

# Pengenalan Pemrograman *E-Commerce* dengan PHP dan MySQL

## **Lisensi Dokumen:**

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

**Dian Andriana**

dian@informatika.lipi.go.id

## **Abstrak**

Aplikasi perangkat lunak komputer dan Internet telah berkembang pesat pada dasawarsa ini, demikian pula dengan aplikasi web dan browser internet maupun intranet. Aplikasi *E-Commerce* telah lama berkembang diawali dengan EDI (*Electronic Data Interchange*) yang telah berkembang dalam lingkup internasional. Dalam makalah ini diuraikan mengenai arsitektur sistem, tool dan konfigurasi yang diperlukan untuk mengimplementasi aplikasi web *e-commerce*, konsiderasi masalah keamanan sistem, perancangan dari sisi diagram alur aplikasi dan perancangan basis data, serta kode program PHP yang diperlukan untuk implementasi aplikasi ini. Digunakan bahasa pemrograman PHP karena kemudahan dalam pemrograman, dan kelengkapan fitur untuk mengimplementasi sistem *e-commerce*, kemampuan untuk *cross platform*, serta kemudahan untuk *deployment* bagi pengembang aplikasi.

## **1. Pendahuluan**

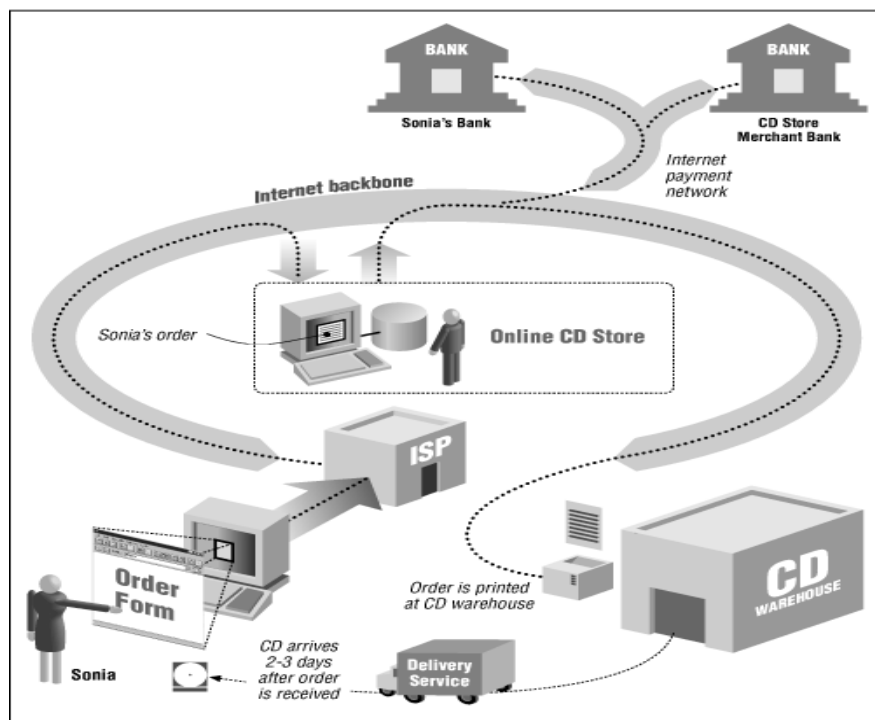
Definisi *E-Commerce* ( *Electronic Commerce*) : *E-commerce* merupakan suatu cara berbelanja atau berdagang secara online atau direct selling yang memanfaatkan fasilitas Internet dimana terdapat website yang dapat menyediakan layanan "*get and deliver*". *E-commerce* akan merubah semua kegiatan marketing dan juga sekaligus memangkas biaya-biaya operasional untuk kegiatan trading (perdagangan) .

Proses yang ada dalam *E-commerce* adalah sebagai berikut :

- Presentasi electronis (Pembuatan Web site) untuk produk dan layanan.
- Pemesanan secara langsung dan tersedianya tagihan.
- Otomasi account Pelanggan secara aman (baik nomor rekening maupun nomor Kartu Kredit).
- Pembayaran yang dilakukan secara Langsung (online) dan penanganan transaksi

Keuntungan yang diperoleh dengan menggunakan transaksi melalui *E-commerce* bagi suatu perusahaan adalah sebagai berikut :

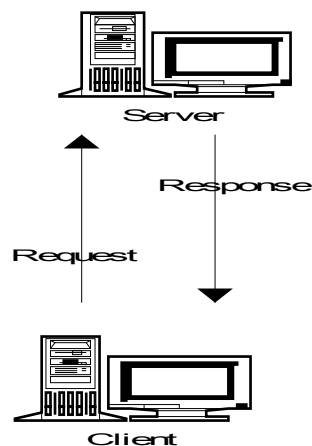
- Meningkatkan pendapatan dengan menggunakan online channel yang biayanya lebih murah.
- Mengurangi biaya-biaya yang berhubungan dengan kertas, seperti biaya pos surat, pencetakan, report, dan sebagainya.
- Mengurangi keterlambatan dengan menggunakan transfer elektronik / pembayaran yang tepat waktu dan dapat langsung dicek.
- Mempercepat pelayanan ke pelanggan, dan pelayanan lebih responsif.



