA

Bismillahirrahmaanirrahim...

Puji dan syukur hanya milik Allah atas berkat rahman dan rahim-Nya lah buku ini dapat diselesaikan dengan keterbatasan yang kami miliki.

Perkembangan internet saat ini begitu pesat, terlihat dari jumlah pertumbuhan pengguna yang mengakses internet atau website serta munculnya website-website baru. Faktor lain semakin yang membuat semakin pesatnya perkembangan internet dan website adalah infrastruktur dan semakin mudahnya akses terhadap internet. Hal inilah yang membuat banyak orang tertarik untuk masuk pada bidang IT atau dalam hal ini internet. Salah satu bidang yang menjadi profesi di bidang internet adalah *Web Programmer*.

Salah satu bahasa pemrograman yang populer untuk pembuatan website atau yang dipakai oleh seorang *Web Programmer* adalah PHP (PHP: *Hypertext Preprocessor*) karena kemudahan dalam mempelajari bahasa pemrograman ini dan bersifat gratis. Ditambah dengan banyaknya dukungan komunitas di internet, sehingga memudahkan bagi para *Web Programmer* untuk menemukan solusi jika mengalami kesulitan dalam mempelajari bahasa pemrograman PHP.

Agar seorang *Web Programmer* tidak membuat dari awal script program PHP, maka disarankan menguasai *framework*. Dengan *framework*, *Web Programmer* tidak harus membuat *script* dari awal. *Framework* sudah menyediakan fungsi atau *library* yang umum digunakan untuk membangun sebuah *website*, seperti fungsi koneksi ke *database*, fungsi pengolahan string, fungsi validasi dan lainnya.

Dari penjelasan di atas, penulis ingin memberikan sumbangsih dalam bentuk pemikiran yang dituangkan dalam bentuk buku ajar ini, dengan judul “Pemrograman Web dengan Menggunakan PHP dan *Framework Codeigniter*”, yang mana diharapkan menjadi

panduan dalam belajar bahasa pemrograman PHP dan Framework Codeigniter.

Penulis menyadari akan banyak sekali kekurangan dari buku ini, baik dari segi penyampaian atau pun isi buku ini, karenanya komentar dan saran sangat penulis nantikan untuk memperbaiki buku ini.

Untuk itu penulis ucapkan terima kasih kepada semua pihak yang telah berpartisipasi dalam penyelesaian buku ini. Khusus kepada penerbit yang telah menerbitkan buku ini.

Bandung, Januari 2016

Penulis,

Supono

Vidiandry Putratama

DAFTAR ISI

PRAKATA	vi
DAFTAR ISI.....	viii
PENDAHULUAN.....	1
 BAB I	
PENGENALAN PHP	3
1.1. Pengertian PHP.....	3
1.2. Sejarah PHP.....	4
1.3. Prinsip Kerja PHP	4
1.4. Kelebihan PHP	5
1.5. Kekurangan PHP.....	6
1.6. Paket Web-Server	6
1.7. Instalasi Web-Server.....	8
1.8. <i>Integrated Development Enviornment (IDE)</i>	12
 BAB II	
SINTAKS DASAR PHP	18
2.1. Sintaks Dasar PHP	18
2.2. Membuat Program Hello World	20
2.3. Tipe data.....	22
2.4. Tipe Floating – Point (Bilangan Rill).....	23
2.5. Tipe Karakter.....	23
2.6. Tipe Data String.....	25
2.7. Tipe Data Array.....	25
2.8. Tipe Object.....	26
2.9. Variabel	27
2.10. Konstanta	30
2.11. Operator	32

BAB III

STATEMEN KONTROL	42
3.1. Statemen Kontrol.....	42
3.2. Struktur Pemilihan	43
3.3. Struktur Perulangan	47

BAB IV

ARRAY DAN AKSES FILE	58
4.1. Array	58
4.2. Array Satu Dimensi dan Multidimensi	69
4.3. Akses File.....	71

BAB V

STRING DAN FUNGSI.....	74
5.1. Fungsi String	74
5.2. Fungsi Tanggal	79
5.3. Ekspresi Reguler.....	80
5.4. Fungsi Pada PHP.....	82
5.5. <i>Include</i> dan <i>Require</i>	84

BAB VI

COOKIES, SESSION DAN CLASS	87
6.1. Session	87
6.2. Cookies	90
6.3. Class	93

BAB VII

PHP DAN MySQL.....	96
7.1. MySQL.....	96
7.2. Basisdata.....	97
7.3. Membuat <i>Database</i> pada PHPMyAdmin.....	98
7.4. Koneksi <i>Database</i> ke MySQL dengan PHP.....	100
7.5. Membuat Form Input Buku Tamu.....	102

7.6.	Menampilkan Data Buku Tamu	104
7.7.	Menghapus Data Buku Tamu.....	105
7.8.	Mengubah Data Buku Tamu	106

BAB VIII

Pengenalan CodeIgniter.....109

8.1.	Pengantar Codeigniter	109
8.2.	Kelebihan Framework Codeigniter	112
8.3.	Instalasi Codeigniter	113
8.4.	Struktur Direktori Codeigniter.....	114
8.5.	Konfigurasi Dasar.....	115
8.6.	Konsep URL pada Codeigniter	118
8.7.	<i>Controller</i>	118
8.8.	<i>View</i>	121

BAB IX

CodeIgniter dengan MySQL..... 125

9.1.	Menentukan Kebutuhan Sistem.....	125
9.2.	Desain <i>Database</i>	127
9.3.	Membuat <i>Database</i>	130
9.4.	Membuat Tabel.....	132
9.5.	Menerapkan Konsep Relasi Antartabel.....	136

BAB X

Proyek Rental Mobil141

10.1.	Persiapan Proyek.....	141
10.2.	Konfigurasi	143
10.3.	Model <i>User (Login)</i>	144
10.4.	<i>Controller Login</i>	147
10.5.	<i>View</i> Halaman <i>Login</i>	148
10.6.	Halaman Utama Aplikasi Rental.....	155
10.7.	Halaman Kelola Data Master (<i>Data Mobil</i>)	157

10.8. Halaman Transaksi Pinjam <i>Mobil</i>	167
10.9. Halaman Transaksi Kembali <i>Mobil</i>	180
10.10. Halaman Cari <i>Data Mobil</i>	186

BAB XI

UPLOAD WEBSITE	192
-----------------------------	------------

11.1. <i>Domain dan Hosting</i>	192
11.2. <i>Membeli Domain dan Hosting</i>	195
11.3. <i>Upload File ke Server</i>	198
11.4. <i>Membuat Database pada Server/Hosting</i>	199

DAFTAR PUSTAKA	203
-----------------------------	------------

BIOGRAFI SINGKAT	205
-------------------------------	------------

PENDAHULUAN

Perkembangan peradaban manusia diiringi dengan perkembangan cara penyampaian informasi (yang selanjutnya dikenal dengan istilah teknologi informasi), mulai dari gambar-gambar yang tak bermakna di dinding-dinding gua, peletakan tonggak sejarah dalam bentuk prasasti, sampai diperkenalkannya dunia arus informasi yang dikenal dengan nama internet dengan media website sebagai media yang menampilkan informasinya.

PHP merupakan bahasa pemrograman untuk membuat website sehingga website menjadi dinamis. Bahasa pemrograman ini paling banyak digunakan untuk membuat website karena mudah untuk dipelajari dan gratis. Dengan semakin banyaknya orang menggunakan bahasa pemrograman ini, maka muncullah yang bernama Framework. Dengan adanya Framework akan memudahkan dalam pembuatan website karena sudah tersedia fungsi-fungsi umum yang biasa digunakan untuk membuat website. Framework yang paling populer dan mudah dipelajari di PHP adalah Codeigniter.

Buku ini dibuat untuk memenuhi kebutuhan siapa saja yang ingin berkecimpung dalam dunia *web developer*, terutama yang ingin mempelajari bahasa pemrograman PHP dan Framework Codeigniter. Dalam buku ini menjelaskan dari mulai sejarah bahasa pemrograman PHP, koneksi dengan *database* MySQL, sampai dengan membuat studi kasus menggunakan (proyek) menggunakan Framework Codeigniter.

Diharapkan setelah membaca buku ini, pembaca akan dapat membuat website dengan menggunakan PHP dan Codeigniter yang terhubung dengan *database* MySQL. Buku ini akan menjembatani mahasiswa untuk memahami tentang bahasa

pemrograman PHP, *database* MySQL, Framework Codeigniter, konsep MVC, dan konsep Object Oriented Programming (OOP).

Adapun bahasan dalam buku ini adalah sebagai berikut:

- BAB I Sejarah, perkembangan, cara kerja bahasa pemrograman PHP, serta berbagai editor (IDE) untuk menulis kode program atau sintak PHP.
- BAB II Membuat program sederhana, variable, tipe data, dan operator.
- BAB III Pernyataan Kontrol yaitu pemilihan, dan perulangan pada PHP
- BAB IV Memahami tentang cara kerja dan penulisan array, serta mengambil data dari file dengan PHP dan menampilkan di website.
- BAB V Manipulasi string, ekspresi regular (ereg), membuat fungsi dan fungsi tanggal.
- BAB VI Session, Cookies dan class.
- BAB VII Koneksi PHP ke *database* MySQL serta manipulasi data menggunakan PHP (tambah, ubah dan hapus)
- BAB VIII Pengenalan Codeigniter, Framework, cara kerja MVC, install dan konfigurasi Codeigniter.
- BAB IX Koneksi *database* MySQL dengan Codeigniter serta perancangan basis data.
- BAB X Implementasi Codeigniter dengan MySQL dengan studi kasus membuat Aplikasi Rental Mobil.
- BAB XI Meng-*upload* website yang sudah dibuat serta bagaimana cara membeli hosting dan domain.

BAB I

PENGENALAN PHP

Setelah mempelajari bab ini pembaca dapat memahami tentang pengertian PHP, sejarah, cara Kerja, kelebihan dan kekurangannya, paket instalasi, dan cara instalasi editor (IDE) untuk menulis kode program PHP, dalam hal ini Sublime.

1.1. Pengertian PHP

Menurut Sibero (2012:49), "PHP (*Personal Home Page*) adalah pemrograman (interpreter) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan".

Menurut Kustiyahningsih (2011:114), "PHP (atau resminya PHP: Hypertext Preprosesor) adalah skrip bersifat Bersifat *server-side* yang ditambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari *Personal Home Page Tools*. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* berarti pengerjaan kode program dilakukan di *server*, baru kemudian hasilnya dikirimkan ke *browser*".

Berdasarkan pengertian di atas, dapat ditarik kesimpulan bahwa PHP (PHP: *Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML.

1.2. Sejarah PHP

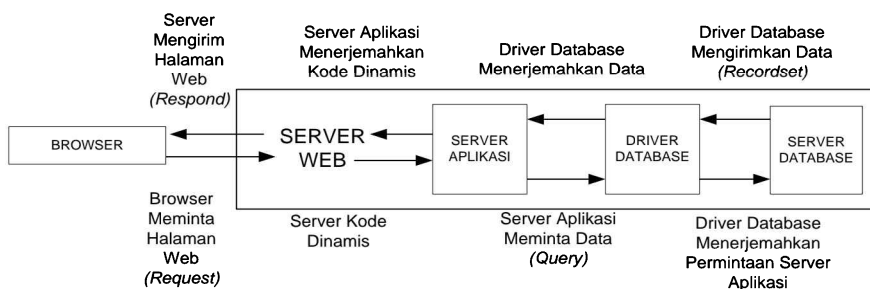
Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

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 dirubah menjadi akronim berulang PHP: *Hypertext Preprocessing*.

PHP difokuskan pada *scripting server-side*, jadi Anda dapat melakukan apa yang bisa dilakukan CGI dengan menggunakan PHP seperti mengambil data inputan form, meng-generate konten halaman dinamis, mengirim dan menerima cookies dan masih banyak lagi. Kemampuan dan *support*-nya untuk *database* juga sangat dapat diandalkan.

1.3. Prinsip Kerja PHP

Bahasa pemrograman PHP merupakan Bahasa pemrograman yang dikategorikan kepada *Server Side Programming*, yang artinya bahasa pemrograman ini memerlukan penerjemah dalam hal ini *web server* untuk menjalankannya. Berikut ini diberikan gambaran tentang cara kerja bahasa pemrograman PHP:



Gambar 1. 1 Prinsip Kerja PHP

Penjelasan Gambar:

1. Client/user mengirimkan file PHP (menggunakan browser) melalui Web Server (Seperti Internet Explorer, Mozilla Firefox, Google Chrome, dll).
2. Web server mendapatkan *request* atau permintaan dari user lalu meneruskan ke Server melalui jaringan internet.
3. Web Server lalu meneruskan permintaan file PHP tersebut ke PHP *processor*. PHP *processor* dapat berupa modul (bagian dari *web-server*) atau terpisah (sebagai CGI/ *Fast - CGI*).
4. Permintaan diproses oleh PHP dan diteruskan ke *database* (jika terdapat permintaan ke *database*), kemudian hasilnya dikirim kembali ke web-server.
5. *Web Server* memaket kembali hasil tersebut dengan menambahkan *HTTP header* dan dikirim kembali ke browser melalui jaringan Internet.
6. Browser memproses HTTP paket dan menampilkannya kembali kepada user sebagai file HTML.

1.4. Kelebihan PHP

Bahasa pemrograman PHP merupakan bahasa pemrograman yang paling banyak digunakan, tentu karena berbagai alasan, salah satunya adalah mempunyai beberapa kelebihan dibandingkan dengan Bahasa pemrograman lainnya yang sejenis. Berikut ini kelebihan bahasa pemrograman PHP:

1. PHP adalah bahasa multiplatform yang artinya dapat berjalan di berbagai mesin dan sistem operasi (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system lainnya.
2. PHP bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.

3. Web Server yang mendukung PHP dapat ditemukan di mana-mana dari mulai apache, IIS, Lighttpd, nginx, hingga Xitami dengan konfigurasi yang relatif mudah dan tidak berbelit-belit, bahkan banyak yang membuat dalam bentuk paket atau package (PHP, MySQL dan Web Server).
4. Dalam sisi pengembangan lebih mudah, karena banyaknya milis–milis, komunitas dan *developer* yang siap membantu dalam pengembangan.
5. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
6. Banyak bertebaran Aplikasi dan Program PHP yang Gratis dan Siap pakai seperti WordPress, PrestaShop, dan lain-lain.
7. Dapat mendukung banyak *database*, seperti MySQL, Oracle, MS-SQL dst.

1.5. Kekurangan PHP

Dari sekian banyaknya kelebihan yang dimiliki oleh bahasa pemrograman PHP, tentu tidak berarti tidak ada kekurangannya. Berikut ini kekurangan bahasa pemrograman web PHP yang mungkin menjadi pertimbangan dalam memilih bahasa pemrograman ini.

1. PHP Tidak mengenal Package
2. Jika tidak di-*encoding*, maka kode PHP dapat dibaca semua orang dan untuk meng-*encoding*-nya dibutuhkan *tool* dari Zend yang mahal sekali biayanya.
3. PHP memiliki kelemahan keamanan. Jadi Programmer harus jeli dan berhati-hati dalam melakukan pemrograman dan Konfigurasi PHP

1.6. Paket Web-Server

Paket *Web-Server* adalah sebuah perangkat lunak *server* yang berfungsi untuk menerima permintaan dalam bentuk situs web

melalui HTTP atau HTTPS dari klien itu, yang dikenal sebagai browser web dan mengirimkan kembali (reaksi) hasil dalam bentuk situs yang biasanya merupakan dokumen HTML (*Oleh Jalan Semut / Wednesday, August 1, 2012 / Posted In __Pengenalan Web And Mobile Computing*). Dari penjelasan di atas, maka dapat ditarik kesimpulan untuk menjalankan file PHP dibutuhkan *web server*.

Berikut ini beberapa *web server* yang banyak kita jumpai dan banyak dipakai oleh orang dalam pengembangan website

1. **XAMPP** terdiri dari Apache web server, MySQL, PHP, Perl, FTP server dan phpMyAdmin. Apache dapat diinstall di berbagai sistem operasi Linux, Solaris, Windows dan Mac OS X.
2. **WampServer** terdiri dari Apache, PHP5 dan MySQL. Sistem operasi yang didukung Windows. WampServer dilengkapi dengan *manager service*, dengan *tray icon* yang memudahkan dalam mengelola server.
3. EasyPHP paket ini terdiri dari Apache server, MySQL, PHPMyAdmin.
4. PHPTriad terdiri Apache, Mysql, PHP dan PHPMyAdmin. Hanya dapat di Windows.
5. FoxServ terdiri dari Apache, mySQL, PHP, PEAR, Zend dengan versi yang terbaru. Sistem Operasi yang dapat diinstal Windows dan Linux. Saat instalasi PHP sebagai modul sedang PEAR dan ZEND sebagai pilihan.
6. PHPDev terdiri dari PHP, Apache, MySQL, PERL, phpMyAdmin. phpdev secara kontinyu memperbaharui versi terbaru dari semua paket yang ada.
7. **AppServ** terdiri dari Apache, MySQL, PHP, dan phpMyAdmin dengan semua komponen yang lengkap. Sistem operasi yang didukung Windows dan Linux.
8. Server2go terdiri dari Apache, PHP dan MySQL. Dapat secara mandiri sebagai server diinstal dalam CD-ROM, USB

Flash-Disk. Besar file 6-45 MB tergantung paket yang ingin digunakan.

9. Apache2Triad terdiri dari Apache2, MySQL, PostgreSQL, OpenSSL, Xmail, SlimFTPd, PHP, Perl dan Python + Apache2TriadCP, PHPmyadmin , PHPPgAdmin , AWStats, UebiMiau, PHPXMail, PHPSFTPd. Semua dalam versi stabil.
10. VertrigoServ terdiri dari PHP development dan server environment untuk Windows secara komplit, juga Apache 2.x.x, PHP 5.x.x, MySQL 5.x.x, dan PhpMyAdmin.
11. Uniform Server terdiri dari paket terakhir dari Apache2, Perl5, PHP5, MySQL5, dan phpMyAdmin. Dapat dijalankan di Windows, tanpa instalasi, cukup di-*unpack* dan jalan.
12. MAMP terdiri dari Apache, PHP and MySQL. Untuk Sistem operasi OSX.
13. TYPO3. Terdiri dari Apache Webserver, PHP dan MySQL. Sistem operasi yang didukung Windows.

1.7. Instalasi Web-Server

Seperti sudah dijelaskan di atas, untuk menjalankan file PHP dibutuhkan sebuah *web-server*. Web-Server yang digunakan pada pembahasan kali ini adalah XAMPP versi 1.8.1, Anda bisa *download* versi yang lebih baru dari yang dibahas buku ini. Berikut adalah langkah – langkah instalasi XAMPP:

1. Download program file master XAMPP di halaman apache friends ini: <https://www.apachefriends.org/download.html>.
2. Jalankan setup.exe XAMPP.
3. Pilih bahasa untuk instalasi yang anda inginkan. Misalnya: English lalu klik OK



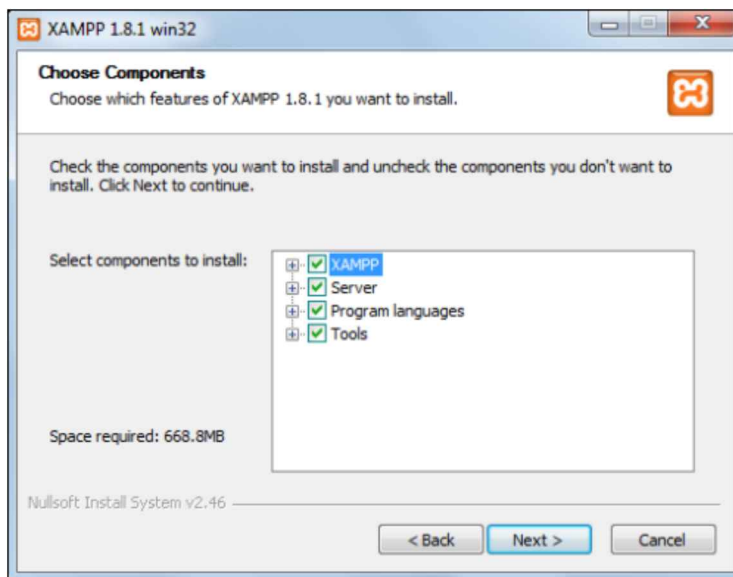
Gambar 1. 2 Pemilihan Bahasa Instalasi

4. Setelah itu akan muncul gambar Welcome Screen XAMPP installer. Lalu klik OK, untuk versi terbaru ada beberapa perubahan proses instalasi, tetapi secara prinsipnya sama saja seperti yang dijelaskan pada buku ini.



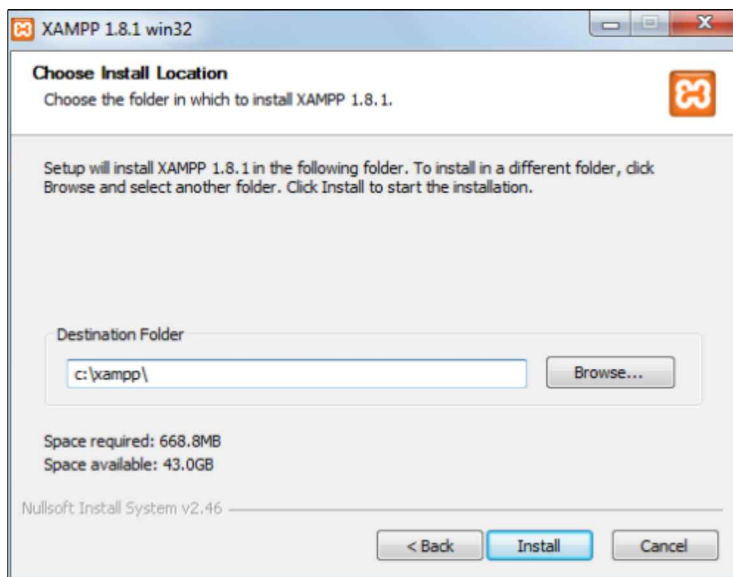
Gambar 1. 3 Welcome Screen Instalasi XAMPP

5. Pilih komponen yang akan di install. Lalu klik Next



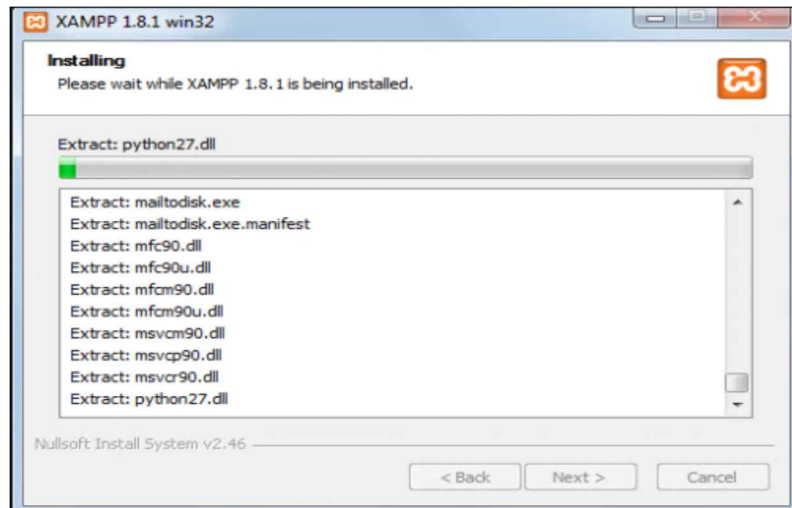
Gambar 1. 4 Komponen Instalasi XAMPP

6. Pilih destination folder (default folder c:\xampp\). Fungsi pemilihan destination folder ini adalah untuk menentukan di mana nanti folder root disimpan. Lalu klik next



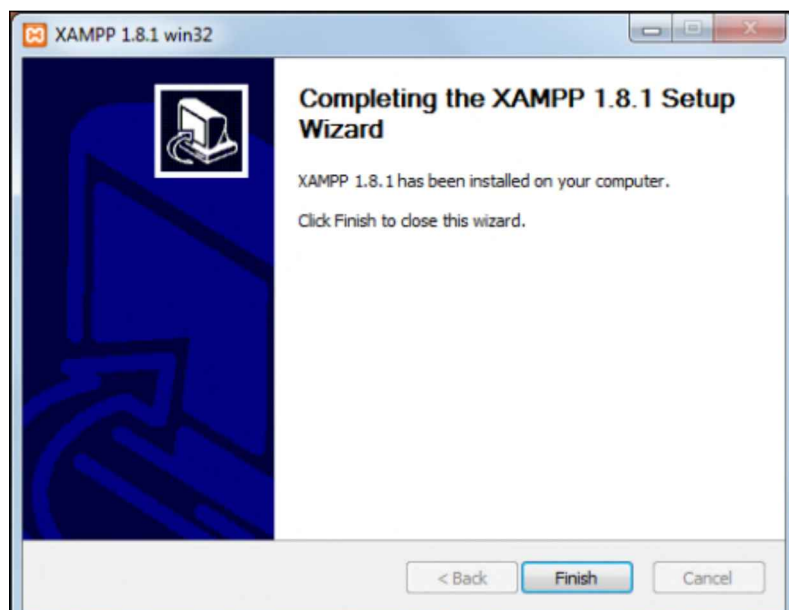
Gambar 1. 5 Destination Foleder Instalasi XAMPP

7. Proses Instalasi XAMPP



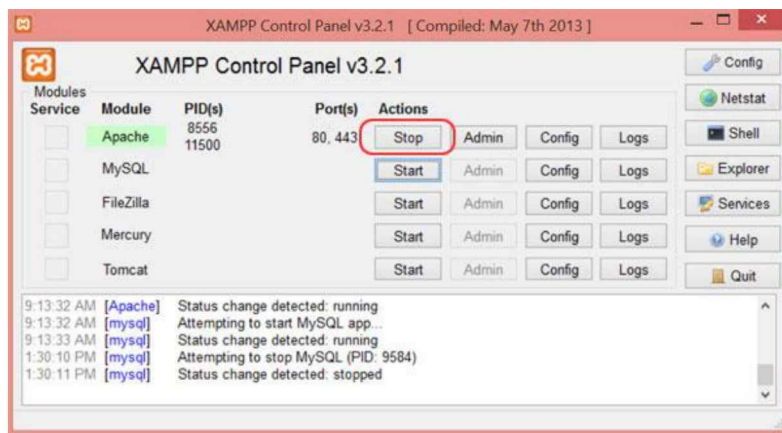
Gambar 1. 6 Proses Instalasi XAMPP

8. Proses Instalasi selesai dilakukan. Lalu Klik Finish



Gambar 1. 7 Pesan Complete Instalitation

9. Untuk menjalankan aplikasi ini, anda bisa langsung mencari ke program aplikasi jika anda menggunakan Windows 8.0 ke atas atau pada desktop Anda. Pada bagian yang dilingkari warna merah Klik tombol START untuk menjalankan web server Apache, sehingga akan tampil PID(s) di sebelah kirinya seperti pada gambar di bawah ini.



Gambar 1. 8 Menjalankan XAMPP

1.8. Integrated Development Environment (IDE)

Integrated Development Environment (IDE) merupakan suatu *software* (perangkat lunak) yang digunakan untuk menulis kode program. IDE ini memiliki fitur-fitur seperti: *code editor*, *compiler*, *debugger*, dan *Graphic User Interface*.

Secara harfiah *Peng-compile-an* program dapat dilakukan pada *console* dan melakukan pembuatan/ pengeditan *syntax* pada notepad, tetapi untuk membangun/ mengembangkan aplikasi-aplikasi besar tanpa bantuan IDE sangatlah sulit. Maka dari itu programmer menggunakan IDE untuk membangun aplikasi-aplikasi besar agar dapat menghasilkan suatu aplikasi secara maksimal.

Dengan bantuan IDE, programmer akan dimudahkan dalam proses penulisan kode program, melakukan debugging, tracking, koreksi terhadap bahasa pemrograman akan lebih mudah, karena beberapa IDE sudah mengimplementasikan *library* untuk mendeteksi bahasa pemrograman tertentu.

IDE yang digunakan untuk bahasa pemrograman berbasis Web sangat banyak. Pada pembahasan saat ini, akan dibahas macam-macam IDE yang sering digunakan diantaranya adalah sebagai berikut:

1. Notepad

Notepad merupakan IDE yang paling sederhana dimiliki dimiliki oleh bahasa system operasi Windows. IDE ini mempunyai kemampuan untuk membuat atau men- *edit* sintaks bahasa pemrograman, dengan kata lain notepad adalah IDE yang mempunyai kemampuan sebagai *text editor*.

2. Notepad ++

Notepad ++ merupakan aplikasi teks editor yang gratis serta *powerful* yang dapat digunakan oleh seorang pengembang aplikasi (*programmer*) untuk menuliskan sebuah kode-kode program. Notepad ++ mendukung banyak bahasa pemrograman, diantaranya: Assembly, C, C++, C#, CSS, HTML, Java, Javascript, Pascal, Perl, PHP, Python, Ruby, Shell, SQL, VB, XML, dan lain sebagainya. Notepad++ ini memiliki banyak kelebihan bila dibandingkan dengan Notepad bawaan Windows yang pertama, seperti memiliki GUI yang baik dan menarik. Selain itu, Notepad++ juga dapat ditambahkan berbagai *plugin* yang bisa semakin mempermudah pekerjaan programmer. Dan juga Notepad++ terdapat versi portabelnya. Dan kelebihan lainnya adalah Notepad++ ini dapat diunduh dan digunakan secara gratis.

3. Dreamweaver

Dreamweaver merupakan perangkat lunak yang digunakan untuk membuat atau meng-*edit* halaman Web. Dreamweaver sangat cocok untuk pengguna pemula atau pengguna yang sedang belajar membuat halaman web. Dreamweaver menyediakan fitur – fitur untuk mempermudah pembuatan halaman web, sehingga tampilan web yang dibuat nanti akan sama dengan tampilan pada saat proses perancangan halaman web. Dreamweaver mendukung beberapa bahasa pemrograman, antara lain: HTML, PHP, CSS, javascript, php, dan sebagainya.

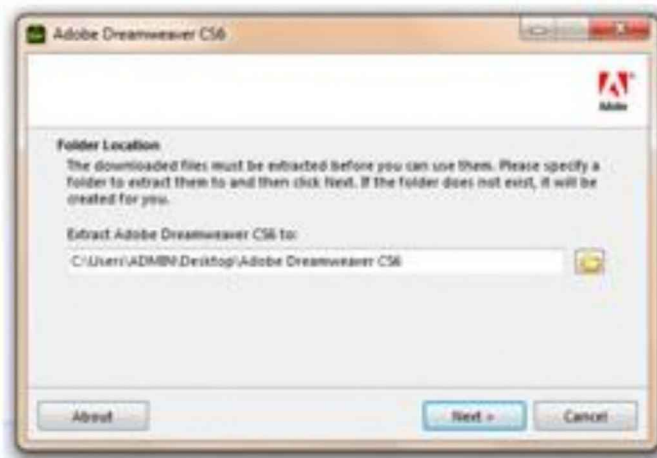
4. Sublime Text

Sublime Text merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau meng-*edit* suatu aplikasi. Sublime Text mempunyai fitur *plugin* tambahan yang memudahkan programmer. Selain itu, Sublime Text juga memiliki desain yang simpel dan keren menjadikan Sublime Text terkesan elegan untuk sebuah *syntax editor*. Selain ringan, IDE ini memiliki kecepatan proses simpan dan buka file. Tidak heran kalau IDE ini paling banyak digunakan terutama di kalangan programmer berbasis web.

Instalasi IDE

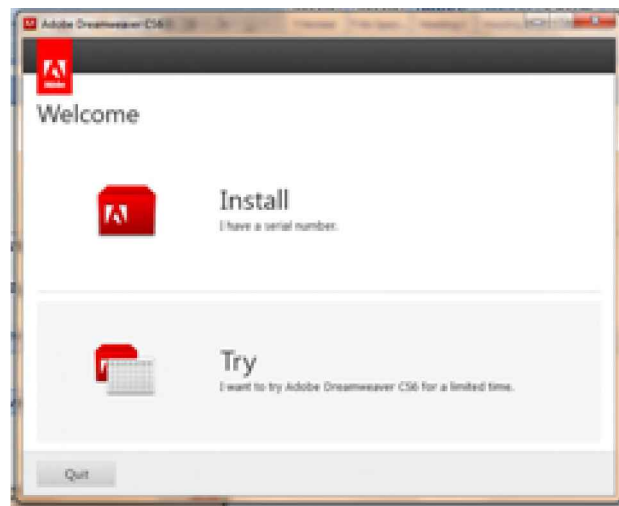
Seperti sudah dijelaskan pada sub bab sebelumnya, terdapat macam-macam IDE yang digunakan untuk membuat suatu aplikasi berbasis Web. Salah satu IDE yang masih banyak digunakan saat ini adalah Dreamweaver. Berikut adalah cara Instalasi Dreamweaver Cs6:

1. Jalankan Aplikasi Dreamweaver CS6, Lalu Pilih Next
2. Tentukan Lokasi File yang anda inginkan, Lalu Next



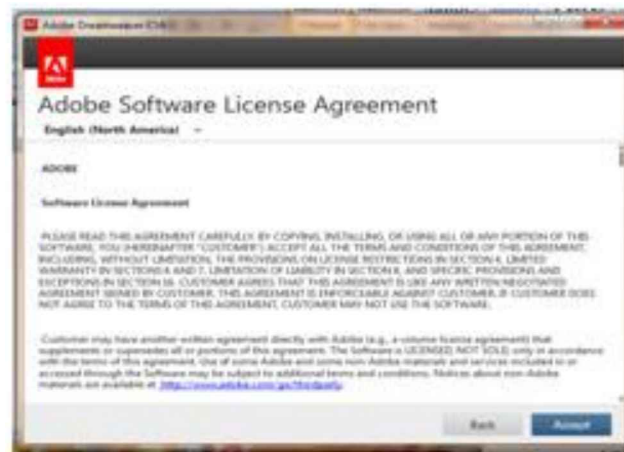
Gambar 1. 9 Menentukan Lokasi Instalasi

3. Pilih Install jika anda memiliki Serial Number, jika tidak cukup memilih pilihan Try untuk mencoba aplikasi selama satu Bulan.



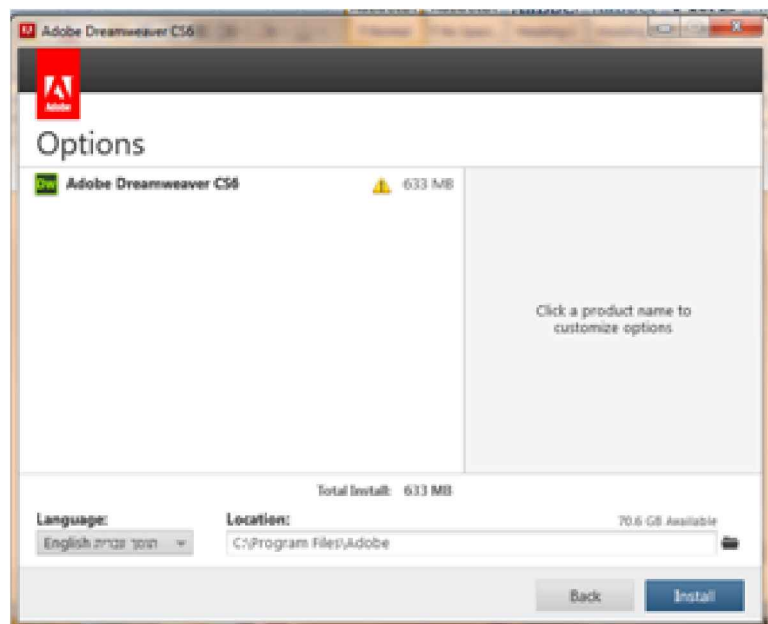
Gambar 1. 10 Instalasi Dreamweaver

4. Pilih Accept untuk menyetujui License Agreement.



Gambar 1. 11 Agreement Instalasi

5. Pilih Install untuk memasang aplikasi Dreamweaver CS6



Gambar 1. 12 Proses Instalasi Selesai

6. Selesai.

Dengan banyaknya pilihan IDE, memungkinkan programmer dapat menilai mana yang cocok sesuai dengan kebutuhan dan perangkat yang digunakan.

Rangkuman/Kesimpulan

Bahasa pemrograman PHP adalah bahasa pemrograman yang diperuntukkan untuk pembuatan web, di mana dikategorikan bahasa berbasis server side karena dibutuhkan web server untuk menjalankannya. Beberapa vendor sudah membuat package installer di mana sudah mempaketkan instalasi PHP-MySQL-Apache dalam sebuah paket seperti XAMPP. Untuk menulis bahasa kode program PHP ini dibutuhkan editor atau IDE untuk memudahkan dalam proses debugging.

Latihan

1. Install IDE Sublime dan analisis cara kerjanya kemudian bandingkan dengan IDE Dreamweaver.
2. Jelaskan pengertian PHP menurut pemahaman Anda !
3. Gambarkan dan Jelaskan Prinsip kerja aplikasi perpustakaan yang dibangun dengan menggunakan bahasa PHP !

BAB II

SINTAKS DASAR PHP

Setelah mempelajari bab ini, pembaca diharapkan dapat menjelaskan dan membuat sintaks dasar di dalam pemrograman PHP, folder tempat menyimpan file PHP, sebagai dasar mempelajari materi-materi selanjutnya. Pembaca dapat menjelaskan dan membuat contoh program menggunakan variabel, tipe data, komentar, dan operator.

2.1. Sintaks Dasar PHP

Sama seperti bahasa pemrograman yang lain, bahasa pemrograman PHP memiliki aturan dasar cara penulisannya. Berikut ini beberapa cara penulisan sintak PHP yang bisa dipakai dalam pembuatan penulisan bahasa pemrograman PHP:

Cara I

`<?php` (menandai awal tag)

Isi script PHP

`?>` (menandai akhir tag)

Cara II

`<?` (menandai awal tag)

Isi script PHP

`?>` (menandai akhir tag)

Cara III

`<%` (menandai awal tag)

Isi script PHP

`%>` (menandai akhir tag)

Cara ini sama dengan tag pada ASP. Opsi ini bisa dilakukan jika nilai `asp_tags` pada `php.ini` bernilai on.

Cara IV

`<script language="php">` menandai awal script

Isi script PHP

`</script>` menandai akhir script

Untuk keseragaman pemakaian kode program PHP, maka dalam buku ini akan menggunakan Cara I, yaitu dengan mempunyai tag awal `<?php` dan diakhiri dengan tag `?>`. Dokumen atau file PHP harus disimpan dengan ekstensi `.php` dan disimpan pada folder root server, dalam hal ini di folder `htdocs` pada XAMPP (`c:/xampp/htdocs`).

Berikut ini contoh dasar penulisan kode PHP:

```
<?php
    echo "Selamat datang di Program PHP";
?>
```

Sintaks di atas adalah sintaks untuk memunculkan tulisan "Selamat datang di Program PHP" pada layar dalam hal ini browser. Fungsi dari sintaks `echo` adalah untuk menampilkan String yang nantinya akan diterjemahkan oleh web server dan ditampilkan pada web browser.

