BAB II

LANDASAN TEORI

2.1. Registrasi

Berasal dari bahasa Inggris '*registration'* yang memiliki arti daftar. Registrasi merupakan proses melakukan pengisian sejumlah hal atau memenuhi persyaratan dari suatu objek yang nantinya dibutuhkan untuk mengikuti suatu kegiatan.[KMS'00]

2.2. Uji Kompetensi

Uji kompetensi berasal dari dua kata, yaitu 'Uji ' dan' Kompetensi'. Uji itu sendiri memiliki pengertian suatu kegiatan yang mencoba untuk mengetahui mutu sesuatu (ketulenen, kecakapan, ketahanan, dsb), sedangkan 'Kompetensi' merupakan kemampuan yang dimiliki oleh suatu objek [KMS'00]. Sehingga dapat disimpulkan bahwa uji kompetensi merupakan kegiatan yang mencoba kemampuan yang sudah dimiliki oleh suatu objek.

2.3. Diklat

Diklat merupakan singkatan dari Pendidikan dan Pelatihan, dimana arti dari 'Pendidikan' adalah suatu proses mengubah sikap dan tata laku seseorang atau kelompok orang dalam usaha mendewasakan manusia melalui upaya pengajaran dan pelatihan, proses, perbuatan, cara mendidik. Dan arti dari 'Pelatihan' merupakan proses, cara, perbuatan melatih, kegiatan atau pekerjaan melatih membiasakan diri agar mampu (dapat) melakukan sesuatu [KMS'00]. Diklat ini

sendiri memiliki artian melakukan suatu kegiatan bertujuan kepada peningkatan kemampuan yang dimiliki oleh seseorang.

2.4. Internet

Singkatan dari *Interconection Network The Network of the Networks.* Diartikan sebagai *a global of computer networks* atau sebuah jaringan komputer dalam skala global / mendunia. Jaringan komputer ini berskala internasional yang dapat membuat masing-masing komputer saling berkomunikasi. *Network* (jaringan antar komputer yang menghubungkan satu komputer dengan komputer yang lainnya) ini membuat jaringan inter-koneksi (*inter-connected network)* yang terhubung melalui protokol TCP/IP (*Transmission Control Protocol / Internet Protocol*; merupakan standar protokol internet). Dikembangkan dan diuji coba pertama kali pada tahu 1969 oleh US Departement of Defense dalam Proyek ARPAnet.[FJ'04]

2.4.1. Web

Web merupakan suatu sistem di internet yang memungkinkan siapapun agar menyediakan informasi. Dengan menggunakan teknologi tersebut maka informasi dapat diakses selama 24 jam dalam satu hari dan dikelola oleh mesin. Untuk mengakses informai yang disediakan web ini, diperlukan berbagai perangkat lunak, yang disebut dengan *web broser. Web broser* merupakan suatu program komputer yang menyediakan fasilitas untuk membaca alamat web di suatu komputer [FJ'04]

2.5. SQL

SQL merupakan singkatan dari *Struktur query Language.* Di dalam dunia *database* istilah *query* dapat diartikan "permintaan data". SQL merupakan bahasa tingkat empat yang berfungsi menampilkan hasil atau melakukan sesuatu pada data yang diinginkan. Dalam SQL tidak perlu mendefinisikan bagaimana data itu diambil atau diproses, seperti halnya bahasa tingkat tiga, hanya perlu memberikan perintah mengenai data apa yang ingin diambil atau dieksekusi. Sehingga tidak perlu memikirkan bagaimana proses data tersebut diambil atau dieksekusi. Cukup dengan meminta data apa yang ingin diambil atau dieksekusi.

Saat ini SQL merupakan bahasa *query* standar di berbagai *software database.* Berbagai *software database* dapat diakses menggunakan bahasa SQL. Sehingga selain MySQL atau Oracle, dapat menggunakan SQL di *software database* apapun.