Gambar 1. Contoh Aplikasi E-Commerce : Pembelian CD dengan Kartu Kredit

## 2. Arsitektur dan Konfigurasi Sistem

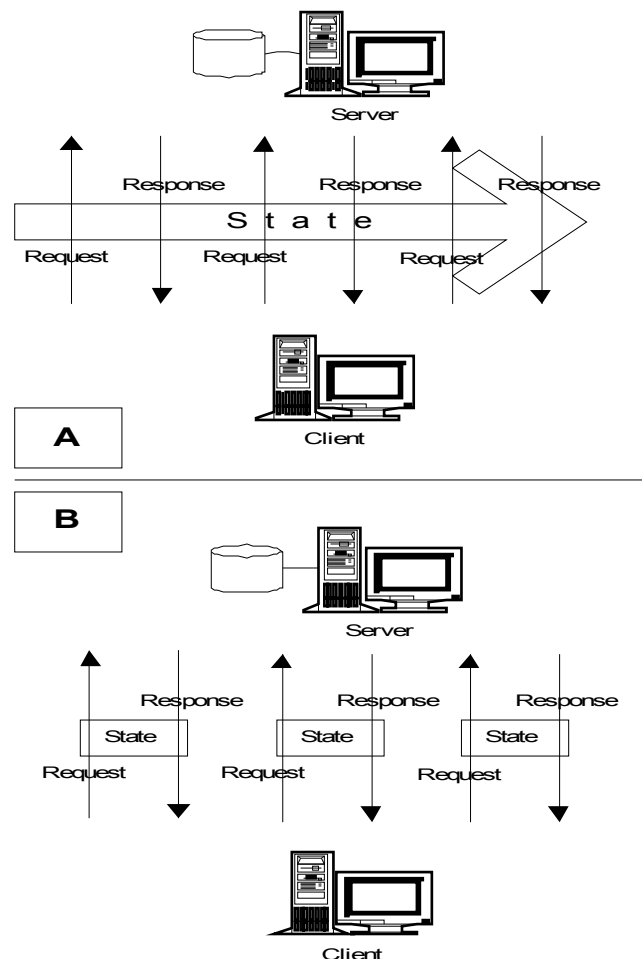
Arsitektur dasar dari aplikasi web ini adalah arsitektur *client/server*. Artinya pemrosesan aplikasi ini dijalankan melibatkan kedua sisi yakni sisi mesin server pusat dan sisi *client*. Hal ini berbeda dengan misalnya aplikasi Microsoft Word yang hanya melibatkan satu sisi saja yaitu sisi *client*. Atau bagi pengguna mesin VAX yang hanya menggunakan sisi server saja sedangkan sisi *client* hanya *dumb terminal* saja yang tidak melakukan pemrosesan apapun di sisi *client*.



Gambar 2. Arsitektur *Client/Server*

## 2.1 Stateless Web Server

Untuk aplikasi *E-Commerce* ini *web server* harus dapat mengingat siapa / identitas pengguna yang sedang melakukan browsing setiap halaman. Pada dasarnya aplikasi web dan protokol HTTP adalah *stateless*. Artinya setiap merespon sebuah *request* HTTP, server akan selesai bekerja (*complete*) dan tidak melakukan pencatatan apa yang telah dilakukan oleh pengguna sebelumnya dan terhadap siapa identitas pengguna. Server memperlakukan informasi permintaan (*request*) secara serial, satu persatu pada saat *request* masuk. Tidak ada koneksi permanen (*persistence*) yang berjalan setelah sebuah halaman telah selesai dilayani / dikerjakan.



Gambar 3. Perbandingan State dalam Sistem Aplikasi: A.State yang kontinyu dalam aplikasi *desktop*, dan B.Stateless protokol dalam aplikasi web

Agar sebuah situs web mempunyai memori / *state*, dalam hal ini aplikasi ini mampu mengingat 'siapa memesan apa', beberapa informasi yang mengidentifikasi pengguna harus dikirim dengan setiap *request* halaman web. Informasi tersebut disimpan dengan menggunakan *session*.

Session tersebut dipergunakan untuk merekam / *tracking* aktivitas pengguna yang melalui sejumlah halaman pada website, misalnya pada jenis aplikasi *Shopping Cart* (kereta belanja). Direkam pula informasi identitas pengguna yang memiliki kereta belanja tersebut.

Dengan PHP, untuk penggunaan session ini mula-mula dilakukan pengaturan pada file `php.ini` yang menunjukkan *session* dimulai (*start*). Dengan ini PHP akan membuat suatu identifier unik dan file yang berkaitan, yang disimpan di server (lokasinya di atur di `php.ini` dan nilai *default*nya di direktori `/tmp`). Kemudian pada saat pengguna berkunjung pada halaman-halaman situs web, semua informasi variabel yang dipilih oleh pengguna akan disimpan dalam file pada server, dan semua script yang dibutuhkan untuk melacak sebagai identifier unik.

Implementasi *session* dapat mempergunakan *cookie* yang disimpan pada sisi *client*, atau dipropagasikan melalui alamat URL.

Untuk penggunaan *cookie*, yaitu dengan *passing* variabel melalui *cookie* yang menyimpan informasi semua elemen barang belanja dan harganya. Namun hal ini memiliki keterbatasan yaitu dari ([http://www.netscape.com/newsref/std/cookie\\_spec.html](http://www.netscape.com/newsref/std/cookie_spec.html)) mengenai spesifikasi *cookie* yang hanya mengijinkan 20 *cookie* per domain dan berukuran hanya 4 bytes per *cookie*. Cara lain adalah dengan memberi identitas / *identifier* unik pada masing-masing pengguna, suatu nilai unik yang mengidentifikasi siapa pengguna tersebut. Sehingga pada saat pengguna menambahkan satu item pada kereta belanja, informasi yang berkaitan dengan identifier unik tadi disimpan di komputer server. Jika menggunakan *cookie* untuk fungsi penyimpanan informasi tadi, diperlukan membuat *string* unik yang akan diletakkan dalam *cookie*, dalam direktori di server akan terdapat sebuah file yang memiliki nama yang sama sebagai ID pengguna yang unik. Dalam file tersebut dapat disimpan semua variabel yang berkaitan dengan pengguna. Contohnya terdapat *array* berisi item-item barang yang ditambahkan oleh seorang pengguna ke dalam kereta belanjanya.

Terdapat keterbatasan penggunaan *cookie*, yakni bila browser pengguna di atur untuk menolak (*reject*) *cookie*.

Metode lain yang dapat digunakan adalah dengan propagasi URL, yaitu dengan mengaktifkan *flag* `-enable-trans-sid` dalam konfigurasi PHP, hal ini berguna agar *session id* akan secara otomatis ditambahkan ke setiap *relative link* pada halaman-halaman web setiap kali *session* telah dimulai.

## 2.2 Konfigurasi Sistem dan Tool Yang Digunakan

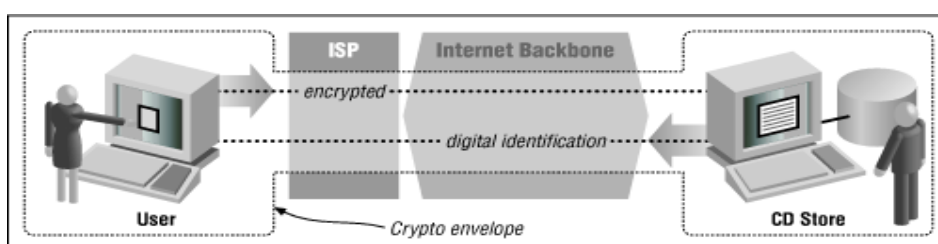
Masalah lain dalam aplikasi ini adalah mengenai aspek keamanan dalam memperoleh informasi dari pengguna, terutama data mengenai penggunaan kartu kredit. Informasi ini perlu diverifikasi oleh institusi yang berkualifikasi dan memerlukan pengaturan konfigurasi serta penggunaan beberapa macam tool.

