

# **UNIVERSITAS GUNADARMA**



## **PENULISAN ILMIAH HALAMAN JUDUL**

### **APLIKASI WEB DENGAN PHP**

Nama : Mark Ade Syukur  
NPM : 31196574  
Jurusan : Manajemen Informatika  
Pembimbing : Enny Susana, SKom, MM

### **PENULISAN PENELITIAN/KERJA PRAKTEK**

**Diajukan Untuk Melengkapi Syarat  
Mencapai Jenjang D III/Setara Sarjana Muda  
Universitas Gunadarma**

**LEMBAR PENGESAHAN**

Judul PI : Aplikasi Web dengan PHP  
Nama : Mark Ade Syukur  
N P M : 31196574  
NIRM : 963137756630607  
Tanggal Sidang : 4 Agustus 1999  
Tanggal Lulus : 4 Agustus 1999

Menyetujui,

Dosen Pembimbing

Koordinator PI

(Enny Susana, SKom, MM)

(Dra. Henny Widowati, MM)

Ketua Jurusan  
Manajemen Informatika

(Marliza Ganefi, SKom, MM)

Untuk bangsaku yang sedang prihatin

## **ABSTRAKSI**

Mark Ade Syukur. 31196574.

APLIKASI WEB DENGAN PHP.

Penulisan Ilmiah. Fakultas Ilmu Komputer. 1999.

Kata Kunci: aplikasi, internet, web, PHP.

( ix + 55 + Lampiran).

Pada Penulisan Ilmiah ini penulis membahas penggunaan PHP sebagai bahasa CGI yang mendukung web sebagai media informasi dinamis, yaitu sebagai sarana untuk mengambil, mengolah, dan menyediakan informasi dengan efektif.

Pembahasan dimulai dengan memperkenalkan PHP, bagaimana bentuk dan susunan bahasanya, dan menerangkan fasilitas apa saja yang disediakan oleh PHP. Pembahasan mengenai PHP sengaja dipisahkan menjadi bagian tersendiri, yang semata-mata untuk memperjelas materi.

Untuk memperjelas pembahasan mengenai PHP, pada bagian berikutnya penulis menjelaskan penerapan PHP melalui sebuah contoh aplikasi web yang pada intinya memanfaatkan PHP.

Semua pembahasan dan penerapan berlandaskan pada teori TCP/IP, HTTP, HTML, Konsep Client-Server, dan beberapa teknologi Internet pada umumnya dan khususnya pada web. Juga disertakan landasan teori mengenai konsep *open source*, dimanfaatkan pada aplikasi yang penulis gunakan sebagai penerapan PHP, dan beberapa teori yang dianggap perlu.

Daftar Pustaka ( 1996 - 1999 )

## **PENGANTAR**

Puji dan syukur penulis limpahkan ke hadirat Tuhan Yang Maha Kuasa, yang dengan karunia, berkat dan kemurahan hatiNya telah memperkenankan penulis menyelesaikan Penulisan Ilmiah ini.

Demikian pula penulis menyampaikan ucapan terimakasih kepada:

1. Ibu Prof. E. S. Margianti, SE, MM, selaku Rektor Universitas Gunadarma.
2. Ibu Marliza Ganefi, SKom, MM, selaku Ketua Jurusan Manajemen Informatika Universitas Gunadarma.
3. Ibu Enny Susana, SKom, MM, selaku Dosen Pembimbing yang telah memberikan bimbingan dan masukan sehingga penulisan ilmiah ini dapat diselesaikan dengan baik.
4. Bapak I Made Wiryana, selaku rekan yang secara rendah hati dan penuh kesabaran bersedia memberikan bimbingan serta ide-idenya kepada penulis.
5. Semua pihak yang tidak bisa disebutkan satu per satu, yang telah membantu menyelesaikan karya ini, termasuk di antaranya kawan-kawan yang telah mendukung, memberi masukan, dan meminjamkan berbagai peralatan kepada penulis.

Sebagai pribadi dan sebagai mahasiswa, tentu masih banyak kesalahan dan kekurangan yang ada pada karya ini. Untuk itu penulis mohon maaf, dan sangat mengharapkan berbagai bentuk masukan, baik untuk karya ini maupun untuk pribadi penulis sendiri.

Semoga karya ini dapat bermanfaat bagi siapa saja yang tertarik untuk mempelajari teknologi pengembangan aplikasi web, dan tentunya agar dapat dikembangkan lebih jauh di masa mendatang.

Jakarta, Juni 1999

Penulis

**DAFTAR ISI**

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN .....	ii
ABSTRAKSI.....	iv
PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	ix
BAB I PENDAHULUAN .....	1
I.1 LATAR BELAKANG MASALAH .....	1
I.2 BATASAN MASALAH .....	2
I.3 TUJUAN PENULISAN .....	2
I.4 METODE PENULISAN.....	3
I.5 SISTEMATIKA PENULISAN .....	3
BAB II LANDASAN TEORI.....	4
II.1 TCP/IP .....	4
II.2 World Wide Web .....	5
II.3 Hypertext Transfer Protocol (HTTP) .....	7
II.4 Hypertext Markup Language (HTML).....	9
II.5 BROWSER DAN SERVER WEB.....	10
II.6 INTERNET SEBAGAI SISTEM CLIENT/SERVER .....	11
II.7 WEB DENGAN INFORMASI YANG DINAMIS .....	15
II.8 STRUCTURED QUERY LANGUAGE .....	17
II.9 KONSEP OPEN SOURCE.....	18
II.10 HTML Cookie.....	21
II.11 PEMBAHASAN PHP.....	24
II.11.1 Tentang PHP.....	24

II.11.2	Apa itu PHP .....	25
II.11.3	Web Dinamis .....	25
II.11.4	Pendekatan Cara PHP .....	27
II.11.5	Dasar-dasar PHP .....	28
II.11.6	Variabel pada PHP .....	30
II.11.7	Cookie pada PHP .....	33
II.11.8	Fungsi pada PHP .....	34
BAB III MASALAH DAN PENERAPAN PHP .....		35
III.1	PERMASALAHAN .....	35
III.2	PENERAPAN PHP .....	36
III.2.1	Peralatan yang Digunakan .....	36
III.2.2	Struktur Database .....	37
III.2.3	Alur Program .....	41
III.2.4	Penjelasan Program .....	44
BAB IV PENUTUP .....		52
IV.1	KESIMPULAN .....	52
IV.2	SARAN .....	52
DAFTAR PUSTAKA .....		54
REFERENSI INTERNET .....		55
Lampiran Kode Program .....		56

**DAFTAR GAMBAR**

<a href="#">Gambar 2.1 - Konsep dasar browser dan server web</a> .....	6
<a href="#">Gambar 2.2 - Konsep Sistem Mainframe</a> .....	12
<a href="#">Gambar 2.3 - CGI secara sederhana</a> .....	16
<a href="#">Gambar 2.4 - Program CGI menjadi penghubung ke sistem database</a> .....	17
<a href="#">Gambar 3.1 - Common Gateway Interface</a> .....	26
<a href="#">Gambar 3.2 - Prinsip kerja PHP dan Apache</a> .....	28
<a href="#">Gambar 3.1 – Tampilan utama “PEMILO 1999”</a> .....	35
<a href="#">Gambar 3.2 - Hubungan antar komponen aplikasi web</a> .....	37
<a href="#">Gambar 3.3 - Hubungan tabel pilihan dengan tabel logo</a> .....	39
<a href="#">Gambar 3.4 - Diagram alur Menu Utama</a> .....	41
<a href="#">Gambar 3.5 - Diagram alur submenu Profil</a> .....	42
<a href="#">Gambar 3.6 - Diagram alur submenu Coblos</a> .....	42
<a href="#">Gambar 3.7 - Diagram alur submenu Tabel</a> .....	43
<a href="#">Gambar 3.8 - Diagram alur submenu 5 Besar</a> .....	43
<a href="#">Gambar 3.9 - Diagram alur submenu Buku Tamu</a> .....	44
<a href="#">Gambar 3.10 - Tampilan form untuk verifikasi</a> .....	45
<a href="#">Gambar 3.11 - Tampilan “kartu suara” PEMILO 1999</a> .....	47
<a href="#">Gambar 3.12 - Tampilan submenu Tabel</a> .....	48
<a href="#">Gambar 3.13 - Tampilan grafis 5 logo terbaik</a> .....	49
<a href="#">Gambar 3.14 - Tampilan pilihan pada submenu Buku Tamu</a> .....	49
<a href="#">Gambar 3.15 - Tampilan “Mengisi Buku Tamu”</a> .....	50
<a href="#">Gambar 3.16 - Tampilan isi Buku Tamu</a> .....	50



**DAFTAR TABEL**

<a href="#">Tabel 3.1 Tabel Pilihan</a> .....	37
<a href="#">Tabel 3.2 Tabel Logo</a> .....	38
<a href="#">Tabel 3.3 Tabel Bukutamu</a> .....	39
<a href="#">Tabel 3.4 Tabel Pemilih</a> .....	40

## **BAB I PENDAHULUAN**

### **I.1 LATAR BELAKANG MASALAH**

Informasi. Adalah salah satu kata kunci pada jaman ini. Semua kegiatan kita memerlukan informasi, dan bisa juga dikatakan bahwa semua kegiatan kita dituntut untuk menghasilkan informasi. Untuk mendapatkan dan menghasilkan informasi, komputer dan teknologinya adalah salah satu alat bantu yang paling tepat. Penggunaan komputer pada berbagai bidang, kalangan dan usia selalu kita jumpai sekarang ini. Tuntutan kebutuhan akan informasi dan penggunaan komputer yang semakin banyak mendorong terbentuknya sebuah jaringan komputer yang mampu melayani berbagai kebutuhan tertentu. Dengan adanya jaringan komputer, pengelolaan informasi dapat berlangsung lebih baik lagi. Berkembangnya teknologi dan kebutuhan akan informasi menyebabkan bertambah kompleksnya informasi yang harus dan yang bisa diolah, sehingga kebutuhan penggunaan beberapa jaringan komputer bersama-sama semakin diperlukan. Penggunaan jaringan secara bersama-sama ini bertumbuh membentuk jaringan komputer yang amat besar yang tersebar di seluruh bagian di muka bumi ini.

Jaringan komputer seperti ini kita kenal dengan nama Internet. Internet bisa diakses dan dimanfaatkan untuk berbagai keperluan, oleh siapa saja, di mana saja, kapan pun kita mau menggunakannya. Berbagai macam teknologi Internet bisa digunakan, salah satunya adalah *World Wide Web* (atau selanjutnya disebut "web" saja) yang mampu menyediakan informasi dalam bentuk teks, gambar, suara, maupun gambar bergerak. Dengan kemampuan seperti ini, web menjadi sangat terkenal dan perkembangannya sangatlah pesat.

Dengan terus melajunya roda perkembangan teknologi, web berkembang menjadi alat bantu yang tidak hanya mampu menyediakan informasi, namun juga mampu untuk mengolah informasi. Proses pengolahan informasi dengan

memanfaatkan teknologi web menyebabkan web menjadi media informasi yang dinamis. Hal ini membutuhkan penggunaan sarana teknis yang menentukan. Sarana tersebut banyak bergantung pada penggunaan perangkat lunak yang kuat, aman, terpercaya, dan cepat. Perangkat lunak penentu yang dibutuhkan antara lain adalah sebuah *server-side scripting language*, atau juga disebut sebagai CGI (Common Gateway Interface), yang banyak tersedia di pasaran seperti: Perl, ASP (Active Server Pages), Cold Fusion, Python, dan PHP.

Pada kesempatan ini penulis hendak membahas mengenai penggunaan bahasa PHP sebagai salah satu perangkat lunak yang turut menentukan bisa digunakannya media web sebagai media pengolah informasi yang dinamis.

Bagaimana bentuk, penggunaan dan pemanfaatan PHP sebagai bahasa yang mampu mendukung web sebagai pengolah media informasi yang dinamis?

## I.2 BATASAN MASALAH

Dalam membahas PHP, penulis hanya akan menerangkan hal berikut:

- apa itu PHP
- bentuk, *syntax* pemrograman dan *grammar* dari PHP
- fasilitas PHP

Penulis tidak bermaksud apapun untuk membandingkan PHP dengan bahasa CGI lainnya.

Untuk menerapkan PHP, penulis membuat sebuah aplikasi web yang semata hanya merupakan contoh pemanfaatan PHP. Aplikasi web ini membutuhkan sistem yang sifatnya *open source*, dan dalam penjelasannya penulis tidak bermaksud untuk membandingkannya dengan produk manapun.

## I.3 TUJUAN PENULISAN

Tujuan Penulisan Ilmiah ini:

- memperkenalkan dan menjelaskan PHP sebagai sebuah bahasa CGI
- menerapkan metode pengaksesan server database serta pengaksesan grafik dinamis menggunakan PHP
- menyumbangkan gagasan konsep *open source* pada sebuah aplikasi pemilihan

#### **I.4 METODE PENULISAN**

Semua materi yang dibahas pada Bab II dan Bab III berdasarkan pada berbagai teori yang diolah dari literatur yang tercantum pada Daftar Pustaka. Sedangkan aplikasi web yang disertakan pada Bab IV adalah gagasan penulis semata, tanpa maksud mengambil dan/atau meniru hasil karya pihak manapun.

Perancangan, penyusunan, dan pembuatan Penulisan Ilmiah beserta aplikasi web yang penulis sertakan ini sepenuhnya dilakukan pada sistem operasi Linux beserta beberapa aplikasinya, sehingga karya ini terjamin kesahihannya dalam hal legalitas.

#### **I.5 SISTEMATIKA PENULISAN**

Bab I *Pendahuluan*, berisi Latar Belakang Masalah, Batasan Masalah, Tujuan Penulisan, Metode Penulisan dan Sistematika Penulisan.

Bab II *Landasan Teori*, berisi semua dasar teori yang untuk selanjutnya digunakan pada bagian pembahasan. Teori yang dibahas antara lain menyangkut TCP/IP, HTTP, HTML, dan juga teori mengenai hal-hal yang mendasari alasan-alasan, kelebihan-kelebihan, serta kekurangan-kekurangan pengembangan aplikasi *client-server* melalui web.

Bab III *Penjelasan PHP*, berisi pembahasan tentang model pemrograman PHP untuk membangun aplikasi web yang dinamis.

Bab IV *Penerapan PHP*, berisi pembahasan secara rinci mengenai aplikasi web yang dibuat dalam tugas akhir ini beserta penjelasan jalannya program.

Bab V *Penutup*, berisi kesimpulan yang diperoleh dari Penulisan Ilmiah ini dan sedikit saran dari penulis.

## BAB II

### LANDASAN TEORI

#### II.1 TCP/IP

Dalam arti yang sederhana, TCP/IP (Transmission Control Protocol/Internet Protocol) adalah nama keluarga protokol jaringan. Protokol adalah sekelompok aturan yang harus diikuti oleh perusahaan-perusahaan dan produk-produk software agar produk mereka itu bisa kompatibel satu dengan yang lainnya. Suatu protokol menentukan bagaimana suatu software berkomunikasi dengan software lain, juga menentukan bagaimana setiap bagian dari keseluruhan paket mengatur perjalanan informasinya.

Internet, dari awal memang sudah dibuat dengan menggunakan TCP/IP pada tingkat protokolnya yang memungkinkan sistem apapun yang terhubung ke dalamnya bisa berkomunikasi dengan sistem lain tanpa mempedulikan bagaimana sistem masing-masing bekerja.

TCP/IP sebenarnya adalah dua macam protokol berbeda. Tidak seperti yang dianggap oleh kebanyakan orang, istilah TCP/IP mengacu kepada seluruh keluarga protokol yang dirancang untuk mentransfer informasi sepanjang jaringan. TCP/IP dirancang untuk menjadi komponen perangkat lunak dari suatu jaringan. Semua bagian di dalam keluarga TCP/IP memiliki tugas tersendiri, misalnya mengirim e-mail, mentransfer file, menyediakan layanan *login* jarak jauh (*remote login*) dan menangani informasi *routing* jaringan.

Protokol TCP bertanggung jawab memecah informasi ke dalam beberapa paket, sedangkan IP bertanggung jawab mengangkut (mentransfer) paket-paket tersebut sesuai tujuannya. Kemudian TCP bertugas menyatukan kembali paket-paket itu ke urutan yang benar.

Layanan dalam TCP/IP yang berbeda dikelompokkan menurut fungsi-fungsinya. Protokol-protokol *transport* mengendalikan pergerakan data antara dua mesin, dan mencakup:

1. TCP (*Transmission Control Protocol*)  
Protokol ini bersifat *connection-based*, artinya kedua mesin pengirim dan penerima tersambung dan berkomunikasi satu sama lain sepanjang waktu.
2. UDP (*User Datagram Protocol*)  
Protokol ini bersifat *connectionless* (tanpa koneksi), artinya data dikirim tanpa kedua mesin penerima dan pengirim saling berhubungan. Ini seperti mengirim surat lewat kantor pos, surat dikirim oleh pengirim namun ia tidak pernah bisa tahu apakah surat tersebut sampai di tujuan atau tidak.

