



### **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

*Website* adalah media penyampaian informasi melalui internet. Macamnya, bisa sebagai penyimpanan informasi komersial (toko *online*), *service* (layanan *web* dan sms), dan penyampaian berita (aplikasi surat kabar *online*). *Website* diciptakan dari serangkaian *script* atau *code* tertentu dari bahasa pemrograman tertentu. Bahasa pemrograman yang dipakai bisa bermacam-macam. Antara lain ASP (*Active Server Page*), dan PHP (*Hypertext Preprocessor*) (FN.Jovan, 2007).

Bahasa Pemrograman PHP adalah bahasa pemrograman yang bekerja dalam sebuah *webserver*. *Script* PHP harus tersimpan dalam sebuah *web server* dan dieksekusi dalamnya *web* (Madcoms Madiun, 2009). PHP memiliki kemiripan dengan bahasa C *programming* dengan menambah cita rasa Perl ke dalamnya. PHP mendukung banyak sekali basis data dan sangat mudah dipelajari. PHP juga dapat menarik data maupun *files* dari basis data, memanipulasi data, dan mengirim data melalui *web browser*. Tidak hanya itu, dengan menggunakan PHP kita dapat memperbaharui basis data, membuat basis data, menghapus basis data, dan melakukan proses matematika. Proses meletakkan kode pada sebuah laman agar *server* mengeksekusinya dikenal sebagai istilah *embedding* (Maxfield, 2000). *Database* (basis data) yang didukung oleh PHP antara lain MySQL, MSSQL, mSQL, Informix, dan Oracle.

Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis. Dalam penulisannya, *script* PHP tidak harus berdiri sendiri namun dapat disisipkan diantara kode HTML. *Script* PHP harus selalu diawali dengan `<? atau <?php` dan diakhiri dengan `?>` (Madcoms Madiun, 2009).

Secara umum, bahasa pemrograman terbagi menjadi dua cabang yaitu *interpreted* dan *compiled*. Bahasa *interpreter* atau penerjemah adalah sebuah program yang membaca berkas yang berisi kode yang akan dieksekusi, dan segera bertindak atasnya. Kode di file disebut *source code*. Secara umum, kode ini dibaca dan dipahami oleh orang (Maxfield, 2000). Sedangkan bahasa *compiled* atau kompiler adalah sebuah program yang membaca berkas sumber dan mengkompilasi kode program menjadi kode biner yang dapat dieksekusi langsung oleh komputer. Kode biner tidak dapat dibaca dan dipahami oleh kebanyakan orang (Maxfield, 2000).

### 3.2. MySQL

Penyimpanan data yang fleksibel dan cepat aksesnya sangat dibutuhkan dalam sebuah *website* yang interaktif dan dinamis. *Database* berfungsi sebagai media penyimpanan data yang diinputkan melalui form *website*. Selain itu, dapat juga menampilkan data yang tersimpan dalam database ke dalam halaman *website*. Jenis *database* yang sangat populer adalah MySQL karena gratis untuk dipergunakan serta dapat berjalan di berbagai *platform*. (Madcoms Madiun, 2009).

*Database* MySQL dapat dibuat dengan menggunakan tampilan jendela phpMyAdmin. Untuk dapat mengakses data dalam *database* MySQL, harus dibuat koneksi terlebih dahulu. Salah satu fungsi untuk melakukan koneksi dengan *database* adalah **mysql\_connect()**; Contohnya:

```
mysql_connect (Nama Host, Nama user, Password);
```

**Keterangan:**

**Nama Host** adalah lokasi di mana MySQL dipublikasikan.

**NamaUser** adalah string dengan isi data nama *user* yang terdaftar dalam MySQL yang digunakan untuk mengakses data pada MySQL.

**Password** → string yang berisikan *password* yang digunakan untuk mengakses *database*. (Madcoms Madiun, 2009)

### 3.3. Interaksi Manusia dengan Komputer

Perkembangan teknologi komputer, membuat hampir semua orang berinteraksi dengan komputer. Sehingga saat ini pengguna teknologi sangat beragam, mulai dari orang awam sampai orang yang ahli dalam menggunakan teknologi. Maka diperlukan *interface* yang mudah dipahami oleh semua kalangan, terutama *user* yang awam.

Interaksi Manusia dan Komputer (IMK) atau *Human-Computer Interaction* (HCI) adalah disiplin ilmu yang berhubungan dengan perancangan, evaluasi, dan implementasi sistem komputer interaktif untuk digunakan oleh manusia, serta studi fenomena-fenomena besar yang berhubungan dengannya (Dix, 1993).

Karakteristik yang memperhatikan *usability* (Shneiderman., 2005)

Tabel 3.1. Karakteristik *usability*

<b>User-centered design</b>	fokus pada <i>user</i> dan <i>task</i> sejak awal desain
<b>Participative design</b>	<i>user</i> sebagai anggota dari tim desainer
<b>Experimental design</b>	terdapat <i>testing usability</i> oleh <i>user</i> secara formal pada percobaan awal, simulasi dan evaluasi <i>prototype</i> secara keseluruhan
<b>Iterative design</b>	desain, <i>testing</i> dan penilaian hasilnya, dan desain ulang sampai hasilnya memenuhi spesifikasi <i>usability</i> yang diinginkan

<b>User-supportive design</b>	<i>training</i> , seleksi (jika perlu) <i>manual</i> , <i>quick reference cards</i> , bantuan dari ‘ahli’ yang ada disekitarnya, sistem bantuan ( <i>help</i> ) seperti: <ul style="list-style-type: none"> <li>- <i>on-line: context-specific help</i></li> <li>- <i>off-line: ‘hot-line’ phone service</i></li> </ul>
-------------------------------	---

Dalam IMK, terdapat 8 *golden rules* (Shneiderman., 2005):

**a. Strive of Consistency**

Paling sering dilanggar, tapi juga paling mudah untuk diperbaiki jika terjadi pelanggaran. Berikut adalah contoh konsistensi.