Dalam membangun aplikasi ini dipergunakan algoritma untuk memelihara (*maintain*) *state*, pengambilan informasi secara *secure* terhadap kartu kredit, menggunakan kode pemrograman khusus dan penggunaan opsi instalasi khusus.

Di bawah ini akan dibahas mengenai teori dasar enkripsi dan sekuriti Web. Kemudian akan dibahas tool *mandatory* untuk instalasi web server Apache.

### 2.2.1 Enkripsi Public-Key / Private-Key

Mesin di web menggunakan skema keamanan Public-key/Private-key. Artinya komputer yang akan berkomunikasi menggunakan data terenkripsi harus memiliki dua buah kunci untuk mengenkripsi data dan mendekripsinya. Pertama, public-key tersedia bagi siapa saja yang ingin melakukan komunikasi terhadapnya. Sehingga siapapun yang ingin melakukan komunikasi terhadap sebuah mesin secara *secure* akan memiliki salinan dari Public key mesin tersebut. Namun public key ini tidak cukup untuk dapat mendekripsi data, masih dibutuhkan Private key yang bersifat rahasia. Misalnya pada pemrosesan kartu kredit dengan sebuah bank, nasabah memiliki Public key bank tersebut dimana ia dapat melakukan dekripsi informasi, namun masih diperlukan Private key yang disimpan oleh bank tersebut, untuk dapat melakukan dekripsi data.



Gambar 4. Pengiriman Data Terenkripsi antara Pengguna dengan Server E-Commerce

### 2.2.2 Sertifikat

Meski masalah keamanan sudah ditangani dengan keberadaan Public key / Private key, masih ada masalah yang perlu diperhatikan yakni pesan / data yang diperoleh adalah benar dari pihak yang memiliki otorisasi, bukan dari pihak lain yang tidak berkepentingan atau yang menyalahgunakan. Untuk itu dibutuhkan pihak ketiga untuk memverifikasi pesan yang datang.

Pesan terenkripsi yang dikirim dan diterima akan memiliki semacam '*signature*', dan verifikasi selanjutnya dilakukan terhadap '*signature*' tersebut. Untuk itu, organisasi yang akan mempergunakan komunikasi melalui web memerlukan kerjasama dengan organisasi lain yang mengeluarkan sertifikat yang memverifikasi pengirim pesan. Organisasi ini pulalah yang memberikan Publik key dan Private key. Salah satu contoh organisasi yang menerbitkan sertifikat sekuriti adalah VeriSign.

### 2.2.3 Secure Protocol

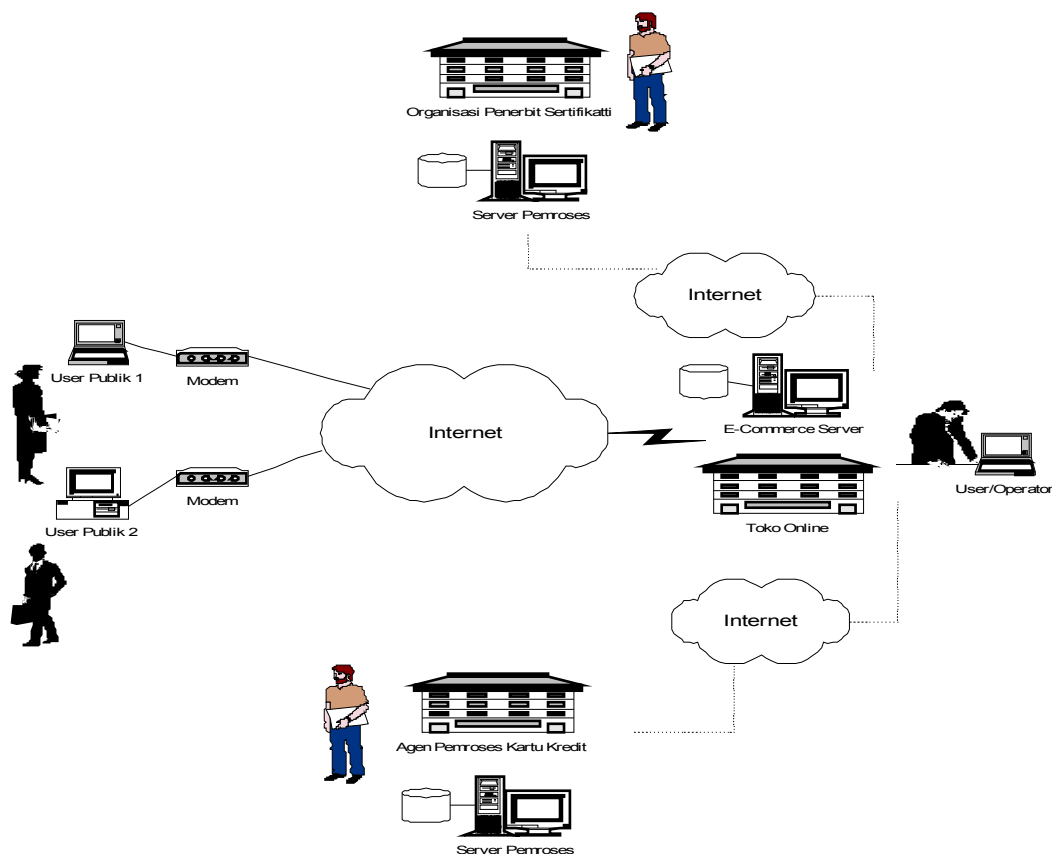
Protokol HTTP secara alamiah bersifat terbuka terhadap penyusupan. Paket-paket data yang melintas melalui *router* Internet dapat disadap dan dibaca. Namun informasi kartu kredit diinginkan agar tidak mudah terbaca. Untuk itu dibutuhkan penggunaan *Secure Socket Layer* atau SSL. SSL adalah protokol tambahan dimana *key* dan sertifikat dari suatu situs *e-commerce* akan ditransfer ke browser atau ke server lain. Melalui SSL, browser akan dapat memverifikasi sertifikat dari situs tersebut sehingga dapat mengetahui identitas pengirim sebenarnya. Tata cara enkripsi ini masih mengandung kelemahan yakni pada aspek sumber daya manusia apabila kurang jujur, yakni apabila terjadi akses tidak sah dilakukan oleh orang yang sudah berada dalam sistem.