Sementara itu ada pula protokol-protokol *routing* untuk menangani pengalamatan (*addressing*) data dan menentukan jalur terbaik untuk mencapai tujuan. Protokol-protokol tersebut juga bertanggung jawab memecah informasi-informasi ukuran besar dan menyusunnya kembali pada tujuan. Protokol-protokol tersebut antara lain:

- IP (*Internet Protocol*) menangani transmisi data yang sebenarnya.
- ICMP (*Internet Control Message Control Protocol*) menangani informasi status untuk IP, seperti *error* (kesalahan) dan perubahan-perubahan dalam perangkat keras jaringan yang mempengaruhi *routing* (penentuan jalur).
- RIP (*Routing Information Protocol*) dan OSPF (*Open Shortest-Path First*), yaitu satu dari berbagai protokol yang menentukan metode *routing* terbaik untuk menyampaikan data.

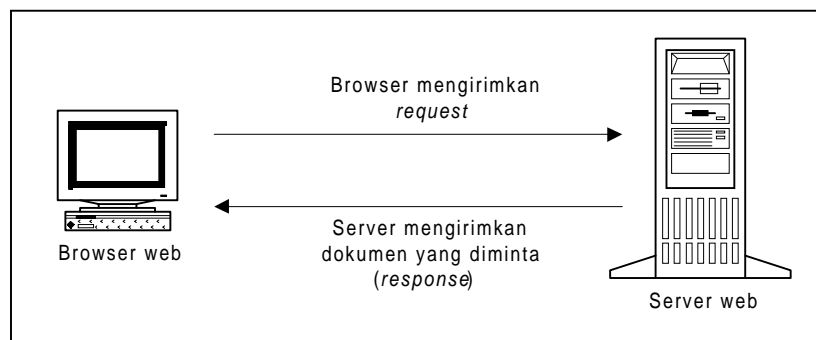
## II.2 World Wide Web

Pada awalnya Internet adalah sebuah proyek yang dimaksudkan untuk menghubungkan para ilmuwan dan peneliti di Amerika, namun saat ini telah tumbuh menjadi media komunikasi global yang dipakai semua orang di muka bumi. Pertumbuhan ini membawa beberapa masalah penting mendasar, di antaranya kenyataan bahwa Internet tidak diciptakan pada jaman *graphical user interface* (GUI) seperti saat ini. Internet dimulai pada masa di mana orang masih menggunakan alat-alat akses yang tidak *user-friendly* yaitu terminal

berbasis teks serta perintah-perintah *command line* yang panjang-panjang serta sukar diingat, sangat berbeda dengan komputer dewasa ini yang menggunakan klik tombol mouse pada layar grafik berwarna.

Kemudian orang mulai berpikir untuk membuat sesuatu yang lebih baik. Popularitas Internet mulai berkembang pesat seperti jamur di musim penghujan setelah standar baru yaitu HTTP dan HTML diperkenalkan kepada masyarakat. HTTP (*Hypertext Transfer Protocol*) membuat pengaksesan informasi melalui protokol TCP/IP menjadi lebih mudah dari sebelumnya. HTML (*Hypertext Markup Language*) memungkinkan orang menyajikan informasi yang secara visual lebih menarik. Pemunculan HTTP dan HTML kemudian membuat orang mengenal istilah baru dalam Internet yang sekarang menjadi sangat populer, bahkan sedemikian populernya sehingga sering dianggap identik dengan Internet itu sendiri, yaitu *World Wide Web* (WWW).

Pada prinsipnya *World Wide Web* (singkatnya cukup disebut "web" saja) bekerja dengan cara menampilkan file-file HTML yang berasal dari *server* web pada program *client* khusus, yaitu *browser* web. Program *browser* pada *client* mengirimkan permintaan (*request*) kepada *server* web, yang kemudian akan dikirimkan oleh *server* dalam bentuk HTML. File HTML berisi instruksi-instruksi yang diperlukan untuk membentuk tampilan. Perintah-perintah HTML ini kemudian diterjemahkan oleh *browser* web sehingga isi informasinya dapat ditampilkan secara visual kepada pengguna di layar komputer.



Gambar 2.1 - Konsep dasar browser dan server web

### II.3 Hypertext Transfer Protocol (HTTP)

Web merupakan terobosan baru sebagai teknologi sistem informasi yang menghubungkan data dari banyak sumber dan layanan yang beragam macamnya di Internet. Pengguna tinggal mengklikkan tombol mousenya pada link-link *hypertext* yang ada untuk melompat ke dokumen-dokumen di berbagai lokasi di Internet. Link-linknya sendiri bisa mengacu kepada dokumen web, *server* FTP (*File Transfer Protocol*), e-mail ataupun layanan-layanan lain.

*Server* dan *browser* web berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini, yaitu HTTP. HTTP bertugas menangani permintaan-permintaan (*request*) dari *browser* untuk mengambil dokumen-dokumen web.

HTTP bisa dianggap sebagai sistem yang bermodel *client-server*. *Browser* web, sebagai *clientnya*, mengirimkan permintaan kepada *server* web untuk mengirimkan dokumen-dokumen web yang dikehendaki pengguna. *Server* web lalu memenuhi permintaan ini dan mengirimkannya melalui jaringan kepada *browser*. Setiap permintaan akan dilayani dan ditangani sebagai suatu koneksi terpisah yang berbeda.

Semua dokumen web dikirim sebagai file teks biasa. Sewaktu mengirimkan *request* kepada *server* web, *browser* juga mengirimkan sedikit informasi tentang dirinya, termasuk jenis-jenis file yang bisa dibaca olehnya. Informasi ini lalu digunakan oleh *server* web untuk menentukan apakah dokumen yang diminta bisa dikirimkan kepada *browser* atau tidak.

Mekanisme kerja HTTP bisa digambarkan sebagai berikut: *Client* (*browser* web) mengirimkan *request* ke *server*, dengan bentuk seperti ini:

```
get /index.html HTTP/1.0
```

kemudian server akan membalas dengan:

```
HTTP/1.0 200 OK
Date: Tue, 04 Feb 1999 23:11:26 GMT
Content-Type: text/html
Content-Length: 1331
...(isi dari dokumen)
Close connection
```



Isi dokumen, yang jenisnya ditentukan pada *header Content-Type* (dalam contoh di atas, sebuah file teks dengan format HTML) selanjutnya akan dibaca oleh *browser* web dan ditampilkan kepada pengguna. Dengan cara ini *browser* web bisa tahu bagaimana ia harus menangani data yang dikirim kepadanya.

HTTP bekerja di atas TCP (Transmission Control Protocol) yang menjamin sampainya data di tujuan dalam urutan yang benar. Bila suatu kesalahan terjadi selama proses pengiriman, pihak pengirim akan mendapat pemberitahuan bahwa telah terjadi ketidakberesan. Karenanya *server* dan *client* tidak harus menyediakan mekanisme untuk memeriksa kesalahan transmisi data, yang berarti mempermudah pekerjaan pemrograman. Namun demikian, HTTP tidak memiliki apa yang disebut *session*, seperti halnya FTP, yang menjaga hubungan antara *server* dan *client* secara konsisten. Setiap halaman web yang dikirim akan melibatkan satu proses penyambungan antara *client* dan *server*, baru kemudian datanya ditransfer. Setelah data selesai ditransfer, koneksi antara *server* dan *client* akan diputus. Sifatnya ini membuat HTTP sering disebut dengan istilah protokol *hit-and-run*.

Suatu halaman web seringkali berisi beberapa file gambar, atau beberapa file-file lain. HTTP memaksa *server* untuk menjalin hubungan baru setiap kali hendak mengirim satu buah file. Ini tidak menguntungkan dan tidak efisien, mengingat proses hubung-putus-hubung semacam ini menyebabkan beban bagi jaringan.

Standar baru protokol HTTP, yaitu HTTP/1.1 yang baru-baru ini diperkenalkan, dirancang untuk mengatasi masalah di atas. Web diarahkan agar mengarah ke penggunaan *persistent connection* (sambungan yang terjaga berkesinambungan) secara lebih efisien. Dalam HTTP/1.1, *server* tidak akan memutuskan hubungan dengan *client* pada akhir penransferan dokumen. Hubungan tetap dibuka untuk melayani bila saja ada *request* lagi dalam waktu yang singkat. Hubungan baru akan diputuskan bila setelah melewati suatu batas waktu tertentu (yang bisa ditentukan oleh administrator *server*) *client* tidak mengirimkan *request* lagi.

Keuntungan lain dari *persistent connection* adalah penggunaan *pipelining*. *Pipelining* adalah proses pengiriman *request* berikutnya segera setelah *request*

sebelumnya dikirimkan tanpa menunggu balasan dari *server* terlebih dahulu. *Server*nya tetap harus melayani setiap *request* secara berurutan, namun ini mengurangi waktu tunda antara setiap *request*. Hasilnya, data akan lebih cepat sampai di tujuan.

Standar HTTP/1.1 ini sekarang sudah mulai dimasyarakatkan dan banyak paket perangkat lunak *server* web komersil dan non-komersil yang sudah mendukung standar baru ini. *Browser-browser* web keluaran terbaru umumnya juga sudah mendukung HTTP/1.1.

#### II.4 Hypertext Markup Language (HTML)

HTML dewasa ini dikenal sebagai bahasa standar untuk membuat dokumen web. Sesungguhnya *Hypertext Markup Language* (HTML) justru tidak dibuat untuk mempublikasikan informasi di web, namun oleh karena kesederhanaan serta kemudahan penggunaannya, HTML kemudian dipilih orang untuk mendistribusikan informasi di web.

Perintah-perintah HTML diletakkan dalam file berekstensi \*.html dan ditandai dengan mempergunakan tag (tanda) berupa karakter "<" dan ">". Tidak seperti bahasa pemrograman berstruktur prosedural seperti Pascal atau C, HTML tidak mengenal *jumping* ataupun *looping*. Kode-kode HTML dibaca oleh *browser* dari atas ke bawah tanpa adanya lompatan-lompatan.

Struktur sebuah dokumen HTML pada dasarnya dibagi menjadi dua bagian besar, yaitu *header* dan *body*. Masing-masing ditandai oleh pasangan container tag <head> dan <body>. Bagian *head* berisikan judul dokumen dan informasi-informasi dasar lainnya, sedangkan bagian *body* adalah data dokumennya. Pengaturan format teks dan pembentukan link dilakukan terhadap obyeknya langsung dengan ditandai oleh tag-tag HTML, seperti terlihat pada contoh berikut:

```
<html><head><title>Ini adalah judul</title></head>
<body bgcolor="#FFFFFF">
<h1>Ini adalah Heading 1</h1>
Ini adalah bagian tubuh dokumen. Semua yang ditulis di sini akan dit
ampilkan ke layar browser
</body></html>
```

HTML diatur oleh Konsorsium WWW (W3C). Semua perubahan atas standar bahasa HTML harus disahkan terlebih dahulu oleh konsorsium ini. Sejauh ini, HTML telah mengalami berbagai revisi sepanjang hidupnya. Standar paling akhir yang sekarang diperkenalkan adalah standar HTML 4.0, yang mendukung antara lain CSS (*cascading style sheet*), *dynamic content positioning* (penempatan isi secara dinamis), *downloadable font* (jenis font yang bisa didownload otomatis) dan sebagainya. Hingga kini, tidak semua *browser* web telah disesuaikan untuk mendukung standar HTML terbaru ini, sehingga banyak masalah inkompatibilitas antara macam-macam *browser* web.

## II.5 BROWSER DAN SERVER WEB

Dalam dunia web, perangkat lunak *client*, yaitu *browser* web mempunyai tugas yang sama yaitu menterjemahkan informasi yang diterima dari *server* web dan menampilkannya pada layar komputer pengguna. Oleh karena HTTP memungkinkan *server* web mengirimkan beragam data, seperti teks atau gambar, *browser* harus bisa mengenali berbagai macam data yang akan diterimanya, dan selanjutnya harus tahu cara untuk menampilkannya dengan benar. Teks harus ditampilkan sebagai teks dan gambar harus ditampilkan sebagai gambar.

Umumnya *browser* web menerima data dalam bentuk HTML. File HTML sebenarnya adalah file teks biasa yang selain berisi informasi yang hendak ditampilkan kepada pengguna, juga mempunyai perintah-perintah untuk mengatur tampilan data tersebut. *Browser*lah yang memiliki kuasa penuh dalam menerjemahkan perintah-perintah tadi. Meskipun sudah dibuat konsensus untuk menstandarkan format dan elemen-elemen HTML, setiap jenis *browser* bisa menterjemahkan file HTML yang sama secara berbeda.

Pada awal pertama kalinya protokol-protokol dasar web dikembangkan yaitu sekitar awal tahun 1990-an, *browser* web pertama yang diperkenalkan adalah Mosaic yang dibuat oleh *National Center for Supercomputing Applications* (NCSA) di Amerika Serikat. Mosaic dimaksudkan agar menjadi sebuah *interface* grafis yang mudah dipergunakan, yang dengan demikian diharapkan dapat mempercepat perkembangan dan dukungan umum akan web. Mosaic langsung

dibuat untuk tiga macam *platform* berbeda, yaitu X Window (untuk lingkungan UNIX dan keluarganya), Microsoft Windows dan Macintosh. Mosaic inilah yang lalu dianggap sebagai legenda yang memacu revolusi web menjadi sedemikian populernya seperti sekarang ini.

Perkembangan jaman serta semakin populernya lingkungan GUI (Graphical User Interface) membuat banyak orang sekarang berlomba-lomba membuat program *browser* yang menarik serta mudah dipakai. *Browser-browser* web modern dilengkapi dengan fasilitas-fasilitas yang mendukung tampilan multimedia berupa audio (suara), animasi 3 dimensi, bahkan video. Program *browser* web yang paling terkenal saat ini adalah Netscape Navigator dan Microsoft Internet Explorer.

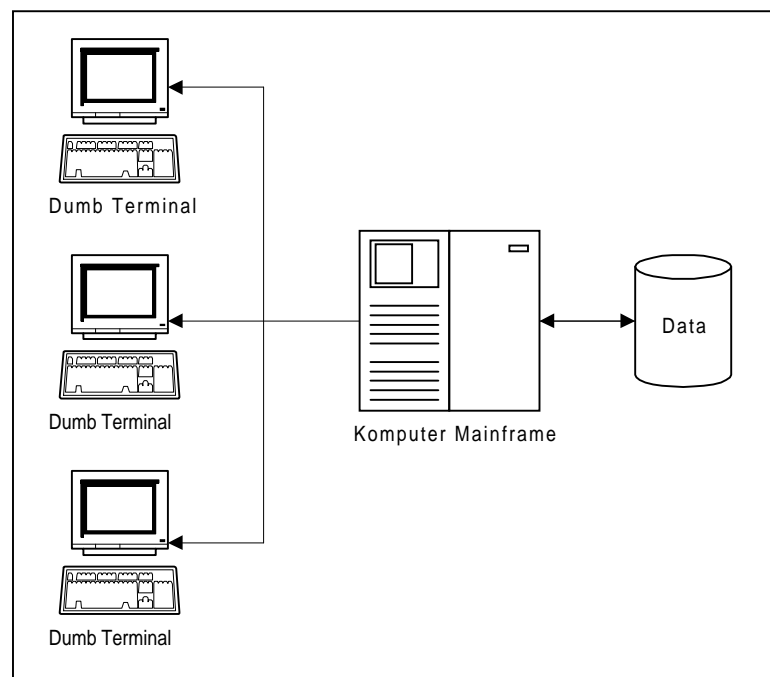
Sementara itu *server* web pada dasarnya adalah perangkat lunak khusus yang bertugas melayani permintaan-permintaan dari *browser* web akan dokumen-dokumen yang tersimpan di dalamnya. Perangkat lunak *server* web sekarang telah tersedia untuk berbagai macam *platform* dan lingkungan sistem operasi Untuk lingkungan UNIX, yang paling populer adalah Apache, Netscape FastTrack dan NCSA HTTPD. Sementara untuk lingkungan Windows tersedia Microsoft Internet Information Server (IIS), Netscape FastTrack, O'Reilly Website dan banyak lagi. Sistem operasi jaringan Novell Netware pun memiliki suatu modul *add-on* yang berfungsi sebagai *server* web, yang bisa dijalankan pada saat *startup* jaringan.

Beberapa perangkat lunak *server* web mempunyai *feature* seperti *server-side programming*, *security control* dan lain sebagainya. Meskipun beragam macamnya, secara fungsional semua jenis *server* web adalah sama saja, yaitu berfungsi melayani permintaan-permintaan dari *browser* web.