- i. Urutan *action* yang konsisten pada situasi yang mirip.
- ii. Istilah yang konsisten untuk *prompt*, menu, layar bantu dan perintah.

Pengecualian (pengetikan *password* atau konfirmasi *delete*) harus dibatasi.

**b. Cater to Universal Usability**

Sebuah sistem harus dapat dipahami secara mudah, dengan

- i. penggunaan singkatan,
- ii. penekanan tombol khusus,
- iii. perintah-perintah tersembunyi,
- iv. fasilitas *macro*
- v. *response time* dan *display rate* yang lebih cepat.

**c. Offer Information Feedback**

Tiap *operator action* harus ada *feedback* dari system, misalnya.

- i. Untuk frequent (sering) dan *minor action: response* secukupnya.
- ii. Untuk *infrequent* (jarang) dan *major action: response* harus lengkap.

#### **d. Design Dialog to Yield Closure**

Urutan *action* sebaiknya diatur dalam kelompok mulai dari awal hingga akhir. Feedback yang informatif pada akhir suatu kumpulan *action* akan melegakan dan memuaskan *user*.

#### **e. Prevent Error(design yg membuat user tidak melakukan error)**

Sebisa mungkin, desainlah sistem sedemikian rupa sehingga *user* tidak mungkin bisa membuat *error* yang serius.

- i. Jika terjadi *error*, sistem harus bisa mendeteksi *error* dan cara mengatasinya.
- ii. *User* hanya perlu mengetikkan bagian perintah yang *error* saja (tidak semuanya).
- iii. Perintah yang mengakibatkan *error* tidak boleh mengubah *state* dari system.
- iv. Sistem harus memberikan instruksi untuk mengembalikan *state* ke keadaan semula sebelum terjadi *error*.

#### **f. Permit Easy Revesal of Action(action bisa di undo)**

Sebisa mungkin, semua *action* harus bisa dibatalkan (*undo*). Dan memuaskan rasa ingin tahu dari *user*.

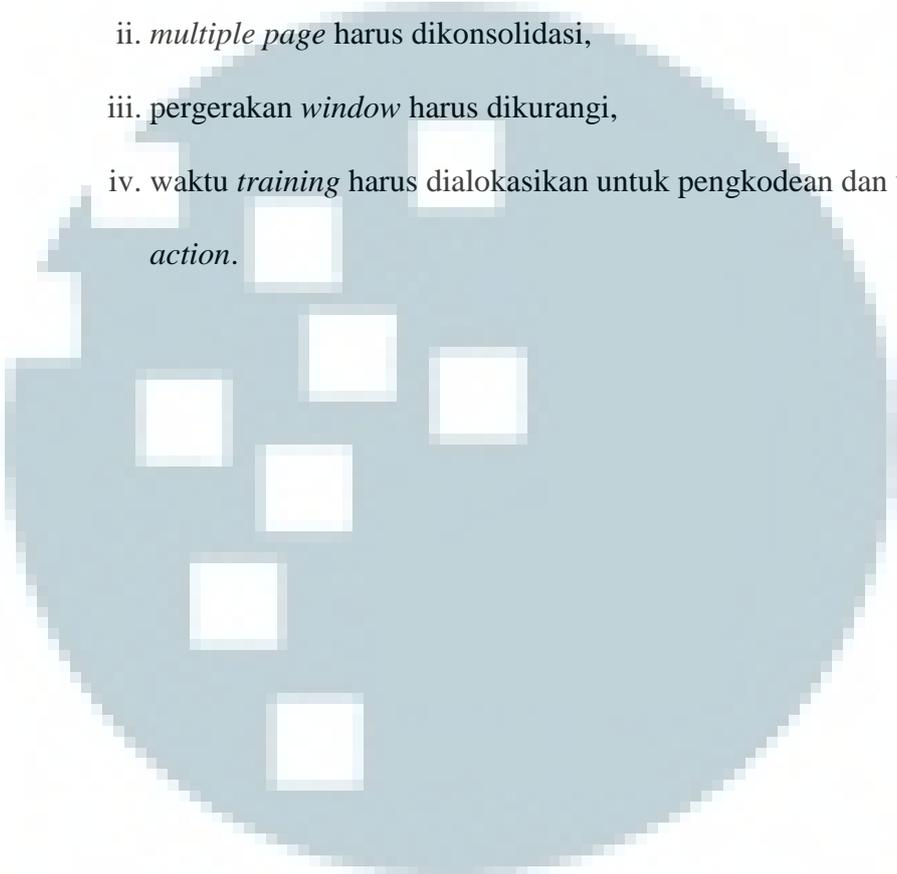
#### **g. Support Internal Locus of Control**

*User* yang sering menggunakan sistem mengharapkan bahwa mereka bisa mengontrol sistem dan *response*-nya. *Data entry* yang bertele-tele, kesulitan dalam mendapatkan informasi, ketidak-mampuan dalam melakukan suatu *action*, semuanya bisa mengakibatkan kekecewaan *user*.

## **h. Reduce Short-Term Memory Load**

Manusia memiliki kemampuan *short-term memory* yang terbatas, sehingga tampilan informasi pada layar komputer haruslah,

- i. sederhana,
- ii. *multiple page* harus dikonsolidasi,
- iii. pergerakan *window* harus dikurangi,
- iv. waktu *training* harus dialokasikan untuk pengkodean dan urutan *action*.



UMN