## 2.2.4 Enkripsi dan Tool Sekuriti

Untuk web server Apache, ditambahkan modul SSL pada saat instalasinya.

Untuk dapat melakukan otorisasi kartu kredit, diperlukan sertifikat. Contoh yang paling sering digunakan adalah VeriSign, yang memiliki layanan PayfloPro.

Setelah Apache dikonfigurasi dengan SSL, maka website aplikasi dapat berkomunikasi dengan browser secara *secure*. Cirinya: URL dimulai dengan `https://`, browser akan mencari Port 443 dan mencari sertifikat. Dalam PHP, banyak fitur yang dapat digunakan untuk dapat berhubungan dengan situs lain. Misalnya fungsi `fopen()`. Namun fungsi-fungsi berhubungan dengan filesystem atau URL tidak mendukung bekerja dengan SSL, sehingga diperlukan kumpulan fungsi khusus atau program diluar PHP. Opsi-opsi dalam PHP4 dapat mendukung layanan proses pembayaran.



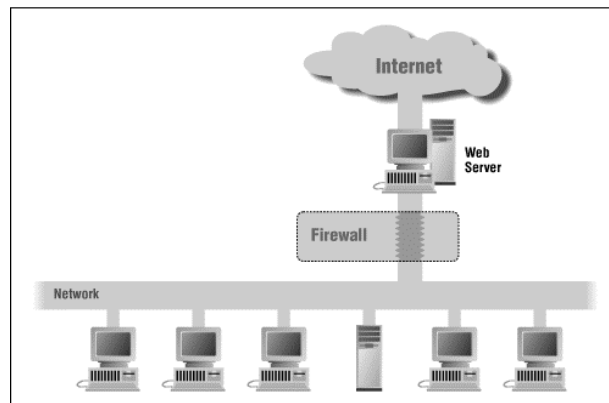
Gambar 5. Komunikasi Antar Situs dalam Aplikasi E-Commerce

## 2.2.5 Penggunaan Firewall

*Firewall* digunakan untuk melindungi jaringan lokal dari serangan luar. Ada beberapa pilihan untuk menempatkan web server :

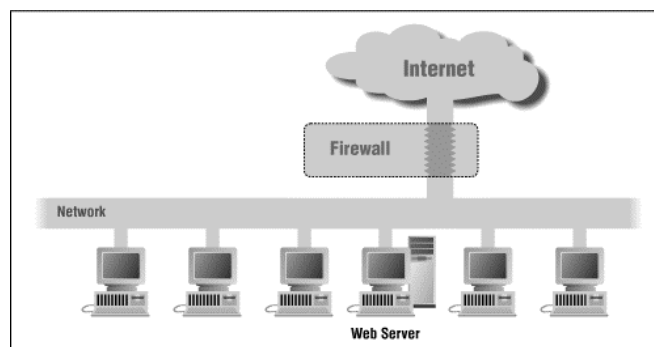
- web server ditempatkan di luar dari *Firewall* (lihat gambar 6), adapun keuntungan dengan menempatkan server diluar dari firewall adalah bahwa web server

mungkin saja menjadi subject penyerangan dari pihak luar; maka mereka "*sniffer*" tidak akan dapat meningkatkan serangan berikutnya untuk merusak server-server lainnya. Dengan kata lain web server tidak akan dapat keuntungan dari segala macam bentuk pelindungan yang di usahakan *firewall*.



Gambar 6. Web Server di Luar *Firewall*

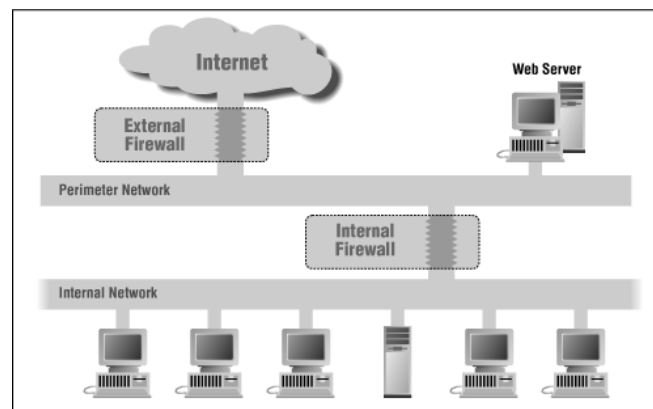
- Web server di dalam *firewall* (lihat gambar 7). Jika diterapkan seperti ini, perlu dikonfigurasi *firewall* menjadi akan melewati transaksi pada TCP port 80, atau dengan membolehkan secara langsung melewati paket maupun dengan menggunakan mekanisme proxy. Keuntungan dari menempatkan web server di dalam *firewall* yaitu *firewall* akan memblok akses dari luar yang menggunakan layanan Internet lainnya, seperti Telnet, FTP. Tetapi apabila penyusup "*sniffer*" tersebut menggunakan kesalahan dari program CGI script, mereka akan mempunyai akses tak terbatas ke jaringan lokal.



Gambar 7. Web Server yang Diletakkan Di dalam *Firewall*

- Pilihan ketiga, yang paling baik, yaitu menggunakan dua *firewall*: satu untuk melindungi jaringan internal / lokal dan yang satunya lagi untuk melindungi web server (lihat gambar 8).





Gambar 8. Webserver yang Diletakkan di Antara *Internal Firewall* dan *External Firewall*

### 2.2.6 PayFloPro dan Cybercash

Untuk penggunaan VeriSign untuk pemrosesan kartu kredit, diperlukan instalasi pustaka kode ( *code library*) yang diperoleh dari VeriSign. Selanjutnya dikompilasi ulang dengan PHP sehingga akan dapat mengenali fungsi-fungsi tersebut (fungsi-fungsi pfpro) . Fungsi-fungsi tersebut akan memproses permintaan (*request*) dan akan mengembalikan jawaban (*response*). Setelah itu *response* tersebut akan dibandingkan dengan kode yang telah diketahui, setelah itu akan diketahui apakah transaksi tersebut sukses atau tidak.