## II.6 INTERNET SEBAGAI SISTEM CLIENT/SERVER

Istilah *client/server* dewasa ini telah demikian populer. Keuntungan utama dari sistem berbasis *client/server* adalah bahwa perangkat keras dan perangkat lunak bisa ditempatkan di mana saja mereka bisa bekerja secara lebih optimal. Dulu, di jama komputer *mainframe*, komputer *mainframe*-lah yang menjadi pusat kendali dan mengerjakan semua proses komputasi. Pengguna berinteraksi

dengan sistem *mainframe* melalui terminal-terminal yang dihubungkan secara langsung ke komputer *mainframe*. Terminal-terminal ini tidak punya kemampuan pemrosesan sama sekali, dan oleh karena itu disebut "terminal dungu" (*dumb terminal*). Terminal dungu tidak lebih dari sekedar perpanjangan kabel untuk keyboard dan layar monitor, dan hanya berfungsi sebagai alat untuk memasukkan dan melihat data saja.



Gambar 2.2 - Konsep Sistem Mainframe

Definisi yang banyak dipergunakan untuk menjelaskan sistem berbasis *client/server* adalah "sistem yang memisahkan antara tugas-tugas komputasi antara proses-proses *client* dan *server*." Dengan sistem *client/server*, kekuatan pemrosesan bisa disebarkan (didistribusikan) ke banyak mesin *client* dan mesin *server* yang terpisah secara fisik (itu sebabnya disebut *distributed system*).

Misalnya sebuah *server* web yang mengambil informasi dari database menampilkan hasilnya pada *client* dengan menggunakan *browser* web. *Server* web dan database bisa saja ditempatkan pada satu mesin saja, namun apabila jumlah *client* yang melakukan akses ke *server* semakin banyak dan melebihi kapasitas mesin *server*, perangkat lunak database dan *server* web bisa saja

dipisahkan dan ditempatkan di mesin kedua, ketiga atau bahkan lebih. Dengan begitu pemrosesan pada sisi *server* dapat disebar ke beberapa mesin, yang memungkinkan efisiensi komputasi. Begitu pula, dengan cara ini, kapasitas *server* bisa dikembangkan dan ditingkatkan sesuai dengan kebutuhan.

Dari sudut pandang lain, sistem berbasis *client/server* juga bisa memanfaatkan *browser* web untuk meringankan kerja *server*. Tugas menampilkan informasi dan menyediakan tampilan pengguna (*user interface*) tidak perlu dilakukan secara langsung oleh *server*, namun diserahkan sepenuhnya kepada *browser* web. Dengan hadirnya teknologi pemrograman *client-side* (yang dijalankan di sisi *client*) seperti Java dan bahasa *client-scripting* seperti JavaScript, fungsi-fungsi lain seperti pemeriksaan/validasi input bisa dilakukan oleh *browser* sebelum data dikirimkan kepada *server*, menjamin data yang dikirim ke *server* tidak keliru. Hal ini mempercepat kerja *server*, karena hanya mengerjakan tugas-tugas yang berguna.

Konsep dasar sistem berbasis *client/server* adalah *balancing* (penyebaran), yaitu proses yang mencegah suatu prosesor mengalami *overload* (terbeban lebih) sementara mesin lainnya justru menganggur.

Jadi, setidaknya dalam teori, sistem berbasis *client/server* memberikan keuntungan yang banyak seperti penggunaan *resource* secara lebih efisien, penyimpanan data terpusat, serta lalu lintas data di dalam jaringan menjadi lebih rendah (dibandingkan dengan sistem yang seluruhnya terpusat). Satu-satunya kelemahan utama dari sistem berbasis *client/server* adalah manajemen dan perawatan mesin-mesin *client* yang membutuhkan *upgrading* serta proses konfigurasi yang memakan waktu dan tenaga. Semua program *client*, misalnya *front-end* untuk suatu database, harus dipasang satu persatu di setiap komputer *client*, dan, apabila pada suatu saat program *client* tersebut harus diubah atau dikembangkan, prosesnya harus diulang di setiap komputer *client*.

Bagaimana dengan Internet, dalam hal ini web? Sebenarnya, secara mudah kita bisa mengatakan bahwa konsep web pada awalnya bisa dipandang mirip dengan konsep jaringan dengan *dumb terminal*; pengguna *browser* web mengirim permintaan ke *server* web, lalu menerima informasi dari *server* berupa dokumen statis yang oleh *browser* hanya perlu ditampilkan saja ke layar.

Meskipun komputer *client* memiliki prosesor sendiri, memori sendiri serta media penyimpanan sendiri, sifat-sifat alami *browser* web sebenarnya memenuhi syarat-syarat sebagai *dumb terminal*. Oleh karena *browser* web punya kemampuan untuk membentuk *user interface* (tampilan pengguna) yang sifatnya grafis, sebagian orang menyebutnya sebagai "*smart dumb terminal*" (terminal dungu yang cerdas, atau tidak terlalu dungu).

Struktur fisik dari Internet sendiri juga tidak berbeda jauh dari model sistem *mainframe*. *Bandwidth* (lebar jalur) besarnya terbatas. Tidak seperti teknologi jaringan lokal (LAN) seperti Ethernet atau Token Ring yang bisa membawa informasi sampai 10 bahkan 100 Mbps, kebanyakan pengguna Internet masih terbatas pada sambungan berorde kilobit per detik. Perusahaan-perusahaan besar mungkin sanggup menyewa sambungan kapasitas besar seperti T1 (1,55 Mbps) ke Internet, namun kebanyakan pengguna yang lebih kecil atau rumah tangga hanya mempunyai sambungan berkecepatan rendah, sekitar 33 – 56 Kbps. Bahkan teknologi ISDN pun (yang sampai saat ini masih tergolong mahal) jarang melampaui 128 Kbps.

Melihat kenyataan bahwa pada mesin-mesin yang menjalankan *browser* pada umumnya adalah komputer-komputer modern yang memiliki potensi pemrosesan dan penyimpanan data yang cukup besar, mesin-mesin *client* seharusnya bisa mengerjakan lebih banyak pekerjaan lagi. Juga, tidak seperti sistem berbasis *client/server* "tradisional", kenyataan bahwa penggunaan *browser* web nyaris tidak membutuhkan perawatan yang mahal, *upgrade* berkala dan konfigurasi yang rumit, membuat web (dan Internet pada umumnya) menjadi tempat yang menarik untuk mengembangkan sistem berbasis *client/server*.

Dari sisi pengelola jaringan, sistem berbasis web mengundang banyak perhatian karena sistem semacam ini bisa mengatasi banyak kekurangan dari sistem-sistem tradisional. Pada intinya, mengembangkan sistem *client/server* di web membawa keuntungan-keuntungan langsung seperti:

1. Tidak ada masalah distribusi program

Pendistribusian berlangsung secara sendirinya, karena setiap salinan dokumen (sebagai satu komponen aplikasi) di-*download* ke mesin *client*

setiap saat mesin *client* membutuhkan dan meminta *update* atau salinan yang lebih baru. Tidak perlu lagi seorang administrator jaringan meng-*install* perangkat lunak *client* untuk setiap komputer yang ada di organisasinya, suatu pekerjaan yang melelahkan dan memboroskan waktu.

2. Efisien

Distribusi otomatis dan tidak perlunya instalasi untuk setiap *client* jelas mempermudah perawatan dan *updating* aplikasi. Perubahan-perubahan pada aplikasi bisa dikerjakan secara terpusat dan bisa langsung diterapkan tanpa perlu menyesuaikan semua *client*. Bahkan tampilan pada pengguna bisa diubah-ubah secara berkala, disesuaikan dengan waktu dan situasi.

3. Fleksibel

*Browser* web tersedia untuk hampir semua macam *platform* mesin dan sistem operasi, baik itu mesin Windows berbasis prosesor Intel, mesin UNIX berbasis prosesor RISC, ataupun komputer Macintosh. Fleksibilitas aplikasi web lebih terjamin, karena tidak perlu lagi mengembangkan program-program *client* yang berbeda untuk tiap macam *platform*.

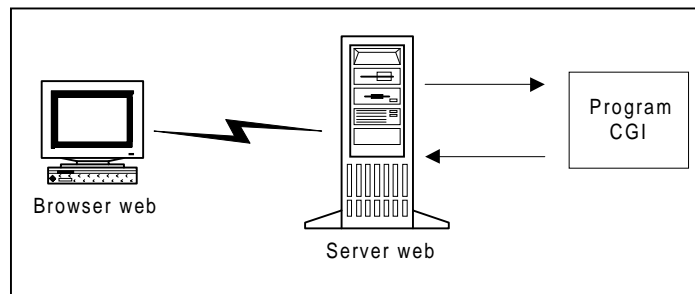
## II.7 WEB DENGAN INFORMASI YANG DINAMIS

Teknologi web tidak berhenti di situ saja. Sejalan dengan perkembangan dan tuntutan pengguna, *server* web tidak hanya duduk diam sambil mengirimkan file-file HTML yang isinya statis belaka, namun *server* web masa kini juga dapat mengirima informasi yang sifatnya dinamis, berubah-ubah sesuai kondisi dan permintaan pengguna. Sekarang orang bisa membangun aplikasi web untuk mengedit file, memasukkan fata, membaca informasi dari *server* database, mengirim e-mail dan bahkan menjadi sebuah "toko virtual" di mana pengguna bisa memilih barang dan membayarnya secara *online* lewat web.

Salah satu kekuatan utama yang memungkinkan developer web membangun aplikasi-aplikasi web yang dinamis adalah kemampuan *server* web untuk mengakses sistem database. *Server-server* web modern sekarang bisa dihubungkan dengan RDBMS (*Relational Database Management System*) seperti



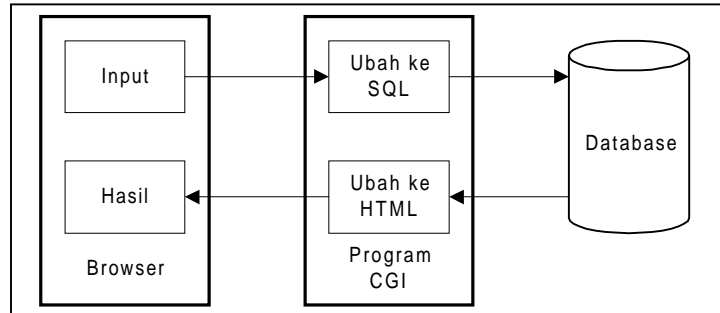
Oracle dan SQL Server tanpa kesulitan. Developer bisa membuat program yang mengakses fungsi-fungsi atau prosedur-prosedur pengaksesan database yang sudah disediakan.



Gambar 2.3 - CGI secara sederhana

Untuk keperluan pengembangan aplikasi web yang dinamis ini, pertama kali diperkenalkan CGI (*Common Gateway Interface*). CGI adalah bagian dari *server web* yang dapat berkomunikasi dengan program lain di luar *server web*. CGI memungkinkan *server web* memanggil suatu program, lalu mengirimkan data-data spesifik dari pengguna ke program tersebut. Hasil proses tadi diterima oleh CGI yang selanjutnya menyerahkannya kepada *server web* untuk kemudian, yang pada gilirannya akan mengirimkan informasi tersebut kembali dalam bentuk HTML ke *browser web* pengguna.

Kebanyakan RDBMS seperti Oracle, Informix dan SQL Server mendukung bahasa SQL (*Structured Query Language*) untuk mengakses dan memanipulasi data yang berada di dalam sistem database relasional. SQL memungkinkan pengaksesan data secara mudah, cepat dan tidak tergantung pada *platform* sistemnya. Program CGI memanfaatkan SQL ini untuk mengakses dan memanipulasi data dalam sistem database. Gambar 2.4 menjelaskan prinsip kerja pengaksesan database melalui CGI.



Gambar 2.4 - Program CGI menjadi penghubung ke sistem database

Seperti terlihat pada gambar 2.4, fungsi program CGI pada *server* web adalah menghasilkan *query* (perintah untuk mengakses database) dalam bentuk SQL kepada sistem RDBMS untuk membaca, memasukkan, menghapus ataupun mengubah data tertentu dari dan ke dalam media penyimpanan data sesuai dengan masukan yang ditentukan oleh pengguna dari *browser* webnya. Program CGI juga bertugas mengubah informasi yang berasal dari RDBMS agar dapat diterima oleh *server* web. *Server* web mengubah bentuk informasi menjadi HTML dan selanjutnya bisa dikirim lagi ke *browser* untuk ditampilkan kepada pengguna.

## II.8 STRUCTURED QUERY LANGUAGE

SQL adalah bahasa standar yang digunakan untuk memanipulasi dan memperoleh data dari sebuah database relasional. SQL membuat pemrogram dan seorang administrator database dapat melakukan hal-hal berikut:

- mengubah struktur sebuah database
- mengubah pengaturan keamanan sistem
- memberikan hak akses kepada pengguna untuk mengakses database atau tabel
- memperoleh informasi dari database
- memutakhirkan isi database

Perintah-perintah SQL dikelompokkan menjadi lima macam:

1. *Data Definition Language (DDL)*  
adalah perintah SQL yang digunakan untuk menjelaskan objek dari database. Dengan kata lain DDL digunakan untuk mendefinisikan kerangka database. Perintahnya adalah:
  - create : untuk membuat/menciptakan objek database
  - alter : untuk memodifikasi/mengubah objek database
  - drop : untuk menghapus objek database
  - Objek database yang dimaksud terdiri dari database, table, index, dan view
2. *Data Manipulation Language (DML)*  
adalah perintah yang digunakan untuk mengoperasikan atau memanipulasi isi database. SQL menyediakan 4 perintah DML:
  - select : digunakan untuk mengambil data dari database
  - delete : digunakan untuk menghapus data pada database
  - insert : menambahkan data ke database
  - update : memodifikasi data pada database
3. Security  
adalah perintah-perintah yang digunakan untuk menjamin keamanan data. Antara lain terdiri atas:
  - grant : memberi akses kepada user tertentu untuk akses ke database
  - revoke : mencabut hak akses dari user
4. Integrity  
adalah perintah-perintah yang digunakan untuk menjaga kesatuan data. Contoh: recover table : untuk memperbaiki tabel pada database
5. Auxilliary  
adalah perintah-perintah pelengkap atau tambahan seperti: unload dan rename.

## II.9 KONSEP OPEN SOURCE

Tujuan dari definisi Open Source adalah untuk melindungi proses Open Source dan menjamin bahwa perangkat lunak yang didistribusikan dengan

menggunakan lisensi *open source* akan tersedia untuk ditinjau secara bebas, dan berkesinambungan mengalami perbaikan secara evolusi, seleksi dan mencapai suatu tingkat kehandalan serta menjaga kemungkinan menjadi produk yang *closed source*. Lisensi ini harus menjamin mencegah orang mengunci perangkat lunak sehingga hanya orang tertentu yang dapat membaca kode sumber dan memodifikasinya. Definisi ini bukan perangkat untuk mengumpulkan biaya lisensi. Penggunaan merk ini bebas dan tetap bebas bagi siapapun yang memenuhi persyaratan. Definisi Open Source sendiri bukanlah lisensi, dan tidak dimaksud sebagai dokumen bernilai hukum (*legal document*). Untuk menjadi Open Source semua syarat dalam definisi Open Source harus dipenuhi bersama pada semua keadaan. Open Source sendiri menjamin hak untuk (Perens, 1999):

- Untuk membuat salinan program, dan mendistribusikan program tersebut
- Untuk mengakses kode sumber, sebelum melakukan perubahan
- Melakukan perbaikan pada program

Definisi di bawah ini mengacu pada *The Open Source Definition Version 1.3* yang awalnya ditulis oleh Bruce Perens dan dikenal sebagai '*The Debian Free Software Guidelines*', setelah diperbaiki dan batasan yang spesifik terhadap Debian dihilangkan maka diperkenalkan sebagai Definisi Open Source. Berikut ini disertakan tulisan asli definisi Open Source dan penjelasan singkat.

1. Redistribusi bebas

"Lisensi ini tidak membatasi pihak manapun untuk menjual atau memberikan perangkat lunak sebagai bagian dari keseluruhan distribusi perangkat lunak yang memuat program dari berbagai sumber berbeda. Lisensi ini tidak membutuhkan royalti atau biaya lainnya sehubungan dengan penjualan dimaksud."

2. Kode sumber

"Program harus memuat kode sumber dan harus memperbolehkan distribusi dalam bentuk kode sumber seperti pada distribusi dalam bentuk yang sudah *dcompile*. Bila bentuk tertentu sebuah produk tidak

didistribusikan dengan kode sumbernya, maka harus secara jelas disediakan kode sumbernya melalui Internet, tanpa pungutan biaya apapun. Kode sumber harus dalam bentuk sedemikian rupa sehingga seorang pemrogram dapat memodifikasi program. "

3. Pekerjaan turunan

"Lisensi harus memperbolehkan adanya modifikasi dan hasil kerja turunan serta pendistribusiannya harus menggunakan persyaratan yang sama dengan lisensi perangkat lunak aslinya."

4. Keutuhan kode sumber dengan pembuatnya

"Lisensi dapat menghalangi kode sumber didistribusi dalam keadaan sudah dimodifikasi hanya bila lisensi memperbolehkan distribusi file *patch* dengan kode sumber untuk maksud memodifikasi program pada saat program itu dibuat. Lisensi harus secara jelas mengizinkan distribusi software yang dibuat dari kode sumber yang sudah dimodifikasi. Lisensi dapat memanfaatkan pekerjaan turunan untuk menggunakan nama atau nomor versi yang berbeda dari software aslinya."

