

- Functional requirements
- Domain modelling
- Behavioral requirements / use case modelling
- Requirements review (milestone 1)

2. Analysis and preliminary design :

- Robustness analysis
- Update domain model
- Tulis kembali use case yang telah dibuat
- Preliminary design review (Milestone 2)

3. Detailed design :

- Sequence diagram
- Update domain model
- Critical design preview

4. Implementation

- Coding / unit testing
- Integration & scenario testing
- Perform code review and model update

b. Kelebihan dari metode *iconix* :

1. Keunggulan utama *ICONIX* (dan membedakannya dengan metodologi lain) adalah pemanfaatan robustness diagram untuk melakukan analisa kehandalan. Dengan analisa kehandalan, akan ditemukan objek-objek baru yang sebelumnya tidak teridentifikasi.

2. *ICONIX* menggunakan *UML* secara elegan, tidak berlebihan. Hal ini pula yang menyebabkan *ICONIX* lebih ringkas dibanding metodologi lainnya.

c. Kelemahan dari metode *iconix* :

1. Dikarenakan *Iconix Process* merupakan proses yang dipicu oleh use case (use case driven) maka penentuan model dan perilaku sistem harus dibangun sejak awal sehingga sulit dikembangkan oleh banyak orang
Iconix Process merupakan metode yang interaktif dan bertahap (iterative-incremental) maka diperlukan iterasi yang banyak saat penentuan permodelan
2. *Iconix process* yaitu suatu metode dimana tidak terlalu banyak membahas pada analisa, design maupun implementasi programnya. Namun lebih melihat kepada kebutuhan pengguna serta menyederhanakan proses tersebut, sehingga proses pengembangan perangkat lunak akan menjadi lebih efisien.

2.2.13.2 Pengertian *Unified Modeling Language* (*UML*)




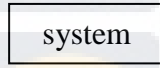

Unified Modeling Language (*UML*) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (*OOP*). Selain itu *UML* adalah bahasa pemodelan yang menggunakan konsep orientasi object. *UML* dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corp.* *UML* menyediakan notasi-notasi yang membantu memodelkan sistem dari

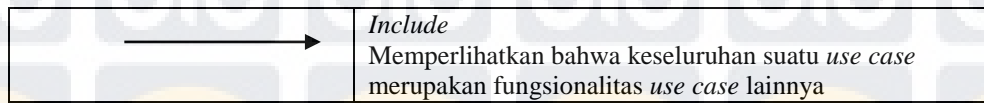
berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. UML dideskripsikan oleh beberapa diagram, diantaranya:

a. Use case Diagram

Use case adalah diagram UML yang memperlihatkan hubungan antara actor dan *use case* yang berjalan pada sistem, *use case* biasanya digunakan untuk memperlihatkan keseluruhan dari kebutuhan sebuah sistem dalam bentuk yang esensial, mengkomunikasikan lingkup pembangunan sebuah proyek, dan untuk memodelkan sebuah analisis dari kebutuhan sebuah sistem dalam bentuk *use case* model. Sebuah *use case* mendeskripsikan urutan – urutan aksi yang menghasilkan hasil yang terukur untuk sang aktor dan untuk mencapai hal tersebut.

Tabel 2.2 Relasi Use Case Diagram

Simbol	Penjelasan
	Aktor Adalah sesuatu atau seseorang yang ada diluar sistem dan ikut berperan serta dalam aktifitas system.
	Dependency Menggambarkan ketergantungan antar <i>use case</i> yang ada
	Use case Memberikan informasi seputar aksi yang dilakukan secara teratur dalam sebuah sistem
	Sistem Berisi seluruh aktifitas sistem pada <i>use case</i> dan ketergantungan serta hubungannya dengan aktor
	Association Menggambarkan interaksi antara aktor dan <i>use case</i> yang ada




Sumber : “*Sistem Manajemen File Berbasis Web Pada Yayasan WWF Indonesia*” Jeffrey, Fatimah A., 2012.

b. Activity Diagram

UML *activity* diagram adalah diagram yang sesuai dengan diagram *flow chart* dan data yang ada pada *flow* diagram yang didapat dari pengembangan yang terstruktur. Biasanya, UML *activity* diagram digunakan untuk mengeksplorasi logika daripada *complex operation*, *complex business rule*, *single use case*, *several use case*, *business process*, *concurrent process*, dan *software process*. (Ambler, 2005).