Selain PayFloPro, dapat juga digunakan Cybercash, yang diinstal sebagai pustaka / *library* pada PHP (fungsi-fungsi cybercash).

### 2.2.7 CURL

Merupakan akronim dari fungsi-fungsi pustaka Client URL. Kode pustaka ini dipergunakan untuk berkomunikasi melalui Internet menggunakan sembarang protokol di sisi lawan. Kode ini mendukung Gopher, Telnet, dan HTTPS. Untuk dapat menggunakan fungsi ini, konfigurasi instalasi PHP harus menyertakan flag `-with-curl`.

## 2.3 Pemrosesan Kartu Kredit

Pemrosesan kartu kredit dilakukan oleh perusahaan yang khusus untuk itu, terdapat beberapa nama perusahaan yang cukup dikenal, namun semuanya memiliki kesamaan cara kerja. Mula-mula dikirim permintaan / *request* dengan informasi kartu kredit: nomor, tanggal kadaluarsa, alamat, dan sebagainya, dan kemudian perusahaan tersebut akan mengirimkan kode kembalian sebagai respon. Kode PHP di sini adalah berfungsi untuk membandingkan kode yang diterima dengan nilai yang didapat sebelumnya dari agen pemrosesan. Untuk aplikasi ini dipergunakan Authorizenet.com sebagai pemroses kartu kredit.

## 3. Rancangan Aplikasi *E-Commerce*

Gambaran aplikasi *e-commerce* akan diuraikan sebagai berikut. Mula-mula aplikasi akan menampilkan daftar barang yang tersedia. Lalu pengguna dapat memilih beberapa item yang ingin dibeli. Pada saat pengguna memilih suatu item barang, identitas barang tersebut dicatat,

dan selanjutnya user dapat melanjutkan berbelanja / memilih item yang lain. Server mengingat item apa saja yang telah dipesan. Pada saat pengguna melanjutkan browsing, server memelihara *track* pengguna tersebut dan pengguna tersebut dapat melakukan *checkout* terhadap item-item yang telah dipesan.

Untuk dapat melaksanakan hal ini, digunakan metode untuk memelihara *state* seperti yang telah dibahas di bagian sebelum ini.

### 3.1 Rancangan Layar

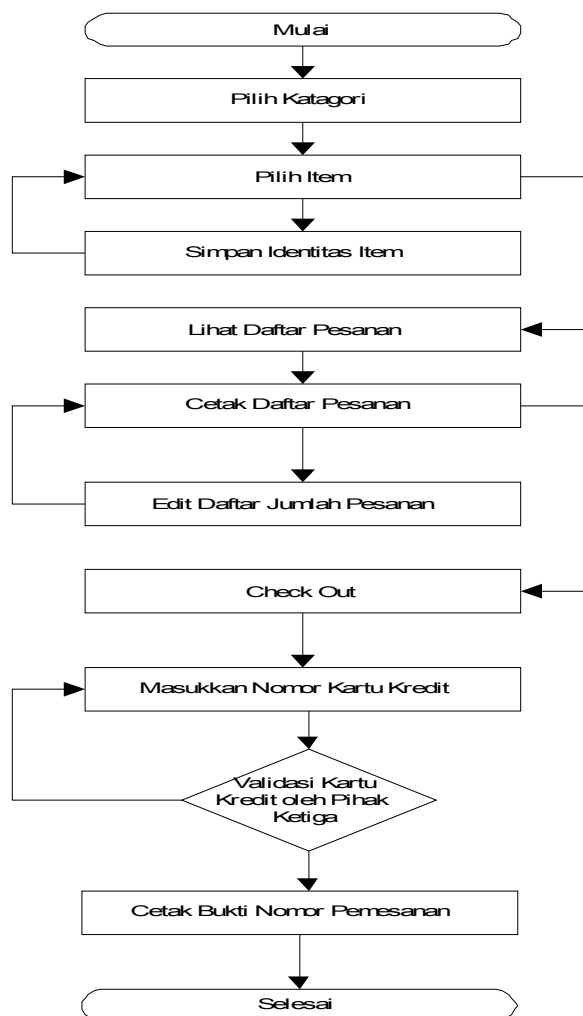
Setiap halaman pada aplikasi ini memiliki tombol yang memungkinkan pengguna untuk langsung melakukan *checkout*. Pada halaman yang menampilkan daftar barang terdapat kumpulan form yang mungkin pengguna untuk memberi indikasi item mana yang akan dibeli. Setiap item dapat ditentukan secara lebih spesifik sesuai jenis barang yang ada, misalnya untuk aplikasi toko *furniture online* terlebih dahulu ditentukan jenis *furniture* meja, terdiri atas meja bulat, meja kotak, meja tulis, dan sebagainya. Form untuk pemesanan menggunakan kotak teks untuk jumlah pesanan, dan tombol 'Order', yang pada contoh meja tadi dicantumkan untuk masing-masing jenis meja.

Selanjutnya ditampilkan satu halaman yang berisi daftar semua item yang sedang berada dalam kereta belanja (*shopping cart*). Halaman ini memungkinkan pengguna untuk menambah atau mengurangi jumlah item yang dipesan, dan menghapus suatu item pesanan.

Pada akhir proses pemesanan, ditampilkan halaman yang mengumpulkan informasi pengguna, preferensi, dan halaman untuk memulai pemrosesan kartu kredit. Halaman ini juga menunjukkan pesan bila ada kesalahan informasi atau terdapat penolakan otorisasi kartu kredit oleh agen pemroses.

Selanjutnya, setelah transaksi selesai diproses, terdapat tanda terima transaksi yang mengkonfirmasi pesanan dan menyampaikan nomor id pesanan kepada pengguna.

Yang penting diperhatikan untuk pengembangan aplikasi e-commerce adalah informasi nomor kartu kredit dan informasi personal lainnya harus aman dan tidak mudah dilihat oleh orang yang tidak berhak.



