

Pemrograman Web Dinamis Menggunakan PHP & MySQL
Oleh: M. Rudyanto Arief

Hak Cipta © 2011 pada Penulis

Editor : Gratia K
Setting : Sri Sulistiyan
Desain Cover : dan_dut
Korektor : Sigit AS / Akter Sudewa

Hak Cipta dilindungi undang-undang.

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis.

Penerbit: CV ANDI OFFSET (Penerbit ANDI)
Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282 Yogyakarta
55281

Percetakan: ANDI OFFSET
Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282 Yogyakarta
55281

Perpustakaan Nasional: Katalog dalam Terbitan (KDT)

Arief, M. Rudyanto

Pemrograman Web Dinamis Menggunakan
PHP & MySQL/M. Rudyanto Arief;

– Ed. 1. – Yogyakarta: ANDI,

20 19 18 17 16 15 14 13 12 11

xii + 452 hlm.; 16 x 23 Cm.

10 9 8 7 6 5 4 3 2 1

ISBN: 978 - 979 - 29 - 2759 - 7

I. Judul

II. Web Design

DDC'21 : 005.365

Kata Pengantar

Puji syukur ke hadirat Allah SWT, hanya dengan curahan rahmat dan hidayah-Nya buku ini dapat diselesaikan. Buku ini dibuat khusus untuk mahasiswa jurusan teknik informatika, manajemen informatika, sistem informasi, calon programmer web yang ingin mempelajari dan menambah pengetahuan tentang bagaimana konsep pemrograman web dan implementasinya. Bahasa pemrograman yang digunakan dalam contoh kasus listing program dalam buku ini adalah menggunakan bahasa pemrograman web PHP. Basis data yang digunakan adalah MySQL serta menggunakan web server Apache. Beberapa contoh kasus di dalam buku ini dibahas dengan sesederhana mungkin supaya pemula bisa mempelajari bahasan materi di dalamnya. Terdapat beberapa contoh program yang digunakan dalam buku ini yang diambil langsung dari beberapa sumber di internet seperti www.w3schools.com, www.ilmukomputer.com, <http://rudyantoarief.com> dan beberapa sumber lain yang dimodifikasi untuk menunjukkan beberapa contoh kasus penerapan materi di dalam buku ini. Jadi, diharapkan para pembaca dapat mengaplikasikan perintah-perintah PHP yang dibahas dalam buku ini untuk membuat aplikasi web dinamis yang melibatkan basis data MySQL. Dalam proses pembuatan buku ini penulis dibantu oleh banyak pihak, karena itu penulis mengucapkan terima kasih kepada:

1. Bapak Drs. M. Suyanto, M.M. selaku Ketua STMIK AMIKOM Yogyakarta yang selalu memberi spirit yang luar biasa agar penulis dapat menulis buku ini.
2. Rekan-rekan dosen, asisten praktikum dan mahasiswa di STMIK AMIKOM Yogyakarta yang selalu memberi masukan mengenai materi yang ada di dalam buku ini yang tidak dapat saya sebutkan satu per satu.
3. Mas Bima Aji Putra Pamungkas (bima@mandahost.com), Mas Tejo Murti (tejomurti@momohum.com), Edison P. Lase (edylase@yahoo.com), Ahlihi Masruro (masruro@gmail.com) yang telah membantu memberikan ide dan menyumbang beberapa contoh listing program untuk buku ini.

4. Terakhir penulis persembahkan buku ini untuk istriku "Neneng Sundarsih Djalla", anakku Muhammad Rayhan Al Farris dan Renata Daffa Azalia.

Penulis menyadari buku ini masih jauh dari sempurna, untuk itu sangat diharapkan saran dan kritik agar buku ini dapat penulis sempurnakan. Beberapa materi yang terdapat di buku ini juga diunggah di <http://rudyantoarie.com>. Korespondensi bisa dilakukan melalui alamat email berikut: arie_rudyanto@yahoo.com atau rudy@amikom.ac.id.

Yogyakarta, Juni 2011

M. Rudyanto Arief

Daftar Isi

KATA PENGANTAR.....	III
DAFTAR ISI	V
BAB 1 PENDAHULUAN	1
1.1 Pembahasan Materi Buku.....	1
1.2 Persiapan Program Aplikasi Pendukung	2
1.3 Perkembangan Web	4
1.4 Web 1.0.....	4
1.5 Web 2.0.....	5
1.6 Content Management System (CMS)	6
1.7 Pengertian Web	7
1.8 Pengertian Situs Web.....	8
1.9 Pengertian World Wide Web (WWW).....	8
1.10 Web Statis dan Web Dinamis.....	8
BAB 2 PEMROGRAMAN WEB	11
2.1 Client Side Scripting	11
2.1.1 Cara Kerja Client Side Scripting.....	12
2.1.2 Kelebihan Client Side Scripting	13
2.1.3 Kelemahan Client Side Scripting	14
2.2 Server Side Scripting	15
2.2.1 Cara Kerja Server Side Scripting.....	16
2.2.2 Kelemahan Server Side Scripting	17
2.2.3 Kelebihan Server Side Scripting	17
2.3 Komponen Penyusun Web	18
2.3.1 Bahasa pemrograman/ scripting language.....	18
2.3.2 Web Editor	19
2.3.3 Web Browser	19
2.3.4 Web Server	19
2.3.5 Database Server	20
2.3.6 Image Editor.....	21

BAB 3 HTML (HYPERTEXT MARKUP LANGUAGE)	23
3.1 Pengantar HTML	23
3.2 Definisi Elemen dan Tag HTML	23
3.3 Penggunaan Tag dan Atribut Tag	24
3.4 Tag-Dasar Html	25
3.5 List	25
3.6 Format Karakter	28
3.7 Font	30
3.8 Karakter Khusus	33
3.9 Grouping Element	34
3.10 Hyperlink	35
3.11 Tabel	36
3.12 Frame	37
BAB 4 PHP (PHP: HYPERTEXT PREPROCESSOR)	43
4.1 Pengantar PHP	43
4.2 Persiapan Dasar	43
4.3 Mengenal Perintah Dasar PHP	44
4.4 Komentar di PHP	45
BAB 5 VARIABEL DAN KONTANTA	49
5.1 Variabel PHP	49
5.2 Konstanta	49
BAB 6 TIPE DATA DAN OPERATOR DI PHP	53
6.1 Tipe Data di PHP	53
6.2 Operator di PHP	53
6.2.1 Operator Aritmatika	53
6.2.2 Penggabungan String	54
6.2.3 Operator Logika	60
6.2.4 Operator Bitwise	61
6.2.5 Operator Lain	62
BAB 7 STRUKTUR KENDALI	63
7.1 Struktur Kendali Percabangan	65
7.2 Struktur Kendali Pengulangan	65
	70

BAB 8 ARRAY	81
8.1 Pendeklarasian Array	81
8.2 Array Multi Dimensi	82
BAB 9 FUNGSI	89
9.1 Fungsi dengan Argumen	91
9.2 Fungsi Built-In	92
9.3 Include dan Require	111
BAB 10 FORM	113
10.1 Perintah Dasar Form	113
10.2 Elemen Input Form	114
10.3 Text Field	114
10.4 Password Field	115
10.5 Radio Button	115
10.6 Check Box	116
10.7 Submit Button	116
10.8 Reset Button	117
10.9 Select List (Drop-down List)	118
10.10 Textarea	119
10.11 Penggabungan Beberapa Elemen Form dalam Satu Halaman	119
10.12 Method Post dan Get	132
10.12.1 Metode Get	133
10.12.2 Metode Post	136
BAB 11 SESSION	143
11.1 Pengertian Session	143
11.2 Mengawali Session	143
11.3 Penggunaan Variabel Session	144
11.4 Menghapus Session	145
11.5 Contoh Penggunaan Session untuk Login	148
BAB 12 PENGANTAR MYSQL	151
12.1 Database, Tabel, Baris dan Kolom	151
12.2 Menggunakan MySQL	152
12.2.1 Struktur Direktori MySQL	152
12.2.2 Tipe Data pada MySQL	153

12.2.3 Mengaktifkan Database MySQL.....	155
12.2.4 Operasi-operasi MySQL.....	156
BAB 13 KONEKSI PHP DAN MYSQL.....	161
BAB 14 PEMBUATAN WEB SEDERHANA.....	167
14.1 Pembuatan Database ponselkita.....	167
14.2 Pembuatan Tabel Brand.....	170
14.3 Membuat Program PHP untuk Koneksi dan Pengolahannya.....	177
14.4 Prosedur Menampilkan Data Brand	180
14.5 Menampilkan Data dengan Format Tabel	185
14.6 Prosedur Menginputkan Data.....	188
14.7 Prosedur Mengubah Data.....	195
14.8 Proses Menghapus Data	203
BAB 15 STUDI KASUS 1 (APLIKASI TOKO BUKU ONLINE).....	211
15.1 Membuat Folder Kerja.....	212
15.2 Membuat Database di Database MySQL.....	212
15.3 Membuat Tabel-Tabel Pendukung di Database tokobuku_online ..	214
15.4 Pembuatan Halaman Web.....	231
15.5 Pembuatan Halaman Control Panel Admin	232
15.5.1 Folder Dashboard.....	232
15.5.2 Folder Penerbit	238
15.5.3 Folder Buku	247
15.5.4 Folder Order	266
15.5.5 Folder Login	277
15.5.6 Folder CSS	280
15.5.7 Folder Lang	306
15.6 Pembuatan Halaman Pengunjung/User	309
15.7 Manual Program Aplikasi Toko Buku Online	330
15.8 Pengelolaan Halaman Admin	331
15.8.1 Pengelolaan Data Penerbit	332
15.8.2 Pengelolaan Data Buku	336
15.8.3 Pengelolaan Halaman Order	337
15.9 Halaman Utama Pengunjung	338
15.10 Pembelian Buku	339
BAB 16 STUDI KASUS 2 (APLIKASI WEB DATABASE E-COMMERCE PONSELKITA/TOKO HANDPHONE ONLINE).....	347

16.1 Rancangan Database.....	348
16.2 Pembuatan Halaman Web	350
16.3 Struktur File PonselKita	405
DAFTAR PUSTAKA.....	407
LAMPIRAN 1 MANUAL INSTALASI XAMPP.....	409
LAMPIRAN 2 INTERNET INFORMATION SERVICES (IIS) WEB SERVER.....	417
LAMPIRAN 3 MANUAL PENGGUNAAN PHPMYADMIN.....	429

Pembahasan Misi Kisi Buku

Pembahasan misi kisi buku ini berfungsi memberi bantuan dalam menyelesaikan soal-soal yang diberikan pada bab dan modul. Pembahasan ini berisi penjelasan mengenai isi dan tujuan pembahasan yang diberikan pada misi kisi buku.

BAB 1

Pendahuluan

Media informasi saat ini berkembang pesat sekali. Sejak komputer ditemukan dan program aplikasi dibuat oleh banyak vendor, muncullah berbagai macam bentuk program aplikasi. Pada generasi awal dikenal program aplikasi berbasis desktop (*desktop based*), lalu di awal tahun 90-an ketika Internet mengalami perkembangan yang cukup pesat dengan banyaknya komputer yang terhubung ke Internet, banyak pula aplikasi dan layanan yang berjalan di Internet. Salah satunya adalah aplikasi berbasis (*web based application*). Seiring dengan itu semakin banyak perusahaan yang mulai melirik Internet sebagai salah satu media pemasaran yang cukup menjanjikan. Sehingga di awal tahun 2000-an banyak perusahaan yang berlomba melakukan promosi di Internet dengan membuat situs web perusahaan untuk tujuan promosi. Era ini disebut awal dari era web generasi 1.0 (web 1.0). Salah satu ciriannya adalah web ini relatif statis dari sisi tampilan dan interaksinya hanya bersifat satu arah. Interaksi terjadi hanya dari pengunjung web yang dapat memberikan komentar pada fasilitas buku tamu yang disediakan di halaman web tersebut.

Web merupakan salah satu bentuk aplikasi yang saat ini perkembangannya cukup pesat, mulai dari era web 1.0 kemudian berkembang web 2.0 dan saat ini sedang proses menuju era web 3.0. Akibat adanya perkembangan era web tersebut, maka secara tidak langsung konsep pemrograman web juga otomatis berkembang.

1.1 Pembahasan Materi Buku

Buku ini membahas semua aspek pemrograman web dari dasar, sehingga sangat cocok untuk pembaca yang baru mulai belajar web dan yang ingin menambah wawasan mengenai pemrograman web. Buku ini juga dirancang dengan sistematis agar dapat dimengerti dengan mudah.

Dalam buku ini materi disusun dengan sistematis dalam susunan bab-bab agar memudahkan dalam mempelajarinya. Walaupun susunannya dibuat terurut namun untuk mempelajarinya boleh tidak terurut. Jika ada

materi yang ingin langsung dipelajari boleh langsung menuju ke materi tersebut tanpa harus membaca materi di bab sebelumnya. Hal ini karena semua bab di buku ini disusun secara urut dan langkah demi langkah seperti model tutorial. Berikut adalah daftar materi yang dibahas di buku ini:

- Pengenalan konsep dasar web dan perkembangannya saat ini (Bab 1).
- Konsep pemrograman web yang meliputi jenis-jenis pemrograman, konsep web dinamis dan statis (Bab 2).
- Pengantar HTML (Bab 3).
- Materi PHP yang merupakan inti dari pemrograman web dinamis juga dibahas lebih detail dalam beberapa bab yang saling berhubungan satu sama lain (Bab 4, Bab 5, Bab 6, Bab 7, Bab 8, Bab 9).
- Materi FORM dibahas khusus di buku ini untuk membuat sebuah halaman web yang interaktif (Bab 10).
- Untuk mengatur keamanan di dalam sebuah halaman web maka digunakan konsep cookies dan session (Bab 11).
- Pengantar database MySQL (Bab 12).
- Fungsi untuk menghubungkan dan mengelola data di database MySQL melalui PHP (Bab 13).
- Pembuatan aplikasi web sederhana yang mampu mengelola (tampilkan data, mengisikan data, mengubah data, menghapus data) (Bab 14).
- Contoh Aplikasi toko buku online (Bab 15).
- Contoh Aplikasi toko handphone online (Bab 16).
- Manual instalasi XAMPP (Lampiran 1).
- Manual instalasi web server Internet Information Service (Lampiran 2).
- Manual cara penggunaan phpMyAdmin (Lampiran 3).

1.2 Persiapan Program Aplikasi Pendukung

Contoh listing program yang dibahas dalam buku ini dapat dicoba menggunakan komputer. Untuk itu didalam komputer harus terinstall beberapa aplikasi pendukung agar contoh listing program di dalam buku ini dapat

dijalankan dan dilihat hasilnya di komputer. Berikut adalah daftar program aplikasi yang harus terinstall di dalam komputer:

1. Sistem Operasi. Sistem operasi yang direkomendasikan adalah Microsoft Windows, Mac OS X, Linux. Selama pembuatan buku ini, semua listing program dibuat menggunakan komputer yang bersistem operasi Microsoft Windows XP atau Seven dan Mac OS X Snow Leopard.
2. Web Browser, setiap komputer yang memiliki sistem operasi Windows dan Mac OS X pasti memiliki web browser default atau browser yang diinstall terpisah. Contoh program di buku ini dicoba dan dijalankan hasilnya dengan menggunakan browser safari, Mozilla Firefox, Internet Explorer, Opera, dan semuanya berjalan dengan baik.
3. Web Server Apache. Saat ini Apache sudah dapat diinstall di banyak sistem operasi. Silakan unduh versi terbaru Apache dari situs resminya dan pilih Apache yang sesuai dengan sistem operasi yang digunakan. Pada buku ini menggunakan Apache for Windows dan Apache for Mac OS X.
4. Modul PHP. Saat ini di situs resmi PHP telah tersedia file master modul PHP untuk masing-masing sistem operasi. Silakan unduh masternya yang sesuai dengan sistem operasi yang Anda gunakan.
5. Database Server MySQL. Saat ini situs resmi MySQL sudah menyediakan master database MySQL yang dapat diunduh secara bebas untuk berbagai macam sistem operasi. Silakan unduh master file database MySQL yang sesuai dengan sistem operasi yang digunakan di komputer masing-masing.
6. XAMPP. Jika tidak mau repot untuk menginstall satu per satu software di atas (web server Apache, modul PHP, dan database MySQL), saat ini telah tersedia banyak sekali aplikasi yang mampu mengintegrasikan ketiga software tersebut dalam satu paket. Contohnya adalah software XAMPP. Software ini tersedia untuk sistem operasi Windows dan sistem operasi Mac OS X. Contoh program pada buku ini dibuat dengan menggunakan XAMPP versi Windows jika menggunakan sistem operasi Windows dan XAMPP versi Mac OS X jika menggunakan komputer yang menggunakan sistem operasi Mac OS X. Jika XAMPP sudah terinstall di dalam komputer maka tidak perlu lagi menginstall software web server Apache, modul PHP, dan database MySQL satu per satu. Ini karena di

dalam XAMPP semua komponen tersebut (web server Apache, modul PHP, database MySQL) sudah tersedia dan langsung siap untuk digunakan tanpa harus melakukan pengaturan tambahan. Penjelasan rinci mengenai cara instalasi dan cara pengaturan XAMPP dapat dilihat di Lampiran 1 di bagian akhir buku ini.

1.3 Perkembangan Web

Perkembangan web saat ini demikian cepat, hal ini disebabkan oleh beberapa faktor di antaranya perkembangan infrastruktur pendukung yang cukup pesat seperti internet, kebutuhan pelaku industri untuk menggunakan web dan Internet sebagai layanan tambahan dalam bisnis mereka, masyarakat yang memandang internet dan web saat ini sebagai kebutuhan dalam semua aspek kehidupan mereka khususnya masyarakat modern di perkotaan.

Tim Berners-Lee pada tahun 1989 menemukan metode baru dalam menghubungkan dokumen satu dengan lainnya di Internet. Selanjutnya metode ini dikembangkan terus sehingga menjadi sebuah web. Pada tahun 1993 web browser yang berbasis grafis ditemukan oleh Marc Andreessen, seorang peneliti di National Center for Supercomputing Applications. Nama web browser-nya adalah Mosaic. Pada tahun 1994, Marc Andreessen mengembangkan Mosaic menjadi sebuah web browser yang lebih baik dan dikenal dengan nama web browser Netscape. Pada saat itu juga Microsoft mengeluarkan produk web browser pertama mereka yaitu Internet Explorer. Saat ini perkembangan web dan relasinya dipercayakan pada sebuah lembaga nirlaba yang berpusat di Amerika, yaitu World-Wide-Web Consortium atau W3C (www.w3.org). Lembaga ini adalah tempat berkumpulnya peneliti dan pengambil kebijakan yang berhubungan dengan web di seluruh dunia. Lembaga ini juga memiliki sebuah situs web yang sering dijadikan referensi untuk belajar secara online (*e-Learning*) yaitu www.w3schools.com. Beberapa materi dan contoh program dalam buku ini banyak diambil dari situs web *e-Learning* W3C tersebut.

1.4 Web 1.0

Web jenis ini sebenarnya tidak berbeda dari web-web lain yang ditemukan di internet saat ini. Konsep web 1.0 pertama kali diperkenalkan oleh Tim O'reilly (www.oreilly.com) pada awal tahun 2000 ketika web mulai

berkembang dan menjadi fenomena pada saat itu. Web menjadi terkenal karena semua perusahaan mulai menggunakan media ini untuk mengembangkan bisnis mereka. Sehingga pada saat itu semua perusahaan membuat web dan berbisnis melalui internet (Era DotCom). Sebelumnya web hanya digunakan untuk kepentingan penyampaian informasi biasa. Pengguna internet menggunakan web hanya untuk kepentingan mencari informasi (*searching*) dan menjelajah internet tanpa tujuan yang jelas (*browsing*). Ketika era DotCom mulai berkembang dan perusahaan mulai berbisnis menggunakan media internet dan web maka muncullah konsep perdagangan elektronis yang disebut e-Commerce. Pada perjalannya ternyata perusahaan yang semula beramai-ramai membuat web mulai jenuh karena ternyata web tidak sebaik yang mereka harapkan. Respons masyarakat terhadap konsep bisnis ini (e-Commerce) ternyata tidak begitu baik karena ditemukan banyak kelemahan di antaranya isu yang berhubungan dengan keamanan ketika pengguna web melakukan transaksi perdagangan melalui web. Beberapa contoh aplikasi yang muncul pada saat itu adalah email, fasilitas mesin pencari, web e-Commerce (www.amazon.com).

1.5 Web 2.0

Web 2.0 diperkenalkan pertama kali oleh Tim O'Reilly dan Dale Dougherty. Sebenarnya secara teknologi web 1.0 dan web 2.0 tidak berbeda. Web 2.0 hanyalah sebuah konsep/cara berpikir terhadap perkembangan web. Konsep ini berawal dari munculnya banyak perusahaan sukses yang berhasil menerapkan teknologi web untuk bisnis mereka. Pendekatannya pun berbeda dari era DotCom yang berkembang pada tahun 2000-an. (Sumber: <http://oreilly.com/web2/archive/what-is-web-20.html>.) Berikut adalah contoh aplikasi web 1.0 dan web 2.0:

- Alta Vista vs. Google
- Hotmail vs. Yahoo Mail
- Mp3.com vs. iTunes
- Ofoto vs. Flickr
- MapQuest vs. Goggle Maps
- Encarta vs. Wikipedia
- Netscape vs. Firefox

- DoubleClick vs. Google AdSense
- Personal Websites vs. Blogging

Teknologi yang digunakan web 2.0:

- Open Data melalui API dan web services
- RSS
- Ajax
- Standarisasi web (XHTML, CSS)

Revolusi dalam Web 2.0

- Penggunaan teknologi yang telah ada sebelumnya dengan cara yang inovatif dan tergolong baru.
- Mengubah cara pandang orang/user terhadap penggunaan web.
- Industri TI yang telah matang.
- Hanya memerlukan lebih banyak inovasi untuk mengamankan posisi (perspektif bisnis).

1.6 Content Management System (CMS)

CMS adalah aplikasi web yang berisikan *template* untuk mengelola isi halaman web secara mudah. Penggunaan CMS tidak memerlukan pengetahuan pemrograman web yang andal karena proses instalasi dan cara penggunaannya sudah *user friendly*. CMS sendiri ada yang dibuat khusus menyesuaikan kasus yang ada (*customized*) dan biasanya berbayar dan ada yang berupa template instan yang fungsionalitasnya dibuat dengan menyesuaikan pada beberapa proses bisnis yang ada di dunia nyata yang dapat digunakan secara gratis. Aplikasi CMS instant yang banyak terdapat di internet saat ini kebanyakan dibuat menggunakan scripting language PHP dan database-nya adalah MySQL. Saat ini perkembangan CMS cukup pesat, banyak vendor yang membuat CMS instant yang didistribusikan secara gratis. Perkembangan CMS instant ini juga dipicu oleh perkembangan web 2.0 yang memungkinkan interaksi dalam arti yang cukup luas antara pengelola web dan pengunjung web. Selain perkembangan teknologi web dan infrastruktur internet, perkembangan pesat CMS juga dipicu oleh kebutuhan masyarakat dan pelaku bisnis yang menginginkan web dapat

mendukung kegiatan bisnis mereka secara mudah dalam hal pengelolaan content, cepat dalam pembuatan web, serta murah dalam pengadaannya. Bahkan saat ini, pengembangan aplikasi web CMS sudah banyak yang menggunakan konsep framework. Dengan konsep ini, sangat dimungkinkan untuk melakukan modifikasi terhadap beberapa fitur CMS instant tersebut karena semuanya dibuat lebih modular. Berikut adalah beberapa contoh web CMS instant yang dibuat oleh vendor dan dapat digunakan secara gratis dan scripting language yang digunakan adalah PHP dengan database server MySQL.

Ada beberapa jenis-jenis CMS, yaitu:

- CMS untuk membuat personal blog, contohnya Wordpress (www.wordpress.com)
- CMS untuk membuat web e-Commerce, contohnya: PrestaShop, OsCommerce, OpenCart, Drupal.
- CMS untuk membuat web e-Learning, contohnya: Moodle
- CMS untuk membuat personal web, contohnya: Joomla!, Mambo
- CMS untuk membuat e-Office, contohnya: Kantaya (Kantor Maya).
- CMS untuk membuat web e-Forum, contohnya: phpBB (www.phpbb.com)

Berikut adalah beberapa jenis framework berbasis scripting language PHP:

- Code Igniter (CI).
- RubyOnRails (RoR).
- CakePHP.
- FuseBox.

1.7 Pengertian Web

Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser. Beberapa jenis browser yang populer saat ini di antaranya: Internet Explorer yang

diproduksi oleh Microsoft, Mozilla Firefox, Opera, dan Safari yang diproduksi oleh Apple.

Browser (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen web dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat di dalam aplikasi browser yang biasa disebut web engine. Semua dokumen web ditampilkan oleh browser dengan cara diterjemahkan.

Situs Web adalah dokumen-dokumen web yang terkumpul menjadi satu kesatuan yang memiliki Unified Resource Locator (URL)/domain dan biasanya *di-publish* di internet atau intranet.

1.8 Pengertian Situs Web

Situs web merupakan kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain/URL (Uniform Resource Locator) yang dapat diakses semua pengguna internet dengan cara mengetikkan alamatnya. Berikut adalah contoh alamat situs web: www.amikom.ac.id, <http://rudyantoarief.com>.

1.9 Pengertian World Wide Web (WWW)

World wide web atau yang biasa disingkat WWW merupakan kumpulan situs web yang dapat diakses di internet yang berisi semua informasi yang dibutuhkan semua pengguna internet. Jadi antara web, situs web, dan WWW sebenarnya sama pengertiannya, hanya ruang lingkupnya yang berbeda.

1.10 Web Statis dan Web Dinamis

Ditinjau dari aspek content atau isi, web dapat dibagi menjadi 2 jenis, yaitu web statis dan web dinamis. Selain dari sisi content/isi, web statis dan web dinamis dapat dilihat dari aspek teknologi yang digunakan untuk membuat jenis web tersebut.