Selain itu, kode program PHP dapat dikombinasikan atau digabungkan dengan kode HTML, ekstensi filenya dapat disimpan dengan menggunakan `.php`

Berikut contoh program PHP yang digabungkan dengan program HTML:

```
<!DOCTYPE html>
<head>
<title>Belajar cara memasukkan kode HTML +
PHP</title>
</head>
<body>
```

```

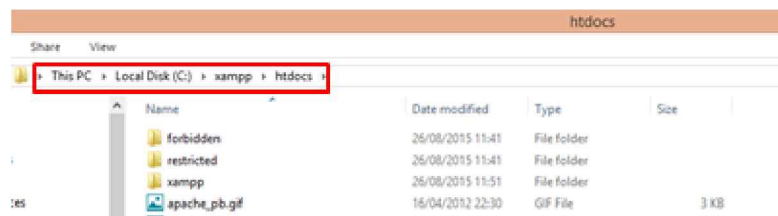
    <p>Kalimat ini dibuat menggunakan HTML saja
</p>
    <?php
        echo "<p>Kalimat ini dibuat menggunakan mode
PHP style</p>";
    ?>
</body>
</html>

```

2.2. Membuat Program Hello World

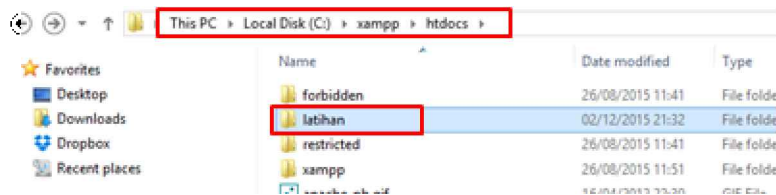
Pada bagian ini kita akan membuat program sederhana untuk menampilkan Kalimat Hello World. Seperti yang sudah dijelaskan pada bab sebelumnya bahwa kode php dapat dibuat atau ditulis dengan menggunakan aplikasi Text Editor apa saja seperti Notepad, Notepad++, Sublime atau *software* PHP Editor. Ada beberapa hal yang perlu diketahui untuk membuat program, yaitu:

1. Direktori tempat (tempat menyimpan program/script) ada di c:/xampp/htdocs (tergantung pada saat instalasi).



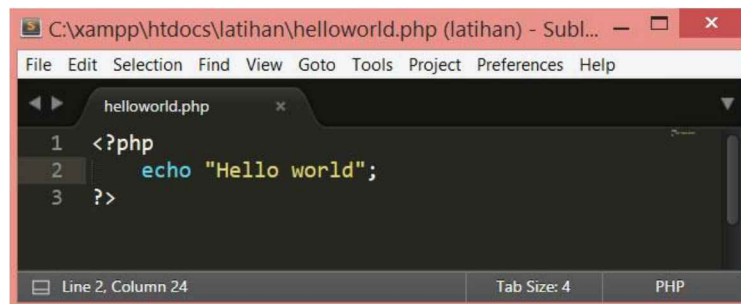
Gambar 2. 1 Direktori Penyimpanan Script

2. Untuk memulai membuat kode program PHP, sebaiknya membuat folder terlebih dahulu agar memudahkan dalam pengelolaan file. Buat folder dan diberi nama "latihan" di c:/xampp/htdocs. Untuk latihan – latihan selanjutnya kita akan menggunakan folder ini untuk menyimpan file – file.



Gambar 2. 2 Membuat Folder Latihan

- Setelah membuat folder dengan nama “latihan” langkah selanjutnya adalah menuliskan sintaks/kode program php. Penulisan kode program php dapat dilakukan di Notepad, Sublime atau aplikasi Dreamweaver. Penulis menggunakan Sublime dalam pembuatan kode program PHP, buka program aplikasi sublime kemudian ketik kode program di atas seperti pada gambar di bawah ini:



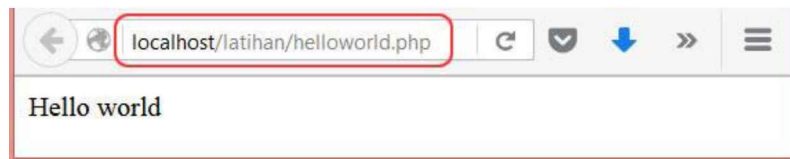
Gambar 2. 3 Menulis Sintak dengan IDE Sublime

Simpan file tersebut di c:\xampp\htdocs\latihan dengan nama helloworld.php ingat harus menggunakan ekstensi.php hasilnya seperti pada gambar di bawah ini:



Gambar 2. 4 Menyimpan File Kode Program PHP

4. Buka web Browser lalu ketikkan localhost/latihan/helloworld.php (pastikan web server anda telah berjalan, apache dijalankan) pada alamat URL seperti pada gambar di bawah ini:



Gambar 2. 5 Menjalankan Sintak PHP

Cara kerja menjalankan kode program PHP adalah sebagai berikut: Jika Anda menyimpan file php di `c:\xampp\htdocs\latihan\kode_program.php` maka untuk menjalankan kode program tersebut, dengan cara buka browser kemudian ketik `localhost/latihan/kode_program.php` atau dengan kata sederhana kita mengubah `c:\xampp\htdocs` menjadi `localhost`.

2.3. Tipe data

Dalam PHP, tipe data dapat dikelompokkan menjadi beberapa bagian, yaitu:

- *Integer* (bilangan bulat)
- *Floating-point* (bilangan real)
- Karakter
- *String*
- *Boolean*
- *Array*
- Objek

Tipe data Integer (Bilangan Bulat)

Tipe data integer adalah tipe data yang mempresentasikan bilangan bulat seperti 3,6,3000,4596 dan sebagainya. Rentang nilai

yang diperbolehkan pada tipe data integer adalah dari -2,147,483,648 sampai dengan +2,147,483,647. Nilai dari tipe data *integer* dapat ditulis dalam bentuk decimal, octal, maupun heksadesimal.

Contoh bilangan bulat yang ditulis dalam bentuk decimal:

```
3000
-578
+98
```

Contoh bilangan bulat yang ditulis dalam bentuk oktal:

```
0755      # desimal = 493
+010      # desimal = 8
```

Contoh bilangan bulat yang ditulis dalam bentuk Heksadisimal:

```
0xFF      # desimal = 255
0x10      # desimal = 10
```

2.4. Tipe Floating – Point (Bilangan Rill)

Tipe *floating-point* adalah tipe data yang merepresentasikan nilai numeric dalam bentuk pecahan atau mengandung angka decimal belakangan koma. Nilai yang diperbolehkan pada tipe *Floating-point* adalah dari 1.7E-308 sampai dengan 1.7E+308 dengan 15 digit akurasi.

Bilangan *floating-point* dapat ditulis dalam notasi sains (*scientific*) dan notasi normal.

Contoh bilangan *floating-point*:

```
3.14
0.017
0.314E1
17.0E-3
```

2.5. Tipe Karakter

Tipe data karakter digunakan untuk merepresentasikan data dengan nilai karakter tunggal, Contoh: 'A','I','U','E','O' dan

sebagainya. Penulisan karakter dalam bahasa pemrograman PHP diapit dengan menggunakan tanda petik tunggal.

Contoh:

```
$karakterku = 'R';
```

```
$angkaku = '8';
```

Bahasa PHP juga mendukung *escape sequence*. *Escape Sequence* merupakan karakter yang diawali dengan karakter *backslash* (\). Meskipun terdiri dari dua karakter atau lebih, *escape sequence* akan dianggap karakter tunggal. *Escape Sequence* memiliki aturan yang khusus dan berbeda-beda, seperti dapat dilihat di tabel di bawah ini:

Tabel 2. 1 Escape Sequence

Escape Sequence	Definisi
\0	Karakter nol atau karakter ber-ASCII nol
\a	Untuk menyembunyikan sound beep
\b	Untuk backspace
\f	Untuk mengganti halaman (formfeed)
\n	Untuk membuat baris baru (newline)
\r	Untuk kembali keawal baris atau carriage return
\t	Untuk tab horizontal, fungsi yang sama ketika melakukan pengetikan di <i>Ms.Word</i> , lalu menekan tombol tab di <i>keyboard</i>
\v	Untuk membuat tab vertikal
\\	Untuk membuat karakter \
\?	Untuk membuat karakter ?
\'	Untuk membuat karakter '
\"	Untuk membuat karakter "
\ooo	Untuk karakter yang nilai oktalnya sebanyak tiga digit oktal
\xhh	Untuk karakter yang heksadesimalnya sebanyak dua digit heksadesimal

2.6. Tipe Data String

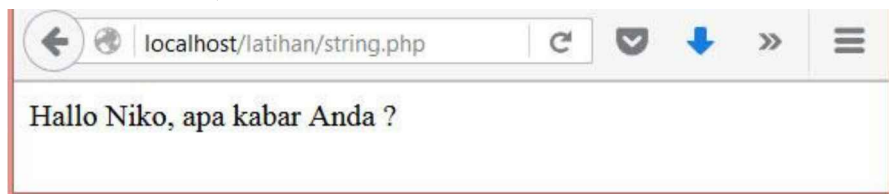
Tipe data *String* merepresentasikan data yang berupa teks (kumpulan karakter). Bahasa PHP mendukung dua tipe *String* yaitu yang diapit oleh petik ganda dan tunggal.

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

Contoh:

```
<?php  
$nama = "Niko";  
echo "Hallo $nama, apa kabar Anda ?";  
?>
```

Saat di eksekusi, kode di atas akan memberikan hasil:



Gambar 2. 6 Menampilkan String

String yang dibuat dengan petik tunggal tidak mendukung proses interpolasi variabel.

2.7. Tipe Data Array

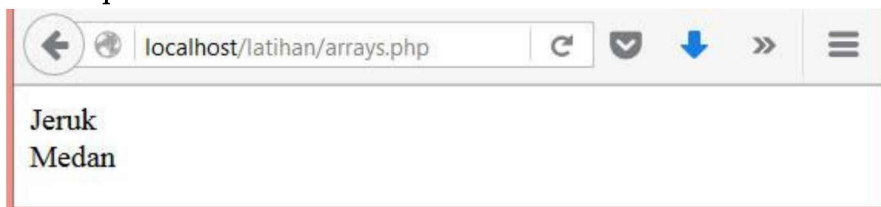
Tipe array merupakan suatu variabel yang menyimpan sekelompok nilai yang memiliki sifat yang sama, yang dapat diidentifikasi atau diakses berdasarkan posisi atau dengan nama yang telah didefinisikan sebelumnya.

Contoh:

```
<?php
//membuat variabel $buah bertipe array, index/key
ditentukan
$buah[2] = "Salak";
$buah[3] = "Jeruk";
$buah[5] = "Durian";
echo $buah[3];
echo "<br>";

//membuat array dengan nama
$alamat['budi'] = "Medan";
$alamat['maudy'] = "Bandung";
$alamat['silalahi'] = "Semarang";
echo $alamat['budi'];
?>
```

Jika di eksekusi, kode program di atas akan memberikan hasil seperti di bawah ini:



Gambar 2. 7 Contoh Tipe Data Array

Untuk tipe data array ini akan dijelaskan pada bab terpisah pada bab berikutnya.

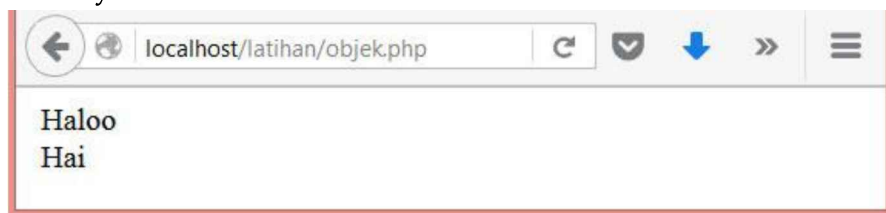
2.8. Tipe Object

Tipe data object dibuat dengan tujuan agar para programmer terbiasa dengan OOP. Tipe data object bisa berupa bilangan, variabel atau fungsi. Data ini dapat dimasukkan ke dalam kode program sehingga meringkas beberapa fungsi dan memperkecil ukuran file.

Agar lebih mudah dapat menggunakan variabel dengan nilai string:

```
<?php
class Nama {
var $jenis = "Haloo";
    function ganti_nama($jenis) {
        $this->jenis = $jenis;
    }
}
$nama = new Nama;
echo $nama->jenis;
$nama->ganti_nama("Hai");
echo "<br>".$nama->jenis;
?>
```

Hasilnya:



Gambar 2. 8 Contoh Tipe Data Object

2.9. Variabel

Variabel adalah kode program yang digunakan untuk menampung nilai tertentu. Nilai yang disimpan di dalam variabel selanjutnya dapat dipindahkan ke dalam *database* atau ditampilkan ke pengguna atau diolah lagi. Dalam PHP, penulisan variabel diawali dengan karakter '\$' (karakter dolar), kemudian diikuti dengan huruf sebagai karakter pertama. Setelah itu, dapat dilanjutkan dengan kombinasi huruf dan angka. Variabel tidak boleh mengandung spasi maupun tanda baca di dalamnya, kecuali underscore ('_'). Variabel pada PHP bersifat "case sensitif", yang

berarti Anda harus memperhatikan penulisan huruf besar dan huruf kecil.

Tabel 2. 2 Penulisan Variabel

Nama Variable Yang Benar	Nama Variable Yang Salah
\$nama_pemakai	\$nama pemakai
\$kota_3	\$3kota
\$user1	\$us\er1

Berikut contoh Pendefinisian Variabel:

```
<?php
$a = true;
$nama = "Andi";
$salamat = "Jl. Sari Asih No 54";
$nilai1 = 4;
$nilai2 = 3.14;
?>
```

Dalam penggunaannya, variabel di php sedikit berbeda dengan kode program pada umumnya, karena pada bahasa pemrograman php suatu variabel tidak perlu dideklarasikan terlebih dahulu. Selain itu, variabel php dapat digunakan untuk menampung nilai yang berasal dari tipe data yang berbeda. Pada contoh di atas, variabel \$a dapat menampung nilai yang berbeda – beda yaitu: Boolean, string, integer dan float.

Variabel Lokal dan Global

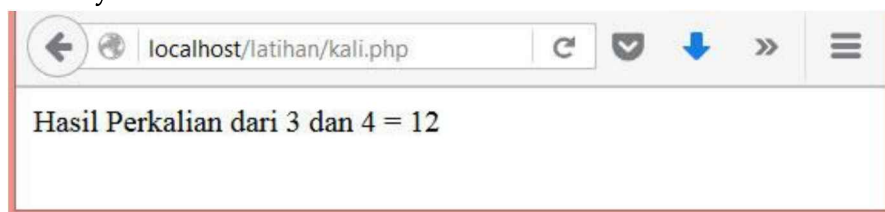
Menurut ruang lingkupnya, variabel di dalam PHP dibedakan menjadi dua jenis yaitu global dan local. Variabel lokal adalah variabel yang hanya dikenal di dalam suatu fungsi tertentu. Variabel lokal tidak akan dikenal oleh fungsi lain maupun oleh kode di luar fungsi.

```

<?php
function perkalian() {
$nilai1 = 3;
$nilai2 = 4;
$hasil = $nilai1 * $nilai2;
echo "Hasil Perkalian dari $nilai1 dan $nilai2 =
$hasil";
}
perkalian();
?>

```

Hasilnya:



Gambar 2. 9 Contoh Variabel Lokal

Pada contoh kode di atas, variabel `$nilai1` dan `$nilai2` adalah variabel yang bersifat lokal dan hanya dikenal di fungsi tersebut (`function perkalian()`). Pengaksesan variabel tersebut dari luar fungsi akan menyebabkan kesalahan pada saat eksekusi.

Berbeda dengan variabel yang bersifat lokal, variabel yang bersifat global akan dikenal oleh seluruh bagian program. Jika suatu fungsi akan menggunakan variabel yang bersifat global, maka perlu ditambahkan kode "global" di depan variabel tersebut. Contoh Penerapan Variabel Global:

```

<?php
$panjang_balok = 10;
$lebar_balok = 20;
$tinggi_balok = 15;
function luas_balok() {
    global $panjang_balok, $lebar_balok;

```

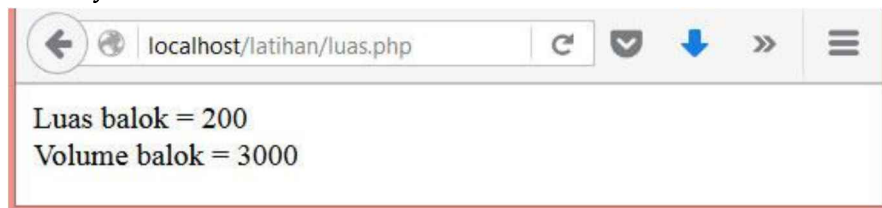
```

    $luas = $panjang_balok * $lebar_balok;
    echo "Luas balok = " . $luas;
}

function volume_balok(){
    global $panjang_balok, $lebar_balok,
    $tinggi_balok;
    $volum = $panjang_balok * $lebar_balok *
    $tinggi_balok;
    echo "Volume balok = " . $volum;
}
echo luas_balok()."<br><br>";
echo volume_balok();
?>

```

Hasilnya:



Gambar 2. 10 Contoh Variabel Global

2.10. Konstanta

Selain menggunakan variabel untuk menampung nilai data, dalam konsep bahasa pemrograman dikenal juga istilah konstanta. Konstanta adalah suatu lokasi penyimpanan yang berisikan nilai tetap yang tidak dapat diubah sepanjang program berjalan. Di dalam kode PHP, konstanta didefinisikan menggunakan fungsi `const` atau `define()`.

Aturan Penulisan Konstanta

Aturan-aturan untuk penulisan konstanta adalah sebagai berikut:

1. Penulisan konstanta dengan keyword `const` hanya dapat digunakan pada lingkungan global PHP. Sehingga kita tidak bisa menggunakannya di dalam function, loop, atau if. Apabila membutuhkan konstanta di dalam maka function harus menggunakan `define`.
2. Mendefinisikan konstanta menggunakan `const` caranya hamper sama dengan membuat variable apabila di dalam variable kita menggunakan tanda dolar (\$) maka dalam konstanta menggunakan `const` atau `define`. Seperti dapat dilihat pada contoh di bawah ini:

Contoh Penulisan Konstanta menggunakan `const`:

```
<?php
function luas_balok(){
define ('panjang_balok', 10);
define ("lebar_balok" , 20);
define ("tinggi_balok" , 15);
    $luas = panjang_balok * lebar_balok;
    echo "Luas balok = " . $luas;
}
luas_balok();
?>
```

Contoh Penulisan Konstanta menggunakan `define`:

```
<?php
const panjang_balok = 15;
const lebar_balok = 23;
const tinggi_balok = 15;
function luas_balok(){
    $luas = panjang_balok * lebar_balok;
    echo "Luas balok = " . $luas;
}

luas_balok();
?>
```

Pada contoh percobaan di atas, dapat dilihat bahwa penulisan/pendeklarasian konstanta menggunakan `const` berada diluar fungsi `luas_balok()` atau bersifat global. Sedangkan pendeklarasian dengan menggunakan `define` dapat di deklarasikan secara global maupun secara lokal.

2.11. Operator

Dalam bahasa pemrograman secara umum, operator digunakan untuk memanipulasi atau melakukan proses perhitungan pada suatu nilai variabel. Operator di dalam bahasa PHP dapat dikategorikan menjadi beberapa kelompok, yaitu:

- a. Operator penugasan
- b. Operator aritmatika
- c. Operator relasional
- d. Operator eksekusi
- e. Operator logika
- f. Operator array
- g. Operator string
- h. Operator bitwise
- i. Operator tipe
- j. Operator kondisional.

Operator Penugasan

Assignment Operator atau biasa disebut Operator penugasan merupakan operator yang digunakan untuk memberi nilai ke suatu variable ke variable lain. Simbol operator ini adalah `(=)`. Berikut ini adalah daftar table dari assignment operator.

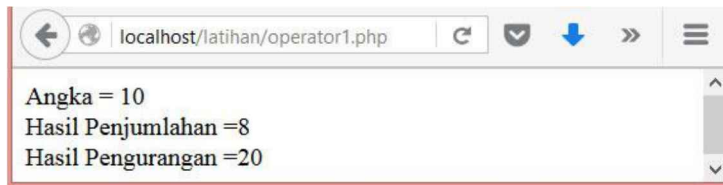
Tabel 2. 3 Operator Penugasan

Operator	Operasi / Fungsi	Contoh
+=	Menambahkan nilai di kiri dengan nilai di kanan	x += 2; x=x+2;
-=	Mengurangi nilai di kiri dengan nilai di kanan	x -= 2; x = x-2;
.=	Melakukan concatenation / operasi gabungan antara nilai di kiri dengan nilai di kanan	x .= "Budi"; x=x. "Budi";
/=	Membagi nilai di kiri dengan nilai di kanan	x /= 2; x =x/2;
%=	Sisa hasil bagi antara nilai di kiri dengan nilai di kanan	x %=2; x = x%2;
!=	Melakukan operasi OR antara nilai di kiri dengan nilai di kanan	x != 2; x = x!2;
^=	Melakukan operasi XOR antara nilai dikiri dengan nilai di kanan	x ^= 2; x = x^2;

Contoh program dengan menggunakan Operator Penugasan:

```
<?php
$angka = 10;
$x = 5;
$y=30;
echo "Angka = $angka <br>";
$x+=3;
echo "Hasil Penjumlahan =$x <br>";
$y-=10;
echo "Hasil Pengurangan =$y";
?>
```

Hasil:



Gambar 2. 11 Contoh Operator Penugasan

Pada program di atas, kita mempunyai variabel \$angka, \$x, \$y. Masing – Masing variabel tersebut telah diisi angka. Untuk variabel \$angka langsung ditampilkan ke layar, sedangkan variabel \$x dan \$y di olah terlebih dahulu sebelum ditampilkan. \$x+=3 artinya \$x = \$x+3 begitu juga dengan \$y+=10 artinya \$y = \$y – 10.

Operator Aritmatika

Operator Aritmatika adalah operator matematis yang terdiri dari operator penambahan, pengurangan, perkalian, pembagian, modulus, plus, dan minus. Berikut operator aritmatika di dalam bahas pemrograman php:

Tabel 2. 4 Contoh Operator Penugasan

Operator	Operasi	Contoh Ekpresi	Hasil
+	Perjumlahan	X=y+5	X=15
-	Pengurangan	X=y-5	X=5
*	Perkalian	X=y*5	X=50
/	Pembagian	X=/5	X=2
%	Sisa Pembagi	X=y%3	X=1
++	Increment	X=y++	X=11
-	Decrement	X=y-	X=9

Kebanyakan operator aritmatika dalam PHP bertipe binary yakni membutuhkan 2 operand, kecuali operator minus (-) dan plus

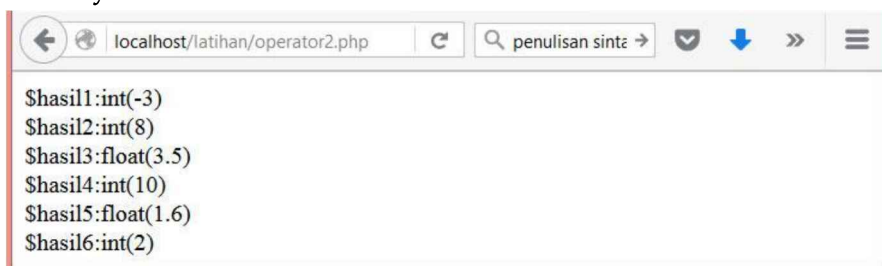
(+) yang merupakan operator tipe unary (hanya membutuhkan 1 operand).

Contoh:

```
<?php
$hasil1= -3;
$hasil2=3+5;
$hasil3=8-4.5;
$hasil4=2*5;
$hasil5=3+8/5-3;
$hasil6=10 % 4;

echo "\$hasil1: "; var_dump($hasil1); //
$hasil1:int(-3)
echo "<br \>";
echo "\$hasil2: "; var_dump($hasil2); //
$hasil2:int(8)
echo "<br \>";
echo "\$hasil3: "; var_dump($hasil3); //
$hasil3:float(3.5)
echo "<br \>";
echo "\$hasil4: "; var_dump($hasil4); //
$hasil4:int(10)
echo "<br \>";
echo "\$hasil5: "; var_dump($hasil5); //
$hasil5:float(1.6)
echo "<br \>";
echo "\$hasil6: "; var_dump($hasil6); //
$hasil6:int(2)
?>
```

Hasilnya:



```
$hasil1:int(-3)
$hasil2:int(8)
$hasil3:float(3.5)
$hasil4:int(10)
$hasil5:float(1.6)
$hasil6:int(2)
```

Gambar 2. 12 Contoh Operator Aritmatika

Pada kode program di atas, menggunakan fungsi `var_dump()` untuk menampilkan hasil perhitungan, sehingga kita bisa melihat tipe data dari masing-masing variabel.

Dari hasil `var_dump()`, terlihat bahwa variabel `$hasil3` dan `$hasil5` bertipe float. Hal ini dikarenakan perhitungan aritmatika pada baris ke-4 dan ke-6 menghasilkan angka desimal, sehingga secara otomatis variabel tersebut tidak dapat ditampung sebagai integer, melainkan harus float. Namun jika hasil operasi matematis tersebut menghasilkan bilangan bulat, PHP akan menyimpannya sebagai tipe data int (integer), seperti variabel `$hasil1`, `$hasil2`, `$hasil4` dan `$hasil6`.

Pada perhitungan baris ke-6 yaitu persamaan `$hasil5=3+8/5-3`, hasilnya adalah 1.6. Hal ini karena operator pembagian memiliki prioritas lebih tinggi daripada operator tambah dan kurang. Operasi `3+8/5-3` dikerjakan oleh PHP sebagai `(3+(8/5))-3`. Namun untuk hal ini, disarankan menggunakan tanda kurung secara tertulis agar memudahkan dalam membaca alur program, dari pada bergantung kepada aturan prioritas operator PHP.

Operator Relasional

Operator Relational digunakan untuk membandingkan nilai dari dua operand. Hasil perbandingan dinyatakan dalam nilai

boolean. TRUE berarti benar, dan FALSE berarti salah. Beberapa jenis relational operator:

`==` : memeriksa apakah operand kanan bernilai sama dengan operand kiri

`>` : memeriksa apakah operand kiri bernilai lebih besar daripada operand kanan

`<` : memeriksa apakah operand kiri bernilai lebih kecil dengan operand kanan

`>=` : memeriksa apakah operand kiri bernilai lebih besar atau sama dengan operand kanan

`<=` : memeriksa apakah operand kiri bernilai lebih kecil atau sama dengan operand kanan

`!=` : memeriksa apakah operand kanan tidak bernilai sama dengan operand kiri

Operator Logika

Untuk menyatakan suatu syarat, kita dapat menggunakan operator logika. Ketika Anda menempuh sekolah SMA kelas 1, tentu Anda pernah belajar tentang logika dalam matematika. Masih ingatkah Anda dengan operator DAN, ATAU dan juga NEGASI, serta tabel kebenaran BENAR-SALAH? Ya... jangan dikira materi tersebut tidak ada gunanya. Materi tersebut akan dipakai dalam konsep pemrograman.

Dalam pemrograman PHP, terdapat beberapa operator logika yang bisa digunakan:

Tabel 2. 5 Operator Logika

Operatot	Makna
<code>&&</code>	AND
<code> </code>	OR
<code>!</code>	Negasi

Sedangkan berikut ini tabel kebenaran dari masing-masing operator:

Logika AND

Tabel 2. 6 Logika AND

Operand 1	Operand 2	Hasil
Benar	Benar	Benar
Benar	Salah	Salah
Salah	Benar	Salah
Salah	Salah	Salah

Logika OR

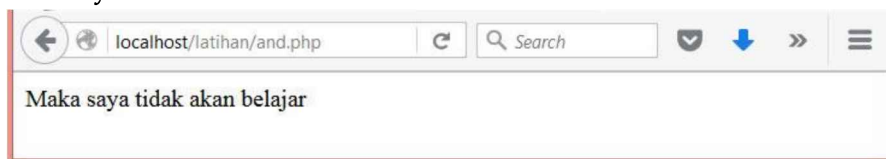
Tabel 2. 7 Logika OR

Operand 1	Operand 2	Hasil
Benar	Benar	Benar
Benar	Salah	Benar
Salah	Benar	Benar
Salah	Salah	Salah

Contoh penggunaan AND:

```
<?php
$nilai = "Makan";
if(($nilai == "Makan") && ($nilai == "Bermain"))
{
echo " Maka saya akan belajar";
}
else
{
echo " Maka saya tidak akan belajar";
}
?>
```

Hasilnya



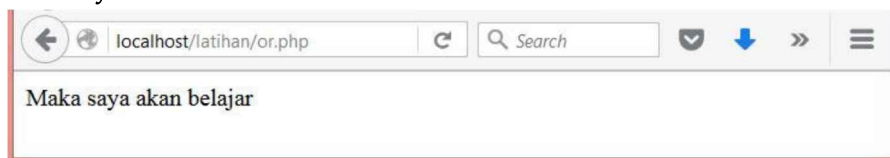
Gambar 2. 13 Contoh Operator Aritmatika menggunakan AND

```

<?php
$nilai = "Makan";
if(($nilai == "Makan") || ($nilai == "Bermain"))
{
echo " Maka saya akan belajar";
}
else
{
echo " Maka saya tidak akan belajar";
}
?>

```

Hasilnya:



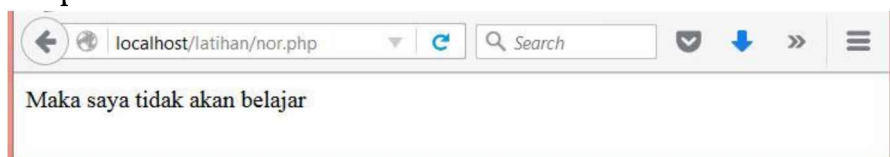
Gambar 2. 14 Contoh Operator Aritmatika menggunakan OR

```

//Menggunakan NOT
<?php
$nilai = "Makan";
if($nilai != "Makan")
{
echo " Maka saya akan belajar";
}
else
{
echo " Maka saya tidak akan belajar";
}
?>

```

Output:



Gambar 2. 15 Contoh Operator Aritmatika menggunakan NOT

Operator String

Dalam PHP, hanya terdapat 1 jenis operator String, yakni operasi penyambungan (concatenation) string. Operator ini menggunakan karakter titik (.).

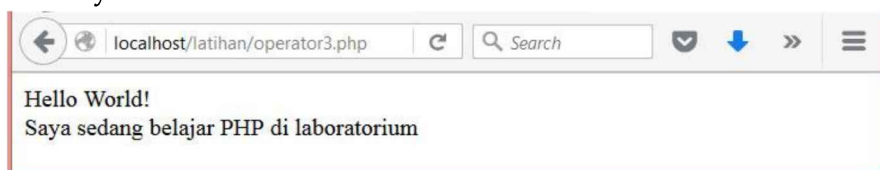
Operator penyambungan string ini membutuhkan 2 inputan yang bertipe data string. Hasil dari operator ini adalah sebuah string yang terdiri dari sambungan kedua string tersebut.

Berikut contoh kode program cara penggunaan operator string dalam PHP:

```
<?php
$a = "Hello ";
$hasil = $a . "World!";
echo $hasil;
echo "<br />";

$a = "belajar ";
$b = "PHP ";
$c = "di laboratorium";
$hasil= "Saya sedang ".$a.$b.$c;
echo $hasil;
?>
```

Hasilnya:



Gambar 2. 16 Contoh Operator String

Rangkuman

PHP adalah bahasa pemrograman yang mempunyai standar penulisan tersendiri, di mana mempunyai cara penulisan yang berbeda dengan yang lainnya. Menyimpan pada folder root (web server) yang sudah diinstall sebelumnya dan menggunakan

ekstensi .php. PHP sama seperti bahasa pemrograman yang lain mempunyai tipe data, operator, komentar, variable dan lainnya.

Latihan

1. Buatlah program biodata menggunakan bahasa pemrograman PHP !
2. Buatlah program hitung Luas dan Keliling Persegi menggunakan bahasa pemrograman PHP !

BAB III

STATEMEN KONTROL

Setelah mempelajari bab ini, pembaca diharapkan dapat menjelaskan pengertian dan jenis-jenis Statemen Kontrol, membuat program menggunakan Struktur Pemilihan (*if* dan *switch*), menjelaskan dan membuat program menggunakan Struktur Perulangan (*for*, *while*, *do-while*, *foreach*) dan dapat menjelaskan dan membuat program menggunakan statement Loncat (*break*, *continue*, *goto*, *return* dan fungsi *exit*).

3.1. Statemen Kontrol

Statemen kontrol adalah statemen-statemen yang digunakan untuk mengontrol atau mengendalikan jalannya eksekusi kode program. Statemen kontrol dapat dibedakan menjadi empat jenis, yaitu:

- Statemen untuk mengontrol pemilihan
- Statemen untuk mengontrol pengulangan
- Statemen untuk memindahkan eksekusi dari bagian kode tertentu ke bagian lain
- Statemen untuk menangani terjadinya eksepsi(kesalahan program)

Pernyataan kontrol pada PHP sendiri bertujuan untuk pengambilan beberapa keputusan dan pengulangan perintah dalam suatu struktur program, maka dari itu pernyataan kontrol pada PHP sendiri merupakan salah satu unsur penting yang selalu kita gunakan.

3.2. Struktur Pemilihan

Struktur pemilihan adalah blok kode program yang dapat menentukan statemen mana yang bernilai *true*, tergantung dari kondisi yang didefinisikan di dalamnya.

Dalam PHP, pemilihan statemen dapat dilakukan dengan statemen “if” atau “switch”

Statemen if

Struktur kontrol *if* memeriksa isi ekspresi, di mana jika terpenuhi atau bernilai benar, maka ia akan menjalankan perintah-perintah yang ada dalam blok.

Bentuk umum pernyataan *if* sederhana, adalah sbb:

```
if (ekspresi1) {  
    perintah2;  
}
```

Pada bentuk umum blok pemilihan di atas, *statemen* akan dieksekusi apabila *kondisi* bernilai *true*. Sebagai catatan bahwa dalam bahasa pemrograman PHP, kondisi yang akan diperiksa dalam blok pemilihan if harus diapit oleh tanda kurung.

Tanda kurung kurawal buka {, dan kurung kurawal tutup }, bersifat opsional. Tanda kurung kurawal buka dan tanda kurung kurawal tutup digunakan jika statemen yang akan di eksekusi lebih dari satu baris.

Bentuk yang lebih luas:

```
if (ekspresi1) {  
    perintah1;  
}  
elseif(ekspresi2) {  
    perintah2;  
}  
else {  
    perintah3;  
}
```

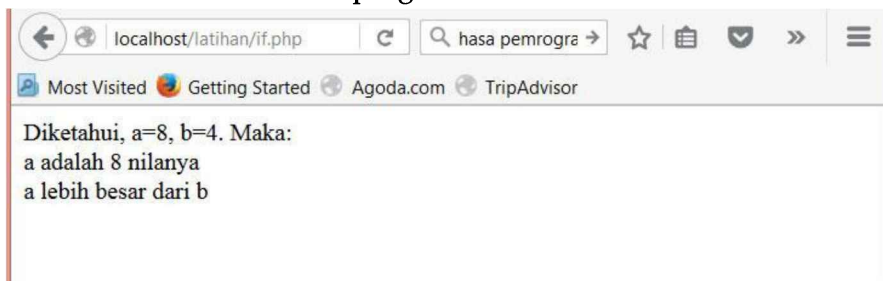
Contoh statemen sederhana:

```
<?php
$a = 8;
$b = 4;
echo "Diketahui, a=$a, b=$b. Maka:<br />";
// if sederhana
if ($a == "8"){
    echo "a adalah 8 nilanya<br />";
} else {
    echo "a tidak sama dengan 8<br />";
}
?>
```

Contoh statemen lebih kompleks:

```
<?php
// if lebih luas
if ($a > $b){
    echo "a lebih besar dari b<br />";
} elseif ($a < $b){
    echo "a tidak lebih kecil dari b<br />";
} elseif ($a == $b){
    echo "a sama dengan b<br />";
} else{
    echo "tidak teridentifikasi";
}
?>
```

Berikut ini hasil dari kode program di atas:



Gambar 3. 1 Menyimpan Statemen IF

Anda bisa menggunakan pernyataan IF dalam bentuk yang pendek, pernyataan IF pendek ini khusus untuk pernyataan IF yang tidak bersarang. Berikut diberikan pernyataan IF normal dan pernyataan IF pendek.

Pernyataan IF normal

```
<?php
echo "Saya ";
if ($umur > 27){
echo "sudah dewasa";
}
else
{
echo "masih muda";
}
?>
```

Pernyataan IF pendek

```
echo "Saya ", ($umur > 27 ? "sudah dewasa": "masih muda");
```

Jika dijalankan kedua kode program di atas, maka hasilnya akan sama.

Statement Switch

Serupa dengan pernyataan if-else if-else, pernyataan switch juga memberikan banyak alternatif pilihan pernyataan. Statemen switch merupakan statement yang dapat digunakan untuk melakukan proses pemilihan di dalam kode program PHP. Bentuk umum dari statemen switch adalah:

Adapun bentuk umum:

```
switch (nilai_ekspresi) {
case nilai1: statement1; break;
case nilai2: statement2; break;
default: statement_n;
}
```

Berikut ini contoh program pada PHP:

```
<?php
$angka = 3;
switch ($angka){
    case 0: $sterbilang = "NOL"; break;
    case 1: $sterbilang = "SATU"; break;
    case 2: $sterbilang = "DUA"; break;
    case 3: $sterbilang = "TIGA"; break;
    case 4: $sterbilang = "EMPAT"; break;
    default: $sterbilang = "Nilai diluar jangkuan!!";
}
echo "Bentuk terbilang dari angka $angka adalah
".$sterbilang;
?>
```

Kode program di atas melakukan inisialisasi dengan variable angka dengan nilai 3, kemudian melakukan switch dengan memberikan perbandingan nilai 3 ke dalam switch case apakah ada nilai 3 pada “case” nya jika ada maka akan menampilkan dalam variable \$sterbilang. Hasilnya seperti pada gambar di bawah ini.



Gambar 3. 2 Tampilan Statement Switch

Dalam penggunaan statemen switch program akan mencari nilai konstan yang cocok atau sama dengan nilai yang terkandung di dalam variabel. Jika ditemukan, maka statemen yang didefinisikan untuk nilai tersebut akan dieksekusi. Untuk bentuk umum di atas, cara kerjanya program akan membandingkan nilai1

dengan nilai variabel. Jika sama, maka program akan mengeksekusi statemen1, dan keluar dari blok pemilihan. Jika tidak sama, maka program akan melanjutkan proses perbandingan terhadap nilai2. Proses ini akan berlangsung sampai ditemukan nilai konstan yang cocok dengan nilai variabel. Jika ternyata semua nilai konstan yang didefinisikan tidak ada yang sama dengan nilai variabel, maka program akan mengeksekusi statemen 5, yaitu statemen yang terdapat pada bagian default.

Berikut contoh lain:

```
<?php
$no_hari = 2;
switch (no_hari) {
case 1: print "Hari ke-1: Minggu \n";
break;
case 1: print "Hari ke-1: Senin \n";
break;
case 1: print "Hari ke-1: Selasa \n";
break;
case 1: print "Hari ke-1: Rabu \n";
break;
case 1: print "Hari ke-1: Kamis \n";
break;
case 1: print "Hari ke-1: Jumat \n";
break;
case 1: print "Hari ke-1: Sabtu \n";
break;
default: print "Nama Hari tidak ditemukan"
break;
}
?>
```

3.3. Struktur Perulangan

Struktur perulangan (atau dalam bahasa inggris disebut dengan loop) adalah instruksi program yang bertujuan untuk mengulang beberapa baris perintah. Dalam merancang perulangan kode program, sedikitnya terdapat tiga komponen dari perulangan yaitu kondisi awal dari perulangan, perintah program yang akan diulang, serta kondisi akhir di mana perulangan akan berhenti.

Statemen For

Pernyataan `for` biasanya digunakan apabila jumlah perulangannya telah pasti. Bentuk Umum pernyataan `for` adalah sebagai berikut:

```
for (ekspresi1; ekspresi2; ekspresi3) {  
    Statemen;  
}
```

ekspresi1 → inisialisasi/penentuan nilai awal konter

ekspresi 2 → perumusan kondisi

ekspresi 3 → operasi penaikan/penurunan konter

Looping dengan *For* disebut juga *determinate loop*, artinya looping yang jumlah perulangannya (iterasi) telah ditentukan di awal *looping*. Ada beberapa bagian penting dari *Looping* ini:

- *Initialization Expression*
Initialization Expression dijalankan satu kali, pada saat looping dimulai. Biasanya bagian ini digunakan untuk menginisialisasi perhitungan
- *Stop Condition*
Stop Condition diperiksa nilainya sebelum setiap iterasi dieksekusi. Apabila *Condition* bernilai *false*, iterasi dihentikan
- *Iterative Expression*
Iterative Expression dilakukan setelah iterasi dieksekusi. Bagian ini biasanya digunakan untuk menambah nilai perhitungan
- *Loop Body*
Loop Body dieksekusi sekali setiap iterasi, merupakan perintah yang ingin kita lakukan berulang – ulang

Berikut ini contoh programnya:

```
<?php  
for ($x=1;$x<=10;$x++) //Nilai Awal, Akhir, Step  
{
```

```

        echo "<font size=$x>Ukuran Huruf<br
/></font>"; //Statement
    }
?>

```

Struktur di atas adalah bentuk perulangan for. Nilai awal adalah batas kondisi awal perulangan, Nilai akhir adalah batas kondisi akhir perulangan, sedangkan Step adalah step perulangan (Tingkat Kenaikan). Statement adalah pernyataan yang diulang.