Gambar 9. Diagram Alir Penggunaan Aplikasi *E-Commerce*



Gambar 10. Rancangan Layar Halaman Depan Contoh Web *E-Commerce*

Toko Furniture Online : Meja				
Nama / Kode	Gambar	Harga	Stok	Pemesanan
Meja Telepon Kotak MK2505		Rp.225.000,-	Ada	Jumlah : <input type="text"/> <input type="button" value="Pesan"/>
Meja Telepon Bulat MB2303		Rp.215.000,-	Kosong	Jumlah : <input type="text"/> <input type="button" value="Pesan"/>
Meja Tulis Besar MT5089		Rp.415.000,-	Ada	Jumlah : <input type="text"/> <input type="button" value="Pesan"/>

Gambar 11. Rancangan Layar Halaman Contoh Daftar Barang

**Toko Furniture Online : Daftar Belanja**

Item	Jumlah	Harga	Total	Hapus ?
Meja Telepon Kotak MK2505	<input type="text" value="1"/>	Rp.225.000,-	Rp.225.000,-	<input type="checkbox"/> Hapus
<b>Grand Total</b>	1		<b>Rp.225.000,-</b>	

Gambar 12. Rancangan Layar Halaman Contoh Daftar Pesanan

**Informasi Pengiriman**

Metode Pengiriman	Biaya
<input checked="" type="radio"/> Pos	Rp.100.000,-
<input type="radio"/> Titipan Kilat (TIKI)	Rp.200.000,-
<input type="radio"/> DHL	Rp.300.000,-

**Informasi Kartu Kredit**

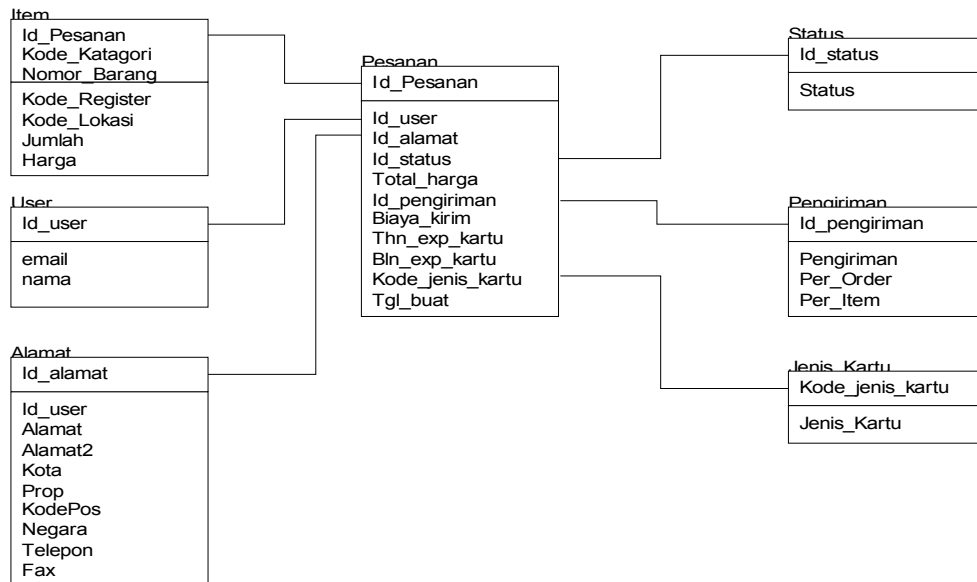
**Kartu Kredit :** ☒ Visa ☐ Master Card

**Nomor :**

**Expires:**

Gambar 13. Rancangan Layar Halaman Contoh Pemrosesan Kartu Kredit

### 3.2 Rancangan Basis Data



Gambar 14. Rancangan Basis Data Aplikasi Web *E-Commerce*

Tabel utama yang digunakan dalam aplikasi ini adalah tabel Pesanan, yang mencatat order pemesanan. Tabel-tabel lain berelasi dengan tabel ini. Tabel Pesanan menyimpan informasi id\_user, id\_alamat, dan semua informasi yang dibutuhkan untuk pembayaran. Tabel Pesanan ini memiliki relasi *one-to-many* dengan tabel Item yang berisi informasi item-item barang yang terdapat dalam sebuah order. Informasi mengenai pengguna dan alamat pengguna dipisahkan, seorang pengguna dapat diidentifikasi dari alamat *e-mail* yang dimilikinya, dan alamat terdiri atas alamat kantor dan rumah. Tabel Pengiriman berisi informasi opsi yang diberikan untuk pengiriman barang. Misalnya terdapat pilihan menggunakan UPS, DHL, TIKI, Pos, dan sebagainya. Tabel Status berisi catatan mengenai status pesanan, yaitu dapat berupa status dikembalikan (*backordered*), dikirim (*shipped*), atau dibatalkan (*cancelled*). Terdapat pula tabel Jenis\_Kartu untuk menyimpan informasi jenis kartu kredit seperti Visa, Mastercard, dan sebagainya.

Tabel 1. Tabel Alamat

Field	Jenis	Kosong	Ekstra	Keterangan
Id_Alamat	Int(11)	Tidak	Auto_increment	PRIMARY KEY
Id User	Varchar(40)	Tidak		KEY
Alamat	Varchar(40)	Ya		
Alamat2	Varchar(40)	Ya		
Kota	Varchar(40)	Ya		
Prop	Varchar(40)	Ya		
KodePos	Varchar(10)	Ya		
Negara	Varchar(20)	Ya		
Telepon	Varchar(20)	Ya		
Fax	Varchar(20)	Ya		

Tabel 2. Tabel Jenis Kartu

Field	Jenis	Kosong	Ekstra	Keterangan
Kode_Jenis_Kartu	Char(3)	Tidak		
Jenis_Kartu	Varchar(30)	Tidak		

Tabel 3. Tabel Pesanan

Field	Jenis	Kosong	Default	Ekstra	Keterangan
Id_pesanan	double	Tidak		Auto_increment	PRIMARY KEY
Id_user	Int (11)	Tidak			KEY
Id_alamat	Int (11)	Tidak			
Id_status	Tinyint (4)	Tidak			
Total_harga	Double	Tidak	'0.00'		
Id_pengiriman	Tinyint (4)	Tidak			
Biaya_kirim	Double	Tidak	'0.00'		
Tahun_exp_kartu	Int (11)	Tidak			
Bulan_exp_kartu	Tinyint (4)	Tidak			
Kode_jenis_kartu	Char(3)	Tidak			
Tgl_buat	Timestamp (14)	Ya			

Tabel 4. Tabel Pengiriman

Field	Jenis	Kosong	Default	Ekstra	Keterangan
Id_Pengiriman	Tinyint(4)	Tidak		Auto_increment	PRIMARY KEY
Pengiriman	Varchar (20)	Tidak			
Per_Order	Double	Tidak	'0.00'		
Per_Item	Double	Tidak	'0.00'		