5. Tidak ada diskriminasi terhadap pribadi atau kelompok tertentu

"Lisensi harus tidak melakukan pembedaan perlakuan terhadap pribadi atau kelompok manapun."

6. Tidak ada diskriminasi terhadap bidang pekerjaan apapun

"Lisensi harus tidak membatasi siapapun untuk menggunakan program secara khusus pada suatu bidang pekerjaan tertentu."

7. Pendistribusian lisensi

"Kewenangan yang melekat pada program harus berlaku juga kepada siapapun program itu didistribusikan tanpa perlu membuat lisensi tambahan oleh pihak bersangkutan."

8. Lisensi harus tidak hanya berlaku terhadap satu produk tertentu  
"Kewenangan yang melekat pada program harus tidak tergantung kepada merk distribusi tertentu saja. Bila program tersebut dipisahkan dari suatu distribusi tertentu dan digunakan atau didistribusi ulang dengan menggunakan lisensi program tersebut, semua pihak yang mana ditujukan oleh distribusi program tersebut mempunyai kewenangan yang sama dengan mereka yang memperolehnya dengan distribusi perangkat lunak aslinya."
9. Lisensi harus tidak mempengaruhi lisensi perangkat lunak lainnya.  
"Lisensi harus tidak melarang software lain yang didistribusi dengan menggunakan perangkat lunak berlisensi. Contohnya, lisensi harus tidak mendesak bahwa program didistribusi dengan menggunakan media yang sama harus merupakan perangkat lunak *open source*."
10. Penyesuaian lisensi dan sertifikasi  
"Perangkat lunak yang menggunakan lisensi yang disertifikasi sesuai dengan Definisi Open Source dapat menggunakan merk dagang Open Source, dan juga tersedia bagi masyarakat umum. Tidak ada lisensi atau software lainnya yang disertifikasi untuk menggunakan merk dagang Open Source."

### II.10 HTML Cookie

*Cookie* adalah sebuah mekanisme yang digunakan oleh *server-side connections* (seperti halnya *script CGI*) untuk menyimpan dan mengambil informasi dari sisi *client*. Tambahan yang bersifat sederhana dan berkesinambungan pada sisi *client* ini menambah jangkauan kemampuan yang aplikasi *client/server* berbasis web.

Ketika *server* mengembalikan sebuah objek HTTP kepada *client*, *server* bisa juga mengirim informasi status yang akan disimpan oleh *client*. Termasuk di dalamnya adalah jangka waktu URL tersebut valid. Permintaan berikutnya (berupa HTTP request) dari *client* yang masih berada dalam jangka waktu yang

valid akan turut mengirim isi dari objek, dari *client* kembali ke *server*. Objek status ini disebut *cookie*.

Sebuah *cookie* diberikan pertama kali kepada *client* dengan menyertakan *header Set-Cookie* sebagai bagian dari sebuah HTTP *response*, yang biasanya hal ini dihasilkan oleh sebuah *script* CGI. Berikut adalah format sebuah *script* CGI yang digunakan untuk memberikan *cookie* kepada *client*:

```
Set-Cookie: NAME=VALUE; expires=DATE;  
path=PATH; domain=DOMAIN_NAME; secure
```

NAME=VALUE

String ini adalah sekelompok karakter (termasuk titik koma, koma, dan spasi). Ini adalah satu-satunya atribut yang dibutuhkan pada *header Set-Cookie*.

expires=DATE

Atribut *expires* berisi string tanggal yang menunjukkan jangka waktu valid dari *cookie* tersebut. Ketika jangka waktu valid sudah dilewati (kadaluwarsa), *cookie* tersebut akan dihapus. String tanggal tersebut harus dalam format:

```
Wdy, DD-Mon-YYYY HH:MM:SS GMT
```

atribut *expires* sifatnya optional. Bila tidak disertakan, *cookie* akan kadaluwarsa ketika sesi pengguna berakhir.

domain=DOMAIN\_NAME

Ketika pencarian dari daftar *cookie* untuk mendapatkan *cookie* yang valid, sebuah perbandingan dari atribut domain *cookie* tersebut dilakukan terhadap nama domain *host* Internet tempat mana URL dijalankan. Bila terdapat *tail matching*, maka *cookie* akan melakukan *path matching* untuk menentukan apakah *cookie* tersebut harus dikirim. *Tail matching* berarti atribut *domain* dibandingkan dengan akhir dari seluruh nama domain *host*. Sebuah atribut *domain* dari "mark.com" akan cocok dengan nama *host* "saya.mark.com", dan juga "aku.bukan.mark.com". Isi default atribut

`domain` adalah nama *host server* tempat mana *server* tersebut menghasilkan *cookie*.

`path=PATH`

Atribut `path` digunakan untuk menunjukkan subset URL pada sebuah domain tempat mana *cookie* tersebut valid. Bila sebuah *cookie* sudah melewati *domain matching*, maka komponen *path* dari URL dibandingkan dengan atribut `path`, dan bila cocok, maka *cookie* valid untuk dikirimkan bersama *request* URL. Path `"/susilo"` dianggap cocok dengan `"/susilowati"` dan juga `"/susilo/wati.html"`. Yang paling umum adalah *path* `"/`. Bila tidak ada atribut `path` disertakan, maka *path* dianggap sama dengan *path* dokumen yang mengandung *header* berisi *cookie* berada.

`secure`

Bila sebuah *cookie* ditandai `secure`, maka *cookie* tersebut hanya akan dikirimkan bila jalur komunikasi dengan *host* adalah jalur yang `"secure"`. Hal ini berarti *cookie* yang `"secure"` hanya dikirimkan ke *server* HTTPS (HTTP melalui SSL).

Beberapa ketentuan *cookie*:

- beberapa *header Set-Cookie* bisa dikirimkan melalui sebuah *server response*.
- *cookie* akan menimpa *cookie* sebelumnya dengan *path* dan nama yang sama.
- batas banyak *cookie* yang dapat disimpan oleh *client*:
- paling banyak 300 buah *cookie*
- paling banyak 4 kilobyte per *cookie*
- paling banyak 20 *cookie* per *server* atau domain

untuk menghapus *cookie*, sebuah *script* CGI bisa mengirimkan *cookie* dengan nama yang sama dan waktu kadaluwarsa yang sudah lewat.



## II.11 PEMBAHASAN PHP

### II.11.1 Tentang PHP

PHP adalah teknologi yang diperkenalkan tahun 1994 oleh Rasmus Lerdorf. Beberapa versi awal yang tidak dipublikasikan digunakan pada situs pribadinya untuk mencatat siapa saja yang mengakses daftar riwayat hidup *onlinenya*. Versi pertama digunakan oleh pihak lain pada awal tahun 1995 dan dikenal sebagai Personal Home Page Tools. Terkandung di dalamnya sebuah *parser engine* (mesin pengurai) yang sangat disederhanakan, yang hanya mampu mengolah *macro* khusus dan beberapa utilitas yang sering digunakan pada pembuatan *home page*, seperti buku tamu, pencacah, dan hal semacamnya. *Parser* tersebut ditulis ulang pada pertengahan 1995 dan dinamakan PHP/FI Version 2. FI (Form Interpreter) sendiri berasal dari kode lain yang ditulis juga oleh Rasmus, yang menterjemahkan HTML dari data. Ia menggabungkan *script* Personal Home Page Tools dengan Form Interpreter dan menambahkan dukungan terhadap server database yang menggunakan format *mSQL*, sehingga lahirlah PHP/FI. PHP/FI tumbuh dengan pesat, dan orang-orang mulai menyiapkan kode-kode programnya supaya bisa didukung oleh PHP.

Sulit memberikan data statistik yang akurat, namun diperkirakan pada akhir 1996 PHP/FI sudah digunakan sedikitnya pada 15.000 situs web di seluruh dunia. Pada pertengahan 1997, angka tersebut berubah menjadi 50.000. Pada saat itu juga terdapat perubahan di dalam pengembangan PHP. PHP berubah dari proyek pribadi Rasmus menjadi sebuah tim yang lebih terorganisasi. *Parse*nya ditulis ulang dari bentuk rancangan awal oleh Zeev Suraski dan Andi Gutmans, dan *parser* baru ini adalah sebagai dasar PHP Version 3. Banyak kode utilitas yang berasal dari PHP/FI *diport* ke PHP3, dan banyak diantaranya sudah selesai ditulis ulang secara lengkap.

Pada pertengahan 1998, baik PHP/FI maupun PHP3 dikemas bersama dengan produk-produk komersial seperti *server* web StrongHold buatan C2 dan Linux RedHat, dan menurut survei yang dilakukan oleh NetCraft, kemungkinan PHP digunakan pada lebih dari 150.000 situs web di seluruh dunia. Sebagai

pembandingan, angka tersebut lebih banyak daripada pengguna *server* web Enterprise Server buatan Netscape di internet.

### II.11.2 Apa itu PHP

PHP singkatan dari Personal Home Page Tools, adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan di sisi *server*. Sebagian besar perintahnya berasal dari C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan *server*. PHP juga dimaksudkan untuk mengganti teknologi lama seperti CGI (Common Gateway Interface).

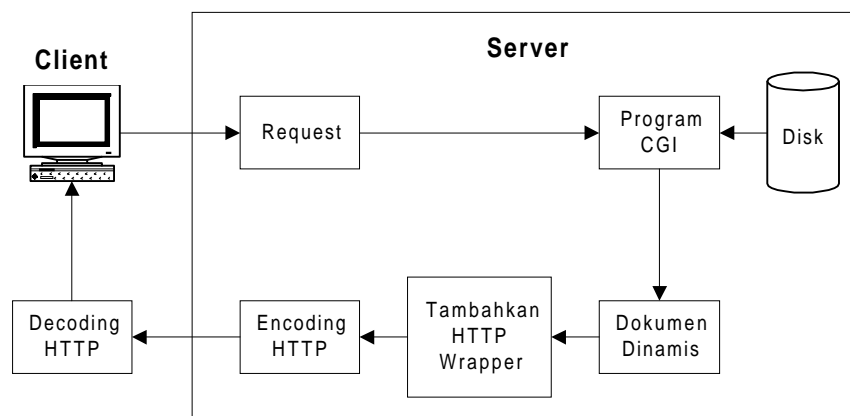
PHP bisa berinteraksi dengan hampir semua teknologi web yang sudah ada. Developer bisa menulis sebuah program PHP yang mengeksekusi suatu program CGI di *server* web lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs web yang besar dan sibuk, karena pemilik masih bisa mempergunakan aplikasi-aplikasi yang sudah terlanjur dibuat di masa lalu dengan CGI, ISAP, atau dengan *script* seperti Perl, Awk atau Python selama proses migrasi ke aplikasi baru yang dibuat dengan PHP. Ini mempermudah dan memperhalus peralihan antara teknologi lama dan teknologi baru.

### II.11.3 Web Dinamis

Saat *server* melayani permintaan dari *browser* web akan suatu dokumen, *server* sebenarnya hanya mengambil suatu file di dalam disk dan melakukan beberapa pekerjaan untuk transmisi seperti menambahkan informasi tipe dokumen, merubah formatnya agar bisa dikirim menggunakan HTTP, yang mengirimkan semuanya ke *browser*. *Browser* web menerima file HTML dan menampilkannya ke layar monitor *client*. Sumbernya tetap berada di *server* dan di sana ia tidak berubah sama sekali. Inilah web yang "biasa-biasa" saja, web yang statis.

Orang kemudian mulai membuat metode baru yang membuat *server* harus melakukan lebih dari sekedar mengirim file. Jika permintaan dari *browser* mengarah ke suatu file program CGI (misalnya Perl, yang biasanya berekstensi

\*.pl), maka *server* mendeteksinya sebagai suatu permintaan untuk menjalankan program di luar *server*. *Server* lalu menjalankan program aplikasi yang dimaksud. Program aplikasi lalu mengeluarkan hasil kerjanya ke *server*, kemudian mengirimkannya kembali ke *browser* dalam bentuk HTML seperti biasa. Bagi *browser* web, informasi yang diterima tetap serupa dengan dokumen HTML statis biasa, namun dokumen tersebut sudah bukan lagi sebuah salinan dari file yang ada di disk *server*, melainkan suatu informasi yang dihasilkan secara *on-the-fly* oleh program aplikasi. Informasi dari program dimasukkan ke dalam dokumen HTML sebelum dikirimkan ke *browser*. Metode ini sering disebut SSI (*Server Side Includes*).



Gambar 3.1 - Common Gateway Interface

Pendekatan cara CGI ini, misalnya dengan *script* Perl, membutuhkan suatu file *template*, yaitu suatu file dokumen yang menjadi tempat penampung informasi hasil kerja program sebelum dikirimkan ke *browser* web. File ini berisi kode-kode khusus yang kemudian akan diganti dengan informasi hasil kerja penerjemah *script* CGI. Hasilnya, dokumen yang dikirim ke *browser* web sebenarnya adalah kombinasi dari informasi dinamis dari program aplikasi serta informasi statis dari file *template* tadi. Developer harus membuat dua file terpisah, yaitu *script* program dan file *templat*nya.

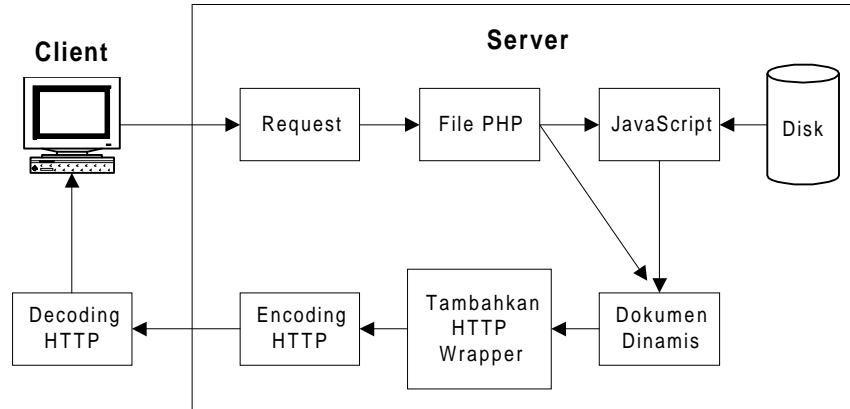
#### II.11.4 Pendekatan Cara PHP

PHP menawarkan solusi yang lebih luwes. Dengan PHP, developer tidak perlu lagi berurusan dengan dua buah file terpisah seperti itu. *Browser* web mengacu secara langsung ke file yang dituju, yang lalu dibaca oleh *server* sebagaimana file HTML statis biasa. Bedanya, sebelum dikirim balik ke *browser* web, *server* web memeriksa isi file dan menentukan apakah ada kode di dalam file tersebut yang harus dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya dimasukkan ke dalam dokumen yang sama. *Server* web bekerja secara langsung terhadap file yang bersangkutan, tidak memanggil *script* terpisah seperti pada metode CGI. Seluruh kode dieksekusi di *server* (oleh karena itu disebut *server-side script*).

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu:

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga developer bisa berkonsentrasi langsung pada penampilan dokumen webnya.
2. Tidak ada proses *compiling* dan *linking*.
3. Berorientasi obyek (*object oriented*).
4. Sintaksis pemrogramannya mudah dipelajari, sangat menyerupai C dan Perl.
5. Integrasi yang sangat luas ke berbagai server database. Menulis web yang terhubung ke database menjadi sangat sederhana. Database yang didukung oleh PHP: Oracle, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePro, Velocis, Informix, dBase, UNIX dbm.

PHP tidak terbatas untuk hanya menghasilkan keluaran HTML. Ia juga bisa digunakan untuk menghasilkan gambar GIF, atau bahkan sumber gambar GIF yang dinamis.



Gambar 3.2 - Prinsip kerja PHP dan Apache

### II.11.5 Dasar-dasar PHP

Seperti yang telah dikemukakan sebelumnya, kode (istilah *script* lebih populer dikalangan pemrogram untuk menyebut kode program yang dijalankan dengan *interpreter*) yang bisa ditangani oleh PHP adalah C. Karena dieksekusi di *server*, PHP disebut sebagai *server-side script*.

Pengertian ini penting karena *browser-browser* web masa kini umumnya juga telah sanggup mengeksekusi *client-side script*, yaitu *script* yang dijalankan di sisi *client*. Pada awalnya C adalah *client-side script*, sehingga untuk membedakannya dari *server-side script* perlu ada suatu mekanisme khusus.

Semua *script* PHP harus ditandai secara khusus dengan *tag* tertentu agar oleh *server* tidak dianggap sebagai *client-side script* yang akan diloloskan ke *browser* web tanpa diolah olehnya.