Web statis adalah web yang isinya/content tidak berubah-ubah. Maksudnya adalah isi dari dokumen web tersebut tidak dapat diubah secara cepat dan mudah. Ini karena teknologi yang digunakan untuk membuat dokumen web ini tidak memungkinkan dilakukan perubahan isi/data.

Teknologi yang digunakan untuk web statis adalah jenis client side scripting seperti HTML, Cascading Style Sheet (CSS). Perubahan isi/data pada halaman web statis hanya dapat dilakukan dengan cara mengubah langsung isinya pada file mentah web tersebut. Misalkan jika sebuah dokumen web dibuat menggunakan script HTML maka perubahan isi dilakukan dengan cara membuka dokumen file web tersebut (yang berekstensi HTML) lalu isinya diganti langsung di dokumen web tersebut. Untuk melakukannya diperlukan pengetahuan yang cukup tentang *client side scripting* atau dengan kata lain hanya dapat dilakukan oleh seorang programmer web yang menguasai perintah-perintah *client side scripting*. Jika dokumen web tersebut sudah diunggah di internet, maka perubahan dilakukan dengan cara membuka file dokumen web tersebut di komputer lokal secara offline untuk selanjutnya diganti isinya. Hasil perubahannya kemudian disimpan dan diunggah kembali ke internet. Jelaslah bahwa perubahan content pada web statis tidak mudah karena harus melalui beberapa proses yang hanya dapat dilakukan oleh seorang user yang memiliki pengetahuan teknis tentang bahasa pemrograman web seperti *client side scripting* dan memiliki pengetahuan tambahan mengenai jaringan komputer dan internet. Contoh web statis di antaranya adalah web profil perusahaan yang lebih dominan menggunakan animasi Flash atau HTML, web kumpulan produk animasi.

Web dinamis adalah jenis web yang content/isinya dapat berubah-ubah seiring saat. Web yang banyak menampilkan animasi flash belum tentu termasuk web dinamis karena dinamis/berubah-ubah isinya tidak sama dengan animasi. Untuk melakukan perubahan data, user cukup mengubah-isinya langsung secara online di internet melalui halaman control panel/administrasi yang biasanya telah disediakan untuk user administrator sepanjang user tersebut memiliki hak akses yang sesuai. Fitur yang disediakan untuk melakukan pengelolaan terhadap content/isi halaman web dinamis biasanya dibuat semudah mungkin, karena user yang akan melakukan perubahan data di halaman web tersebut kemungkinan bukanlah user yang menguasai detail teknis bahasa pemrograman dan database atau biasa diistilahkan user biasa/*naïve user*. Untuk membuat web dinamis diperlukan beberapa komponen yaitu client side scripting (HTML, JavaScript, Cascading Style Sheet), server side scripting seperti PHP, program basis data seperti MySQL untuk menyimpan data-datanya. Contoh web dinamis banyak sekali di internet, di antaranya: situs web berita, situs web e-Commerce, situs web e-Banking.

BAB 2

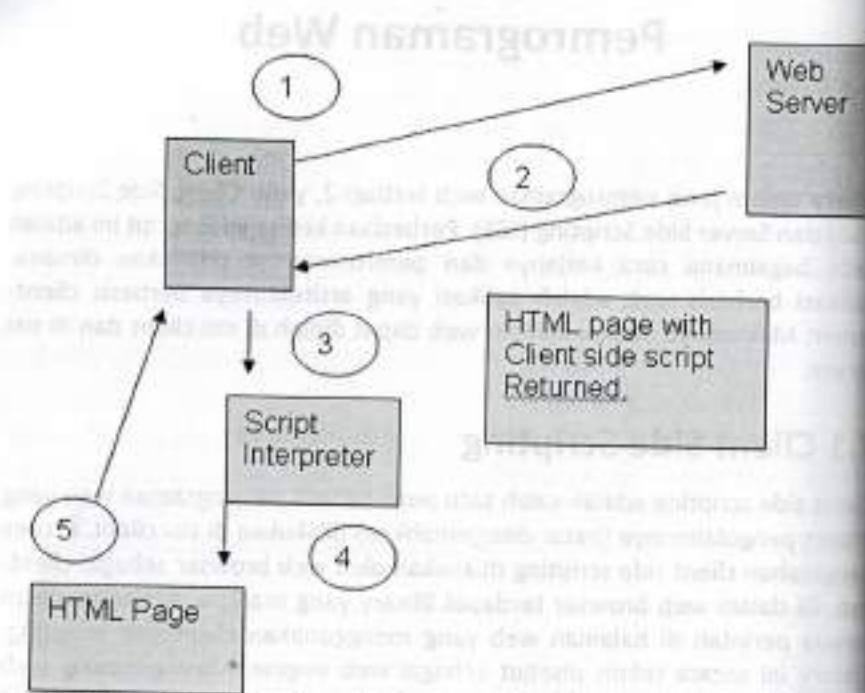
Pemrograman Web

Secara umum jenis pemrograman web terbagi 2, yaitu Client Side Scripting (CSS) dan Server Side Scripting (SSS). Perbedaan kedua jenis script ini adalah pada bagaimana cara kerjanya dan pemrosesannya dilakukan dimana. Aplikasi berbasis web adalah aplikasi yang arsitekturnya berbasis client-server. Maksudnya adalah aplikasi web dapat diolah di sisi client dan di sisi server.

2.1 Client Side Scripting

Client side scripting adalah salah satu jenis bahasa pemrograman web yang proses pengolahannya (baca: diterjemahkan) dilakukan di sisi client. Proses pengolahan client side scripting dilakukan oleh web browser sebagai clientnya. Di dalam web browser terdapat library yang mampu menerjemahkan semua perintah di halaman web yang menggunakan client side scripting. Library ini secara teknis disebut sebagai web engine. Masing-masing web browser memiliki web engine yang berbeda-beda. Itulah mengapa script yang sama dapat ditampilkan dengan layout berbeda-beda di web browser yang berbeda, karena masing-masing web engine menggunakan metode penerjemahan yang sedikit berbeda. Untuk menghindari hal tersebut maka gunakanlah web browser yang telah mengikuti standar dan telah disertifikasi oleh world wide web consortium (W3C). Selain itu, gunakanlah style penulisan perintah client side scripting yang standar sesuai dengan W3C. Misal untuk penulisan HTML gunakanlah style penulisan Extensible HyperText Markup Language (XHTML) yang sudah menjadi standar internasional. Berikut adalah contoh-contoh client side scripting: HyperText Markup Language (HTML), Extensible HyperText Markup Language (XHTML), Cascading Style Sheet (CSS), JavaScript, Extensible Markup Language (XML).

2.1.1 Cara Kerja Client Side Scripting



Gambar 2.1 Skema cara kerja client side scripting

(Sumber: <http://www.build-your-website.co.uk/Client-Scripting.htm>)

Berikut adalah cara kerja client side scripting:

1. Client/user melakukan request untuk mengakses sebuah dokumen web (contoh: <http://www.amikom.ac.id/index.htm> atau <http://localhost/index.htm>) melalui web browser yang ada di komputernya. Request user tersebut akan dicari di web server, tempat dokumen web tersebut tersimpan. Lokasi web server mungkin saja berada di internet atau di komputer lokal (localhost). Secara logik pencarinya dengan cara memanggil domain/URL dari web server tersebut.
2. Jika web server tempat dokumen tersebut tersimpan sudah ditemukan, maka web server akan melakukan pengecekan terhadap dokumen web yang di-request oleh user. Jika dokumen tersebut di dalamnya berisikan

client side scripting maka web server tidak akan melakukan pemrosesan apa pun terhadap dokumen web tersebut. Dokumen web tersebut langsung dikembalikan ke client dalam format halaman HTML, untuk selanjutnya diproses di sisi client (web browser).

3. Dokumen tersebut diterjemahkan oleh client/web browser. Komponen yang berfungsi menerjemahkannya adalah komponen script interpreter yang biasa juga disebut web engine.
4. Hasil pemrosesan (baca: terjemahan) dokumen tersebut adalah berbentuk halaman web dalam format HTML.
5. Halaman web yang telah selesai diproses akan dikembalikan kembali ke client dalam format halaman HTML yang dapat dilihat di halaman utama web browser, halaman web dalam format HTML.

Kesimpulannya adalah pada client side scripting pemrosesannya dilakukan di sisi client yang dalam hal ini komponen client-nya adalah web browser dan komponen servernya adalah web server. Konsep client-server di sini tidak harus dipisahkan secara fisik, yaitu harus ada komputer client dan harus ada komputer server yang terpisah dan dihubungkan melalui jaringan. Mungkin saja client-servernya adalah secara logik, yaitu komponen client (web browser) dan server (web server) berada di satu komputer yang sama (localhost). Prosesnya terpisah secara logik tetapi fisiknya berada dalam satu mesin/komputer yang sama.

2.1.2 Kelebihan Client Side Scripting

Berikut adalah beberapa alasan kelebihan jika menggunakan client side scripting:

1. Mudah untuk dipelajari dan digunakan, artinya untuk mempelajari client side scripting cukup mudah.
2. Tidak membutuhkan pengetahuan pemrograman yang tinggi atau pengalaman pemrograman yang cukup ahli.
3. Perubahan dan pemrosesan kode programnya lebih cepat karena dilakukan langsung di sisi client/komputer host tanpa perlu melakukan proses di sisi server melalui jaringan internet. Artinya kode sumber tersebut tidak perlu dikirim ke server melalui jaringan internet untuk diproses cukup dilakukan di browser yang terinstall di komputer client.

Tentunya hal ini dapat terjadi jika spesifikasi komputer host cukup tinggi untuk melakukan pemrosesannya karena semua kode sumber akan diproses menggunakan sumber daya yang ada di komputer host.

- Mampu menampilkan layout dan desain halaman web yang lebih interaktif dan user friendly. User dapat berinteraksi dengan halaman web melalui form isian yang disediakan.

2.1.3 Kelemahan Client Side Scripting

Berikut adalah beberapa alasan kelemahan jika menggunakan client side scripting:

- Kode programnya dapat dilihat melalui browser, sehingga dapat dikatakan tidak aman jika konteksnya adalah ingin melindungi kode sumbernya dari pihak lain. Melalui menu view page source yang ada di browser, maka user dapat melihat kode sumber dari dokumen web tersebut di internet.
- Karena pemrosesannya dilakukan di sisi client/komputer host maka semua sumber daya yang ada di komputer host tersebut (memory, CPU usage) akan digunakan secara maksimal. Hal ini mengakibatkan client side scripting sangat tergantung pada spesifikasi komputer host (*machine dependent*). Maksudnya, jika spesifikasi komputer host rendah (memory, CPU, storage media) maka dokumen web yang menggunakan client side scripting akan lambat diproses. Sebaliknya, jika spesifikasi komputer host tinggi maka pemrosesan dokumen webnya akan cepat.
- Masalah kompatibilitas web browser menjadi isu yang perlu diperhatikan. Karena beberapa web browser menggunakan web engine yang berbeda maka ada kemungkinan client side scripting akan diterjemahkan berbeda oleh masing-masing web browser tersebut. Jadi, sangat disarankan agar dokumen web yang dibuat menggunakan client side scripting diuji coba lebih dulu di web browser yang populer.
- Minim fitur untuk pengaksesan ke sumber daya komputer. Misalkan untuk menulis ke sebuah file di komputer, membaca isi file, membuat file/direktori di harddisk komputer, dan mengakses port-port di komputer tidak dapat dilakukan oleh client side scripting.

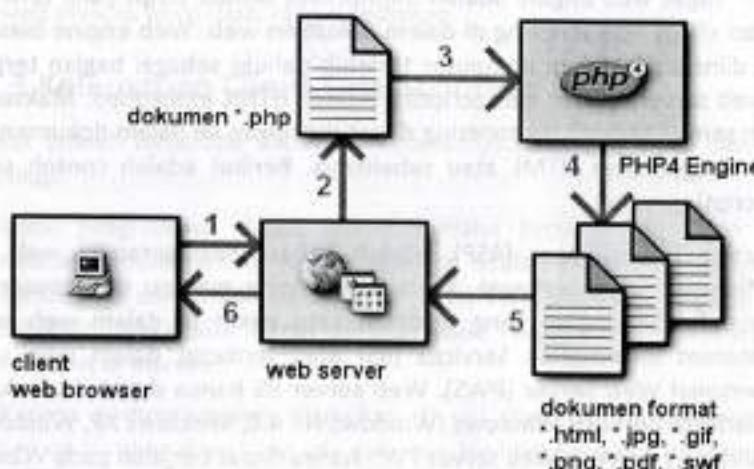
2.2 Server Side Scripting

Server side scripting adalah bahasa pemrograman web yang pengolahannya (baca: diterjemahkan) dilakukan di sisi server. Maksud server di sini adalah web server yang di dalamnya telah mengintegrasikan komponen web engine. Tugas web engine adalah memproses semua script yang termasuk kategori client side scripting di dalam dokumen web. Web engine biasanya harus diinstall di dalam komputer terlebih dahulu sebagai bagian terpisah dari web server. Server side scripting adalah HTML *embedded*. Maksudnya adalah semua server side scripting dapat disisipkan ke dalam dokumen web yang menggunakan HTML atau sebaliknya. Berikut adalah contoh server side scripting, yaitu:

- Active Server Pages (ASP), adalah bahasa pemrograman web milik Microsoft yang berbayar. Web engine yang mampu memproses ASP adalah ASP engine yang terdapat satu paket di dalam web server Internet Information Services (IIS) atau terdapat dalam web server Personal Web Server (PWS). Web server IIS hanya dapat dijalankan di platform berbasis Windows (Windows NT 4.0, Windows XP, Windows 7, Windows Server). Web server PWS hanya dapat berjalan pada Windows 95/98. Kedua aplikasi web server tersebut dapat ditemukan dalam CD Installer masing-masing Windows. Web server IIS dan PWS merupakan aplikasi tambahan yang harus diinstall tersendiri dan tidak otomatis terinstall ketika proses instalasi Windows dilakukan. Jenis script yang digunakan untuk ASP adalah VBScript dan Jscript (JavaScript yang dikembangkan secara khusus oleh Microsoft).
- PHP: Hypertext Preprocessor (PHP), adalah bahasa pemrograman web berbasis open source. Penemuannya adalah Rasmus Lerdorf pada tahun 1995. Pemrosesan script PHP dilakukan oleh PHP engine yang harus diinstall secara terpisah. Web server yang mendukung PHP adalah web server Apache dan web server IIS. Untuk web server IIS harus dilakukan beberapa pengaturan terlebih dahulu. Hal ini dikarenakan web server IIS produk buatan Microsoft sementara PHP adalah produk open source sehingga untuk bisa berjalan di lingkungan web server Windows maka PHP perlu diperkenalkan terlebih dahulu.
- Java Server Pages (JSP), adalah bahasa pemrograman web berbasis server side scripting yang berbasis platform Java. Jadi perintah-perintah

dalam JSP hampir sama dengan bahasa pemrograman Java. Web server yang mendukung JSP adalah web server Apache Tomcat.

2.2.1 Cara Kerja Server Side Scripting



Gambar 2.2 Cara kerja server side scripting (PHP)
(Sumber: www.google.com)

Berikut cara kerja server side scripting:

1. User melakukan request dari sisi client melalui web browser untuk mengakses sebuah halaman web (misal: www.amikom.ac.id).
2. Proses request ini akan dicari di web server, khususnya URL/domain halaman web tersebut. Lokasi dari dokumen web tersebut di web server mungkin saja berada di internet.
3. Jika dokumen tersebut sudah ditemukan, maka web server akan mengecek apakah dokumen yang di-request oleh user menggunakan client side scripting atau server side scripting. Jika dokumen web tersebut berisikan dokumen client side scripting maka web server akan mengarahkan pemrosesannya ke sisi client/web browser. Sebaliknya, jika berisikan server side scripting maka pemrosesannya diarahkan

kepada web engine. Pada Gambar 2.2, server side scripting-nya menggunakan PHP, maka pemrosesannya dilakukan oleh PHP engine/Zend engine.

4. Web engine akan memproses (baca: menerjemahkan) dokumen server side scripting tersebut menjadi format dokumen HTML.
5. Selanjutnya hasil pemrosesan web engine akan dikembalikan lagi ke web server.
6. Kemudian dilanjutkan oleh web server ke client/user yang me-request dokumen tersebut. Format akhir dokumen tersebut adalah format HTML. Hal ini terjadi karena web browser hanya dapat menampilkan halaman web dalam format HTML.

2.2.2 Kelemahan Server Side Scripting

Berikut adalah kelemahan server side scripting:

- * Karena semua pemrosesan dilakukan di sisi server maka dibutuhkan spesifikasi komputer server yang cukup tinggi agar dapat memproses server side scripting secara cepat. Karena itu dibutuhkan investasi yang tidak sedikit untuk pengadaannya.
- * Dibutuhkan kemampuan pemrograman yang baik untuk mempelajari server side scripting.
- * Tidak memiliki kemampuan untuk membuat layout/desain halaman web yang menarik.

2.2.3 Kelebihan Server Side Scripting

Berikut adalah keunggulan server side scripting:

- * Aman. Hal ini karena kode sumber server side scripting disimpan di web server yang ada di sisi server, sehingga user/pengunjung tidak dapat melihat kode sumber server side scripting dari sisi client/web browser. Hal ini berbeda dengan client side scripting yang dapat dilihat kode sumbernya dari sisi web browser.
- * Meminimalkan traffic di jaringan. Pada saat user melakukan request ke server maka yang dikirimkan ke user adalah hasil pemrosesannya saja. Karena pemrosesan dokumennya sudah dilakukan di sisi server, maka

- data yang mengalir dari client ke server atau sebaliknya adalah relatif kecil dan tidak membebani bandwidth di jaringan.
- Pemrosesannya lebih cepat karena spesifikasi hardware untuk mesin server biasanya lebih tinggi (bisa menjadi kelemahan juga).
 - Mampu mendukung banyak program basis data/database management system (DBMS).
 - Mampu mengelola sumber daya yang ada di komputer baik perangkat keras maupun perangkat lunak lainnya.
 - Mampu dijalankan di semua sistem operasi (*cross platform*).
 - Tidak bergantung pada jenis web browser yang akan digunakan, karena semua script dikelola di sisi server/web server.

2.3 Komponen Penyusun Web

Berikut ini adalah komponen-komponen yang dapat menyusun sebuah aplikasi web. Komponen ini identik dengan perangkat lunak dan bahasa pemrograman yang digunakan untuk membuat aplikasi web baik web statis maupun web dinamis.

2.3.1 Bahasa Pemrograman/*Scripting Language*

Secara garis besar ada 2 kategori script yang dibutuhkan untuk membuat aplikasi web, yaitu:

1. Client side scripting, contohnya adalah: Hypertext Markup Language (HTML), HTML5, XHTML, Cascading Style Sheet (CSS), JavaScript, VBScript, jQuery. Untuk client side scripting eksekusinya (baca: diterjemahkan) dilakukan di sisi client yaitu oleh web browser langsung. Ini karena di dalam web browser sudah terdapat library yang mampu mengenali semua perintah-perintah client side scripting. Library ini disebut web engine.
2. Server side scripting, contohnya adalah: Active Server Pages (ASP), PHP: Hypertext Preprocessor (PHP), Java Server Pages (JSP). Untuk server side scripting eksekusinya dilakukan di sisi server oleh sebuah modul yang disebut web engine. Contohnya jika menggunakan script PHP maka engine-nya adalah PHP engine/Zend engine. Jika menggunakan

ASP maka engine-nya sudah termasuk di dalam web server Internet Information Service (IIS) yang diinstall terpisah di Microsoft Windows.

Untuk membuat web statis maka script yang digunakan cukup jenis client side scripting. Untuk membuat web dinamis maka script yang digunakan adalah gabungan antara client side scripting dan server side scripting (pilih salah satu).

2.3.2 Web Editor

Merupakan program aplikasi yang berfungsi untuk mengetikkan perintah-perintah dokumen web baik client side scripting maupun server side scripting. Saat ini banyak tersedia web editor mulai dari yang paling sederhana hingga yang lebih *smart*. Mulai dari web editor yang berbayar hingga yang gratis. Contoh web editor adalah: Notepad, Notepad++, Macromedia Dreamweaver,

2.3.3 Web Browser

Merupakan program yang berfungsi untuk menampilkan dokumen-dokumen web dalam format HTML. Bagaimana halaman web yang dibuat ditampilkan sangat tergantung pada web engine yang digunakan oleh masing-masing browser. Semua jenis web browser yang ada saat ini mengikuti standarisasi yang dibuat oleh World Wide Web Consortium (W3C) yang merupakan badan independen yang mengurus semua hal yang berkaitan dengan web di dunia. Berikut adalah contoh web browser yang cukup populer beserta web engine yang digunakan:

- Web Engine WebKit: Safari, Google Chrome
- Web Engine Trident: Microsoft Internet Explorer, Maxthon.
- Web Engine Gecko: Mozilla Firefox.
- Web Engine Presto: Opera.

2.3.4 Web Server

Web server adalah program aplikasi yang memiliki fungsi sebagai tempat menyimpan dokumen-dokumen web. Jadi semua dokumen web baik yang ditulis menggunakan client side scripting maupun server side scripting tersimpan di dalam direktori utama web server (*document root*). Berikut

adalah beberapa contoh web server: web server Apache yang mendukung PHP, web server Microsoft Internet Information Service (IIS) yang mendukung ASP dan PHP, web server Apache Tomcat yang mendukung Java Server Pages (JSP). Port yang digunakan oleh web server di komputer default-nya adalah port 80 untuk web server Apache dan web server Microsoft Internet Information Services (IIS), sedangkan untuk web server Apache Tomcat menggunakan port 8080 untuk default-nya. Web server ada yang mendukung beberapa platform/sistem operasi, maksudnya adalah vendor pembuat web server tersebut menyediakan file instalasi web server untuk mendukung banyak sistem operasi dan ada yang hanya menyediakan dukungan file instalasi untuk satu sistem operasi saja. Kebanyakan web server yang berbasis open source seperti Apache mampu mendukung banyak sistem operasi. Sedangkan untuk web server yang berbayar biasanya tidak menyediakan dukungan untuk banyak sistem operasi. Web server Apache mampu berjalan di beberapa sistem operasi seperti Microsoft Windows, Linux, Solaris, Mac OS X. Web server IIS hanya dapat berjalan pada sistem operasi Microsoft Windows yang berbasis sistem file NTFS (Windows XP ke atas, Windows NT 4.0 ke atas). Web server IIS digunakan untuk mengolah script ASP.

Pembahasan lebih rinci mengenai web server dan cara konfigurasinya dibahas di Lampiran 2 di bagian akhir buku ini.

2.3.5 Database Server

Database server adalah program yang digunakan untuk menyimpan data yang akan diolah di halaman web. Database biasa disebut Database Management System (DBMS) adalah sebuah aplikasi yang menjembatani user dan data-data di dalam database. Dengan menggunakan DBMS user mampu mengolah data-data di dalam database secara mudah dan cepat. Arsitektur DBMS ada yang stand alone dan ada yang client-server. Untuk aplikasi web, jenis DBMS yang digunakan adalah yang berarsitektur client-server. DBMS merupakan komponen opsional dalam pembuatan aplikasi web. Maksudnya, keberadaan komponen ini di dalam komputer sangat tergantung kebutuhan dari web yang akan dibuat. Jika web tersebut menyimpan data-datanya di file dan bukan di database, maka database server tidak dibutuhkan. Sebaliknya, jika web yang dibuat butuh media penyimpanan data untuk membuat web menjadi lebih dinamis maka database server dibutuhkan. Untuk selanjutnya disebut database server.

Kenapa jenis DBMS yang digunakan harus database server, adalah karena aplikasi web adalah aplikasi client-server yang dapat diakses oleh banyak user dari berbagai tempat sepanjang ada koneksi internet. Untuk mendukung aplikasi web tersebut maka dibutuhkan aplikasi DBMS yang berarsitektur client-server juga. Database server/DBMS, ada yang berbasis *open source*/gratis dan ada yang *closed source*/berbayar. Ada beberapa hal yang perlu diperhatikan ketika akan menggunakan database untuk menyimpan data-data di halaman web:

- Jenis DBMS yang digunakan adalah yang berbasis client server/database server.
- Sistem operasi yang digunakan, banyak DBMS yang dapat mendukung banyak sistem operasi. Contoh di buku ini menggunakan MySQL versi Microsoft Windows.
- Open source atau Closed source
- File instalasi yang dipilih, apakah yang versi install (EXE) atau yang versi terkompress (ZIP/RAR).
- Platform prosesor, apakah berbasis 32 bit atau 64 bit.

Berikut adalah contoh DBMS server yang beredar di dunia nyata saat ini:

- Closed Source/berbayar: Microsoft SQL Server, Oracle 10g, DB2 (IBM)
- Open Source/gratis: MySQL, PostGreSQL

2.3.6 Image Editor

Image Editor adalah program aplikasi yang berfungsi untuk melakukan pengolahan gambar/image. Biasanya gambar dibutuhkan di halaman web untuk mempercantik halaman web. Selain program pengolah gambar, sebuah dokumen web seringkali disisipkan animasi di dalamnya sehingga membutuhkan program pengolah animasi juga. Saat ini banyak terdapat program pengolah gambar, video, dan animasi baik yang berbayar maupun yang gratis berbasis open source. Berikut adalah contoh program aplikasinya: Microsoft Paint, Adobe Photoshop, Adobe Flash, Adobe Premiere, Grab (Mac OS X), Corel Draw. Program tersebut tentunya bersifat pilihan/opsional yang keberadaannya sangat tergantung pada kebutuhan rancangan web yang akan dibuat. Jika dokumen web yang dibuat tidak menyertakan

gambar atau dokumen multimedia lainnya maka program aplikasi pengolah gambar/video/animasi mungkin saja tidak perlu diinstall di dalam komputer.

BAB 3

HTML (Hypertext Markup Language)

3.1 Pengantar HTML

HTML atau HyperText Markup Language merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web. Dokumen ini dikenal sebagai web page. Dokumen HTML merupakan dokumen yang disajikan pada web browser.