Tabel 2.3 Relasi Pada Activity Diagram

Simbol	Penjelasan
	<i>Initial State</i> Menggambarkan dimulainya suatu aktivitas pada <i>activity</i> diagram
	<i>Decision Point</i> Menentukan kapan alur aktivitas menjadi bercabang
	<i>Transition Line</i> Menampilkan hubungan atau relasi antar state dan bagaimana alur sistem berjalan sebagaimana mestinya
	<i>Action State</i> Menggambarkan hasil eksekusi dari <i>atomic</i> .
	<i>Synchronization bar</i> Memperlihatkan dua atau lebih aktivitas yang dapat dilakukan secara paralel.
	<i>Fork</i> <i>Symbol</i> ini menunjukkan awal aktivitas paralel.



	<p><i>Final State</i> Menandakan akhir daripada alur kerja suatu sistem dalam <i>activity diagram</i>.</p>
---	--



Sumber : “*Sistem Manajemen File Berbasis Web Pada Yayasan WWF Indonesia*” Jeffrey, Fatimah A., 2012.

c. Class Diagram

UML *Class diagram* memperlihatkan kelas – kelas daripada sebuah sistem serta hubungan antar kelas itu sendiri. Operasi – operasi dan atribut yang ada didalam kelas biasanya digunakan untuk, eksplorasi wilayah konsep dalam rangka membuat batasan wilayah model sistem, menganalisa kebutuhan dalam kerangka konseptual/analisa model, dan menggambarkan *design object oriented* atau *object based software* secara mendetail. *Classdiagram* terdiri dari satu atau lebih kelas yang dapat mendukung spesifikasi yang dapat menjelaskan model *elements*, termasuk kelas, hubungan antar kelas dan rancangan antar muka.(Ambler, 2005).

Tabel 2.4 Relasi Pada Class Diagram

Simbol	Penjelasan			
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;"> <<Interface>> Class2 </td> </tr> <tr> <td style="text-align: center;"> Atribut 1 Atribut 2 </td> </tr> <tr> <td style="text-align: center;"> Operasi 1 Operasi 2 </td> </tr> </table>	<<Interface>> Class2	Atribut 1 Atribut 2	Operasi 1 Operasi 2	<p><i>Class</i> Deskripsi dari objek yang terbagi atas 3 bagian, yaitu nama <i>class</i> pada bagian atas, atribut pada bagian tengah, dan operasi pada bagian bawah.</p>
<<Interface>> Class2				
Atribut 1 Atribut 2				
Operasi 1 Operasi 2				
	<p><i>Aggregation</i> Bentuk spesial dari hubungan asosiasi yang memiliki hubungan spesifik antara kumpulan dan sebuah bagian. Agregasi digambarkan dengan wajik tidak berisi.</p>			
	<p><i>Association</i> Mengambarkan hubungan yang terstruktur antar <i>class</i> yang saling berelasi.</p>			

	Generalization Relasi yang memperlihatkan satu kelas dapat lebih <i>general</i> atau lebih spesifik dari kelas lainnya.
	Multiplicity Menggambarkan jumlah objek yang berpartisipasi dalam hubungan antar <i>class</i> .

Sumber : “*Sistem Manajemen File Berbasis Web Pada Yayasan WWF Indonesia*” Jeffrey, Fatimah A., 2012.

1. Multiplicity

Multiplicity pada asosiasi adalah jumlah banyaknya obyek sebuah *class* yang berelasi dengan sebuah obyek *class* lain yang berasosiasi dengan *class* tersebut.

Tabel 2.5 *Multiplicity*

Indikator	Arti	Keterangan / Contoh
0..1	Kosong atau satu	
0..*	Kosong atau lebih	
0..n	Kosong atau n, n > 1	0..3
1	Hanya satu	
1..*	Lebih dari sama dengan satu	
1..n	Satu atau n, n > 1	1..5
*	Banyak atau many	
N	Hanya N dimana N lebih dari satu	9
n..*	Lebih dari sama dengan n, n > 1	7..*
n..m	Lebih dari sama dengan n dan kurang dari sama dengan m, dimana n dan m lebih dari satu	3..10





Sumber : “*Sistem Manajemen File Berbasis Web Pada Yayasan WWF Indonesia*” Jeffrey, Fatimah A., 2012.