Software database mulai bermunculan seiring dengan bertambahnya kebutuhan akan *database server*. Salah satu *database server* yang banyak dipakai ialah MySQL. MySQL adalah multiuser *database* yang menggunakan bahasa *Structured Query Language* (SQL) yang berasal dari Skandinavia. MySQL terdiri atas *server* SQL, *client* program untuk mengakses *server*, *tools* untuk administrasi dan *interface* program untuk menulis program sendiri.

MySQL mempunyai banyak hal yang dapat ditawaran sebagai bahan pertimbangan dalam merencanakan sistem *database,* yaitu:

1. Kecepatan, banyak ahli berpendapat MySQL merupakan *server* tercepat

2. Kemudahan penggunaan, MySQL mempunyai performa tinggi namun merupakan *database* yang *simple* sehingga mudah di-*setup* dan dikonfigurasi
3. Harga, MySQL cendrung gratis untuk pengguna tertentu.
4. Mendukung *query language,* MySQL (*Struktured Query Connectivity)* buatan Microsoft
5. MySQL juga dapat siakses lewat protokol ODBC (*Open Database Connectivity*) buatan Microsoft
6. Kapabilitas, banyak *client* dapat mengakses *server* dalam satu waktu, *Client* dapat menggunakan banyak *database* secara simultan
7. Konektifitas dan sekuritas, *database* MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu
8. Protabilitas, MySQL dapat berjalan dalam banyak varian UNIK dengan baik, sebaik seperti berjalan di sitem non-UNIK
9. Distribusi yang terbuka, MySQL mudah didapatkan dan memiliki *source code* yang boleh disebarluaskan sehingga dapat dikembangkan lebih lanjut.

2.5.1. Perintah CREATE

Perintah-perintah dalam SQL, tergolong mudah, tetapi bila tabel yang dimiliki begitu *fleksibel* maka akan perintah yang digunakan lebih sulit untuk membuat SQL tersebut. Bentuk dasar dari perintah ini adalah :

CREATE TABLE <table name> (<columnNameN> <co1. N type> <co1. N detail>, ...) ;

2.5.2. Perintah SELECT

Perintah *SELECT* merupakan perintah dasar untuk menampilkan data. Perintah ini memiliki format sebagai berikut :

SELECT * FROM <tableName> WHERE <ColumnNameN=ValueN>, . . . ;

2.5.3. Perintah INSERT

Perintah untuk menyisipan data yang telah dibuat dalam database di sebut *INSERT*. Ada dua bentuk dasar untuk perintah *INSERT* ini, sebagai berikut:

INSERT INTO <tableName> SET <columnNameN=ValueN>, ... ;

INSERT INTO <tableName> (columnNameN) , ... VALUES (value), . . ;

2.5.4. Perintah DELETE

Perintah *DELETE* ini digunakan untuk menghapus isi dari SQL. Menghapus isi SQL mudah dan berbahaya. Bentuk umum penulisan perintah *DELETE* :

DELETE FROM <tableName> WHERE <columnNameN=ValueN>, . . . ;

2.5.5. Perintah UPDATE

Perintah *UPDATE* ini digunakan apabila melakukan perubahan dari isi SQL, bentuk penulisannya adalah :

UPDATE <tableName> SET <ColumnNameN=ValueN>, ... ;

2.6. PHP

PHP (*Personal Home Page*) merupakan salah satu fasilitas yang bersifat *server-side* dan relatif lebih aman dari pembajakan dan penyalahgunaan. Bentuk penulisan untuk PHP ini tidak jauh berbeda dengan HTML ataupun *JavaScript,*

pada saat pengeksekusian sama halnya dengan *script* yang lain memungkinkan untuk mengontrol apa yang akan di tampilkan pada *window browser.*

Kelebihan PHP dibandingkan dengan bahasa pemograman yang lain adalah mudah dibuat dan cepat dijalankan. PHP dapat berjalan dalam *web server* yang berbeda dan dalam sistem operasi yang berbeda pula.

