

Abstract

Lately, there are plenty of Event Organizer which established individually or in a group. This Event Organizer helps various events which usually have problems in advertising their event to the public or selling the tickets of their events.

With so many form builder whose using AJAX technology, this application is made to help event organizer to advertise an event into a particular website. Basically, this application is focusing in form builder which will be used by event organizer as a mode to create a form of events selling tickets.

Mechanism of this application start by an event organizer who needs to sign up first to register his/her personal datas by filling in the data form. After the event organizer is registered, then the event organizer will have an access to log in. Event organizer can create an event info which contain by the events details that he/she made. This page will be seen by the guest to know what is the event details. After that, event organizer will create an online selling ticket form using form builder, which have a drag and drop system that will make it easier for the event organizer to create a form without needed to understand programing language. This is the page where guest can buy ticket online.

Keywords : *Event organizer, event, form builder, guest, online ticket, AJAX, drag and drop.*

Daftar Isi

| | |
|---|-----------|
| Pernyataan Orisinalitas Laporan Penelitian | i |
| Kata Pengantar | ii |
| Abstrak..... | iii |
| Abstract..... | iv |
| Daftar Isi..... | v |
| Daftar Gambar | viii |
| Daftar Table | x |
| Daftar Simbol | xi |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Perumusan Masalah..... | 1 |
| 1.3 Tujuan..... | 1 |
| 1.4 Batasan Masalah..... | 2 |
| 1.5 Sistematika Penulisan | 2 |
| 1.6 Time Schedule..... | 3 |
| BAB II LANDASAN TEORI | 4 |
| 2.1.1 Sejarah Aplikasi Web | 4 |
| 2.1.2 PHP | 5 |
| 2.1.3 AJAX | 6 |
| 2.1.4 My SQL | 10 |
| 2.1.5 CSS (Cascading Style Sheets) | 11 |
| BAB III ANALISA DAN PEMODELAN..... | 12 |
| 3.1 Pendahuluan | 12 |
| 3.1.1 Identifikasi..... | 12 |
| 3.1.2 Overview Sistem..... | 12 |
| 3.2 Keputusan Desain Perangkat Lunak Secara Keseluruhan | 13 |
| 3.2.1 Usecase | 13 |

| | | |
|-------------------------------------|-------------------------------|----|
| 3.2.2 | <i>Activity diagram</i> | 21 |
| 3.2.3 | Sequence Diagram | 30 |
| 3.2.4 | Class Diagram | 38 |
| 3.2.5 | Entity Relationship Diagram | 39 |
| 3.2.6 | Perancangan User Interface | 40 |
| 3.2.7 | Site Map | 46 |
| 3.3 | Bisnis Proses | 46 |
| BAB IV PERANCANGAN DAN IMPLEMENTASI | | 48 |
| 4.1 | <i>Event organizer</i> | 48 |
| 4.1.1 | Sign Up | 48 |
| 4.1.2 | Login | 48 |
| 4.1.3 | Main Menu | 49 |
| 4.1.4 | <i>Create Event Form</i> | 50 |
| 4.1.5 | <i>Event Info Form</i> | 50 |
| 4.1.6 | <i>Form Builder</i> | 51 |
| 4.1.7 | Ticketing Form | 52 |
| 4.2 | Admin Side | 52 |
| 4.2.1 | Main Menu | 52 |
| 4.2.2 | <i>Manage EO</i> | 53 |
| 4.2.3 | <i>Manage Input</i> | 53 |
| 4.2.4 | <i>Create New Input</i> | 54 |
| 4.2.5 | Manage Advertisement | 54 |
| 4.3 | Penjelasan AJAX | 55 |
| BAB V PENGUJIAN | | 57 |
| 5.1 | Black-box Testing | 57 |
| 5.2 | White Box Testing | 59 |
| 5.2.1 | Pengujian Method AddNewMember | 60 |
| 5.2.2 | Pengujian Method Login | 60 |
| 5.2.3 | Pengujian Method GetEvent | 60 |