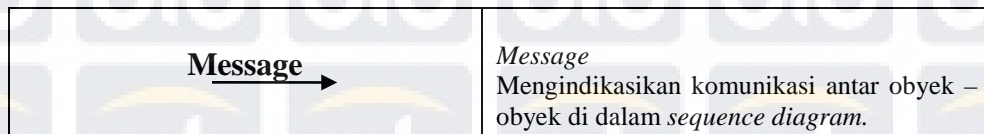
d. Sequence Diagram

UML *Sequence diagrams* adalah model diagram yang dirancang agar menjadi diagram yang dinamis, selain itu diagram ini juga digunakan sebagai alat komunikasi antar diagram. UML *Sequence diagrams* biasanya digunakan

untuk, validasi menyempurnakan *logic* dan menjelaskan secara detail suatu *use case* skenario. Penggunaan nama yang disesuaikan dengan *use cases* skenario dapat mengindikasikan atau mendeskripsikan bagaimana sebuah sistem berjalan, mengeksplorasi desain UML, karena *sequence* diagram menyajikan diagram agar dapat dimengerti secara visual bagaimana sebuah sistem berjalan secara terstruktur. Mendeteksi hambatan-hambatan yang mungkin timbul diantara *designobject-oriented*. Dengan melihat pesan yang dikirim dari sebuah objek dan juga dengan melihat secara detail berapa lama waktu yang dibutuhkan untuk suatu proses dengan proses yang lainnya berjalan. (Ambler, 2005).

Tabel 2.6 *Simbol-simbol Sequence Diagram*

Simbol	Penjelasan
	<i>Actor</i> <i>Actor</i> diurutkan sebagai kolom. Simbol <i>actor</i> pada <i>sequence</i> diagram sama dengan <i>actor</i> pada <i>use case</i> diagram
	<i>Activation</i> Digunakan untuk mengindikasi sebuah obyek yang akan melakukan sebuah aksi.
	<i>Lifeline</i> Mengindikasikan keberadaan sebuah obyek dalam baris waktu. <i>Lifeline</i> adalah sebuah garis putus – putus yang ditarik dari sebuah obyek
	<i>Object</i> <i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama obyek yang didalamnya diawali dengan sebuah titik koma.



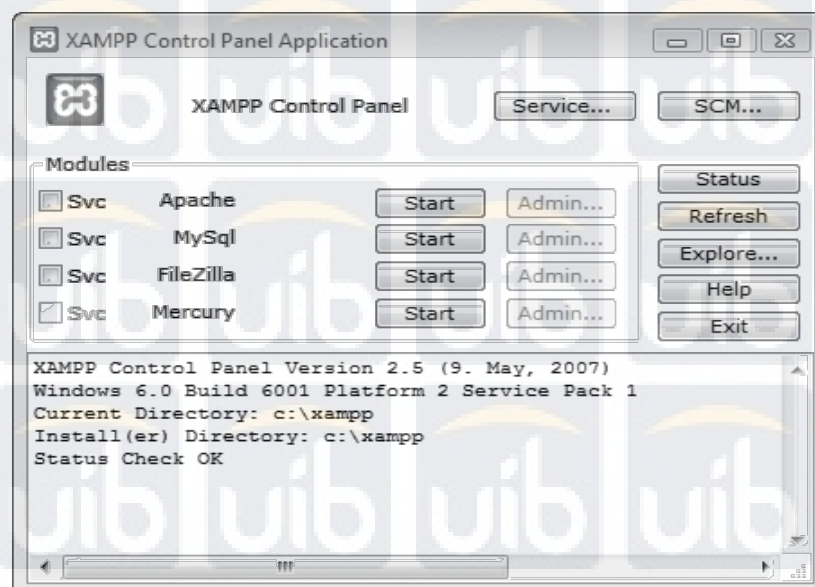
Sumber : “Sistem Manajemen File Berbasis Web Pada Yayasan WWF Indonesia” Jeffrey, Fatimah A., 2012.