Karena PHP adalah sebuah bahasa HTML-*emedded* maka harus ada sebuah cara untuk memisahkan bagian mana *file* tersebut yang merupakan modus PHP, atau dengan kata lain sebuah cara untuk keluar dari HTML (*Escapping from HTML*) dan masuk ke modus PHP. Ada tiga cara untuk keluar dari HTML ke PHP yaitu :

1. <? . . . . . ?>
2. <?php . . . . . ?>
3. <script language=”php”> . . . .</script>

Dimana titik-titik diisi dengan perintah-perintah PHP. Cara pertama akan banyak digunakan karena paling ringkas. Cara kedua dapat dipergunakan untuk kombinasi XML (*extented Markup Language*), sebuah bahasa pengembangan dari HTML. Sedangkan cara ketiga untuk mengantisipasi editor-editor yang tidak dapat menerima *tag1* dan *tag2,* contohnya *Frontpage.*

Contoh penulisan PHP :

```
<HTML>
<Head><TITLE>Today’s Date</TITLE></HEAD><BODY>
<P>Today’s Date (according to this Web server) is <?php
Echo ( date (“1, F ds Y. “) ) ; ?>
</BODY></HTML>
```

Setiap variabel dalam PHP diawali dengan karakter $, misalnya variabel x dapat ditulis $x. Nama variabel dalam PHP bersifat *case-sensitive,* artinya huruf besar dengan huruf kecil dibedakan, dan berlaku untuk variabel saja, tetapi PHP bukan bahasa yang *case-sensitive.*

PHP mengenal variabel dengan berbagai macam tipe :

1. Skalar, Variabel nilai tunggal, variabel skalar dapat bertipe
    a. *Integer* (bilangan bulat)
    b. Double, bilangan real (*floating-point)*
    c. String (karakter)
2. *Array,* variabel yang memiliki nilai dengan berbagai tipe skalar dan memiliki indeks.
3. objek, variabel yang memiliki *property* dan *method*.

Ada enam macam *operator* dalam PHP yaitu :

1. Operator Arimatika.

    Operator aritmatika merupakan operator yang berhubungan dengan fungsi matematika. Oprerator ini dapat dilihat pada tabel 2.1.

Contoh	Nama	Hasil
$a +Sb	Penjumlahan	Jumlah $a dengan $b
$a - $b	Pengurangan	$a dikurangi $b
$a * $b	Perkalian	Hasil kali $a dan $b
$a / $b	Pembagian	$a dibagi $b
$a % $b	Modulus (sisa pembagi)	Sisa dari $a dibagi $b

Tabel 2.1. Operator Aritmatika

2. Operator *String*

   Hanya ada satu operator *string* dalam PHP yaitu penggabungan (“,”)

3. Operator Penugasan (*assigment*)

   Operator penugasan dalam PHP adalah “=”, artinya *operand* di sebelah kiri diisi dengan *operand* di sebelah kanan.

4. Operator *Bitwise*

   Operator *bitwise* mengubah bit-bit tertentu dalam sebuah variabel *integer* menjadi *on* atau *off*. Operator ini dapat juga di lihat pada tabel 2.2

Contoh	Nama	Hasil
$a & $b	And	Bit yang on dikedua $a dan $b menjadi on
$a | $b	Or	Bit yang on disalah satu $a atau $b menjadi on
~$a	Not	Bit yang on di $a menjadi tidak on

Table 2.2. Operator *bitwise*

5. Operator Logika

   Operator logika akan membandingkan *true* dan *false* yang didefinisikan juga sebagai 0 dan 1. Tabel 2.3. menunjukan daftar operator ini

Contoh	Nama	Hasil
$a and $b	And	Benar jika $a dan $b benar
$a or $b	Or	Benar jika $a atau $b benar
$a xor $b	Xor	Benar jika $a atau $b benar,

		tetapi tidak jika keduanya benar
!$	Not	Benar jika $a tidak benar dan kebalikannya
$a && $b	And	Sama dengan and
$a || $b	Or	Sama dengan or

Tabel 2.3. Operator logika

6. Operator Pembandingan

Operator ini membandingkan beberapa variable yang menghasilkan *true* atau *false.* Dapat dilihat pada Tabel 2.4.