Tabel 5. Tabel Status

Field	Jenis	Kosong	Ekstra	Keterangan
Id_Status	Tinyint(4)	Tidak	Auto_increment	PRIMARY KEY
Status	Varchar(20)	Tidak		

Tabel 6. Tabel User

Field	Jenis	Kosong	Ekstra	Keterangan
Id_User	int(11)	Tidak	Auto_increment	PRIMARY KEY
email	Varchar(255)	Tidak		UNIQUE
nama	Varchar(80)	Tidak		

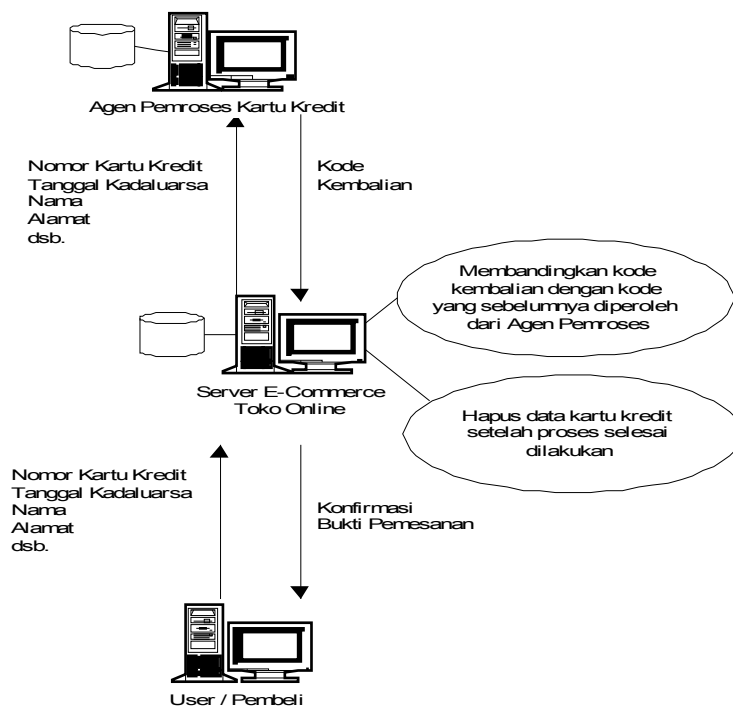
Tabel 7. Tabel Item

Field	Jenis	Kosong	Keterangan
Id_pesanan	double	Tidak	PRIMARY KEY
Kode_Katagori	Tinyint(4)	Tidak	PRIMARY KEY
Nomor_Barang	Double	Tidak	PRIMARY KEY
Kode_Register	Varchar(20)	Ya	
Kode_Lokasi	Varchar(20)	Ya	
Jumlah	Int(11)	Tidak	
Harga	Double	Tidak	

### 3.3 Keamanan Nomor Kartu Kredit dalam Basis Data

Nomor kartu kredit tidak disimpan di dalam basis data, karena dapat mengurangi keamanan. Perangkat komputer dapat dinilai sebagai suatu alat yang tidak aman, karena dapat diakses oleh orang lain. Bila digunakan ISP (Internet Service Provider) untuk *hosting* web site ini, tentunya akan sangat diragukan keamanannya, karena ISP menggunakan *shared server* yang digunakan secara bersama, meskipun telah menggunakan konfigurasi *secure server* yang ditawarkan oleh ISP tersebut. Nomor kartu kredit yang tersimpan dalam basis data akan dapat diakses oleh orang lain yang mempunyai akses terhadap mesin server. Juga tidak ada kebutuhan untuk menyimpan nomor kartu kredit dalam bentuk apapun. Yang diperlukan oleh aplikasi ini adalah memvalidasi nomor tersebut saja, dan segera menghapusnya dari memori setelah selesai. Jika nomor kartu kredit ini tidak pernah dituliskan ke hardisk, akan menjadi sangat aman karena akan terhindar dari kemungkinan pencurian oleh yang tidak berhak.





Gambar 15. Komunikasi Antar Situs dan Penghapusan Informasi Kartu Setelah Proses Selesai

## 4. Kode Program

Pemrograman untuk aplikasi *e-commerce* ini diimplementasikan dengan PHP. Fungsi-fungsi utama yang dipergunakan dalam kode program di sini adalah fungsi yang berhubungan dengan *session* dan fungsi yang berkaitan dengan pustaka *cURL*.

Konsep pemrograman berorientasi objek digunakan dalam kode program ini. Digunakan sifat *inheritance*, yakni jika sebuah kelas / *class* mewarisi properti dan metode (*properties and methods*) dari kelas induk / *parent class*, ia memiliki akses terhadap semua metode dan properti dari induknya. Dan sebuah aplikasi dapat dibangun dengan memperluas / *extending* sebuah kelas berdasarkan kelas lain yang telah ada.

### 4.1 Fungsi Session

Fungsi `session_register()` digunakan untuk menyatakan memulai *session*, sekaligus didefinisikan variabel apa saja yang akan disimpan dalam *session*. Fungsi `session_register()` diletakkan pada baris pertama program, karena fungsi ini mengirim *cookies* yang merupakan salah satu tipe dari HTTP header. Jika suatu tipe header dikirim setelah teks dikirim ke browser akan mengakibatkan error.

```
<?
session_register("var1");
$var1 = "nilai1";
?>
```

Pada waktu diakses pertama kali, halaman tersebut akan memulai session. Akan dikirim *cookie* atau *session id* yang akan ditambahkan ke dalam *relative link*. Perintah *session\_register* akan memerintahkan PHP untuk melakukan pencarian variabel \$var1 pada file session. Jika ada, variabel tersebut akan tersedia / *available* secara global, atau dapat pula diakses melalui array \$HTTP\_SESSION\_VARS. Setelah halaman tersebut diproses, nilai terakhir dari variabel yang terdaftar akan dituliskan ke file session.

Fungsi-fungsi yang digunakan untuk session ini :

Session\_destroy() : fungsi untuk menonaktifkan session dan semua variabel yang berkaitan dengannya.

Session\_unregister() : fungsi untuk menghapus nilai dari suatu variabel dalam file session.

Session\_set\_save\_handler() : fungsi yang memungkinkan untuk mengatur sendiri metode penyimpanan (*storing*), pengambilan (*retrieving*), dan penulisan (*writing*) *session handler*.