Gambar 3. 3 Tampilan Statement For

Statemen Perulangan dalam Perulangan

Dalam bahasa pemrograman, sering kali terjadi di mana kode perulangan tersebut berada dalam perulangan yang lain. Untuk lebih jelasnya perhatikan kode PHP di bawah ini yang menjelaskan Struktur Perulangan dalam Perulangan.

```

<?php
for ($x=1; $x<=10; $x++)
{
    for ($y=1; $y<=$x; $y++)
    {
        echo " # "; //Statement
    }
    echo "<br />"; //Statement
}
?>

```

Contoh di atas terlihat statement diproses oleh dua perulangan yaitu perulangan dengan ekspresi X dan Y.

Statemen While

Pernyataan *while* akan melakukan memproses secara berulang terhadap pernyataan berdasarkan atas kondisi.

```
while (kondisi) {  
    Statemen;  
}
```

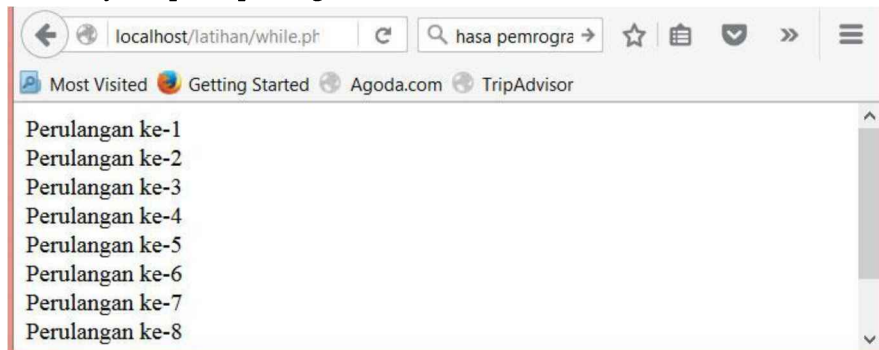
While Loop juga dikenal dengan istilah *indeterminate loop*, artinya jumlah pengulangannya tidak ditentukan pada awal perulangan. *While loop* lebih sederhana daripada *For Loop* karena hanya memiliki dua bagian, yaitu:

- *Stop Condition*
Stop Condition diperiksa nilainya sebelum setiap iterasi dieksekusi. Selama *stop condition* bernilai *true*, perintah dalam *loop body* akan dilakukan berulang-ulang. Iterasi akan dihentikan apabila *stop condition* bernilai *false*
- *Loop Body*
Sama seperti pada *for loop*, *loop body* dilaksanakan satu kali setiap iterasi. *Loop* di atas akan berjalan terus tanpa henti, karena tidak ada perintah yang mengubah nilai *stop condition*.

Contoh struktur sederhana dalam perulangan *while* dapat dilihat pada kode program di bawah ini:

```
<?php  
$i = 1;  
while ($i <= 10) {  
    echo "Perulangan ke-$i<br />";  
    $i++;  
}??>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 3. 4 Tampilan Statement While

Statemen Do-While

Perulangan dengan perintah Do-While merupakan kebalikan dari struktur perulangan While. Jika pada struktur While, pendeteksian ekspresi dilakukan pada awal perulangan, Pada struktur Do-While ini pendeteksian ekspresi dilakukan setelah proses satu loop dilakukan. Untuk lebih jelasnya, perhatikan struktur perulangan Do-While di bawah ini.

```
<?php
$x=1; // Nilai Variable
do
{
echo "$x. Nurul Imam<br />"; //Statement
$x++;
}
while ($x<=10); //Ekspresi
?>
```

Ekspresi merupakan kondisi yang harus dipenuhi sebagai syarat terjadinya perulangan. Jika kondisi pada ekspresi masih terpenuhi, maka perulangan akan masih dilakukan

Statemen Foreach

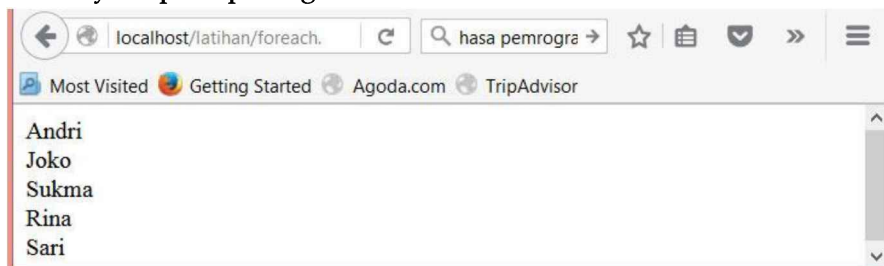
Array merupakan tipe data yang sering digunakan dalam membuat program menggunakan PHP. Kemampuan array dalam menyimpan banyak data dalam satu variabel akan sangat berguna untuk menyederhanakan dan menghemat penggunaan variabel.

Untuk menampilkan dan memproses data dari array, kita bisa memanfaatkan perulangan for, seperti contoh berikut ini:

```
<?php
$nama =
array("Andri", "Joko", "Sukma", "Rina", "Sari");

for ($i=0; $i <5; $i++)
{
    echo "$nama[$i]";
    echo "<br />";
}
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 3. 5 Tampilan Statement Foreach

Contoh di atas membuat perulangan for sebanyak 5 kali, dengan variabel counter \$i dimulai dari angka 0 (karena index array dimulai dari angka 0). Namun sebagai cara alternatif untuk menampilkan array, saya akan mengubah kode di atas dengan menggunakan perulangan foreach:

```
<?php
$nama =
array("Andri", "Joko", "Sukma", "Rina", "Sari");
```

```
foreach ($nama as $val)
{
    echo "$val";
    echo "<br />";
}
?>
```

Pada contoh di atas, saya mendefinisikan variabel array `$nama` dengan format singkat, dan tanpa mendefinisikan key secara tertulis. Variabel merupakan variabel perantara dalam contoh di atas. Perulangan tersebut akan diulang sebanyak data yang terdapat di dalam array, sehingga kita tidak perlu harus menghitung seberapa banyak perulangan yang harus dilakukan.

Jika anda membutuhkan nilai key dari array untuk dapat diproses, maka PHP menyediakan bentuk kedua dari perulangan `foreach`, dengan format dasar penulisan sebagai berikut:

```
foreach ($nama_array as $key => $value)
{
    statement ($key...$value...)
}
```

Perbedaan dengan format sebelumnya, di sini PHP menyediakan variabel perantara kedua, yaitu variabel `$key`. Variabel `$key` ini menampung nilai key dari array.

Statemen Break

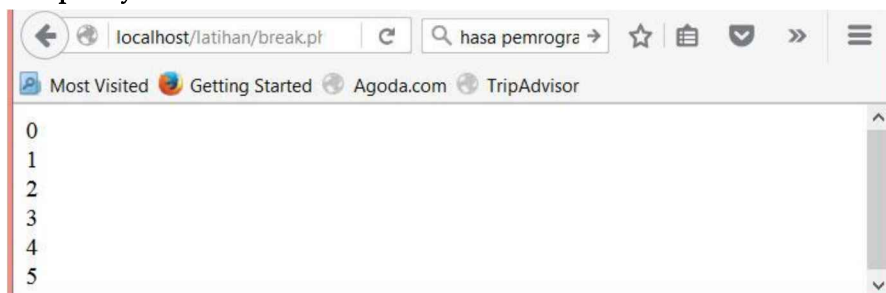
Perintah `break` jika digunakan di dalam perulangan berfungsi untuk 'menghentikan paksa' proses perulangan yang berlangsung. Ketika proses perulangan berjalan, ada kalanya kita ingin segera keluar dari perulangan jika sebuah kondisi tertentu telah terpenuhi, sehingga sisa proses perulangan tidak perlu dijalankan. Sebagai contoh Misalkan kita memiliki nama-nama mahasiswa yang tersimpan di dalam sebuah array atau di dalam *database*. Proses pencarian sederhana dapat dirancang dengan

melakukan pencocokan secara berulang dimulai dari nama pertama, kedua, dan seterusnya. Perulangan ini akan dilakukan sebanyak daftar mahasiswa yang ada. Akan tetapi, jika nama yang dicari telah ditemukan, proses perulangan seharusnya dapat dihentikan saat itu juga, karena tujuan pencarian nama telah selesai.

Untuk keperluan inilah PHP menyediakan instruksi break. Break berfungsi sebagai perintah kepada web server untuk menghentikan perulangan secara prematur, yaitu menghentikan perulangan di luar dari yang direncanakan. Untuk lebih jelasnya dapat melihat contoh di bawah ini:

```
<?php
for ($i=0; $i <100; $i++)
{
    if ($i==13)
    {
        break;
    }
    echo $i;
    echo "<br />";
}
?>
```

Outputnya:



Gambar 3. 6 Tampilan Statement Break

Dalam program di atas, saya membuat perulangan for dari 0 sampai 100, dan dalam keadaan normal, perintah for (\$i=0; \$i <100; \$i++) akan memproses perulangan sebanyak 100 kali.

Namun pada baris ke-4 saya menambahkan sebuah struktur IF yang menyatakan bahwa jika nilai variabel counter \$i sama dengan 6, maka break. Perintah break akan membuat perulangan for langsung dihentikan, dan kita hanya menghasilkan perulangan sampai angka 6.

Statemen Continue

Continue berfungsi untuk melewati satu iterasi / putaran dalam rangkaian loop. Untuk lebih jelasnya perhatikan contoh di bawah ini:

```
<?php
// melakukan continue pada $i == 2
for($i = 0; $i<5; $i++){
    if ($i == 2);
    continue;
}

echo "Nilai i:$i<br>";
?>
```

Kode tersebut akan melakukan break pada saat i bernilai 2, sehingga akan menghasilkan output sebagai berikut:

Nilai i: 0

Nilai i: 1

Nilai i: 3

Nilai i: 4

Loop Selesai

Statemen Return

Perintah return berfungsi untuk memerintahkan kode untuk keluar dari fungsi atau mengembalikan nilai dari fungsi tersebut

kepada yang memanggilnya. Untuk lebih jelasnya dapat dilihat di contoh di bawah ini:

```
<?
function testReturn(){
    For($i = 0;$1<5; $i++){
        //melakukan return pada $i==2
    if($i == 2)
        return;
    }
    echo("Nilai I: $i<br>");
}
//jalankan function
testReturn();
echo("Function selesai");
?>
```

Output dari sistem di atas adalah

Nilai i: 0

Nilai I: 0

Function Selesai

Perhatikan bahwa looping selesai tidak dijalankan. Ini menunjukkan bahwa setelah return dieksekusi, program langsung keluar dari function dan mengeksekusi perintah setelah function, yaitu echo("Function Selesai");

Statemen Exit

Exit digunakan untuk menghentikan keseluruhan script php dan keluar dari blok program tersebut. Fungsi exit ini banyak digunakan untuk melakukan debugging terhadap sebuah kode program. Untuk lebih jelasnya dapat dilihat contoh di bawah ini:

```
<?
function testExit(){
    for($i = 0;$1<5; $i++){
        //melakukan return pada $i==2
        if($i == 2){
```

```

        exit;
    }
    echo("Nilai I: $i<br>");
}
Echo("Loop Selesai");
}
//jalankan function
testExit();
echo("Function selesai");
?>

```

Output dari sistem di atas adalah

Nilai i: 0

Nilai I: 0

Function Selesai

Perhatikan bahwa baris echo("Fuction Selesai"); tidak dijalankan

Rangkuman

Jenis-jenis Statemen Kontrol terdiri dari Struktur Pemilihan (if dan switch), Struktur Perulangan (*for*, *while*, *do-while*, *foreach*) dan statemen Loncat (*break*, *continue*, *go to*, *return* dan fungsi *exit*). Setiap struktur pemilihan memiliki fungsi yang berbeda secara kinerja, tetapi ada beberapa yang mempunyai hasil yang sama. Pemilihan struktur penulisan tergantung dari kasus atau persoalan dapat diselesaikan dengan struktur pemilihan mana.

Latihan:

1. Buatlah program untuk mendeteksi seseorang dikatakan Orang Tua, Dewasa atau anak-anak menggunakan bahasa PHP. Pilih struktur pemilihan yang paling cocok untuk kasus ini.
2. Buatlah Deret Bilangan prima menggunakan pengulangan For dan While menggunakan bahasa PHP.

BAB IV

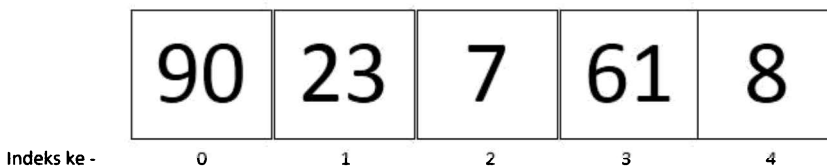
ARRAY DAN AKSES FILE

Setelah mempelajari bab ini, pembaca dapat memahami dapat membuat array dengan menggunakan bahasa pemrograman PHP kedalam suatu halaman web, baik itu array satu dimensi atau array dua dimensi. Selain itu dapat memahami bagaimana PHP dapat mengakses file termasuk mengubah dan menampilkan isi file.

4.1. Array

Array merupakan sebuah variable kompleks yang digunakan untuk menyimpan sederetan data lebih dari satu informasi dan disimpan dalam satu buah nama variable. Perbedaannya dengan variable biasa adalah kemampuannya untuk menyimpan lebih banyak data, dan memiliki perbedaan cara penulisan di mana array memiliki indeks yang ditulis dalam sebuah kotak []. Indeks adalah suatu penanda yang menjadi ciri khas dari informasi/data dan menjadi pembeda dengan informasi/data yang lain yang disimpan dalam satu array yang sama. Indeks bisa berupa angka (integer) atau huruf (string) dan index dimulai dari angka nol.

Konsep pengertian array merupakan sederetan nilai yang diberikan index seperti ilustrasi pada gambar di bawah ini:



Gambar 4. 1 Ilustrasi Array Sederhana

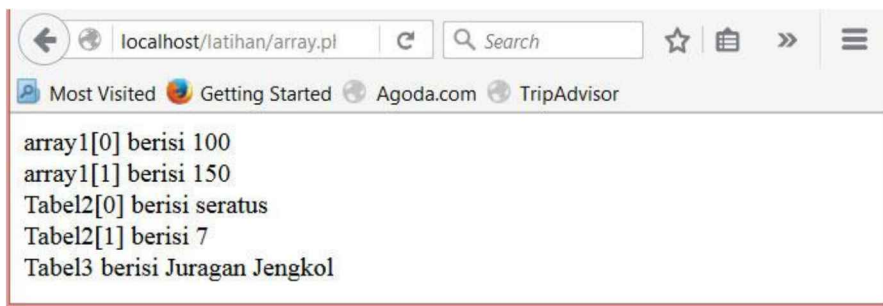
Contoh Penulisan Array

```
<?php
    //Inisialisasi Array cara 1
    $array1[] = 100; //array1[0] akan berisi nilai
100
    $array1[] = 150; //array1[1] akan berisi nilai
150
    //Cara 2
    $tabel2 = array("seratus",7); //string dan
integer
    //Cara lain
    $tabel3 =
array("nol"=>"Juragan","satu"=>"Jengkol");
    //Hasilnya
    echo "array1[0] berisi $array1[0] <br>";
    echo "array1[1] berisi $array1[1] <br>";
    echo "Tabel2[0] berisi $tabel2[0] <br>";
    echo "Tabel2[1] berisi $tabel2[1] <br>";
    echo "Tabel3 berisi $tabel3[nol]
$tabel3[satu]<br>";
?>
```

Berikut penjelasan kode program di atas:

```
$array1[] = 100; //array1[0] akan berisi nilai 100
$array1[] = 150; //array1[1] akan berisi nilai 150
```

PHP akan membuatkan sebuah array dengan nama array1, jika tanda siku [] tidak diisi maka pernyataan pertama merupakan array ber-index 0 dan yang selanjutnya akan berurutan yaitu 1,2,3 dan seterusnya. Kode program di atas jika dijalankan akan menghasilkan seperti pada gambar di bawah ini:



Gambar 4. 2 Tampilan Array Sederhana

Dari contoh-contoh di atas dapat dilihat bahwa array memiliki struktur yang terdiri dari nama array, indeks array dan isi array, `$(nama_array)[$indeks]=(isi_array);`

Misalkan kita akan membuat array sebagai berikut: array ke 1 berisi 2001, array ke 2 berisi 2002, array ke 3 berisi 2003 dan array ke 4 berisi 2004. Maka kita bisa membuat dengan penulisan seperti di bawah ini:

```
<?php
$nilai_array = array(1=>2001, 2=>2002, 3=>2003);
// atau bisa seperti ini:
$nilai_array2 = array(1=>2001, 2002, 2003);
?>
```

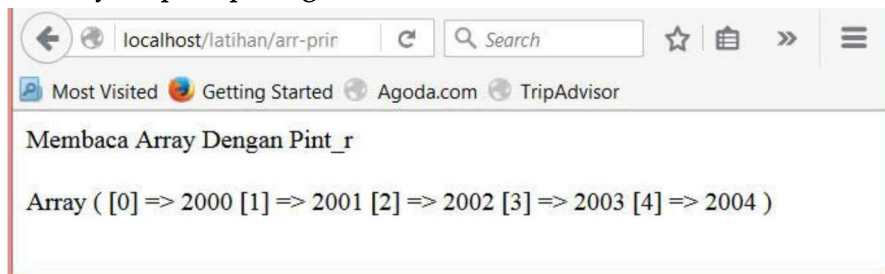
Membaca Array

Ada dua kode program yang digunakan untuk membaca array dalam bahasa PHP, diantaranya adalah `print_r($namaArray);` dan `var_dump($namaArray);` sedangkan `foreach ($namaArray as $indeksArray => $isiArray) {blok pernyataan;}` digunakan untuk menampilkan semua data yang ada dalam sebuah array secara berurutan dan menampilkan isi array berdasarkan indeks yang bersesuaian. Sedangkan `foreach ($namaArray as $isiArray) {blok pernyataan;}` untuk menampilkan array berdasarkan isi array-nya saja.

Membaca Array dengan perintah print_r

```
<?php //Misal ada Array Tahun
    $tahun[]=2000;
    $tahun[]=2001;
    $tahun[]=2002;
    $tahun[]=2003;
    $tahun[]=2004;
    echo "<b>Membaca Array Dengan Pint_r</b><br/><br/>";
    print_r($tahun);
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 3 Menampilkan Array Dengan Perintah print_r

Membaca Array dengan var_dump

```
<?php //Misal ada Array Hobi
    $hobi=array("makan", "tidur", "nangis", "merenung");
    echo "<b>Membaca Array Dengan
Var_dump</b><br/><br/>";
    var_dump($hobi);
?>
```

Hasilnya seperti pada gambar di bawah ini:

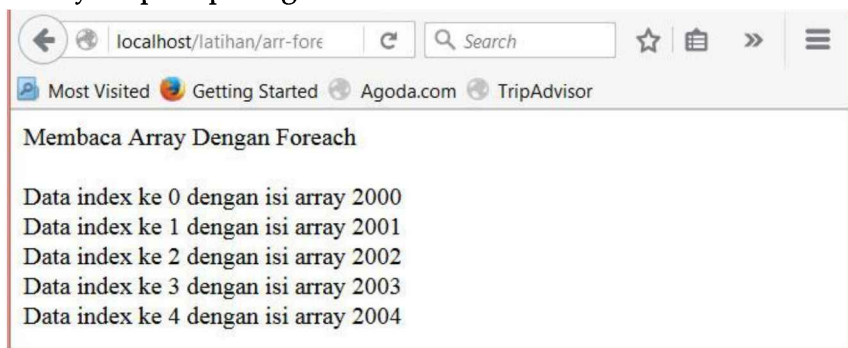


Gambar 4. 4 Menampilkan Array Dengan Perintah var_dump

Membaca Array dengan pengulangan foreach

```
<?php //Misal ada Array Tahun
    $tahun[]=2000;
    $tahun[]=2001;
    $tahun[]=2002;
    $tahun[]=2003;
    $tahun[]=2004;
    echo "Membaca Array Dengan Foreach<br/><br/>";
    foreach($tahun as $indeks => $isi)
    {
        echo "Data index ke $indeks dengan isi array
    $isi<br/>";
    }
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 5 Menampilkan Array Dengan foreach

Mengubah Isi Array

Mengubah isi array secara sederhana adalah melakukan overwrite dari array yang ada, kemudian menampilkannya kembali. Misalkan kita memiliki array yang `$tahun[2]="2002"`, isi array '2002' akan dirubah menjadi '0000 Berubah', dengan cara `$hobi[2]="000 Berubah"`; berikut contoh kode programnya:

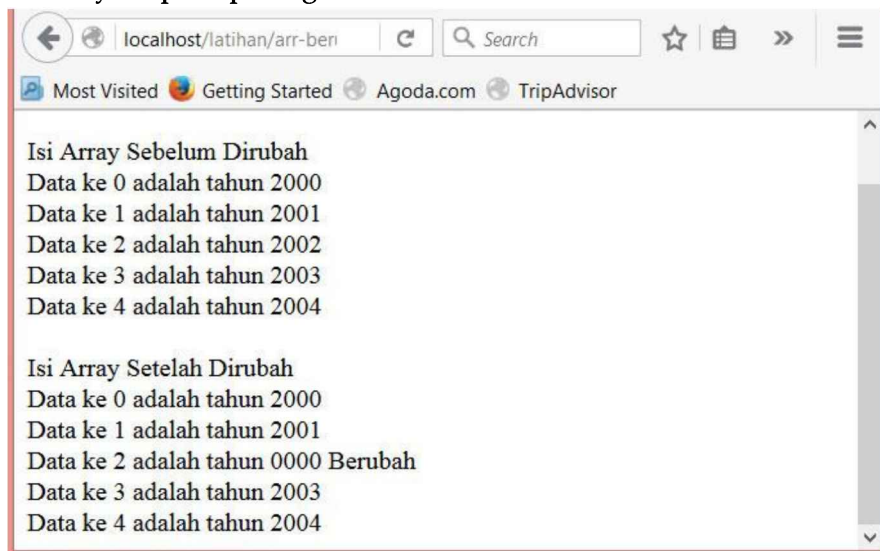
```
<?php
    //Misal ada Array Tahun
    $tahun[]=2000;
```

```

$tahun[]=2001;
$tahun[]=2002;
$tahun[]=2003;
$tahun[]=2004;
echo "<b>Mengubah Array</b><br/><br/>";
echo "Isi Array Sebelum Dirubah<br/>";
foreach($tahun as $indeks => $isi)
{
    echo "Data ke $indeks adalah tahun $isi<br/>";
}
//Indeks ke 2 dirubah jadi string
$tahun[2]="0000 Berubah";
echo "<br/>Isi Array Setelah Dirubah<br/>";
foreach($tahun as $indeks => $isi)
{
    echo "Data ke $indeks adalah tahun $isi<br/>";
}
?>

```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 6 Mengubah Isi Array

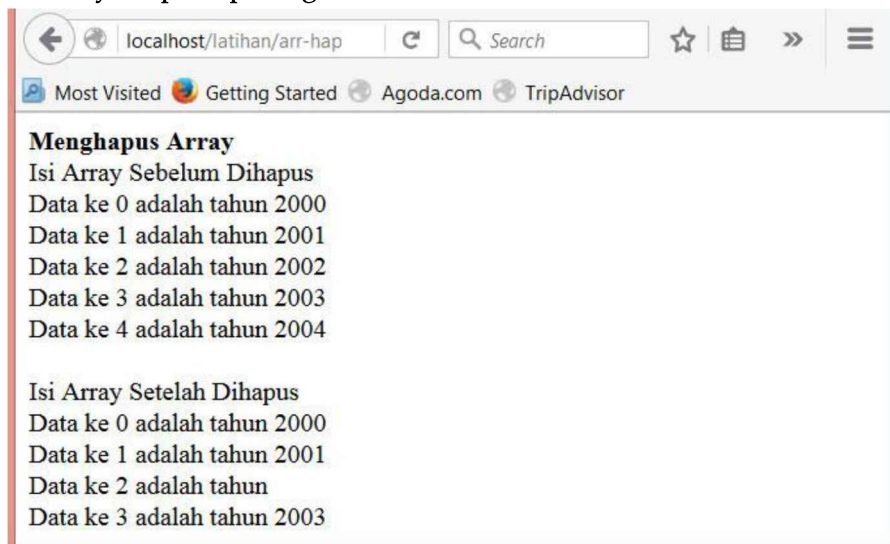
Menghapus Array

Menghapus sebuah array dengan mengisi array kosong pada data array yang diinginkan dengan perintah `unset($namaArray[indeksArray]);` jika tidak disertakan `indeksArray` maka efeknya adalah akan menghapus semua isi array yang bersangkutan.

Contoh Menghapus Isi Array

```
<?php
//Misal ada Array Tahun
$tahun[]=2000;
$tahun[]=2001;
$tahun[]=2002;
$tahun[]=2003;
$tahun[]=2004;
echo "<b>Menghapus Array</b><br/>";
echo "Isi Array Sebelum Dihapus<br/>";
foreach($tahun as $indeks => $isi)
{
    echo "Data ke $indeks adalah tahun
$isi<br/>";
}
$tahun[2]=""; //Indeks ke 2 dihapus, cara1
unset($tahun[4]); //indeks ke 4 dihapus, cara2
echo "<br/>Isi Array Setelah Dihapus<br/>";
foreach($tahun as $indeks => $isi)
{
    echo "Data ke $indeks adalah tahun
$isi<br/>";
}
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 7 Menghapus Data Isi Array

Mengurutkan Isi Array

Array dapat ditampilkan secara khusus dengan menggunakan fungsi dalam PHP, yaitu:

Tabel 4. 1 Mengurutkan Array

Fungsi Pengurutan	Hasil Eksekusi
sort(namaarray)	Pengurutan berdasarkan nilai, secara ascenden, dan mengubah indeks mulai dari nol sesuai urutan yang baru
asort(namaarray)	Pengurutan berdasarkan nilai tanpa mengubah indeks
rsort(namaarray)	Pengurutan berdasarkan nilai, secara descenden/terbalik, dan mengubah indeks dari nol sesuai urutan yang baru
arsort(namaarray)	Pengurutan terbalik berdasarkan nilai tanpa mengubah indeks
ksort(namaarray)	Pengurutan berdasarkan indeks

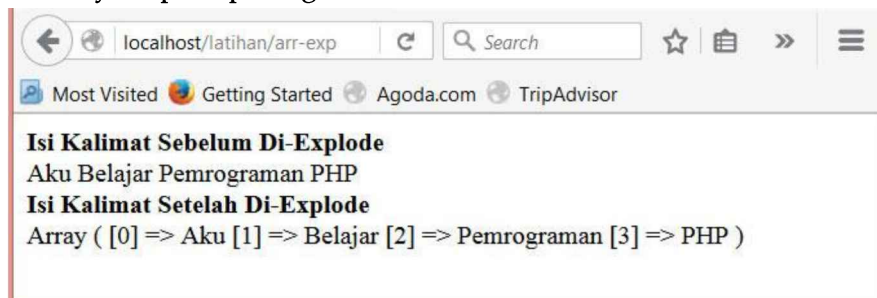
Mengubah String Menjadi Array

Pengolahan data seringkali membutuhkan pengonversian dari array menuju string, maupun sebaliknya. Dalam PHP, fungsi untuk mengubah string menjadi Array adalah Explode dengan struktur `$namaArray=explode(pemisah,$variabelString)`.

Contoh kode program di bawah adalah melakukan membuat string menjadi array dengan menggunakan fungsi explode dengan pemisah space.

```
<?php
//Misal ada Variabel Kalimat berisi
$kalimat="Aku Belajar Pemrograman PHP";
//Dipecah pake fungsi explode
$katakan=explode(" ",$kalimat);
echo "<b>Isi Kalimat Sebelum Di-Explode</b><br/>";
echo "$kalimat";
echo "<br/><b>Isi Kalimat Setelah Di-
Explode</b><br/>";
print_r($katakan);
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 8 Mengubah String Menjadi Array

Mengubah Array Menjadi String

Menggunakan Implode untuk mengubah sebuah array menjadi sebuah variabel string, fungsi ini kebalikan dari Explode dengan struktur `$namaVariabel = implode (penghubung, $namaArray)`.

Contoh programnya seperti pada kode program di bawah ini:

```
<?php
//Misal ada oray seperti berikut
$hobiku[0]="nguseup";
$hobiku[1]="bobogohan";
$hobiku[2]="nongton";
$hobiku[3]="mengbal";
echo "<b>Isi Array Sebelum Di-Implode</b><br/>";
print_r($hobiku);
echo "<br><b>Penjelasan Isi Array</b><br/>";
foreach($hobiku as $indeks => $isi)
{
    echo "Hobiku ke $indeks adalah <i>$isi</i><br/>";
}
echo "<b>Isi Array Setelah Di-Implode</b><br/>";
$kalimat=implode(" ", $hobiku);
echo "Hobiku adalah $kalimat";
?>
```

Memecah Array

Dalam PHP, memecah Array artinya adalah membuat sebuah array baru yang isinya adalah merupakan bagian dari sebuah array yang lain. Atau dengan kata lain, jika terdapat sebuah array maka bagian isi dari array tersebut dapat dimasukkan kedalam sebuah array baru. Untuk melakukan hal ini, maka diperlukan fungsi PHP yaitu fungsi `array_slice`, dengan struktur `$namaArray=array_slice($namaArray,N1,N2)`, di mana N1 adalah awal indeks dari array lama yang ingin dimasukkan ke dalam array baru. N2 adalah jumlah data yang ingin dimasukkan ke dalam array baru dari array lama (dihitung sejak awal indeks yang diinginkan).

Berikut ini contoh kode programnya:

```
<?php
//Misal ada oray seperti berikut
$hobiku[0]="nguseup";
$hobiku[1]="bobogohan";
$hobiku[2]="nongton";
```

```

$hobiku[3]="mengbal";
echo "<b>Array Sebelumnya:</b><br/>";
Print_r($hobiku);
echo "<br/><b>Dipecah Menjadi:</b><br/>";
//mengambil array ke 1 dan 2 sebagai sub array
$sub_hobiku=array_slice($hobiku,1,2);
Print_r($sub_hobiku);
?>

```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 9 Memecah String

Array dari variable \$sub_hobiku mengambil sebagian dari array \$hobiku, dimulai dari indeks ke 1 sampai ke 2. Indeks pada array sub_hobiku di urutkan dari pertama lagi, artinya pada array hobiku untuk indeks ke 1 menjadi indeks ke 0 di sub-hobiku, begitu juga untuk array-array lainnya, indeksnya mengikuti wadah array yang baru.

Menggabungkan Array

Adalah membuat sebuah array baru yang isinya adalah gabungan dari dua buah array yang sudah ada atau lebih menggunakan fungsi `array_merge` dengan struktur `$namaArray=array_merge($array1,$array2,...)`.

Berikut ini contoh kode program untuk menggabungkan array:

```

<?php
//Misal Array Hobi berisi
$hobi[0]="mancing";
$hobi[1]="membaca";
$hobi[2]="nonton";

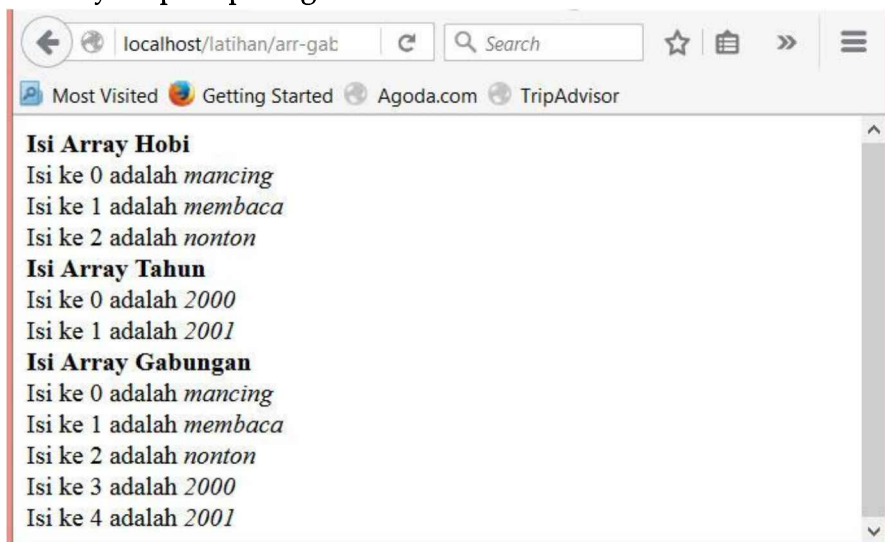
```

```

//Misal Array Tahun berisi
$tahun=array(2000,2001);
//Kedua Array digabung menjadi gabungArray
$gabungArray =array_merge($hobi,$tahun);
//Hasilnya
echo "<b>Isi Array Hobi</b><br/>";
foreach($hobi as $indeks => $isi_hobi)
{ echo "Isi ke $indeks adalah
<i>$isi_hobi</i><br/>";}
echo "<b>Isi Array Tahun</b><br/>";
foreach($tahun as $indeks => $isi_tahun)
{ echo "Isi ke $indeks adalah
<i>$isi_tahun</i><br/>";}
echo "<b>Isi Array Gabungan</b><br/>";
foreach($gabungArray as $indeks => $isi)
{
    echo "Isi ke $indeks adalah <i>$isi</i><br/>";
}
?>

```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 10 Menggabungkan String

4.2. Array Satu Dimensi dan Multidimensi

Array satu dimensi adalah elemen-elemen array dapat diakses oleh bahasa pemrograman PHP dengan menggunakan

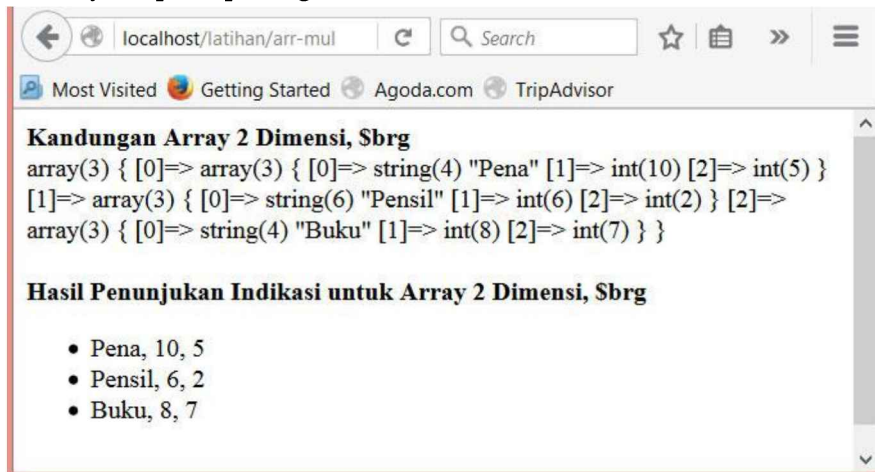
suatu indeks tertentu dan bertipe sama. Contoh array satu dimensi adalah array yang digunakan contoh pada sub bab-sub bab sebelumnya. Sedangkan array 2 dimensi bahkan array multidimensi merupakan dimensi sering kali digambarkan/dianalogikan sebagai sebuah matriks atau bentuk grid dengan istilah yang sederhana array 2 dimensi adalah array dalam array. Contoh programnya seperti pada kode program di bawah ini:

```
<?php
$brg = array
(
array("Pena", 10, 5),
array("Pensil", 6, 2),
array("Buku", 8, 7),
);

echo "<b>Kandungan Array 2 Dimensi, " . chr(36) .
"brg</b><br>";
echo var_dump($brg) . "<br><br>";

echo "<b>Hasil Penunjukan Indikasi untuk Array 2
Dimensi, " . chr(36) . "brg</b>";
echo "<ul>";
    echo "<li>".$brg[0][0].", ".$brg[0][1].",
".$brg[0][2]."</li>";
    echo "<li>".$brg[1][0].", ".$brg[1][1].",
".$brg[1][2]."</li>";
    echo "<li>".$brg[2][0].", ".$brg[2][1].",
".$brg[2][2]."</li>";
echo "</ul>";
echo "<br>";
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 11 Array Dua Dimensi

4.3. Akses File

Dalam management file dan direktori, PHP menyediakan banyak fungsi, yaitu lebih dari 70 fungsi, terutama yang berhubungan dengan manajemen file seperti create, write, append dan delete. Untuk melakukan penyimpanan data dengan PHP, maka kita dapat menggunakan beberapa fungsi bawaan yang telah ada pada PHP yaitu: `fopen`, `fputs`, `fread`, `fwrite`, `fgets`.

Selain menggunakan beberapa fungsi bawaan pada PHP, kita juga menggunakan beberapa mode akses file, yaitu:

Tabel 4. 2 Fungsi Akses File

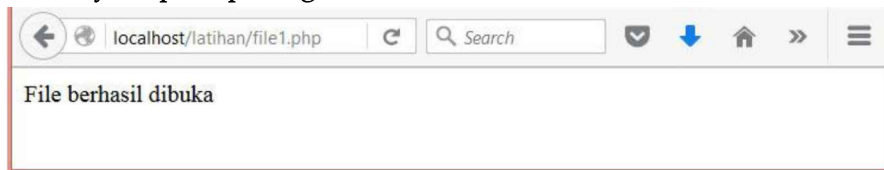
Mode	Keterangan
r	Membuka file untuk dibaca
r+	Membuka file untuk dibaca dan ditulis. Data akan ditulis di awal file.
w	Hanya untuk tulis file, isi file lama dihapus, jika file belum ada maka akan di-create
w+	Untuk baca dan tulis file, isi file lama dihapus, jika file belum ada maka akan di-create
a	Hanya untuk menambahkan isi file, pointer berada di akhir

Mode	Keterangan
	file, jika file belum ada maka di-create
a+	Untuk membaca dan menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create

Untuk mengakses file, maka kita harus siapkan sebuah file dengan nama file_data.txt pada folder latihan kita, kemudian isi dengan kalimat "Ini File". Pada kode program di bawah diilustrasikan bagaimana membuka dan menutup sebuah file.

```
<?php
$namafile = "file_data.txt"; //definikan nama file
$handle = fopen ($namafile, "r"); // mode yang
digunakan untuk eksekusi file
if (! $handle) { // cek kondisi jika file nya tidak ada
maka
    echo "File tidak dapat dibuka, atau file belum
ada"; // tampilkan pesan file tidak ada
} else { // selain itu, jika file nya ada maka
    echo "File berhasil dibuka"; // tampilkan pesan
file berhasil dibuka
}
fclose ($handle); //tutup
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 4. 12 Membuka File

Beberapa fungsi akses file dengan PHP sebagai berikut:

Membuka & membuat file

`fopen ($namafile, $mode);`

Keterangan: \$namafile merupakan nama file yang akan dibuat, sedangkan \$mode merupakan mode akses file

Contoh Program mengakses (membuka) file dengan mode w

Menulis di file

Untuk menulis ke file, dapat menggunakan fungsi `fwrite($handle, $isi)` dan `fputs($handle, $isi)`. Kedua fungsi ini memiliki fungsi yang sama

Membaca isi file

fungsi `fread($handle, [$panjang]);` akan membaca isi file setiap jumlah byte tertentu.

Fungsi `fgets($handle, [$panjang]);` akan membaca isi file tiap baris.