Contoh	**Nama**	**Hasil**
$a == $b	Equal	Benar jika $a sama dengan $b
$a != $b	Not Equal	Benar jika $a tidak sama dengan $b
$a < $b	Less Than	Benar jika $a kurang dari $b
$a > $b	Greather Than	Benar jika $a lebih dari $b
$a <= $b	Less than or equal to	Benar jika $a kurang atau sama dengan $b
$a >= $b	Greather than or equal to	Benar jika $a lebih atau sama dengan $b

Tabel 2.4. Operator Pembandingan

Perintah pencabangan dalam PHP ada dua macam yaitu :

1. *IF*

   Else dan *elseif* untuk beberapa kondisi

2. *Switch*

Case untuk pencabangan yang mengevaluasi apakah variabel / ekspresi bernilai tertentu.

Perintah pengulangan dapat digunakan untuk perintah berulang, ada tiga macam perintah pengulangan dalam PHP, yaitu :

1. *For*

   Digunakan untuk mengulangi perintah dengan jumlah pengulangan yang sudah diketahui.

2. *While*

   Digunakan untuk pengulangan sebuah perintah selama kondisi tertentu.

3. *Do ... While*

   Sama dengan perintah *While* hanya saja pada akhir ada pemeriksaan kondisi.

Disamping membuat fungsi-fungsi sendiri dalam sebuah program, misalnya fungsi dengan menggunakan argumen (*parameter*). Dapat juga menggunakan fungsi-fungsi yang sudah disediakan oleh PHP. PHP memiliki ratusan fungsi-fungsi yang akan sangat membantu. Fungsi ini biasa disebut *Built-in Function* dalam PHP. Untuk dapat menggunakan sebuah fungsi, kita harus dapat membaca dan memahami spesifikasi, seperti tipe data, format untuk memasukan dan keluaran yang dibutuhkan suatu fungsi. Disamping itu masukan ada yang bersifat *mandatory* (harus ada) dan *optional* (boleh tidak ada) .

Beberapa contoh *Built-in Function* antara lain :

1. Fungsi-fungsi *Array*

PHP memiliki banyak fungsi-fungsi yang dapat digunakan untuk memanipulasi sebuah *array*, misalnya *sort* dan *count*.

2. Fungsi-fungsi tanggal dan jam

   PHP menyediakan fungsi-fungsi yang menghasilkan *string* dan waktu tertentu dengan format tertentu. Fungsi-fungsi ini misalnya *gatedale, date.*

3. Fungsi-fungsi matematika

   PHP juga menyediakan fungsi-fungsi trigonometri, logaritma dan lain sebagainya.

4. Fungsi-fungsi Variabel

   Fungsi-fungsi variabel dari PHP berguna untuk memanipulasi tipe-tipe variabel, seperti mengetahui tipe, mengubah tipe dan lain sebagainya.

Sebuah situs yang interaktif memiliki halaman yang memungkinkan *user* memasukan informasi-informasi tertentu, untuk dapat melakukan itu kita memerlukan *form,* Sebuah *form* biasanya file HTML, karena itu sebuah *form* dapat dibuat dengan menggunakan tag-tag HTML, ada dua tag HTML yang sangat penting untuk membuat *form* yaitu <FORM> dan <inout> properti yang penting dalam tag <FORM> adalah :

1. *ACTION* , menyatakan nama file yang akan digunakan untuk memberikan respon terhadap *form* (file PHP).
2. *NAME,* menyatakan nama dari *form* bersangkutan.
3. *METHOD,* menyatakan bagaimana data dalam *form* dikirim, ada dua macam method yang digunakan : *GET* mengirimkan data dengan

mencantumkan isi *form* sebagai parameter dalam URL dan method tidak mencantumkan data pada URL.