Pengenalan Kode HTML

```
<HTML>
</HTML>
```

Masing-masing baris di atas disebut tag. Tag adalah kode yang digunakan untuk me-mark-up (memoles) teks ASCII menjadi file HTML. Setiap tag diapit dengan tanda kurung runcing. Ada tag pembuka yaitu `<HTML>` dan ada tag penutup yaitu `</HTML>` yang ditandai dengan tanda slash (garis miring) di depan awal tulisannya. Tag di atas memberikan kaidah bahwa yang akan ditulis di antara kedua tag tersebut adalah isi dari dokumen HTML.

Dalam penamaan sebuah dokumen yang akan ditampilkan pada web browser maka nama yang digunakan harus diakhiri dengan ekstensi (.html) atau (.htm). Ekstensi dokumen HTML awalnya 3 karakter, adalah untuk mengakomodasi sistem penamaan dalam DOS.

Pengaturan warna di halaman HTML menggunakan mode kombinasi RGB (Red, Green, Blue) dimana setiap warna ditampilkan dalam dua digit nilai heksadesimal (0, 1, 2, ..., F). Setiap bagian dua digit kode menunjukkan banyaknya intensitas dari kombinasi warna Merah, Hijau, dan Biru.

Heksadesimal	Warna
#000000	Black
#FFFFFF	White
#FF0000	Red
#00FF00	Green
#0000FF	Blue
#FF00FF	Magenta

3.2 Definisi Elemen dan Tag HTML

Sebuah dokumen HTML disusun oleh beberapa elemen atau lebih dikenal dengan komponen-komponen dasar. Elemen dapat berupa teks murni, atau bukan teks, atau keduanya. Elemen atau komponen tersebut misal head, body, paragraf, list, dll.

Untuk menandai sebuah elemen dalam suatu dokumen HTML digunakan tag. Tag HTML terdiri dari sebuah [(<) (nama tag) (>)] contoh <h1>, tag pada umumnya berpasangan (misalnya <h1> dengan </h1>), tanda / pada tag pasangan memberikan tanda bahwa tag tersebut merupakan pembatas akhir elemen yang dibuka oleh tag awal. Jadi secara umum penulisan sebuah tag adalah <nama tag> ...</nama tag> selain itu dalam penamaan tag tidak menganut case sensitive.

Elemen dasar yang harus dimiliki dalam pembuatan dokumen HTML adalah tag html, tag body, dan tag head. Elemen head berisi informasi tentang dokumen, sedangkan elemen body berisi teks dan atau elemen lainnya. Lebih jelasnya adalah sebagai berikut:

```
<html>
<head>
....informasi dokumen....
</head>
<body>
....informasi yang di tampilkan
pada halaman browser.....
</body>
</html>
```

3.3 Penggunaan Tag dan Atribut Tag

Dalam pembuatan dokumen HTML, penulisan elemen yang diawali dan diakhiri tanda tag HTML memiliki syarat penulisan yaitu:

- Tag HTML diapit dengan dua karakter kurung sudut (< dan >)
- Tag HTML secara normal selalu berpasangan, misal (<i>...</i>)
- Tag html tidak 'case sensitive'. sama dengan
- Jika dalam suatu tag ada tag lagi, maka penulisan tag akhir tidak boleh bersilang, harus berurutan. Misalnya <i>..Tebal dan Miring..</i>

Tag dapat mempunyai atribut. Atribut menyatakan sesuatu tentang tag tersebut. Atribut digunakan untuk mengubah default pemformatan dokumen dengan tag yang bersangkutan.

Tag berikut tidak mempunyai atribut: <body>. Tag <body> ini tidak menggunakan atribut, sehingga dokumen HTML ditampilkan sebagaimana default dari tampilan HTML.

Tag <body bgcolor="red">. Maka tag <body> ini memiliki atribut berupa warna background merah. Secara umum tag dengan atributnya adalah

```
<namatag
atrl="nilai_atrl"atr2="nilai_atr2"....>isi</namatag>
```

3.4 Tag-Tag Dasar HTML

HTML

Merupakan tag dasar yang mendefinisikan bahwa dokumen adalah dokumen html. Tag ini sebagai tag pertama dalam dokumen html. Penulisan tag seperti berikut ini:

```
<html>
.....
</html>
```

Head

Merupakan tag berikutnya setelah tag html, digunakan untuk menuliskan keterangan tentang dokumen web yang akan ditampilkan. Penulisan tag seperti berikut ini:

```
<html>
<head>
.....
</head>
.....
</html>
```

Title

Merupakan tag di dalam head yang digunakan untuk menuliskan judul dari dokumen HTML, yang akan muncul pada caption halaman browser jika halaman tersebut diakses. Penulisan tag seperti berikut ini:

```
<html>
<head>
<title> Judul Dokumen </title>
</head>
.....
</html>
```

Body

Merupakan section utama dalam dokumen web. Pada section ini semua isi dokumen yang akan ditampilkan di dalam browser harus dituliskan. Penulisan tag seperti berikut ini:

```
<html>
<head>
<title> Judul Dokumen </title>
</head>
<body>
...isi dokumen...
</body>
</html>
```

Heading

Tag heading (hx) digunakan untuk memformat heading (judul dan subjudul) dari suatu halaman web. Ada enam buah heading yang dikenal HTML, yaitu dari (h1) sampai dengan (h6).

```
<html>
<head>
<title>Latihan 5</title>
</head>
<body>
<h1> Heading 1 </h1>
<h2> Heading 2 </h2>
<h3> Heading 3 </h3>
<h4> Heading 4 </h4>
<h5> Heading 5 </h5>
<h6> Heading 6 </h6>
</body>
</html>
```

Horizontal Ruler

Tag Horizontal ruler `<hr>` berfungsi untuk menampilkan garis horizontal tiga dimensi di dalam halaman web Anda. Tag horizontal ruler juga tidak memerlukan elemen penutup `</hr>`.

Atribut elemen horizontal ruler:

align :	[left center right] default center
size :	pixel (tebal garis, default 2)
width :	panjang (lebar garis, pixel atau persen, default 100%) noshade (garis solid)

Contoh:

```
<html>
<head>
<title>Horisontal Ruler</title>
</head>
<body bgcolor="#222222" text="#FF0000">
<h1> Ecommerce Book </h1>
```

```
<hr width="80%">
</body>
</html>
```

Paragraph

Tag paragraph `<p>` berfungsi layaknya untuk pengaturan antarparagraf dalam halaman web Anda. Dalam elemen paragraf terdapat atribut: `align=[left | center | right]` yang berfungsi sebagai pengatur perataan paragraf, jadi Anda cukup memilih salah satu dari ketiga pilihan tanpa harus memberi kurung buka dan tutup, dan default-nya adalah `left`.

contoh:

```
<html>
<head>
<title>Paragraph</title>
</head>
<body bgcolor="#222222" text="#FF0000">
<h1>Ecommerce Book</h1>
<hr width="85%" align="left">
<h4> Hot New Releases in Books </h4>
<p>Outliers: The Story of Success</p>
</body>
</html>
```

3.5 List

List Item merupakan perintah yang digunakan dalam HTML untuk membuat daftar atau mengelompokkan kata [list].

Ordered List (Numbering). Ordered list digunakan untuk membuat daftar dimana setiap bagiannya memiliki nomor secara terurut. Ordered list dimulai dengan tag `` dan diakhiri dengan ``. Untuk menyatakan tiap bagiannya digunakan tag `` dan tidak diperlukan tag penutup. Atribut elemen list yaitu:

`TYPE = [1 | a | A | i | I] { default 1 }`

contoh:

```
<html>
<head>
<title>Order List</title>
</head>
<body>
<h2>Kategori Buku</h2>
<ol type="a">
<li>Multimedia
<li>Database
<li>Pemrograman Web
<li>Ekonomi
<li>Hukum
</ol>
</body>
</html>
```

Unordered List (Bullet). Berbeda dengan ordered list, dalam unordered list tidak dijumpai urutan dalam suatu daftar. Setiap bagian dari unordered list ditandai dengan tanda bullet. Untuk membuat daftar dengan tanda ini digunakan tag awal `` dan tag penutup `` dan untuk tiap bagiannya digunakan tag `` tanpa tag penutup. Atribut elemen list yaitu:

`TYPE = [disc | square | circle] { default disc }`

```
<html>
<head>
<title>Unorder List</title>
</head>
<body>
<h2>Kategori Buku</h2>
<ul type=" circle">
<li>Multimedia
<li>Database
<li>Pemrograman Web
<li>Ekonomi
<li>Hukum
</ul>
</body>
</html>
```

Break. Tag Break `
` berfungsi untuk memberikan baris baru suatu paragraf dalam halaman web Anda. Tag break tidak memerlukan tag penutup break.

contoh:

```
<html>
<head>
<title>Break</title>
</head>
<body bgcolor="#222222" text="#FF0000" >
<h1>Ecommerce Book</h1>
<hr width="85%" align="left">
<h4> Hot New Releases in Books </h4>
<p>Outliers: The Story of Success<br>Against All
Enemies<br>Thirteen Reasons Why<br>The Walking Dead
Volume 14: No Way Out TP<br>Other</p>
</body>
</html>
```

3.6 Format Karakter

Format terhadap karakter yang ditampilkan akan sangat berguna untuk membuat sebuah dokumen HTML yang menarik. Pemformatan karakter tersebut mencakup beberapa hal:

Logical Format. Logical format akan menerapkan layout dokumen secara logis dan terstruktur. Tag-tag yang termasuk logical format adalah sebagai berikut:

- `<cite>`, digunakan untuk menandai suatu kutipan (Citation).
- `<code>`, digunakan untuk menampilkan kode-kode pemrograman, misalnya bahasa PHP.
- ``, digunakan untuk menandai suatu teks yang ditekankan oleh penulis.
- `<kbd>`, digunakan untuk menandai suatu teks yang harus dimasukkan oleh user melalui keyboard.
- `<samp>`, digunakan untuk menandai suatu teks yang dimaksudkan sebagai contoh.

- ``, digunakan untuk menandai bagian yang terpenting dari suatu teks.
- `<var>`, digunakan untuk menampilkan nama variabel.
- `<dfn>`, digunakan untuk menandai sebuah subdefinisi dari daftar definisi.

Semua tag tersebut memerlukan tag penutupnya masing-masing.

```
<html>
<head>
<title> Logical Format </title>
</head>
<body>
<cite> Albert Einstein</cite> adalah tokoh ilmuwan
Fisika. <br><br>
Untuk menampilkan isi variabel di pemrograman PHP
adalah :<code> echo "$content";</code> <br><br>
Saya berkata . "saya bisa<em>pasti bisa</em>."
<br><br>
Untuk melihat direktori, ketikkan <kbd>dir </kbd>.
<br><br>
Pada abjad terdapat 5 huruf vokal, yaitu
<samp>AEIUIO</samp>. <br> <br>
Kata terakhir yang diucapkannya adalah <strong>Aku
Mencintaimu!</strong> <br><br>
Kode program tersebut diiterasi sebanyak
<var>N</var> kali . <br><br>
<dfn>Kambing adalah hewan pemakan rumput. </dfn>
</body>
</html>
```

Physical Format. Physical format adalah format terhadap fisik suatu font. Tag-tag yang termasuk physical format adalah sebagai berikut:

- ``, untuk menampilkan huruf tebal.
- `<i>`, untuk menampilkan huruf miring.
- `<u>`, untuk menampilkan garis bawah pada teks.
- `<tt>`, untuk menampilkan huruf seperti huruf mesin ketik.
- `<strike>`, untuk menampilkan garis horizontal pada bagian tengah huruf.

- f. `<big>`, untuk menampilkan ukuran huruf yang lebih besar.
- g. `<small>`, untuk menampilkan ukuran huruf yang lebih kecil.
- h. `<sub>`, untuk menampilkan subscript.
- i. `<sup>`, untuk menampilkan superscript.

```

<html>
<head>
<title> Memformat Karakter </title>
</head>
<body>
<b> kalimat ini akan dicetak tebal. </b><br>
<i> kalimat ini akan dicetak miring </i><br>
<u> kalimat ini akan bergaris bawah </u><br>
<strike> kalimat ini akan bergaris tengah
</strike><br>
<tt>menampilkan huruf seperti huruf mesin
ketik.</tt><br>
Untuk menampilkan <sub>subscript </sub><br>
Untuk menampilkan <sup>superscript </sup>
</body>
</html>

```

Tag `<BLOCKQUOTE>`. Untuk membuat indentasi paragraf, kita bisa menggunakan `BLOCKQUOTE`. Dengan perintah `<Blockquote>` browser akan menampilkan teks menjorok ke dalam (mengindentasi teks) atau menampilkan teks.

```

<html>
<head>
<title>Quotes/indentasi</title>
</head>
<body>
<font color="#222222" size="7">Quote atau
Indentasi</font>
<p>
Dengan HTML, Anda dapat mengatur indentasi text yang
Anda buat dengan menggunakan tag BLOCKQUOTE
<blockquote>Ini adalah paragraf yang menggunakan
blockquote, Anda dapat melihat perbedaanya ketika
script ini Anda jalankan.

```

`</blockquote>`

```

Nah, Anda sudah mengetahui perbedaannya kan??Jadi,
belajar HTML itu tidak sulit dan Anda bisa mencoba
banyak hal disini. </p>
</body>
</html>

```

Preformatted Text. Preformatted text `<PRE>` digunakan untuk menampilkan teks sama seperti yang Anda ketikkan dalam dokumen HTML. Browser akan menampilkan teks tersebut dalam font monospaced, yaitu seperti terlihat dalam teks yang ditampilkan oleh aplikasi berbasiskan teks.

Sebagai contoh, cobalah latihan berikut ini:

```

<html>
<head>
<title>Formatting Teks</title>
</head>
<body>
<font color="#222222" size="6">
Hencoba PreFormatted Text
</font>
<p>
<pre>
menampilkan teks sama seperti yang Anda ketikkan
dalam dokumen HTML
</pre>
</p>
</body>
</html>

```

3.7 Font

Dengan tag `` Anda bisa menentukan format tampilan font dalam dokumen HTML seperti color, size, style dan lainnya.

3.7.1 Ukuran Font

Atribut SIZE digunakan untuk mengatur ukuran font. Nilai font dimulai dengan nilai 1 untuk ukuran huruf terkecil dan nilai 7 untuk ukuran paling besar.

b. Jenis Font

Atribut FACE digunakan untuk mengatur jenis huruf yang diinginkan. Atribut FACE harus diisi dengan string jenis font seperti Arial, Times New Roman, Verdana dan lain sebagainya.

c. Warna Font

Atribut COLOR digunakan untuk mengatur warna font yang diinginkan. Untuk memberi nilai pada atribut color ada dua cara, yaitu:

1. Dengan menyebutkan nama warna seperti red, green dan blue.
2. Dengan menggunakan nilai RGB (Red Green Blue) dari satu warna misalnya FF0000 untuk red, 00FF00 untuk green dan 0000FF untuk Blue.

```
<html>
<head>
<title>Formatting Teks</title>
</head>
<body>
<p><font color="#FF0066" size="7">
Menggunakan Font dan Atributnya</font></p>
<p>
<font face="Verdana"> Teks ini menggunakan atribut
FACE</font><br>
<font size="+3">Teks ini menggunakan atribut
SIZE</font><br>
<font color="#990000">Teks ini menggunakan atribut
COLOR</font>
</p>
</body>
</html>
```

3.8 Karakter Khusus

Tanda lebih kecil (<) dan lebih besar (>) merupakan tag HTML, lalu bagaimana cara untuk menuliskan tanda tersebut ke dalam dokumen web kita? Caranya adalah dengan menuliskan entitas karakter yang mempunyai 3 bagian: sebuah ampersand (&), sebuah nama entitas atau sebuah # dan nomor entitas, dan sebuah titik koma. Contoh:

Untuk menampilkan tanda lebih kecil kita harus menuliskan: < atau <

Baasil	Nama Entitas	Nomor Entitas	Keterangan
&	©	©	Copyright
&	®	®	Registered Trademark
*	×	×	Tanda kali
+	±	±	Plus minus

```
<html>
<head>
<title>Memformat Karakter</title>
</head>
<body>
ampersand &amp;lt;br>
laquo; angle quotation mark (left)<br>
angle quotation mark (right) &raquo; <br>
quot;tanda kutip &quot;<br>
lebih kecil &lt;<br>
lebih besar &gt;<br>
tanda kali &times;<br>
tanda bagi &divide;<br>
copyright &copy;<br>
registered trademark &reg;<br>
trademark &#8482;<br>
non-breaking &nbsp;&nbsp; space<br>
</p>
</body>
</html>
```

3.9 Grouping Element

Tag DIV dan SPAN digunakan untuk mengelompokkan element-element HTML. Span digunakan untuk mendefinisikan inline content, sementara div digunakan untuk blocklevel content.

```
<html>
<head>
<title>Div dan Span</title>
</head>
<body>
```

```

<div>
Divisi 1
<p>
Div tag digunakan untuk mengelompokkan group element
biasanya untuk blocklevel
element.
</p>
</div>
<div align="center">
Divisi 2
<p>
paragraf ini ditulis dengan alignment center.
</p>
</div>
<span style="font:verdana; color:#FF0000; font-
size:20">
baris ini dalam span dengan warna merah.
</span>
</body>
</html>

```

3.10 Hyperlink

Dengan hyperlink ini berbagai halaman web yang berbeda dapat saling dihubungkan. Untuk membuat link Anda dapat menggunakan perintah anchor (). Untuk membuat link ke dokumen HTML lain digunakan perintah berikut:

` Teks pada browser `

Bisa juga membuat link dalam sebuah dokumen HTML. Untuk itu perlu disediakan nama anchor lokasi tujuan dari link tersebut. Fungsi anchor sangat berguna jika dokumen HTML yang Anda buat memuat banyak informasi. Nama anchor dibuat dengan menambahkan atribut "name".

Contoh:

` HTML dan CSS `

Cara melakukan link ke bagian tersebut yaitu:

` Teks pada browser `

3.11 Tabel

Tabel banyak digunakan karena dapat menampilkan informasi dengan bentuk yang ringkas dan mudah dibaca. Untuk membuat tabel digunakan tag awal `<table>` dan tag penutup `</table>`.

Tag `<table>` memiliki beberapa bagian penting, yaitu:

- a. `<caption>...</caption>` digunakan untuk membentuk judul tabel. Judul tabel ini akan terletak di luar tabel, bisa di bagian atas atau di bagian bawah.
- b. `<th>...</th>` digunakan untuk meletakkan judul tabel di bagian paling atas atau paling kiri dari suatu tabel. Tabel header akan dicetak dalam huruf tebal.
- c. `<tr>...</tr>` dipakai untuk membentuk baris pada suatu tabel.
- d. `<td>...</td>` digunakan sebagai tempat menulis data atau informasi dalam tabel.

```

<!DOCTYPE html>
<html>
<head>
<title>Tabel</title>
</head>
<body>
<table width="363" border="1" bordercolor="#0000FF">
<caption>Membuat Tabel</caption>
<tr>
<td width="120">Baris 1 Kolom 1</td>
<td width="110">Baris 1 Kolom 2 </td>
</tr>
<tr>
<td>Baris 2 Kolom 1 </td>
<td>Baris 2 Kolom 2 </td>
</tr>
<tr>
<td>Baris 3 Kolom 1 </td>
<td>Baris 3 Kolom 2 </td>
</tr>
</table>
</body>
</html>

```

Atribut untuk tag `<table>` adalah sebagai berikut:

- ALIGN: Posisi horizontal tabel
- BACKGROUND: Menentukan gambar latar belakang tabel
- BGCOLOR: Menentukan warna latar belakang tabel
- BORDER: Menentukan tebal bingkai tabel
- BORDERCOLOR: Menentukan warna bingkai tabel
- BORDERCOLORLIGHT: Menentukan warna depan bingkai tabel
- BORDERCOLORDARK: Menentukan warna bayangan bingkai tabel
- CELLSPACING: Menentukan jarak spasi antarsel
- CELLPADDING: Menentukan jarak isi sel dengan bingkai
- HEIGHT: Menentukan tinggi tabel
- WIDTH: Menentukan lebar tabel
- VALIGN: Menentukan posisi vertikal teks dalam sel

Atribut untuk tag `<td>` adalah:

- ALIGN: Posisi horizontal teks dalam sel
- BACKGROUND: Menentukan gambar latar belakang sel
- BGCOLOR: Menentukan warna latar belakang sel
- BORDER: Menentukan tebal bingkai sel
- BORDERCOLOR: Menentukan warna bingkai sel
- BORDERCOLORLIGHT: Menentukan warna depan bingkai sel
- BORDERCOLORDARK: Menentukan warna bayangan bingkai sel
- COLSPAN: Menentukan jumlah kolom yang digabung
- HEIGHT: Menentukan tinggi sel
- NOWRAP: Menentukan teks agar tetap satu baris
- ROWSPAN: Menentukan lebar tabel
- VALIGN: Menentukan posisi vertikal teks dalam sel

3.12 Frame

Frame digunakan untuk membagi jendela browser menjadi beberapa bagian dan masing-masing bagian terdiri dari dokumen HTML tersendiri. Frame pertama kali diperkenalkan oleh Netscape Navigator 2.0.

Untuk membuat sebuah frame, tag yang digunakan adalah:

- Tag `<frameset>` mempunyai beberapa atribut:
 - rows: Membuat frame secara mendatar (baris) sekaligus mendefinisikan lebar masing-masing.
 - cols: Membuat frame secara vertical (kolom) sekaligus mendefinisikan lebar masing-masing.
 - bordercolor: Mendefinisikan warna bingkai frame.

Untuk mendefinisikan lebar frame pada atribut ROWS dan COLS ada tiga cara yang dapat digunakan, yaitu dengan nilai tetap, nilai persentase, dan nilai proporsional. Misalnya ingin dibuat tiga buah frame, maka penggunaannya adalah sebagai berikut:

- ✓ Dengan nilai tetap
`<frameset rows="100,240,400">`
- ✓ Dengan nilai persentase
`<frameset cols="30%,40%,30%">`
- ✓ Dengan nilai proporsional
`<frameset rows="*,2*,3*">`

Dengan nilai tetap, setiap frame didefinisikan dalam ukuran pixel. Cara ini kurang baik karena jika resolusi monitor user tidak sama dengan resolusi monitor pada saat perancangan frame, maka frame yang terbentuk akan menjadi kacau. Dengan nilai persentase, lebar setiap frame didefinisikan sebagai persentase dari lebar jendela browser sehingga resolusi monitor tidak akan berpengaruh. Dengan nilai proporsional, lebar setiap frame juga didefinisikan sebagai persentase dari lebar jendela browser, hanya saja pembiacanya agak berbeda, misal nilai proporsionalnya adalah `*,2*,*` jumlah bintang seluruhnya adalah $1+2+1=4$, frame pertama akan memiliki lebar $\frac{1}{4}$ dari jendela browser, frame kedua $\frac{2}{4}$ atau $\frac{1}{2}$ dari jendela browser, dan frame ketiga $\frac{1}{4}$ dari jendela browser.

b. Tag <frame> mempunyai beberapa atribut:

- src: Menentukan nama file HTML yang akan digunakan sebagai isi frame.
- marginheight: Menentukan batas atas dan bawah antara dokumen dengan bingkai dalam pixel.
- marginwidth: Menentukan batas kiri dan kanan antara dokumen dengan bingkai dalam pixel.
- scrolling: Menentukan apakah frame dapat memiliki scroll bar. Nilainya Yes, No, Auto.
- noresize: Jika atribut ini disebutkan, maka ukuran frame tidak dapat diubah.
- name: Mendefinisikan nama dari objek frame; bisa sebagai tujuan link.
- border: Menentukan ukuran bingkai frame.

c. Tag <noframe>, digunakan untuk menampilkan sebuah alternatif dokumen jika browser tidak mendukung penggunaan frame. Sebuah frame dapat didefinisikan di dalam frame lain.

Contoh:

```
<html>
<head>
<title>Frame Kiri</title>
</head>
<body bgcolor ="#222222">
<h1>Frame sebelah kiri </h1>
</body>
</html>
```

Simpan dengan nama: framekiri.html

```
<html>
<head>
<title>Frame Kanan</title>
</head>
<body bgcolor ="#333333">
<h1>Frame sebelah kanan </h1>
</body>
</html>
```

Simpan dengan nama: framekanan.html

```
<html>
<head>
<title>Contoh Penggunaan Frame</title>
</head>
<frameset cols=30%,**>
<frame src="framekiri.html">
<frame src="framekanan.html">
</frameset>
</html>
```

Simpan dengan nama: frame.html

BAB 4

PHP (PHP: Hypertext Preprocessor)

4.1 Pengantar PHP

PHP (PHP: Hypertext Preprocessor) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membentuk halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web.

PHP termasuk dalam *Open Source Product*, sehingga *source code* PHP dapat diubah dan didistribusikan secara bebas. Versi terbaru PHP dapat diunduh secara gratis di situs resmi PHP: <http://www.php.net>. PHP juga dapat berjalan pada berbagai web server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami. PHP juga mampu lintas *platform*. Artinya PHP dapat berjalan di banyak sistem operasi yang beredar saat ini, di antaranya: Sistem Operasi Microsoft Windows (semua versi), Linux, Mac OS, Solaris. PHP dapat dibangun sebagai modul pada web server Apache dan sebagai binary yang dapat berjalan sebagai CGI (*Common Gateway Interface*). PHP dapat mengirim HTTP header, dapat mengatur cookies, mengatur authentication dan redirect users.

Salah satu keunggulan yang dimiliki oleh PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam software sistem manajemen basis data/Database Management System (DBMS), sehingga dapat menciptakan suatu halaman web yang dinamis. PHP mempunyai koneksi yang baik dengan beberapa DBMS antara lain Oracle, Sybase, mSQL, MySQL, Microsoft SQL Server, Solid, PostgreSQL, Adabas, FilePro, Velocis, dBase, Unix dbm, dan tak terkecuali semua database ber-interface ODBC. PHP juga memiliki integrasi dengan beberapa library eksternal yang dapat membuat

Anda melakukan segalanya dari dokumen PDF hingga mem-parse XML. PHP mendukung komunikasi dengan layanan lain melalui protokol IMAP, SNMP, NNTP, POP3 atau bahkan HTTP. Bila PHP berada dalam halaman web Anda, maka tidak lagi dibutuhkan pengembangan lingkungan khusus atau direktori khusus. Hampir seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan utama adalah konektivitas basis data dengan web. Dengan kemampuan ini kita akan mempunyai suatu sistem basis data yang dapat diakses dari web.