Rangkuman

Array merupakan kumpulan data dengan tipe data yang sama. Array digunakan untuk menyimpan data yang cukup banyak dengan tipe data yang sama. Kita bisa mengolah array seperti memisahkan, menggabungkan, menghapus, mengurutkan dan manipulasi lainnya terhadap array.

Latihan

1. Buat program untuk mencari nilai max dan rata-rata dari 3 buah nilai inputan, gunakan fungsi dengan 3 nilai tsb dikirimkan dalam bentuk array sebagai parameter ke fungsi. Berikan juga return value ke program utama
2. Buat program kalkulator sederhana, gunakan fungsi dengan parameter yang dikirim ke fungsi berupa: 2 buah bilangan yang akan diproses (dalam bentuk array), operator bilangan. Berikan juga header dan footer yang dipanggil dengan perintah `include`.

BAB V

STRING DAN FUNGSI

Setelah mempelajari bab ini, pembaca diharapkan dapat membuat program untuk manipulasi string dan ekspresi regular. Selain itu juga dapat membuat fungsi dan class pada bahasa pemrograman PHP.

5.1. Fungsi String

Manipulasi String adalah proses rekayasa terhadap sebuah teks baik itu kata, kalimat atau paragraph oleh PHP sehingga hasilnya sesuai dengan keinginan. Misalnya membalikkan kata pada sebuah kalimat dapat dilakukan oleh PHP contoh lain adalah anda dapat menghitung jumlah karakter yang terdapat pada string, memfilter karakter atau kata tertentu yang terdapat di dalam string, mengganti karakter tertentu pada string dengan karakter lain dan lainnya. PHP menyediakan lebih dari 90 fungsi untuk manipulasi string, ada beberapa fungsi dasar untuk memanipulasi string dengan PHP yang harus anda ketahui. berikut adalah beberapa fungsi php yang berfungsi untuk memanipulasi string.

Menampilkan String

Untuk menampilkan data string dapat menggunakan perintah echo untuk mencetak data string dalam jendela browser, selain itu, dapat juga menggunakan perintah Print atau Printf.

Contoh berikut penulisan perintah print adalah:

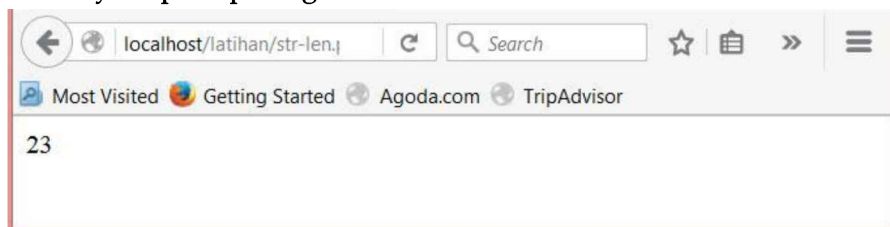
```
<?php
echo "maharani";
//Sama dengan
echo "maharani";
//Sama dengan
echo ("maharani");
?>
```

Menghitung Jumlah Karakter

Untuk menghitung jumlah dari sebuah string anda dapat menggunakan fungsi `strlen()`. Untuk contoh dan cara penulisan dari fungsi `strlen` untuk menghitung jumlah karakter di string php dapat di perhatikan pada contoh berikut ini:

```
<?php
$kalimat = "Belajar pemrograman PHP";
echo strlen($kalimat);
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 5. 1 Menghitung Jumlah Karakter

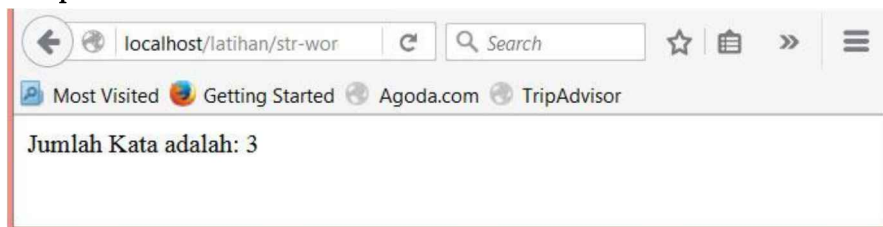
Menghitung Jumlah Kata

Untuk menghitung jumlah kata yang terdapat pada sebuah string anda bisa menggunakan fungsi yang sudah di sediakan oleh php yaitu fungsi `str_word_count()`. berikut adalah cara penulisan fungsi `str_word_count()` untuk menghitung jumlah kata pada sebuah string.

Contoh kode programnya seperti pada gambar di bawah ini:

```
<?php
$kalimat = "Belajar pemrograman PHP";
echo "Jumlah Kata adalah: ".str_word_count($kalimat);
?>
```

Output:



Gambar 5. 2 Menghitung Jumlah Kata

Membalik String

Fungsi `strrev` digunakan untuk membalikkan urutan karakter pada string. langsung saja untuk cara penulisan fungsi `strrev()` pada php seperti contoh berikut ini:

```
<?php
$kalimat = "Belajar pemrograman PHP";
echo strrev($kalimat);
?>
```

Output:

rajaleB namargormep PHP

Memfilter dan mengganti string

Fungsi `str_replace` berfungsi sebagai pengganti string. misalnya di dalam sebuah string terdapat kata atau huruf yang ingin di ganti dengan kata lain maka bisa menggunakan fungsi `str_replace()`. untuk format penulisan nya seperti contoh berikut ini.

```
Str_replace("kata yang ingin diganti","kata
pengganti","si string");
```

Dapat di lihat pada contoh penulisan fungsi `str_replace()` php di atas bahwa fungsi `str_replace` memiliki 3 parameter. parameter pertama adalah kata yang ingin di ganti, parameter kedua adalah kata yang digunakan sebagai pengganti, dan

parameter ketiga adalah isi string. untuk contoh penggunaannya perhatikan pada contoh di bawah ini.

```
<?php
$kalimat = "Belajar pemrograman PHP";
echo str_replace("Belajar", "Tutorial", $kalimat);
?>
```

Output:

Tutorial Pemrograman PHP

Dapat di lihat pada contoh di atas. awalnya isi dari variabel string kalimat adalah “Belajar Pemrograman PHP”. kemudian dengan fungsi str_replace kita mengganti kata “Belajar” dengan “Tutorial”. Sehingga hasil output dari str_replace() adalah “Tutorial pemrograman PHP”.

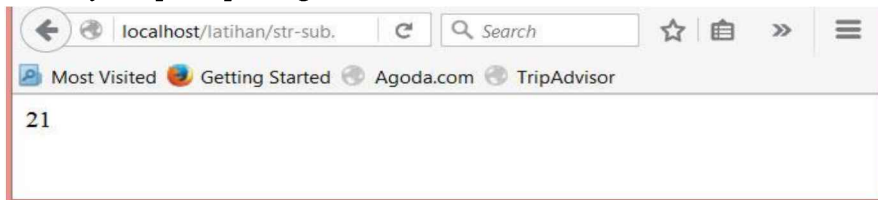
Mengambil Sebagian String

Untuk mengambil sebagian string pada sebuah kalimat atau kata, gunakan fungsi substr. Fungsi banyak digunakan pada kalangan programmer PHP dan sangat memudahkan.

Contoh ilustrasi seperti ini, NIP sebuah perusahaan terdiri dari tahun-tanggal_lahir-bulan_lahir dan nomor urut seperti ini: 19842108001 yang berarti 4 digit pertama adalah taun lahir, dua digit berikutnya adalah tanggal lahir, dua digit berikutnya adalah bulan lahir dan empat digit berikutnya adalah nomor urut. Anda diminta untuk mengambil tanggal lahir orang tersebut. Gunakan fungsi substr() seperti pada kode program di bawah ini:

```
<?php
$nip = "19832108001";
$sambil_tanggal = substr($nip, 4, 2);
echo $sambil_tanggal;
// 21
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 5.3 Mengambil String

Contoh program lain:

```
<?php
echo substr('abcdef', 1);      // bcdef
echo substr('abcdef', 1, 3);   // bcd
echo substr('abcdef', 0, 4);   // abcd
echo substr('abcdef', 0, 8);   // abcdef
echo substr('abcdef', -1, 1);  // f
?>
```

Fungsi-fungsi string dalam PHP bisa dilihat di website resmi PHP di alamat ini: <http://ca.php.net/manual/en/ref.strings.php>.

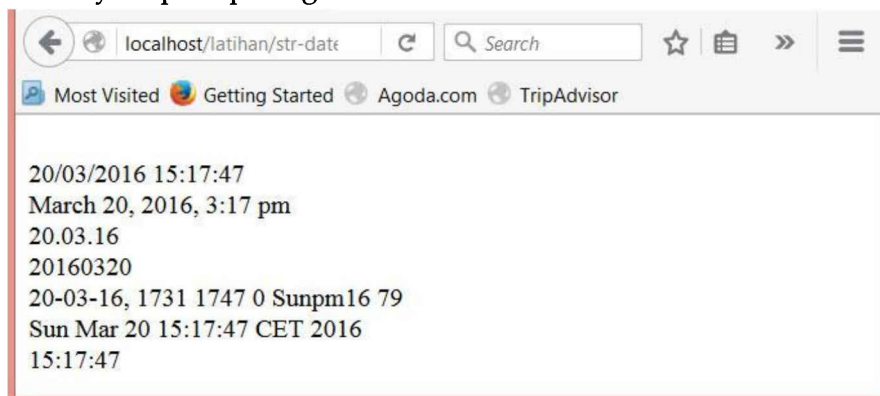
- `number_format()` – mengatur format angka.
- `str_replace()` – mengganti semua string dalam pola menjadi suatu string.
- `strlen()` – menghitung panjang string.
- `strpos()` – mencari posisi pertama sebuah string dalam string.
- `strev()` – membalik string.
- `strstr()` – mencari posisi pertama sebuah karakter dalam string.
- `strtolower()` – mengubah string menjadi huruf kecil (lower-case).
- `strtoupper()` – mengubah string menjadi huruf kapital (upper-case)
- `substr()` - memotong string
- dan lain-lain.

5.2. Fungsi Tanggal

Fungsi dalam Operasi tanggal dalam PHP yang utama dan sering digunakan adalah fungsi `date()`. Fungsi ini akan menghasilkan tanggal dan waktu server sekarang. Script di bawah akan menampilkan data waktu sekarang dengan berbagai tipe.

```
<?php
echo "<br>". date("d/m/Y H:i:s"); // Tanggal Sekarang
echo "<br>". date("F j, Y, g:i a");
echo "<br>". date("d.m.y");
echo "<br>". date("Ymd");
echo "<br>". date('j-m-y, it is w Day z ');
echo "<br>". date("D M j G:i:s T Y");
echo "<br>". date("H:i:s");
?>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 5. 4 Fungsi Tanggal

Beberapa pilihan parameter dari fungsi `date()` ini dapat dilihat pada tabel berikut ini:

Parameter	Keterangan	Contoh Nilai
Hari		
d	Tanggal 2 digit	01 s/d 31
D	Tiga digit dengan hari	Sun, Mon, ...
j	Tanggal tanpa nol	1 d/d 31
l	Nama hari lengkap	Sunday, Monday, ...
N	Nama hari digit 1	1 s/d 7

Parameter	Keterangan	Contoh Nilai
Minggu		
W	Urutan minggu dlm tahun	Minggu ke 1 tahun
Bulan		
F	Nama Bulan	Januari, Februari,...
m	Urutan dengan angka	01 s/d 12
Tahun		
Y	Empat Digit	2009, 2010
y	Dua digit	09, 10
Waktu		
A	Uppercase AM dan PM	AM, PM
H	Jam format 24	00 s/d 23
i	Menit	00 s/d 59
s	Detik	00 s/d 59

5.3. Ekspresi Reguler

Ekspresi reguler menyediakan mekanisme untuk memanipulasi dan mencocokkan string. Sebagai gambaran, anda bisa memeriksa suatu string yang mengandung karakter, suku kata, atau kata tertentu, atau bahkan diawali dan diakhiri dengan string tertentu.

1. Mengenal fungsi `ereg` pada PHP, tersedia fungsi `ereg` yang dapat digunakan untuk menangani ekspresi reguler.
Bentuk dasar pemanggilan fungsi ini: `ereg(pola, sumber);`
Fungsi ini menghasilkan nilai
 - **True** : Apabila string pada pola cocok dengan string yang terdapat pada sumber
 - **False** : Apabila pola tidak cocok dengan string sumber
2. Memeriksa suatu substring dalam suatu string Fungsi `ereg` dapat digunakan untuk memeriksa suatu substring berada dalam suatu string atau tidak.

Contoh pemanggilan fungsi:

`ereg ("PUTER", "KOMPUTER");` akan menghasilkan nilai **TRUE**, sebab string `"PUTER"` terdapat pada string

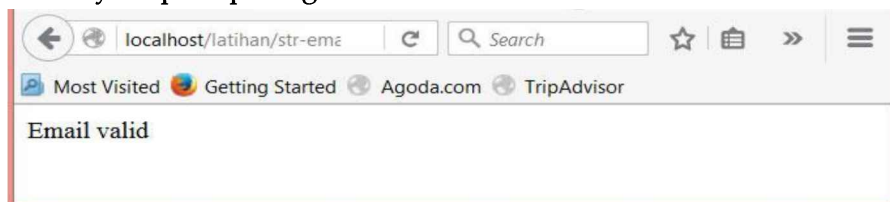
"KOMPUTER". Adapun: `ereg ("PUTER","Komputer");` akan menghasilkan hasil FALSE karena string "PUTER" tidak terdapat pada string "komputer"

Fungsi `ereg` ini paling banyak digunakan dan contoh yang paling banyak dipakai adalah untuk melakukan validasi email, apakah seseorang menulis alamat email dengan benar? Yaitu mengandung karakter @ kemudian . dan lainnya.

```
<?php
$email = "budi@yahoo.com"; // membaca value email dari
string
$pattern = "^.+@.+\.com$";
if (ereg($pattern, $email))
{
    echo "Email valid";
}
else echo "Email tidak valid";
?>
```

Penjelasan kode program di atas adalah `^.+` maksudnya adalah alamat email diawali dengan sebarang karakter dengan jumlah karakter 1 atau lebih. `@` maksudnya adalah dalam alamat email harus memuat simbol @. Setelah tanda @ berikutnya boleh diisi dengan sebarang karakter dengan panjang karakter 1 atau lebih (Lihat simbol `.+`). Berikutnya ada simbol `\.com$`. Simbol ini bermakna bahwa email (yang dianggap valid) harus berakhiran `.com`. Mengapa tanda titik di depan 'com' ini harus diawali dengan tanda slash (`\`)?. Seandainya tidak diberi tanda slash, maka titik ini akan dianggap simbol untuk pattern yang bermakna sebarang karakter tunggal (lihat penjelasan di atas).

Hasilnya seperti pada gambar di bawah ini:



Gambar 5. 5 Fungsi Ereg Untuk Validasi Email

5.4. Fungsi Pada PHP

Fungsi merupakan suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya terpisah dari program yang memanggilnya. Fungsi merupakan elemen utama dalam bahasa PHP karena bahasa PHP sendiri terbentuk dari kumpulan fungsi- fungsi. Fungsi banyak diterapkan dalam program-program PHP yang terstruktur. Keuntungan penggunaan fungsi dalam program yaitu program akan memiliki struktur yang jelas (mempunyai readability yang tinggi) dan juga akan menghindari penulisan bagian program yang sama.

Jenis Fungsi di dalam PHP

Seperti kita ketahui PHP menyediakan banyak fungsi bulit in seperti fungsi date, fungsi time, fungsi string dan fungsi lainnya. Namun dari sekian banyak fungsi ternyata kita juga membutuhkan fungsi-fungsi yang tidak terdapat dalam bawaan php (fungsi built in), jadi mau tidak mau kita harus membuat fungsi sendiri atau istilah nya *user defined function* (UDF). Namun apa sebenarnya tujuan membuat fungsi adalah menghemat pengetikan script php yang berulang-ulang dan menyederhanakan penulisan syntax program. Ketika program sudah mulai bertambah kompleks maka penggunaan fungsi mutak diperlukan sehingga akan mempermudah pekerjaan kita ketika membutuhkan suatu action yaitu cukup dengan memanggil fungsi yang telah kita buat.

Terdapat dua jenis fungsi di dalam PHP, yaitu

1. Fungsi tanpa nilai balik
2. Fungsi dengan nilai balik

Mendefinisikan Fungsi

Bentuk umum pendefinisian fungsi dalam PHP

```
function nama_fungsi(parameter1,...,n) {  
statement2;  
}
```

Perhatikan contoh di bawah ini:

```
function tulis($s){  
echo $s;
```

Pada kode di atas, fungsi di definisikan tanpa nilai balik dengan nama tulis(). Fungsi ini memiliki parameter, yaitu \$. fungsi ini bertugas untuk menampilkan isi dari variabel \$s ke layar. Untuk fungsi tanpa nilai balik, pada umumnya statemen retur tidak disertakan. Jika disertakan, statemen tersebut tidak boleh diikuti dengan ekspresi/nilai tertentu.

Adapun cara pemanggilan fungsi di atas adalah

```
tulis("PHP\n");  
tulis("Pascal\n");  
tulis("Python");
```

Hasil dari program di atas adalah sebagai berikut:

```
-----  
PHP  
Pascal  
Python  
-----
```

Untuk fungsi dengan nilai balik, contoh pendefinisiannya adalah sebagai berikut:

```
<?php  
function kali($a,$b){  
    return ($a * $b);  
}  
$hasil = kali(10,2);  
echo "Hasil Kali: ".$hasil;  
?>
```

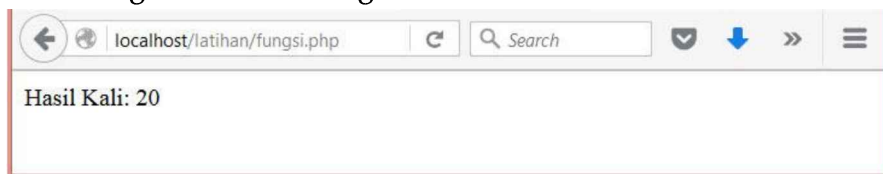
Pada kode di atas, pendefinisian fungsi dengan nama kali(). Fungsi tersebut memiliki dua parameter yang berperan sebagai masukan untuk fungsi, yaitu \$a dan \$b. Tugas dari fungsi tersebut

adalah untuk mengalikan bilangan yang tersimpan di dalam variabel \$a dan \$b. Variabel \$hasil merupakan variabel lokal yang hanya dikenal di dalam badang fungsi.

Cara pemanggilan fungsi di atas adalah sebagai berikut:

```
$hasil1 = kali(10,2);
```

Hasil Program adalah sebagai berikut:



Gambar 5. 6 Membuat Fungsi dengan PHP

Contoh lain mencetak bilangan ganjil

```
<?php
function cetak_ganjil () {
    for ($i=0; $i<100; $i++) {
        if ($i%2 == 1) {
            echo "$i ";
        }
    }
}
//pemanggilan fungsi
cetak_ganjil();
?>
```

5.5. Include dan Require

Fungsi include() dan require() merupakan fungsi yang digunakan untuk menyertakan file php lain ke dalam suatu kode program PHP. Hal sangat membantu proses pemrograman karena tidak perlu menulis ulang program PHP, cukup dengan memanggilnya saja dengan fungsi ini.

Pada dasarnya fungsi `<include()` dan `require()` memiliki kegunaan yang sama, hanya saja pada fungsi `include()`, jika file yang disertakan ternyata tidak ditemukan (karena lokasi yang salah atau memang file tidak ada) maka program atau perintah PHP selanjutnya (setelah `include`) akan tetap dijalankan walaupun ditampilkan error.

Untuk mencobanya, kita harus membuat minimal 2 file di mana 1 file untuk di sertakan (`include/require`) pada satu file lainnya. Kita membuat file `header.php` dan `main.php`, file `header.php` akan dimuat pada kode program `main.php`

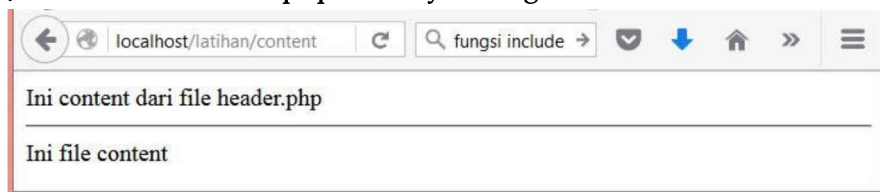
Berikut ini kode program `header.php`

```
<?php
echo "Ini content dari file header.php <br>";
echo "<hr>";
?>
```

Selanjutnya kita membuat file `content.php` sebagai berikut:

```
<?php
include "header.php";
echo "Ini DARI file content";
?>
```

Jalankan file `content.php`, hasilnya sebagai berikut:



Gambar 5. 7 Fungsi Include

Rangkuman

Bahasa pemrograman PHP sudah memiliki fungsi yang lengkap, dari mulai fungsi pengelolaan string sampai dengan fungsi tanggal, selain menyediakan fungsi yang sudah ada, PHP juga menyediakan fasilitas untuk membuat fungsi sendiri.

Jika sebuah file digunakan berulang-ulang, seperti file koneksi ke *database*, file header dan lainnya. Maka buatlah file tersebut terpisah dan jika membutuhkan tinggal memanggil dengan fungsi `include()` atau `require()` sesuai dengan kebutuhan dari program tersebut.

Latihan

1. Buatlah program untuk menghitung jumlah huruf pada kalimat "POLITEKNIK POS INDONESIA" !
2. Buatlah sebuah program untuk menghitung Luas dan Keliling Trapesium!
3. Buatlah sebuah class yang dapat menampilkan biodata dengan menggunakan bahasa pemrograman PHP !

BAB VI

COOKIES, SESSION DAN CLASS

Setelah membaca bab ini, diharapkan mampu memahami dan mengimplementasikan konsep dari Cookies, Session dan Class. Selain dapat mengimplementasikan, juga mampu menerapkan pada studi kasus tertentu (fungsinya).

6.1. Session

Session dapat diartikan sebagai sebuah variabel global yang diciptakan dalam server PHP pada saat sesi awal membuka sebuah web page dan berlaku hingga Anda menutup web tersebut. Session tidak dibatasi oleh konfigurasi browser klien seperti halnya cookies karena tersimpan di server. Session akan otomatis menciptakan sebuah ID unik untuk setiap user pada saat dijalankan.

Kegunaan session adalah untuk menyimpan informasi login yang berlaku hanya dalam satu sesi dan menyimpan catatan order barang dalam transaksi online.

Dengan session memungkinkan programmer menyimpan informasi user secara semi-permanen, artinya selama masa tertentu informasi akan tersimpan. Penyimpanan isi variabel session berada di server, jadi tidak bisa diakses secara langsung oleh client. Dalam aplikasi berbasis web, session banyak digunakan sebagai autentifikasi login, seperti aplikasi email, sosial networking, dll. Dalam penanganan session terdapat beberapa proses yang perlu diperhatikan:

- Proses pembuatan session
- Proses pemeriksaan session
- Proses penghapusan session

Penulisan *script*-nya:

```
session_start() #pembuatan session  
session_destroy() #penghapusan session  
$_SESSION['nama_variable'] #set nama sesssion
```

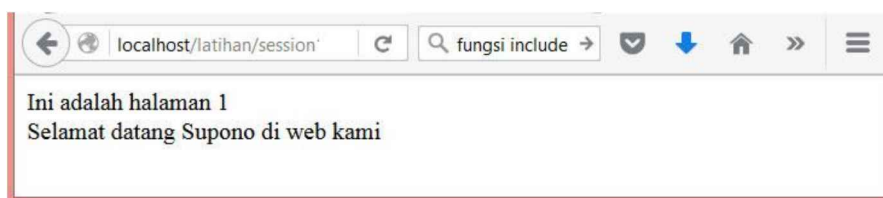
Untuk ilustrasi penggunaan session sebagai berikut:

Buat sebuah file dengan nama session1.php file ini digunakan untuk membuat session

Ini adalah halaman 1


```
<?php  
session_start();  
$_SESSION['name'] = "Supono";  
if ($_SESSION['nama'] != "") {  
    echo "Selamat ".$_SESSION['nama']." datang di  
web kami";  
} else {  
    echo "Maaf, silahkan login terlebih dahulu";  
}  
?>
```

Setelah itu buat file yang kedua dengan nama file session2.php file ini berguna untuk mengecek apakah session dari halaman 1 masih ada atau tidak.



Gambar 6. 1 Menciptakan Session

Berikutnya buat file session2.php untuk mengecek session yang sudah dibuat.

Ini adalah Halaman 2

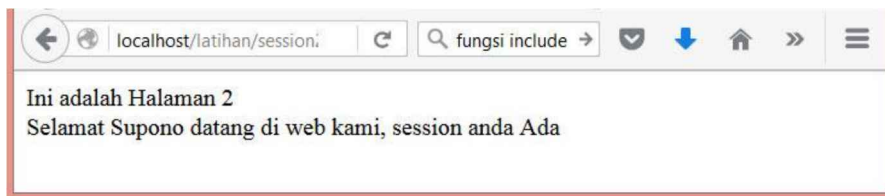

```
<?php  
session_start();
```

```

if ($_SESSION['nama'] != ""){
    echo "Selamat ".$_SESSION['nama']." datang di
web kami, session anda Ada";
} else {
    echo "Maaf, silahkan login terlebih dahulu";
}
?>

```

Anda akses halaman 1 (session1.php) kemudian anda akses halaman 2 (session1.php) maka nama Supono akan tetap muncul pada halaman 2, karena session menyimpan data tersebut.



Gambar 6. 2 Mengecek Session

Berikutnya buat halaman ketiga dengan nama session3.php fungsi halaman ini untuk menghapus session yang dibuat pada halaman 1 (session1.php)

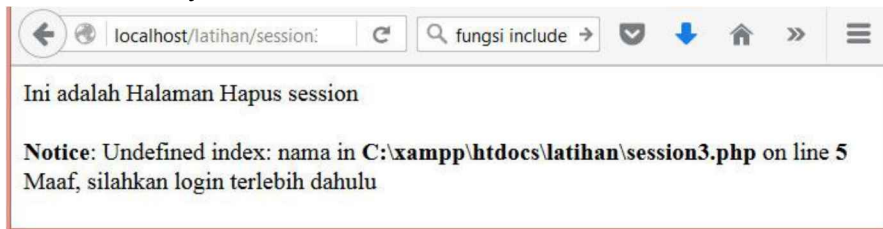
Ini adalah Halaman Hapus session


```

<?php
session_start();
session_destroy();
if ($_SESSION['nama'] != ""){
    echo "Selamat ".$_SESSION['nama']." datang di
web kami, session anda Ada";
} else {
    echo "Maaf, silahkan login terlebih dahulu";
}
?>

```

Berikut hasilnya



Gambar 6. 3 Hapus Session

Pada halaman di atas terlihat bahwa ada error karena index “nama” tidak terdefinisi, artinya index nama tersebut sudah di hapus dari system. Jika Anda mengakses halaman session2.php akan menjumpai error yang sama.

6.2. Cookies

Cookies adalah informasi yang disimpan di komputer klien dalam sebuah file kecil yang diletakkan pada folder tertentu oleh browser. Cookies dapat diakses kapan pun melalui page PHP selama cookies ini masih tersimpan. Hal yang perlu diingat tentang penggunaan cookies bahwa ada beberapa komputer yang tidak menerima akses cookies karena user mematikan fitur cookies tersebut pada browser. Namun, jika cookies ini memang diperlukan kita bisa mengeceknya dengan PHP dan jika statusnya dimatikan pada browser, kita bisa mengaktifkan fungsi cookies ini kembali pada setting-an browser.

Cookies dapat digabungkan dengan session dalam penggunaannya, sehingga keamanan user dalam proses login akan lebih aman karena ada 2 lapis keamanan di sisi client dan di sisi server.

Kegunaan Cookies

Kegunaan dari Cookies adalah sebagai berikut:

1. Menyimpan username dan password login agar pengguna tidak selalu harus mengisikannya pada saat membuka page (halaman web). Biasanya akan muncul checkbox "Remember Me" atau "Ingat Saya Selalu" yang terdapat pada bagian bawah box login.
2. Untuk mencatat konfigurasi yang dilakukan oleh user. Seperti: warna tema, jenis huruf, pilihan bahasa, dll.
3. Untuk mengetahui apakah pengunjung pernah datang atau belum ke halaman yang sedang dibuka (seperti yang digunakan pada hit counter/penghitung jumlah pengunjung pada blog).

Deklarasi Cookies dalam PHP

Cookies harus dideklarasikan sebelum halaman ditampilkan, yang artinya dituliskan sebelum <html>. Secara mendasar untuk membuat sebuah cookies ditulis dengan susunan, sbb:

```
setcookie(name, value, expire, path, domain);
```

Kemudian dalam implementasinya dapat kita tuliskan, sbb:

```
<?php
$nilai = 'Martabak Mesir';

// Cara Pertama
setcookie("Enak", $nilai);

// Cara Kedua
setcookie("Enak", $nilai, time()+3600); /* Berlaku
Satu Jam */

// Cara Ketiga
setcookie("Enak", $nilai, time()+3600, "/Nama
Folder/", "martabakmesirmaknyosselalusihati.com");
?>
```

Dan untuk mengambil nilai dari cookies dapat dilakukan dengan cara, sbb:

```
<?php
$enak = $_COOKIE["Enak"];
echo $enak;
?>
```

Sedangkan, untuk menonaktifkan cookies yang sudah dibuat dapat dilakukan dengan perintah, sbb:

```
<?php
// Mengubah nilai waktu menjadi mundur
setcookie("Enak", "", time()-3600);
?>
```

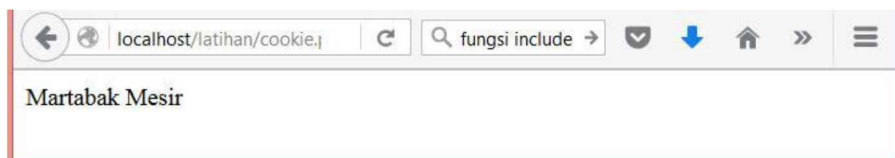
Berikut contoh lengkap programnya menggunakan cara kedua, dengan memberikan masa berlaku cookie adalah 1 jam.

```
<?php
$nilai = 'Martabak Mesir';

// Cara Kedua
setcookie("Enak", $nilai, time()+3600); /* Berlaku
Satu Jam */

$enak = $_COOKIE["Enak"];
echo $enak;
?>
```

Hasilnya seperti gambar di bawah ini, jika kita memanggil `$_COOKIE["enak"]` selama waktu 1 jam ke depan, maka akan selalu menampilkan martabak enak. Jika lebih dari satu jam maka akan terhapus atau error (undefined).



Gambar 6. 4 Cookie

6.3. Class

Class adalah '*cetak biru*' atau '*blueprint*' dari **object**. Class digunakan hanya untuk membuat kerangka dasar. Yang akan kita pakai nantinya adalah hasil cetakan dari class, yakni **object**.

Sebagai analogi, **class** bisa diibaratkan dengan *laptop* atau *notebook*. Kita tahu bahwa *laptop* memiliki ciri-ciri seperti *merk*, memiliki *keyboard*, memiliki *processor*, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah *laptop*. Selain memiliki ciri-ciri, sebuah laptop juga bisa dikenakan tindakan, seperti: *menghidupkan laptop* atau *mematikan laptop*.

Class dalam analogi ini adalah gambaran umum tentang sebuah benda. Di dalam pemrograman nantinya, contoh class seperti: *koneksi_database* dan *profile_user*.

Di dalam PHP, penulisan **class** diawali dengan *keyword class*, kemudian diikuti dengan *nama dari class*. Aturan penulisan nama **class** sama seperti aturan penulisan *variabel* dalam PHP, yakni diawali dengan huruf atau *underscore* untuk karakter pertama, kemudian boleh diikuti dengan huruf, *underscore* atau angka untuk karakter kedua dan selanjutnya. Isi dari **class** berada dalam tanda kurung kurawal.

Berikut adalah contoh penulisan **class** dalam PHP:

```
<?php
class namaClass
{
    Deklarasikan dan definisikan properties di sini
    Definisikan semua method di sini
}
?>
```

Berikut ini contoh kode program dalam pembuatan class

```
<?php
class Mobil
{
    // atribut
```

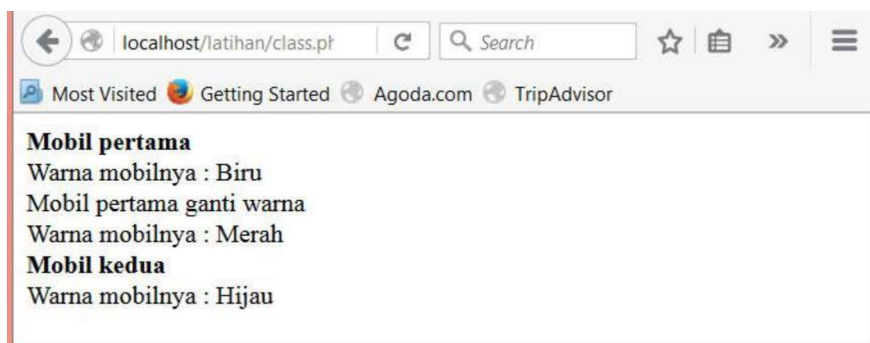
```

var $warna = "Biru";
var $merk = "BMW";
var $harga = "10000000";
// method
function gantiWarna ($warnaBaru)
{
$this->warna = $warnaBaru;
}
function tampilWarna ()
{
echo "Warna mobilnya: " . $this->warna;
}
}

// penggunaan class
$a = new Mobil();
$b = new Mobil();
echo "<b>Mobil pertama</b><br>";
$a->tampilWarna();
echo "<br>Mobil pertama ganti warna<br>";
$a->gantiWarna("Merah");
$a->tampilWarna();
echo "<br><b>Mobil kedua</b><br>";
$b->gantiWarna("Hijau");
$b->tampilWarna();
?>

```

Hasilnya seperti pada gambar di bawah ini:



Gambar 6. 5 Class pada PHP

Rangkuman

Session digunakan untuk menyimpan informasi pengguna di sisi server, biasanya digunakan untuk autentifikasi seseorang untuk login, sedangkan cookies digunakan untuk menyimpan informasi pengguna di sisi client dalam hal ini browser yang dengan menentukan berapa lama informasi tersebut disimpan. Bahasa pemrograman PHP sudah mendukung Pemrograman Berbasis Objek atau OOP dalam hal ini pembuatan class dan sifat-sifat dari pemrograman OOP sudah dapat diimplementasikan.

Latihan

Berikut ini latihan untuk lebih memahami implementasi dari bab ini:

1. Buat sebuah form login kemudian, simpan variabel username ke dalam session dan cookies dengan waktu 5 menit, kemudian masuk ke halaman utama yang memiliki 3 halaman yaitu home, kontak dan logout. Halaman home dan contact untuk mengecek session, sedangkan logout untuk menghapus session dan cookies.
2. Buat sebuah Class dengan nama Apel di mana di dalamnya terdapat method untuk mengubah warna, menentukan rasa dan jumlah apel.

BAB VII

PHP DAN MySQL

Setelah mempelajari bab ini, pembaca diharapkan dapat membuat program membuat *database*, tabel dan dapat melakukan koneksi dengan *database* MySQL dengan kode program PHP serta melakukan manipulasi data dengan PHP.

7.1. MySQL

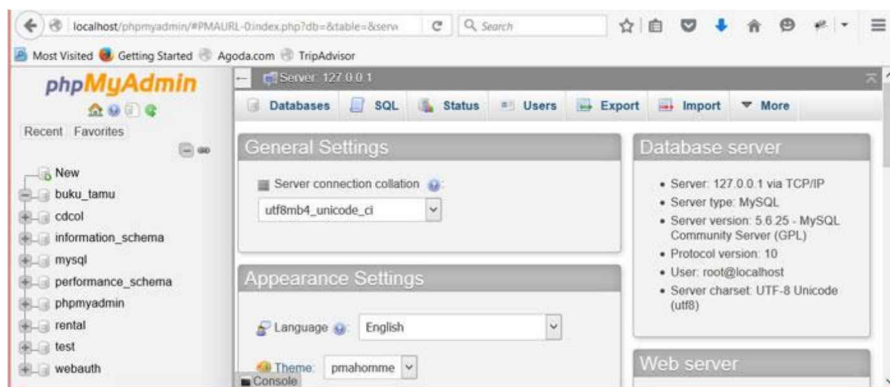
MySQL adalah sistem manajemen *database* SQL yang bersifat Open Source dan paling populer saat ini. Sistem *Database* MySQL mendukung beberapa fitur seperti multithreaded, multi-user, dan SQL *database* manajemen sistem (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, andal dan mudah digunakan.

Kelebihan My SQL adalah sebagai berikut:

1. Source MySQL dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan *database* dapat dilakukan dengan mudah.
4. MySQL merupakan program yang multithreaded, sehingga dapat dipasang pada server yang memiliki multiCPU.
5. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, dsb.
6. Bekerja pada berbagai platform.
7. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
8. Memiliki sistem sekuriti yang cukup baik dengan verifikasi host.
9. Mendukung ODBC untuk sistem operasi Windows.
10. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

MySQL dan PHP merupakan sistem yang saling terintegrasi. Maksudnya adalah pembuatan *database* dengan menggunakan sintak PHP dapat di buat. Sedangkan input yang di masukkan melalui aplikasi web yang menggunakan *script server side* seperti PHP dapat langsung dimasukkan ke *database* MySQL yang ada di server dan tentunya web tersebut berada di sebuah web server.

Pada saat instalasi aplikasi XAMPP *database* MySQL sudah terinstal, jadi kita tinggal menggunakannya (tidak install lagi). Untuk mengeceknya, kita menggunakan aplikasi PHPMyAdmin <http://localhost/phpmyadmin>

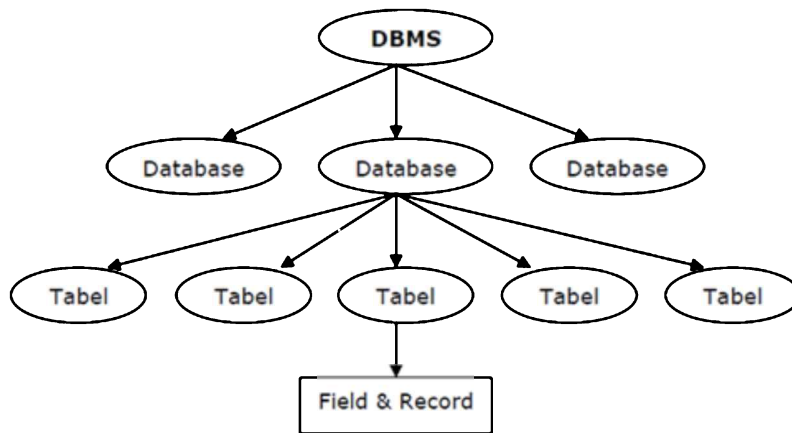


Gambar 7. 1 Tampilan MySQL dengan GUI PhpMyAdmin

7.2. Basisdata

Basis data (atau *database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut

Dalam konsep *database*, urutan atau hierarki *database* sangatlah penting. Urutan atau hierarki *database* digambarkan dalam gambar sbb:

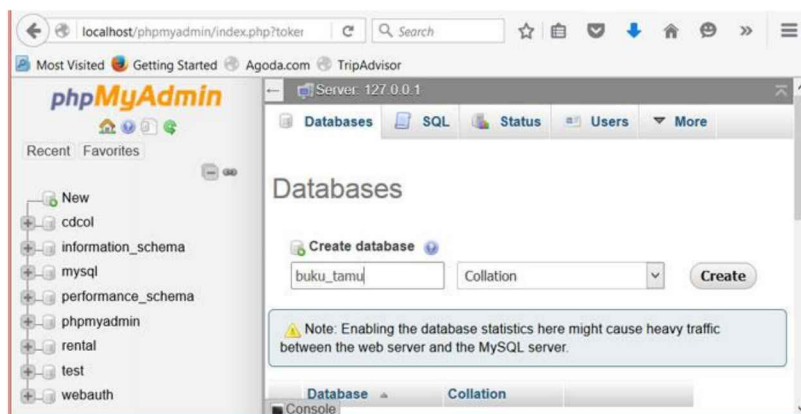


Gambar 7. 2 Konsep Database

7.3. Membuat Database pada PHPMyAdmin

Sebelum membuat *database*, pastikan kita akan membuat aplikasi apa yang berhubungan dengan *database*, setelah itu baru kemudian merancang *database* dan tabel-tabelnya. Pada modul ini kita akan merancang aplikasi “BUKU TAMU” dengan nama *database* “buku_tamu” dengan satu tabel “kontak” yang terdiri dari form input nama, alamat, no telepon, dan keterangan.

Buat *database* dengan nama “buku_tamu” kemudian klik tombol “Create” di PHPMyAdmin seperti gambar di bawah ini:



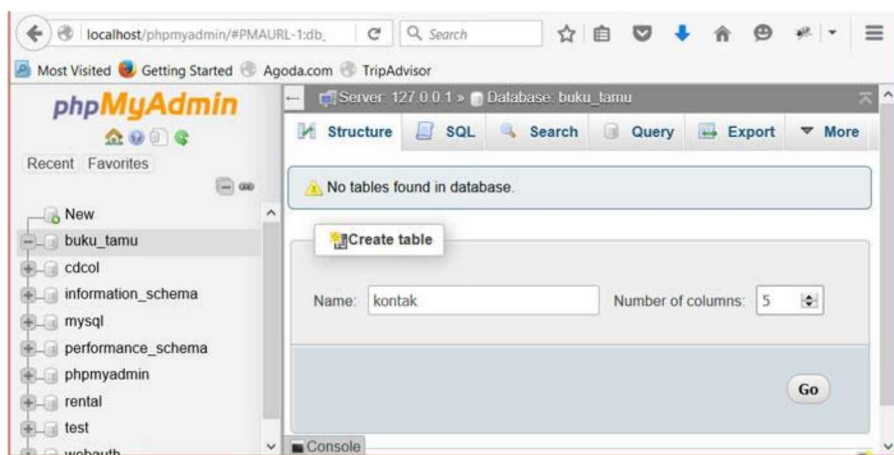
Gambar 7. 3 Membuat Database Buku Tamu

Selanjutnya kita buat perancangan tabelnya, dalam hal ini kita akan membuat satu tabel aja, dengan namanya “kontak” dengan field seperti pada skema di bawah ini:

Tabel 7. 1 Tabel Kontak

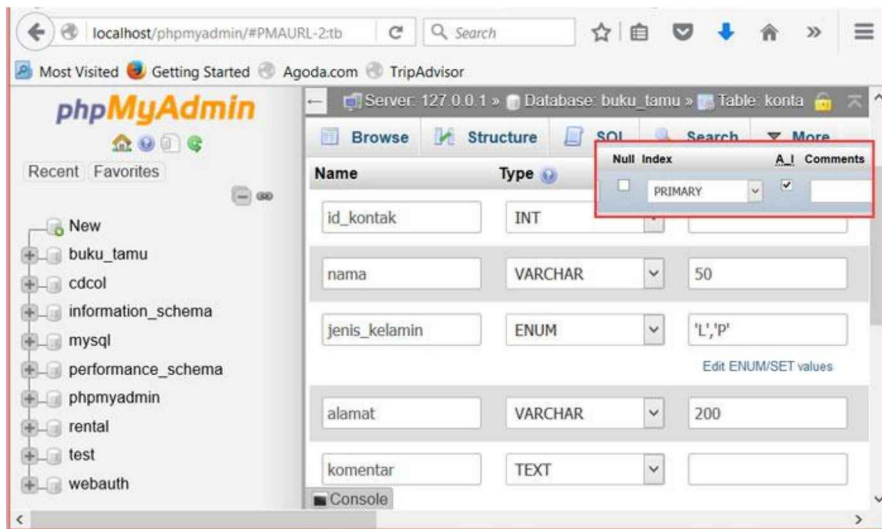
Filed	Type	Length	Keterangan
id_kontak	int	5	Primary Key & Autoincrement
nama	varchar	150	Nama pengisi buku tamu
alamat	varchar	200	Alamat
jenis_kelamin	enum('L','P')		Jenis kelamin
komentar	text		Komentar pengisi buku tamu

Buat tabel dengan nama Kontak dengan jumlah field 5, aplikasikan ke program PHPMyAdmin seperti gambar di bawah ini:



Gambar 7. 4 Membuat Tabel Kontak

Setelah mengisi nama tabel dan jumlah filednya, kemudian klik tombol “GO” selanjutnya kita disuruh untuk mengisi field, tipe data dan length seperti pada gambar di bawah ini:



Gambar 7. 5 Membuat Tabel Kontak

Sampai saat ini kita telah bisa membuat *database* dan tabel di aplikasi PHPMyAdmin. Sekarang saatnya melakukan koneksi ke *database* MySQL dengan PHP dan kemudian mencoba untuk melakukan manipulasi data, seperti tambah data, ubah data dan simpan *database*.

7.4. Koneksi Database ke MySQL dengan PHP

Untuk melakukan koneksi ke *database* MySQL menggunakan *script* PHP maka kita harus tau *host*, *username*, *password* dan nama *database*-nya di MySQL. Simpan dengan nama koneksi.php:

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$dbnm = "buku_tamu";
$conn = mysql_connect ($host, $user, $pass);
if ($conn) {
    echo "Koneksi sukses ke server MySQL<br>";
    $buka = mysql_select_db ($dbnm);
```

```

        if ($buka) {
            echo "Database sudah terpilih<br>";
        } else {
            echo "Database gagal dipilih";
        }
    } else {
        echo "Koneksi ke Server Gagal";
    }
?>

```

Kode program di atas digunakan untuk melakukan koneksi dengan *database* MySQL. Variabel di bawah dapat diubah sesuai kebutuhan.