| | |
|--|------|
| 5.2.4 Pengujian Method NewEvent | 61 |
| 5.2.5 Pengujian Method ShowEventInfo..... | 61 |
| 5.2.6 Pengujian Method ProccessDelete..... | 61 |
| 5.2.7 Pengujian Method Proccess | 62 |
| 5.2.8 Pengujian Method UpdateFontType | 62 |
| 5.2.9 Pengujian Method UpdateFontColor | 62 |
| 5.2.10 Pengujian Method UpdateFontSize | 63 |
| 5.2.11 Pengujian Method UpdateFormProperty | 63 |
| 5.2.12 Pengujian Method UpdateInfoProperty..... | 63 |
| 5.2.13 Pengujian Method LoadTipeInfo..... | 64 |
| 5.2.14 Pengujian Method SaveForm | 64 |
| 5.2.15 Pengujian Method Deleteltem | 64 |
| 5.2.16 Pengujian Method Addltem | 64 |
| 5.2.17 Pengujian Method NewInputValue | 65 |
| 5.2.18 Pengujian Method DeleteEO | 65 |
| 5.2.19 Pengujian Method DeletelInput | 65 |
| 5.3 Kesimpulan Pengujian..... | 66 |
| BAB VI KESIMPULAN DAN SARAN..... | 67 |
| 6.1 Kesimpulan..... | 67 |
| 6.2 Saran..... | 67 |
| Daftar Pustaka | xiii |
| Lampiran | xiv |

Daftar Gambar

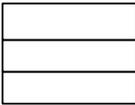
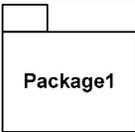
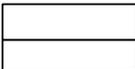
| | |
|--|----|
| Gambar II-1 visualisasi AJAX | 8 |
| Gambar III-1 Usecase Diagram Form Builder..... | 14 |
| Gambar III-2 Activity diagram Proses Sign Up | 22 |
| Gambar III-3 Activity diagram Proses Login | 23 |
| Gambar III-4 Activity diagram Proses Submit Form..... | 24 |
| Gambar III-5 - Activity diagram Proses Tambah Form | 25 |
| Gambar III-6 - Activity diagram Proses Ubah Form | 25 |
| Gambar III-7 - Activity diagram Proses Hapus Form | 26 |
| Gambar III-8 - Activity diagram Proses Tambah Event Info..... | 27 |
| Gambar III-9 - Activity diagram Proses Ubah event info | 27 |
| Gambar III-10 - Activity diagram Proses Hapus Event Info | 28 |
| Gambar III-11 - Activity diagram Proses Tambah EO (user) | 29 |
| Gambar III-12 - Activity diagram Proses Hapus EO | 29 |
| Gambar III-13 - Sequence Diagram Sign Up | 30 |
| Gambar III-14 - sequence diagram login | 31 |
| Gambar III-15 - sequence diagram submit | 32 |
| Gambar III-16 - sequence diagram Atur Event Info | 33 |
| Gambar III-17 - sequence diagram atur form..... | 35 |
| Gambar III-18 - sequence diagram atur EO..... | 37 |
| Gambar III-19 - class diagram | 38 |
| Gambar III-20 Entity Relationship Diagram | 39 |
| Gambar III-21 - Desain Antar muka | 40 |
| Gambar III-22 - Desain Antar Muka(2)..... | 40 |
| Gambar III-23 - Desain Antar Muka(3)..... | 41 |
| Gambar III-24 - Desain Antar Muka(4)..... | 41 |
| Gambar III-25 - Desain Antar Muka(5)..... | 42 |
| Gambar III-26 - Desain Antar Muka (6)..... | 43 |
| Gambar III-27 - Desain Antar Muka (7)..... | 44 |
| Gambar III-28 - Desain Antar Muka (8)..... | 45 |
| Gambar III-29 - Site Map | 46 |
| Gambar IV-1 Interface Register | 48 |
| Gambar IV-2 Interface Login..... | 49 |
| Gambar IV-3 Interface Main Menu..... | 49 |

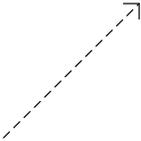
| | |
|--|----|
| Gambar IV-4 Interface Create New Event | 50 |
| Gambar IV-5 Interface Event Info Form..... | 51 |