4.2 Persiapan Dasar

Sebelum memulai untuk mencoba dan mengetikkan contoh-contoh perintah PHP yang disertakan sebagai contoh di buku ini, ada beberapa hal yang perlu dipersiapkan/diinstall di dalam komputer, di antaranya:

- Sistem Operasi Windows. Contoh-contoh kode program di buku ini dibuat di komputer sistem operasi Windows. Semua versi Windows dapat digunakan. Perlu diketahui bahwa PHP dapat digunakan di hampir semua sistem operasi (Windows, Linux, Mac OS, Solaris). Kebetulan saja pada buku ini semua contohnya dibuat pada komputer sistem operasi Windows.
- Browser. Browser dibutuhkan untuk menjalankan dan menampilkan contoh-contoh kode program yang terdapat di dalam buku ini. Berikut adalah beberapa program aplikasi browser yang cukup populer: Mozilla Firefox, Internet Explorer, Google Chrome, Opera, Safari. Untuk menghindari masalah kompatibilitas sebaiknya gunakan browser versi terbaru yang dapat diunduh secara gratis di masing-masing situs web resmi vendor yang membuat browser tersebut.
- XAMPP. Merupakan aplikasi yang mengintegrasikan beberapa aplikasi utama web di dalamnya. Dalam XAMPP terdapat instalasi modul PHP, MySQL, web server Apache. Selain XAMPP, saat ini terdapat banyak program aplikasi sejenis yang beredar di internet, seperti: phptriad, wamp, mamp, apache2triad. Selain itu dapat juga melakukan instalasi secara terpisah untuk modul PHP (php.net), MySQL (mysql.com), web server apache (apache.org). Buku ini menggunakan XAMPP. Petunjuk bagaimana instalasinya disajikan di bagian lampiran di buku ini. Versi

terbaru XAMPP dapat diunduh di <http://sourceforge.net/projects/xampp/>.

Web Editor. Merupakan aplikasi yang digunakan untuk mengetikkan kode-kode program PHP, HTML, Javascript. Saat ini banyak sekali program aplikasi web editor, di antaranya: Notepad, Notepad++, Adobe Dreamweaver. Buku ini menggunakan Adobe Dreamweaver dan Notepad sebagai web editornya.

Tips:

- Sebelum Anda mencoba script-script yang ada di halaman berikut, buatlah sebuah folder kerja Anda.
- Folder ini menjadi tempat penyimpanan file-file PHP Anda. Setiap mengetikkan script di tiap sesi, simpanlah sesuai dengan nama file yang ada di atas script tersebut.
- Contoh: File : "Lat01.php" berarti Anda harus menyimpan file tersebut dengan nama "Lat01.php".
- Jangan lupa untuk memberi ekstensi *.PHP di setiap nama file yang Anda simpan.

4.3 Mengenal Perintah Dasar PHP

PHP termasuk dalam *HTML-embedded*, artinya kode PHP dapat ditempatkan pada sebuah halaman HTML. Ada beberapa cara untuk menuliskan script PHP, yaitu:

1. ?php
Script PHP
?>
2. ?>
Script PHP
?>
3. <script language =
"php">
Script PHP Anda
</script>
4. <%
Script PHP Anda
%>

Cara pertama merupakan format yang dianjurkan tetapi mungkin cara ini akan sering digunakan karena lebih ringkas. Cara yang ketiga

digunakan untuk mengantisipasi editor-editor yang tidak dapat menentukan kedua cara di atas. Selain itu kita juga bisa menggunakan cara penulisan ASP, tetapi tentu saja ada beberapa konfigurasi yang perlu dilakukan.

Parser PHP bekerja membaca file HTML sampai ditemukan penanda khusus yang memberitahukan untuk menerjemahkan skrip berikut sebagai kode PHP. Parser PHP akan mengeksekusi semua perintah dalam blok kode PHP tersebut. Dengan cara inilah maka kode skrip PHP dapat ditempelkan pada dokumen HTML. Teks lainnya yang berada di luar tag PHP akan dianggap sebagai skrip HTML biasa.

Berikut ini contoh sederhana pemakaian bahasa PHP yang disisipkan dalam halaman HTML:

File: Lat01.php

```
<html>
  <head>
    <title>Skrip PHP pertama</title>
  </head>
  <body>
    <?php
      echo "Ini Skrip PHP pertamaku!";
    ?>
  </body>
</html>
```

Contoh lain:

File: Lat02.php

```
<html>
  <head>
    <title>Penyisipan PHP Pada HTML</title>
  </head>
  <body>
    Hello Selamat Datang di STMIK AMIKOM Yogyakarta
    <br>
    <?php
      // Berikut ini adalah kode PHP yang disisipkan
      echo "<b> Tempat Kuliah Orang Berdasarkan ";
    </?php>
```

```
<b> Unggul dalam Trend Teknologi Informasi.</b>";
  </body>
</html>
```

3.4 Komentar di PHP

Jika ada perintah yang tidak akan dieksekusi di PHP, maka perintah tersebut akan dijadikan sebagai komentar. Penggunaan komentar sangat dianjurkan untuk memberi penjelasan/keterangan pada kode-kode program PHP yang dituliskan dalam sebuah dokumen web. Ada dua cara penggunaan komentar di PHP:

1. Komentar Satu Baris, menggunakan tanda garis miring double/double slash [//]
2. Komentar Lebih dari satu baris, menggunakan tanda [/*] dan diakhiri dengan [*/].

File: Lat02_1.php

```
<html>
  <head>
    <title>Skrip PHP pertama</title>
  </head>
  <body>
    <?php
```

// ini adalah komentar satu baris

echo "Ini Skrip PHP pertamaku!";

/*
 ini adalah
 contoh komentar
 lebih dari satu baris
*/

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

BAB 5

Variabel dan Konstanta

5.1 Variabel PHP

Pada setiap bahasa pemrograman pasti akan kita temui konsep variabel. Variabel adalah sebuah tempat di memori untuk menyimpan data yang nilainya dapat berubah-ubah selama program dijalankan. Tidak seperti pada kebanyakan bahasa pemrograman lain yang mengharuskan kita untuk mendeklarasikan variabel terlebih dahulu, variabel dalam PHP tidak harus dideklarasikan sebelum variabel tersebut digunakan. Variabel diwakili oleh kata tertentu dengan aturan penulisan sebagai berikut:

1. Variabel dimulai dengan tanda dollar (\$).
2. Harus dimulai dengan huruf atau underscore (_).
3. Tidak boleh menggunakan tanda baca.
4. Case sensitive atau huruf besar dan huruf kecil berbeda.
5. Jangan menggunakan kata yang merupakan fungsi built-in PHP.

Contoh-contoh penulisan variabel:

Benar	Salah
\$variabel	\$var!abel
\$_nama	\$-nama
\$angkatan09	\$09angkatan
\$tahun_masuk	\$tahun-masuk

Contoh penggunaan variabel

Nama File	latihan/variabel.php
Keterangan	Contoh penggunaan variabel
<pre><html> <head> <title>Menggunakan variabel</title> </head> <body> <?php \$jurusan = "Teknik Informatika"; print("Jurusan : \$jurusan
"); \$jurusan = "Sistem Informasi"; print("Jurusan : \$jurusan
"); ?> </body> </html></pre>	

Nama File	latihan/variabel2.php
Keterangan	Contoh penggunaan variabel
<pre><html> <head> <title>Penyisipan PHP Pada HTML</title> </head> <body> Selamat Datang di Halaman Web Kami!
 <?php // Berikut ini adalah inisiasi beberapa variabel \$kampus = "STMIK AMIKOM Yogyakarta"; \$jurusan = "SI Teknik Informatika"; Salamat = "J. Ringroad Utara, Condong Catur"; ?> <?php echo "\$kampus, \$jurusan, Salamat."; ?> </body> </html></pre>	

Dari kedua contoh di atas variabel.php dan variabel2.php dapat dilihat ada 2 metode yang bisa digunakan untuk menampilkan output. variabel.php

menggunakan fungsi **print**, sedangkan pada variabel2.php menggunakan fungsi **echo**. Untuk mengetahui struktur dasar **echo** dan **print** silakan lihat bab 9 tentang fungsi built-in PHP.

Berikut ini adalah sebuah contoh mengenai betapa luwesnya penggunaan variabel dan tipe data dalam PHP. Simpan kode ini dalam nama variabel3.php.

Nama File	latihan/variabel3.php
Keterangan	Contoh penggunaan variabel dan tipe data
<pre><?php // Contoh variabel \$a \$a = "Testing"; // Kini \$a adalah variable jenis String echo "Nilai a adalah \$a (string)
"; \$a = 55; // Kini \$a adalah variable jenis Integer echo "Nilai a berubah menjadi \$a (Integer)
"; \$a = 7.5; // Kini \$a adalah variable jenis floating point echo "Nilai a sekarang menjadi \$a (floating point)
"; ?></pre>	

5.2 Konstanta

Konstanta adalah variabel yang nilainya tetap. Konstanta hanya diberi nilai pada awal program dan nilainya tidak pernah berubah selama program berjalan. PHP telah mendefinisikan beberapa konstanta, misalnya: **PHP_VERSION**, yaitu konstanta yang memberikan informasi tentang versi PHP yang digunakan. Selain konstanta yang telah disediakan oleh PHP juga dapat dibuat konstanta sendiri. Aturan penulisan konstanta adalah sebagai berikut:

```
define("nama_konstanta","nilai_konstanta")
```

Contoh penggunaan konstanta

Nama File	latihan/konstanta.php
Keterangan	Contoh penggunaan konstanta
	<pre><html> <head> <title>Menggunakan konstanta</title> <head> <body> <?php define("KAMPUS", "STMIK AMIKOM YOGYAKARTA"); print(KAMPUS); ?> </body> </html></pre>

BAB 6

Tipe Data dan Operator di PHP

6.1 Tipe Data di PHP

PHP mengenal beberapa macam tipe data antara lain Integer, Floating Point, dan String. Floating point lebih dikenal dengan nama double atau decimal. Penulisan string selalu diawali dengan tanda petik ganda ("") atau tanda petik tunggal ('').

Contoh-contoh penulisan tipe data yaitu sebagai berikut:

Tipe Data	Contoh	Keterangan
Integer	\$aks = 24; \$nilai = -5;	Bilangan bulat
Double	\$ipk = 3.75; \$pajak = 15.00;	Bilangan real
String	\$jurusan = "Teknik Informatika"; \$jenjang = "Strata Satu"	Karakter, kalimat

6.2 Operator di PHP

Operator adalah simbol yang digunakan untuk memanipulasi data, seperti penambahan, pengurangan, perkalian, perbandingan, atau penugasan. Ada operator yang menggunakan satu operand, ada juga yang menggunakan dua operand. Operand adalah data yang dioperasikan atau dimanipulasi. Operand dan operator bersama-sama membentuk suatu ekspresi (ungkapan).

Operator dapat dikelompokkan dalam 4 kategori, yaitu:

- Operator Aritmatika adalah operator yang berhubungan dengan fungsi matematika.
- Operator Logika adalah operator yang membandingkan TRUE dan FALSE.

3. Operator Bitwise adalah operator yang membandingkan binary.
4. Ada juga operator yang sering digunakan namun tidak termasuk dalam kelompok di atas, kita dapat mempelajarinya dalam kelompok lain.

6.2.1 Operator Aritmatika

Operator aritmatika merupakan operator yang berhubungan dengan fungsi matematika. Operator ini sering digunakan dalam program yang akan dibuat.

Operator	Operasi
+	Penambahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa pembagian
++	Penambahan dengan 1
--	Pengurangan dengan 1

Berikut ini akan diberikan contoh penggunaan operator aritmatika di atas.

Misalkan, nilai $y=6$:

Operator	Operasi	Contoh	Output
+	Penambahan	$x=y+4$	$x=10$
-	Pengurangan	$x=y-4$	$x=2$
*	Perkalian	$x=y*2$	$x=12$
/	Pembagian	$x=y/4$	$x=3$
%	Sisa pembagian	$x=y \% 2$	$x=0$
++	Penambahan dengan 1	$x=y++$	$x=7$
--	Pengurangan dengan 1	$x=y--$	$x=5$

Catatan : untuk nilai pecahan selalu diambil nilai bulatnya saja. Misal nilai yang didapat 13,85 maka apabila diambil nilai bulatnya akan menjadi 13.

Berikut ini tabel precedence of arithmetic operator dari PHP.

Prioritas	Operator	Keterangan
Pertama	()	Operator yang terdapat pada tanda kurung akan dihitung lebih dahulu.
Kedua	* / %	Operator berikut ini akan dijalankan setelah operator yang terletak di tanda kurung. Apabila terdapat lebih dari satu operator, maka urutan pengerjaannya dilakukan mulai dari kiri sampai kanan.
Ketiga	+ -	Operator ini merupakan operator terakhir yang dijalankan oleh PHP setelah operator yang terletak di tanda kurung dan operator * / %. Apabila terdapat lebih dari satu operator, maka urutan pengerjaannya dilakukan mulai dari kiri sampai kanan.

Berikut ini contoh apabila menginginkan nilai variabel \$x bernilai 15 maka perlu memberikan sepasang tanda kurung pada operator + sehingga script PHP menjadi sebagai berikut :

```
<?php
$x = (2 + 3) * 4 - 5;
echo "$x";
?>
```

listing program berikut ini akan memberikan contoh penggunaan operator aritmatika.

Nama File	latihan/OpAritmatika.php
Keterangan	Contoh penggunaan operator aritmatika dan tipe data
	<pre><html> <head> <title>Operator Aritmatika</title> </head> <body> <?php \$gaji = 2000000; \$spajak = 0.1; \$gajibersih = \$gaji - (\$gaji*\$spajak); echo "Gaji sebelum pajak = Rp. \$gaji
"; echo "Gaji yang dibawa pulang = Rp.</pre>

```
Sgajibersih";  
    ?>  
    </body>  
</html>
```

Listing berikut adalah bentuk pengembangan dari semua materi yang ada di contoh sebelumnya, simpan dengan nama: pesan.php

Nama File	latihan/pesan.php
Keterangan	Contoh penggunaan operator aritmatika dan tipe data
<pre> <?php // inisiasi variable yang digunakan // nama peralatan \$mainboard = "Mainboard"; \$processor = "Processor"; \$ram = "Ram"; \$hardisk = "Hardisk"; // harga per unit peralatan \$harga_mainboard = \$_POST['harga_mainboard']; \$harga_processor = \$_POST['harga_processor']; \$harga_ram = \$_POST['harga_ram']; \$harga_hardisk = \$_POST['harga_hardisk']; // jumlah peralatan yang ada \$jumlah_mainboard = \$_POST['jumlah_mainboard']; \$jumlah_processor = \$_POST['jumlah_processor']; \$jumlah_ram = \$_POST['jumlah_ram']; \$jumlah_hardisk = \$_POST['jumlah_hardisk']; // total harga per jenis peralatan \$total_mainboard = \$jumlah_mainboard * \$harga_mainboard; \$total_processor = \$jumlah_processor * \$harga_processor; \$total_ram = \$jumlah_ram * \$harga_ram; \$total_hardisk = \$jumlah_hardisk * \$harga_hardisk; // hitung peralatan total \$total_harga = \$total_mainboard + \$total_processor + \$total_ram + \$total_hardisk; // besar diskon untuk peralatan </pre>	

```

diskon = $_POST['diskon'];
// jumlah total diskon yang diberikan kepada
// user
nilai_diskon = ($diskon * $total_harga)/100;
// jumlah yang harus dibayar
total_harga_dibayar = $total_harga -
nilai_diskon;


```

title>Daftar Peralatan Yang Dibeli</title>

</head>

<body>

<center>

<form action=" pesan.php" method="POST">

Nama Hardware	Jumlah	Harga Satuan	Jumlah Harga
<?php echo \$mainboard; ?></td>	<input name="jumlah_mainboard" type="text" size="3" value="<?php echo \$jumlah_mainboard; ?>"></td>	<input name="harga_mainboard" type="text" size="15" value="<?php echo \$harga_mainboard; ?>"></td>	<?php echo \$total_mainboard; ?></td>
<?php echo \$processor; ?></td>	<input name="jumlah_processor" type="text" size="3" value="<?php echo \$jumlah_processor; ?>"></td>		

</form>

```

<td align="right"><input name="harga_processor"
type="text" size="15" value="<?php echo
$harga_processor; ?>"></td>
<td align="right"><?php echo $total_processor;
?></td>
</tr>
<tr>
<td align="left"><?php echo $ram; ?></td>
<td align="right"><input name="jumlah_ram"
type="text" size="3" value="<?php echo
$jumlah_ram; ?>"></td>
<td align="right"><input name="harga_ram"
type="text" size="15" value="<?php echo
$harga_ram; ?>"></td>
<td align="right"><?php echo $total_ram; ?></td>
</tr>
<tr>
<td align="left"><?php echo $hardisk; ?></td>
<td align="right"><input name="jumlah_hardisk"
type="text" size="3" value="<?php echo
$jumlah_hardisk; ?>"></td>
<td align="right"><input name="harga_hardisk"
type="text" size="15" value="<?php echo
$harga_hardisk; ?>"></td>
<td align="right"><?php echo $total_hardisk;
?></td>
</tr>
<tr>
<td colspan="3" align="right">Total Harga</td>
<td align="right"><?php echo $total_harga; ?></td>
</tr>
<tr>
<td colspan="3" align="right">
Diskon <input name="diskon" type="text" size="2"
value="<?php echo $diskon; ?>">%</td>
<td align="right"><?php echo $nilai_diskon;
?></td>
</tr>
<tr>
<td colspan="3" align="right">Jumlah harus
dibayar</td>
<td align="right"><?php echo $total_harga_dibayar;
?></td>

```

```

</tr>
<td colspan="4" align="center">
<input type="submit" value="OK" />
<input type="reset" value="Batal" />
</td>
</tr>
</table>
</form>
</center>
</body>
</html>

```

Tampilan di Dreamweaver

Daftar Pemesanan Hardware Komputer			
Nama Hardware	Jumlah	Harga Satuan	Jumlah Harga
MB	<input type="text" value="1"/>	<input type="text" value="500000"/>	<input type="text" value="500000"/>
Processor	<input type="text" value="1"/>	<input type="text" value="450000"/>	<input type="text" value="450000"/>
Ram	<input type="text" value="1"/>	<input type="text" value="250000"/>	<input type="text" value="250000"/>
Hardisk	<input type="text" value="1"/>	<input type="text" value="350000"/>	<input type="text" value="350000"/>
Total Harga <input type="text" value="1200000"/>			
Diskon <input type="text" value="5"/> % <input type="text" value="60000"/>			
Jumlah harus dibayar <input type="text" value="1140000"/>			
<input type="button" value="OK"/> <input type="button" value="Batal"/>			

Tampilan di Browser

Daftar Pemesanan Hardware Komputer			
Nama Hardware	Jumlah	Harga Satuan	Jumlah Harga
Mainboard	<input type="text" value="1"/>	<input type="text" value="500000"/>	<input type="text" value="500000"/>
Processor	<input type="text" value="1"/>	<input type="text" value="450000"/>	<input type="text" value="450000"/>
Ram	<input type="text" value="1"/>	<input type="text" value="250000"/>	<input type="text" value="250000"/>
Hardisk	<input type="text" value="1"/>	<input type="text" value="350000"/>	<input type="text" value="350000"/>
Total Harga <input type="text" value="1200000"/>			
Diskon <input type="text" value="5"/> % <input type="text" value="60000"/>			
Jumlah harus dibayar <input type="text" value="1140000"/>			
<input type="button" value="OK"/> <input type="button" value="Batal"/>			

Daftar Pemesanan Hardware Komputer			
Nama Hardware	Jumlah	Harga Satuan	Jumlah Harga
Mainboard	1	500000	500000
Processor	1	450000	450000
Ram	1	280000	280000
Hardisk	1	350000	350000
Total Harga		1550000	
Diskon 5 %		77500	
Jumlah harus dibayar		1472500	
<input type="button" value="OK"/> <input type="button" value="Batal"/>			

6.2.2 Penggabungan String

Nama File	latihan/GabungString.php
Keterangan	Contoh penggabungan string
	<pre><?php // inisiasi variabel \$ampus = "STMIK AMIKOM Yogyakarta"; \$jurusan = "Teknik Informatika"; \$strata = "Strata Satu"; // alternatif pertama \$alt1 = \$ampus . " " . \$strata . ", " . \$jurusan . "."; // alternatif kedua \$alt2 = \$jurusan . ", " . \$ampus . " " . \$strata . "."; ?> <html> <head> <title>Menggabungkan String</title> </head> <body></pre>

String yang pertama adalah:

<?php echo \$alt1; ?>

</body>

String yang kedua adalah:

<?php echo \$alt2; ?>

</body>

</html>

Tampilan di browser:



6.3.3 Operator Logika

Operator ini akan membandingkan TRUE atau FALSE. Seperti bahasa C, PHP mendefinisikan *False* dengan 0 dan *True* dengan 1.

Operator	Operasi
==	Sama dengan
!=	Tidak sama dengan
<	Lebih kecil
>	Lebih besar
<=	Lebih kecil sama dengan
>=	Lebih besar sama dengan
AND atau &&	Logika AND
OR atau	Logika OR
XOR	Logika XOR
!	Logika NOT

Tabel kebenaran operasi logika:

p	q	p AND q	p OR q	p XOR q	!p
True	True	True	True	False	False
True	False	False	True	True	False
False	True	False	True	True	True
False	False	False	False	False	True

6.2.4 Operator Bitwise

Operator ini digunakan untuk memanipulasi bit-bit dari nilai data. Operator ini beserta artinya dapat dilihat pada tabel berikut:

Operator	Operasi
>>	Pergeseran bit ke kanan
<<	Pergeseran bit ke kiri
~	Komplemen satu atau NOT
&	Bitwise AND
	Bitwise OR
^	Bitwise XOR (Exclusive OR)

Hubungan bit dan hasilnya untuk operator bitwise logika:

BIT		AND	OR	XOR
b1	b2			
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

Hasil 12 dan 10:

$$\begin{array}{r}
 1 & 1 & 0 & 0 & (12) \\
 8 & 1 & 0 & 1 & 0 & (10) \\
 \hline
 1 & 0 & 0 & 0 & (08)
 \end{array}$$

6.3.5 Operator Lain

Besides operator di atas, PHP masih mempunyai operator lain. Berikut ini beberapa operator yang sering digunakan dalam script.

Operator	Arti
.	Penggabungan string
=	Penugasan/pengisian nilai
\$	Mengacu pada variabel

BAB 7

Struktur Kendali

Struktur kendali merupakan pengatur aliran program, mempunyai rangkaian perintah yang harus ditulis untuk memenuhi beberapa keadaan, yaitu:

- Mengulang suatu perintah jika suatu kondisi terpenuhi.
- Melanjutkan sebuah pernyataan bila kondisi terpenuhi.
- Memilih sebuah pilihan dari beberapa alternatif bila kondisi terpenuhi.

Struktur kendali dapat dibagi menjadi dua jenis, yaitu struktur kendali percabangan (pengambilan keputusan) dan struktur kendali pengulangan (looping).

7.1 Struktur Kendali Percabangan

Struktur kendali percabangan (pengambilan keputusan) adalah struktur kendali yang berfungsi untuk melakukan pemilihan atas perintah yang akan dijalankan sesuai dengan kondisi tertentu. Ada empat perintah percabangan dalam PHP, yaitu if, if..else, if..elseif, dan switch.

• Perintah IF

Perintah IF digunakan untuk menjalankan satu atau lebih perintah berdasarkan suatu kondisi. Sintaks penulisan perintah IF adalah sebagai berikut:

if (kondisi)

pernyataan yang akan dijalankan apabila kondisi
benar

Dalam bentuk ini, bagian *pernyataan* akan dijalankan hanya jika bagian *kondisi* bernilai benar. Berikut ini adalah contoh penggunaan struktur kendali IF :

File: Lat12.php

```
<html>
  <head>
    <title>Struktur Kendali IF</title>
  </head>
  <body>
    <?php
      $x = 12;
      if ($x > 10)
      {
        print("Selamat Siang");
      }
    </?>
    </body>
</html>
```

Perintah IF - ELSE

Perintah IF - ELSE digunakan untuk memilih salah satu pernyataan berdasarkan suatu kondisi. Perintah ini akan menjalankan pernyataan tertentu bila kondisi bernilai benar dan akan menjalankan pernyataan yang lain jika kondisi bernilai salah. Sintaks penulisannya adalah sebagai berikut:

```
IF (kondisi)
{
  pernyataan_1
}
ELSE
{
  pernyataan_2
}
```

Pada bentuk ini *pernyataan_1* dijalankan jika *kondisi* bernilai benar, dan *pernyataan_2* dijalankan apabila *kondisi* bernilai salah. Contoh penggunaan perintah IF - ELSE, sebagai berikut:

File: Lat14.php

```
<html>
  <head>
    <title>Struktur Kendali IF...ELSE</title>
  </head>
  <body>
    <?php
      $x = 9;
      if ($x > 10)
      {
        print("Selamat Siang");
      }
      else
      {
        print("Selamat Pagi");
      }
    </?>
    </body>
</html>
```

Perintah IF - ELSEIF

Perintah IF - ELSEIF digunakan untuk menjalankan suatu pernyataan dengan melibatkan lebih dari satu kondisi. Sintaks penulisannya sebagai berikut:

```
IF (kondisi_1)
{
  pernyataan_1
}
ELSEIF (kondisi_2)
{
  pernyataan_2
}
ELSE
{
  pernyataan_3
}
```

Pada bentuk ini *pernyataan_1* dijalankan kalau *kondisi_1* bernilai benar, dan *pernyataan_2* dijalankan apabila *kondisi_2* bernilai benar.