2.2.14 Perangkat Lunak Pendukung (*Software*)

2.2.14.1 XAMPP

XAMPP merupakan sebuah paket instalasi untuk PHP, Apache, dan MySQL. Dengan menggunakan XAMPP, *user* tidak perlu lagi repot menginstal ketiga *software* itu secara terpisah. XAMPP sama seperti PHP yang sudah sering digunakan oleh *programmer Web* dalam membuat aplikasi *Web*. (Ibrahim, 2008).

Dibawah ini ditampilkan *mainscreen* dari aplikasi xampp yang digunakan untuk perancangan aplikasi ini pada Gambar 2.2



Gambar 2.2 *MainScreen XAMPP*

2.2.14.2 Apache

ServerHTTP Apache atau *ServerWeb/WWW Apache* adalah *Webserver* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta *platform* lainnya) yang berguna untuk melayani dan menfungsikan situs *Web*. Protokol yang digunakan untuk melayani fasilitas *Web/www* ini menggunakan HTTP. *Webserver* adalah suatu program (dan juga mesin yang menjalankan program) yang mengerti protokol HTTP dan dapat menanggapi permintaan – permintaan dari *Web browser* yang menggunakan protokol tersebut. (Febrian, 2007).

Beberapa dukungan *Apache* :

- a. Kontrol Akses. Kontrol ini dapat dijalankan berdasarkan nama host atau nomor IP
- b. CGI (*Common Gateway Interface*),
- c. PHP (*Personal Home Page/PHP Hypertext Processor*); *Apache* mendukung PHP dengan menemukannya sebagai salah satu modulnya (*mod_php*). Hal ini membuat kinerja PHP menjadi lebih baik.
- d. SSI (*Server Side Includes*)
- e. *Virtual host*.

2.2.14.3 MySQL

MySQL(my structure Query language) adalah sebuah program pembuat database yang bersifat open source, artinya siapa saja boleh menggunakan dan tidak di cekal.

2.2.14.4 PHP MyAdmin

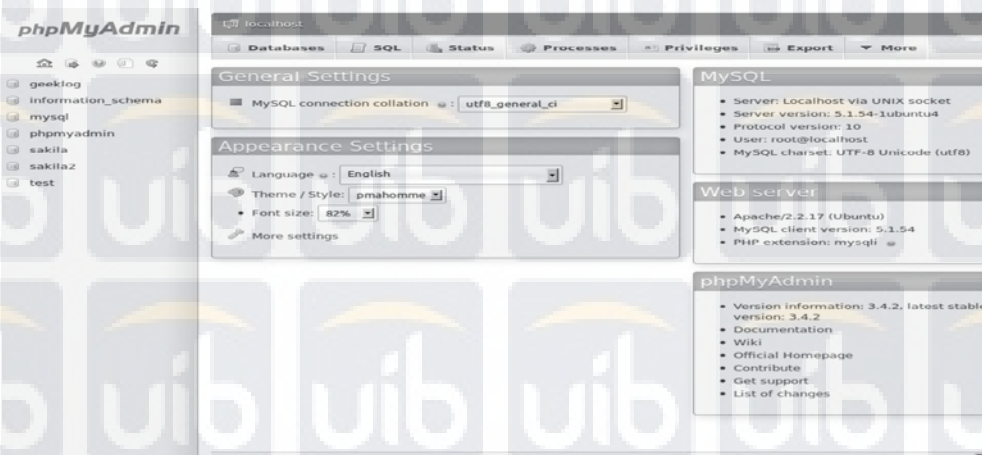
PhpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrogramanPHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring Jagat Jembar (*World Wide Web*). PhpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain).