Daftar Tabel

| | |
|--|-------------------------------------|
| Tabel I-1 Time Schedule | Error! Bookmark not defined. |
| Tabel III-1 <i>Usecase</i> : Ajukan Form | 15 |
| Tabel III-2 <i>usecase</i> : Menambah Info | 15 |
| Tabel III-3 <i>usecase</i> : Mengubah Data Info | 16 |
| Tabel III-4 <i>usecase</i> : Menghapus Data Info | 17 |
| Tabel III-5 <i>usecase</i> : Menambah Form | 18 |
| Tabel III-6 <i>usecase</i> : Mengubah Form | 18 |
| Tabel III-7 <i>usecase</i> : Menghapus Form | 19 |
| Tabel III-8 <i>usecase</i> : Menghapus EO | 21 |

Daftar Simbol

| SIMBOL | GAMBAR | KETERANGAN |
|------------------|---|---------------|
| Usecase Diagram |  | Actor / User |
| |  | Usecase |
| Activity diagram |  | Kondisi Awal |
| |  | Kondisi Akhir |
| |  | State |
| |  | Pilihan |
| |  | Alur |
| |  | Transition |
| Class Diagram |  | Class |
| |  | Package |
| |  | Interface |

| | | |
|--|---|--------------------------|
| |  | Extends / generalization |
| |  | Binary association |
| |  | dependency |

BAB I PENDAHULUAN

Di bawah ini akan dijelaskan persyaratan produk aplikasi yang dibangun, yang terdiri dari tujuan dari pembuatan aplikasi, ruang lingkup proyek, definisi, akronim, dan singkatan, referensi, *overview*, perspektif produk, fungsi produk, karakteristik pengguna, batasan-batasan, asumsi dan ketergantungan, dan penundaan persyaratan.

1.1 Latar Belakang

Pada saat ini, dunia hiburan di tanah air semakin marak. Beberapa pihak menjadikan hal ini sebagai peluang bisnis yang menguntungkan. Seperti banyaknya *event-event organizer* yang bermunculan, dan berlomba untuk menyelenggarakan acara membuat persaingan bisnis ini cukup ketat.

Salah satu kendala yang dihadapi oleh pihak penyelenggara dan konsumen adalah proses penjualan tiket. Sistem yang masih banyak kita kenal adalah penjualan tiket di counter-counter dan sistem penitipan tiket di beberapa tempat yang bersangkutan. Tetapi hal ini tetap menyulitkan konsumen, karena konsumen diharuskan pergi ke tempat penjualan tiket, yang belum tentu dekat dengan daerah tempat tinggal konsumen.

Melihat dari hal-hal di atas (penulis) bermaksud membuat suatu web ticket box yang berisikan event info builder dan *registration* form builder. Di mana pihak penyelenggara dapat menjual tiket acara melalui web tersebut dengan mudah. Konsumen pun akan merasakan lebih efisien tanpa harus pergi ke tempat penjualan tiket. Pada web ini pun pihak sponsor tidak akan merasa dirugikan, karena event organizer dapat memasukkan iklan sponsor ke dalam web.

1.2 Perumusan Masalah

Perumusan yang terdapat pada program ini adalah:

1. Bagaimana menggunakan bahasa pemrograman AJAX dalam PHP?
2. Bagaimana membuat suatu halaman web yang dinamis sehingga mudah untuk dimengerti dan digunakan oleh user?
3. Bagaimana membuat form builder yang diimplementasikan pada suatu web?

1.3 Tujuan

Dengan kemajuan teknologi yang ada pada saat ini, maka dibuatlah aplikasi Pemanfaatan Page Builder Dan Form Builder Untuk Implementasi Aplikasi Ticket Box

Menggunakan Teknologi Ajax ini untuk mempermudah system penjualan tiket acara secara online, sehingga pengunjung atau guest lebih mudah untuk mendapatkan tiket. Melalui web ini juga, diharapkan dapat mempermudah *event organizer* dalam mempublikasikan acara yang akan diselenggarakan.

1.4 Batasan Masalah

Batasan masalah yang terdapat pada program ini adalah:

1. Pembuatan aplikasi program menggunakan bahasa pemrograman PHP dan Ajax.
2. Aplikasi ini menekankan pada pembuatan form builder untuk mempermudah user dalam membuat event info.
3. Pada aplikasi ini tidak membahas masalah sekuritas dan pembayaran online.

1.5 Sistematika Penulisan

Sistematika penulisan terdiri dari enam bab, yaitu :

BAB I PENDAHULUAN

Bab ini memberikan gambaran mengenai Latar Belakang Masalah, Penjelasan Instansi, Perumusan Masalah, Tujuan, Pembatasan Masalah, dan Sistematika Penulisan.