Sedangkan *Pernyataan_3* akan dijalankan apabila *kondisi_1* dan *kondisi_2* bernilai salah. Contoh program yang menggunakan perintah IF ELSEIF adalah sebagai berikut:

File: Lat16.php

```
<html>
  <head>
    <title>Struktur Kendali IF...ELSEIF</title>
  </head>
  <body>
    <?php
      $waktu = getdate();
      if ($waktu[hours] <= 10)
      {
        print("Selamat Pagi");
      }
      elseif ($waktu[hours] <= 15)
      {
        print("Selamat Siang");
      }
      elseif ($waktu[hours] <= 18)
      {
        print("Selamat Sore");
      }
      else
      {
        print("Selamat Malam");
      }
    ?>
  </body>
</html>
```

KASUS: Buatlah script untuk menentukan kelulusan seorang siswa terhadap Mata Kuliah PI dengan kisaran nilai : > 70 nilainya "A", >= 50 nilainya "B", >=30 nilainya "C", >= 10 nilainya "D", selain itu pesannya Tidak Lulus. Simpanlah dengan nama Lat17.php

8 Perintah switch

Perintah *switch* digunakan sebagai alternatif pengganti dari perintah *if...elseif*. Dengan perintah ini program percabangan akan lebih mudah dibuat dan dipelajari. Sintaks penulisan perintah *switch* adalah sebagai berikut:

```
switch (kondisi)
{
  case konstanta_1:
    pernyataan_1;
    break;
  case konstanta_2:
    pernyataan_2;
    break;
  default:
    pernyataan_3;
}
```

Perintah *switch* akan menyeleksi kondisi yang diberikan dan membandingkan hasilnya dengan konstanta-konstanta yang berada pada *case*. Pembandingan akan dimulai dari *konstanta_1* sampai konstanta terakhir. Jika hasil dari kondisi sama dengan nilai konstanta tertentu, maka pernyataan pada konstanta tersebut akan dijalankan sampai ditemukan pernyataan *break*. Jika hasil dari kondisi tidak ada yang sama dengan konstanta-konstanta yang diberikan, maka pernyataan pada *default* yang akan dijalankan. Berikut ini contoh penggunaan struktur kendali dengan perintah *switch*:

File: Lat18.php

```
<html>
  <head>
    <title>Struktur Kendali dengan Switch</title>
  </head>
  <body>
    <?php
      $english = date("l");
      switch($english_day)
      {
        case "Monday":
          $indonesian = "Senin";
          break;
```

```

case "Tuesday":
    $indonesian = "Selasa";
    break;
case "Wednesday":
    $indonesian = "Rabu";
    break;
case "Thursday":
    $indonesian = "Kamis";
    break;
case "Friday":
    $indonesian = "Jumat";
    break;
case "Saturday":
    $indonesian = "Sabtu";
    break;
default:
    $indonesian = "Minggu";
}
print("<h2>Hari ini adalah hari  
$indonesian</h2>")
?>
</body>
</html>

```

Catatan: nilai pada fungsi date("l") adalah huruf L, bukan angka 1. Untuk lebih jelas lihat Bab 9 bagian Fungsi Tanggal dan Jam.

Latihan:

Buatlah script untuk menentukan kelulusan seperti pada contoh "Lat17.php" dengan menggunakan **switch**. Lalu simpan dengan nama file Lat19.php.

7.2 Struktur Kendali Pengulangan

Struktur kendali pengulangan digunakan untuk mengulangi suatu perintah sebanyak yang diinginkan. Ada tiga jenis perintah pengulangan dalam PHP yaitu FOR, WHILE, dan DO – WHILE.

Perintah FOR

Perintah FOR digunakan untuk mengulangi suatu perintah dengan jumlah pengulangan yang sudah diketahui. Pada perintah ini tidak perlu menuliskan kondisi untuk diuji. Kita hanya perlu menuliskan nilai awal dan akhir variabel penghitung. Nilai variabel penghitung akan secara otomatis ditambah atau berkurang tiap kali sebuah pengulangan dilaksanakan. Contoh penulisan perintah FOR adalah sebagai berikut:

```

for (nilai_awal; nilai_akhir; penambahan/pengurangan)
{
    pernyataan yang dijalankan
}

```

Struktur pengulangan dengan FOR :

Lat18.php

```

<!DOCTYPE html>
<html>
    <head>
        <title>Struktur pengulangan dengan FOR</title>
    </head>
    <body>
        </php>
        for($i=1; $i<=7; $i++)
        {
            print("<h$i>Ini adalah contoh heading
</h$i>");
        }
    </body>
</html>

```

Lat19.php

```

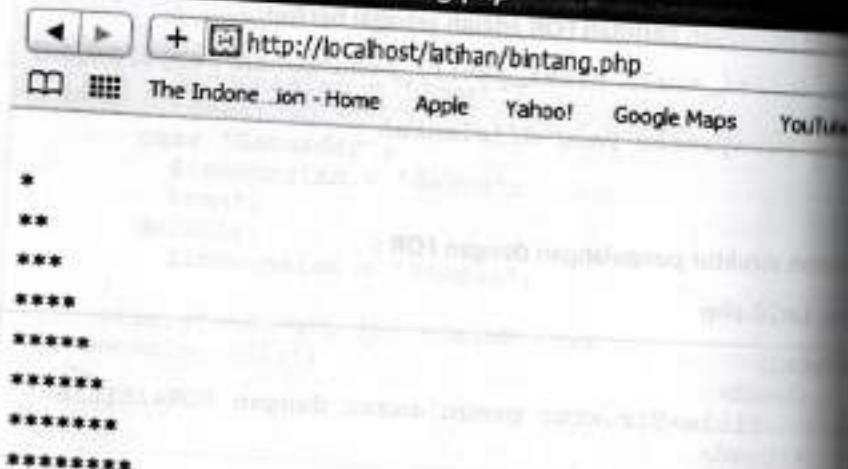
for ($i=1; $i<10; $i++)
{
    for ($x=1; $x<$i; $x++)
    {
        echo '**';
    }
}

```

```
echo'<br />';
}
?>
```

Tampilan di browser:

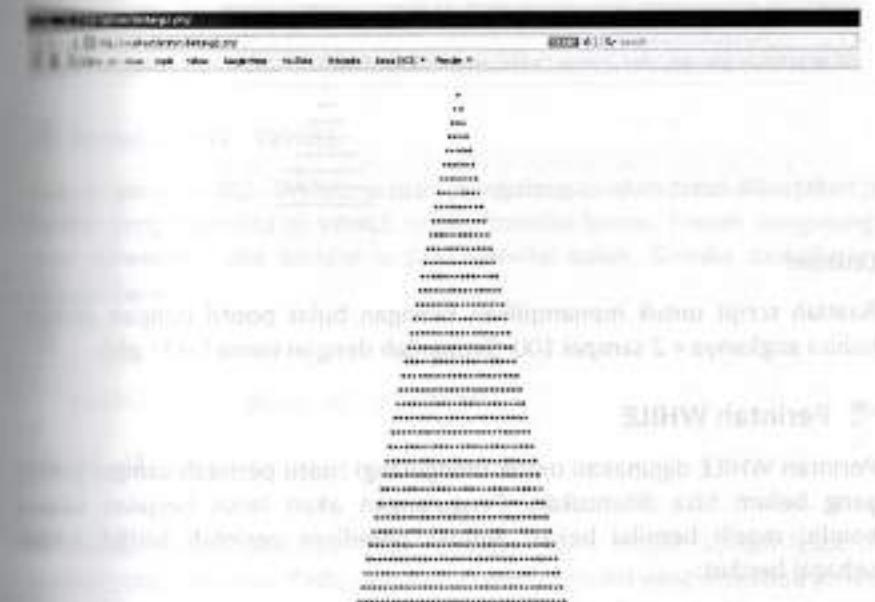
④ <http://localhost/latihan/bintang.php>



File: bintang2.php

```
<center>
<?
for($i=1;$i<100;$i++)
{
    for($x=1;$x<$i;$x++)
    {
        echo '**';
    }
    echo '<br/>';
}
?>
</center>
```

Tampilan di browser:



file: bintang3.php

```
<center>
<?
for($i=1;$i<15;$i=$i+2)
{
    for($j=0;$j<$i;$j++)
    {
        echo '**';
    }
    echo '<br />';
}
?>
</center>
```

Tampilan di browser:



Latihan:

Buatlah script untuk menampilkan bilangan bulat positif dengan pertambahan angkanya = 2 sampai 100. Simpanlah dengan nama Lat21.php

④ Perintah WHILE

Perintah WHILE digunakan untuk mengulangi suatu perintah sampai jumlah yang belum bisa ditentukan. Pengulangan akan terus berjalan selama kondisi masih bernilai benar. Sintaks penulisan perintah WHILE adalah sebagai berikut:

```
WHILE (kondisi)
{
    pernyataan yang akan dijalankan
}
```

Contoh struktur pengulangan dengan WHILE:

File: Lat22.php

```
html>
<head>
    <title>Struktur pengulangan dengan
WHILE</title>
</head>
<body>
<?php
<
    $i=1;
    while ($i <= 9)
    {
        print("Ini pengulangan yang ke-$i<br>");
        $i++;
    }
}</body>
</html>
```

```
}
?>
</body>
</html>
```

⑤ Perintah DO - WHILE

Dengan perintah DO - WHILE, proses pengulangan akan terus dikerjakan jika kondisi yang diperiksa di WHILE masih bernilai benar. Proses pengulangan akan dihentikan jika kondisi sudah bernilai salah. Sintaks penulisannya sebagai berikut:

```
DO
{
    pernyataan yang dijalankan
}
WHILE (kondisi);
```

Perbedaan antara perintah WHILE dengan DO - WHILE adalah letak dari kondisi yang diperiksa. Pada perintah WHILE, kondisi yang diperiksa terletak di awal perulangan, sehingga sebelum masuk ke dalam perulangan WHILE kondisi harus bernilai benar. Sedangkan pada perintah DO - WHILE, kondisi diperiksa di akhir perulangan. Ini berarti paling sedikit sebuah perulangan akan dilakukan oleh perintah DO - WHILE, karena untuk masuk ke perulangan tidak ada kondisi yang harus dipenuhi. Contoh struktur pengulangan dengan DO - WHILE:

File: Lat23.php

```
<html>
<head>
    <title>Struktur pengulangan dengan
do...while</title>
</head>
<body>
<?php
    $i = 1;
    do
    {
        print("Ini juga pengulangan yang ke-
$i<br>");
        $i++;
    }
}</body>
</html>
```

```

}
while ($i <= 6):

    $j = 5;
    do
    {
        print("<br>Perulangan ini kondisinya tidak
terpenuhi<br>");
        $j++;
    }
    while($j <= 3);
?>
</body>
</html>

```

Contoh-contoh script perulangan

File: perulangan ganti baris+tabel.php

```





```

Tampilan di browser:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50

file: perulangan ganti baris.php

```





```

Tampilan di browser:

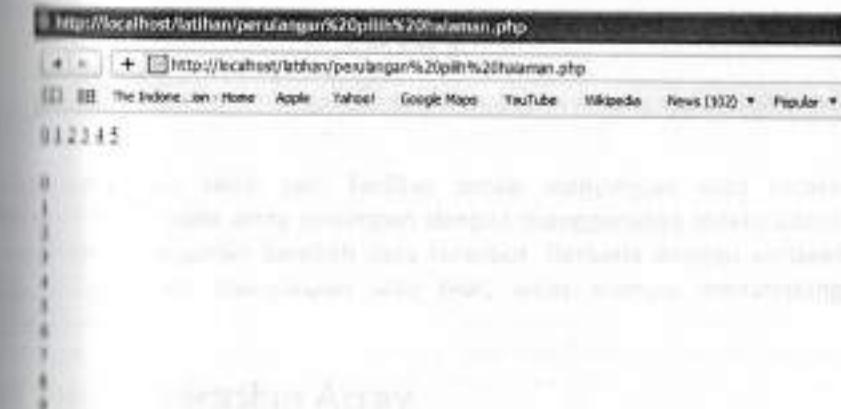
1.	2.	3.	4.	5.
6.	7.	8.	9.	10.
11.	12.	13.	14.	15.
16.	17.	18.	19.	20.
21.	22.	23.	24.	25.
26.	27.	28.	29.	30.
31.	32.	33.	34.	35.
36.	37.	38.	39.	40.
41.	42.	43.	44.	45.
46.	47.	48.	49.	50.

File: perulangan pilih halaman.php

```

<a href="perulangan_pilih_halaman.php?depan=0">0</a>
<a href="perulangan_pilih_halaman.php?depan=1">1</a>
<a href="perulangan_pilih_halaman.php?depan=2">2</a>
<a href="perulangan_pilih_halaman.php?depan=3">3</a>
<a href="perulangan_pilih_halaman.php?depan=4">4</a>
<a href="perulangan_pilih
halaman.php?depan=5">5</a><br />
<br />
<?
$depan=$_GET['depan'];
for($i=0;$i<=9;$i++){
    echo (( $depan*10)+$i). "<br />";
}
?>
```

Tampilan di browser:



Perulangan Array

Perulangan array adalah perulangan yang dilakukan pada setiap elemen dalam sebuah array. Perulangan array ini biasanya digunakan untuk menampilkan data yang tersimpan dalam array.

Perulangan array ini biasanya dilakukan dengan menggunakan loop for. Dalam loop for, kita dapat mengakses setiap elemen dalam array dengan menggunakan indeksnya.

Contoh penggunaan perulangan array dalam PHP:

```

$angka = array(1, 2, 3, 4, 5);
for ($i = 0; $i < 5; $i++) {
    echo $angka[$i];
}
```

Dalam contoh di atas, kita memiliki array yang berisi angka 1 hingga 5. Kita menggunakan loop for dengan indeks \$i yang dimulai dari 0 hingga 4. Setiap kali loop berjalan, kita mencetak nilai pada indeks \$i dari array \$angka.

Hasilnya akan menjadi:

BAB 8

Array

Array merupakan salah satu fasilitas untuk menyimpan data secara dinamis. Data pada array tersimpan dengan menggunakan indeks untuk memudahkan pencarian kembali data tersebut. Berbeda dengan variabel yang hanya dapat menyimpan satu nilai, array mampu menampung banyak nilai.

8.1 Pendeklarasian Array

Objek yang terdapat dalam array disebut elemen-elemen array dan letak posisi masing-masing elemen array tersebut ditunjukkan oleh suatu indeks. Setiap elemen array memiliki batas atas dan batas bawah, dimana data akan tersimpan di antara kedua batas tersebut. Semua elemen array yang tersimpan memiliki tipe data yang sama. Array dapat berdimensi satu, dua, tiga atau lebih. Array berdimensi satu (*one-dimensional array*) mewakili bentuk vektor. Array berdimensi dua (*two-dimensional array*) mewakili bentuk suatu matrix. Array berdimensi tiga (*(three-dimensional array)* mewakili bentuk suatu ruang.

Berapa cara mendeklarasikan array:

file lat24.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Menggunakan Array</title>
</head>
<body>
    <?php
        #Cara pendeklarasian array pertama
        $warna
        =array("merah","kuning","hijau","biru");

        print("<p>Pendeklarasian array pertama:<br>");
        print($warna[2]);
    </?php>
</body>
</html>
```

```

#Cara pendeklarasian array kedua
$kota[] = "Jakarta";
$kota[] = "Bandung";
$kota[] = "Surabaya";
$kota[] = "Solo";
$kota[] = "Semarang";
$kota[] = "Jogjakarta";

print("<p>Pendeklarasian array kedua:<br>");
print($kota[5]);

#Cara pendeklarasian array ketiga
$kota[0] = "Jakarta";
$kota[5] = "Bandung";
$kota[2] = "Surabaya";
$kota[] = "Solo";
$kota[] = "Semarang";
$kota[gudeg] = "Jogjakarta";

print("<p>Pendeklarasian array ketiga:<br>");
print("$kota[5] <br>");
print("$kota[7] <br>");
print("Saat ini saya sedang kuliah di
$kota[gudeg]");
?>
</body>
</html>

```

8.2 Array Multidimensi

File: Lat25.php

```

<html>
<head>
    <title>Array Multidimensi</title>
</head>
<body>
<?php
    $kota = array(
        "jatim"=>array("Surabaya", "Malang", "Jember",
        "Bondowoso"),

```

```

        "jabar"=>array("Bandung", "Bogor", "Tangerang"),
        "jogja"=>array("Sleman", "Bantul", "Kulonprogo",
        "yogyakarta"));
        print("Saya berasal dari ".$kota[jabar][2]. "
        dan sekarang saya kuliah di
        ".$kota[jogja][1]);
?>
</body>
</html>

```

8.2.1 Array php

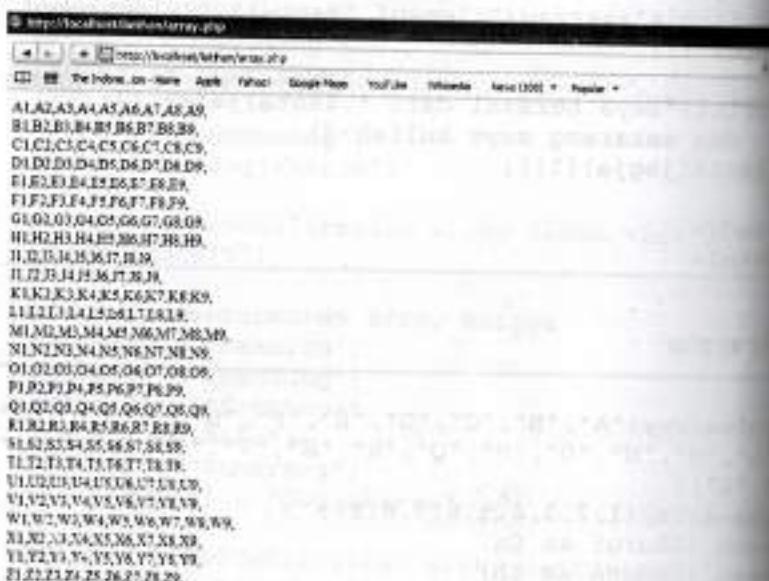
```

$abjad=array("A","B","C","D","E","F","G","H","I","J",
            "K","L","M","N","O","P","Q","R","S","T","U","V","W","X",
            "Y","Z");
$angka=array(1,2,3,4,5,6,7,8,9);
echo ($urut as $a)
echo ($angka as $b)

$abjad,*;
if ($b == 9)
{
    echo "<br>";
}

```

Tampilan di browser



The screenshot shows a web browser window with the URL <http://localhost/tampildiarray.php>. The page displays a list of names in uppercase letters from A to Z, each followed by a short description. The browser's address bar and navigation buttons are visible at the top.

```

A,A1,A2,A3,A4,A5,A6,A7,A8,A9,
B,B1,B2,B3,B4,B5,B6,B7,B8,B9,
C,C1,C2,C3,C4,C5,C6,C7,C8,C9,
D,D1,D2,D3,D4,D5,D6,D7,D8,D9,
E,E1,E2,E3,E4,E5,E6,E7,E8,E9,
F,F1,F2,F3,F4,F5,F6,F7,F8,F9,
G,G1,G2,G3,G4,G5,G6,G7,G8,G9,
H,H1,H2,H3,H4,H5,H6,H7,H8,H9,
I,I1,I2,I3,I4,I5,I6,I7,I8,I9,
J,J1,J2,J3,J4,J5,J6,J7,J8,J9,
K,K1,K2,K3,K4,K5,K6,K7,K8,K9,
L,L1,L2,L3,L4,L5,L6,L7,L8,L9,
M,M1,M2,M3,M4,M5,M6,M7,M8,M9,
N,N1,N2,N3,N4,N5,N6,N7,N8,N9,
O,O1,O2,O3,O4,O5,O6,O7,O8,O9,
P,P1,P2,P3,P4,P5,P6,P7,P8,P9,
Q,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,
R,R1,R2,R3,R4,R5,R6,R7,R8,R9,
S,S1,S2,S3,S4,S5,S6,S7,S8,S9,
T,T1,T2,T3,T4,T5,T6,T7,T8,T9,
U,U1,U2,U3,U4,U5,U6,U7,U8,U9,
V,V1,V2,V3,V4,V5,V6,V7,V8,V9,
W,W1,W2,W3,W4,W5,W6,W7,W8,W9,
X,X1,X2,X3,X4,X5,X6,X7,X8,X9,
Y,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,
Z,Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9.

```

Berikut contoh penerapan Array dan Operator Aritmatika.

Nama file: ArrayFunction.php

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Operator Aritmatika dan Array</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>

<body>

<?
//Fungsi tampilan tanggal
$english_day = getdate();
if ($english_day[hours] <= 10){
    $ucapan = "Selamat pagi";
}

```

```

elseif ($english_day[hours] <= 18){
    $ucapan = "Selamat Siang";
}
else{
    $ucapan = "Selamat malam";
}

$message = date("d M Y") . ", " . date("h:i") . ", " .
$ucapan;

//tampilkan Jumlah Barang
$jum_barang = array(12 , 10 , 12, 1, 2);

//tampilkan harga barang
$harga_barang = array(500000, 25000, 25000, 7500,
1000);

//total harga per unit setelah diDiskon
for ($i = 0; $i < 5; $i++){
    if($jum_barang[$i] >= 12){
        $bayar[$i] = $jum_barang[$i] *
$harga_barang[$i];
        $diskon = ($bayar[$i] * 15)/100 ;
        $total[$i] = $bayar[$i] - $diskon;
    }
    elseif ($jum_barang[$i] >= 10){
        $bayar[$i] = $jum_barang[$i] *
$harga_barang[$i];
        $diskon = ($bayar[$i] * 10)/100 ;
        $total[$i] = $bayar[$i] - $diskon;
    }
    else {
        $total[$i] = $jum_barang[$i] *
$harga_barang[$i];
    }
}

//total harga yang harus dibayar
$bayar = 0;
for ($d=0 ; $d<5 ; $d++){
    $bayar = $bayar + $total[$d] ;
}

//harga yang kena pajak

```

```

if ($bayar < 5000000){
    $tax = 10;
    $bayar_pajak = ($bayar * 10)/100;
    $bayar_baru = $bayar - $bayar_pajak;
}
else{
    $tax = 0;
    $bayar_baru = $bayar;
}
?>
<table width="618" border="1">
    <tr>
        <td colspan="5"><div align="center"><strong>MINI
MARKET AMIKOM YOGYAKARTA </strong></div></td>
    <tr>
        <td colspan="5"><font color="red">
        <?
            echo $message;
        ?></font></td>
    <tr>
        <td width="43"><strong>NO</strong></td>
        <td width="132"><strong>BARANG</strong></td>
        <td width="111"><strong>JUMLAH</strong></td>
        <td width="117"><strong>HARGA</strong></td>
        <td width="181"><strong>SUB TOTAL </strong></td>
    </tr>
    <tr>
        <td>1</td>
        <td>Printer</td>
        <td align="right"><? echo $jumlah_barang[0]; ?></td>
        <td align="right"><? echo $harga_barang[0];
?></td>
        <td align="right"><? echo $total[0]; ?></td>
    </tr>
    <tr>
        <td>2</td>
        <td>Mouse</td>
        <td align="right"><? echo $jumlah_barang[1]; ?></td>
        <td align="right"><? echo $harga_barang[1];
?></td>
        <td align="right"><? echo $total[1]; ?></td>
    
```

```

</tr>
<tr>
    <td>3</td>
    <td>Keyboard</td>
    <td align="right"><? echo $jumlah_barang[2]; ?></td>
    <td align="right"><? echo $harga_barang[2];
?></td>
    <td align="right"><? echo $total[2]; ?></td>
</tr>
<tr>
    <td>4</td>
    <td>Gula pasir </td>
    <td align="right"><? echo $jumlah_barang[3]; ?></td>
    <td align="right"><? echo $harga_barang[3];
?></td>
    <td align="right"><? echo $total[3]; ?></td>
</tr>
<tr>
    <td>5</td>
    <td>Sabun Cuci </td>
    <td align="right"><? echo $jumlah_barang[4]; ?></td>
    <td align="right"><? echo $harga_barang[4];
?></td>
    <td align="right"><? echo $total[4]; ?></td>
</tr>
<tr>
    <td colspan="4">Total</td>
    <td align="right"><? echo $bayar; ?></td>
</tr>
<tr>
    <td colspan="4">Pajak</td>
    <td align="right"><? echo $tax; ?></td>
</tr>
<tr>
    <td colspan="4">Total yang dibayar</td>
    <td align="right"><? echo $bayar_baru; ?></td>
</tr>
<tr>
    <td colspan="5"><center>Kami menyediakan segala
macam kebutuhan Anda</center></td>
</tr>
</table>

```

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

Tampilan di browser:

MINI MARKET AMIKOM YOGYAKARTA				
14 Jun 2011, 03:43, Selamat Song				
NO	BARANG	JUMLAH	HARGA	SUB TOTAL
1	Printer	12	500000	\$100000
2	Mouse	10	25000	225000
3	Keyboard	12	25000	255000
4	Gula pasir	1	7500	7500
5	Sabun Cuci	2	15000	30000
Total				\$617500
Pajak				0
Total yang dibayar				\$617500
Kami menyediakan segala macam kebutuhan Anda				

BAB 9

Fungsi

Dalam pembuatan program seringkali kita membutuhkan sekumpulan perintah yang akan digunakan berulang kali. Tentunya sangat merepotkan apabila perintah-perintah tersebut harus diketik ulang setiap kali akan digunakan. Hal ini dapat dihindari dengan menggunakan *subrutin*. Subrutin adalah sekumpulan perintah yang diberi nama dan kemudian dapat kita panggil sewaktu-waktu. Dalam pemrograman dikenal dua macam subrutin, yaitu prosedur dan fungsi.

Perbedaan antara kedua macam subrutin ini adalah: fungsi mengembalikan nilai tertentu, sedangkan prosedur tidak. Bahasa PHP hanya mengenal perintah untuk membuat fungsi, yaitu *function*. Pada sebuah subrutin diperlukan argumen, yaitu nilai-nilai yang harus dimasukkan pada saat subrutin tersebut dipanggil. Tetapi jangan khawatir, kita dapat membuat prosedur dengan menggunakan perintah *function* tetapi tidak dengan menggunakan perintah *return*. Perintah *return* digunakan untuk mengembalikan nilai tertentu. Standar penulisan fungsi adalah:

```
function nama_fungsi(argumen)
{
    kode perintah
}
```

Berikut ini contoh penggunaan fungsi baik yang menggunakan perintah *return* maupun yang tidak menggunakan perintah *return*, perhatikan hasilnya.