4. *TARGET*, menyatakan *frame* mana yang akan menampung respon, lihat pada tag *FRAME*.

Untuk membuat masukan kita menggunakan tag <INPUT> properti-properti yang penting untuk <INPUT> antara lain adalah :

1. *NAME,* menyatakan nama dan input..
2. *TYPE,* menyatakan jenis masukan,ada banyak jenis masukan antara lain :
    a. *TEXT*, merupakan masukan yang diinput oleh *user*.
    b. *PASSWORD*, seperti halnya *TEXT* yang diketikan dan *user* dimunculkan sebagai karakter ”*”, agar tidak dibaca oleh orang lain.
    c. *SUBMIT,* merupakan tombol yang jika *user* mengkliknya maka *form* akan dikirimkan dan respon file PHP akan dijalankan.
    d. *RADIO,* berupa pilihan satu tanda bulat.
    e. *CHECKBOX, check* pada kotak pilihan
    f. *SELECT,* berupa *combo box*
    g. *VALUE,* menyatakan nilai yang dimiliki oleh masukan tersebut. Nilai tersebut akan dimunculkan di *browser*.

PHP mempunyai kemampuan untuk mengakses contain dan suatu *database* ke dalam *web*. *Database* yang dapat diakses lewat PHP3 adalah *database* ODBC, *database server* seperti MYSQL dan juga Oracle.

2.7. Basis Data (database)

Basis data *(database)* merupakan kumpulan data yang saling berhubungan satu sama lain, tersimpang diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Sampai dengan membentuk *database, database* sangat penting dalam pembuatan program, karena nantinya akan memudahkan pengaksesan file didalam desain *database* dan mempunyai jenjang mulai dari *bit, charakter, filed, record, file, dan* kemudian membentuk *database.* Yang akan di jelaskan dibawah ini :

1. *Bit*

   Merupakan satuan terkecil dalam hitungan biner. Satu bit menunjukan dasar yang mewakili nilai 0 atau 1.

2. *Character*

   Merupakan lambang-lambang yang terdiri dari huruf, angka, serta lambang-lambang lainnya, dibentuk dari susunan bit.

3. *Field*

   Field terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis.

4. *Database*

   Representatif kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi yang tidak perlu)

5. *File*

   Arsip yang disimpan pada suatu media yang terdiri dari kumpulan karakter, dan di dokumentasikan dalam bentuk data digital oleh komputer.

2.8. Pengertian Client / Server

Sistem *Client-server* biasanya berjalan pada setidaknya dua sistem komputer yang berbeda, satu komputer bertindak sebagai *client* dan lainnya sebagai server, tetapi *Client* dan *Server* juga biasa berada pada satu sistem komputer biasanya sebuah *Server* melayani beberapa komputer *Client*, walaupun mungkin ada juga yang hanya melayani satu *client* saja. Fungsi *client- server* biasanya dilakukan oleh file *server* kecuali apabila dibutuhkan kinerja yang maksimal maka digunakanlah *server* khusus. *Client* biasanya berupa sebuah komputer dekstop yang terhubung dalam jaringan, apabila pemakai ingin mengambil dan menyimpan informasi bagian aplikasi *Client* akan mengeluarkan permintaan dan akan di kirimkan (biasanya melalui jaringan) kepada *Server, Server* kemudian menjalankan permintaan dan mengirimkan informasi kapada *Client.*

Sebuah *database* tidak dapat menggantikan *Client/Server*, walaupun sistem *Client/Server* sering menggunakan *database* untuk melakukan aktifitas *Server,* aplikasi yang dirancang untuk menggunakan aktivitas *Server* menggunakan Acces, Mcrosoft FoxPro, Program data jaringan lainnya semuanya adalah contoh-contoh aplikasi *database* jaringan karena semua pemrosesan dilakukan oleh *Client.*