BAB II LANDASAN TEORI

Bab ini berisi uraian tentang teori-teori dasar yang menjadi landasan pembahasan dari seluruh materi yang ada, baik yang berasal dari buku teks maupun berasal dari situs web.

BAB III ANALISA DAN PERANCANGAN

Bab ini berisi gambaran tentang proses yang mencakup struktur organisasi, proses bisnis, perancangan fitur-fitur, perancangan *Usecase* beserta langkah-langkahnya dan merancang sistem basis data sebagai solusi untuk permasalahan tersebut. Dan juga pembahasan mengenai cara pembuatan custom object untuk pembuatan User Interface (UI) aplikasi.

BAB IV PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi tentang Kumpulan screenshot dari proyek yang dibuat beserta penjelasan dari tiap fungsi (method) utama yang dibuat.

BAB V PENGUJIAN

Bab ini berisi tentang Laporan dari pengujian tiap class/fungsi/method yang dibuat (whitebox testing) dan laporan

dari kuisoner (blackbox testing) yang diberikan pada minimal 20 responden (beserta bukti identitas diri).

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi tentang pengetahuan yang didapat setelah mengerjakan karya ilmiah ini, baik berupa penegasan/pembuktian atau pengetahuan baru.

1.6 Time Schedule

Proyek web ini direncanakan akan selesai dalam waktu tujuh bulan.

Tabel I-1 Time Schedule

| | Tahap Pengerjaan Proyek | Januari | | | | Februari | | | | Maret | | | | April | | | | Mei | | | | Juni | | | | Juli | | | |
|---|---|---------|---|---|---|----------|---|---|---|-------|---|---|---|-------|---|---|---|-----|---|---|---|------|---|---|---|------|---|---|---|
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | Pengumpulan Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Penyusunan laporan Bab I | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Penyusunan Laporan Bab II | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Perancangan desain perangkat lunak | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Penyusunan laporan bab III, pembuatan program | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | Pembuatan program dan database | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

BAB II LANDASAN TEORI

Dalam mengembangkan program ini dibutuhkan suatu sistem yang dapat menangani user dalam menggunakan aplikasi web ini.

Beberapa hal yang mendukung kegiatan analisis dan perancangan sistem yang dikembangkan diantaranya adalah mengenai penggunaan bahasa pemrogramannya, yakni bahasa pemrograman PHP dan AJAX, serta database yang akan menampung data aplikasi ini.

2.1.1 Sejarah Aplikasi Web

Seiring dengan perkembangan internet pada awal tahun 1990-an dan di ditemukannya HTTP (*Hyper Text Transfer Protokol*) yang digunakan untuk mengirimkan data di internet, sejak itulah sejarah aplikasi web dimulai. Pada waktu itu tipe dokumen standar yang digunakan di internet adalah HTML (*Hyper Text Markup Language*). HTML adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser (Sebuah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh webserver). HTML tidak dirancang untuk membuat sebuah aplikasi web yang kompleks melainkan hanya untuk menampilkan content dan formatnya dalam bentuk text dan image dalam bentuk yang statis.

Aplikasi web adalah sebuah aplikasi yang menggunakan kemampuan webserver untuk memproses data dan mengirimkannya ke *user* melalui network/internet. Berbeda dengan aplikasi desktop, dimana harus melakukan instalasi di sisi client, untuk dapat menggunakan aplikasi tersebut. Aplikasi web hanya membutuhkan sebuah web browser yang digunakan hanya untuk menampilkan data yang diterima dari server.

Karena keterbatasan diatas akhirnya muncullah beberapa teknologi yang dapat membuat aplikasi web lebih dinamis. Teknologi ini dapat dikelompokkan menjadi dua jenis

1. Teknologi *Client-Side*. Teknologi ini membuat aplikasi web menjadi dinamis dengan melakukan manipulasi data disisi *user*. Contoh teknologi ini adalah JavaScript, Java Applet, Flash dan lain-lain. Keuntungan yang ditawarkan teknologi ini adalah mempermudah dan memberikan “keinteraktifan” pada web *user* interface.
2. Teknologi *Server-Side*. Teknologi ini menawarkan ke-interaktifan data dari sisi server, karena semua data di manipulasi di sisi server. Client hanya menerima output

dari server dalam bentuk HTML biasa. Contoh teknologi ini adalah *CGI, PHP, ASP, JSP, ColdFusion* dan lain lain.