Contoh program tanpa perintah *return*:

File: Lat26.php

```
<html>
<head>
```

```

<title>Membuat fungsi tanpa perintah
return</title>
</head>
<body>
<?php
    function amikom()
    {
        print("www.amikom.ac.id <br /> Visinya adalah
Menjadi
        Perguruan Tinggi Komputer Terbaik di Asia");
    }

    amikoms();
?>
</body>
</html>

```

Contoh program dengan perintah `return`:

File: Lat27.php

```

<html>
<head>
    <title>Membuat fungsi dengan perintah
return</title>
</head>
<body>
<?php
    function cetak_tebal($input)
    {
        $teks = "<b>";
        $teks .= "$input";
        $teks .= "</b>";
        return($teks);
    }
    print("Kalimat ini belum dicetak tebal
<br>\n");
    print(cetak_tebal("Kalau yang ini sudah dicetak
tebal"));
?>
</body>
</html>

```

9.1 Fungsi dengan Argumen

Argumen adalah suatu nilai yang dimasukkan ke dalam sebuah fungsi. Secara default sebuah argumen bersifat *pass by value*, yang berarti hanya nilainya saja yang dibutuhkan sehingga nilai pada variabel tersebut tidak mengalami perubahan setelah fungsi dijalankan. Perhatikan contoh berikut:

File: Lat28.php

```

<html>
<head>
    <title>Fungsi dengan argumen pass by
value</title>
</head>
<body>
<?php
    function luas($panjang, $lebar)
    {
        $luas = $panjang*$lebar;
        return($luas);
    }

    print("Luasnya = ".luas(3,9));
?>
</body>
</html>

```

Selain argumen yang telah dijelaskan di atas, ada juga argumen yang bersifat *pass by reference*, yang digunakan ketika kita ingin mengubah nilai sebuah argumen. Argumen yang bersifat *pass by reference* ini nilainya akan berubah setelah fungsi dijalankan. Untuk membuat argumen *pass by reference* kita akan menggunakan karakter "&" di depan nama argumennya. Untuk lebih jelas perhatikan contoh berikut:

File: Lat29.php

```

<html>
<head>
    <title>Fungsi dengan argumen pass by
reference</title>
</head>

```

```

<body>
<?php
function sambung(&$string)
{
    $string .= "dan tambahannya";
}
$str = "Ini sebuah contoh string ";
sambung($str);
echo $str;
?>
</body>
</html>

```

9.2 Fungsi Built-In

Bagian ini akan membahas fungsi-fungsi yang telah disediakan oleh PHP. Kita juga dapat mempelajari fungsi-fungsi lainnya dalam berbagai manual yang banyak terdapat di internet.

□ Fungsi-fungsi Umum

echo *string_first, string_second, ..., string_last*

Fungsi *echo* sering digunakan untuk mengirim satu atau lebih parameter yang dipisahkan dengan tanda baca koma (,) ke browser. Sebagai contoh:

File: Lat30.php

```

<html>
<head>
<title>Fungsi echo</title>
</head>
<body>
<?php
echo "Aku datang", 1, 2.0, 3.5, " kamu pergi";
?>
</body>
</html>

```

print (*string_output*)

Fungsi ini hampir sama dengan fungsi *echo* yaitu untuk mengirim output ke browser. Sebagai contoh:

File: Lat31.php

```

<html>
<head>
<title>Fungsi print</title>
</head>
<body>
<?php
print("Hai semua, kunjungi www.amikom.ac.id ya
");
?>
</body>
</html>

```

printf (*string_format, argumen ...*)

Fungsi ini hampir sama dengan fungsi *print*, yaitu untuk mengirim output ke browser. Fungsi ini sering digunakan untuk menampilkan hasil yang bentuknya bisa diatur.

Format	Keterangan
D	Integer, notasi desimal
B	Integer, notasi binary
O	Integer, notasi oktal
X	Integer, notasi hexadesimal, dinyatakan dalam huruf kecil
X	Integer, notasi hexadesimal, dinyatakan dalam huruf kapital
C	Karakter yang nilai ASCII-nya dinyatakan dalam argumen
S	String
F	Double (bilangan real)

Contoh penggunaan fungsi *printf* adalah:

File: Lat32.php

```
<html>
  <head>
    <title>Fungsi printf</title>
  </head>
  <body>
    <?php
      printf("%s %d %f <br>\n", "string", 10, 3.14);
    ?>
  </body>
</html>
```

copy(string_source, string_destination)

Fungsi ini digunakan untuk menyalin suatu file yang ditentukan dalam argumen *source* (lokasi asal file) menuju ke *destination* (lokasi tujuan). Akan menghasilkan *true* jika berhasil.

File: Lat33.php

```
<html>
  <head>
    <title>Fungsi copy</title>
  </head>
  <body>
    <?php
      if (copy("data.txt","tmp/data.txt"))
      {
        print("data.txt disalin ke /tmp");
      }
      else
      {
        print("data.txt tidak dapat disalin !");
      }
    ?>
  </body>
</html>
```

include(string_filename)

Judul *filename* merupakan nama file yang akan dijalankan dengan fungsi *include*. Fungsi ini sering digunakan untuk menyisipkan file yang dibutuhkan dalam argumen. Amati contoh di bawah ini:

File: Lat34.php

```
<html>
  <head>
    <title>Fungsi include</title>
  </head>
  <body>
    <?php
      include("fungsi_print.php");
    ?>
  </body>
</html>
```

phpinfo()

Dengan fungsi ini akan didapatkan informasi tentang versi PHP, pembuatnya, sistem operasi pada web server, konfigurasi variabel, dan sebagainya.

File: Lat35.php

```
<html>
  <head>
    <title>Fungsi phpinfo()</title>
  </head>
  <body>
    <?php
      phpinfo();
    ?>
  </body>
</html>
```

version()

Fungsi ini hanya akan memberi informasi mengenai versi PHP yang kita gunakan.

Contoh:

File: Lat36.php

```
<html>
<head>
    <title>Fungsi phpversion()</title>
</head>
<body>
<?php
    printf("Versi PHP-ku adalah %s", phpversion());
?>
</body>
</html>
```

strlen(string_text)

Fungsi ini akan mengembalikan nilai integer yang merupakan panjang string dalam argumen text. Lihat contoh berikut ini:

File: Lat37.php

```
<html>
<head>
    <title>Fungsi strlen</title>
</head>
<body>
<?php
    print(strlen("stmikamikom"));
?>
</body>
</html>
```

ord(string_character)

Fungsi ini akan mengembalikan nilai ASCII dari karakter pada argumen. Jadi argumennya hanyalah sebuah karakter. Perhatikan contoh di bawah ini:

File: Lat38.php

```
<html>
<head>
```

```
<title>Fungsi ord</title>
</head>
<body>
<?php
    print(ord("B"));
?>
</body>
</html>
```

strtolower(string_text)

Fungsi ini akan mengubah argumen menjadi huruf kecil semua. Untuk jelasnya Anda dapat mencoba contoh berikut ini:

File: Lat39.php

```
<html>
<head>
    <title>Fungsi strtolower</title>
</head>
<body>
<?php
    print(strtolower("WWW.RudyantoArief.Com"));
?>
</body>
</html>
```

strtoupper(string_text)

Fungsi ini merupakan kebalikan dari fungsi *strtolower*. Semua huruf dalam argumen text yang bertipe string akan diubah menjadi huruf kapital dan menjadi keluaran dari fungsi ini. Perhatikan contoh di bawah ini:

File: Lat40.php

```
<html>
<head>
    <title>Fungsi strtoupper</title>
</head>
<body>
<?php
```

```

print(strtoupper("WWW.RudyantoArief.Com"));
?>
</body>
</html>

```

8 Fungsi Tanggal dan Jam

date(string_format, integer_timestamp)

Fungsi **date** mengembalikan tanggal atau waktu tergantung dari argumen yang dimasukkan. Daftar *string_format* terdapat dalam tabel di bawah ini. Argumen *timestamp* bersifat *optional*.

Kode	Keterangan
A	am atau pm
A	AM atau PM
D	Tanggal (<i>date</i>)
D	Nama hari (singkatan)
F	Nama bulan
H	Jam 01 sampai 12
H	Jam 00 sampai 23
I	Menit
J	Tanggal
L	Nama hari (lengkap)
M	Bulan (dalam angka)
M	Bulan (singkatan)
Y	Tahun dalam dua digit
Y	Tahun dalam empat digit
Z	Hari ke- dari tahun

Perhatikan contoh:

File: Lat41.php

```

<html>
<head>
    <title>Fungsi date</title>
</head>
<body>
<?php
    print("Yogyakarta, ".date("d F Y"));
    print("<br>\n");
    print(date("h:i A l"));
?>
</body>
</html>

```

getdate(integer_timestamp)

Fungsi **getdate** digunakan untuk menghasilkan waktu dengan keluaran bertipe *array*. Argumen *timestamp* bersifat *optional*. Daftar elemen **getdate** dapat dilihat dalam tabel berikut:

Elemen	Keterangan
hours	Jam
mday	Hari
minutes	Menit
mon	Bulan dalam digit
month	Bulan
seconds	Detik
wday	Hari dalam digit
weekday	Hari
yday	Hari ke- dari tahun
year	Tahun

Perhatikan contoh berikut:

File: Lat42.php

```
<html>
<head>
<title>Ucapan selamat dengan fungsi
getdate</title>
</head>
<body>
<?php
$waktu = getdate();
if($waktu[hours] <= 9)
{
    echo "Selamat Pagi";
}
elseif($waktu[hours] <= 14)
{
    echo "Selamat Siang";
}
elseif($waktu[hours] <= 18)
{
    echo "Selamat Sore";
}
else
{
    echo "Selamat Malam";
}
?>
</body>
</html>
```

`Sleep(integer_seconds)`

Fungsi `sleep` menyebabkan proses tertunda selama nilai pada argumen `seconds`, seperti tampak pada contoh berikut:

File: Lat43.php

```
<html>
<head>
<title>Fungsi sleep</title>
</head>
<body>
<?php
```

```
print(microtime());
sleep(10);
print("<br>\n");
print(microtime());
?>
</body>
</html>
```

`Sleep(integer_microseconds)`

Fungsi ini menyebabkan proses tertunda selama nilai pada argumen `microseconds`. Perhatikan contoh berikut ini:

File: Lat44.php

```
<html>
<head>
<title>Fungsi usleep</title>
</head>
<body>
<?php
print(microtime());
usleep(50);
print("<br>\n");
print(microtime());
?>
</body>
</html>
```

`Fungsi Matematika`

`Abs(number_value)`

Fungsi ini akan menghasilkan nilai mutlak dari nilai pada argumen. Jika nilai argumen bernilai positif, maka fungsi `abs` akan mengembalikan nilai tersebut. Jika nilai argumen bernilai negatif, fungsi `abs` akan mengalikan dengan negatif satu (-1). Perhatikan contoh berikut ini:

File: Lat45.php

```
<html>
  <head>
    <title>Fungsi abs</title>
  </head>
  <body>
    <?php
      print(abs(-10));
    ?>
  </body>
</html>
```

cos(double_angle)

Fungsi ini akan mengembalikan nilai cosinus dari nilai sudut (dalam radian). Perhatikan contoh di bawah ini:

File: Lat46.php

```
<html>
  <head>
    <title>Fungsi cos</title>
  </head>
  <body>
    <?php
      print(cos(4 * pi()));
    ?>
  </body>
</html>
```

exp(double_power)

Fungsi ini akan mengembalikan nilai eksponen dari argumen. Lihat contoh berikut ini:

File: Lat47.php

```
<html>
  <head>
    <title>Fungsi exp</title>
  </head>
  <body>
```

```
<?php
  print(exp(1));
?>
</body>
</html>
```

log10(double_value)

Fungsi ini akan mengembalikan nilai log10 dari argumen. Perhatikan contoh berikut:

File: Lat48.php

```
<html>
  <head>
    <title>Fungsi log10</title>
  </head>
  <body>
    <?php
      print(log10(100));
    ?>
  </body>
</html>
```

pi()

Fungsi ini akan memberikan nilai pendekatan dari π ($22/7$). Perhatikan contoh berikut:

File: Lat49.php

```
<html>
  <head>
    <title>Fungsi pi()</title>
  </head>
  <body>
    <?php
      print(pi());
    ?>
  </body>
</html>
```

```
pow(double_base, double_power)
    ;
```

Fungsi ini digunakan untuk menghasilkan nilai basis pangkat nilai power. Perhatikan contoh berikut:

File: Lat50.php

```
<html>
  <head>
    <title>Fungsi pow()</title>
  </head>
  <body>
    <?php
      print (pow(2, 10));
    <?>
    </body>
  </html>
```

round(double_value)

Fungsi ini akan mengubah nilai pada argumen (bertipe *double*) menjadi integer yang terdekat (pembulatan). Perhatikan contoh berikut:

File: Lat51.php

```
<html>
  <head>
    <title>Fungsi round()</title>
  </head>
  <body>
    <?php
      print (round(2, 4)."<br>");
      print (round(2, 5)."<br>");
      print (round(2, 6)."<br>");
    <?>
    </body>
  </html>
```

sin(double_angle)

Fungsi ini menghasilkan nilai sinus dari nilai sudut (dalam radian) pada argumen.

Perhatikan contoh berikut:

File: Lat52.php

```
<html>
  <head>
    <title>Fungsi sin()</title>
  </head>
  <body>
    <?php
      print (sin(2 * pi()));
    <?>
    </body>
  </html>
```

sqrt(double_value)

Fungsi ini digunakan untuk mencari nilai akar dari argumen value.

Perhatikan contoh berikut:

File: Lat53.php

```
<html>
  <head>
    <title>Fungsi sqrt()</title>
  </head>
  <body>
    <?php
      print (sqrt(100.0));
    <?>
    </body>
  </html>
```

tan(double_angle)

Fungsi ini akan menghasilkan nilai tangen dari sudut (dalam radian) pada argumen angle. Perhatikan contoh berikut:

File: Lat54.php

```
<html>
  <head>
    <title>Fungsi tan()</title>
  </head>
  <body>
    <?php
      print (tan(2 * pi()));
    ?>
  </body>
</html>
```

0 Fungsi Array dan Variabel

```
define(string_constant, string_value, boolean
non_case_sensitive)
```

Nilai pada argumen bisa bertipe string, integer, atau double. Argumen *case sensitive* bersifat opsional. Default fungsi ini adalah *case sensitive*. Contoh implementasinya pada penulisan program adalah:

File: Lat55.php

```
<html>
  <head>
    <title>Fungsi define()</title>
  </head>
  <body>
    <?php
      define("ini", "SAYA SEORANG");
      define("itu", "MAHASISWA AMIKOM");
      print(ini . "<br>\n");
      print(itu . "<br>\n");
    ?>
  </body>
</html>
```

• *empty(variable)*

Fungsi ini akan mengembalikan nilai true jika variabel belum mempunyai nilai, dan sebaliknya false jika variabel telah diberi nilai. Perhatikan contoh berikut:

File: Lat56.php

```
<html>
  <head>
    <title>Fungsi empty()</title>
  </head>
  <body>
    <?php
      if(empty($nama))
        print("Tolong isi nama Anda segera!");
      else
        print("Nama Anda adalah $nama");
    ?>
  </body>
</html>
```

• *array(expression)*

Fungsi ini akan mengembalikan nilai true jika ekspresi adalah array, dan sebaliknya false. Perhatikan contoh berikut:

File: Lat57.php

```
<html>
  <head>
    <title>Fungsi is_array()</title>
  </head>
  <body>
    <?php
      $isi = array("buku", "pensil", " pena");
      if(is_array($isi))
        print("Variabel isi adalah variable array");
    ?>
  </body>
</html>
```

```

        }
    ?>
</body>
</html>

is_double(expression)

```

Fungsi ini akan mengembalikan nilai true jika nilai pada ekspresi bertipe double. Perhatikan contoh berikut:

File: Lat58.php

```

<html>
<head>
    <title>Fungsi is_double()</title>
</head>
<body>
<?php
    $dollar = 8000.25;
    if(is_double($dollar))
    {
        print("Variabel dollar bertipe double");
    }
?>
</body>
</html>

```

is_integer(expression)

Fungsi ini akan mengembalikan nilai true jika nilai pada ekspresi bertipe integer. Perhatikan contoh berikut:

File: Lat59.php

```

<html>
<head>
    <title>Fungsi is_integer()</title>
</head>
<body>
<?php
    $harga = 8000;
?>

```

```

if(is_integer($harga))
{
    print("Variabel harga bertipe integer");
}
?>
</body>
</html>

is_string(expression)

```

Fungsi ini akan mengembalikan nilai true jika nilai pada ekspresi bertipe string. Perhatikan contoh berikut:

File: Lat60.php

```

<html>
<head>
    <title>Fungsi is_string()</title>
</head>
<body>
<?php
    $hero = "Spiderman";
    if(is_string($hero))
    {
        print("Variabel hero bertipe string");
    }
?>
</body>
</html>

```

isset(variable)

Fungsi ini akan mengembalikan nilai true jika variabel telah memiliki nilai, dan nilainya false jika variabel belum diberi nilai. Perhatikan contoh berikut:

File: Lat61.php

```

<html>
<head>
    <title>Fungsi isset()</title>

```

```

</head>
<body>
<?php
    if(isset($nama))
    {
        print("Nama Anda adalah $nama");
    }
    else
    {
        print("Tolong isi nama Anda segera!");
    }
?>
</body>
</html>

count(variable_array)

```

Fungsi ini akan mengembalikan nilai berupa jumlah elemen array di argumen variabel. Jika variabel belum diisi maka fungsi akan mengembalikan nilai 0. Jika variabel bukan array, fungsi akan mengembalikan nilai null. Perhatikan contoh berikut:

File: Lat62.php

```

<html>
    <head>
        <title>Fungsi count()</title>
    </head>
    <body>
        <?php
            $hero = array("Spiderman", "Wolvrine",
'Cyclop');
            print(count($hero));
        ?>
    </body>
</html>

```

3.3 include dan Require

require() dan *include()* itu benar-benar sama cara kerjanya, tetapi tentu saja tidak, sebab jika sama fungsinya tentu tidak selayaknya ada dua fungsi. Perbedaan mendasar antara kedua fungsi ini adalah:

require() akan selalu digantikan oleh isi dari file yang ditunjuk dalam fungsi ini dan tidak dapat digunakan dalam percabangan/perkondisionan (seperti perkondisionan "jika ini maka require file itu"). Karena file yang ditunjuk akan selalu direferensi tanpa peduli struktur/aliran script.

include() akan mengatur pembacaan file yang ditunjuk sesuai dengan kondisi struktur/aliran script, sehingga fungsi ini dapat digunakan pada percabangan/perkondisionan.

Menurut ciri-ciri di atas, *require()* akan sesuai digunakan untuk menyertakan file yang berisikan variabel dan fungsi-fungsi global yang selanjutnya akan dipakai pada seluruh bagian dari script utama. Sementara *include()* akan digunakan untuk menyisipkan kode program/ script atau tag yang dibutuhkan pada program/script utama, misalkan untuk header atau footer setiap halaman dalam sebuah situs.

Penting untuk diingat bahwa yang penting untuk kedua fungsi ini, parser PHP akan meninggalkan mode PHP dan kembali ke mode HTML standar pada saat membaca file yang ditunjukkan oleh kedua fungsi ini. Itu sebabnya pada contoh di atas, file yang ditunjuk oleh fungsi-fungsi ini selalu dimulai dengan tag <?php dan diakhiri dengan tag ?> untuk mengembalikan mode file ke mode script PHP.

BAB 10 Form

Form merupakan salah satu elemen HTML yang digunakan untuk menerima input pengunjung. Melalui form pengunjung sebuah halaman web dapat melakukan interaksi dengan web. Pengunjung dapat memasukkan informasi melalui elemen-elemen input seperti textfield, checkbox, radio button, dropdown list, submit button dan masih banyak lainnya. Selain itu pengunjung juga dapat melakukan pemilihan data menggunakan elemen select list atau dropdown list dengan istilah combo box atau drop-down list. Form berfungsi untuk membuat sebuah halaman web menjadi web interaktif dan dinamis.

10.1 Perintah Dasar Form

Atribut	Fungsi
	<p>Metode pengiriman data ke tujuan, yaitu:</p> <p>GET: Mengirim data pada server dengan cara meletakkan data pada bagian akhir URL yang ditunjuk</p> <p>POST: Mengirim datanya secara terpisah</p>
	Menentukan lokasi dari script yang akan memproses data dari form.

10.2 Elemen Input Form

Elemen yang paling penting dalam form adalah elemen input. Elemen ini memungkinkan pengunjung memasukkan data melalui halaman form yang datanya akan dikirimkan ke server. Elemen input digunakan untuk menyampaikan informasi yang dimasukkan oleh pengunjung. Elemen input dapat berupa dari berbagai macam cara tergantung dari atributnya, seperti jenis text like checkbox, password, radio button, submit button, dan lainnya.

10.3 Text Field

Merupakan elemen input yang memungkinkan pengunjung memasukkan text dalam satu baris. Lebar karakter text field defaultnya adalah 20 karakter.

Perintah dasar:

```
<input type="text" />
```

Contoh penggunaan:

```
<form>
First name: <input type="text" name="firstname" />
Last name: <input type="text" name="lastname" />
</form>
```

Tampilannya di browser:

First name:	<input name="firstname" type="text"/>
Last name:	<input name="lastname" type="text"/>

Keterangan: form sifatnya invisible, tidak terlihat di browser.

10.4 Password Field

Perintah sama dengan text field yaitu untuk memasukkan satu baris teks dalam format ciphertext/password. Karakter yang ditampilkan di password field akan diubah menjadi tanda bintang (*) atau bulatan (●).

Perintah dasar:

```
<input type="password" />
```

Contoh penggunaan:

```
<input type="password" name="pwd" />
```

Tampilannya di browser (sebelum diisi dan setelah diisi):

password:	<input name="pwd" type="password"/>
password:	*****

10.5 Radio Button

Perintah form yang membolehkan pengunjung memilih satu saja pilihan data dari pilihan data yang terbatas. Seperti pemilihan jenis kelamin (sex atau gender).

Perintah dasar:

```
<input type="radio" />
```

Contoh penggunaan:

```
<input type="radio" name="sex" value="male" /> Male<br/>
<input type="radio" name="sex" value="female" /> Female
```

Tampilannya di browser:

- Male
- Female

10.6 Check Box

Digunakan untuk memilih beberapa pilihan data sekaligus atau hanya satu pilihan dari daftar pilihan yang ada.

Perintah dasar:

```
<form>
<input type="checkbox" name="vehicle" value="Bike">
I have a bike<br />
<input type="checkbox" name="vehicle" value="Car">
I have a car
</form>
```

Contoh penggunaan:

- I have a bike
- I have a car

10.7 Submit Button

Digunakan untuk mengirimkan data dari halaman web ke server. Data dikirimkan ke halaman lain yang telah ditentukan pada atribut `action` form. File yang ditentukan pada atribut `action` biasanya akan melalui suatu aksi tertentu yang akan mengolah data yang dimasukkan dari pengguna.

Perintah dasar:

```
<input type="submit" />
```

Contoh penggunaan:

```
<form name="input" action="proses_input.php"
method="get">
username: <input type="text" name="user" />


```

Tampilan di browser:

Username:

Penjelasan: jika pengunjung memasukkan username pada textfield dan mengklik tombol submit, maka data yang dimasukkan akan dikirim ke halaman "proses_input.php".

10.8 Reset Button

Digunakan untuk membatalkan semua proses pengisian data yang dilakukan pengguna dalam input form.

Perintah dasar:

```
<input type="reset" />
```

Contoh penggunaan:

```
<form name="input" action="proses_input.php"
method="get">
username: <input type="text" name="user" />


```

Tampilan di browser:

Username:

Username:

Username:

Keterangan: jika pengunjung memasukkan username pada textfield kemudian mengklik tombol reset (gambar ke-2 dari tampilan browser di atas), maka data yang dimasukkan akan dibersihkan/reset dari textfield.

10.9 Select List (drop-down list)

Digunakan untuk menampilkan daftar pilihan dalam bentuk drop-down list.

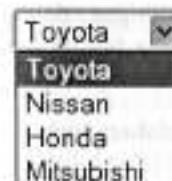
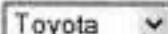
Perintah dasar:

```
<select>
  ...
</select>
```

Contoh penggunaan:

```
<form name="input" action="proses_input.php">
<select name="mobil">
<option value="toyota">Toyota</option>
<option value="nissan">Nissan</option>
<option value="honda">Honda</option>
<option value="mitsubishi">Mitsubishi</option>
</select>
</form>
```

Tampilan di browser:



10.10 Textarea

Elemen TEXTAREA berfungsi sebagai input kontrol form untuk memasukkan teks lebih dari satu baris. Elemen ini mempunyai atribut yaitu name, cols, dan rows. Atribut name mendefinisikan nama input kontrol form elemen textarea. Atribut cols mendefinisikan jumlah kolom textarea yang terlihat, dan atribut rows mendefinisikan jumlah baris textarea yang terlihat. Elemen ini harus berada di dalam value berisikan nilai elemen FORM.

Perintah dasar:

```
<textarea>
  name="name"
  cols="number"
  rows="number" >
  .....
</textarea>
```

10.11 Penggabungan Beberapa Elemen Form dalam Satu Halaman

Berikut adalah contoh penerapan semua elemen input form dan bagaimana pengolahan variabel yang dimasukkan di elemen form tersebut. Ada 3 file yang harus dibuat, yaitu:

- File style.css yang berisikan script css (cascading style sheet) yang berfungsi untuk memformat tampilan halaman form.

- File form.html yang menampilkan sebuah halaman input dengan beberapa contoh elemen input di dalamnya. Action pada file ini akan diarahkan ke file lain yang berisikan script php.
- File view.php yang berfungsi untuk melakukan pengolahan terhadap masukan nilai yang dilakukan di halaman form.html.
- Buatlah ke-3 file tersebut dan simpan di dalam direktori kerja yang sudah dibuat untuk menyimpan semua dokumen latihan di buku. Simpan di dalam dokumen root web server apache yaitu di folder "htdocs".