Adapun fitur-fitur yang terdapat dalam phpMyAdmin adalah, antarmuka berbasis *Web*. Dukungan banyak fitur MySQL:

- a. Menelusuri dan drop basisdata (*database*), tabel, pandangan (*view*), bidang (*fields*) dan indeks.
- b. Membuat, menyalin, drop, dan mengubah nama basis data, tabel, kolom dan indeks.
- c. Pemeliharaan *server*, basis data dan tabel, dengan *server* konfigurasi.
- d. Melaksanakan, mengedit dan penunjuk pernyataan-SQL, bahkan *batch-queries*
- e. Mengelola pengguna MySQL dan hak istimewa.
- f. Mengelola prosedur penyimpanan.

- g. Impor data dari CSV dan SQL
- h. Ekspor data ke berbagai format: CSV, SQL, XML, PDF, ISO / IEC 26300 *OpenDocument Text* dan Spreadsheet, Word, Excel, LATEKS.
- i. Membuat grafik PDF dari tampilan basis data anda.
- j. Membuat kompleks *query* menggunakan *Query-by-example*. (QBE)
- k. Pencarian global dalam basis data.
- l. Transformasi data disimpan ke dalam format yang menggunakan satu set fungsi yang telah ditetapkan, seperti menampilkan data *blob-data* atau *download-link*.

Adapun tampilan *mainscreen* dari phpMyAdmin ditampilkan pada Gambar 2.3 dibawah ini.



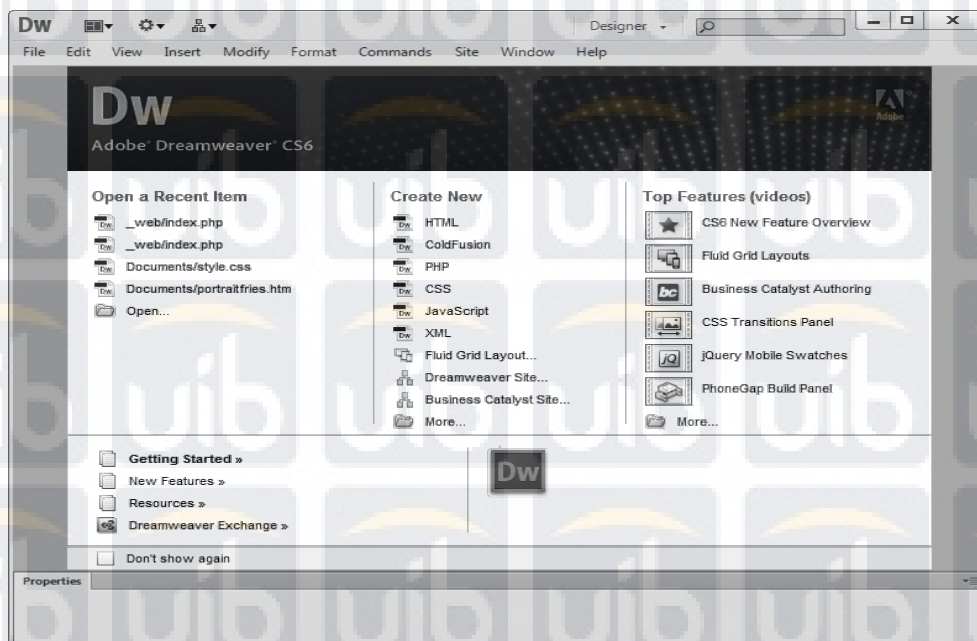
Gambar 2.3 MainScreen phpMyAdmin

2.2.14.5 Adobe Dreamweaver

Adobe *Dreamweaver* merupakan salah satu *software* dari kelompok adobe yang banyak digunakan untuk mendesain situs *Web*. Adapun adobe *Dreamweaver* itu sendiri adalah sebuah HTML editor professional untuk mendesain secara

visual atau mengelola situs atau halaman *Web*. *Dreamweaver* memiliki performa yang lebih baik dan memiliki tampilan yang memudahkan anda untuk membuat halaman *Web*, baik dalam jendela desain maupun dalam jendela kode. *Dreamweaver* didukung dengan cara pemakaian yang praktis dan standar, dan juga didukung untuk pengembangan penggunaan HTML, CSS3, XML, dan RSS dan kemudahan-kemudahan lain yang diperlukan.

Berikut ini ditampilkan *mainscreen* dari *adobe Dreamweaver CS6* pada Gambar 2.4.



Gambar 2.4 *Adobe Dreamweaver CS6*