\$host adalah server host di mana *database* berada, dalam hal ini MySQL ada pada server dengan nama localhost.

\$user adalah user untuk mengakses *database*, secara default ketika install MySQL memberikan user root sebagai user, anda bisa menambahkannya sesuai kebutuhan.

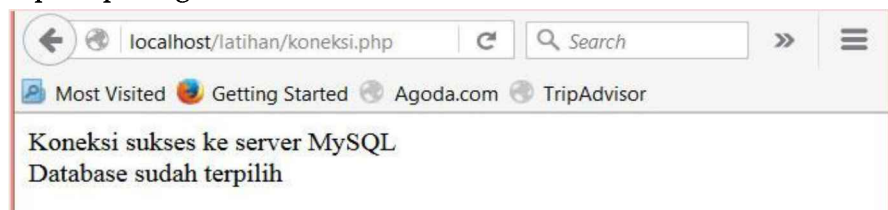
\$pass adalah password untuk mengakses *database*, dalam hal ini kosong (secara default)

\$dbmn adalah variable untuk menyimpan nama *database*.

Mysql_connect merupakan fungsi bawaan PHP untuk melakukan koneksi ke *database* MySQL dengan parameter host, username dan password.

Mysql_select_db merupakan fungsi untuk memilih salah satu *database* yang ada di server MySQL.

Kode program di atas jika dijalankan akan menghasilkan seperti pada gambar di bawah ini:



Gambar 7. 6 Koneksi ke MySQL

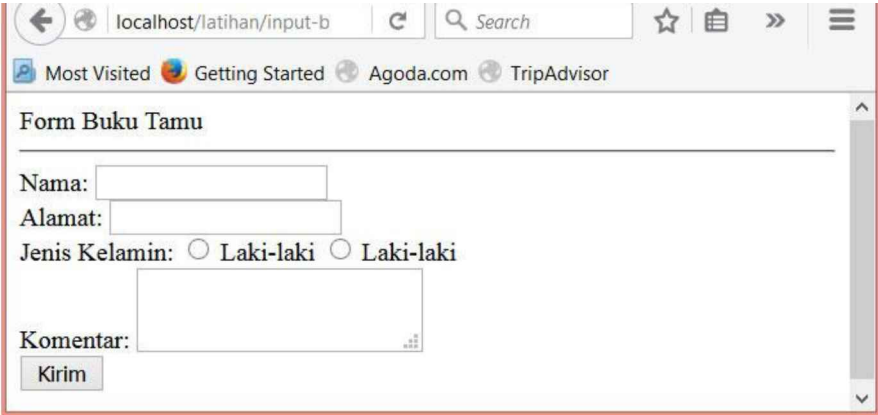
7.5. Membuat Form Input Buku Tamu

Kode program di bawah ini adalah kode program untuk membuat Form Input Buku Tamu. Beri nama file dengan nama `input-buku-tamu.php`.

```
Form Buku Tamu<hr />
<form method="post" action="simpan-buku-tamu.php">
Nama: <input type="text" name="nama" /><br />
Alamat: <input type="text" name="alamat" /><br />
Jenis Kelamin: <input type="radio" name="kelamin"
value="L" /> Laki-laki <input type="radio"
name="kelamin" value="P" /> Perempuan<br />
Komentar: <textarea name="komentar"
cols="20"></textarea><br />
<input type="submit" value="Kirim" />
</form>
```

Kode program di atas merupakan kode program HTML standar untuk menampilkan input form yang nantinya data dari form tersebut akan dikirim ke tabel kontak yang sudah dibuat pada bagian sebelumnya.

Hasilnya seperti gambar di bawah ini:

A screenshot of a web browser window displaying a form titled "Form Buku Tamu". The browser's address bar shows "localhost/latihan/input-b". The form contains the following elements: a text input field for "Nama:", a text input field for "Alamat:", a "Jenis Kelamin:" section with two radio buttons (the first is selected and labeled "Laki-laki", the second is labeled "Perempuan"), a text area for "Komentar:", and a "Kirim" button at the bottom left. The browser's toolbar includes back, forward, search, and other navigation icons.

Gambar 7. 7 Input Buku Tamu

Pada script di atas, data dari form akan dikirim ke file yang bernama `simpan-buku-tamu.php` di bawah ini script untuk

menyimpan buku tamu dan SQL untuk menyimpan buku tamu dari form

```
<?php
// ambil file koneksi
include "koneksi.php";
$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
$jenis_kelamin = $_POST['kelamin'];
$komentar = $_POST['komentar'];
$query = "INSERT INTO kontak VALUE
('','$nama','$alamat','$jenis_kelamin','$komentar')";
$sql = mysql_query($query);
if ($sql){
    echo "Data berhasil disimpan";
} else {
    echo "Data gagal disimpan";
}
?>
```

Berikut ini penjelasan kode program di atas:

```
include "koneksi.php";
```

Digunakan untuk mengambil (load) file koneksi yang sudah dibuat, jadi untuk memasukkan data ke *database* perlu koneksi terlebih dahulu, nah agar kita tidak menulis ulang program koneksi, maka dibuat terpisah agar kalau dibutuhkan tinggal memanggil.

```
$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
```

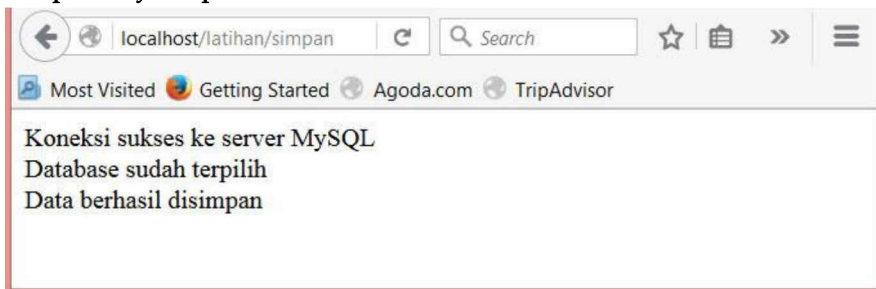
Terdapat variable `$nama` yang digunakan untuk menampung kiriman (POST) dari form dengan nama input "nama". Metode post di sesuaikan dengan method yang ada pada form. Begitupun untuk variable berikutnya yaitu alamat.

```
$query = "INSERT INTO kontak VALUE
('','$nama','$alamat','$jenis_kelamin','$komentar')";
```

Digunakan untuk menyimpan sebuah perintah SQL yang nantinya variable \$query ini akan dikirimkan ke *database* untuk di eksekusi dengan perintah di bawah ini:

```
$sql = mysql_query($query);
```

Melakukan perintah query ke *database*, jika berhasil maka tampilannya seperti di bawah ini:



Gambar 7. 8 Tampilan Hasil Input Buku Tamu

7.6. Menampilkan Data Buku Tamu

Setelah melakukan input data buku tamu ke dalam *database*, maka yang dilakukan selanjutnya adalah menampilkan buku tamu yang telah diinputkan ke dalam halaman website. Kode program untuk menampilkan data buku tamu adalah sebagai berikut.

Buat script di bawah ini dengan nama file tampil-buku-tamu.php

```
<?php
// ambil file koneksi
include "koneksi.php";
$query = "SELECT * FROM kontak";
$sql = mysql_query($query);
while ($row = mysql_fetch_array($sql)){
print $row['nama']." | ".$row['alamat']." | <a
href='hapus-buku-
tamu.php?id=".$row['id_kontak']."'>Hapus</a> <br />";
}
?>
```

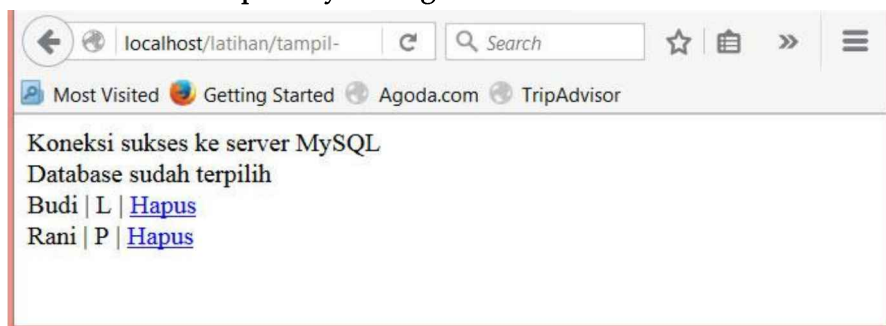
Berikut penjelasan dari kode program di atas:

```
query = "SELECT * FROM kontak";
```

Merupakan SQL untuk menampilkan seluruh data dari tabel kontak.

```
while ($row = mysql_fetch_array($sql)) {
```

Melakukan pengulangan selama data (row) ada pada tabel kontak. Hasil tampilannya sebagai berikut:



Gambar 7. 9 Tampilan Data Buku Tamu

7.7. Menghapus Data Buku Tamu

Untuk menghapus data, di bawah ini adalah cara untuk menghapus data. Buat script di bawah ini dengan nama file hapus-buku-tamu.php

```
<?php
include "koneksi.php";
$kontak_id = $_GET['id'];
$sql = "DELETE FROM kontak WHERE kontak_id =
'$kontak_id'";
$query = mysql_query($sql);
if ($query){
    print "Data buku tamu berhasil dihapus";
} else {
    print "Data buku tamu gagal dihapus";
}
?>
```

7.8. Mengubah Data Buku Tamu

Sebelum mengubah buku tamu, maka kita harus memilih data yang akan dibuat. Oleh karena itu pertama-tama adalah menampilkan data buku tamu beserta link untuk ubah datanya seperti script di bawah ini:

```
<?php
// ambil file koneksi
include "koneksi.php";
$query = "SELECT * FROM kontak";
$sql = mysql_query($query);
while ($row = mysql_fetch_array($sql)){
print $row['nama']." | ".$row['alamat']." | <a
href='form-ubah-buku-
tamu.php?id=".$row['id_kontak']."'>Hapus</a> <br
/>";}
?>
```

Simpan dengan nama file list-ubah-buku-tamu.php

Untuk mengubah buku tamu, perlu dibuat suatu form untuk memfasilitasi kegiatan perubahan data tersebut. Di bawah ini adalah kode program pembuatan form ubah data buku.

```
<body>
<?php
include "koneksi.php";
$id_kontak = $_GET['id_kontak'];
$sql = "SELECT * FROM kontak WHERE id_kontak =
'$id_kontak'";
$res = mysql_query($sql);
$row = mysql_fetch_array($res);
?>
Form Buku Tamu<hr />
<form method="post" action="edit-buku-tamu.php">
ID: <input type="text" name="id_kontak"
value="<?php print $row['id_kontak']?>" /><br />
Nama: <input type="text" name="nama" value="<?php
print $row['nama']?>" /><br />
Alamat: <input type="text" name="alamat"
value="<?php print $row['alamat']?>" /><br />
<input type="submit" value="Kirim" />
```

```
</form>
</body>
</html>
```

Setelah membuat kode program untuk membuat form perubahan data, selanjutnya membuat form untuk melakukan update ke *database* dengan nama file edit-buku-tamu.php Sesuai dengan tujuan form ketika melakukan submit data form. Berikut di bawah ini adalah kode program untuk mengubah data buku tamu.

```
<body>
<?php
include "koneksi.php";
$nama = $_POST['nama'];
$alamat = $_POST['alamat'];
$id_kontak = $_POST['id_kontak'];
$sql = "UPDATE kontak SET nama = '$nama', alamat = '$alamat' WHERE id_kontak = '$id_kontak'";
$res = mysql_query($sql);
if ($res){
    print "Update data sukses";
} else {
    print "Update data gagal";
}
?>
</body>
```

Rangkuman

Bahasa pemrograman PHP mendukung hampir semua *database*, terutama MySQL. Untuk melakukan koneksi ke *database* MySQL dibutuhkan *username*, *password*, *host*, dan nama *database*-nya. Setelah sukses melakukan koneksi ke server MySQL dan memilih salah satu *database*, maka PHP dapat melakukan manipulasi data seperti tambah, ubah, tampil dan hapus dengan perintah SQL.

Latihan

1. Buat script untuk melakukan ubah data buku tamu tersebut di atas.
2. Buatlah menu utama untuk melakukan manipulasi data buku tamu, seperti tambah, hapus, tampil dan ubah. Kemudian di setiap halaman, buat link untuk kembali ke menu utama. Halaman menu utama dari buku tamu tersebut terlihat seperti gambar di bawah ini:



Gambar 7. 10 Menu Buku Tamu

BAB VIII

Pengenalan CodeIgniter

Setelah mempelajari bab ini, pembaca diharapkan mampu mengetahui apa itu Framework CodeIgniter, konsep MVC, kelebihan dan kekurangannya, melakukan instalasi CodeIgniter, dan bagaimana membuat web sederhana menggunakan Framework CodeIgniter.

8.1. Pengantar CodeIgniter

CodeIgniter adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. CodeIgniter memudahkan *developer* atau pengembang web untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuat dari awal. CodeIgniter dirilis pertama kali pada tanggal 28 Februari 2006. Versi terakhir adalah 3.0 (saat penulis membuat buku ini) yang dapat dilihat di website resminya di www.codeigniter.com.

Sebelum membahas lebih jauh tentang salah satu framework PHP ini, mungkin kalian ada yang masih bingung apa itu *Framework*, apa itu MVC (*Model, View, Controller*)? *Framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau class dari awal. Seorang programmer tidak perlu membuat dari awal fungsi-fungsi seperti fungsi koneksi ke *database*, fungsi string dan lainnya.

Ada beberapa alasan mengapa menggunakan Framework:

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola standar yang ada).
3. Umumnya framework menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple *database*, scaffolding, pengaturan session, error handling, dll).
4. Lebih bebas dalam pengembangan jika dibandingkan CMS (*Content Management System*).

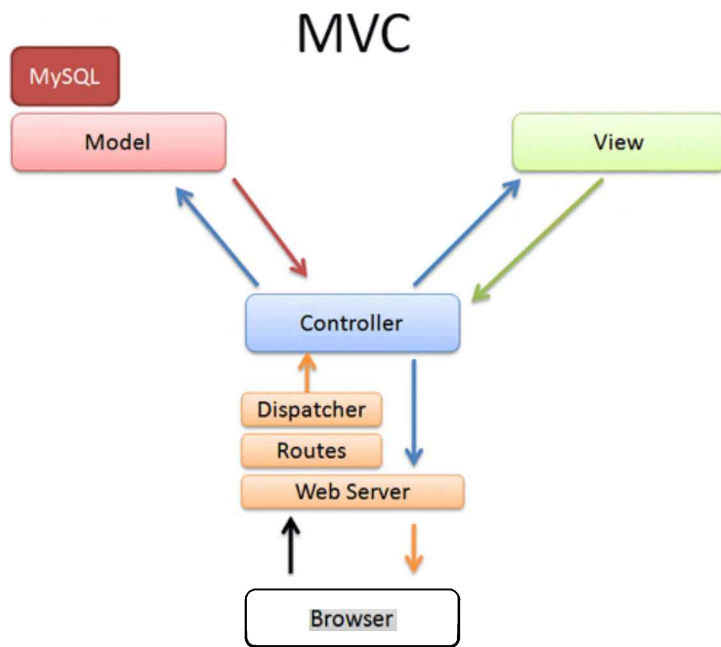
Sekarang masuk ke pengertian MVC (*Model View Controller*).

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi.

Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu:

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.



Gambar 8. 1 Cara Kerja MVC

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developer*-nya, yaitu *programmer* yang menangani bagian model dan *controller*, sedangkan *designer* yang menangani bagian *view*, sehingga penggunaan arsitektur MVC dapat meningkatkan *maintainability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara *programmer* dan *designer* dalam menangani variabel-variabel yang akan ditampilkan.

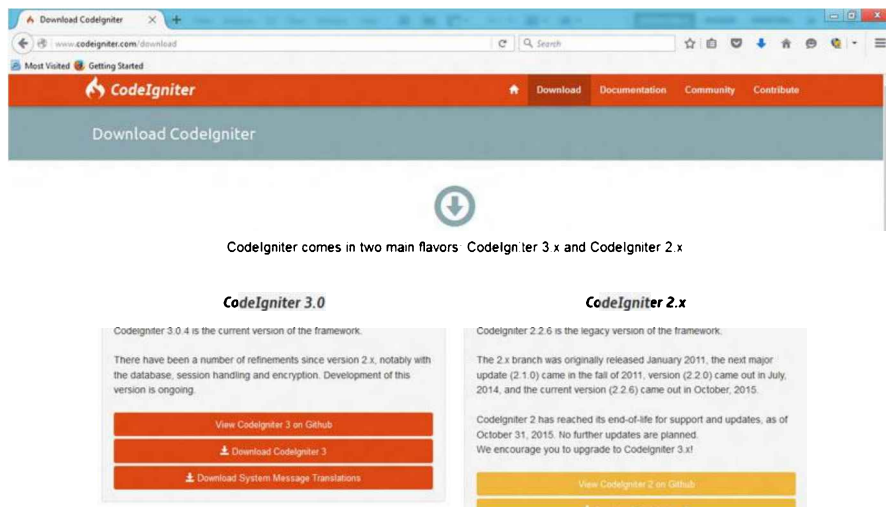
8.2. Kelebihan Framework Codeigniter

Ada beberapa kelebihan CodeIgniter (CI) dibandingkan dengan Framework PHP lain:

1. Performa sangat cepat: salah satu alasan tidak menggunakan framework adalah karena eksekusinya yang lebih lambat daripada PHP from the scratch, tapi CodeIgniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan framework yang paling cepat dibanding framework yang lain.
2. Konfigurasi yang sangat minim (nearly zero configuration): tentu saja untuk menyesuaikan dengan *database* dan keleluasaan routing tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan codeigniter dengan setting standard, anda hanya perlu mengubah sedikit saja file pada folder config.
3. Memakai konsep MVC: Codeigniter memakai konsep MVC (Model View Controller), konsep modern yang banyak dipakai oleh framework PHP lainnya. Dengan adanya MVC, Pengerjaan antara logika dengan layout telah dipisahkan, sehingga antara si programmer dan designer dapat santai melakukan tugasnya.
4. Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
5. Dokumentasi yang sangat lengkap: Setiap paket instalasi codeigniter sudah disertai user guide yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.
6. Dan banyak lagi yang lainnya.

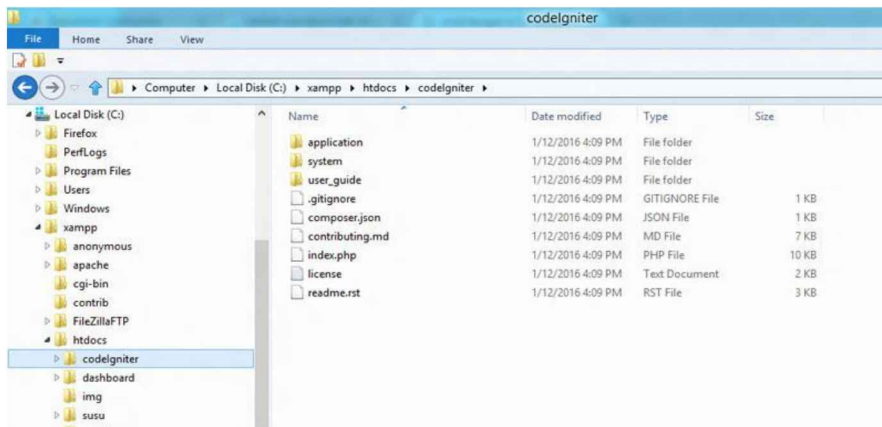
8.3. Instalasi Codeigniter

Sebelum melakukan instalasi, Anda harus men-*download* file instalasi Codeigniter dari website resmi Codeigniter yaitu <http://codeigniter.com/download> kemudian klik menu download dan pilih versi terbarunya.



Gambar 8. 2 Halaman Download Codeigniter

Setelah file download selesai di download, kemudian extract ke folder server root kita (dalam hal ini wamp). Simpan di `c:/wamp/www/` misalnya kita ubah hasil extract kita ke dalam nama folder codeigniter maka hasilnya seperti ini: `c:/wamp/www/codeigniter`.



Gambar 8. 3 Menyimpan Folder Codeigniter

Dalam implementasi tutorial pembuatan buku ini, penulis menggunakan Sistem Operasi Windows 8.1 dan menggunakan XAMPP versi 3.2.1

8.4. Struktur Direktori Codeigniter

Setelah kita mencopykan dan me-rename nama folder, saatnya kita mengenalkan fungsi-fungsi dalam folder dalam folder system:

Tabel 8. 1 Folder Pada Codeigniter

Application	Cache	Menyimpan cache
	Config	File-file konfigurasi
	Controller	Menyimpan file controller
	Core	File core
	Helper	File-file helper
	Hook	Satuan file yang diloat
	Languange	File bahasa
	Libraries	File librari
	Logs	File log
	Models	File-file model
	Third_party	File pihak ketiga / plugin
	View	File view
System	Menyimpan cache	
User Guide	File untuk akses user guide / panduan	

Pada prakteknya kita hanya akan berhubungan pada 3 folder yang diberikan huruf tebal pada tabel di atas, bukan berarti kita tidak membahas lebih detail folder yang lain, tetapi untuk level pemula kita akan berhubungan dengan Model, View dan Controller.

8.5. Konfigurasi Dasar

Di sini akan dijelaskan konfigurasi dasar Codeigniter, di mana file-filenya tersimpan di folder application/config. Di sini akan dibahas mengenai empat file konfigurasi penting yang akan menentukan berjalannya Codeigniter, diantaranya: config.php, database.php, autoload.php dan routes.php akan dijelaskan selanjutnya.

config.php

Berisi konfigurasi dasar aplikasi yang Anda buat. Buka file ini, konfigurasi yang pertama adalah base URL dari aplikasi Anda pada baris 26

```
26 $config['base_url']="http://example.com/";
```

Ganti dengan base url root aplikasi Anda:

```
26 $config['base_url']="http://localhost/codeigniter/";
```

Setelah itu pada baris 26 hilangkan index.php

```
38 $config['index_page'] = "index.php";
```

Hilangkan nama index.php nya seperti script di bawah ini:

```
38 $config['index_page'] = "";
```

Keuntungan menggunakan cara ini (setting url root) adalah kita tidak perlu menulis ulang secara lengkap URL jika diperlukan, misalnya ketika menuliskan *action* pada form, link, URL pada alamat sumber gambar dan kepentingan lainnya yang berhubungan dengan URL.

Setelah mengonfigurasi file `config.php`, anda sudah bisa melihat hasil dari setting kita, silahkan akses aplikasi framework kita di `http://localhost/codeigniter` hasilnya seperti gambar di bawah ini:



Gambar 8. 4 Halaman Jika Sukses Install Codeigniter

database.php

Jika website akan terhubung ke *database*, maka dibutuhkan tambahan konfigurasi yaitu konfigurasi file untuk melakukan koneksi ke *database* dalam hal ini MySQL. Abaikan file ini jika website yang dibangun tidak terhubung dengan *database*. File ini berada di `application/config/database.php`. File ini berisi semua informasi yang diperlukan dalam melakukan setting konfigurasi *database*, seperti *host*, *username*, *password*, jenis *database* yang digunakan, dan lainnya.

```
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'buku_tamu',
82     'dbdriver' => 'mysqli',
    'dbdriver' =>,
```

Pada baris 40 diisi dengan nama *host*-nya, pada baris 82 jika Anda menggunakan SQL Server, maka tulis *dbdriver*nya adalah *mssql*. Codeigniter mendukung beberapa *database*, seperti MySQL,

MS SQL, PostgreSQL, Oracle, SQLite dan ODBC. Setting ini disesuaikan dengan konfigurasi *database* yang ada pada komputer Anda. Untuk praktikum kali ini kita menggunakan *database* dan setting yang pernah kita lakukan di modul sebelumnya.

autoload.php

File ini digunakan untuk menspesifikasikan file apa saja yang akan di load secara otomatis saat user mengakses aplikasi ini. Misalnya yang biasa menggunakan autoload adalah file *database*, sehingga tidak setiap membuat file kita melakukan load file koneksi *database*, maka dengan menempatkan di autoload, maka secara otomatis akan di load pada saat aplikasi diakses. Dalam file ini kita bisa melakukan load terhadap library, helper, plugin, config, language atau model. Autoload ini mirip seperti fungsi include pada modul pemrograman struktural sebelumnya. File autoload ada di application/config/autoload.php misalnya kita akan melakukan autoload file koneksi ke *database* yang sudah kita buat sebelumnya, tinggal ketik saja "*database*" pada array library. Jika ingin melakukan autoload model, berarti cari autoload['model'].

```
61 $autoload['libraries'] = array('database');
```

Pada baris 42 di atas menjelaskan bahwa kita melakukan load file library *database* (setting *database*), sehingga di setiap file yang akan kita buat tidak perlu melakukan pemanggilan fungsi koneksi *database*.

route.php

File ini digunakan untuk melakukan remap URL request agar mengarah ke suatu fungsi tertentu pada sebuah controller, atau dapat diartikan ketika pertama kali akses website (mengakses index.php) controller apa yang akan dieksekusi pertama kali. Secara default, route.php diisi dengan controller welcome. File ini berada di application/config/routes.php

```
53 $route['default_controller'] = 'welcome';
```

Ketika kita mengakses `http://localhost/codeigniter` maka aplikasi akan diarahkan langsung ke controller `welcome` dan secara lengkap URL nya adalah `http://localhost/codeigniter/index.php/welcome`

8.6. Konsep URL pada Codeigniter

Konsep URL pada codeigniter berbeda dengan sistem URL pada pemrograman structural yang dijelaskan pada bab-bab sebelumnya di buku ini, Codeigniter atau disebut CI ini memiliki URL yang unik. Konsep URL yang dirancang Codeigniter ini bertujuan untuk memudahkan search engine dalam mencari content atau isi dan akan lebih mudah dibaca atau userfriendly. Di bawah ini akan diilustrasikan struktur URL dan penerjemahannya per segmen atau bagian.

Contoh: **`http://localhost/codeigniter/index.php/berita/detail/1`**

URL di Codeigniter mempunyai pendekatan MVC (Model View Controller) yang tampak seperti URL di atas, yang jika dipecah akan menjadi beberapa bagian atau segmen seperti di bawah ini:

- Segemen "**`http://localhost/codeigniter/index.php`**" merupakan root aplikasi.
- Segemen URL berikutnya adalah "**`berita`**" yang merupakan Controller (nama Class) yang buat.
- Segemen berikutnya adalah "**`detail`**" merupakan fungsi atau method dari class Controller `berita`.
- Segemen terakhir adalah angka "**`1`**" yang merupakan parameter dari fungsi `berita` yang dilewatkan melalui URL.

8.7. *Controller*

Controller adalah sebuah kelas dengan nama file bebas yang dapat merepresentasikan dengan URL, nama *class* pada *controller*

merupakan nama file tersebut, misalnya nama filenya Admin.php maka nama class-nya adalah “class Admin” Controller juga disebut sebagai logic programming.

Contoh: <http://localhost/codeigniter/index.php/berita>

Pada contoh di atas, URL akan mencari controller berita.php controllernya adalah berita yang mana controller ini akan disimpan di folder codeigniter/application /controllers/berita.php dan harap diingat bahwa nama Class (controller) harus sama dengan nama filenya, seperti contoh penjelasan di atas.

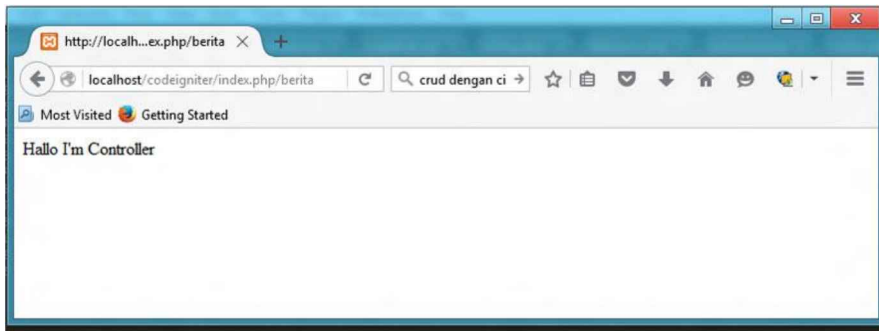
Membuat controller sederhana

Untuk lebih memahami penggunaan controller, buat sebuah file dengan nama berita.php kemudian simpan di folder controller dengan alamat di codeigniter/ application/controllers/ kemudian buka dan isi dengan script di bawah ini:

```
<?php
class Berita extends CI_Controller{
    function index()
    {
        echo "Hallo I'm Controller";
    }
    function detail($nama)
    {
        echo "Hallo Your name is: ".$nama;
    }
}
?>
```

Sekarang buka browser dan akses alamat di bawah ini:

<http://localhost/codeigniter/index.php/berita> jika sukses maka akan tampil kata “**Hallo I'm Controller**”. Secara otomatis ketika mengakses controller berita, maka fungsi yang akan ditampilkan pertama kali adalah fungsi dengan nama index.



Gambar 8. 5 Akses Controller

Penjelasan Script:

1. `class Berita extends CI_Controller{` membuat sebuah class dengan nama Berita (harus sama dengan nama filenya yaitu `berita.php`) kemudian `extends` dari sebuah kelas induk bernama `CI_Controller`.
2. `function index()` membuat sebuah fungsi atau method dalam sebuah class dengan nama `index` yang tidak memiliki parameter di dalamnya. Jika nama fungsinya `index`, maka secara otomatis jika mengakses controller ini, maka yang pertama kali di eksekusi adalah fungsi `index`. Untuk mengaksesnya cukup sampai dengan `berita localhost/codeigniter/berita`
3. `echo "Halo I'm Controller";` untuk menampilkan pada browser.
4. `function detail($nama)` membuat fungsi bernama `detail` yang mempunyai satu parameter. Fungsi ini harus dipanggil lengkap seperti ini: `localhost/codeigniter/berita/detail/budi` (`detail` adalah nama fungsinya dan `budi` adalah nilai parameternya).

Class constructor

Tujuan membuat *constructor* adalah mengoverriding constructor yang ada pada *parent class* (class induk *constructor*), sehingga kita harus melakukannya secara manual. Ada perbedaan

pembuatan *constructor* di antara PHP 4 dan PHP 5. Jika pada PHP 4 nama *constructor* adalah nama *class*-nya, sedangkan pada PHP 5 nama *constructor* adalah `__construct()`. Berikut ini contoh *class constructor* pada PHP 5 (penulis mempertimbangkan bahwa PHP 4 sudah jarang digunakan).

```
<?php
class Berita extends CI_Controller{
function __construct()
{
    parent::__construct();
    // Your own constructor code
}
function index()
{
    echo "Hallo I'm Controller";
}
}
?>
```

8.8. View

View adalah halaman *presentation* atau halaman desain HTML-nya, di mana di *view*-lah data akan ditampilkan, *view* ini akan menerima *variable* dari *controller* kemudian ditampilkan di *view*. Cara kerjanya adalah *controller* akan menunjuk *view* mana yang akan dikirim data dari *controller* dengan fungsi `$this->load->view("nama_view")`; nama *view* adalah nama file berekstensi php (.php) yang disimpan pada folder *application/ view/*. Di bawah ini adalah penggunaan *view* dan *contoller*, *contoller* yang tadi kita ubah seperti di bawah ini:

```
<?php
class Berita extends CI_Controller{
function index()
{
    $this->load->view('hello');
}
function detail($nama)
{
    $data['is_name'] = $nama;
    $this->load->view('hasil', $data);
}
}
?>
```

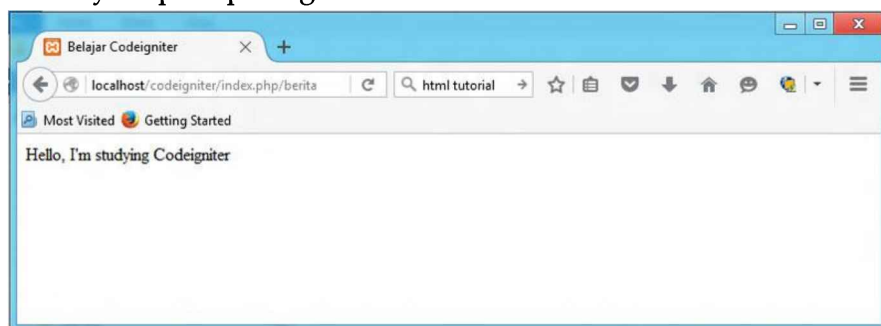
Ada dua penulisan yang berbeda pemanggilan view oleh controller pada script di atas, yaitu “`$this->load->view('hello');`” dan `$this->load->view('hasil', $data);` jika terdapat `$data` di belakangnya berarti controller selain memanggil view, juga mengirimkan data ke view tersebut. Pada contoh di atas data yang dikirim adalah `$data['is_nama'] = $nama`.

Selanjutnya kita akan buat file view-nya yang akan disimpan di folder view, yaitu di `codeigniter/application/views/` dengan nama sesuai dengan isi dari load view di atas, yaitu `hallo.php` dan `hasil.php`

Berikut ini file `hallo.php`

```
<!DOCTYPE html>
<html>
<head>
<title>Belajar Codeigniter</title>
</head>
<body>
    Hello, I'm studying Codeigniter
</body>
</html>
```

Hasilnya seperti pada gambar di bawah ini:



Gambar 8. 6 Pembuat View tanpa mengirimkan data

Jika view tersebut menerima pengiriman data dari controller, maka buatlah script php pada view tersebut seperti di bawah ini:

```
<body>
    Hai <?php echo $is_name?>, apa kabar
</body>
</html>
```

Terlihat bahwa hanya isi array-nya saja yang dituliskan pada *view* tersebut.

Sampai di sini kita telah bisa membuat *Controller*, kemudian variabel yang ada di *Controller* dikirim ke *View*. Pada controller kita bisa melakukan proses programming *logic* atau perhitungan yang hasilnya akan dikirimkan ke *views* dalam bentuk array variable. Untuk membuat model, dibutuhkan pembuatan *database* dan table, karena model ini menghubungkan controller dengan *database*. Pembuatan model akan dibahas pada bab berikutnya yaitu Codeigniter koneksi dengan *database* MySQL.

Rangkuman/Kesimpulan

Codeigniter adalah sebuah Framework PHP dengan konsep MVC, di mana model hubungannya dengan *database*, *view* hubungannya dengan tampilan HTML dan *Controller* hubungannya dengan *logic programming*. Untuk membuat sebuah halaman website menggunakan Codeigniter, diperlukan minimal satu *controller* dan satu *view*. Jika berhubungan dengan *database* maka dibutuhkan model.