Nama file: style.css

```
@charset "utf-8";
/* CSS Document */
th {
    background-color:#33FF00;
    font-family:Calibri;
    font-size:24px;
}

.table1 {
    background-color:#0099CC;
    border-color:#CC0066;
}

.table2 {
    background-color:#CCFFCC;
}

#text {
    font-family:Calibri;
    font-size:20px;
    color:#000033;
}
```

Nama file: form.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Untitled Document</title>
<link href="style.css" rel="stylesheet"
type="text/css">
</head>

<body>
<form action="view.php" method="get">





```

```

Laki-laki &nbsp;&nbsp;&nbsp;
    <input type="radio" name="jk"
value="Perempuan" />
    Perempuan</td>
</tr>
<tr>
    <td width="25%" id="text">ASAL</td>
    <td align="center"><b>:</b></td>
    <td>
        <select name="asal">
            <option value="-">-</option>
            <option
value="Yogyakarta">Yogyakarta</option>
            <option value="Jakarta">Jakarta</option>
            <option value="Jawa Timur">Jawa
Timur</option>
        </select>
    </td>
</tr>
<tr>
    <td width="25%" id="text">NO TELP</td>
    <td align="center"><b>:</b></td>
    <td><input name="telp" type="text" id="telp"
size="50" maxlength="13" /></td>
</tr>
<tr>
    <td width="25%" id="text">ALAMAT EMAIL</td>
    <td align="center"><b>:</b></td>
    <td><input name="email" type="text"
id="email" size="50" /></td>
</tr>
<tr>
    <td width="25%" id="text">PASSWORD</td>
    <td align="center"><b>:</b></td>
    <td><input name="password" type="password"
id="password" size="50" /></td>
</tr>
<tr>
    <td width="25%" valign="top"
id="text">HOBBY</td>
    <td align="center" valign="top"><b>:</b></td>
    <td><table width="100%" border="0">
        <tr>

```

```

            <td width="48%">
                <input type="checkbox" name="hb1"
value="Sepakbola" />Sepakbola
                <br />
                <input type="checkbox" name="hb2"
value="Futsal" />Futsal
                <br />
                <input type="checkbox" name="hb3"
value="Berenang" />Berenang
            </td>
            <td width="52%"><input type="checkbox"
name="hb4" value="Membaca" />Membaca
                <br />
                <input type="checkbox" name="hb5"
value="Jogging" />Jogging
                <br />
                <input type="checkbox" name="hb6"
value="nGoWes" />nGoWes</td>
        </tr>
    </table></td>
</tr>
<tr>
    <td width="25%" valign="top"
id="text">DESKRIPSI DIRI</td>
    <td align="center"
valign="top"><b>:</b></td>
    <td>
        <textarea name="desk" cols="50"
rows="5" /></textarea>
    </td>
</tr>
</table></td>
</tr>
<tr>
    <td align="center">
        <input type="submit" value="OK"
&nbsp;&nbsp;&nbsp;
        <input type="reset" value="Batal" />
    </td>
</tr>
</table>
</form>
</body>
</html>

```

Nama file: view.php

```

<?php
//menangkap value form yang disimpan dalam sebuah variabel
$nim = $_GET['nim'];
$nama = $_GET['nama'];
$jkelamin = $_GET['jkelamin'];
$asal = $_GET['asal'];
$telepon = $_GET['telepon'];
$email = $_GET['email'];
$password = $_GET['password'];
$jk = $_GET['jk'];
$deskripsi = $_GET['deskripsi'];

//menampilkan dari variabel diatas dalam browser
echo $nim . '<br />';
echo $nama . '<br />';
echo $jkelamin . '<br />';
echo $asal . '<br />';
echo $telepon . '<br />';
echo $email . '<br />';
echo $password . '<br />';
echo $jk . '<br />';
echo $deskripsi . '<br />';
?>

```

Tampilan di browser (form.html):

FORM REGISTRASI

NIM:

NAMA:

JENIS KELAMIN:

Laki-laki Perempuan

TELP:

ALAMAT EMAIL:

PASSWORD:

HOBBI:

Sepak bola Makan Jogging Belajar

DESKRIPSI DIRI:

WALAU DI SUDAH MULAI KERJAKAN, SAYA YETU...

OK | Batal

Tampilan di browser (view.php):

FORM REGISTRASI

NIM:

NAMA:

JENIS KELAMIN:

Laki-laki Perempuan

TELP:

ALAMAT EMAIL:

PASSWORD:

HOBBI:

Sepak bola Makan Jogging Belajar

DESKRIPSI DIRI:

WALAU DI SUDAH MULAI KERJAKAN, SAYA YETU...

OK | Batal

http://localhost/Sekolah/Beasiswa.php?nim=10.11.12.13&nama=Mohammed+Rofiqudin&label=NIM&label>Nama&label=TTL

The Belajar Jari Home Apple Safari Google Play YouTube YouTube Subtitle Wikipedia News (SD) People

10.11.12.13
Mohammed Rofiqudin Arif
Yogakarta
0374-884701
rody@unikom.ac.id

Lahir di STMIK AMIKOM YOGAKARTA, Tempat Kelahiran: Bogor, web: www.unikom.ac.id
Sekolah: Sekolah
Membaca

Contoh Lain:

Nama File: beasiswa.html

```
<html>
<head>
<title>Form Beasiswa Sederhana</title>
<style type="text/css">
body { background : url(bg.gif) repeat-y top center #000; width : 800px; margin : 0 auto; padding : 10px ;}
a:link { text-decoration:none; color:white;}
a:hover { color:black; background-color:#fff; font-style:italic;}
fieldset { border: 2px solid #333; width: 700px; background-color: transparent; margin:40px; }
legend { background: #222; border: 2px solid #333; padding: 5px; }
td { font-size: 12px; padding:5px; }
.ijo { margin: 0 auto;color:white; }
#label { font-style: italic; width:150px; }
</style>
</head>
<body>
<form action="Form" method="post">
<fieldset>
<legend><a href="">Form Pengajuan Beasiswa</a></legend>
<br/>
<table border="0" width="680px" class="ijo">
```

```
<tr>
<td id="label">NIM</td>
<td></td>
<td><input size="20" type="text" name="Nim" value="" /></td>
</tr>
<tr>
<td id="label">NAMA</td>
<td></td>
<td><input size="45" type="text" name="Nama" value="" /></td>
</tr>
<tr>
<td id="label">TTL</td>
<td></td>
<td><input size="20" type="text" />
<select title="Date">
<option>Tanggal</option>
<option>1</option>
<option>2</option>
<option>3</option>
<option>4</option>
<option>5</option>
<option>6</option>
<option>7</option>
<option>8</option>
<option>9</option>
<option>10</option>
<option>11</option>
<option>12</option>
<option>13</option>
<option>14</option>
<option>15</option>
<option>16</option>
<option>17</option>
<option>18</option>
<option>19</option>
<option>20</option>
<option>21</option>
<option>22</option>
<option>23</option>
<option>24</option>
<option>25</option>
```

```

        <option>26</option>
        <option>27</option>
        <option>28</option>
        <option>29</option>
        <option>30</option>
        <option>31</option>
    </select>
    <select>
        <option>Bulan</option>
        <option>Januari</option>
        <option>Februari</option>
        <option>Maret</option>
        <option>April</option>
        <option>May</option>
        <option>Juni</option>
        <option>Juli</option>
        <option>Agustus</option>
        <option>September</option>
        <option>Oktober</option>
        <option>November</option>
        <option>Desember</option>
    </select>
    <select>
        <option>Tahun</option>
        <option>2011</option>
        <option>2010</option>
        <option>2009</option>
        <option>2008</option>
        <option>2007</option>
        <option>2006</option>
        <option>2005</option>
        <option>2004</option>
        <option>2003</option>
        <option>2002</option>
        <option>2001</option>
        <option>2000</option>
        <option>1999</option>
        <option>1998</option>
        <option>1997</option>
        <option>1996</option>
        <option>1995</option>
        <option>1994</option>
        <option>1993</option>

```

```

        <option>1992</option>
        <option>1991</option>
        <option>1990</option>
        <option>1989</option>
        <option>1988</option>
        <option>1987</option>
        <option>1986</option>
        <option>1985</option>
        <option>1984</option>
        <option>1983</option>
        <option>1982</option>
        <option>1981</option>
        <option>1980</option>
    </select>
</td>
</tr>
<tr>
    <td id="label">JENIS KELAMIN</td>
    <td>:</td>
    <td>
        <input type="radio"
        >Kelamin" value="Laki - laki" />Laki-laki
        <input type="radio"
        >Kelamin" value="Perempuan" />Perempuan
    </td>
</tr>
<tr>
    <td id="label">ALAMAT</td>
    <td>:</td>
    <td><input size="45" type="text"
        >Alamat" value="" /></td>
</tr>
<tr>
    <td id="label">PENGHASILAN ORTU</td>
    <td>:</td>
    <td>
        <select>
            <option>-</option>
            <option>Dibawah 1
            </option>
            <option>1 - 2 Juta</option>
            <option>2 - 3 Juta</option>
            <option>3 - 4 Juta</option>

```

```

<option>Diatas 4 Juta</option>
    </select>
  </td>
</tr>
<tr>
    <td width="20%" valign="top" id="label">KELUARGA YANG MASIH HIDUP</td>
    <td valign="top">:</td>
    <td>
        <table width="100%" class="ijo">
            <tr>
                <td width="50%">
                    <input type="checkbox" name="Hobi" />BAPAK<br/>
                    <input type="checkbox" name="Hobi" />IBU<br/>
                    <input type="checkbox" name="Hobi" />KAKAK
                </td>
                <td width="50%">
                    <input type="checkbox" name="Hobi" />ADIK<br/>
                    <input type="checkbox" name="Hobi" />PAMAN<br/>
                    <input type="checkbox" name="Hobi" />BIBI<br/>
                    ...
                </td>
            </tr>
        </table>
    </td>
</tr>
<tr>
    <td valign="top" id="label">ALASAN</td>
    <td>
        Alasan anda mengambil beasiswa ...</td>
    </td>
</tr>
<tr>
    <td colspan="2"></td>
    <td>

```

```

        <input type="Submit" name="Submit" value="Submit" />
        <input type="reset" name="Reset" value="Reset" />
    </td>
</tr>
<tr>
    <td colspan="3" style="text-align:center; vertical-align:bottom; font-size:small; font-style:italic; border-top:1px solid black; padding-top:5px; margin-top:5px;">
        <small>Copyright © 2008 by Siti Hajar. All rights reserved. This document may not be reproduced in whole or in part without written permission from the author. Last updated: 2008-08-15</small>
    </td>
</tr>

```

Bentuk di Dreamweaver:



Bentuk website di makalah ini menggunakan tampilan yang sederhana dan mudah dipahami. Penggunaan radio button untuk jenis kelamin memungkinkan pengguna untuk memilih hanya satu pilihan saja. Pengguna juga dapat memilih beberapa hobi dengan menggunakan checkbox. Untuk menambahkan pilihan hobi, pengguna dapat menambahkan checkbox baru ke dalam bagian hobi. Untuk menambahkan pilihan gender, pengguna dapat menambahkan radio button baru ke dalam bagian gender. Untuk menambahkan pilihan alasan mengambil beasiswa, pengguna dapat menambahkan textarea baru ke dalam bagian alasan. Untuk menambahkan tombol submit dan reset, pengguna dapat menambahkan input type="submit" dan input type="reset" ke dalam bagian tombol. Untuk menambahkan copyright, pengguna dapat menambahkan komentar pada bagian footer.

Tampilan di Browser:

10.12 Method Post dan Get

Pada elemen form terdapat 2 method yang dapat digunakan untuk mengirimkan data dari halaman form ke server. GET dan POST adalah metode pengiriman data dari suatu client ke server melalui browser. Masing-masing diterima oleh server dalam bentuk variabel array `$_GET` dan `$_POST` yang bersifat public. Jika method post yang dipilih, maka semua variabel nilai yang dimasukkan di elemen input yang ada di form akan dikirim ke server tidak akan ditampilkan di bagian address bar browser. Sedangkan jika method get yang digunakan, maka semua variabel nilai yang dimasukkan di dalam form ketika dikirim akan ditampilkan di bagian address bar browser. Pada contoh penggunaan form di atas (form.html), method yang digunakan adalah method post. Ketika nilai variabelnya dikirimkan ke file view.php, maka semua nilai variabel yang dimasukkan dan diproses oleh file view.php, maka semua nilai variabel yang dimasukkan akan ditampilkan pada bagian address bar browser. Jika data yang dikirimkan ke server merupakan data yang sangat rahasia, sebaiknya gunakan method post yang digunakan.

10.12.1 Metode Get

Contoh GET

Metode GET adalah sebuah fungsi yang digunakan untuk mengumpulkan data dari sebuah form yang dikirimkan menggunakan metode GET. Data yang dikirimkan dari sebuah form menggunakan metode GET dapat dilihat oleh semua orang, maksudnya adalah nilai dari form tersebut dapat dilihat dengan jelas pada jendela Address Bar pada browser. Metode GET memiliki keterbatasan terhadap banyaknya informasi yang akan dikirim. Metode GET tidak cocok untuk nilai variabel yang sangat besar. Metode GET dapat digunakan untuk menyimpan variabel yang nilainya lebih dari 255 karakter.

Berikut adalah contoh penggunaan metode GET (sumber: www.w3schools.com).

File	latihan/get.php
fungsi	Untuk menghubungkan php ke database tokobuku online

```

<?php
    $name = $_GET['fname'];
    $age = $_GET['age'];

    echo "Name: $name";
    echo "Age: $age";
?>

```

Jika user mengklik tombol submit, URL akan mengirimkan nilai form ke server seperti berikut:

`http://localhost/latihan/welcome.php?fname=rudyanto&age=32`

`http://localhost/latihan/welcome.php?fname=rudyanto&age=32`

Welcome rudyanto.
You are 32 years old!

File "welcome.php" dapat menggunakan fungsi `$_GET` untuk mengepulkan data. Nama dari field yang ada di form otomatis akan menjadi key pada array `$_GET`.

Nama File	latihan/welcome.php
Keterangan	Untuk menangkap nilai variabel dari form get.php Welcome <?php echo \$_GET["fname"]; ?>. You are <?php echo \$_GET["age"]; ?> years old!

http://nama_host/nama_program.php?variabel1=nilai1&variabel2=nilai2&variabel3=nilai3..dst..

Contoh:

Misal kita membuat program php sebagai berikut:

Nama File	latihan/form.html
Keterangan	<!-- query_form.html --> <h2>Form Isian</h2> <form method="POST" action="query2.php"> NIM: <input type="TEXT" name="nim"> Nama <input type="TEXT" name="nama" size="1"> Nilai: <input type="TEXT" name="nilai" size="5"> <input type="SUBMIT" value="Kirim"> </form>

Setelah file diupload ke server maka akan muncul halaman seperti berikut.

File yang diproses oleh file berikut dengan menggunakan method GET.

File	latihan/query.php
Konten	<pre>query.php Selamat Datang
; Data yang dikirim dengan metode GET adalah sebagai berikut:
; NIM : '.\$_GET['nim'].
; Nama: '.\$_GET['nama'].
; Nilai: '.\$_GET['nilai'].
;</pre>

berikut kita masukan pada URL-bar suatu browser,
<localhost/query.php?nim=07.11.1679&nama=Bima&nilai=99>

Outputnya adalah sebagai berikut:

Selamat Datang

Data yang dikirim dengan metode GET adalah sebagai berikut:

NIM: 07.11.1679

Nama: Bima

Nilai: 99

Programmer harus berhati-hati dengan masalah security saat mengirimkan variabel melalui url, karena nama variabel dan nilainya tidak tampak jelas oleh user. User dengan leluasa memasukkan nilai variabel dengan karakter yang bisa membahayakan website seakan database.

Angiriman variabel melalui metode GET hanya efisien untuk jumlah data yang sedikit. Dapat dibayangkan berapa panjang url yang harus dibuat untuk menampung jumlah data yang banyak. Sementara browser memiliki batas maksimum panjang url yang dapat dituliskan.

Haruskan melakukan tahap sanitize terlebih dahulu terhadap variabel `$_GET` maupun `$_POST` sebelum diproses.

10.12.2 Metode Post

Fungsi `$_POST` digunakan untuk mengumpulkan nilai variabel dari sebuah form yang dikirimkan menggunakan metode post. Informasi yang dikirimkan dari sebuah form menggunakan metode POST tidak dapat dilihat oleh orang lain pada jendela address bar di browser dan tidak memiliki pembatasan pada banyaknya informasi yang dikirim. Ukuran default nilai metode POST adalah 8 Mb. Untuk mengubahnya dapat dilakukan pada pengaturan `post_max_size` di dalam file `php.ini`.

Untuk mengirim dengan metode POST, maka harus dibuat form terlebih dahulu. Misal dibuat sebuah html dengan nama `query_form.html` sebagai berikut:

Nama File	latihan/query_form.html
Keterangan	
<pre><!-- query_form.html --> <h2>Form Isian</h2> <form method="POST" action="query2.php"> NIM:
 <input type="TEXT" name="nim">
 Nama
 <input type="TEXT" name="nama" size="1">
 Nilai:
 <input type="TEXT" name="nilai" size="5">
 <input type="SUBMIT" value="Kirim"> </form></pre>	

Catatan: option pada action tag form adalah `query2.php`, yang berfungsi menerima data yang dikirimkan oleh browser. Source program misalnya seperti berikut ini.

Nama File	latihan/query2.php
Keterangan	
<pre><?php //query2.php echo 'Selamat Datang
'; echo 'Data yang dikirim dengan metode POST adalah sbb:
';</pre>	

```
echo 'NIM : '.$_POST['nim'].'<br/>';
echo 'Nama: '.$_POST['nama'].'<br/>';
echo 'Nilai : '.$_POST['nilai'].'<br/>';
?>
```

Bila url `http://localhost/query_form.html` dimasukan ke URL-bar, maka tampak pada layar sebagai berikut:

Form Isian

Nama:	<input type="text"/>
Jenis Kelamin:	<input type="checkbox"/>
Nilai:	<input type="text"/>
	<input type="button" value="Kirim"/>

Isikan setiap field pada form di atas dengan nilai-nilai, misal '07.11.1679' pada field NIM, 'Bima' pada Nama, dan '100' pada Nilai. Kemudian klik tombol 'Kirim', maka server akan menerima form tersebut lalu memproses `query2.php`, sehingga outputnya sebagai berikut:

Selamat Datang

Data yang dikirim dengan metode POST adalah sebagai berikut:

NIM : 07.11.1679

Nama: Bima

Nilai : 100

- Sama seperti metode GET, programmer harus berhati-hati dengan masalah security. Seseorang user dapat mengisi sembarang nilai dari suatu field sehingga dapat membahayakan website bahkan database.
- Disarankan melakukan tahap sanitize terlebih dahulu terhadap variabel `$_GET` maupun `$_POST` sebelum diproses.

Listing berikut merupakan contoh penggunaan form sebagai interface masukan data kemudian akan diproses menggunakan php. Misal dibuat sebuah file html dengan nama `form2.html` sebagai berikut:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Form Registrasi</title>
</head>
<body>
<form action="view.php" method="GET">
<table width="65%" align="center">
<tr><th>FORM REGISTRASI</th></tr>
<tr>
<td><table width="100%">
<tr>
<td width="25%">NIM</td>
<td width="5%" align="center"><b>:</b></td>
<td width="70%">
<input name="nim" type="text" id="nim"
size="50" maxlength="10"/>
</td>
</tr>
<tr>
<td width="25%">NAMA</td>
<td align="center"><b>:</b></td>
<td><input name="nama" type="text"
id="nama" size="50" /></td>
</tr>
<tr>
<td width="25%">JENIS KELAMIN</td>
<td align="center"><b>:</b></td>
<td>
<input type="radio" name="jk"
value="Laki-laki" />
Laki-laki &nbsp;&nbsp;&nbsp;&nbsp;
<input type="radio" name="jk"
value="Perempuan" />
Perempuan</td>
</tr>
<tr>
<td width="25%">ASAL</td>

```

```

<td align="center"><b>:</b></td>
<td>
<select name="asal">
<option value="-"></option>
<option
value="Yogyakarta">Yogyakarta</option>
<option
value="Jakarta">Jakarta</option>
<option value="Jawa Timur">Jawa
Timur</option>
</select>
</td>
</tr>
<tr>
<td width="25%">NO TELP</td>
<td align="center"><b>:</b></td>
<td><input name="telp" type="text"
id="telp" size="50" maxlength="13" /></td>
</tr>
<tr>
<td width="25%">ALAMAT EMAIL</td>
<td align="center"><b>:</b></td>
<td><input name="email" type="text"
id="email" size="50" /></td>
</tr>
<tr>
<td width="25%">PASSWORD</td>
<td align="center"><b>:</b></td>
<td><input name="password"
type="password" id="password" size="50" /></td>
</tr>
<tr>
<td width="25%" valign="top">HOBBY</td>
<td align="center"
valign="top"><b>:</b></td>
<td><table width="100%" border="0">
<tr>
<td width="48%">


```

```

<br />
<input type="checkbox" name="hb3"
value="Berenang" />Berenang
</td>
<td width="52%><input type="checkbox" name="hb4"
value="Membaca" />Membaca
<br />
<input type="checkbox" name="hb5" value="Jogging"
/>Jogging
<br />
<input type="checkbox" name="hb6" value="nGoding"
/>Battminton</td>
</tr>
</table></td>
</tr>
<tr>
    <td width="25%" valign="top">DESKRIPSI
DIRI</td>
    <td align="center"
valign="top"><b>:</b></td>
    <td>
        <textarea name="desk" cols="50"
rows="5"></textarea>
    </td>
    </tr>
</table></td>
</tr>
<tr>
    <td align="center">
        <input type="submit" value="OK"
/>&nbsp;&nbsp;&nbsp;
        <input type="reset" value="Batal" />
    </td>
</tr>
</table>
</form>
</body>
</html>

```

Action tag form adalah view.php, yang berfungsi menerima data yang dikirimkan oleh browser. Source program misalnya seperti berikut ini.

```

// menangkap value form yang disimpan dalam sebuah variabel
$nim = $_GET['nim'];
$nama = $_GET['nama'];
$asal = $_GET['asal'];
$telp = $_GET['telp'];
$email = $_GET['email'];
$password = $_GET['password'];
$jk = $_GET['jk'];
$desk = $_GET['desk'];
$hb1 = $_GET['hb1'];
$hb2 = $_GET['hb2'];
$hb3 = $_GET['hb3'];
$hb4 = $_GET['hb4'];
$hb5 = $_GET['hb5'];
$hb6 = $_GET['hb6'];

```

Menampilkan dari variabel diatas dalam browser

```

echo "NIM : $a<br />";
echo "Nama : $b<br />";
echo "Kota : $c<br />";
echo "No.Telp : $d<br />";
echo "Email : $e<br />";
echo "Password : $f<br />";
echo "Jenis Kelamin : $g<br />";
echo "Deskripsi Diri : $h'<br />";
echo "Hobby 1 : $i<br />";
echo "Hobby 2 : $j<br />";
echo "Hobby 3 : $k<br />";
echo "Hobby 4 : $l<br />";
echo "Hobby 5 : $m<br />";
echo "Hobby 6 : $n<br />";
?>

```

BAB 11

Session

III Pengertian Session

Session merupakan salah satu fasilitas yang ada pada PHP yang digunakan untuk menyimpan data sementara ke dalam variabel (bisa disebut variabel session) sehingga data tadi dapat diakses oleh client selama variabel session tersebut tidak dikosongkan atau dihilangkan. Nilai variabel di dalam session ini akan tersimpan di sisi server (web server). Hal ini berbeda dengan cookies yang sebenarnya disimpan di sisi client (browser). Jadi, session relatif lebih aman dibandingkan dengan cookies karena nilai session tidak mudah diakses oleh siapapun.

Session banyak digunakan untuk identifikasi user (seperti pada halaman login), menyimpan konfigurasi dan menyimpan sebuah nilai (seperti untuk counter). Tetapi, session tidak bisa digunakan untuk hal-hal lainnya seperti penggunaan variabel biasa.

III Mengawali Session

Jika menunjukkan bahwa suatu halaman menggunakan session maka awal halaman harus ada awal session yaitu dengan menggunakan `session_start()` yang menunjukkan halaman tersebut mengandung session. Jika suatu halaman tidak diawali dengan `session_start()` dan terdapat session di dalam halaman tersebut memproses variabel session, maka session tersebut tidak akan terbaca atau bernilai null atau kosong. Hal ini lah yang membuat memungkinkan muncul error pada halaman tersebut.

Contoh penggunaannya seperti berikut:

```
Code Split Design Title: Untitled Document
1 <?
2 session_start();
3 //ini halaman yang akan menggunakan session
4 |
```

11.3 Penggunaan Variabel Session

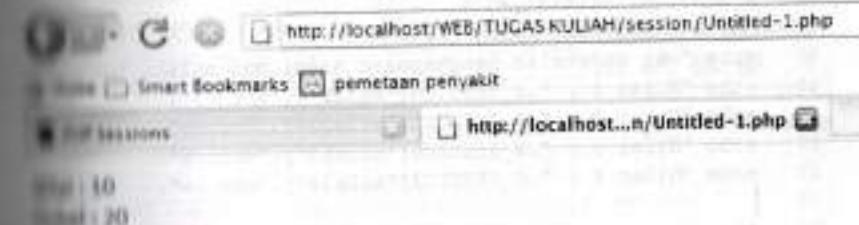
Prinsip dari penggunaan variabel session sama halnya dengan penggunaan variabel seperti biasanya. Variabel session dapat diisi sebuah nilai dan dilakukan operasi perhitungan, juga dapat diambil atau ditampilkan nilainya. Tetapi agar variabel session tersebut dapat digunakan maka harus pada pembahasan di atas, halaman tersebut harus dimulai duluan dengan `session_start()`.

Setelah session dimulai maka variabel session sudah dapat digunakan. Penggunaannya menggunakan format `$_SESSION['nm_var']`. `nm_var` tersebut merupakan nama dari variabel session yang dibuat dan diisi sesuai dengan nama yang akan digunakan. digunakan untuk identifikasi user seperti pada halaman yang memerlukan login sebaiknya nama dari variabel session tersebut lain dari yang lain agar lebih aman.