Untuk itu, PHP mempunyai empat buah cara untuk membedakannya dari *script* HTML dan memasuki "mode PHP":

```
<? echo ("ini adalah cara paling sederhana untuk memproses instruksi SGML\n"); ?>
```

```
<?php echo ("bila Anda hendak menyajikan dokumen XML, lakukanlah dengan cara ini\n"); ?>
```

```
<script language="php">
echo ("beberapa editor HTML (seperti FrontPage) tidak menyukai pemrosesan instruksi");
</script>
```

```
<% echo ("Dimulai dari PHP 3.0.4 Anda dapat menggunakan tag ala ASP"
); %>
```

Blok-blok *script* ini ditempatkan di file yang sama dengan file HTML biasa, namun untuk memudahkan pengenalan, semua file *script* PHP diberi ekstensi \*.php3, bukannya \*.html. *Script* yang dimaksudkan agar dieksekusi di *browser* (yaitu *client-side script*) biasanya ditandai dengan tag `<script>` dan `</script>`. Contohnya dapat dilihat berikut ini:

```
<html>
<head><title>Percobaan PHP</title></head>
<body>
Ini adalah contoh script yang dijalankan di server:<br>
<? echo "Ini script yang dijalankan di server." ?>
dan di bawah ini adalah script yang dijalankan di browser: <br>
<script language=javascript>
    document.writeln("Ini script yang dijalankan di browser!");
</script>
</body>
</html>
```

Adapun demikian script yang berada di antara `<script>` dan `</script>` juga bisa dieksekusi di sisi *server*. Untuk itu ditambahkan atribut `runat=` pada tag `<script>` ini. Contohnya:

```
<script language=javascript runat=server>
    ...kode program...
</script>
```

Ini diperlukan terutama bila kita hendak menulis suatu subrutin atau fungsi di dalam file PHP, karena semua *script* yang ada di antara tag `<?>` dan `?>` akan langsung dieksekusi begitu file dibaca, oleh karenanya tidak cocok bila kita hendak menempatkan fungsi atau subrutin yang hanya perlu dibaca bila dipanggil.

Secara gamblang, diagram di bawah ini akan menjelaskan bagaimana perbedaan antara *tag-tag script* `<? dan ?>` serta `<script>` dan `</script>`:

```
<html>
<head> ... </head>
<body>
<?
... → script PHP
?>
<script runat=server>
... → server-side script
</script>
<script>
... → client-side script
</script>
</body>
</html>
```

Pemisahan antar instruksi berlaku sama seperti pada C atau Perl, yaitu menggunakan ':' (titik koma) sebagai pemisah antar *statement*. Tag penutup ('?>') juga menyatakan akhir *statement*, sehingga kedua perintah berikut ini sama artinya:

```
<?
echo "Percobaan...";
?>

<? echo "Percobaan..." ?>
```

### II.11.6 Variabel pada PHP

PHP mendukung beberapa jenis variabel sebagai berikut:

1. integer. Variabel berjenis integer bertujuan untuk menyimpan bilangan bulat (tidak termasuk bagian pecahan dan/atau pemangkatannya)
2. double. Double digunakan untuk menyimpan bilangan bernilai pecahan dan juga bilangan pemangkatan.
3. string. String merupakan jenis data karakter yang disimpan sebagai nomor pada memori komputer. Nilai yang disimpan adalah nilai ASCII karakter string tersebut.

4. array. Array adalah sebuah set variabel yang mempunyai jenis data yang sama. Array mengandung komponen yang dinamakan elemen. Elemen-elemen sebuah array disimpan pada lokasi tertentu pada memori.
5. object. Jenis variabel object adalah berdasarkan gambaran objek pada dunia nyata, yang mempunyai "status" dan "tingkah laku". Sebuah variabel object menyimpan statusnya dalam bentuk variabel dan tingkah lakunya berupa parameter.
6. pdfdoc (hanya bila dukungan PDF diaktifkan)
7. pdfinfo (hanya bila dukungan PDF diaktifkan)

Jenis variabel biasanya tidak ditentukan oleh pemrogram, melainkan ditentukan oleh PHP saat *runtime*, tergantung pada konteks apa variabel itu digunakan. Misalkan variabel `coba` berisikan nilai string, maka variabel `coba` berjenis string, atau integer bila berisi nilai integer. Contoh:

```
$coba = "0"; // $coba adalah sebuah variabel string (ASCII 48)
$coba++; // $coba adalah string "1" (ASCII 49)
$coba += 1; // $coba berubah menjadi integer (2)
$coba = $coba + 1.3; // $coba berubah menjadi double (3.3)
```

#### II.11.6.1 KONVERSI STRING

Ketika sebuah variabel berisikan nilai string diproses sebagai numerik, nilai dan jenis hasilnya ditentukan sebagai berikut: string tersebut akan diproses sebagai integer, kecuali bila mengandung karakter '.', 'e', atau 'E', maka akan diproses sebagai jenis double. Bila string diawali dengan data numerik yang valid, maka nilai data numerik itulah yang digunakan, selain itu nilai string = 0 (nol). Data numerik valid adalah tanda bilangan (positif atau negatif) yang sifatnya optional, diikuti oleh satu atau lebih digit (boleh mengandung desimal), diikuti oleh keterangan eksponen (juga optional). Eksponen adalah sebuah 'e' atau 'E', diikuti oleh satu atau lebih digit. Contoh:

```
$coba = 1 + "10.5"; // $coba berjenis double (11.5)
$coba = 1 + "-1.3e3"; // $coba berjenis double (-1299)
$coba = 1 + "mark-1.3e3"; // $coba berjenis integer (1)
$coba = 1 + "mark3"; // $coba berjenis integer (1)
```



```
$coba = 1 + "10 anak sapi"; // $coba berjenis integer (11)
$coba = "10.0 sapi " + 1; // $coba berjenis integer (11)
$coba = "10.0 sapi " + 1.0; // $coba berjenis double (11)
```

Jenis variabel tergantung kepada ekspresi yang kedua, bila ekspresi pertama berupa sebuah string.

#### II.11.6.2 INISIALISASI VARIABEL

Inisialisasi variabel pada PHP dilakukan dengan cara memberinya sebuah nilai. Namun pada jenis variabel array dan object, digunakan cara yang sedikit berbeda.

##### II.11.6.2.1 Menginisialisasi Variabel Array

Sebuah array bisa diinisialisasi dengan dua cara: dengan memberikannya sebuah nilai secara berurutan, dan menggunakan *counstruct* array(). Contoh:

```
$nama[] = "Joko"; // $nama[0] = "Joko"
$nama[] = "Supriyanto"; // $nama[1] = "Supriyanto"
```

Seperti pada C dan Perl, elemen array diawali dengan 0, bukan 1.

##### II.11.6.2.2 Menginisialisasi Variabel Object

Inisialisasi variabel berjenis object dilakukan dengan menggunakan statement *new* untuk menyisipkan object ke sebuah variabel. Contoh:

```
class kuda {
    function kudalah() {
        echo "Menjadi seekor kuda.";
    }
}
$keledai = new kuda;
$keledai -> kudalah();
```

### II.11.6.3 VARIABEL LUAR

Ketika sebuah form dikirim menuju sebuah *script* PHP, semua variabel yang ada pada form tersebut secara otomatis juga bisa digunakan oleh PHP. Sebagai contoh adalah form berikut:

```
<form action="coba.php3" method="post">
  Nama: <input type="text" name="nama"><br>
  <input type="submit">
</form>
```

Ketika dikirim, PHP akan membuat sebuah variabel `$nama`, yang berisi apapun yang diisikan melalui field "Nama:" pada form.

Ketika mengirim sebuah form, juga bisa menggunakan sebuah gambar daripada sekedar tombol "Kirim" dengan menggunakan tag seperti:

```
<input type="image" src="gambar.gif" name="sub">
```

Ketika pengguna mengklik dimanapun pada gambar, form yang bersangkutan akan dikirimkan ke *server* dengan tambahan dua buah variabel, yaitu `sub_x` dan `sub_y`. Variabel-variabel ini menyimpan koordinat tempat pengguna mengklik gambar.

### II.11.6.4 VARIABEL ENVIRONMENT

PHP secara otomatis membuat variabel environment bisa digunakan sebagai variabel normal. Contoh:

```
echo $HOME; /* menunjukan penggunaan variabel environment $HOME, bila diaktifkan */
```

### **II.11.7 Cookie pada PHP**

PHP secara gamblang mendukung *HTTP Cookies* seperti ditetapkan oleh Netscape's Spec. Pada PHP, *cookie* bisa diatur melalui fungsi `setcookie()`. *Cookie* adalah bagian dari *header* HTTP, jadi fungsi `setcookie()` harus dipanggil sebelum hasil apapun dikirim kembali ke *browser*. Setiap *cookie* yang dikirim ke

*server* dari *client* akan diubah secara otomatis menjadi sebuah variabel PHP, sama seperti cara pengiriman data menggunakan GET dan POST.

Untuk memberikan beberapa nilai pada sebuah *cookie*, tambahkan [] pada nama *cookie*. Sebagai contoh:

```
SetCookie ("cookieku[]", "Coba man...", time()+3600);
```

### II.11.8 Fungsi pada PHP

Sebuah fungsi didefinisikan menggunakan perintah:

```
function coba ($arg_1, $arg_2, ..., $arg_n) {  
    echo "Contoh fungsi.\n";  
    return $nilai;  
}
```

Fungsi harus didefinisikan terlebih dulu sebelum mereka digunakan. Pengembalian nilai dari suatu fungsi menggunakan perintah `return`. Semua jenis data bisa digunakan sebagai nilainya, termasuk list dan object.

## BAB III

### MASALAH DAN PENERAPAN PHP

#### III.1 PERMASALAHAN

Aplikasi yang dikembangkan sebagai penerapan PHP bernama "Pemilo 1999", yaitu sebuah aplikasi pemilihan, yang menangani pemilihan logo partai-partai yang mengikuti PEMILU RI 1999. Pengunjung dapat melihat profil logo-logo partai, memilih logo yang disukainya (penulis menyebut pemilih sebagai "pencoblos"), melihat hasil pemilihan, dan memberi tanggapan tentang aplikasi ini. Pemilihan berlangsung selama satu minggu, dimana selama waktu pemilihan tersebut dapat dilihat klasemen sementara pemilihan dan 5 logo terbaik. Pemungutan suara bisa dilakukan darimana pun pemilih menghendaknya, selama ia terhubung ke Internet.

Untuk keamanan, PEMILO 1999 menggunakan fasilitas *cookie* sebagai alat untuk menentukan bahwa seorang pemilih hanya boleh dan hanya bisa sekali menyoblos selama waktu pemilihan. Begitu pula dengan nomor logo pilihan setiap penyoblos dienkripsi dengan menggunakan teknologi *md5* dari RSA Inc. Berikut tampilan utama "PEMILO 1999":



Gambar 3.1 - Tampilan utama "PEMILO 1999"

## III.2 PENERAPAN PHP

### III.2.1 Peralatan yang Digunakan

Pada dasarnya Pemilo 1999 adalah aplikasi *client/server* yang menggunakan media Internet. Untuk dapat berjalan dengan baik, aplikasi ini membutuhkan beberapa komponen utama, yaitu:

1. *server* web sebagai *server*
2. *browser* web sebagai *client*
3. *server* database
4. bahasa CGI, yang disatukan pada *server* web

Adapun perangkat lunak yang digunakan pada aplikasi ini adalah sebagai berikut:

- Apache versi 1.3.6, sebuah *server* web berkemampuan tinggi.
- sebagai *client*, digunakan *browser* web Netscape Navigator versi 4.61
- MySQL versi 3.22, sebuah RDBMS (*Relational Database Managenet System*) multi-user berkemampuan tinggi yang dirancang untuk bekerja pada berbagai macam *platform*.
- PHP versi 3.0, yang merupakan bagian dari *server* web
- Linux dengan *kernel* 2.2.9, sebagai sistem operasi *server* yang handal dan sifatnya *open source*
- JavaScript sebagai *client-side scripting language*.

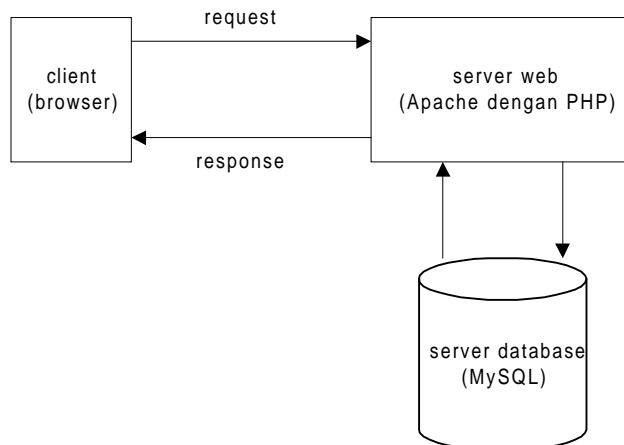
Perangkat keras yang digunakan untuk mengembangkan dan mengujicoba aplikasi "PEMILO 1999" ini adalah:

- Sebuah mesin berbasis Intel untuk mengoperasikan *server*.
- Beberapa mesin dengan beragam sistem operasi sebagai *client*.
- Jaringan *private* TCP/IP.

Karena sifatnya pemilihan, maka aplikasi ini dijalankan pada sistem *open source* dengan tujuan agar keseluruhan proses pemilihan dapat berlangsung secara murni jujur dan adil. Penggunaan Linux sebagai sistem operasi adalah

agar dapat menggunakan fungsi-fungsi PHP yang membutuhkan beberapa library, seperti untuk pembuatan grafik.

Adapun hubungan antar komponen di atas digambarkan sebagai berikut:



Gambar 3.2 - Hubungan antar komponen aplikasi web

**III.2.2 Struktur Database**

“PEMILO 1999” mempunyai tugas inti sebagai berikut:

1. menyimpan informasi mengenai logo partai
2. menyimpan nomor pilihan pemilih
3. menyortir logo partai dengan penyoblos terbanyak
4. menyimpan komentar pemilih

Oleh karena itu, penulis membuat sebuah database bernama `pemilu` yang terdiri dari 3 (tiga) buah tabel, yakni tabel `logo`, tabel `pilihan`, dan tabel `bukutamu`.

III.2.2.1 TABEL PILIHAN

Tabel ini mempunyai struktur sebagai berikut:

Tabel 3.1 Tabel Pilihan

Field	Jenis	Ukuran	Keterangan
<code>ipadd</code>	karakter	15	alamat IP penyoblos
<code>no_pilihan</code>	karakter	2	nomor urut logo yang dicoblos

Tabel ini dibuat dengan definisi berikut:

```
create table pilihan (
    IPadd varchar(15) not null,
    no_pilihan char(2) not null
);
```

Jumlah record pada tabel ini bertambah sesuai banyaknya pencoblosan. Alamat IP pencoblos dicatat pada field `ipadd` dan nomor logo partai pilihannya dicatat pada field `no_pilihan`.

### III.2.2.2 TABEL LOGO

Sebetulnya untuk tugas mencatat dan menghitung pemilihan, aplikasi ini bisa dibuat hanya dengan menggunakan satu buah tabel, yaitu tabel `pilihan`. Namun untuk menampilkan fasilitas Lima Besar, proses ini membutuhkan penyortiran berdasarkan nomor logo partai yang dicoblos. Dengan memperhitungkan unsur kestabilan dan kecepatan, maka untuk proses ini digunakan satu tabel lagi, yaitu tabel `logo`.

Tabel ini mempunyai struktur sebagai berikut:

Tabel 3.2 Tabel Logo

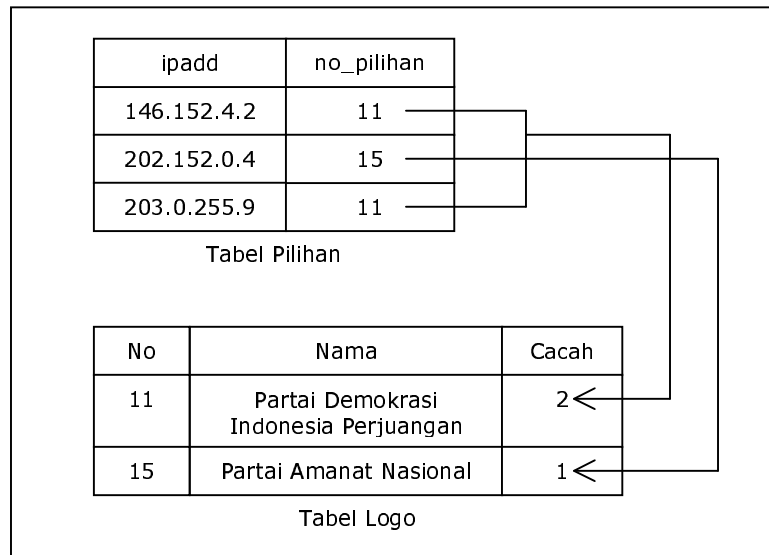
Field	Jenis	Ukuran	Keterangan
no	numerik	2	nomor urut logo
nama	karakter	50	nama logo
cacah	numerik	5	pencacah logo

Tabel ini dibuat dengan definisi berikut:

```
create table logo (
    no smallint(2) not null,
    nama varchar(45) not null,
    cacah int(11) not null default '0'
);
```

Jumlah record pada tabel ini tetap, yaitu 48 record, sebanyak jumlah logo partai peserta Pemilu 1999. Field yang berubah setiap saat ada pengunjung yang melakukan penyoblosan adalah field `cacah`, yang berfungsi sebagai penghitung untuk mencatat banyak pemilih yang menyoblos logo tersebut. Dengan demikian setiap jumlah penyoblos suatu

logo partai berubah, secara otomatis akan dicacahkan pada tabel logo sesuai logo yang bersangkutan. Untuk lebih jelasnya bisa dilihat pada gambar berikut:



Gambar 3.3 - Hubungan tabel pilihan dengan tabel logo

**III.2.2.3 TABEL BUKUTAMU**

Tabel ini mempunyai struktur sebagai berikut:

Tabel 3.3 Tabel Bukutamu

Field	Jenis	Ukuran	Keterangan
no	numerik	5	nomor urut entri
nama	karakter	50	nama
email	karakter	50	alamat <i>e-mail</i>
pekerjaan	karakter	20	pekerjaan
alamat	karakter	50	alamat rumah
komentar	karakter	256	komentar
url	karakter	50	alamat situs



Tabel ini didefinisikan dengan perintah berikut:

```
create table bukutamu (
  no mediumint(8) default '0' not null auto_increment,
  nama varchar(30) default '' not null,
  email varchar(30) default '' not null,
  alamat varchar(30) default '' not null,
  pekerjaan varchar(30) default '',
  komentar text default '' not null,
  url varchar(50),
  primary key (no)
);
```

Tabel ini mempunyai kunci utama yaitu field `no`. Tabel ini diisi ketika ada pengunjung yang memberikan tanggapannya melalui fasilitas Buku Tamu.