Tugas/Latihan

Buat sebuah *controller* bernama kontak, di mana pada *controller* tersebut memiliki 3 fungsi yang dengan spesifikasi sebagai berikut:

1. Fungsi *index*
Tidak memiliki parameter, halaman ini berisi informasi alamat email dan kontak anda.
2. Fungsi *tampil*
Memiliki satu parameter, yaitu nomor telepon. Fungsi ini selain melakukan *load view* juga mengirimkan data nomor telepon, alamat dan email yang masing-masing memiliki variabel dan ditampilkan pada *view*.
3. Fungsi *download*
Tidak memiliki parameter, halaman ini berisi *list file* untuk di-*download*.

BAB IX

CODEIGNITER DENGAN MySQL

Setelah mempelajari bab ini, pembaca diharapkan mampu merancang sebuah system informasi sederhana, membuat *database*, membuat tabel serta merelaksikannya dengan *database* MySQL dan program bantu phpMyAdmin, membuat model pada Codeigniter, menghubungkan model dan menampilkan data menggunakan Codeigniter, serta dapat membuat fungsi CRUD (Create, Read, Update, dan Delete) data.

9.1. Menentukan Kebutuhan Sistem

Pada bagian ini akan mendefinisikan kebutuhan system dari sebuah aplikasi yang akan dibangun menggunakan Codeigniter dan *database* MySQL. Aplikasi yang akan dibuat adalah Sistem Informasi Rental Mobil PT. Alim Rugi, berikut ini wawancara seorang analis dengan pemilih rental mobil PT. Alim Rugi dengan nama Tukimin.

Analisis : Pak Tukikin, sebagaimana yang sudah dijadwalkan sebelumnya bahwa hari ini kita akan melakukan wawancara terkait kebutuhan aplikasi yang akan bapak bangun untuk rental mobil bapak. Ada beberapa pertanyaan yang akan saya ajukan ke Bapak, dan tolong kerja samanya Bapak memberikan informasi yang benar dan bersama dengan datanya.

Tukimin : OK, silahkan. Untuk data akan disiapkan oleh asisten saya.

Analisis : Apa tujuan dan harapan yang bapak capai dengan sistem informasi rental mobil ini?

Tukimin : Saya menginginkan sistem informasi yang dibuat mampu menampilkan informasi siapa saja yang

meminjam, tanggal berapa sampai dengan tanggal berapa dan jenis mobil apa yang dipinjam. Selain itu saya juga menginginkan biodata dari tiap peminjam tersimpan secara komputerisasi dan sewaktu-waktu bias di print. Dan juga saya ingin ada fungsi pencarian data mobil apakah sedang dipinjam atau ada.

Analisis : Seperti apa proses peminjaman mobil di perusahaan bapak saat ini?

Tukimin : Untuk saat ini masih dicatat di buku besar. Prosesnya seperti ini, peminjam datang ke tempat rental kami, kemudian menanyakan apakah ada mobil kosong, kalau ada mereka akan mendaftarkan diri dengan memberikan KTP asli, setelah itu melakukan transaksi DP dan mobil dibawa oleh peminjam.

Analisis : Kalau proses pengembaliannya bagaimana pak?

Tukimin : Langsung mengembalikan dengan menukarkan kunci dengan kartu identitas yang ditinggal pada saat peminjaman.

Analisis : Bagaimana kalau terlambat mengembalikannya?

Tukimin : Akan dikenakan denda per hari keterlambatan sebesar 100 ribu rupiah. Tetapi untuk sistem informasi yang akan dibangun tidak perlu ada fitur denda, cukup dapat melakukan pencarian dan rekap data peminjam, mobil dan transaksi.

Analisis : Alhamdulillah, terima kasih pak.

Tukimin : Tapi saya ingin data aman, artinya tidak sembarangan orang bisa akses aplikasi ini.

Analisis : Siap pak, bisa menggunakan hak akses.

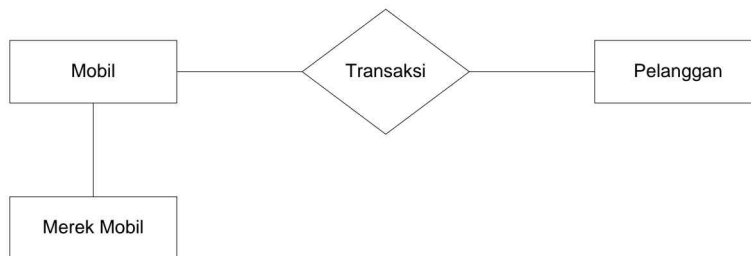
Selanjutnya adalah membuat rekap dari hasil pembicaraan di atas, di mana kurang lebih seperti berikut:

1. Tujuan pembuatan aplikasi adalah membuat kelola data peminjam, mobil dan transaksi.

2. Memiliki fitur pencarian data mobil yang menghasilkan mobil tersebut sedang dipinjam atau tersedia.
3. Membuat aplikasi transaksi peminjaman dan pengembalian mobil, tanpa ada system denda.
4. Mempunyai fitur login.

9.2. Desain Database

Berikut ini adalah desain *database* berdasarkan data pada sub bab sebelumnya. Nama *database* yang akan dibangun adalah “rental”.



Gambar 9. 1 Hubungan Antar Tabel

Dengan table-table sebagai berikut:

1. Tabel Mobil

Tabel 9. 1 Tabel Mobil

Filed	Type	Length	Note
Nomor_kendaraan	Varchar	15	Primary Key
Nomor_mesin	Varchar	30	
Id_merek_mobil	Int		FK dari table merek
Tahun_beli	Int		
Nama_mobil	Varchar	30	

Keterangan:

- *Nomor_kendaraan*: berisi nomor kendaraan mobil, di mana akan dibuat Primary Key dan bersifat unik, artinya tidak

ada nomor kendaraan yang sama. Contoh datanya adalah: Z 4589 FY

- *Nomor Mesin*: berisi informasi nomor rangka mesin dari kendaraan tersebut.
- *Id_merek_mobil*: berisi id merek mobil yang merupakan foreign key dari table lain.
- *Tahun_beli*: tahun beli dari kendaraan tersebut, biasanya peminjam menanyakan tahun keluaran mobil tersebut apakah mobil baru atau lama.
- *Nama_mobil*: berisi informasi nama mobilnya, misalnya: Toyota Avanza

2. Merek Merek

Tabel 9. 2 Tabel Merk

Filed	Type	Length	Note
Id_merek	Int		Primary Key, Autoincrement
Nama_merek	Varchar	50	

Keterangan:

- *Id_merek*: berisi informasi id merek yang dari mobil tertentu yang dibuat auto increment artinya setiap memasukan data baru maka akan bertambah otomatis datanya. Misalnya 1, 2, 3, 4, dan seterusnya.
- *Nama_merek*: berisi informasi merek dari mobil yang umum dijumpai, misalnya: Honda, Toyota, Mitsubishi dan lain-lain.

3. Tabel Pelanggan

Tabel 9. 3 Tabel Pelanggan

Filed	Type	Length	Note
Id_pelanggan	Int		Primary Key, Autoincrement
Nama_pelanggan	Varchar	100	
Jenis_kelamin	Enum	"L", "P"	
Tempat_lahir	Varchar	50	

Filed	Type	Length	Note
Tanggal_lahir	Date		
Nomor_identitas	Varchar	50	
Alamat	Varchar	225	
Nomor_telp	Varchar	50	

Keterangan:

- *Id_pelanggan*: berisi informasi id pelanggan yang dibuat auto increment artinya setiap memasukan data baru maka akan bertambah otomatis datanya. Misalnya 1, 2, 3, 4, dan seterusnya.
- *Nama_pelanggan*: berisi informasi nama pelanggan.
- *Jenis_kelamin*: berisi jenis kelamin dengan tipe data Enum, artinya hanya bisa memilih L (laki-laki) dan P (perempuan).
- *Tempat_lahir*: berisi informasi tempat lahir pelanggan.
- *Tanggal_lahir*: berisi informasi tanggal lahir.
- *Nomor_identitas*: berisi informasi nomor identitas dari pelanggan, dapat bisa berisi nomor KTP, Kartu Pelajar, nomor SIM atau kartu identitas lainnya.
- *Alamat*: berisi informasi alamat dari pelanggan
- *Nomor_telp*: berisi informasi kontak dari pelanggan.

4. Tabel Transaksi

Tabel 9. 4 Tabel Transaksi

Filed	Type	Length	Note
Id_transaksi	Int		Primary Key, Autoincrement
Nomor_kendaraan	Varchar	15	
Id_pelanggan	Int		
Tanggal_pinjam	Date Time		
Lama_pinjam	Int		
Tanggal_kembali	Date time		
Status	Enum	"K", "B"	Default "B"

Keterangan:

- *Id_transaksi*: berisi id transaksi yang dibuat secara auto incremen.
- *Nomor_kendaraan*: merupakan Foreign Key dari table mobil.
- *Id_pelanggan*: berisi id pelanggan yang merupakan foreign key dari table pelanggan
- *Tanggal_pinjam*: berisi tanggal pinjam mobil
- *Lama_pinjam*: berisi data lama pinjam dalam hitungan hari.
- *Tanggal_kembali*: berisi data kembali mobil yang dipinjam.
- *Status*: berisi informasi apakah mobil tersebut sudah kembali atau masih dipinjam. "K" artinya "Kembali" dan "B" artinya "Belum kembali".

5. Tabel user

Tabel 9. 5 Tabel User

Filed	Type	Length	Note
Id_user	Int		Primary Key, Autoincrement
Username	Varchar	32	
Password	Varchar	32	
Nama	Varchar	50	

Keterangan:

- *Username*: berisi id username untuk user yang akan login ke system.
- *Password*: berisi data password yang dienkripsi menggunakan md5 sehingga tidak terlihat secara jelas pada table user.

9.3. Membuat Database

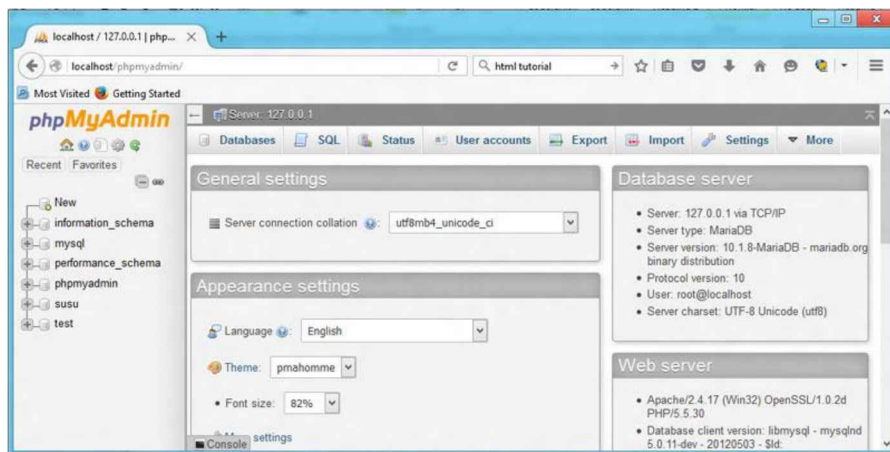
MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi

di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Beberapa kelebihan MySQL antara lain:

- Free (bebas di-download)
- Stabil dan tangguh
- Fleksibel dengan berbagai pemrograman
- Security yang baik
- Dukungan dari banyak komunitas
- Kemudahan management *database*
- Mendukung transaksi
- Perkembangan software yang cukup cepat.

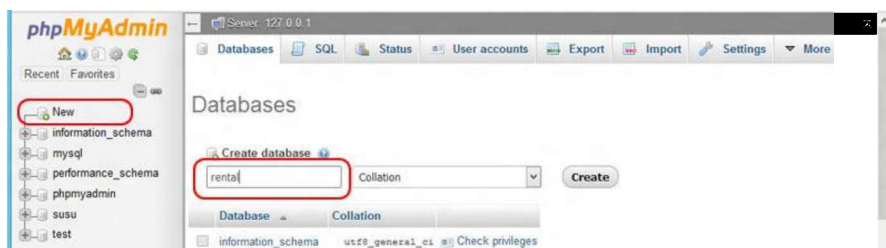
Pada buku ini, penggunaan *database* MySQL menggunakan phpMyAdmin yang bisa diakses langsung dengan mengetikkan localhost/phpmyadmin pada URL browser kita.



Gambar 9. 2 Halaman PHPMyAdmin

Untuk membuat *database* baru, klik link “New” pada bagian jendela kiri phpMyAdmin kemudian akan muncul pada bagian

kanan “Create Database”, kemudian isikan nama *database* yang akan kita buat, yaitu “Rental” seperti pada gambar di bawah ini:



Gambar 9. 3 Membuat Database Baru

9.4. Membuat Tabel

Pada kasus di atas kita akan membuat sebanyak 5 tabel, dengan rincian 4 tabel akan dibuat storage engine dengan tipe “InnoDB” yang mendukung fitur “Referential Integrity” yang artinya akan betul-betul menjaga konsistensi data dalam table *database*. Konsistensi yang dimaksud adalah hubungannya dengan melakukan transaksi seperti hapus dan ubah data pada table tersebut. Sedangkan table user akan di-*setting* store engine tipenya adalah “MyISAM”, karena tidak memerlukan penggunaan fitur Referential Integrity.

1. Membuat tabel Mobil

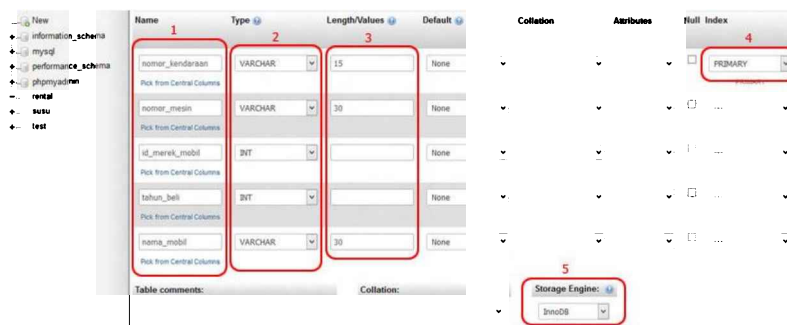
Untuk membuat tabel pada phpMyAdmin, caranya sebagai berikut:

- Klik nama *database* yang sudah kita buat pada bagian kiri, maka akan muncul di sebelah kanan “Create table”.
- Isikan nama tabelnya yaitu tabel “mobil”.
- Kemudian isi pada Number of columns 5 sesuai dengan desain *database* yang sudah kita buat pada sub bab sebelumnya.
- Klik tombol “GO” pada bagian kanan seperti pada gambar berikut ini



Gambar 9. 4 Membuat Tabel Baru

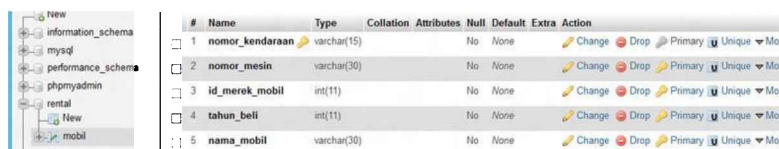
- Yang berikutnya menentukan nama field/kolom dari tabel.



Gambar 9. 5 Form Pengisian Spesifik Tabel

Isikan nama field pada bagian nomor 1 pada gambar, kemudian beri tipe data sesuai dengan yang direncanakan sesuai dengan nomor 2, kemudian lebar dari tipe data pada nomor 3, setelah itu berikan tambahan “Primary Key” pada nomor 4, serta jangan lupa setting store engine dengan tipe “InnoDB” seperti pada nomor 5 pada gambar 10 di atas.

- Jika pembuatan tabel sesuai dengan perancangan yang sudah dibuat pada sub bab sebelumnya, klik tombol “Save” pada bagian bawah halaman.



Gambar 9. 6 Tabel Mobil Jadi

2. Tabel Merek

Untuk membuat tabel baru, lakukan seperti langkah-langkah di bawah ini:

- Klik nama *database* “rental” pada bagian kiri halaman phpMyAdmin. Setelah itu muncul “Create table” pada halaman bagian kanan.
- Isikan nama tabel yang dimaksud, yaitu: “merek” dan jumlah kolom sebanyak 2.
- Pada tabel merek ada keterangan “Auto Increment” pada kolom id_merek, untuk melakukannya check pada bagian A_I sebelah kanan “Primary Key” seperti pada gambar di bawah ini:



Gambar 9. 7 Membuat Tabel Merek

- Hasilnya akan terlihat seperti pada gambar di bawah ini:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
☐	1	id_merek	int(11)		No	None	AUTO_INCREMENT	Change Drop Primary
☐	2	nama_merek	varchar(50)		No	None		Change Drop Primary

Gambar 9. 8 Hasil Tabel Merek

3. Tabel Pelanggan

Sama seperti pada pembuatan tabel sebelumnya, untuk membuat tabel baru klik terlebih dahulu nama *database* sebelah kiri, kemudian isikan nama tabel dan jumlah kolom. Pada tabel pelanggan terdapat tipe data “Enum”, berikut ini cara mengisinya:

Pick from Central Columns

jenis_kelamin ENUM "L", "P" None

Pick from Central Columns Edit ENUM/SET values

tempat_lahir INT 50 None

Gambar 9. 9 Membuat Tipe data Enum

Berikut ini hasil dari pembuatan tabel pelanggan

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
☐ 1	id_pelanggan	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
☐ 2	nama_pelanggan	varchar(100)			No	None		Change Drop Primary
☐ 3	jenis_kelamin	enum('L', 'P')			No	None		Change Drop Primary
☐ 4	tempat_lahir	int(50)			No	None		Change Drop Primary
☐ 5	tanggal_lahir	date			No	None		Change Drop Primary
☐ 6	nomor_identitas	varchar(50)			No	None		Change Drop Primary
☐ 7	alamat	varchar(225)			No	None		Change Drop Primary
☐ 8	nomor_telp	varchar(50)			No	None		Change Drop Primary

Gambar 9. 10 Hasil Tabel Pelanggan

4. Tabel Transaksi

Untuk membuat tabel transaksi, perintahnya sama seperti yang dilakukan pada bagian sebelumnya. Berikut ini hasil dari pembuatan tabel transaksi.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
☐ 1	id_transaksi	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
☐ 2	id_pelanggan	int(11)			No	None		Change Drop Primary
☐ 3	nomor_kendaraan	varchar(15)			No	None		Change Drop Primary
☐ 4	tanggal_pinjam	datetime			No	None		Change Drop Primary
☐ 5	lama_pinjam	int(11)			No	None		Change Drop Primary
☐ 6	tanggal_kembali	datetime			No	None		Change Drop Primary
☐ 7	status	enum('K', 'B')			No	B		Change Drop Primary

Gambar 9. 11 Hasil Tabel Transaksi

5. Tabel User

Untuk tabel user harap diingat bahwa store engine menggunakan MyISAM. Berikut hasil dari tabel User.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
☐	1	id_user	int(11)		No	None	AUTO_INCREMENT	Change Drop Primary Unique
☐	2	username	varchar(32)		No	None		Change Drop Primary Unique
☐	3	password	varchar(32)		No	None		Change Drop Primary Unique
☐	4	nama	varchar(50)		No	None		Change Drop Primary Unique

Gambar 9. 12 Hasil dari Tabel User

9.5. Menerapkan Konsep Relasi Antartabel

Pada bagian sebelumnya sudah dibahas (desain *database*) bahwa ada beberapa tabel yang saling berelasi antara satu tabel dengan tabel yang lainnya yang dikenal dengan Referential Integrity Constrain. Dalam sebuah RDBMS disebut dengan “Foreign Key” agar dapat saling terhubung satu tabel dengan tabel yang lainnya.

Tujuan dari menerapkan relasi tabel ini agar konsistensi data akan tetap terjaga. Contoh kasus yang menjadi inkonsistensi data (tidak konsisten data) seperti kasus berikut: Suatu ketika Amin memiliki id_pelanggan 212 yang sudah melakukan transaksi peminjaman pada tanggal tertentu. Karena ada sesuatu hal, id_pelanggan atas nama Amin diubah menjadi 350. Karena tidak dibuatkan relasinya, sehingga di tabel transaksi id_pelanggan 212 akan tetap ada dan tidak berubah menjadi 350. Dengan adanya system referential integrity akan memungkinkan jika data master berubah, maka secara otomatis data pada tabel lain pun akan berubah, termasuk apabila dihapus.

Pada desain *database*, beberapa tabel yang berelasi adalah sebagai berikut:

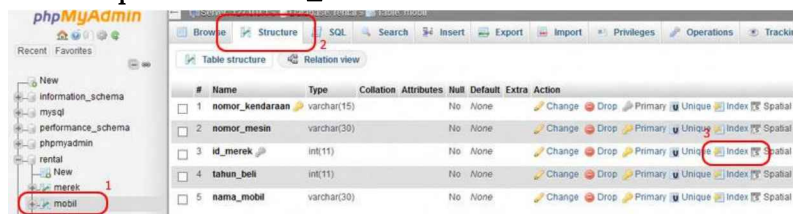
- Mobil dengan Merek,
- Mobil dengan Transaksi, dan
- Transaksi dengan Pelanggan.

Berikut ini cara membuat relasi menggunakan phpMyAdmin dengan mengaktifkan referential Integrity Constrains.

1. Mobil dengan Merek

Berikut ini tahapan bagaimana membuat relasinya

- Pilih tabel “mobil” pada *database* “rental” seperti pada nomor 1 pada gambar di bawah.
- Pilih tab “Structure” pada bagian atas halaman phpMyAdmin seperti pada nomor 2 pada gambar di bawah.
- Pada kolom “Action” pilih “index” seperti pada nomor 3 pada gambar di bawah. Fungsi index ini untuk memberikan index kepada kolom id_merek.



Gambar 9. 13 Pembuatan Relasi Mobil dengan Merek

- Selanjutnya pilih menu “Strukture” pada bagian “Relationship View” yang ada pada bagian bawah kanan menu Struktur menu tas (lihat gambar sebelumnya). Kemudian pada bagian “Foreign key constraints”.
- Pada bagian “Column” pilih “id_merek” pilih rental dan pada bagian kanannya pilih “id_merek” yang artinya tabel mobil berhubungan dengan tabel merek spesifik pada kolom id_merek. Pilih pada bagian kanan “ON DELETE” dan “ON UPDATE” dipilih “CASCADE”.
- Kemudian klik tombol “Save” pada bagian bawah.



Gambar 9. 14 Relation View Mobil dengan Merek

Fungsi dari ON DELETE dan ON UPDATE CASCADE adalah apabila data pada tabel induk dihapus, maka data pada tabel anak akan ikut terhapus. Demikian juga apabila tabel induk diubah/update maka data pada tabel anaknya akan diubah/update.

2. Transaksi dengan Mobil

- Pilih tabel transaksi, lakukan seperti pada bagian di atas.



Gambar 9. 15 Relationship antara Mobil dan Transaksi

- Setelah itu klik tombol “Save” pada bagian bawah.

3. Transaksi dengan Pelanggan

- Pilih tabel transaksi, kemudian lakukan seperti pada bagian sebelumnya.

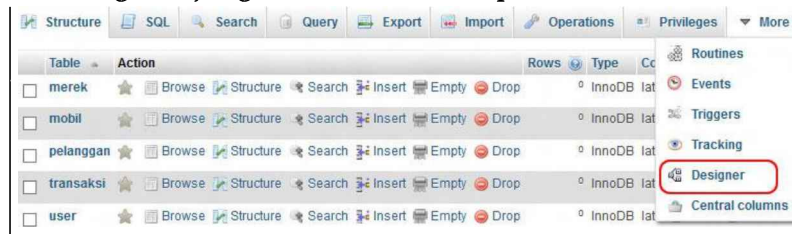


Gambar 9. 16 Relationship Transaksi dengan Pelanggan

- Klik tombol “Save”.

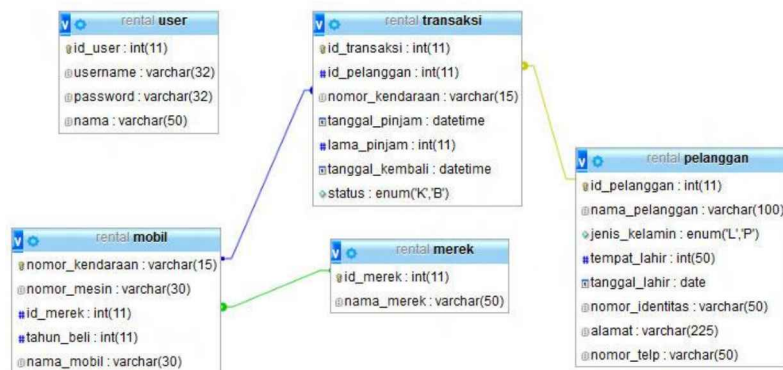
Jika semuanya sudah benar, maka kita akan memastikan apakah semua yang dilakukan untuk merelasikan tabel tersebut sudah benar-benar terimplementasi. Berikut ini cara mengeceknya:

- Pilih *database* “rental” yang ada pada sebelah kiri jendela phpMyAdmin.
- Pada bagian ujung kanan atas menu, pilih “More”.



Gambar 9. 17 Melihat Hasil Relasi

- Pilih “Designer”, akan tampil seperti pada gambar di bawah ini (terdapat garis relasi).



Gambar 9. 18 Hasil Relasi Antar Tabel

Gambar di atas menghubungkan satu tabel dengan tabel yang lain akan terhubung secara otomatis berdasarkan pembuatan relasi pada poin sebelumnya.

Rangkuman/Kesimpulan

Untuk membuat sebuah sistem informasi, hal pertama yang harus dilakukan adalah melakukan analisis dan observasi terhadap sistem informasi yang akan dibangun. Setelah melakukan analisis selanjutnya memetakan dalam bentuk perancangan *database* (dalam hal ini ERD), setelah itu implementasikan ke dalam *database*.

Latihan

Sebuah sekolah menengah atas (SMA) akan membuat sebuah Sistem Informasi Perpustakaan berbasis web. Selain untuk informasi koleksi katalog, sistem informasi perpustakaan ini juga dijadikan untuk proses transaksi yaitu pinjam dan kembali buku. Yang dilakukan oleh operator. Pengguna dari sistem informasi perpustakaan ini adalah: Anggota, Petugas Perpustakaan, dan Admin dari perpustakaan. Adapun koleksi dari katalog berupa buku, majalah, jurnal dan ebook.

Dari deskripsi di atas, Anda diminta untuk merancang *database*, tabel dan relasi antar tabelnya. Setelah selesai melakukan perancangan selanjutnya implementasikan ke dalam *database* MySQL.

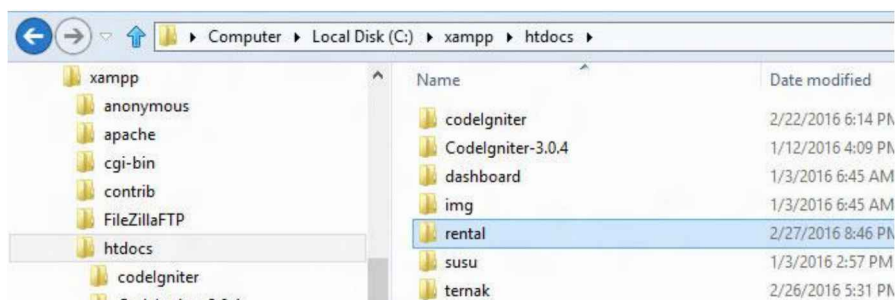
BAB X

PROYEK RENTAL MOBIL

Pada bab ini akan membahas bagaimana memulai membuat proyek sederhana, yaitu membuat aplikasi Rental Mobil menggunakan Codeigniter. Mulai dari persiapan, manajemen folder, editor atau IDE yang digunakan untuk menulis kode program, sampai dengan sistem administrasi pada aplikasinya. Diharapkan setelah membaca dan mencoba bab ini, akan dapat membuat aplikasi sederhana CRUD (create-read-update-delete).

10.1. Persiapan Proyek

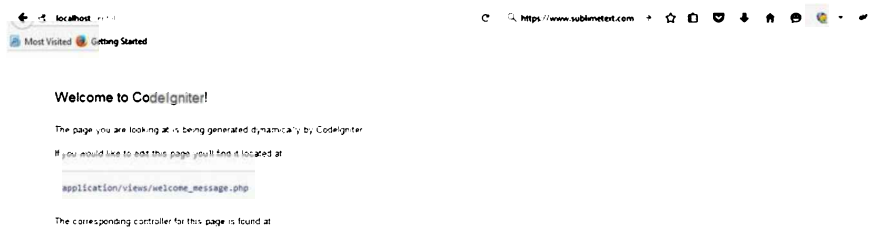
Sebelum masuk ke aplikasi, disiapkan terlebih dahulu aplikasi codeigniter yang masih bersih atau belum dikonfigurasi sama sekali. Anda bisa download atau mengekstrak file zip hasil download. Simpan di folder htdocs dan ubah namanya dari "CodeIgniter-3.0.4" menjadi "rental" seperti terlihat pada gambar di bawah ini.



Gambar 10. 1 Membuat Folder Proyek "Rental"

Kemudian akses pada browser untuk memastikan bahwa folder tersebut benar-benar hasil dari rename folder codeigniter.

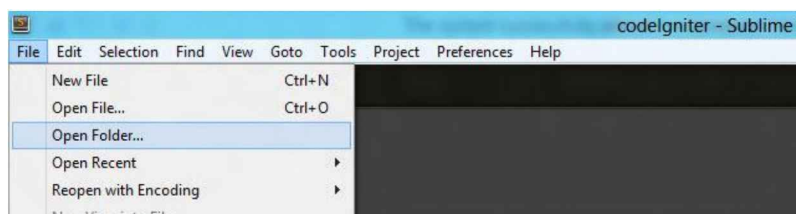
Jika benar maka hasilnya akan tampak seperti pada gambar di bawah ini:



Gambar 10. 2 Akses Folder Rental melalui Broser Sukses

Pada bagian ini, penulis beranggapan bahwa Anda sudah memiliki pengetahuan tentang pemrograman HTML dan CSS. Penulis tidak akan detail menjelaskan pembuatan form login, tetapi lebih kepada fungsi Codeigniternya, sehingga pembuatan form login atau tampilan lainnya dianggap sudah menguasai. Jika Anda belum menguasai tentang bahasa pemrograman HTML dan CSS, disarankan membaca-baca kembali tentang bahasa pemrograman HTML dan CSS yang banyak di toko buku atau internet. Dalam penulisan script PHP, penulis menggunakan editor (IDE) sublime (<https://www.sublimetext.com>) dan menggunakan system operasi Windows.

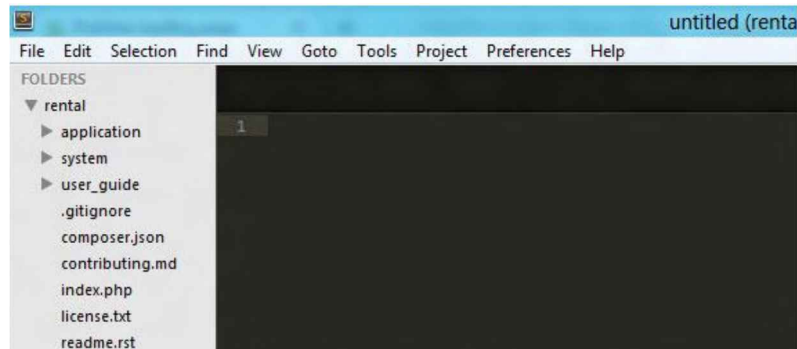
1. Buka aplikasi sublime, buka aplikasi rental dengan menggunakan Open Folder.



Gambar 10. 3 Buka Folder Proyek Rental

2. Pastikan hasilnya terbuka folder pada bagian sebelah kiri ada folder rental seperti pada gambar di bawah ini. Jika

sudah seperti gambar di bawah ini, maka kita siap untuk memulai proyek aplikasi rental.



Gambar 10. 4 Aplikasi Rental Siap Digunakan pada Sublime

10.2. Konfigurasi

Sebelum melakukan scripting, ada beberapa hal yang harus dikonfigurasi pada beberapa file.

- Karena kita berhubungan dengan *database*, maka perlu melakukan konfigurasi pada file *database.php* yang berada pada folder *rental/application/config* pada baris 76 lakukan penambahan seperti pada script di bawah ini:

```
$db['default'] = array(  
    'dsn'      => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'rental',
```

Pada bagian username tambahkan username = “root”, password tidak diubah atau tetap dalam keadaan kosong dan *database* diisi dengan nama *database* yang sudah kita buat sebelumnya yaitu “rental”.

- Pada file `autoload.php` yang berada dalam folder yang sama dengan file `database.php` tambahkan script di bawah ini pada baris 61:

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

File `autoload.php` ini akan dijalankan pertama kali ketika aplikasi dijalankan (auto load). Maka koneksi `database` dan fungsi `session` akan dijalankan pertama kali.

- Pada file `autoload.php` tambahkan pada baris 86 seperti pada script di bawah ini:

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

Digunakan untuk kebutuhan pembuatan URL dan pembuatan Form HTML.

- Pada file `config.php` tambahkan pada baris 26 seperti pada script di bawah ini:

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

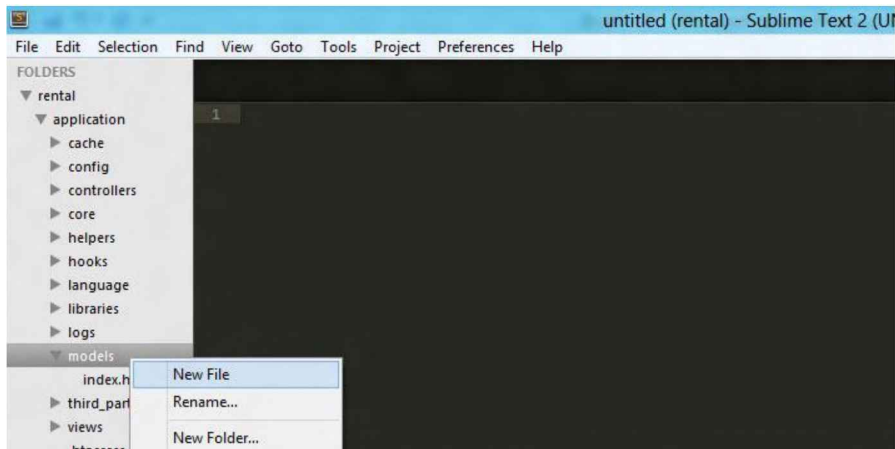
Untuk menentukan url root.

10.3. Model User (Login)

Seperti pada penjelasan sebelumnya, fungsi dari model adalah untuk melakukan hubungan dengan `database`, dalam hal ini melakukan manipulasi tabel yang ada di `database`. Model user ini digunakan untuk melakukan validasi terhadap user atau pengguna yang melakukan login dengan mencocokkan username dan password pada tabel user yang sudah dibuat. Untuk membuat model, lakukan langkah-langkah sebagai berikut:

1. Pada aplikasi Sublime yang sudah ada proyek folder rental di sebelah kiri, klik dua kali (double klik) rental, akan terbuka di bawahnya.
2. Pilih folder “application”.

3. Pilih folder “model” di dalam folder application.
4. Klik kanan pada folder model, kemudian pilih “New File” seperti pada gambar di bawah ini:



Gambar 10. 5 Membuat File Model

5. Buat script seperti pada bagian di bawah ini:

```
<?php
class MUser extends CI_Model {
    public $table = "user";
    public function __construct()
    {
        parent::__construct();
    }
    public function CheckUser($username,
$password)
    {
        $query = $this->db->get_where($this-
>table, array('username'=>$username,
'password'=>$password));
        if ($query->num_rows() > 0)
        {
            return true;
        } else{
            return false;
        }
    }
}
```

Simpan dengan nama file MUser.php dan pastikan file tersebut harus berada di dalam folder rental/application/models/MUser.php

6. Penjelasan Script.

```
class MUser extends CI_Model {
```

Untuk membuat sebuah kelas dengan nama MUser.

```
public $table = "user";
```

Untuk mendeklarasikan sebuah variable tabel dengan isi variable tabelnya adalah user. Fungsi ini digunakan untuk menyimpan nama tabel, untuk mengaksesnya dengan menggunakan script \$this->table.

```
public function __construct()
{
    parent::__construct();
}
```

Merupakan fungsi konstruktor, di mana fungsi inilah yang pertama kali dieksekusi ketika memanggil kelas MUser.

```
public function CheckUser($username, $password)
{
    $query = $this->db->get_where($this->table,
array('username'=>$username, 'password'=>$password));
    if ($query->num_rows() > 0)
    {
        return true;
    } else{
        return false;
    }
}
```

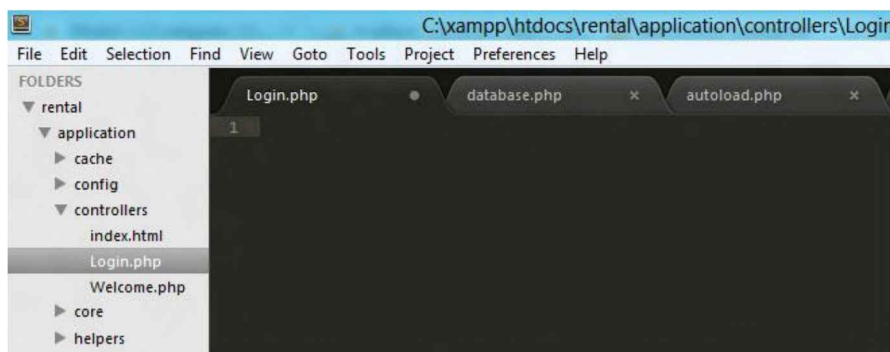
Digunakan untuk memeriksa username dan password ketika login. Fungsi ini memiliki dua parameter yaitu username dan password yang ditangkap kemudian dikirim untuk dicocokkan pada tabel user apakah ada di tabel atau tidak ada.

Script selanjutnya merupakan pengecekan, JIKA query ditemukan atau hasil > 0 maka fungsi akan mengembalikan nilai

TRUE atau benar, dan jika tidak maka akan mengembalikan nilai FALSE atau salah yang berarti login gagal.

10.4. Controller Login

Fungsi controller pada Framework Codeigniter adalah sebagai perantara atau pengontrol pergerakan aliran data dari model menuju view dan sebaliknya. Untuk membuat controller klik kanan pada folder controller di aplikasi sublime yang berada di folder application/controller buat sebuah file dengan nama login.php sehingga tampil seperti pada gambar di bawah ini:



Gambar 10. 6 Membuat Controller Login

Selanjutnya buat kode program seperti pada kode program di bawah ini:

```
<?php
class Login extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
    }

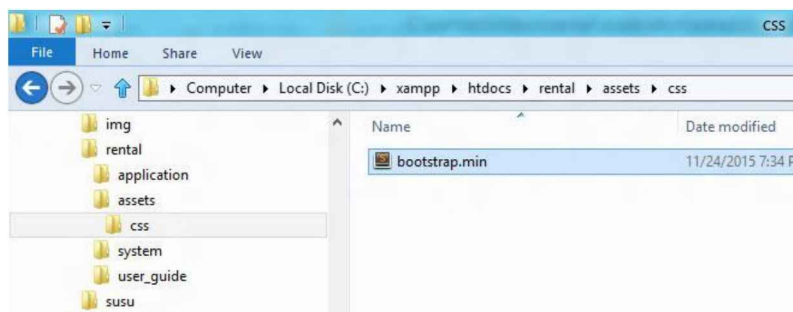
    public function index()
    {
        $this->load->view('login');
    }
}
?>
```

Fungsi index adalah fungsi yang diakses setelah fungsi constructor. Sama seperti pada sistem web root, index adalah bagian yang akan diakses pertama kali dan akan mengabaikan file-file lain dalam sebuah folder. Jadi ketika seseorang mengakses controller Login, maka akan ditampilkan fungsi index. Fungsi index ini adalah melakukan load view atau memerintahkan untuk menampilkan halaman login pada folder view (rental/application/view) yang akan dibuat pada bagian selanjutnya.

