



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

LANDASAN TEORI

3.1. PHP

PHP merupakan singkatan rekursif (akronim berulang) dari PHP Hypertext Preprocessor. PHP merupakan salah satu bahasa pemrograman berbasis *server* (*Server side scripting*), yang berarti PHP harus tersimpan di dalam sebuah server dan dieksekusi di dalam *server* tersebut. Saat ini PHP menjadi pilihan paling populer di kalangan pengembang *website*, karena kemudahannya (*syntax* PHP mirip bahasa pemrograman C), sifat PHP yang *free* alias gratis, banyak didukung oleh *webserver* antara lain IIS, APACHE, Lighttpd, nginx, hingga Xitami dengan konfigurasi yang relatif mudah, serta PHP mendukung berbagai akses basis data seperti : MySQL, MSSQL, MSQL, Informix, dan Oracle.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisannya kode sumber ini menjadi sumber terbuka, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP.

Dalam penggunaannya PHP membutuhkan file HTML untuk berkolaborasi dalam membentuk sebuah webpage. Penggunaan PHP dalam file HTML ditandai dengan *tag* penanda sebagai pembuka PHP yang berbentuk `<?php` dan *tag* penutup `?>`. Seperti yang sudah disebutkan di atas bahwa PHP merupakan *script* yang dieksekusi di sisi server, bukan pada sisi *client*, jadi saat *browser* merequest sebuah halaman *website*, *webserver* akan mengolah fungsi PHP yang ada di dalamnya, jika terdapat sebuah fungsi untuk mengakses *database*, maka modul PHP akan mengakses ke *database server* tempat *database* yang diminta, selanjutnya dokumen yang diminta dikembalikan dalam bentuk HTML, sehingga *source code* PHP tidak terbaca dari sisi *client*.

3.2. MYSQL

MYSQL merupakan sebuah perangkat lunak manajemen basis data atau lebih dikenal dengan istilah DBMS (*database management system*). MYSQL sendiri termasuk salah satu database populer yang digunakan saat ini. Penggunaannya sendiri cukup banyak dalam dunia nyata seperti penggunaan sebagai sebuah tempat penyimpanan, yang nantinya akan diakses oleh sebuah aplikasi (aplikasi ini dapat berupa aplikasi dari bahasa *Java*, *Visual Basic*, *C#*, *PHP*, dll). MYSQL sendiri lebih sering dijumpai bersama dengan aplikasi PHP sehingga sering disebut *dynamic duo*.

Dalam melakukan pengaksesan ke database MYSQL, Haruslah disertakan suatu fungsi sebagai penghubung aplikasi dengan database, dimana fungsi

tersebut mencakup alamat dari *database server*, *username* (pemilik *database*) dan *password* dari *user* yang dimaksud. Jika koneksi antara database dan aplikasi telah terjalin, aplikasi mulai dapat melakukan manipulasi data dalam database tersebut seperti : *insert*, *update*, *delete* data di dalam *database*, serta membaca (*read*) data untuk ditampilkan di *browser*. Proses ini sering disebut CRUD (*create*, *read*, *update*, *delete*).

3.3. CSS, Javascript, JQuery

CSS merupakan style sheet yang berfungsi untuk melakukan modifikasi pada tampilan HTML. Dengan adanya CSS, sebuah halaman *website* dapat memiliki sebuah tampilan yang menarik. Layaknya sebuah fitur word prosesor, CSS dapat mengatur ukuran huruf, desain huruf, ukuran gambar, *footer*, *header* sebuah dokumen HTML. Penulisan CSS dapat dibagi menjadi tiga kelompok anatar lain : *Inline Style Sheet*, *Embedded Style Sheet*, dan *Linked Style Sheet*. Sedangkan manfaat CSS adalah sebagai berikut:

- a. Mempersingkat penulisan tag HTML.
- b. Mempercepat *rendering*.
- c. Mudah dan cepat dalam *maintenance*.

JavaScript pertama kali dikembangkan oleh Brendan Eich dari Netscape dibawah nama *Mocha*, yang nantinya namanya diganti menjadi *LiveScript*, dan akhirnya menjadi *JavaScript*. JavaScript merupakan scripting yang berjalan pada

sisi *client* yang artinya semua eksekusi *javaScript* dilakukan pada sisi *client*, bukan pada sisi *server* seperti *PHP*. Dalam penulisannya *javaScript* dapat ditaruh bersamaan dengan file *HTML*, misalnya pada *tag body* atau *head* atau dari eksternal *resources*.

Jquery merupakan salah satu *framework javaScript* yang dirilis tahun 2006, dan menjadi *framework javaScript* paling banyak digunakan saat ini karena alasan kemudahannya. *Jquery* mendukung fitur-fitur antara lain : manipulasi *CSS*, teknik *AJAX*, *DOM*, efek animasi, *event*. Semua fitur-fitur tersebut dapat diimplementasikan pada *tag-tag HTML*.

3.4. Interaksi Manusia dengan Komputer

Interaksi manusia dan komputer merupakan disiplin ilmu yang membahas bagaimana sebuah sistem berinteraksi dengan baik dengan manusia, dengan memberikan kemudahan kepada penggunaannya berdasarkan desain dari sistem tersebut. Dalam interaksi manusia dan komputer patut diperhatikan poin-poin berikut :

a. Strive of Consistency

Kekonsistenan pada sebuah sistem yang ada, misalnya konsisten dalam urutan suatu pengerjaan yang memiliki tugas mirip satu sama lain.

b. *Cater to Universal Usability*

Kemudahan pengguna dalam memahami isi dari sistem tersebut, misalnya penggunaan singkatan yang ada atau simbol-simbol di dalam sistem harus mencerminkan aksi yang dilakukan oleh objek tersebut.

c. *Offer Information Feedback*

Untuk setiap aksi yang berhasil atau gagal dilakukan, ditampilkan sebuah pemberitahuan (*feedback*) bagi user, agar user mengetahui hasil dari aksi yang dilakukannya.

d. *Design Dialog to Yield Closure*

Pengaturan urutan aksi dari awal sampai akhir dengan urutan dengan feedback di akhir aksi.

e. *Prevent Error*

Desain dari sistem haruslah dapat mencegah pengguna melakukan kesalahan atau *error* yang fatal. Jika sampai user melakukan kesalahan, sebisa mungkin sistem tersebut dapat mengembalikan ke keadaan sebelum *error* tersebut terjadi. Sistem juga harus dapat mendeteksi *error* tersebut.

f. *Permit Easy Reversal of Action*

Sebisa mungkin sebuah aksi dalam sistem tersebut dapat dibatalkan.

g. *Support Internal Locus of Control*

Sistem harus berfungsi sebagaimana mestinya dengan respon yang tidak terlalu lama, agar pengguna tidak merasa kecewa

h. Reduce Short-Term Memory Load

Pengguna dalam hal ini manusia memiliki daya ingat yang terbatas, untuk itu desainlah sebuah sistem yang sederhana, tidak terlalu banyak desain-desain yang tidak perlu, seperti banyaknya pergerakan window.