Metode *session handler* yang dipilih adalah yang manajemen session berdasarkan *file-based*. Namun metode ini tidak sesuai digunakan untuk lingkungan tercluster (*clustered environment*) dimana beberapa mesin bekerja secara bersama untuk melayani satu situs, untuk lingkungan seperti ini tidak dapat menggunakan *local filesystem*.

Session\_encode(): fungsi untuk menuliskan variabel ke dalam basis data, variabel tersebut harus terlebih dahulu diubah formatnya ke dalam format yang dimengerti oleh basis data. Fungsi session\_encode berguna untuk mengubah format ini.

```
$str = session_encode(string)
```

Session\_decode() : fungsi untuk membalik proses *encoding* di atas, sehingga variabel dikembalikan kedalam representasi PHP.

## 4.2 Fungsi-fungsi cURL

Untuk aplikasi ini diperlukan komunikasi dengan layanan validasi kartu kredit, dilakukan dengan fungsi cURL. Cara kerjanya adalah mula-mula fungsi ini akan mengirim pesan yang *secure* melalui HTTPS, dan layanan yang memvalidasi kartu kredit tersebut akan mengembalikan response, yang kemudian diproses lebih lanjut dengan PHP.

Fungsi cURL yang digunakan :

Curl\_init() : fungsi ini mengembalikan nilai integer yang serupa dengan nilai *identifier* kembalian yang dikembalikan oleh mysql\_connect() atau pointer file yang dikembalikan oleh fopen(). Pada kasus seperti ini disebut dengan cURL handle, atau ch. Pada argument tunggal pada fungsi ini diberikan URL yang akan diakses.

```
Int curl_init ([string url])
```

```
$cc_company_url =  
https://secure.process.site/transact.dll?exp=foo&cardtype=bar
```

```
$ch = curl_init($cc_company_url);
```

Fungsi ini akan memulai session cURL. Panggilan pada URL ini tidak akan berfungsi hingga fungsi `curl_exec` dieksekusi.

`Curl_setopt()` : sebelum komunikasi URL dieksekusi, perlu diset salah satu opsi cURL yaitu opsi `CURLOPT_RETURNTRANSFER`. Opsi ini untuk mengembalikan hasil dari *request* https ke dalam variabel PHP.

```
Curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
```

`Curl_exec()` : fungsi ini untuk mengeksekusi transfer. Sebuah argumen digunakan yaitu berasal dari hasil kembalian fungsi `curl_init()` dan digunakan pula pengesetan opsi-opsi lain.

```
Bool curl_exec (int ch)
```

`Curl_close()`: fungsi ini menutup koneksi cURL menggunakan *curl handle* :

```
Void curl_close (int ch)
```

Kumpulan fungsi-fungsi ini yang menjalankan transaksi dan mengembalikan hasil ke dalam variabel `$data`.

```
$ch = curl_init($authorize_net_url);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
$data = curl_exec($ch);  
curl_close($ch);
```

## 5. Hasil dan Pembahasan

Prototipe ini dibangun menggunakan perangkat lunak *open source* serta memiliki karakteristik *cross-platform*. Dari hasil diskusi dengan pengguna, diperoleh masukan mengenai kebutuhan pengguna akan perangkat lunak *e-commerce*, namun spesifik pada suatu produk tertentu, serta kebutuhan untuk memperkaya fungsi-fungsi multimedia sehingga secara visual lebih menarik.

Diperlukan kerja sama dengan supplier / distributor produk komersil untuk dapat menawarkan produk-produknya melalui *e-commerce*. Juga jasa kurir diperlukan untuk dapat melakukan layanan antar kepada konsumen dengan cepat.

Bagi lembaga penelitian atau lembaga pendidikan, perlu juga memperhatikan masalah HKI dalam layanan *e-commerce* untuk produk-produk karya intelektual, misalnya untuk layanan *e-commerce* dalam situs web *digital library*. Diperlukan kerjasama dengan pemilik hak intelektual tersebut untuk menawarkan produk-produk karya intelektual tersebut secara komersil.

Dalam sebuah seminar mengenai *e-commerce* di Bandung dikemukakan bahwa faktor terpenting dalam aplikasi *e-commerce* adalah *delivery*. Adanya sistem distribusi *multi level marketing* juga diinformasikan dapat memotong jalur *delivery* supaya lebih cepat sampai ke tangan konsumen.

## 6. Kesimpulan

Pengembangan aplikasi *e-commerce* bagi sebuah perusahaan / lembaga merupakan proses yang cukup kompleks. Melibatkan beberapa organisasi / situs dalam penanganan sekuriti dan otorisasi.

Perangkat lunak aplikasi *e-commerce* dalam dunia bisnis dapat mendukung pemotongan rantai distribusi sehingga konsumen dapat memperoleh suatu produk dengan harga yang lebih murah. Jenis antarmuka web dipilih dengan pertimbangan fleksibilitas implementasi perangkat lunak ini yang dapat dilakukan di jaringan intranet maupun internet, kemudahan untuk *deployment*, serta kemampuan *cross platform*.

Dalam makalah ini telah diuraikan mengenai arsitektur sistem, tool dan konfigurasi yang diperlukan untuk mengimplementasi aplikasi web *e-commerce*, konsiderasi masalah keamanan sistem, perancangan dari sisi diagram alur aplikasi dan perancangan basis data, serta kode program PHP yang diperlukan untuk implementasi aplikasi ini.

### ***Daftar Pustaka***

1. <http://www.mysql.com>
2. <http://www.php.net>
3. Greenspan, Jay, and Bulger, Brad, "MySQL/PHP Database Application", M&T Books, Foster City CA USA, 2001.
4. Fery Soswanto, "E-Commerce dengan memanfaatkan Sistem Operasi Linux".

## Biografi Penulis



**Dian Andriana.** Lahir di Surabaya, 10 Januari 1973. Setelah dua tahun di SMUN 5 Surabaya, menamatkan SMU di SMU PGII 1 Bandung tahun 1991. Menyelesaikan program S1 di Jurusan Teknik Informatika Institut Teknologi Bandung pada tahun 1997. Saat ini bekerja sebagai peneliti pada instansi Lembaga Ilmu Pengetahuan Indonesia (LIPI), tepatnya di Pusat Penelitian Informatika. Kompetensi pada bidang Software Engineering, Database dan Web Engineering. Berpengalaman mengembangkan Sistem Informasi aplikasi database under Windows dan Linux.