#### III.2.2.4 TABEL TAMBAHAN

Sebagai keamanan, PEMILO 1999 perlu mengakses sebuah tabel yang berisi data calon pemilih yang sah, agar hanya pemilih yang sah sajalah yang mempunyai hak untuk menyoblos. Tabel ini bisa terletak di dalam maupun di luar database `pemilo`. Tabel ini penulis beri nama `pemilih`, yang mempunyai struktur sebagai berikut:

Tabel 3.4 Tabel Pemilih

Field	Jenis	Ukuran	Keterangan
<code>noktp</code>	karakter	15	nomor KTP penyoblos
<code>nama</code>	karakter	50	nama sesuai pada KTP

Tabel ini didefinisi dengan perintah berikut:

```
create table pemilih (
  noktp varchar(20) not null,
  nama varchar(30) not null,
  primary key (id),
  unique idx_nama (nama)
);
```

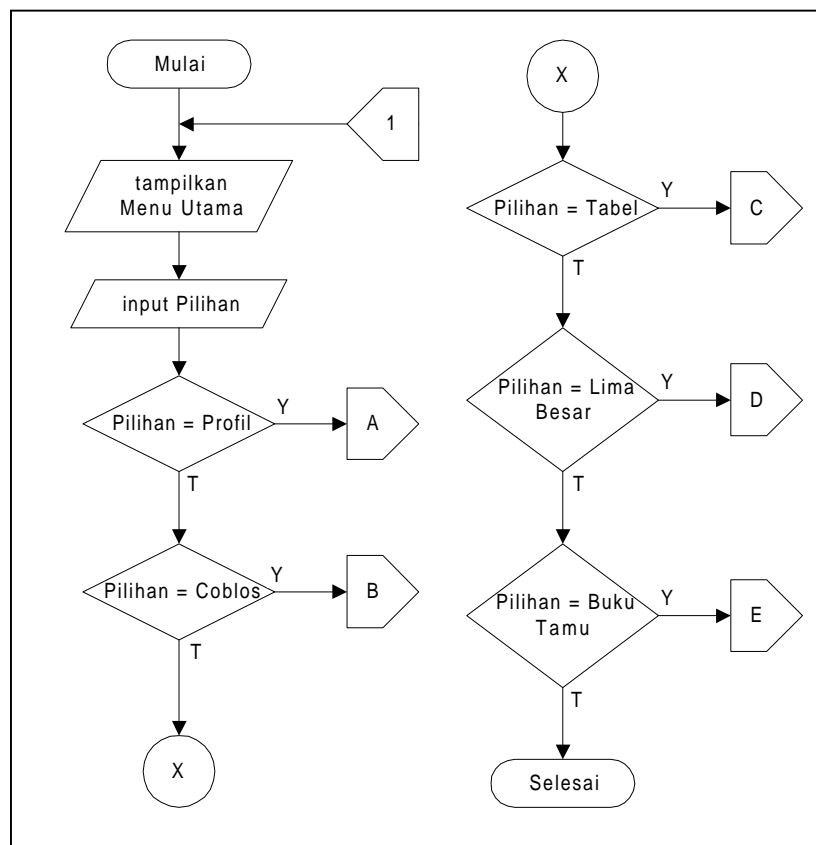
Tabel ini menggunakan index `idx_nama` berdasarkan field `nama`, dan kunci utama pada field `noktp`.

### III.2.3 Alur Program

Secara umum, program mempunyai menu utama yang menampilkan pilihan 5 submenu, yaitu:

1. Profil, menampilkan profil logo-logo partai,
2. Coblos, melakukan penyoblosan logo yang disukai,
3. Tabel, melihat perolehan suara tiap logo partai,
4. 5 Besar, melihat 5 logo partai terbaik, dan
5. Buku Tamu, mengisi dan melihat tanggapan para pengunjung.

Berikut adalah alur utama yang menggambarkan ke-5 submenu tersebut:



Gambar 3.4 - Diagram alur Menu Utama

Bila pengunjung memilih "Profil", pengunjung harus memilih profil logo partai apa yang akan dilihatnya, kemudian program akan menampilkan profil logo partai yang dikehendaki. Alurnya sebagai berikut:

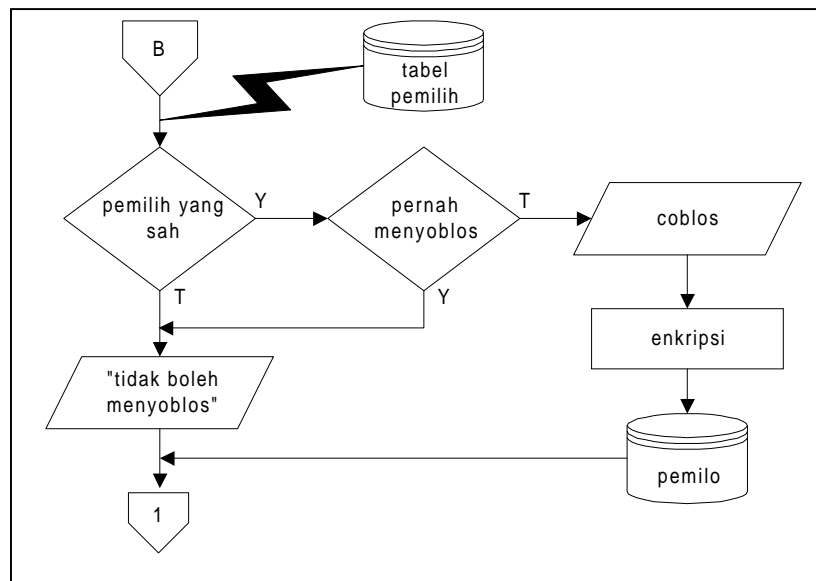


Gambar 3.5 - Diagram alur submenu Profil

Bila pengunjung memilih "Coblos", maka program akan menghubungkan dengan database eksternal (yang berisi tabel `pemilih`) untuk menguji:

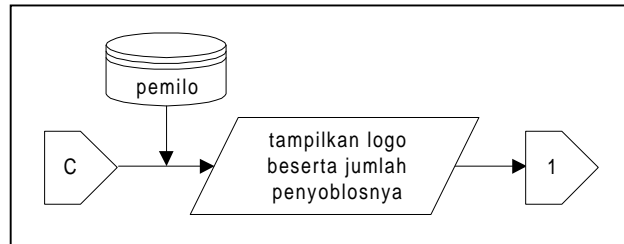
1. apakah pengunjung terdaftar sebagai pemilih yang sah
2. bila sah, apakah pengunjung sudah pernah mencoblos dalam kurun 1 (satu) minggu masa pencoblosan

Bila kedua syarat tersebut terpenuhi, proses pencoblosan berlangsung, setelah itu kembali ke menu utama. Berikut alur programnya:



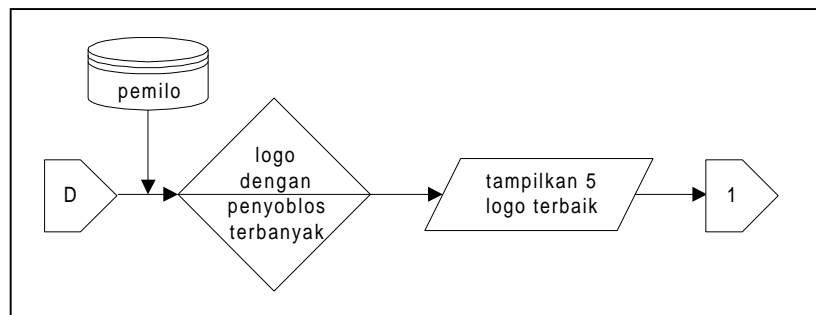
Gambar 3.6 - Diagram alur submenu Coblos

Bila pengunjung memilih "Tabel", maka akan program akan mengakses database untuk menampilkan ke-48 logo partai beserta jumlah suara yang mencoblos masing-masing logo tersebut. Alur programnya:



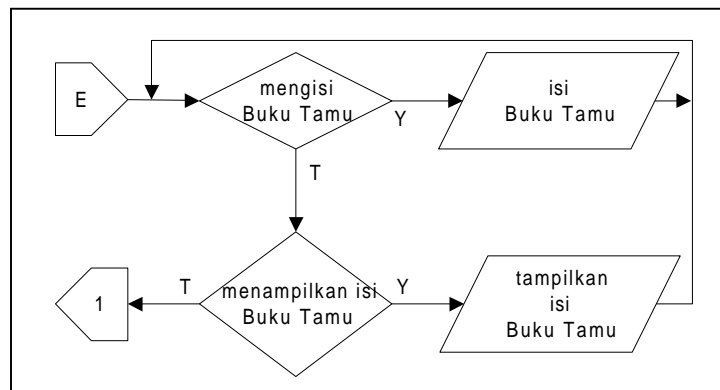
Gambar 3.7 - Diagram alur submenu Tabel

Bila pengunjung memilih "5 Besar", program akan mengakses database yang sudah disortir, lalu menampilkan grafik 5 logo terbaik. Bentuk alurnya sebagai berikut:



Gambar 3.8 - Diagram alur submenu 5 Besar

Bila pengunjung memilih "Buku Tamu", program akan melakukan proses mengisi Buku Tamu atau menampilkan isi Buku Tamu, sesuai dengan permintaan pengunjung. Alur programnya sebagai berikut:



Gambar 3.9 - Diagram alur submenu Buku Tamu

### III.2.4 Penjelasan Program

Kode *script* aplikasi "PEMILO 1999" selengkapnya telah dilampirkan pada Lampiran. Sub bab ini akan menjelaskan satu per satu setiap bagian utama dari program tersebut.

#### III.2.4.1 MENU UTAMA

Menu utama yang ditampilkan menggunakan dua buah *frame* sebagai *user interface*, sehingga pengunjung dapat lebih mudah melakukan navigasi dalam aplikasi ini. Adapun kode utama yang mendeklarasikan *frame* tersebut:

```

<framesetcols="119,*" framespacing="0" border="0" frameborder="0">
  <frame name="menu" scrolling="no" noresize target="utama"
    src="menu.htm">
  <frame name="utama" scrolling="yes" src="utama.htm">
  <noframes>
    <body>
      Maaf, halaman ini menggunakan frame, sedangkan browser Anda
      tidak mendukung fasilitas frame.
    </body>
  </noframes>
</frameset>

```

*Frame* yang berisi menu utama diberi nama *menu*, sedangkan yang berisi submenu bila menu yang bersangkutan dipilih bernama *utama*. Bilamana *browser* client tidak mendukung fasilitas *frame*, maka akan ditampilkan pesan "Maaf, halaman ini menggunakan frame, sedangkan browser Anda tidak mendukung fasilitas frame."

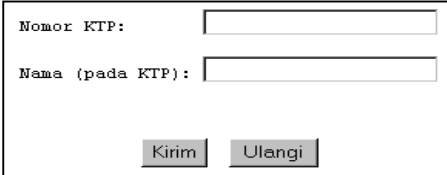
Frame menu dirancang tanpa menggunakan *scrollbar*, sedangkan *utama* bisa menampilkan *scrollbar* bila diperlukan.

#### III.2.4.2 SUBMENU PROFIL

Submenu Profil hanyalah berisi teks mengenai profil masing-masing logo partai dan tidak mengandung proses apapun, sehingga tidak dijelaskan lebih lanjut pada bagian ini.

#### III.2.4.3 SUBMENU COBLOS

Bilamana pengunjung memilih menu ini, akan ditampilkan:



The image shows a web form for verification. It contains two text input fields. The first is labeled "Nomor KTP:" and the second is labeled "Nama (pada KTP):". Below the input fields are two buttons: "Kirim" and "Ulangi".

Gambar 3.10 - Tampilan form untuk verifikasi

Setelah diisi, dan mengklik tombol "Kirim", maka data yang diberikan pengunjung akan diperiksa ke database eksternal, dengan menggunakan *script* berikut:

```
function login($id_pemilih, $nama_pemilih) {
    $db      = "pemilu";
    $SQL     = "SELECT * FROM pemilih WHERE id ='$id_pemilih'";
    $koneksi = konek_db();
    $query   = mysql_db_query($db, $SQL, $koneksi);
    $row     = mysql_fetch_array($query);
    if (($row["id"] == $id_pemilih)
        AND ($row["nama"] == $nama_pemilih)
        AND ($id_pemilih != "")) {
        SetCookie("pemilo", "$id_pemilih:$nama_pemilih", time()+60);
        $nilai = 1;
    }
    else { $nilai = 0; }
    return $nilai;
}
```

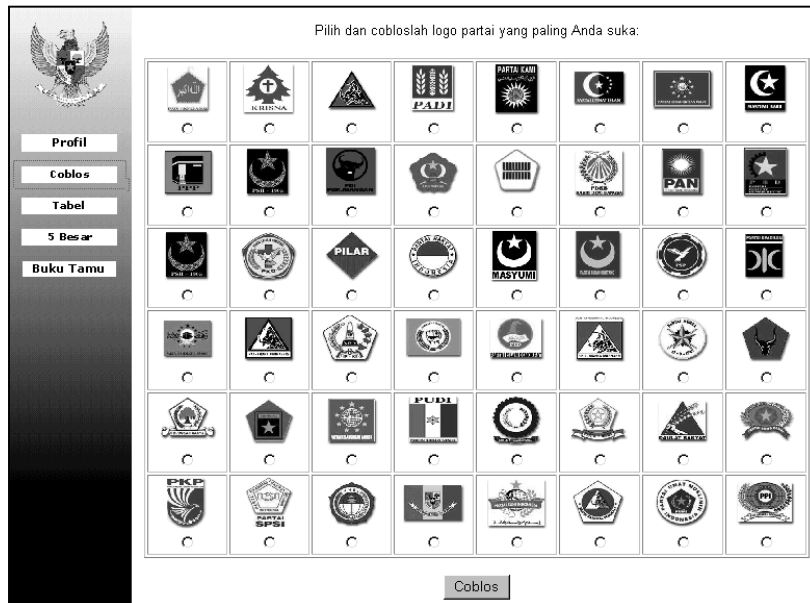
Apabila pengunjung tidak terdaftar pada tabel `pemilih`, maka akan muncul pesan "Maaf, Anda tidak terdaftar sebagai pemilih yang sah.", sedangkan bila pengunjung sudah pernah mencoblos sebelumnya (selama waktu pemilihan), akan ditampilkan pesan "Maaf, Anda hanya boleh sekali mencoblos". *Script* yang memproses pengujian ini adalah sebagai berikut:

```
function periksa($cookie) {
    $auth = explode(":", $cookie);
    $db = "pemilo";
    $koneksi = konek_db();
    $SQL = "SELECT * FROM pemilih WHERE id = '$auth[0]'";
    $query = mysql_db_query($db, $SQL, $koneksi);
    $row = mysql_fetch_array($query);
    if (($row["id"] == $auth[0])
        AND ($row["nama"] == $auth[1])
        AND ($auth[0] != "")) {
        $nilai = 1;
    }
    else {
        $nilai = 0;
    }
    return $nilai;
}
```

Bila kedua pengujian ini berhasil dilewati, maka akan *cookie* bernama `pemilo` akan ditempatkan pada *client* (*cookie* ini akan kadaluwarsa dalam waktu satu minggu), dengan perintah berikut:

```
SetCookie("pemilo", "$id_pemilih:$nama_pemilih", time()+604800);
```

Bila pengunjung adalah pemilih yang sah dan belum pernah melakukan pencoblosan sebelumnya, maka akan ditampilkan "kartu suara" untuk melakukan pencoblosan, dengan tampilan sebagai berikut:



Gambar 3.11 - Tampilan "kartu suara" PEMILO 1999

Pencoblosan dilakukan dengan cara mengklik logo yang disukai kemudian mengklik tombol "Coblos". Dengan demikian logo yang dipilih tersebut telah dicoblos. Bila pengunjung hanya mengklik tombol "Coblos" tanpa terlebih dahulu mengklik logo partai yang disukainya (atau bisa disebut "golput"), maka akan ditampilkan pesan "Maaf, sebelum mencoblos Anda harus memilih logo partai dulu.". Proses pengiriman dengan pengenkripsian dilakukan dengan *script* berikut:

```

if ($partai=="") {
    echo "Maaf, sebelum mencoblos Anda harus memilih logo partai
        dulu.\n";
    ?><br><br>Ayo <a href="coblos.htm">nyoblos</a>.<?
}
else {
    $sql="INSERT INTO pilihan (ipadd,md5(no_pilihan)) VALUES
        ('$ip','$partai)";
    $hasil=mysql_query($sql);
    echo "Terimakasih Anda telah mencoblos.\n";
    ?><br><br><a href="utama.htm">Kembali</a> ke halaman utama.<?
}

```

Setelah menyoblos dengan sukses, maka akan ditampilkan pesan "Terimakasih Anda telah mencoblos", lalu pengunjung dibawa kembali ke Menu Utama.