10.5. View Halaman Login

Fungsi View pada Framework Codeigniter adalah menampilkan halamannya. Pada bagian ini kita akan membuat halaman login dengan menggunakan CSS bootstrap dan HTML. Sebelum membuat kode program, ada beberapa hal yang harus disiapkan dalam pembuatan form login ini, yaitu:

- Buat sebuah folder dengan nama "assets" di dalam folder rental. Fungsi folder ini digunakan untuk menyimpan file-file tambahan yang digunakan dalam pembuatan website, seperti file Stylesheet (.css), JavaScript (.js), file gambar, font atau file tambahan lain yang dibutuhkan.
- Buat folder css di dalam folder assets, kemudian download file bootstrap.min.css di <http://getbootstrap.com> sehingga akan tampil seperti pada gambar di bawah ini:



Gambar 10. 7 Membuat Folder Assets

Berikut ini kode program yang ditulis pada views/login.php

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Login Admin Rental</title>
    <link href="<?php echo
base_url() ?>assets/css/bootstrap.min.css"
rel="stylesheet">
  </head>
  <body>
    <div class="container col-xs-8">
      <h2>Login App. Rental</h2>
      <?php $pesan = $this->session-
>flashdata('pesan');?>
      <?php if (@$pesan):?>
        <div class="alert alert-danger">
          <strong>Login Salah!</strong> Username
atau Password Anda salah!.
        </div>
      <?php endif?>
      <form role="form" class="form-horizontal"
action="<?php echo site_url() ?>/Login/validasi"
method="post">
        <div class="form-group">
          <div class="col-xs-4">
            <label for="username">Username</label>
            <input type="username" class="form-
control" placeholder="Username" name='username'>
          </div>
        </div>
        <div class="form-group">
          <div class="col-xs-4">
            <label
for="password">Password:</label>
            <input type="password"
class="form-control" placeholder="Password"
name='password'>
          </div>
        </div>
      </form>
    </div>
  </body>
</html>
```

```

        </div>
    </div>
    <div class="form-group">
        <div class="col-xs-4">
            <button type="submit" class="btn btn-
primary">Login</button>
        </div>
    </div>
</form>
</div>
</body>
</html>

```

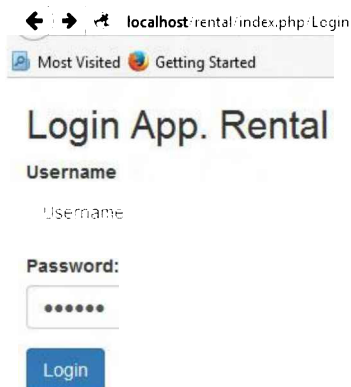
Untuk mengakses halaman login di atas, dengan cara ketik di browser

<http://localhost/rental/index.php/Login>

Konsep URL pada Framework Codeigniter adalah URL root kemudian controller nya, kemudian nama method/fungsi dalam controller di atas. Jika pada Controller tersebut ada fungsi index, maka cukup sampai nama controller aja mengaksesnya maka fungsi index akan otomatis diakses pertama kali. Jadi contoh di atas bisa diakses seperti di bawah ini:

<http://localhost/rental/index.php/Login/index>

Hasilnya akan tampil seperti pada gambar di bawah ini:



Gambar 10. 8 Halaman Login

Lakukan update kode program pada bagian controller Login.php sehingga kode programnya seperti pada kode program di bawah ini:

```
<?php
class Login extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->model('MUser');
    }

    public function index()
    {
        if ($this->session->userdata('logged') ==
true){
            redirect('rental');
        } else {
            $this->load->view('login');
        }
    }

    public function validasi()
    {
        $this->load->library('form_validation');
        $this->form_validation->
>set_rules('username', 'Username', 'required');
        $this->form_validation->
>set_rules('password', 'Password', 'required');
        if ($this->form_validation->run() == true){
            $username = $this->input->
>post('username');
            $password = $this->input->
>post('password');
            if ($this->MUser->
>CheckUser($username,$password) == true){
                $data =
array('username'=>$username, 'logged'=>true);
```

```

        $this->session->
>set_userdata($data);
        redirect('rental');
    } else {
        $this->session->
>set_flashdata('pesan', 'Username atau Password
Anda salah');
        redirect('Login');
    }
} else {
    $this->load->view('login');
}
}

public function logout()
{
    $this->session->session_destroy();
    redirect('Login', 'refresh');
}
}
?>

```

Untuk mencobanya, silahkan login dengan menggunakan username dan password yang salah. Jika username atau password yang dimasukan salah maka akan muncul pesan kesalahan yang tampil di atas form login seperti pada gambar di bawah ini:



Gambar 10. 9 Pesan Jika Login Salah

Penjelasan kode program dijelaskan pada bagian bawah.

Pada fungsi konstruktor terdapat kode program sebagai berikut:

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

Untuk memanggil model MUser, kalau pada PHP biasa mirip dengan fungsi include atau require untuk memanggil sebuah file (load file). Berikutnya pada fungsi index di atas ada kode program pengecekan apakah sudah login atau belum dengan mengecek isi session. Jika ada nilai “logged” bernilai benar, maka akan langsung redirect atau berpindah ke controller “rental” seperti pada kode program di bawah, jika tidak login maka tampilkan view login.

```
if ($this->session->userdata('logged') == true){  
    redirect('rental');  
} else {  
    $this->load->view('login');  
}
```

Berikutnya adalah fungsi validasi, bertujuan untuk mencocokkan input form yang dimasukan oleh user kemudian dicocokkan dengan data yang ada di tabel user pada *database* rental.

```
$this->load->library('form_validation');
```

Digunakan untuk memanggil fungsi validasi form. Codeigniter sudah membuahkan library untuk sistem validasi form, apakah harus diisi, format email, angka dan lainnya sudah disediakan fungsinya oleh Codeigniter, kita tinggal menggunakannya.

```
$this->form_validation->set_rules('username',  
'Username', 'required');
```

Kode program di atas membuat aturan/role terhadap nama input form “username” bahwa input tersebut harus diisi (*required*) atau tidak boleh kosong.

```
if ($this->form_validation->run() == true){
```

Kode program di atas berfungsi untuk menjalankan (*run*) validasi form. Kode berikutnya adalah untuk mengambil nilai dari form yang disimpan dalam sebuah variabel.

```
$username = $this->input->post('username');
```

Kode berikutnya adalah mencocokkan *username* dan *password* ke tabel user dengan bantuan model User dengan fungsi CheckUser. Perlu diingat bahwa *controller* untuk berhubungan dengan *database* harus lewat model (walaupun *controller* bisa saja langsung ke *database*).

```
if ($this->MUser->CheckUser($username,$password) == true){
```

Jika *username* dan *password* sama dengan *database* (*true*), maka kode program selanjutnya adalah membuat *array* berisi *username* dan *logged* yang akan disimpan dalam sebuah *session*, yang nantinya nilai *session* ini digunakan untuk mengecek apakah dia sudah login atau belum. Kemudian kode program akan mengarah ke *controller* rental. Jika salah maka akan membuat sebuah *session* sementara (*temporary*) untuk menyimpan pesan kesalahan dengan kode program di bawah ini:

```
$this->session->set_flashdata('pesan', 'Username atau Password Anda salah');
```

Fungsi yang terakhir adalah fungsi *logout*, digunakan untuk mengakhiri sebuah *session* (*destroy*). Setelah *session* dihancurkan kode program diarahkan kembali ke *controller login* dengan kondisi *refresh*, artinya akan melakukan reload halaman dengan utuh.

```
$this->session->session_destroy();  
redirect('Login', 'refresh');
```

10.6. Halaman Utama Aplikasi Rental

Halaman utama aplikasi rental akan terlihat atau terbuka apabila pengguna sukses melakukan *login* (*username* dan *password* benar). Halaman ini akan menampilkan menu-menu sesuai dengan apa yang kita rancang sebelumnya, yaitu menu peminjaman, pengembalian, input data master (peminjam, mobil) dan menu pencarian data mobil. Sesuai dengan skema sebelumnya, jika *login* sukses, maka akan langsung ke *controller* rental.

Buat file pada folder *controller* dengan nama file *Rental.php* kemudian tulis kode program seperti pada gambar di bawah ini:

```
<?php  
class Rental extends CI_Controller {  
    public function __construct()  
    {  
        parent::__construct();  
        if ($this->session->userdata('logged') !=  
true) {  
            redirect('Login');  
        }  
    }  
  
    public function index()  
    {  
        $data['home'] = true;  
        $this->load->view('dashboard', $data);  
    }  
}  
?>
```

Pada bagian fungsi *constructor* terdapat pengecekan apakah *user* tersebut sudah *login* atau belum, jika belum *login* maka akan melakukan *redirect* ke *controller login* dan menampilkan *form login*. Jika *user login* maka akan menampilkan *view dashboard*.
\$data['home'] = true;

Digunakan untuk mengirimkan nilai ke *view* bahwa *variable home* adalah *true*, sehingga akan ditangkap oleh *view* untuk memberikan efek aktif pada navigasi atau menu pada halaman *dashboard* bagian atas.

Selanjutnya kita akan membuat *view dashboard*. Pada halaman *dashboard* ini akan mengambil file *header.php* dan *footer.php*, tujuannya agar tidak menulis berulang ketika kita membuat file berikutnya. Berikut ini kode program file *header.php*

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Sistem Informasi Rental</title>
    <link href="<?php echo
base_url()??>assets/css/bootstrap.min.css"
rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <h2>Sistem Informasi Rental Mobil</h2>
      <div class="row">
        <nav class="navbar navbar-default">
          <ul class="nav navbar-nav">
            <li class="<?php echo (@$home == true
? "active": "")?>"><a href="<?php echo
site_url()??>/Rental">Home</a></li>
            <li class="<?php echo (@$pinjam ==
true ? "active": "")?>"><a href="<?php echo
site_url()??>/Pinjam">Pinjam</a></li>
            <li class="<?php echo (@$kembali ==
true ? "active": "")?>"><a href="<?php echo
site_url()??>/Kembali">Kembali</a></li>
            <li class="<?php echo (@$master ==
true ? "active": "")?>"><a href="<?php echo
site_url()??>/Master">Master</a></li>
            <li class="<?php echo (@$cari == true
? "active": "")?>"><a href="<?php echo
site_url()??>/Cari">Cari Mobil</a></li>
            <li><a href="<?php echo
site_url()??>/Login/logout">Logout</a></li>
          </ul>
        </nav>
      </div>
    </div>
```

```

        </nav>
    </div>
    <div class="row">

```

Berikut ini kode program halaman footer.php

```

        </div>
    </div>
</body>
</html>

```

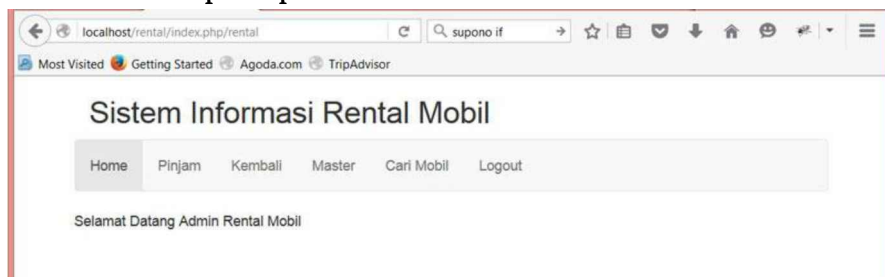
Berikut ini kode program halaman dashboard.php yang merupakan gabungan dari header.php dan footer.php seperti di bawah ini

```

<?php $this->load->view('header.php')?>
    Selamat Datang Admin Rental Mobil
<?php $this->load->view('footer.php')?>

```

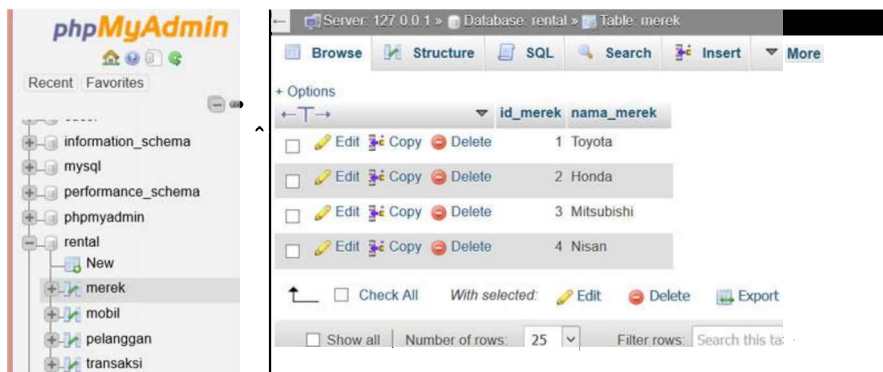
Berikut ini tampilan pada *browser* halaman *dashboard*



Gambar 10. 10 Halaman Dashbaord Rental Mobil

10.7. Halaman Kelola Data Master (Data Mobil)

Halaman ini digunakan untuk pengelolaan data mobil (tambah, ubah dan hapus data mobil). *Controller* yang digunakan adalah *controller* master yang harus dibuat, sebelum membuat *controller* kita akan membuat model untuk data mobil. Untuk merek mobil dapat dimasukan datanya secara manual dari phpMyAdmin seperti gambar di bawah ini:



Gambar 10. 11 Data Merk Mobil pada phpMyAdmin

Berikut ini kode program model untuk MMobil.php yang terdiri dari fungsi tambah, ubah, hapus, detail dan jumlah data dari table mobil.

```
<?php
class MMobil extends CI_Model{
    public $table = "mobil";
    function __construct(){
        parent::__construct();
    }

    public function
setData($nomor_kendaraan,$nomor_mesin,$id_merek,$t
ahun_beli,$nama_mobil){
        $this->nomor_kendaraan =
$nomor_kendaraan;
        $this->nomor_mesin = $nomor_mesin;
        $this->id_merek = $id_merek;
        $this->tahun_beli = $tahun_beli;
        $this->nama_mobil = $nama_mobil;
    }

    public function add(){
        $arrayData = array(
            'nomor_kendaraan'=>$this-
>nomor_kendaraan,
```

```

        'nomor_mesin'=>$this->nomor_mesin,
    >nomor_mesin,
        'id_merek'=>$this->id_merek,
        'tahun_beli'=>$this->tahun_beli,
        'nama_mobil'=>$this->nama_mobil,
    );
    return $this->db->insert($this->table,
$arrayData);
}

    public function edit($id){
        $arrayData = array(
            'nomor_kendaraan'=>$this->nomor_kendaraan,
    >nomor_kendaraan,
            'nomor_mesin'=>$this->nomor_mesin,
    >nomor_mesin,
            'id_merek'=>$this->id_merek,
            'tahun_beli'=>$this->tahun_beli,
            'nama_mobil'=>$this->nama_mobil,
        );
        $this->db->where('nomor_mesin', $id);
        return $this->db->update($this->table,
$arrayData);
    }

    public function getList(){
        $query = $this->db->get($this->table);
        return $query->result();
    }

    public function delete($id){
        return $this->db->delete($this->table,
array('id'=>$id));
    }

    function detail($id){
        $condition = array("nomor_mesin"=>$id);
        $query = $this->db->get_where($this->table,$condition);
    }

```

```

        if($query->num_rows() > 0){
            return $query->row();
        } else {
            return false;
        }
    }

    public function getTotal(){
        return $this->db->count_all_results($this->tnews);
    }
}
?>

```

Menampilkan Data Mobil

Yang berikutnya adalah membuat halaman controller untuk menampilkan list mobil. Buat controller dengan nama Master.php seperti pada kode program di bawah ini:

```

<?php
class Master extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        if ($this->session->userdata('logged') !=
true){
            redirect('Login');
        }
    }

    public function index()
    {
        $data['master'] = true;
        $this->load->view('master',$data);
    }
}
?>

```

Berikut ini *view* untuk kode program untuk *view master.php* seperti di bawah ini:

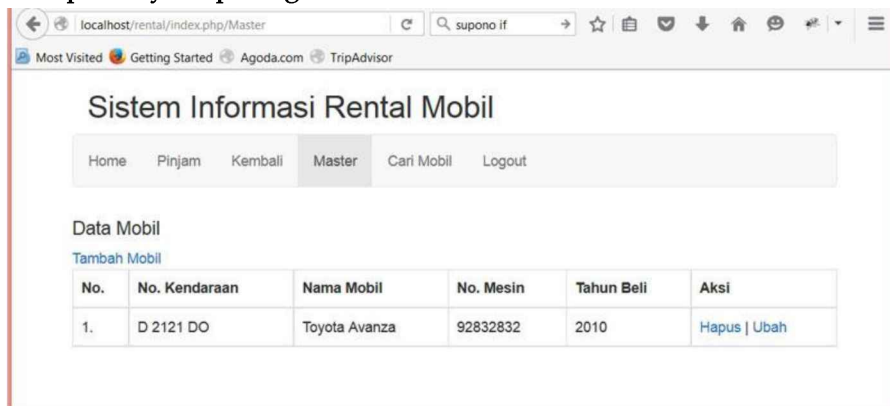
```
<?php $this->load->view('header.php')?>
<h4>Data Mobil</h4>
<a href="<?php echo
site_url()?>/Master/detail">Tambah Mobil</a>
<?php if ($result):?>
<table class="table table-bordered">
<thead>
<tr>
<th>No.</th>
<th>No. Kendaraan</th>
<th>Nama Mobil</th>
<th>No. Mesin</th>
<th>Tahun Beli</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<?php $i=1;?>
<?php foreach ($result as $row):?>
<tr>
<td><?php echo $i?>.</td>
<td><?php echo $row-
>nomor_kendaraan?></td>
<td><?php echo $row-
>nama_mobil?></td>
<td><?php echo $row-
>nomor_mesin?></td>
<td><?php echo $row-
>tahun_beli?></td>
<td>
<a href="<?php echo
site_url()?>/Master/delete/<?php echo $row-
>nomor_kendaraan?>">Hapus</a> |
<a href="<?php echo
site_url()?>/Master/detail/<?php echo $row-
>nomor_mesin?>">Ubah</a>
</td>
</tr>
<?php $i++?>
<?php endforeach?>
</tbody>
</table>
```

```

<?php else: ?>
    Tidak ada data
<?php endif?>
<?php $this->load->view('footer.php')?>

```

Tampilannya seperti gambar di bawah ini:



Gambar 10. 12 Tampilan Data Mobil

Tambah Data Mobil

Selanjutnya kita akan melakukan tambah data mobil, untuk tambah data mobil kita hanya menambahkan fungsi di *controller* yang sudah ada yaitu di *controller* Master.php dengan nama fungsi detail(). Berikut ini kode program untuk fungsi detail() pada controller Master:

```

public function detail($id=NULL){
    $data['master'] = true;
    $this->load->model('MMerk','merek');
    $data['merk'] = $this->merek->getList();
    $this->load->library('form_validation');
    $submit = $this->input->post('submit');
    if ($submit){
        $nomor_kendaraan = $this->input-
>post('nomor_kendaraan');
        $nomor_mesin = $this->input-
>post('nomor_mesin');

```

```

        $id_merek = $this->input-
>post('id_merek');
        $tahun_beli = $this->input-
>post('tahun_beli');
        $nama_mobil = $this->input-
>post('nama_mobil');
        $this->form_validation-
>set_rules('nomor_kendaraan', 'Nomor Kendaraan',
'required');
        $this->form_validation-
>set_rules('nama_mobil', 'Nama Mobil',
'required');

        if ($this->form_validation->run()==FALSE) {
            $data['errors'] = TRUE;
        } else {
            if ($id){
                $data['detail'] = $this->MMobil-
>detail($id);
            }
            $this->MMobil-
>setData($nomor_kendaraan,$nomor_mesin,$id_merek,$
tahun_beli,$nama_mobil);
            if ($id){
                $this->MMobil->edit($id);
                redirect('Master');
            } else {
                $this->MMobil->add();
            }
            redirect('Master');
        }
    } else {
        if ($id){
            $data['detail'] = $this->MMobil-
>detail($id);
        }
    }
    $this->load->view('master-detail', $data);
}

```

Fungsi di atas selain untuk melakukan tambah data, juga digunakan untuk melakukan edit/ubah data mobil.

```
$this->load->library('form_validation');
```

Kode program di atas digunakan untuk melakukan validasi terhadap form, yang mengizinkan *form* tersebut apakah harus diisi, panjang teks, validasi email, dan lainnya.

Selanjutnya membuat *view* dari tampilan tambah data mobil sebagai berikut:

```
<?php $this->load->view('header.php')?>
<h4>Tambah Data Mobil</h4>
<form class="form-horizontal" role="form"
method="post" action="<?php echo
site_url()?>/Master/detail">
  <div class="form-group">
    <label class="control-label col-sm-2"
for="noken">Nomor Kendaraan:</label>
    <div class="col-sm-10">
      <input type="text" class="form-control"
name="nomor_kendaraan" value="<?php if (@$detail){
echo @$detail->nomor_kendaraan;} else { echo
set_value('nomor_kendaraan');}?>">
    </div>
  </div>
  <div class="form-group">
    <label class="control-label col-sm-2"
for="nomes">Nomor Mesin:</label>
    <div class="col-sm-10">
      <input type="text" class="form-control"
name="nomor_mesin" value="<?php if (@$detail){
echo @$detail->nomor_mesin;} else { echo
set_value('nomor_mesin');}?>">
    </div>
  </div>
  <div class="form-group">
    <label class="control-label col-sm-2"
for="merk">Merk Mobil:</label>
```

```

        <div class="col-sm-10">
            <select class="form-control"
name="id_merek">
                <?php foreach ($merk as $mrk):?>
                    <option value="<?php echo $mrk-
>id_merek?"><?php echo $mrk-
>nama_merek?"></option>
                <?php endforeach?>
            </select>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2"
for="thn">Tahun Beli:</label>
        <div class="col-sm-10">
            <input type="text" class="form-control"
name="tahun_beli" value="<?php if (@$detail){ echo
@$detail->tahun_beli;} else { echo
set_value('tahun_beli');}?>">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2"
for="namo">Nama Mobil:</label>
        <div class="col-sm-10">
            <input type="text" class="form-control"
name="nama_mobil" value="<?php if (@$detail){ echo
@$detail->nama_mobil;} else { echo
set_value('nama_mobil');}?>">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2"
for="nomes"></label>
        <div class="col-sm-2">
            <input type="submit" class="form-control"
name="submit" value="Simpan">
        </div>
    </div>

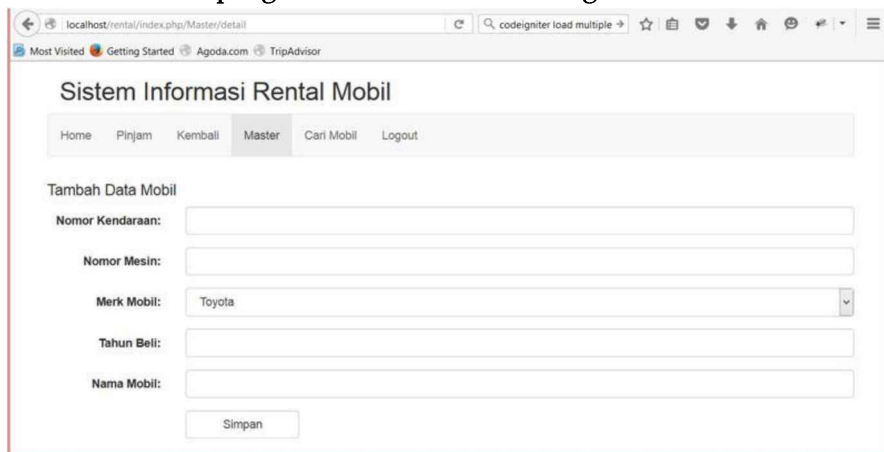
```

```
</form>
<?php $this->load->view('footer.php')?>
```

Terdapat kode program seperti di bawah ini, yang bertujuan untuk memberikan nilai jika kondisi dalam keadaan/status edit (menampilkan data), dan `set_value` digunakan apabila pengguna menekan tombol submit, tetapi ada kesalahan/tidak diisi sebagian, maka data yang sebelumnya tidak akan hilang pada form tersebut, akan ditampilkan kembali dengan fungsi ini.

```
<?php if (@$detail){ echo @$detail->nomor_mesin;}
else { echo set_value('nomor_mesin');}?>
```

Hasil dari kode program di atas adalah sebagai berikut:



Gambar 10. 13 Tampilan Tambah Data Mobil

Ubah Data

Proses ubah data sama halnya seperti tambah data, dan fungsi ubah data sudah disatukan dengan tambah data. Yang membedakan adalah jika ubah data membawa parameter ID (`nomor_mesin`) yaitu mobil dengan nomor mesin mana yang akan dibuat, sedangkan kalau tambah data tidak membawa parameter.

Hapus Data

Untuk dapat melakukan hapus data, maka harus ditambahkan satu buah fungsi pada controller Master seperti di bawah ini:

```
public function delete($id){  
    $this->MMobil->delete($id);  
    redirect('Master');  
}
```

Fungsi di atas akan membawa ID (nomor_mesin) ke model dan pada model akan dicari apakah ID dengan nomor mesin tersebut ada atau tidak ada. Jika ada maka akan menghapusnya

10.8. Halaman Transaksi Pinjam Mobil

Proses transaksi pinjam mobil dilakukan sebagai berikut: admin atau operator akan memasukan nomor kendaraan pada form yang disediakan, kemudian sistem akan mencari ke *database* apakah nomor kendaraan tersebut ada atau tidak, jika ada apakah mobil dengan nomor kendaraan tersebut sedang dipinjam atau tidak. Jika sedang dipinjam maka akan ditampilkan bahwa mobil tersebut sedang dipinjam, jika tidak sedang dipinjam maka akan memberikan informasi memasukan nomor identitas pelanggan (jika pelanggan tersebut sudah mendaftar sebelumnya), jika belum mendaftar maka akan didaftarkan terlebih dahulu.

Kelola Pelanggan

Sebelum membuat fungsi transaksi peminjaman, kita akan menyiapkan fungsi untuk penambahan data pelanggan, yaitu:

1. Membuat model untuk table pelanggan.
2. Membuat controller-nya
3. Membuat view untuk registrasi/pendaftaran pelanggan.
4. Menambahkan menu Pelanggan pada menu bar (header.php)

Berikut ini kode untuk membuat model pelanggan dengan nama file MPelanggan.php, untuk saat ini kita hanya menggunakan tambah dan menampilkan data pelanggan, untuk menambahkan fungsi lain seperti hapus dan ubah secara konsepnya sama seperti pada model Master mobil, kode program model MPelanggan sebagai berikut:

```
<?php
class MPelanggan extends CI_Model{
    public $table = "pelanggan";
    function __construct(){
        parent::__construct();
    }

    public function
setData($nama_pelanggan,$jenis_kelamin,$tempat_lahir,$tanggal_lahir,$nomor_identitas,$alamat,$nomor_telp){
        $this->nama_pelanggan =
$nama_pelanggan;
        $this->jenis_kelamin = $jenis_kelamin;
        $this->tempat_lahir = $tempat_lahir;
        $this->tanggal_lahir = $tanggal_lahir;
        $this->nomor_identitas =
$nomor_identitas;
        $this->alamat = $alamat;
        $this->nomor_telp = $nomor_telp;
    }

    public function add(){
        $arrayData = array(
            'nama_pelanggan'=>$this->nama_pelanggan,
            'jenis_kelamin'=>$this->jenis_kelamin,
            'tempat_lahir'=>$this->tempat_lahir,
```

```

        'tanggal_lahir'=>$this->tanggal_lahir,
        'nomor_identitas'=>$this->nomor_identitas,
        'alamat'=>$this->alamat,
        'nomor_telp'=>$this->nomor_telp,
    );
    return $this->db->insert($this->table, $arrayData);
}

public function getList(){
    $query = $this->db->get($this->table);
    return $query->result();
}
}

```

Selanjutnya membuat controller untuk menambah data pelanggan dan menampilkan data pelanggan dengan nama file **Pelanggan.php**, berikut kode programnya:

```

<?php
class Pelanggan extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        if ($this->session->userdata('logged') != true){
            redirect('Login');
        }
        $this->load->model('MPelanggan');
    }

    public function index()
    {
        $data['result'] = $this->MPelanggan->getList();
        $data['pelanggan'] = true;
        $this->load->view('pelanggan', $data);
    }
}

```

```

    }
    public function add($id=NULL){
        $data['master'] = true;
        $this->load->model('MMerk','merek');
        $data['merk'] = $this->merek->getList();
        $this->load->library('form_validation');
        $submit = $this->input->post('submit');
        if ($submit){
            $nomor_kendaraan = $this->input-
>post('nomor_kendaraan');
            $nomor_mesin = $this->input-
>post('nomor_mesin');
            $id_merek = $this->input-
>post('id_merek');
            $tahun_beli = $this->input-
>post('tahun_beli');
            $nama_mobil = $this->input-
>post('nama_mobil');
            $this->form_validation-
>set_rules('nomor_kendaraan', 'Nomor Kendaraan',
'required');
            $this->form_validation-
>set_rules('nama_mobil', 'Nama Mobil',
'required');

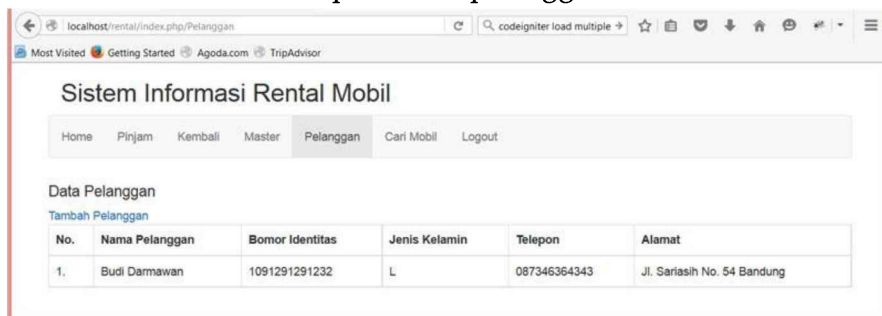
            if ($this->form_validation-
>run()==FALSE){
                $data['errors'] = TRUE;
            } else {
                $this->MMobil-
>setData($nomor_kendaraan,$nomor_mesin,$id_merek,$
tahun_beli,$nama_mobil);
                $this->MMobil->add();
                redirect('Pelanggan');
            }
        }
        $this->load->view('pelanggan-detail',
$data);
    }
}

```

Berikut ini kode program pada view dengan nama pelanggan.php

```
<?php $this->load->view('header.php')?>
    <h4>Data Pelanggan</h4>
    <a href="<?php echo
site_url()?>/Pelanggan/detail">Tambah
Pelanggan</a>
    <?php if ($result):?>
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>No.</th>
                    <th>Nama Pelanggan</th>
                    <th>Bomor Identitas</th>
                    <th>Jenis Kelamin</th>
                    <th>Telepon</th>
                    <th>Alamat</th>
                </tr>
            </thead>
            <tbody>
                <?php $i=1;?>
                <?php foreach ($result as $row):?>
                    <tr>
                        <td><?php echo $i?>.</td>
                        <td><?php echo $row-
>nama_pelanggan?></td>
                        <td><?php echo $row-
>nomor_identitas?></td>
                        <td><?php echo $row-
>jenis_kelamin?></td>
                        <td><?php echo $row-
>nomor_telp?></td>
                        <td><?php echo $row->alamat?></td>
                    </tr>
                    <?php $i++?>
                <?php endforeach?>
            </tbody>
        </table>
    <?php else: ?>
        Tidak ada data
    <?php endif?>
<?php $this->load->view('footer.php')?>
```

Berikut ini hasil dari tampilan data pelanggan



Gambar 10. 14 Tampilan Tambah Data Pelanggan

Transaksi Peminjaman

Setelah kita punya fungsi untuk menambah data pelanggan, selanjutnya adalah melakukan proses transaksi peminjaman mobil. Seperti biasa kita akan membuat model dari table transaksi dengan nama MTransaksi.php dengan kode program sebagai berikut:

```
<?php
class MTransaksi extends CI_Model{
    public $table = "transaksi";
    function __construct(){
        parent::__construct();
    }

    public function
setData($id_pelanggan,$nomor_kendaraan,$tanggal_pi
njam,$lama_pinjam,$tanggal_kembali){
    $this->id_pelanggan = $id_pelanggan;
    $this->nomor_kendaraan =
$nomor_kendaraan;
    $this->tanggal_pinjam =
$tanggal_pinjam;
    $this->lama_pinjam = $lama_pinjam;
    $this->tanggal_kembali =
$tanggal_kembali;
}
```

```

        public function add(){
            $arrayData = array(
                'id_pelanggan'=>$this->id_pelanggan,
                'nomor_kendaraan'=>$this->nomor_kendaraan,
                'tanggal_pinjam'=>$this->tanggal_pinjam,
                'lama_pinjam'=>$this->lama_pinjam,
                'tanggal_kembali'=>$this->tanggal_kembali,
                'status'=>"B",
            );
            return $this->db->insert($this->table, $arrayData);
        }

        public function getList($status=NULL){
            $this->db->select('*');
            $this->db->from($this->table);
            $this->db->join('pelanggan', 'pelanggan.id_pelanggan = '.$this->table.'.id_pelanggan', 'left');
            if ($status){
                $this->db->where('status', $status);
            }
            $this->db->order_by('id_transaksi', 'DESC');
            $query = $this->db->get();
            return $query->result();
        }

        function cekStatus($nomor_kendaraan, $status){
            $condition = array("status"=>$status, "nomor_kendaraan"=>$nomor_kendaraan);

```

```

        $query = $this->db->get_where($this->table,$condition);
        if($query->num_rows() > 0){
            return true;
        } else {
            return false;
        }
    }
}

```

Pada bagian fungsi getList terdapat query untuk melakukan join dengan tabel lain, yaitu tabel transaksi dengan tabel pelanggan untuk menampilkan nama siapa yang melakukan peminjaman atas nomor kendaraan tersebut. Berikutnya pada fungsi cekStatus berfungsi untuk melakukan pengecekan terhadap nomor kendaraan dan status pada saat proses peminjaman apakah nomor kendaraan tersebut sedang dipinjam atau tidak, jika ada maka akan bernilai TRUE dan jika tidak ada akan bernilai FALSE.

Selanjutnya membuat controller Pinjam.php dengan menuliskan kode program seperti di bawah ini:

```

<?php
class Pinjam extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        if ($this->session->userdata('logged') !=
true){
            redirect('Login');
        }
        $this->load->model('MTransaksi');
    }

    public function index()
    {
        $data['result'] = $this->MTransaksi->getList();
        $data['pinjam'] = true;
        $this->load->view('pinjam',$data);
    }
}

```

```

    public function add($id=NULL){
        $data['pinjam'] = true;
        $data['result'] = $this->MTransaksi->getList();
        $this->load->library('form_validation');
        $submit = $this->input->post('submit');
        if ($submit){
            $nomor_kendaraan = $this->input-
>post('nomor_kendaraan');
            $id_pelanggan = $this->input-
>post('id_pelanggan');
            $lama_pinjam = $this->input-
>post('lama_pinjam');
            $this->form_validation-
>set_rules('nomor_kendaraan', 'Nomor Kendaraan',
'required');
            $this->form_validation-
>set_rules('id_pelanggan', 'ID Pelanggan', 'required');
            $this->form_validation-
>set_rules('lama_pinjam', 'Lama Pinjam', 'required');
            if ($this->form_validation->run()===FALSE){
                $data['errors'] = TRUE;
            } else {
                $this->load->model('MMobil','mobil');
                $this->load-
>model('MPelanggan','pelanggan');
                // cek Apakah Nomor Kendaraan Ada di
tabel Mobil
                $error = array();
                if (!$this->mobil-
>findByNoKend($nomor_kendaraan)){
                    $error[] = "Nomor Kendaraan tidak
ditemukan di database";
                }
                // cek apakah ID pelanggan ada di tabel
pelanggan
                if (!$this->pelanggan-
>findByIdPel($id_pelanggan)){
                    $error[] = "ID Pelanggan tidak
ditemukan di database";
                }
            }
        }
    }

```

```

        // cek apakah sedang dipinjam atau
        tidak
        if ($this->MTransaksi-
>cekStatus($nomor_kendaraan,"B")){
            $error[] = "Mobil tersebut sedang
dipinjam";
        }
        if (!$error){
            $tanggal_pinjam = date('Y-m-d
H:i:s');
            $tanggal_kembali_str =
strtotime($tanggal_pinjam."+ $lama_pinjam days");
            $tanggal_kembali = date("Y-m-d
H:i:s",$tanggal_kembali_str);
            $this->MTransaksi-
>setData($id_pelanggan,$nomor_kendaraan,$tanggal_pinjam
,$lama_pinjam,$tanggal_kembali);
            $this->MTransaksi->add();
            redirect('Pinjam');
        } else {
            $data['error'] = $error;
        }
    }
}
$this->load->view('pinjam', $data);
}

public function delete($id){
    $this->MMobil->delete($id);
    redirect('Master');
}
}
?>

```

Ketika lolos dari pengecekan validasi form, maka dihadapkan pada 3 kali pengecekan, yaitu:

1. Pengecekan terhadap Nomor Kendaraan pada tabel Mobil, dengan menambahkan fungsi find By No Kend (\$nomor_kendaraan) pada model M Mobil.
function findByNoKend(\$nomor_kendaraan){

```

        $condition =
array("nomor_kendaraan"=>$nomor_kendaraan);
        $query = $this->db->get_where($this->
table,$condition);
        if($query->num_rows() > 0){
            return true;
        } else {
            return false;
        }
    }
}

```

2. Pengecekan terhadap ID Pelanggan pada tabel pelanggan, oleh karena itu tambahkan juga fungsi find By Id Pel(\$id_pelanggan) pada model MPelanggan.

```

function findByIdPel($id_pelanggan){
    $condition =
array("id_pelanggan"=>$id_pelanggan);
    $query = $this->db->get_where($this->
table,$condition);
    if($query->num_rows() > 0){
        return $query->row();
    } else {
        return false;
    }
}

```

3. Dan yang terakhir melakukan pengecekan terhadap tabel transaksi apakah Mobil yang dimaksud akan dipinjam itu sedang dipinjam atau tidak.