Untuk contoh penggunaan variabel session:

```
Code Split Design Title: Untitled Document
1 <?
2 session_start();
3 //ini halaman yang akan menggunakan session
4 //pengisian variabel session
5 $_SESSION['nilai']=10;
6
7 //operasi perhitungan pada variabel session
8 $dobel=$_SESSION['nilai']*2;
9
10 //menampilkan isi variabel session
11 echo "Nilai : ".$_SESSION['nilai']."<br />";
12 echo "Dobel : ".$dobel;
13
14 ?>
```

Jadi hasil dari script di atas:



11.4 Menghapus Session

Jika variabel session digunakan, variabel tersebut dapat dihapus. Ada dua cara untuk menghapus session, yang pertama yang banyak dilakukan yaitu hanya dengan memberikan atau mengganti nilai dari variabel session menjadi null atau kosong. Cara tersebut nantinya seperti `$_SESSION['nm_var']=""`.

Cara di atas dapat juga dengan menggunakan `unset($_SESSION['nm_var'])`. Cara tersebut untuk menghapus variabel session. Jika kita ingin menghapus semua variable session

yang mungkin ada banyak variabel session, maka dapat digunakan fungsi `session_destroy()`.

Berikut contoh penggunaannya:

```
Untitled-1.php Untitled-2.php Untitled-3.php
Code Split Design Title: Untitled-1.php

1 <?
2 session_start();
3 //Sebelum endus, penghapusan session
4 //sebelum penghapusan
5 echo "Sebelum dihapus <br />";
6 echo "Nilai 1 : ".$_SESSION['nilai1']."<br />";
7 $_SESSION['nilai2']=9;
8 $_SESSION['nilai3']=5;
9 $_SESSION['nilai4']=7;
10 echo "Nilai 2 : ".$_SESSION['nilai2']."<br />";
11 echo "Nilai 3 : ".$_SESSION['nilai3']."<br />";
12 echo "Nilai 4 : ".$_SESSION['nilai4']."<br />";
13
14 //penghapusan session nilai dan nilai4
15 $_SESSION['nilai3']=""; //Setelah penghapusan nilai dan nilai3
16 unset($_SESSION['nilai1']);
17 echo "<br />Setelah penghapusan nilai dan nilai3 <br />";
18 echo "Nilai 1 : ".$_SESSION['nilai1']."<br />";
19 echo "Nilai 2 : ".$_SESSION['nilai2']."<br />";
20 echo "Nilai 3 : ".$_SESSION['nilai3']."<br />";
21 echo "Nilai 4 : ".$_SESSION['nilai4']."<br />";
22
23 ?>
```

<http://localhost.../Untitled-2.php> PHP

Sebelum dihapus

Nilai 1 : 10

Nilai 2 : 9

Nilai 3 : 8

Nilai 4 : 7

Setelah penghapusan nilai dan nilai3

Nilai 1 :

Nilai 2 : 9

Nilai 3 :

Nilai 4 : 7

Penggunaan session_destroy();

```
Untitled-2.php Untitled-3.php
Code Split Design Title: Untitled-3.php

session_start();
session_destroy();
//Sebelum endus, penghapusan session menggunakan session_destroy()
<br />Setelah menggunakan session_destroy() <br />
Nilai 1 : ".$_SESSION['nilai1']."<br />";
Nilai 2 : ".$_SESSION['nilai2']."<br />";
Nilai 3 : ".$_SESSION['nilai3']."<br />";
Nilai 4 : ".$_SESSION['nilai4']."<br />"
```

Hasil dari script

 http://localhost.../Untitled-3.php

Setelah menggunakan session_destroy()

Nilai 1 :

Nilai 2 :

Nilai 3 :

Nilai 4 :

10.5 Contoh Penggunaan Session untuk Login

Pertama, buat dulu form loginnya dengan html dan form actionnya proses.php. Tampilan form loginnya kira-kira seperti ini:

User ID:	<input type="text"/>
Password:	<input type="password"/>
	<input type="button" value="Login"/> <input type="button" value="Reset"/>

Setelah itu buat halaman proses.php yang isinya seperti ini:

```
session_start();
require_once '../config/db.php';
ini_set('magic_quotes_gpc','');
$pengguna=stripslashes($_POST['uid']);
$pass=stripslashes($_POST['password']);

$pengguna=mysql_real_escape_string($pengguna);
$pass=mysql_real_escape_string($pass);
$SQL="mysql_query("select id_user,username from user where
username='$pengguna' and pass='".md5($pass)."'")or die("Cek kembali
username dan password");
if(mysql_num_rows($SQL)>0){
    $cek=mysql_fetch_array($SQL);
    if($cek['adm_user']=='cek[id_user]');
    if($cek['adm_nama']=='cek[username]');
    mysql_close($koneksi);
    header("Location: ./index.php");
}
else{
    <script>window.alert("Pasangan User ID dan password tidak
cocok!</script>
    window.location='./login.php'</script>";
}
```

Buat halaman yang hanya boleh diakses oleh pengguna, di awalnya tuliskan script seperti pada gambar berikut:

```
session_start();
if(!isset($_SESSION['adm_user'])){
    header("Location: ./login.php");
}
```

BAB 12

Pengantar MySQL

MySQL dibangun oleh sebuah perusahaan Swedia bernama MySQL AB yang pada saat itu bernama TcX DataKonsult AB sekitar tahun 1994-1995, namun sebenarnya teknologi pokok kodenya sudah ada sejak 1979. Awalnya TcX membuat MySQL dengan tujuan mengembangkan aplikasi web untuk klien. TcX merupakan sebuah perusahaan pengembang software dan konsultan database. Pada akhirnya MySQL sudah diakuisisi oleh Oracle Corp.

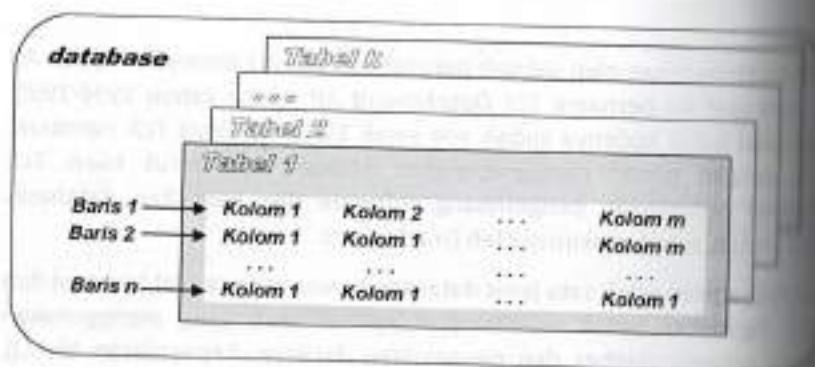
MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan MySQL sebagai sumber dan pengelolaan datanya. Kepopuleran MySQL ini tentu saja karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengelola database-nya sehingga mudah untuk digunakan, kinerja query MySQL juga cukup mencukupi untuk kebutuhan database perusahaan-perusahaan besar maupun kecil. MySQL juga bersifat *open source* dan *free* (Anda tidak perlu membayar untuk menggunakan MySQL) pada berbagai platform (kecuali Windows, yang bersifat *shareware*). MySQL didistribusikan dengan lisensi *open source* GPL (General Public License) mulai versi 3.23, pada bulan Mei 2000. Software MySQL bisa diunduh di <http://www.mysql.org> atau www.mysql.com.

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP merupakan sebuah pasangan software pengembangan aplikasi web yang baik. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web sedangkan pengembangannya menggunakan bahasa pemrograman script PHP.

1. Database, Tabel, Baris dan Kolom

MySQL termasuk RDBMS (*Relational Database Management System*). Itulah sebabnya istilah tabel, baris, dan kolom digunakan pada MySQL. Pada dasarnya sebuah database mengandung satu atau sejumlah tabel. Tabel tersebut memiliki sejumlah kolom dan baris, dimana setiap kolom berisi sekumpulan data.

pulan data yang memiliki tipe yang sejenis, dan baris merupakan sekumpulan data yang saling berkaitan dan membentuk informasi. Biasanya juga disebut sebagai *field* dan informasi yang tersimpan dalam setiap baris disebut *record*.



12.2 Menggunakan MySQL

12.2.1 Struktur Direktori MySQL

Software MySQL secara default akan diletakkan pada direktori C:\MySQL jika diinstall pada sistem operasi Windows. Direktori ini bisa saja tidak sesuai dengan keinginan pada saat instalasi. Apabila instalasi dilakukan dengan menggunakan software PHPTriad (paket software yang menggabungkan tiga aplikasi: Apache Web Server, PHP, dan MySQL), maka software MySQL terletak di dalam direktori C:\Apache\MySQL.

Struktur direktori MySQL:

MySQL

- bench
- bin
- data
- docs
- examples
- include
- lib
- scripts
- share

Sub-direktori yang paling penting dalam struktur direktori MySQL adalah *bin* dan *data*. Sub-direktori *bin* merupakan direktori yang menyimpan semua program database MySQL, sedangkan sub-direktori *data* digunakan untuk menyimpan data dan file-file yang dibutuhkan oleh MySQL untuk menyimpan database. Setiap database MySQL dibuatkan sebagai sub-direktori di dalam sub-direktori *data* ini.

12.2.2 Tipe Data pada MySQL

Ada delapan tabel tipe data yang dapat digunakan untuk field-field tabel pada MySQL.

Tipe Data	Ukuran	Keterangan
tinyint	1 byte	Nilai integer yg sangat kecil
smallint	2 bytes	Nilai integer yang kecil
mediumint	3 bytes	Integer dengan nilai medium
int	4 bytes	Integer dengan nilai standar
bigint	8 bytes	Integer dengan nilai besar
decimal	4 bytes	Bilangan desimal dengan

		single-precision			
DOUBLE	8 bytes	Bilangan desimal dengan double-precision		3 bytes	"-832.59:59" sampai "838-59:59"
DECIMAL(M,D)	M bytes (D+2, if M < D)	Bilangan float (desimal) yang dinyatakan sebagai string		8 bytes	"1000-01-01 00:00:00" sampai "9999-12-31 23:59:59"
CHAR(M)	M bytes, 1 <= M <= 255	String karakter dengan panjang yang tetap		4 bytes	Range: 19700101000000 (suatu nilai tanggal pada tahun 2037)
VARCHAR(M)	L+1 bytes, L <= M and 1 <= M <= 255	String karakter dengan panjang yang tidak tetap		1 byte	1901 sampai 2155
TINYBLOB	L+1 bytes, L < 2^8	BLOB (Binary Large Object) yang sangat kecil			Nilai kosong (hampa)
BLOB	L+1 bytes, L < 2^16	BLOB berukuran kecil			
MEDIUMBLOB	L+1 bytes, L < 2^24	BLOB berukuran sedang			
LONGBLOB	L+1 bytes, L < 2^32	BLOB berukuran besar			
TINYTEXT	L+1 bytes, L < 2^8	String teks yang sangat kecil			
TEXT	L+1 bytes, L < 2^16	String teks berukuran kecil			
MEDIUMTEXT	L+1 bytes, L < 2^24	String teks berukuran medium			
LONGTEXT	L+1 bytes, L < 2^32	String teks berukuran besar			
ENUM('v1','v2',...)	1 or 2 bytes, (65535 values max)	Enumerasi, kolom dapat diajukan dengan 1 member enum			
SET('val1','val2',...)	1, 2, 3, 4 or 8 bytes, (64 max)	Himpunan, kolom dapat diajukan dengan beberapa nilai anggota himpunan			
DATE	3 bytes	"1000-01-01" sampai "9999-12-31"			

4.3 Mengaktifkan Database MySQL

MySQL berjalan pada mode DOS sehingga untuk menggunakan MySQL terlebih dahulu masuk ke mode DOS. Kita mengasumsikan program MySQL berada pada direktori C:\Apache\MySQL, dan untuk kesederhana kita masuk ke direktori MySQL dengan mengetikkan mysql\bin pada DOS prompt. Setelah berada pada C:\Apache\MySQL\bin, langkah selanjutnya adalah mengaktifkan MySQL (MySQL-daemon). Jika server MySQL telah aktif, diteruskan memanggil program MySQL yang berjalan di sisi client. Untuk lebih jelas ikuti langkah-langkah berikut:

Masuk ke mode DOS

cd\apache\mysql\bin (enter)

mysql (enter)

mysql (enter)

Jika langkah-langkah di atas dilakukan dengan benar, kita akan dibawa ke lingkungan MySQL client, yang ditandai dengan munculnya tampilan seperti berikut:

WINDOWS>cd\apache\mysql\bin

Apache\MySQL\bin>mysql

Welcome to the MySQL monitor. Commands end with ;

```
Your MySQL connection id is 2 to server version:  
3.23.42  
Type 'help;' or '\h' for help. Type '\c' to clear  
the buffer.  
mysql>
```

Pada layar monitor akan ditampilkan versi MySQL yang digunakan (pada contoh ini versi yang digunakan adalah versi 3.23.42), dan adanya *prompt* mysql> menunjukkan bahwa program MySQL telah siap untuk menerima masukan berupa perintah-perintah SQL. Kita dapat mengetikkan perintah-perintah SQL pada *prompt* tersebut. Dalam penulisan perintah SQL, satu baris perintah selalu diakhiri dengan tanda titik koma (;). Sedangkan untuk keluar dari lingkungan MySQL dapat dilakukan dengan mengetikkan salah satu perintah: *exit*, *quit*, atau *\q*.

12.2.4 Operasi-operasi MySQL

• Membuat Database

Sebelum melakukan proses pembuatan tabel dan manipulasi data, database harus diciptakan terlebih dahulu. Dalam pembuatan database pemberian nama database harus unik. Ini berarti tidak boleh menggunakan nama database yang sudah ada. Untuk mengetahui daftar database yang terdapat pada MySQL, dapat digunakan perintah berikut:

```
SHOW DATABASES;
```

Sedangkan perintah untuk membuat database adalah sebagai berikut:

```
CREATE DATABASE nama_database;
```

Agar dapat melakukan operasi-operasi terhadap suatu database, harus dilakukan koneksi ke database tersebut terlebih dahulu. Koneksi ke database dapat dilakukan dengan menggunakan perintah *use*, sebagai berikut:

```
USE nama_database;
```

Jika koneksi ke database berhasil akan muncul tanggapan dari MySQL, sebagai berikut:

```
Database changed
```

Menghapus database dapat dilakukan dengan perintah:

```
DROP nama_database;
```

• Membuat Tabel

Suatu database ditempati oleh tabel-tabel yang merupakan tempat untuk menyimpan data. Membuat tabel di dalam database MySQL dapat dilakukan dengan mengetikkan perintah SQL berikut:

```
CREATE TABLE nama_tabel (
```

```
nama_field1 tipe_data1(ukuran/nilai) atribut,  
nama_field2 tipe_data2(ukuran/nilai) atribut,  
...);
```

• Mengubah Struktur Tabel

Menambah field:

```
ALTER TABLE 'nama_tabel' ADD 'field'  
tipe_data[(ukuran/nilai) atribut];  
ALTER TABLE 'anama_tabel' ADD 'field' tipe_data  
FIRST;  
ALTER TABLE 'nama_tabel' ADD 'field' tipe_data  
AFTER field_posisi;
```

Mengubah nama field:

```
ALTER TABLE 'nama_tabel' CHANGE 'field_lama'  
'field_baru' tipe_data[(ukuran/nilai) atribut];
```

Menghapus field:

```
ALTER TABLE 'nama_tabel' DROP 'field';
```

Menentukan primary key:

```
ALTER TABLE 'nama_tabel' ADD PRIMARY KEY
    (field_kunci);
```

Mengubah primary key:

```
ALTER TABLE 'nama_tabel' DROP PRIMARY KEY, ADD
PRIMARY KEY (field_kunci);
```

Menghapus primary key:

```
ALTER TABLE 'nama_tabel' DROP PRIMARY KEY;
```

Memasukkan Data

Memasukkan data pada suatu tabel yang telah dibuat dilakukan dengan menggunakan pernyataan *insert*, dengan sintaks penulisannya sebagai berikut:

```
INSERT INTO nama_tabel (field1, field2,...) VALUES
(nilai_field1,nilai_field2,...);
```

```
INSERT INTO nama_tabel VALUES
(nilai_field1,nilai_field2,...);
```

Menampilkan Data

Perintah yang digunakan untuk menampilkan data dari tabel adalah *select*. Sintaks penulisannya sebagai berikut:

```
SELECT field1,field2,... FROM nama_tabel;
```

Perintah di atas akan menampilkan semua data yang terdapat pada field yang ditentukan. Untuk melihat semua data yang ada pada suatu tabel, sintaks penulisannya adalah:

```
SELECT * FROM nama_tabel;
```

Untuk menyaring hasil seleksi data digunakan kata kunci *where* sehingga record yang ditampilkan hanyalah record yang sesuai dengan kriteria yang diinginkan. Sintaks penulisannya sebagai berikut:

```
SELECT field1,field2,... FROM nama_tabel WHERE
```

kriteria;

Mengurutkan Data

Hasil query dapat diurutkan sesuai kebutuhan dengan menggunakan kata kunci *order by*. Sintak penulisannya sebagai berikut:

```
SELECT field1,field2,... FROM nama_tabel ORDER BY
kriteria;
```

Untuk mengurutkan data dengan urutan terbalik (besar ke kecil) dapat digunakan kunci tambahan *desc*, dengan sintaks penulisan sebagai berikut:

```
SELECT field1,field2,... FROM nama_tabel ORDER BY
kriteria DESC;
```

Mengubah Data

Pengubahan data pada tabel berfungsi untuk memodifikasi nilai kolom (*field*) dari suatu record. Perintah SQL yang digunakan untuk mengubah data adalah sebagai berikut:

```
UPDATE nama_tabel SET nama_field1=nilai_baru1,
nama_field2=nilai_baru2, ... WHERE kriteria;
```

Untuk memodifikasi nilai suatu kolom secara keseluruhan, digunakan perintah *update* tanpa menentukan kriterianya:

```
UPDATE nama_tabel SET nama_field1=nilai_baru1,
nama_field2=nilai_baru2, ...;
```

• Menghapus Data

Penghapusan data pada tabel berfungsi untuk menghapus suatu record dengan kriteria tertentu. Perintah SQL untuk menghapus data pada tabel adalah `delete`, dengan sintaks penulisan sebagai berikut:

```
DELETE FROM nama_tabel WHERE kriteria;
```

Untuk menghapus seluruh record pada suatu tabel, digunakan perintah `delete` tanpa menentukan kriterianya:

```
DELETE FROM nama_tabel;
```

• Backup Data

```
mysqldump -u [username] [nama-database] > [nama-
file.sql];
mysqldump -u root test > [test.sql];
```

• Restore Data

```
mysql -u [username] [nama-database] < [nama-
file.sql];
mysql -u root test2 < test.sql;
```

• Membuat User dan Hak Aksesnya

```
GRANT ALL PRIVILEGES ON *.* TO 'user-name'
IDENTIFIED BY 'user-password';
```

• Menghapus Hak Akses User

```
REVOKE ON *.* FROM 'user-name';
```

BAB 13

Koneksi PHP dan MySQL

PHP memiliki fungsi-fungsi yang digunakan untuk mengakses database MySQL. Fungsi-fungsi tersebut berguna untuk melakukan koneksi dan manipulasi database MySQL melalui program PHP.

`mysql_connect`

Fungsi: Membuka koneksi ke server database MySQL
 Penulisan: resource `mysql_connect` ([string hostname [:port] [:path/to/socket] [, string username [, string password]]])
 Contoh: `mysql_connect("localhost","root","")`

`mysql_close`

Fungsi: Memutuskan koneksi dengan server database MySQL
 Penulisan: Bool `mysql_close` ([resource link_identifier])
 Contoh: `$koneksi=mysql_connect("localhost","root","");
 mysql_close($koneksi);`

`mysql_select_db`

Fungsi: Memilih database MySQL yang akan di akses
 Penulisan: bool `mysql_select_db` (string database_name [, resource link_identifier])
 Contoh: `<?php
 $link =
 mysql_connect("localhost","root","");
 mysql_select_db("namedb");
 ?>`

```

        or die("Gagal Koneksi ke Server
Database!");
mysql_select_db("test",$link)
        or die("Gagal Memilih Database!");
?>

```

mysql_query

Fungsi: Mengeksekusi perintah query

Penulisan: resource **mysql_query** (string query [, resource link_identifier])

Contoh:

```
<?php
mysql_query (" SELECT * FROM buku");
?>
```

mysql_num_rows

Fungsi: Menampilkan jumlah baris yang dihasilkan oleh perintah *mysql_query* yang menggunakan SELECT.

Penulisan: int **mysql_num_rows** (resource result)

Contoh:

```
<?php
$link = mysql_connect("localhost",
"username", "password");
mysql_select_db("database", $link);
$result = mysql_query("SELECT * FROM
buku", $link);
$num_rows = mysql_num_rows($result);
print("$num_rows Baris");
?>
```

mysql_num_fields

Fungsi: Menampilkan jumlah field yang dihasilkan oleh perintah *mysql_query* yang menggunakan SELECT

Penulisan: int **mysql_num_fields** (resource result)

mysql_affected_rows

Fungsi: Menampilkan jumlah baris yang terpengaruh oleh operasi SQL sebelumnya (INSERT, DELETE, dan UPDATE).

Penulisan: int **mysql_affected_rows** (resource result)

resource_result menyatakan variabel yang digunakan pada pemanggilan fungsi *mysql_connect*.

mysql_fetch_row

Fungsi: Menghasilkan array yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil. Fungsi ini bersifat membaca baris berikutnya dalam suatu himpunan hasil.

Penulisan: array **mysql_fetch_row** (resource result)

mysql_fetch_array

Fungsi: Sama seperti *mysql_fetch_row*. Hanya saja setiap kolom akan disimpan dua kali pada array hasil. Yang pertama memiliki indeks angka yang dimulai dari nol, dan yang kedua memiliki indeks berdasarkan nama *field*.

Penulisan: array **mysql_fetch_array** (resource result [, int result_type])

mysql_fetch_field

Fungsi: Memperoleh informasi suatu kolom.

Penulisan: object **mysql_fetch_field** (resource result [, int field_offset])

field_offset menyatakan *nomor_kolom* menyatakan nomor kolom yang ingin didapatkan. Penomoran ini dimulai dari nol dan sifatnya opsional. Jika argumen ini tidak disebutkan, maka kolom berikutnya yang akan didapatkan.

Contoh:

```
<?php
$link =
mysql_connect("localhost","root","");
mysql_select_db("workshop",$link);

$result = mysql_query ("select * from
staff");
$i = 0;
while ($i < mysql_num_fields ($result))
{
    echo "Information for column
$i:<BR>\n";
    $meta = mysql_fetch_field ($result);
    if (!$meta)
    {
        print("No information
available<BR>\n");
    }
    print("<PRE>
blob:           $meta->blob
max_length:     $meta->max_length
multiple_key:   $meta->multiple_key
name:           $meta->name
not_null:       $meta->not_null
numeric:        $meta->numeric
primary_key:   $meta->primary_key
table:          $meta->table
type:           $meta->type
unique_key:     $meta->unique_key
unsigned:       $meta->unsigned
zerofill:       $meta->zerofill
</PRE>");
```

```
$i++;
}
mysql_free_result ($result);
?>
```

mysql_data_seek

Fungsi: Memindahkan pointer pada suatu himpunan hasil supaya menunjuk ke baris tertentu.

Penulisan: bool **mysql_data_seek** (resource result_identifier, int row_number)

mysql_create_db

Fungsi: Menciptakan database baru.

Penulisan: int **mysql_create_db** (string database_name [, resource link_identifier])

mysql_drop_db

Fungsi: Menghapus database

Penulisan: Bool **mysql_drop_db** (string database_name [, resource link_identifier])

mysql_list_dbs

Fungsi: Menghasilkan daftar database yang ada di server

Penulisan: resource **mysql_list_dbs** ([resource link_identifier])

Contoh:

```
<?php
$link =
mysql_connect("localhost","root","");
$db_list = mysql_list_dbs($link);
while ($row =
mysql_fetch_object($db_list))
```

```

    print($row->Database . "<br>\n");
}
?>

```

mysql_list_tables

Fungsi: Memperoleh daftar nama tabel dalam suatu database MySQL

Penulisan: resource **mysql_list_tables** (string database [, resource link_identifier])

Contoh:

```

<?php
$koneksi =
mysql_connect("localhost","root","");
$DB = "database";
mysql_select_db($DB);
$tables=mysql_list_tables($DB);
while
(list($bla)=mysql_fetch_array($tables))
{
    print($bla." are the tablenames<br>");
}
?>

```

mysql_list_fields

Fungsi: Memperoleh daftar nama kolom.

Penulisan: resource **mysql_list_fields** (string database_name, string table_name [, resource link_identifier])

Contoh:

```

<?php
$koneksi =
mysql_connect("localhost","root","");
mysql_select_db("workshop",$koneksi);
$fields=mysql_list_fields
("workshop",("staff",$koneksi));
$columns = mysql_num_fields($fields);
for ($i = 0; $i < $columns; $i++){ echo
mysql_field_name($fields, $i)
. "<br>\n"; }?>

```

BAB 14

Pembuatan Web Sederhana

Pembahasan kali ini akan membahas pembuatan web sederhana yang menggabungkan antara elemen form HTML, database MySQL dan PHP. Berikut adalah langkah-langkahnya:

1. Membuat database dan tabel-tabel yang diperlukan, yaitu database ponselkita dan satu tabel yaitu tabel brand dengan 2 kolom (IDBrand, Brand).
2. Membuat rancangan form HTML.
3. Membuat perintah PHP untuk melakukan koneksi dan pengolahannya.

14.1 Pembuatan Database ponselkita

Ada 2 cara untuk membuat database di MySQL, yaitu dengan menggunakan aplikasi phpMyAdmin yang berbasis web atau menggunakan perintah SQL.

1. Membuat database menggunakan phpMyAdmin
 - Aktifkan browser, kemudian ketik URL berikut pada address bar yang ada di browser: <http://localhost/phpMyAdmin/>, untuk mengaktifkan phpMyAdmin.
 - Pada halaman utama phpMyAdmin, buat database dengan nama "ponselkita". Isikan nama database yang akan dibuat pada bagian "Create new database" kemudian klik tombol "Create".