### III.2.4.4 SUBMENU TABEL

Penampilan Tabel dilakukan dengan cara langsung mengambil data dari database, sehingga hasil yang ditampilkan bersifat mutakhir. Proses ini dilakukan menggunakan *script* berikut:

```
<?
while ($baris = mysql_fetch_array($hasil)) {
    $permintaan="select * from pilihan where no_pilihan=".$n;
    $partai_ybs=mysql_num_rows(mysql_query($permintaan));
    $persentase=round($partai_ybs/$tot_pilihan*100);
    ?>
    <tr>
        <td width="10%" align="center"><?
            printf ("%d",$baris[0]);
        ?></td>
        <td width="70%"><?
            printf ("%s",$baris[1]);
        ?></td>
        <td width="10%" align="center"><?
            printf ("%d",$baris[2]);
        ?></td>
        <td width="10%" align="center"><?
            printf ("%d",$persentase);
        ?></td></tr>
    <?
    $n++;
}
?></table>
```

Berikut cuplikan tampilan submenu Tabel:



The screenshot shows a web application interface. On the left is a vertical sidebar menu with a logo at the top and five menu items: 'Profil', 'coblos', 'Tabel', '5 Besar', and 'Buku Tamu'. The 'Tabel' item is highlighted. To the right of the sidebar is a table with four columns: 'No.', 'Nama', 'Jumlah', and '%'. The table contains 18 rows of data representing various political parties in Indonesia.

No.	Nama	Jumlah	%
1	Partai Indonesia Baru	3	4
2	Partai Kristen Nasional	24	34
3	Partai Nasional Indonesia	0	0
4	Partai Aliansi Demokrat Indonesia	1	1
5	Partai Kebangkitan Muslim Indonesia	0	0
6	Partai Ummat Islam	3	4
7	Partai Kebangkitan Umat	1	1
8	Partai Masyumi Baru	0	0
9	Partai Persatuan Pembangunan	0	0
10	Partai Syarikat Islam Indonesia	0	0
11	Partai Demokrasi Indonesia Perjuangan	4	6
12	Partai Abul Yatama	0	0
13	Partai Kebangsaan Merdeka	0	0
14	Partai Kasih Demi Bangsa	0	0
15	Partai Amanat Nasional	23	32
16	Partai Rakyat Demokratik	0	0
17	Partai Syarikat Islam Indonesia 1905	0	0
18	Partai Katolik Demokrat	0	0

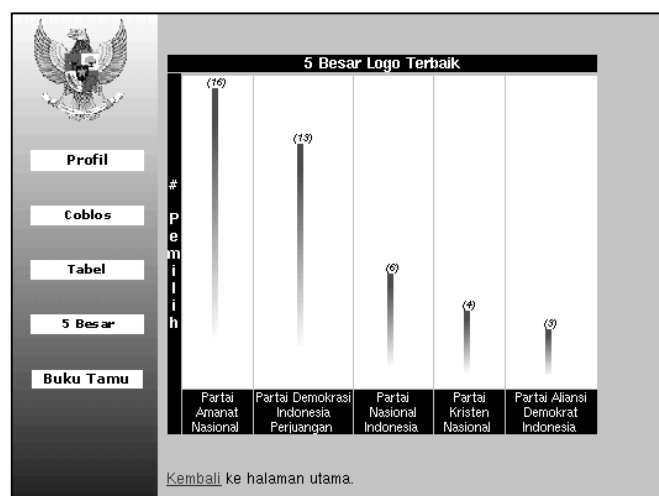
Gambar 3.12 - Tampilan submenu Tabel

### III.2.4.5 SUBMENU 5 BESAR

Untuk penyortiran, digunakan *query* SQL berikut:

```
select * from logo order by cacah desc;
```

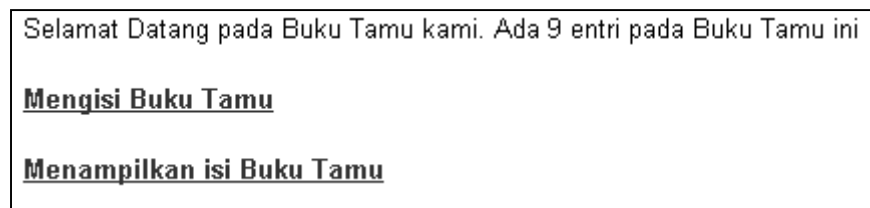
Untuk menampilkan 5 logo terbaik secara grafis, perintahnya bisa dilihat pada Lampiran. Berikut adalah tampilan submenu 5 Besar:



Gambar 3.13 - Tampilan grafis 5 logo terbaik

### III.2.4.6 SUBMENU BUKU TAMU

Bila pengunjung memilih submenu ini, maka akan tampil:



Gambar 3.14 - Tampilan pilihan pada submenu Buku Tamu

Angka 9 (sembilan) menandakan sudah ada 9 entri isi Buku Tamu, dilakukan dengan *query* SQL berikut:

```
select max(id) from buketamu;
```

Bila "Mengisi Buku Tamu" dipilih, maka akan ditampilkan:

Silakan isi Buku Tamu kami di bawah ini.

Nama Anda:

Alamat e-mail:

Alamat web:

Pekerjaan:

Alamat:

Komentar Anda:

Gambar 3.15 - Tampilan "Mengisi Buku Tamu"

Setelah diisi lengkap dan mengklik tombol "Kirim", maka entri akan ditambahkan ke database dan ditampilkan pesan dengan nama sesuai yang diisikan pada Buku Tamu. Perintah ini menggunakan *script*:

```
<?
Terimakasih, kami sangat menghargai tanggapan Anda, <?echo
    $nama;?>.
?>
```

Bila "Menampilkan isi Buku Tamu" yang dipilih, maka akan ditampilkan seluruh isi Buku Tamu yang diurutkan berdasarkan tanggal kirim:

**Isi Buku Tamu**

---

**Nama:**Boris Becker  
**E-mail:**bbecker@inet.de  
**Halaman Web:**www.borisbecker.com  
**Pekerjaan:**Pemain tenis  
**Alamat:**Pedalaman Jerman  
**Komentar:**  
 Mark, Buku Tamu-mu ini menyebarkan

---

**Nama:**James Bond  
**E-mail:**jbond@headquarters.gov  
**Halaman Web:**www.jamesbond.com  
**Pekerjaan:**Agen Rahasia & Playboy  
**Alamat:**"Rahasia"  
**Komentar:**  
 Harusnya Buku Tamu ini lebih rahasia, dimana tamu-tamunya rahasia, semua tombol-tombolnya dirahasiakan, gitu lah.

---

**Nama:**Runi Rompies  
**E-mail:**runi52@hotmail.com  
**Halaman Web:**www.runi.co.id  
**Pekerjaan:**Wartawati  
**Alamat:**Jakarta  
**Komentar:**

Gambar 3.16 - Tampilan isi Buku Tamu

Koneksi database ditutup dengan menggunakan perintah:

```
mysql_close(mysql_connect($server,$uid,$pwd));
```

## BAB IV

### PENUTUP

#### IV.1 KESIMPULAN

PHP sangat menarik bagi kalangan developer web. PHP adalah teknologi baru dalam pengembangan aplikasi berbasis web, yang dirancang dari awal untuk menyediakan kemampuan-kemampuan bagi developer web untuk membangun dokumen-dokumen HTML yang dinamis, baik itu untuk World Wide Web atau *intranet* perusahaan.

Sebenarnya PHP adalah *interpreter* (penerjemah) *sever-side script* yang bisa menerima bahasa pemrograman yang mirip C. Model pemrograman PHP berorientasi pada obyek sehingga mudah dan nyaman digunakan oleh para developer web.

Sifat PHP yang *compile-free*, membuatnya sangat fleksibel. Dengan PHP, akses database bisa dilakukan dengan mudah melalui semua interface databasenya, termasuk salah satu di dalamnya adalah ODBC. PHP tidak tergantung pada jenis *browser*, karena semua proses dilakukan di sisi *server*, dan *browser* hanya menerima HTML hasilnya. PHP juga tidak tergantung pada jenis *platform*, karena bisa dijalankan pada berbagai *platform server* web.

#### IV.2 SARAN

Memang dewasa ini telah banyak teknologi pemrograman aplikasi web yang telah dikembangkan, seperti CGI, ASP, HTMLScript, Cold Fusion, Intrabuilder, dan banyak lagi. PHP bisa jadi memiliki keunggulan yang tidak dimiliki oleh teknologi pemrograman web yang lain, terutama adalah kemudahan pemakaiannya.

Pada saat Tugas Akhir ini disusun, sudah dibuat versi baru dari PHP yaitu PHP 4.0/Zend.

Singkatnya, PHP adalah suatu solusi menawan untuk membangun aplikasi web tanpa perlu pusing kepala. PC Magazine, sebuah majalah komputer

terkemuka di dunia, pada edisi bulan Januari 1999 menobatkan PHP sebagai “teknologi pemrograman web yang sangat unggul”.

Sebagai kaum intelektual yang peduli terhadap bangsanya, penulis sangat mengharapkan berkembangnya penggunaan perangkat lunak *open source*, yang sangat cocok digunakan pada masa krisis seperti ini, dan tidak luput juga pada aplikasi-aplikasi yang sifatnya pemilihan, seperti halnya PEMILO 1999.

**DAFTAR PUSTAKA**

- Anonim, *JavaScript 1.2 Reference*, Netscape Communications, USA, 1998.
- Danesh, A., *JavaScript in a Week*, Sams Publishing, Indianapolis, USA, 1996.
- Graham, Ian S., *HTML Sourcebook 2<sup>nd</sup> Edition*, John Wiley & Sons, Inc, New York, USA, 1996.
- Gundavaram, S., *CGI Programming on the World Wide Web*, O'Reilly Associates, Sebastopol California, USA, 1996.
- Hethmon, P.S., *Illustrated Guide to HyperText Transfer Protocol (HTTP)*, Prentice Hall, New Jersey, USA, 1997.
- Kurniawan, D., *HTML 3 untuk Publikasi di Internet*, BPFE Yogyakarta, Indonesia, 1997.
- Nugroho, L.E., *Software Development in A Web-Based Environment*, Forum Teknik, Yogyakarta, Indonesia, 1997.
- Perkins, J., *Teach Yourself SQL*, Sams Publishing, Indianapolis, USA, 1996.
- Warren, M. dan Lynch, D., *Active Web Database Programming*, Sams Publishing, Indianapolis, USA, 1997.

## **REFERENSI INTERNET**

davida@detron.se, 1998, *MySQL Reference Manual*, <http://www.mysql.com>

PHP.ORG, 1999, *PHP3 Manual*, <http://www.php3.org>

Wiryana, M., 1999, *Pemilu Luber Jurdil dan Open Source*, <http://nakula.rvs.uni-bielefeld.de/made/artikel/PEMILU>



## Lampiran

### Kode Program

#### INDEX.HTML

```

*****
<html><head><title>PEMILU - Pemilihan Logo Partai</title></head>
<frameset cols="119,*" framespacing="0" border="0" frameborder="0">
  <frame name="menu" scrolling="no" noresize target="utama"
    src="menu.htm">
  <frame name="utama" scrolling="yes" src="utama.htm">
</frameset>
<body>
  Maaf, halaman ini menggunakan frame, sedangkan browser Anda
  tidak mendukung fasilitas frame.
</body>
</frameset></html>
*****

```

#### MENU.HTML

```

*****
<html><head>
<base target="utama">
<style type="text/css">
<!--
  A:link      {font-family: Verdana;
              font-weight: bold;
              font-size: 8pt;
              text-decoration: none;}
  A:visited  {font-family: Verdana;
              font-weight: bold;
              font-size: 8pt;
              text-decoration: none;}
-->
</style>
<script language="javascript">
<!--//
if (document.images) {
  profil1      = new Image;
  profil1.src  = "profil1.gif";
  profil2      = new Image;
  profil2.src  = "profil2.gif";
  coblos1      = new Image;
  coblos1.src  = "coblos1.gif";
  coblos2      = new Image;
  coblos2.src  = "coblos2.gif";
  tabel1       = new Image;
  tabel1.src   = "tabel1.gif";
}

```

```
tabel2      = new Image;
tabel2.src  = "tabel2.gif";
lbesar1    = new Image;
lbesar1.src = "lbesar1.gif";
lbesar2    = new Image;
lbesar2.src = "lbesar2.gif";
bukutamu1  = new Image;
bukutamu1.src = "bukutamu1.gif";
bukutamu2  = new Image;
bukutamu2.src = "bukutamu2.gif";
}

function GantiGambar(imgid, imgnew) {
    if (document.images) {
        document.images[imgid].src=eval(imgnew + ".src");
    }
}
//-->
</script>
</head>
<body bgcolor=#FFFFFF link=#000000 vlink=#000000
    background="blakang.jpg" leftmargin=0 rightmargin=0>
<center>

<br><br>
<a href="profil.htm"
    onMouseOver="GantiGambar('profil1','profil2'); return true"
    onMouseOut="GantiGambar('profil1','profil1'); return true"
    target="utama">
    
</a>
<br>
<a href="coblos.htm"
    onMouseOver="GantiGambar('coblos1','coblos2'); return true"
    onMouseOut="GantiGambar('coblos1','coblos1'); return true"
    target="utama">
    
</a>
<br>
<a href="tabel.php3"
    onMouseOver="GantiGambar('tabel1','tabel2'); return true"
    onMouseOut="GantiGambar('tabel1','tabel1'); return true"
    target="utama">
    
</a>
<br>
<a href="5besar.php3"
    onMouseOver="GantiGambar('lbesar1','lbesar2'); return true"
    onMouseOut="GantiGambar('lbesar1','lbesar1'); return true"
    target="utama">
    
</a>
<br>
```

```

<a href="bukutamu.php3"
  onMouseOver="GantiGambar('bukutamu1','bukutamu2'); return true"
  onMouseOut="GantiGambar('bukutamu1','bukutamu1'); return true"
  target="utama">
  
</a>
</body></html>
#*****

```

### UTAMA.HTML

```

#*****
<html><head><title>Selamat Datang</title></head><body>
<center>Selamat Datang...silakan pilih...menu di sebelah kiri..
<p align="center">
  
</p>
</body></html>
#*****

```

### DAFTAR.HTML

```

#*****
<html><body>
<form NAME="mendaftar" ACTION="daftar.php3" METHOD=POST>
  Nomor KTP:      <INPUT TYPE=text NAME=noktp><br>
  Nama (pada KTP): <INPUT TYPE=text NAME=nama><br>
  <INPUT TYPE=submit VALUE=Kirim>
  <INPUT TYPE=reset VALUE=Ulangi>
</form>
</body></html>
#*****

```

### DAFTAR.PHP3

```

#*****
<?

function konek_db() {
  $pengguna = "root";
  $server   = "localhost";
  $koneksi = mysql_connect($server, $pengguna);
  return $koneksi;
}

function login($id_pemilih, $nama_pemilih) {
  $db      = "pemilu";
  $SQL     = "SELECT * FROM pemilih WHERE id ='$id_pemilih'";
  $koneksi = konek_db();