Yang berikutnya adalah membuat view-nya untuk transaksi peminjaman dengan nama file pinjam.php dengan kode program seperti di bawah ini:

```

<?php $this->load->view('header.php')?>
<h4>Transaksi Peminjaman Mobil</h4>
<div class="alert alert-danger">
    <strong>Ada Kesalahan, silahkan cek di bawah
ini:<br></strong>
    <?php echo validation_errors(); ?>
    <?php if (@$error):?>

```

```

        <?php foreach ($error as $er=>$value):?>
            <?php echo $value; echo "<br>";?>
        <?php endforeach?>
    <?php endif?>
</div>
<form class="form-horizontal" role="form"
method="post" action="<?php echo
site_url()?>/Pinjam/add">
    <div class="form-group">
        <label class="control-label col-sm-2"
for="noken">Nomor Kendaraan:</label>
        <div class="col-sm-2">
            <input type="text" class="form-control"
name="nomor_kendaraan" value="<?php echo
set_value('nomor_kendaraan');?>">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2"
for="noken">ID Pelanggan:</label>
        <div class="col-sm-2">
            <input type="text" class="form-control"
name="id_pelanggan" value="<?php echo
set_value('id_pelanggan');?>">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2"
for="noken">Lama Pinjam (hari):</label>
        <div class="col-sm-1">
            <input type="text" class="form-control"
name="lama_pinjam" value="<?php echo
set_value('lama_pinjam');?>">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-2"
for="nomen"></label>
        <div class="col-sm-2">

```

```

        <input type="submit" class="form-control"
name="submit" value="Simpan">
    </div>
</div>
</form>

<?php if ($result):?>
<table class="table table-bordered">
    <thead>
        <tr>
            <th>No.</th>
            <th>No. Kendaraan</th>
            <th>Nama Peminjam</th>
            <th>Tanggal Pinjam</th>
            <th>Tanggal Kembali</th>
            <th>Status</th>
        </tr>
    </thead>
    <tbody>
        <?php $i=1;?>
        <?php foreach ($result as $row):?>
            <tr>
                <td><?php echo $i?>.</td>
                <td><?php echo $row-
>nomor_kendaraan?></td>
                <td><?php echo $row->nama_pelanggan?></td>
                <td><?php echo $row->tanggal_pinjam?></td>
                <td><?php echo $row-
>tanggal_kembali?></td>
                <td><?php echo ($row->status == "B" ?
"Belum Kembali": "Sudah Kembali");?></td>
            </tr>
            <?php $i++?>
        <?php endforeach?>
    </tbody>
</table>
<?php else: ?>
    Tidak ada data
<?php endif?>
<?php $this->load->view('footer.php')?>

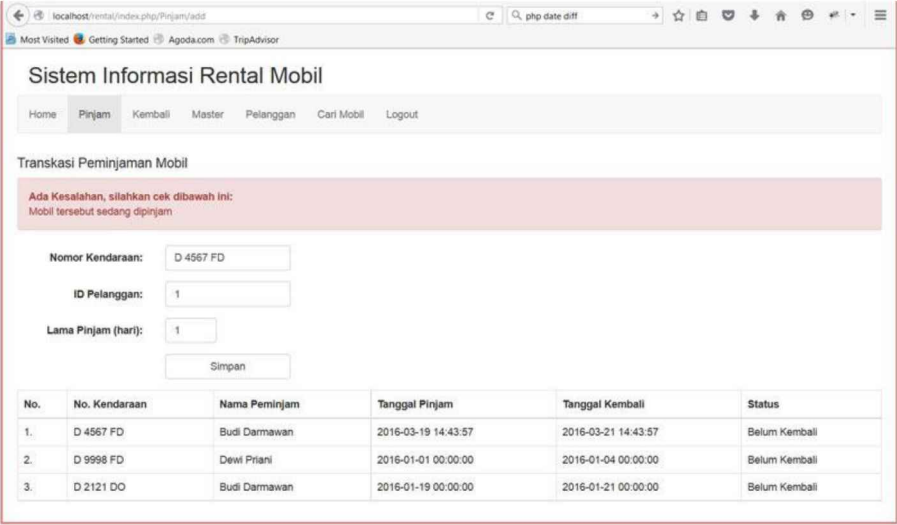
```

Pada kode di atas terdapat kode program seperti berikut ini:

```
<?php echo validation_errors(); ?>
```

Kode program di atas digunakan untuk menampilkan error yang berasal dari form jika tidak diisi. Pada tampilan ini terdiri dari form pengisian Nomor Kendaraan, ID Pelanggan dan Lama peminjaman yang semuanya wajib diisi. Pada bagian bawah form terdapat data siapa aja yang meminjam dan belum mengembalikan.

Berikut ini hasilnya:



No.	No. Kendaraan	Nama Peminjam	Tanggal Pinjam	Tanggal Kembali	Status
1.	D 4567 FD	Budi Darmawan	2016-03-19 14:43:57	2016-03-21 14:43:57	Belum Kembali
2.	D 9998 FD	Dewi Priani	2016-01-01 00:00:00	2016-01-04 00:00:00	Belum Kembali
3.	D 2121 DO	Budi Darmawan	2016-01-19 00:00:00	2016-01-21 00:00:00	Belum Kembali

Gambar 10. 15 Tampilan Tambah Data Pelanggan

10.9. Halaman Transaksi Kembali Mobil

Halaman ini digunakan untuk melakukan transaksi pengembalian mobil yang dipinjam. Ilustrasinya adalah sebagai berikut:

1. Peminjam (pelanggan dalam hal ini) datang langsung ke tempat peminjaman mobil untuk mengembalikan mobil yang dimaksud dengan membawa surat kendaraan (STNK).
2. Petugas (dalam hal ini admin atau operator perusahaan rental) memasukan nomor kendaraan yang tertera pada STNK ke system.

3. Sistem akan mengecek apakah nomor kendaraan tersebut ada atau tidak, jika ada di cek juga apakah peminjaman melebihi dari tanggal pengembalian atau tidak.
4. Jika melebihi tanggal yang ditentukan maka akan dikenakan denda. Pada studi kasus ini tidak terdapat pengelolaan denda.

Untuk model masih menggunakan MTransaksi.php, tetapi akan ditambahkan beberapa fungsi sesuai dengan kebutuhan untuk kelola transaksi pengembalian mobil. Pada prinsip datanya, pengembalian mobil merupakan update data peminjaman di tabel transaksi dari status “B” (belum kembali) menjadi “K” (kembali). Oleh karena ini kita akan membuat fungsi update ini pada model MTransaksi dan mengubah fungsi getList dengan menambahkan parameter nomor_kendaraan sebagai berikut:

```
public function  
getList($status=NULL,$nomor_kendaraan=NULL) {  
    $this->db->select('*');  
    $this->db->from($this->table);  
    $this->db->join('pelanggan',  
'pelanggan.id_pelanggan = '.$this->  
>table.'.id_pelanggan', 'left');  
    if ($status){  
        $this->db->where('status', $status);  
    }  
    if ($nomor_kendaraan){  
        $this->db->where('nomor_kendaraan',  
$nomor_kendaraan);  
    }  
    $this->db->order_by('id_transaksi', 'DESC');  
    $query = $this->db->get();  
    return $query->result();  
}
```

Fungsi di atas melakukan pencarian data berdasarkan dengan dua parameter, yaitu sedang dipinjam (B) dan berdasarkan

nomor kendaraan dan ditampilkan berdasarkan transaksi yang terbaru (order_by).

Berikut ini fungsi tambahan dengan nama fungsi `kembalikan()` pada model `MTransaksi`:

```
function kembalikan($id_transaksi){
    $arrayData = array(
        'status'=>"K",
    );
    $this->db->where('id_transaksi',
$id_transaksi);
    return $this->db->update($this->table,
$arrayData);
}
```

Fungsi di atas melakukan update terhadap status mobil dengan memberikan nilai K. Selanjutnya kita membuat controller untuk transaksi kembali dengan nama `Kembali.php` dengan kode program seperti di bawah ini:

```
<?php
class Kembali extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        if ($this->session->userdata('logged') !=
true){
            redirect('Login');
        }
        $this->load->model('MTransaksi');
    }

    public function index()
    {
        $data['result'] = $this->MTransaksi-
>getList("B");
        $data['kembali'] = true;
        $this->load->view('kembali',$data);
    }
}
```

```

        public function search($id=NULL){
            $data['kembali'] = true;
            $this->load->library('form_validation');
            $submit = $this->input->post('submit');
            if ($submit){
                $nomor_kendaraan = $this->input-
>post('nomor_kendaraan');
                $this->form_validation-
>set_rules('nomor_kendaraan', 'Nomor Kendaraan',
'required');
                if ($this->form_validation-
>run()==FALSE){
                    $data['errors'] = TRUE;
                } else {
                    $error = array();
                    // cek apakah sedang dipinjam atau
tidak
                    if (!$this->MTransaksi-
>cekStatus($nomor_kendaraan,"B")){
                        $error[] = "Tidak ada
peminjaman mobil dengan nomor kendaraan
".$nomor_kendaraan;
                    }
                    if (!$error){
                        $data['result'] = $this-
>MTransaksi->getList("B",$nomor_kendaraan);
                    } else {
                        $data['error'] = $error;
                        $data['result'] = $this-
>MTransaksi->getList("B");
                    }
                }
            }
            $this->load->view('kembali', $data);
        }

        function kembalikan($id_transaksi){
            $this->MTransaksi-
>kembalikan($id_transaksi);

```

```

        $this->session->set_flashdata('pesan',
'Mobil sudah dikembalikan');
        redirect("Kembali");
    }
}

```

Terdapat fungsi `kembalikan(id_transaksi)` digunakan untuk jika mengklik *link* `kembalikan` pada list peminjaman, maka akan diarahkan ke fungsi ini dan membawa ID transaksi. Berikut ini *controller* untuk file `kembali.php`

```

<?php
class Kembali extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        if ($this->session->userdata('logged') !=
true) {
            redirect('Login');
        }
        $this->load->model('MTransaksi');
    }

    public function index()
    {
        $data['result'] = $this->MTransaksi-
>getList("B");
        $data['kembali'] = true;
        $this->load->view('kembali',$data);
    }

    public function search($id=NULL) {
        $data['kembali'] = true;
        $this->load->library('form_validation');
        $submit = $this->input->post('submit');
        if ($submit){
            $nomor_kendaraan = $this->input-
>post('nomor_kendaraan');

```

```

        $this->form_validation-
>set_rules('nomor_kendaraan', 'Nomor Kendaraan',
'required');
        if ($this->form_validation-
>run()==FALSE) {
            $data['errors'] = TRUE;
        } else {
            $error = array();
            // cek apakah sedang dipinjam atau
tidak
            if (!$this->MTransaksi-
>cekStatus($nomor_kendaraan,"B")){
                $error[] = "Tidak ada
peminjaman mobil dengan nomor kendaraan
".$nomor_kendaraan;
            }
            if (!$error){
                $data['result'] = $this-
>MTransaksi->getList("B",$nomor_kendaraan);
            } else {
                $data['error'] = $error;
                $data['result'] = $this-
>MTransaksi->getList("B");
            }
        }
    }
    $this->load->view('kembali', $data);
}

function kembalikan($id_transaksi){
    $this->MTransaksi-
>kembalikan($id_transaksi);
    $this->session->set_flashdata('pesan',
'Mobil sudah dikembalikan');
    redirect("Kembali");
}
}

```

Maka tampilannya seperti pada gambar di bawah ini, gambar di bawah menampilkan data pada saat suatu mobil dikembalikan (terdapat pesan sukses).

The screenshot shows a web browser at the URL `localhost/rental/index.php/Kembali`. The page title is "Sistem Informasi Rental Mobil". The navigation menu includes "Home", "Pinjam", "Kembali" (active), "Master", "Pelanggan", "Cari Mobil", and "Logout".

Under the heading "Transaksi Pengembalian Mobil", there is a green success message: "Kembali Mobil! Mobil sudah dikembalikan." Below this is a form for "Nomor Kendaraan:" with an input field and a "Simpan" button.

A table displays the rental transactions:

No.	No. Kendaraan	Nama Peminjam	Tanggal Pinjam	Tanggal Kembali	Aksi
1.	D 4567 FD	Budi Darmawan	2016-03-20 00:25:10	2016-03-21 00:25:10	Kembalikan

Gambar 10. 16 Tampilan Tambah Data Pelanggan

10.10. Halaman Cari Data Mobil

Bagian ini adalah pencarian data mobil termasuk mobil tersebut statusnya sedang dipinjam atau sedang tidak dipinjam. Menu ini sangat penting terutama untuk proses pengecekan status mobil secara cepat berada pada posisi apa. Alur programnya adalah pertama kode program akan memeriksa ke tabel mobil, jika ada akan dilakukan pengecekan ke tabel transaksi apakah status mobil tersebut sedang dipinjam atau tersedia. Untuk proses pencarian dengan nomor kendaraan sudah tersedia pada modul MMobil.

Penambahan fungsi berada di MTransaksi, yaitu melakukan pengecekan apakah mobil dengan nomor kendaraan tersebut sedang dipinjam atau tidak. Berikut ini penambahan kode program pada model MTransaksi sebagai berikut:

```
function getStatus($nomor_kendaraan) {
    $condition =
array("nomor_kendaraan"=>$nomor_kendaraan);
    $this->db->order_by('status', 'DESC');
    $this->db->limit(1);
```

```

        $query = $this->db->get_where($this->table,$condition);
        if($query->num_rows() > 0){
            $result = $query->row();
            if ($result->status == "K"){
                $status = "Mobil Ada";
            } else {
                $status = "Mobil Sedang
Dipinjam";
            }
            return $status;
        } else {
            return false;
        }
    }
}

```

Fungsi kode program di atas mencari nomor kendaraan sesuai dengan input yang diberikan, dengan diurutkan berdasarkan status terbesar dan batasnya 1. Setelah menemukan kemudian melakukan pengecekan jika statusnya “K” maka akan diberikan status “Mobil Ada”, sedangkan jika bukan “K” maka diberikan status “Mobil Sedang Dipinjam”.

Setelah memodifikasi model transaksi, selanjutnya membuat controller Cari.php kemudian tuliskan kode program seperti di bawah ini:

```

<?php
class Cari extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        if ($this->session->userdata('logged') !=
true){
            redirect('Login');
        }
        $this->load->model('MMobil');
        $this->load->model('MTransaksi',
'transaksi');
    }
}

```

```

    }

    public function index()
    {
        $data['result'] = $this->MMobil-
>getList();
        $data['cari'] = true;
        $this->load->view('cari',$data);
    }

    public function search(){
        $data['cari'] = true;
        $this->load->library('form_validation');
        $submit = $this->input->post('submit');
        if ($submit){
            $nomor_kendaraan = $this->input-
>post('nomor_kendaraan');
            $this->form_validation-
>set_rules('nomor_kendaraan', 'Nomor Kendaraan',
'required');
            if ($this->form_validation-
>run()==FALSE){
                $data['errors'] = TRUE;
            } else {
                $data['result'] = $this->MMobil-
>findByNoKend($nomor_kendaraan);
            }
        }
        $this->load->view('cari', $data);
    }
}

```

Kode program di atas mirip dengan kode program pada kasus pencarian sebelumnya, selanjutnya kita membuat untuk view-nya dengan nama cari.php dengan kode program seperti di bawah ini:

```

<?php $this->load->view('header.php')?>
<h4>Transaksi Pengembalian Mobil</h4>

```

```

        <?php $pesan = $this->session-
>flashdata('pesan');?>
        <?php if (@$pesan):?>
        <div class="alert alert-success">
            <strong>Kembali Mobil!</strong> Mobil
sudah dikembalikan.
        </div>
        <?php endif?>
        <form class="form-horizontal" role="form"
method="post" action="<?php echo
site_url()?>/Cari/search">
            <div class="form-group">
                <label class="control-label col-sm-2"
for="noken">Nomor Kendaraan:</label>
                <div class="col-sm-2">
                    <input type="text" class="form-control"
name="nomor_kendaraan" value="<?php echo
set_value('nomor_kendaraan');?>">
                </div>
            </div>
            <div class="form-group">
                <label class="control-label col-sm-2"
for="nomen"></label>
                <div class="col-sm-2">
                    <input type="submit" class="form-control"
name="submit" value="Simpan">
                </div>
            </div>
        </form>

        <?php if ($result):?>
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>No.</th>
                    <th>No. Kendaraan</th>
                    <th>Nama Mobil</th>
                    <th>No. Mesin</th>
                    <th>Tahun Beli</th>

```

```

        <th>Status</th>
    </tr>
</thead>
<tbody>
<?php $i=1;?>
<?php foreach ($result as $row):?>
    <tr>
        <td><?php echo $i?>.</td>
        <td><?php echo $row-
>nomor_kendaraan?></td>
        <td><?php echo $row->nama_mobil?></td>
        <td><?php echo $row->nomor_mesin?></td>
        <td><?php echo $row->tahun_beli?></td>
        <td><?php echo $this->transaksi-
>getStatus($row->nomor_kendaraan)?></td>
    </tr>
    <?php $i++?>
<?php endforeach?>
</tbody>
</table>
<?php else: ?>
Tidak ada data
<?php endif?>
<?php $this->load->view('footer.php')?>

```

Sehingga hasilnya seperti pada gambar di bawah ini:

No.	No. Kendaraan	Nama Mobil	No. Mesin	Tahun Beli	Status
1.	D 2121 DO	Toyota Avanza	92832832	2010	Mobil Ada
2.	D 4567 FD	Toyot Agya	564322	2008	Mobil Sedang Dipinjam
3.	D 9998 FD	Toyot Xenia	664322	2007	Mobil Ada

Gambar 10. 17 Tampilan Tambah Data Pelanggan

Sampai dengan di sini kita sudah mengimplementasikan perancangan yang sudah dirancang pada bab sebelumnya, dan implementasinya sudah dilakukan dengan menggunakan Codeigniter. Anda dapat menambah fitur-fitur lain misalnya denda, dan lainnya sesuai dengan kebutuhan.

Kesimpulan

Untuk membuat aplikasi berbasis web dengan Codeigniter diperlukan persiapan perancangan yang matang, gunakan editor yang ringan misalnya Sublime agar proses programming lebih cepat. Untuk membuat aplikasi yang aman dibutuhkan system login agar tidak setiap orang dapat akses, gunakan framework CSS (misalnya Bootstrap) agar pembuatan layout lebih cepat dan mempercepat dalam proses pembangunan sebuah website.

Untuk hubungan dengan *database* gunakan model, baik itu untuk tambah, ubah, hapus, dan cari data ke tabel. Untuk pengelolaan (handling) dari form atau aksi dari tampilan gunakan controller, termasuk memberikan nilai ke tampilan gunakan controller. Untuk menampilkan data ke layout gunakan view.

Latihan

Setelah Anda mengimplementasikan perancangan pada bab sebelumnya, maka pada latihan ini Anda diharapkan mampu mengimplementasikan perancangan pada latihan di bab sebelumnya. Gunakan nama di htdocs yang berbeda dan *database* yang berbeda untuk implementasi latihan ini.

BAB XI

UPLOAD WEBSITE

Pada bab ini akan membahas bagaimana meng-*upload* website yang sudah jadi ke server, sehingga semua orang bisa mengakses website yang sudah kita buat. Selain itu dibahas juga tentang domain dan hosting serta bagaimana cara pemesanannya. Diharapkan setelah mempelajari bab ini pembaca dapat memesan domain dan hosting serta meng-*upload* web yang sudah dibuat.

11.1. Domain dan Hosting

Pada bagian ini akan dibahas tentang apa itu domain dan hosting, dan mungkin ini sudah menjadi bagian dari kehidupan kita, tetapi kita kurang menyadari secara pengertian dari hosting dan domain ini.

Domain

Domain adalah nama unik yang diberikan untuk mengidentifikasi nama server di internet. Nama Domain diberikan untuk memberikan kemudahan bagi pengguna internet untuk mengingat suatu komputer server di internet. Nama *server* biasanya menggunakan deretan nomor atau IP address, misalnya *server* Google mempunyai nomor IP 209.85.175.106 ini sangat menyulitkan untuk mengingatnya, oleh karena itu orang lebih mudah mengingat google.com dari pada 209.85.175.106. Jika kita akses di URL nomor IP dan google.com hasilnya akan sama.

Top Level Domain (TLD)

Top Level Domain Name adalah deretan kata di belakang nama domain seperti .com (*dot commercial*), .net (*dot network*), .org

(*dot organization*), .edu (*dot education*), .gov (*dot goverment*), dan .mil (*dot military*).

Ada dua macam Top Level Domain, yaitu Global Top Level Domain (gTLD) dan Country Code Top Level Domain (ccTLD). gTLD adalah seperti yang diungkapkan di atas dan ccTLD adalah TLD yang diperuntukkan untuk masing-masing negara, seperti Indonesia dengan kode ID (co.id, net.id, or.id) atau Singapura dengan kode SG (com.sg, net.sg, dsb).

Contoh TDL:

- .com : di gunakan untuk kepentingan komersial atau perusahaan.
- .net : di gunakan untuk kepentingan network infrastruktur.
- .org : di gunakan untuk kepentingan organisasi.
- .info : di gunakan untuk kepentingan informasional website.
- .name : digunakan untuk kepentingan keluarga atau personal.
- .edu : digunakan untuk kepentingan website pendidikan, terbatas hanya untuk pendidikan.
- .mil : di gunakan untuk kepentingan website angkatan bersenjata, terbatas hanya untuk Militer.
- .biz : di gunakan untuk kepentingan Bisnis.
- .tv : di gunakan untuk Entertainment seperti Televisi, Radio, majalah.
- .travel : di gunakan untuk Bisnis Pariwisata.

Second Level Domain (SLD)

Second Level Domain Name (SLD) adalah nama domain yang anda daftarkan. Misalnya nama domain yang anda daftarkan adalah domainku.com, maka domainku adalah SLD dan .com-nya adalah TLD.

Third Level Domain

Third Level Domain adalah nama sebelum Second Level Domain dan Top Level Domain. Misalnya nama domain yang anda

miliki adalah domainku.com, maka anda dapat menambahkan nama lain sebelum domainku, yaitu mail.domainku.com atau search.domainku.com.

Hosting

Hosting adalah suatu jasa penyediaan *server* dan aplikasinya kepada pelanggan-pelanggannya. Satu *server* ini dipakai untuk banyak *client* supaya biayanya terjangkau (*share hosting*) ada juga yang satu *server* dipakai oleh sendiri (*dedicated*). *Hosting* ini biasa dipakai untuk keperluan penyimpanan data web yang kita buat atau untuk kebutuhan *mail server*. *Web server* bisa dipakai sebagai *company profile*, *product catalog*, *gallery*, portal komunitas, portal berita, portal perdagangan dan sebagainya. *Mail server* dipakai sebagai sarana komunikasi elektronik antara seorang dengan yang lain baik dalam perusahaan yang sama maupun berbeda perusahaan.

Web server dan *mail server* dan diletakkan di dalam *server* yang sama, tetapi bisa juga dipisah menjadi dua buah *server*. Pemisahan maupun penggabungan tergantung dari kebijakan masing-masing penyedia jasa. Penggabungan memiliki keuntungan mudahnya *maintenance*, sedangkan pemisahan memiliki keuntungan dalam hal pengaturan *bandwidth*.

Ada beberapa jenis layanan *hosting* yaitu *shared hosting*, *VPS* atau *Virtual Dedicated Server*, *dedicated server*, *colocation server*.

1. *Shared Hosting* adalah menggunakan *server hosting* bersama sama dengan pengguna lain satu *server* dipergunakan oleh lebih dari satu nama domain. Artinya dalam satu *server* tersebut terdapat beberapa *account* yang dibedakan antara *account* satu dan lainnya dengan *username* dan *password*.
2. *VPS*, *Virtual Private Server*, atau juga dikenal sebagai *Virtual Dedicated Server* merupakan proses virtualisasi dari lingkungan *software* sistem operasi yang dipergunakan oleh

server. Karena lingkungan ini merupakan lingkungan virtual, hal tersebut memungkinkan untuk menginstal sistem operasi yang dapat berjalan di atas sistem operasi lain.

3. *Dedicated Server* adalah penggunaan *server* yang dikhususkan untuk aplikasi yang lebih besar dan tidak bisa dioperasikan dalam *shared hosting* atau *virtual dedicated server*. Dalam hal ini, penyediaan server ditanggung oleh perusahaan *hosting* yang biasanya bekerja sama dengan vendor.
4. *Colocation Server* adalah layanan penyewaan tempat untuk meletakkan *server* yang dipergunakan untuk *hosting*. *Server* disediakan oleh pelanggan yang biasanya bekerja sama dengan vendor.

11.2. Membeli Domain dan Hosting

Agar *website* kita dipublikasi dan semua orang bisa mengakses menggunakan internet, maka harus di simpan di *server* internet, selain di simpan di *server* juga kita harus memberi nama *server*-nya (domain). Untuk membeli domain dan *hosting* harus ke penyedia layanan internet atau dikenal dengan ISP (*Internet Service Provider*). Banyak sekali penyedia atau yang menjual domain dan hosting, Anda bisa melakukan pencarian dengan mengetik pada mesin pencarian (Google) dengan *keyword*: “jasa web hosting”, akan muncul banyak sekali.

Pada studi kasus pembelian domain dan hosting ini penulis mengambil salah satu penyedia ISP Domosquare <https://www.domosquare.com> yang beralamat di Kota Jogjakarta. Secara umum prinsip pemesanan di semua ISP sama, yaitu kita harus daftar terlebih dahulu kemudian melakukan order.

Pendaftaran

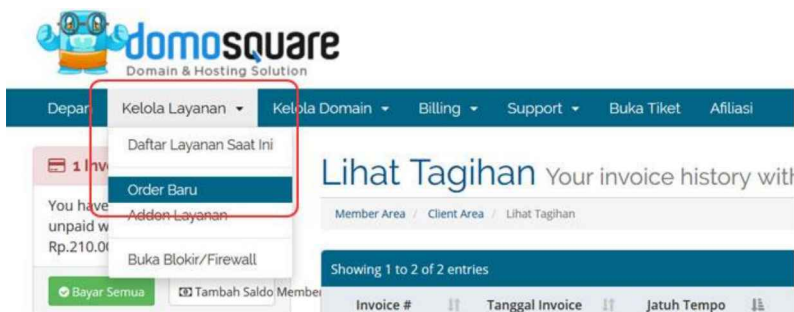
Berikut ini tahapan dalam melakukan pendaftaran account baru di domosquare.com

1. Buka website domosquare.com kemudian pilih **Member Login** dan kemudian pilih menu **Daftar** pada bagian bawah.



Gambar 11. 1 Tampilan domosquare.com

2. Muncul form registrasi, lakukan pengisian data Anda sampai dengan selesai. Jika sudah di-submit, maka akan ada pemberitahuan ke email untuk aktivasi member dan Anda sudah bisa login. Jika sudah login maka lakukan order untuk domain dan hosting baru pada menu seperti gambar di bawah ini:



Gambar 11. 2 Order Hosting dan Domain

3. Kemudian memilih paket yang sesuai dengan kebutuhan dan memasukkan nama domain yang diinginkan, setelah itu

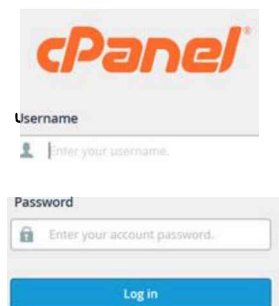
lakukan *checkout* dan *invoice* akan dikirim ke email untuk melakukan pembayaran.

4. Lakukan pembayaran, jika sudah konfirmasi pembayaran melalui email/sms/tlp ke pihak ISP maka *account* cPanel untuk akses ke *server* akan diberikan.
5. Jika kesulitan dalam proses order domain dan *hosting*, maka Anda bisa memanfaatkan fasilitas *chatting* yang ada di sudut kanan bawah, maka mereka akan melakukan order dan Anda cukup menunggu *invoice* pembayarannya.

Akses ke Server

Akses *web server* hanya bisa dilakukan jika Anda sudah punya *account* dari domain dan *hosting* yang sudah dibeli. Berikut ini contoh akses *web server* dengan domain yang sudah dibeli.

Untuk akses ke server ketik nama domain kemudian diberikan akhiran */cpanel*, misalnya nama domainnya budi.com maka akses ke *server*-nya adalah budi.com/cpanel akan tampil seperti gambar di bawah ini:



Gambar 11. 3 Akses ke Server via Website

Masukkan *username* dan *password* yang diberikan oleh ISP untuk *login* ke *server*, jika sudah *login* Anda akan masuk ke halaman cPanel atau *control panel* pengelolaan *hosting* dari *website*.

11.3. Upload File ke Server

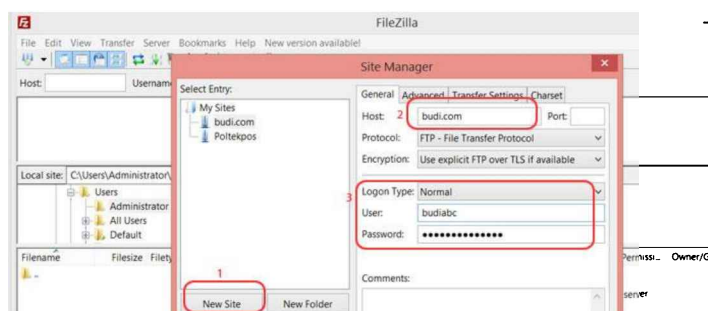
Untuk melakukan *upload* file ke *server* bisa menggunakan dua acara yaitu:

1. Melalui web browser dengan masuk ke cPanel pada menu File Manager.
2. Menggunakan aplikasi lain yaitu dengan aplikasi FileZilla

Yang paling banyak digunakan adalah cara ke-2, alasannya adalah satu kali aksi bisa meng-*upload* banyak file ke server. Untuk dapat menggunakan cara ke 2, harus menginstal terlebih dahulu aplikasi FileZilla. Aplikasi ini bersifat gratis dan dapat di download secara bebas dari internet.

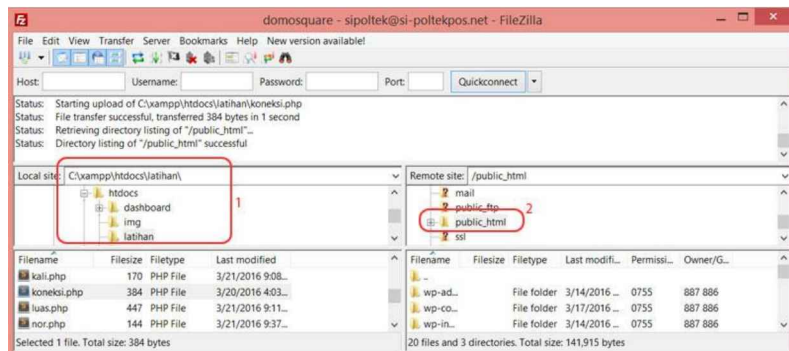
Berikut ini langkah-langkah upload menggunakan aplikasi FileZilla.

1. Pastikan aplikasi FileZilla sudah terinstal, buka aplikasi tersebut, kemudian seting aplikasi FileZilla agar dapat terkoneksi dengan server hosting kita, yaitu ke menu **File** → **Site Manager** → Pilih **New Site** pada bagian bawah, kemudian isi **Host** nya dengan nama domain lengkap misalnya budi.com, kemudian pada bagian **Logon Type** pilih normal dan isikan **User** dan **Password** sesuai dengan yang diberikan dari ISP.



Gambar 11. 4 Akses ke Server via Website

2. Jika sukses Anda sudah siap melakukan *upload* file atau website yang di-*local* kita ke *web server (hosting)* seperti pada gambar berikut ini:



Gambar 11.5 Akses ke Server via Website

- Arahkan folder atau file mana yang akan di upload ke server, pada nomor 1 di gambar di atas adalah diarahkan ke folder root di htdocs\latihan
 - Arahkan di *web server hosting* ke folder dengan nama public_html seperti terlihat pada nomor 2 di gambar di atas.
 - Untuk memulai upload klik kanan pada bagian folder yang akan di upload, misalnya di folder latihan (gambar nomor 1) kemudian pilih upload. Tunggu proses upload sampai dengan selesai.
3. Web sudah bisa diakses oleh public dengan nama budi.com dengan catatan budi.com tidak terhubung dengan *database*, kalau terhubung dengan *database* maka kita harus membuat *database* di server ISP nya

11.4. Membuat Database pada Server/Hosting

Kebanyakan *website* sekarang sudah menggunakan *database* dalam menyimpan datanya, sehingga kita harus menyesuaikan dengan *server hosting* kita. Oleh karena itu pada bagian ini akan dibahas bagaimana membuat *database* di *server hosting*.

1. Buka halaman cPanel dengan mengakses namadomain/cpanel misalnya budi.com/cpanel dan masukkan username dan password.
2. Pada bagian bawah (menu *database*) pilih *Database Wizard* seperti pada gambar di bawah ini:



Gambar 11. 6 Akses ke Server via Website

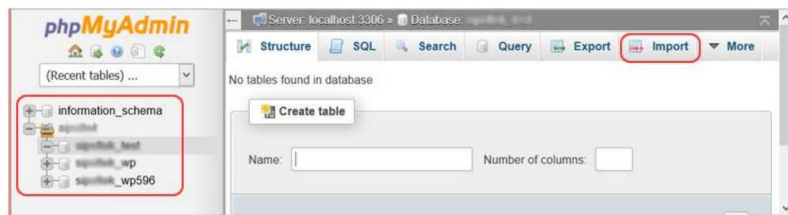
3. Selanjutnya lakukan pengisian sesuai dengan perintah yang diberikan dan harap dicatat apa yang diketikkan, karena konfigurasi *database* ini akan dipakai pada file koneksi php kita untuk koneksi ke *database* server. Jika menemukan tampilan Privilege seperti di bawah ini, lakukan *check* semuanya

☒ ALL PRIVILEGES	
☒ ALTER	☒ ALTER ROUTINE
☒ CREATE	☒ CREATE ROUTINE
☒ CREATE TEMPORARY TABLES	☒ CREATE VIEW
☒ DELETE	☒ DROP
☒ EVENT	☒ EXECUTE

Gambar 11. 7 Akses ke Server via Website

4. Setelah selesai maka kita akan mendapatkan username, password dan nama *database* yang kita buat pada menu *Database Wizard* ini yang akan kita cocokkan ke file koneksi.

5. Selanjutnya melakukan import *database* agar *database* di server hosting sama dengan di *database* local. Kembali ke cPanel masuk ke menu phpMyadmin di sebelah kanan *Database Wizard*.



Gambar 11. 8 Import Database

Pilih nama *database* yang sudah dibuat pada tahapan sebelumnya, kemudian pilih Import pada bagian kanan atas kemudian ambil file .sql nya untuk melakukan import.

6. *Database* local dan server sudah sama jika import *database* sudah sukses

Tahapan terakhir adalah melakukan konfigurasi file koneksi agar sesuai dengan *database* server, lakukan pada file *datadatabase.php* untuk di Codeigniter dan samakan username, password dan *database* sesuai dengan *Database Wizard*.

Kesimpulan

Agar bisa diakses oleh public, website harus disimpan pada server yang terhubung dengan internet dan mempunyai IP public, yaitu dengan membeli domain dan hosting ke penyedia domain dan hosting. Setelah melakukan pembelian maka akan diberikan akses ke server dengan username dan password.

Untuk upload file ke server hosting, dapat menggunakan aplikasi lain yaitu FileZilla agar memudahkan dalam proses uploading dalam jumlah yang banyak. Untuk konfigurasi *database* server harus masuk ke cPanel dengan memasukkan username dan

password yang didapat saat melakukan pembelian domain dan hosting.

Latihan

1. Upload file latihan kemudian seperti array dan lainnya, kemudian bagaimana cara akses langsung ke file pada server hosting.
2. Buat subdomain misalnya support.budi.com bagaimana langkah-langkah membuat sub domain.

DAFTAR PUSTAKA

- Agung Julisman, (2014). "Sistem Aplikasi Travel dengan Angulast JS dan Codeigniter". Penerbit Lokomedia.
- Awan Pribadi Basuki, (2011). "Membangun Web Berbasis PHP dengan Framework Codeigniter". Penerbit Lokomedia.
- Bernadus Herdi Sirenden (2012). "Buat Sendiri Aplikasi Petamu Menggunakan Codeigniter dan Google Maps API". Penerbit Andi Yogyakarta.
- Betha Sidik, (2012). "Framework Codeigniter". Penerbit Informatika Bandung.
- Budi Raharjo, (2015). "Belajar Otodidak MySQL (Teknik Pembuatan dan Pengelolaan *Database*)". Penerbit Informatika
- Bunafit Nugroho, (2004). "Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL (Studi Kasus, Membuat Sistem Informasi Pengolahan Data Buku)". Penerbit Gaya Media.
- R.H Sianipar, (2015). "Membangun Web dengan PHP dan MySQL untuk Pemula dan Programmer". Penerbit Informatika
- Wahana Komputer, (2011). "Mudah dan Cepat Membuat Website dengan Codeigniter". Penerbit Andi Yogyakarta.
- www.duniaikom.com."Tutorial Belajar PHP Part 39: Pengertian Fungsi (Function) PHP dan cara penggunaan Fungsi PHP". Diakses 2 September 2015
- <http://fandhie.com/tutorial-belajar-membuat-fungsi-php-php-function/>. "Tutorial: Belajar Membuat Fungsi PHP (PHP Function)".
- <http://getbootstrap.com/css>, "CSS Bootstrap Framework".

<http://imdi15.blogspot.co.id/2014/05/fungsi-echo-post-dan-get-pada-php-serta.html> "Fungsi ECHO, \$, POST dan GET pada PHP serta Pengenalan PHP".

<http://www.codeigniter.com/docs>, "Codeigniter Documentation".

http://www.codeigniter.com/user_guide, "User Guide Codeigniter".

<http://www.php.net/docs.php>, "PHP Documentation Official".

<http://www.w3schools.com/bootstrap>, "CSS Bootstrap Framework Tutorial".

BIOGRAFI SINGKAT

Supono, lahir di Garut, 21 Agustus 1983 dari keluarga sederhana. Setelah lulus SMA, kemudian melanjutkan program studi D3 di Politeknik Pos Indonesia dengan mengambil jurusan Teknik Informatika, lulus tepat waktu selama 3 tahun. Setelah itu melanjutkan program sarjana (S1) di Universitas Widyatama dengan mengambil jurusan yang sama dan lulus 3 tahun. Pendidikan terakhirnya adalah lulusan Program Magister (S2) Institut Teknologi Bandung dengan jurusan Informatika yang lulus tahun 2013.

Kegiatan sehari-hari dihabiskan dengan mengajar di perguruan tinggi Politeknik Pos Indonesia, dan memberikan bimbingan pelajaran di salah satu lembaga pendidikan (kursus) bernama Masagi di Bandung.

Virdiandry Putratama, biasa di panggil Virdi. Lahir di Pontianak pada tanggal 4 September 1986. Saya anak pertama dari dua bersaudara. Merantau ke Kota Bandung untuk menuntut ilmu di sebuah perguruan tinggi bernama Politeknik Pos Indonesia. Saya mengambil Jurusan Manajemen Informatika dengan gelar A.Md. Setelah lulus jenjang D3 tersebut saya melanjutkan ke jenjang S1 di Sekolah Tinggi Sains dan Teknologi Bandung selama 2 tahun. Pendidikan terakhir saya adalah Program Magister (S2) Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI yang telah saya selesaikan di tahun 2016

Sejak tahun 2012 sampai 2016 saya adalah tenaga pengajar di Politeknik Pos Indonesia, baru tahun 2016 saya diangkat sebagai dosen tetap Politeknik Pos Indonesia sampai sekarang

INFORMASI DOWNLOAD SOURCE CODE

- Anda dapat men-*download* semua latihan yang ada di buku ini dengan link download di bawah ini:
<https://dl.dropboxusercontent.com/u/1213312/books/php-codeigniter/source-code-buku-php-codeigniter.rar>
- Jika ada pertanyaan dapat mengirimkan email ke:
mr.supono@gmail.com

Pemrograman Web dengan menggunakan PHP dan FRAMEWORK CODEIGNITER

Perkembangan internet saat ini begitu pesat, terlihat dari jumlah pertumbuhan pengguna yang mengakses internet atau website serta meunculnya website-website baru. Faktor lain semakin yang membuat semakin pesatnya perkembangan internet dan website adalah infrastruktur dan semakin mudahnya akses terhadap internet. Hal inilah yang membuat banyak orang tertarik untuk masuk pada bidang IT atau dalam hal ini internet. Salah satu bidang yang menjadi profesi dibidang internet adalah Web Programmer.

Salah satu bahasa pemrograman yang populer untuk pembuatan website atau yang dipakai oleh seorang Web Programmer adalah PHP (*PHP: Hypertext Preprocessor*) karena kemudahan dalam mempelajari bahasa pemrograman ini dan bersifat gratis. Ditambah dengan banyaknya dukungan komunitas di internet, sehingga memudahkan bagi para Web Programmer untuk menemukan solusi jika mengalami kesulitan dalam mempelajari bahasa pemrograman PHP.

Agar seorang Web Programmer tidak membuat dari awal script program PHP, maka disarankan menguasai framework. Dengan framework, Web Programmer tidak harus membuat *script* dari awal. Framework sudah menyediakan fungsi atau *library* yang umum digunakan untuk membangun sebuah website, seperti fungsi koneksi ke database, fungsi pengolahan *string*, fungsi validasi dan lainnya.

Dari penjelasan diatas, penulis ingin memberikan sumbangsih dalam bentuk pemikiran yang dituangkan dalam bentuk buku ajar ini, dengan judul “Pemrograman Web Dengan Menggunakan PHP dan *Framework Codeigniter*”, yang mana diharapkan menjadi panduan dalam belajar bahasa pemrograman PHP dan *Framework Codeigniter*.



Penerbit Deepublish (CV BUDI UTAMA)
Jl. Rajawali, Gang Elang 6 No.3, Drono, Sardonoharjo, Ngaglik, Sleman
Jl. Kaliurang Km 9,3 Yogyakarta 55581
Telp/Fax : (0274) 4533427
Anggota IKAPI (076/DIY/2012)
cs@deepublish.co.id @penerbitbuku_deepublish
Penerbit Deepublish www.penerbitbukudeepublish.com