```

```

$query    = mysql_db_query($db, $SQL, $koneksi);
$row      = mysql_fetch_array($query);
if (($row["id"] == $id_pemilih)
    AND ($row["nama"] == $nama_pemilih)
    AND ($id_pemilih != "")) {
    SetCookie("pemilo","$id_pemilih:$nama_pemilih",time()+604800);
    $nilai = 1;
}
else {
    $nilai = 0;
}
return $nilai;
}

function periksa($cookie) {
    $auth    = explode(":", $cookie);
    $db      = "pemilo";
    $koneksi = konek_db();
    $SQL     = "SELECT * FROM pemilih WHERE id = '$auth[0]'";
    $query   = mysql_db_query($db, $SQL, $koneksi);
    $row     = mysql_fetch_array($query);
    if (($row["id"] == $auth[0])
        AND ($row["nama"] == $auth[1])
        AND ($auth[0] != "")) {
        $nilai = 1;
    }
    else {
        $nilai = 0;
    }
    return $nilai;
}

if ( login($noktp, $nama) == 0) {
    echo "Maaf, Anda tidak terdaftar sebagai pemilih yang sah.";
}
elseif ( periksa($pemilo) == 1) {
    echo "Maaf, Anda hanya boleh sekali menyoblos.";
}
else {
    login($noktp,$nama);
    ?>
    <html><head><title>Pemilu</title></head>
    <body><center>
    Pilih dan cobloslah logo partai yang paling Anda suka:
    <form action="coblos.php3" method="POST">
    <table border="1" width="640" height="480">
    <tr>
    <td align="center">
    
    <br><center>
    <input type="radio" value="1" name="partai">
    </td>
    <td align="center">
    
        <br><center>
        <input type="radio" value="2" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="3" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="4" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="5" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="6" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="7" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="8" name="partai">
    </td>
</tr>
<tr>
    <td align="center">
        
        <br><center>
        <input type="radio" value="09" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="10" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="11" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="12" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="13" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="14" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="15" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="16" name="partai">
    </td>
</tr>
<tr>
    <td align="center">
        
        <br><center>
```

```
        <input type="radio" value="17" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="18" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="19" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="20" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="21" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="22" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="23" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="24" name="partai">
    </td>
</tr>
<tr>
    <td align="center">
        
```

```
        <br><center>
        <input type="radio" value="25" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="26" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="27" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="28" name="partai"></center>
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="29" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="30" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="31" name="partai">
    </td>
    <td align="center">
        
        <br><center>
        <input type="radio" value="32" name="partai">
    </td>
</tr>
<tr>
```



```
<td align="center">
  
  <br><center>
  <input type="radio" value="33" name="partai"></center>
</td>
<td align="center">
  
  <br><center>
  <input type="radio" value="34" name="partai">
</td>
<td align="center">
  
  <br><center>
  <input type="radio" value="35" name="partai">
</td>
<td align="center">
  
  <br><center>
  <input type="radio" value="36" name="partai">
</td>
<td align="center">
  
  <br><center>
  <input type="radio" value="37" name="partai">
</td>
<td align="center">
  
  <br><center>
  <input type="radio" value="38" name="partai">
</td>
<td align="center">
  
  <br><center>
  <input type="radio" value="39" name="partai">
</td>
<td align="center">
  
  <br><center>
  <input type="radio" value="40" name="partai">
</td>
```

```
</tr>
<tr>
  <td align="center">
    
    <br><center>
    <input type="radio" value="41" name="partai">
  </td>
  <td align="center">
    
    <br><center>
    <input type="radio" value="42" name="partai">
  </td>
  <td align="center">
    
    <br><center>
    <input type="radio" value="43" name="partai">
  </td>
  <td align="center">
    
    <br><center>
    <input type="radio" value="44" name="partai">
  </td>
  <td align="center">
    
    <br><center>
    <input type="radio" value="45" name="partai">
  </td>
  <td align="center">
    
    <br><center>
    <input type="radio" value="46" name="partai">
  </td>
  <td align="center">
    
    <br><center>
    <input type="radio" value="47" name="partai">
  </td>
  <td align="center">
    
  </td>

```

```

                <br><center>
                <input type="radio" value="48" name="partai">
            </td>
        </tr>
    </table><br>
    <input type="Submit" name="Submit" value="Coblos">
</form>
</body></html>
<?
}
?>
#*****

```

### COBLOS.PHP3

```

#*****
<html><body><?
$ip=getenv("REMOTE_ADDR");
mysql_connect("localhost","root");
mysql_select_db("pemilu") or die("Ada kesalahan database, hubungi Ad
min.");
if ($partai=="") {
    echo "Maaf, sebelum mencoblos Anda harus memilih logo partai dulu
    .\n";
    ?><br><br>Ayo <a href="coblos.htm">nyoblos</a>.<?
}
else {
    $sql="INSERT INTO pilihan (ipadd,no_pilihan) VALUES ('$ip','$part
    ai')";
    $hasil=mysql_query($sql);
    echo "Terimakasih Anda telah menyoblos.\n";
    ?><br><br><a href="utama.htm">Kembali</a> ke halaman utama.<?
}
?></body></html>
#*****

```

### TABEL.PHP3

```

#*****
<html><body><?
mysql_connect("localhost","root");
mysql_select_db("pemilo") or die("Database ngaco, hubungi Admin");
$tot_pilihan=mysql_num_rows(mysql_query("select * from pilihan"));
$hasil = mysql_query("SELECT * FROM logo");
$n=1;
?>
<center>
<table border="1" width="67%">
    <tr><td width="10%" align="center">No.    </td>
        <td width="70%" align="center">Nama    </td>
        <td width="10%" align="center">Jumlah </td>
        <td width="10%" align="center">%      </td>
    </tr>

```

```

<?
while ($baris = mysql_fetch_array($hasil)) {
    $permintaan="select * from pilihan where no_pilihan=".$n;
    $partai_ybs=mysql_num_rows(mysql_query($permintaan));
    $persentase=round($partai_ybs/$tot_pilihan*100);
    ?>
    <tr>
        <td width="10%" align="center"><?
            printf ("%d",$baris[0]);
        ?></td>
        <td width="70%"><?
            printf ("%s",$baris[1]);
        ?></td>
        <td width="10%" align="center"><?
            printf ("%d",$baris[2]);
        ?></td>
        <td width="10%" align="center"><?
            printf ("%d",$persentase);
        ?></td>
    </tr>
    <?
    $n++;
}
?>
</table></body></html>
#*****

```

### 5BESAR.PHP3

```

#*****
<html><body>
<?
mysql_connect("localhost","root");
mysql_select_db("pemilo") or die("Ada kesalahan database, hubungi Ad
min.");
for ($n=1 ; $n<=48 ; $n++) {
    $q1="select no_pilihan from pilihan where no_pilihan=".$n;
    $h1=mysql_num_rows(mysql_query($q1));
    $q2="update logo set cacah=$h1 where no=$n";
    mysql_query($q2);
}
$h2 = mysql_query("select * from logo order by cacah desc");
for ($n=0 ; $n<=4 ; $n++) {
    $b = mysql_fetch_row($h2);
    $nilai[$n] = $b[2];
    $nama[$n] = $b[1];
}
?>
<br><br>
<?

```

```
function grafik_html(
    $nama,
    $nilai,
    $bar,
    $harga,
    $dnilai=0,
    $dbar=0
) {
    $er = error_reporting(1);
    $harga = grafik_default_hv($harga);
    mulai_grafik($harga, $nama);
    grafik_tegaklurus($nama, $nilai, $bar, $harga);
    akhir_grafik();
    error_reporting($er);
}

function init_grafik_html() {
    $harga = array(
        "labelv"=>"",
        "labelh"=>"",
        "cellpadding"=>"",
        "cellspacing"=>"",
        "border"=>"",
        "lebar"=>"",
        "background"=>"",
        "fcolorv"=>"",
        "fcolorh"=>"",
        "bgcolorv"=>"",
        "bgcolorh"=>"",
        "fstylev"=>"",
        "fstyleh"=>"",
        "nilaitaktampil"=>"",
        "skala"=>"",
        "namabgcolor"=>"",
        "nilaibgcolor"=>"",
        "namafcolor"=>"",
        "nilaifcolor"=>"",
        "namafstyle"=>"",
        "nilaifstyle"=>"",
    );
    return($harga);
}

function mulai_grafik($harga, $nama) {
    print '<TABLE';
    print ' CELLPADDING="" . $harga["cellpadding"] . "'";
    print ' CELLSPACING="" . $harga["cellspacing"] . "'";
    print ' BORDER="" . $harga["border"] . "'";
    if ($harga["lebar"] != 0) {
        print ' lebar="" . $harga["lebar"] . "'";
    }
    if ($harga["background"]) {
        print ' BACKGROUND="" . $harga["background"] . "'";
    }
    print '>';
}
```

```

    if (($harga["labelv"]) || ($harga["labelh"])) {
        $rowspan = 3;
        $colspan = SizeOf($nama) + 1;
    }
    print '<TR><TD ALIGN=CENTER VALIGN="CENTER" ' ;
    if (! $harga["background"]) {
        print 'BGCOLOR="' . $harga["bgcolorh"] . '"';
    }
    print ' COLSPAN="' . $colspan . '">';
    print '<FONT COLOR="' . $harga["fcolorh"] . '" STYLE="' . $harga[
        "fstyleh"] . '">';
    print "<B>" . $harga["labelh"] . "</B>";
    print '</FONT></TD></TR>';
    print '<TR><TD ALIGN="CENTER" VALIGN="CENTER" ' ;
    if (! $harga["background"]) {
        print 'BGCOLOR="' . $harga["bgcolorv"] . '"';
    }
    print ' ROWSPAN="' . $rowspan . '">';
    print '<FONT COLOR="' . $harga["fcolorv"] . '" STYLE="' . $harga[
        "fstylev"] . '">';
    print "<B>" . $harga["labelv"] . "</B>";
    print '</FONT></TD>';
}

function akhir_grafik() {
    print "</TABLE>";
}

function grafik_default_hv($harga) {
    if (! $harga["fcolorv"]) {
        $harga["fcolorv"] = "#000000"; }
    if (! $harga["fcolorh"]) {
        $harga["fcolorh"] = "#000000"; }
    if (! $harga["bgcolorv"]) {
        $harga["bgcolorv"] = "#FFFFFF"; }
    if (! $harga["bgcolorh"]) {
        $harga["bgcolorh"] = "#FFFFFF"; }
    if (! $harga["cellpadding"]) {
        $harga["cellpadding"] = 0; }
    if (! $harga["cellspacing"]) {
        $harga["cellspacing"] = 0; }
    if (! $harga["border"]) {
        $harga["border"] = 0; }
    if (! $harga["skala"]) {
        $harga["skala"] = 1; }
    if (! $harga["namabgcolor"]) {
        $harga["namabgcolor"] = "#FFFFFF"; }
    if (! $harga["nilaibgcolor"]) {
        $harga["nilaibgcolor"] = "#FFFFFF"; }
    if (! $harga["namafcolor"]) {
        $harga["namafcolor"] = "#000000"; }
    if (! $harga["nilaifcolor"]) {
        $harga["nilaifcolor"] = "#000000"; }
    return ($harga);
}

```

```

function grafik_tegaklurus($nama, $nilai, $bar, $harga) {
    print "<TR>";
    for( $i=0;$i<SizeOf($nilai);$i++ ) {
        print '<TD ALIGN="CENTER" VALIGN="BOTTOM" ' ;
        if (! $harga["background"]) {
            print ' BGCOLOR="' . $harga["nilaibgcolor"] . "' ;
        }
        print ">";
        if (! $harga["nilaitaktampil"]) {
            print '<I><FONT SIZE="-2" COLOR="' . $harga["nilaifcolor"]
                . "' ' ;
            print ' STYLE="' . $harga["nilaifstyle"] . "'>( ' ;
            print $nilai[$i] . "></FONT></I><BR>";
        }
        ??
        <IMG SRC="<? echo $bar[$i] ?>" lebar=5 HEIGHT="<?
        if ($nilai[$i] != 0) {
            echo $nilai[$i] * $harga["skala"];
        }
        else {
            echo "1";
        }
        ?>">
        </TD>
        <?
    }
    print "</TR><TR>";
    for( $i=0;$i<SizeOf($nilai);$i++ ) {
        ??
        <TD ALIGN="CENTER" VALIGN="TOP"
        <?
        if (! $harga["background"]) {
            print ' BGCOLOR="' . $harga["namabgcolor"] . "' ;
        }
        ??
        >
        <FONT
            SIZE="-1"
            COLOR="<? echo $harga["namafcolor"] ?>"
            STYLE="<? echo $harga["namafstyle"] ?>"
        >
        <? echo $nama[$i] ?>
        </FONT>
        </TD>
        <?
    }
}

print "</TR>";
$terbesar = 10;
$bar      = array();

for( $i=0;$i<SizeOf($nilai);$i++ ) {
    $bar[$i] = "pelangi_merah.gif";
}

```

```

$nilai_grafik = init_grafik_html();
$nilai_grafik["labelh"] = "5 Logo Terbaik";
$nilai_grafik["labelv"] = "#<P>P<BR>e<BR>m<BR>i<BR>l<BR>i<BR>h";
$nilai_grafik["fcolorv"] = "#FFFFFF";
$nilai_grafik["fcolorh"] = "#FFFFFF";
$nilai_grafik["bgcolorv"] = "#000000";
$nilai_grafik["bgcolorh"] = "#000000";
$nilai_grafik["lebar"] = 350;
$nilai_grafik["cellspacing"] = "1";
$nilai_grafik["skala"] = 150 / $terbesar;
$nilai_grafik["namafcolor"] = "#FFFFFF";
$nilai_grafik["namabgcolor"] = "#000000";
grafik_html($nama, $nilai, $bar, $nilai_grafik);
?>
<br><br>
<a href="utama.htm">Kembali</a> ke halaman utama.
</body></html>
*****

```

### BUKUTAMU.PHP3

```

*****
<html><body>
<?
#$MP="/usr/lib/sendmail -t";
#$mail="ccunning";
mysql_connect("localhost","root");
mysql_select_db("pemilo")
    or die("Ada kesalahan database, hubungi Admin.");

if ($argv[0]=="tambahkan"):
    ?>
    Silakan isi Buku Tamu kami di bawah ini..<br>
    <FORM NAME="bukutamu" ACTION="<?echo $PHP_SELF?>" METHOD=POST>
        <INPUT TYPE=hidden NAME=cmd VALUE=kirim>
        Nama Anda: <INPUT TYPE=text NAME=nama><br>
        Alamat e-mail: <INPUT TYPE=text NAME=email><br>
        Alamat web: <INPUT TYPE=text NAME=url><br>
        Pekerjaan: <INPUT TYPE=text NAME=pekerjaan><br>
        Alamat: <INPUT TYPE=text NAME=alamat><br>
        Komentar Anda:
            <TEXTAREA NAME=komentar COLS=60 ROWS=6></TEXTAREA>
        <INPUT TYPE=submit VALUE=Kirim><br>
        <INPUT TYPE=reset VALUE=Ulangi>
    </FORM>
    <?
elseif ($argv[0]=="tampilkan"):
    echo "<H2>Isi Buku Tamu</H2>";
    $hasil=mysql_query("select nama, email, url, pekerjaan, alamat, k
        omentar from bukutamu");
    while ($b=mysql_fetch_row($hasil)) {
        echo "<HR>";
        echo "<B>Nama:</B>$b[0]";
        echo "<BR><B>E-mail:</B><A HREF=\"mailto:$b[1]\">$b[1]</A>";
        echo "<BR><B>Halaman Web:</B><A HREF=\"\$b[2]\">$b[2]</A>";
    }

```



```

        echo "<BR><B>Pekerjaan:</B>${b[3]}";
        echo "<BR><B>Alamat:</B>${b[4]}";
        echo "<BR><B>Komentar:</B>";
        echo "<BR>${b[5]}";
    }
elseif (isset($cmd) && $cmd=="kirim"):
#   $fd=popen ($MP,"w");
#   fputs ($fd,"Untuk: $mail\n");
#   fputs ($fd,"Judul: Guestbook  Addition\n");
#   fputs ($fd,"$nama ($email) sudah dimasukkan ke dalam Buku Tamu,
#       terimakasih.\n");
#   fputs ($fd,"Halaman Web: $url\n");
#   fputs ($fd,"Pekerjaan: $pekerjaan\n");
#   fputs ($fd,"Alamat: $alamat\n");
#   fputs ($fd,"$komentar\n");
#   pclose ($fd);
$komentar=addslashes("$komentar");
mysql_query("insert into bukutamu (
        nama,
        email,
        url,
        pekerjaan,
        alamat,
        komentar
    ) values (
        '$nama',
        '$email',
        '$url',
        '$pekerjaan',
        '$alamat',
        '$komentar'
    )
");
??
<P>Terimakasih, kami sangat menghargai tanggapan Anda, <?echo $na
ma;?>.
<br><br>
<a href="utama.htm">Kembali</a> ke halaman utama.
<?
else:
$hasil=mysql_query("select max(id) from bukutamu");
$b=mysql_fetch_row($hasil);
$banyak=$b[0];
if ($banyak=="") {
    $isi="Saat ini tidak ada entry sama sekali.";
}
elseif ($banyak=="1") {
    $isi="Saat ini baru ada sebuah entri.";
}
else {
    $isi="Saat ini ada $nomor entri.";
}
}

```

```
echo "<P>Selamat Datang pada Buku Tamu kami. Ada $banyak entri pa  
da Buku Tamu ini";  
echo "  
<H4>  
<A HREF=\"\$PHP_SELF?tambahkan\">Mengisi Buku Tamu</A>  
</H4>";  
echo "  
<H4>  
<A HREF=\"\$PHP_SELF?tampilkan\">Menampilkan isi Buku Tamu</A>  
</H4>";  
endif;  
?></body></html>  
#*****
```