Untuk mendapatkan aplikasi web yang baik biasanya programmer menggunakan kedua teknologi diatas, supaya dapat menghasilkan sebuah aplikasi web yang powerful, karena kedua teknologi tersebut menawarkan keuntungan yang sangat berbeda antara satu dan lainnya.

Keuntungan aplikasi web:

- Aplikasi web lebih mudah dan murah untuk digunakan. Dengan aplikasi web, kita dapat mengurangi biaya untuk mengimplementasi/menginstall aplikasi disisi *client*.
- Aplikasi web lebih mudah dan murah untuk di upgrade. Biaya perawatan dan upgrade lebih murah karena hanya mengubah di sisi server.
- Aplikasi web lebih fleksibel. Dengan aplikasi web, kita tidak perlu memikirkan masalah sistem operasi, karena secara default sebuah sistem operasi sudah dilengkapi *web browser*.

2.1.2 PHP

PHP adalah bahasa pemrograman *script* yang banyak dipakai saat ini. PHP banyak dipakai untuk memprogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessing/Form Interpreter*. Dengan perilsan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

Kelebihan PHP adalah :

- a) Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b) Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
- c) Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
- d) Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena referensi yang banyak.
- e) PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

Dalam PHP juga dikenal 8 tipe data yaitu, *Integer, Double, Boolean, String, Object, Array, Null, dan Resource*.

2.1.3 AJAX

AJAX merupakan sebuah kombinasi dari beberapa teknologi ((X)HTML, CSS, JavaScript, *Document Object Model* DOM) dan XMLHttpRequest) yang membuat sebuah halaman web dapat di-refresh dari server tanpa harus me-refresh seluruh bagian dari halaman web, sehingga meningkatkan *experience* seorang user ketika menggunakannya. AJAX bukanlah sebuah teknologi, tetapi sebuah teknik dalam aplikasi web.

JavaScript

JavaScript dibuat oleh Brendan Eich (netscape) dan dimasukkan kedalam versi 2.0 dari *web browser* Netscape. JavaScript dikembangkan dari bahasa LiveScript yang dikembangkan oleh Netscape untuk digunakan pada versi awal dari web browser mereka. Kode dari JavaScript ditempatkan dalam kode HTML dari halaman web diinterpretasikan dan dieksekusi oleh *web browser* ketika halaman web ditampilkan.

Asynchronous JavaScript and XMLHTTP, atau disingkat AJAX, adalah suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web interaktif. Tujuannya adalah untuk memindahkan sebagian besar interaksi pada komputer *web surfer*, melakukan pertukaran data dengan server di belakang layar, sehingga halaman web tidak harus dibaca ulang secara keseluruhan setiap kali seorang pengguna melakukan perubahan. Hal ini akan meningkatkan interaktivitas, kecepatan, dan *usability*. AJAX merupakan kombinasi dari:

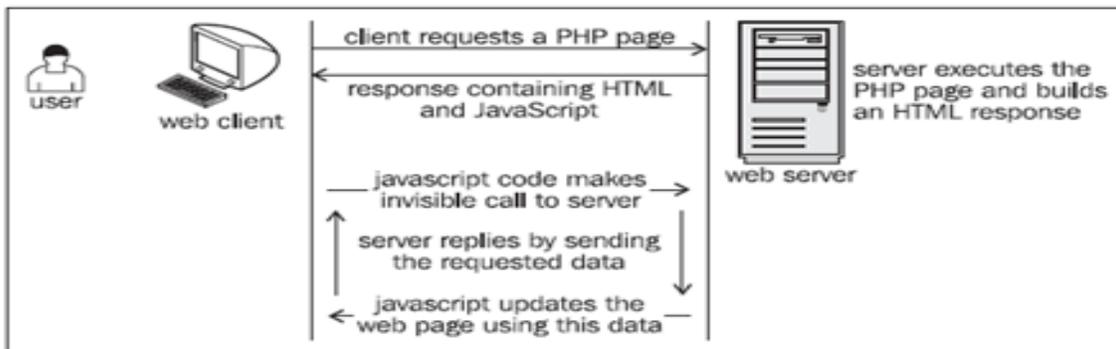
- DOM yang diakses dengan *client side scripting language*, seperti VBScript dan implementasi ECMAScript seperti JavaScript dan JScript, untuk menampilkan secara dinamis dan berinteraksi dengan informasi yang ditampilkan. DOM adalah sebuah dokumen yang merepresentasikan dokumen XML ataupun halaman web sebagai sebuah kumpulan objek-objek yang saling berelasi yang dapat dimanipulasi secara dinamis. Halaman web terstruktur sebagai sebuah hirarki tree yang terdiri dari root node, parent, dan cabang-cabangnya. Setiap elemen HTML direpresentasikan oleh sebuah cabang yang dapat diakses melalui JavaScript.
- Objek XMLHTTP dari Microsoft atau XMLHttpRequest yang lebih umum di implementasikan pada beberapa browser. Objek ini berguna sebagai kendaraan pertukaran data asinkronus dengan *web server*. Pada beberapa *framework* AJAX, element HTML IFrame lebih dipilih daripada XMLHTTP atau XMLHttpRequest untuk melakukan pertukaran data dengan *web server*.
- XML umumnya digunakan sebagai dokumen *transfer*, walaupun format lain juga memungkinkan, seperti HTML, *plain text*. XML dianjurkan dalam pemakaian teknik AJAX karena kemudahan akses penanganannya dengan memakai DOM
- JSON dapat menjadi pilihan alternatif sebagai dokumen *transfer*, mengingat JSON adalah JavaScript itu sendiri sehingga penanganannya lebih mudah

Pada intinya AJAX itu merupakan gabungan beberapa teknologi yang bertujuan untuk menghindari *page reload*. Dengan menghindari *page reload*, kita dapat menghindari paradigma *click-and-wait* serta memberikan sebuah fitur yang cukup kompleks pada *website*

seperti validasi data secara realtime, *drag and drop* dan fitur-fitur lain yang belum dimiliki web biasa.

Dengan AJAX, suatu aplikasi web dapat mengambil data kemudian diolah di client melalui *request asynchronous* HTTP yang diinisiasi oleh Javascript, sehingga dapat mengupdate bagian-bagian tertentu dari web tanpa harus memanggil keseluruhan halaman web. Request ini dapat dieksekusi dalam beberapa cara dan beberapa format transmisi data. Dikombinasikannya cara pengambilan data remote dengan interaktivitas dari *Document Object Model* (DOM) telah menghasilkan generasi terbaru dari aplikasi web yang mengebrak aturan-aturan tradisional tentang apa yang dapat terjadi di dalam web.

AJAX memberikan keseimbangan lebih baik antara fungsionalitas client dengan fungsionalitas server ketika mengeksekusi aksi yang diminta oleh *user*. Hingga saat ini, fungsionalitas di sisi client dan server dipandang sebagai fungsionalitas yang terpisah yang bekerja hanya satu dalam suatu waktu untuk merespon aksi dari *user*. Dengan adanya AJAX, dihasilkan solusi untuk menyeimbangkan loading web antara client dan server dengan cara memperkenankan komunikasi secara background ketika *user* bekerja dengan page saat itu. Gambar di bawah ini adalah visualisasi dari apa yang terjadi ketika halaman web dengan AJAX diminta oleh *user*:



Gambar II-1 visualisasi AJAX

AJAX dibangun dari beberapa teknologi yang telah disediakan oleh web browser modern, seperti Mozilla Firefox, Internet Explorer, atau Opera, sehingga client tidak perlu menginstall modul extra untuk menjalankan *website* AJAX. AJAX ini dikonstruksi dari:

- JavaScript adalah unsur terpenting dari AJAX, yang dapat membangun fungsionalitas client site. Dalam fungsi-fungsi JavaScript akan banyak digunakan Document Object Model (DOM) untuk memanipulasi bagian dari halaman HTML.
- Objek XMLHttpRequest memperkenalkan JavaScript untuk mengakses server secara asynchronous, sehingga *user* dapat melanjutkan pekerjaan, sementara waktu proses dilakukan di background. Mengakses server singkatnya adalah membuat request HTTP sederhana untuk suatu file atau script yang terletak di server. Request HTTP mudah dibuat dan tidak menyebabkan permasalahan yang berhubungan dengan firewall.
- Teknologi server-side diperlukan untuk menangani request yang datang dari client JavaScript.

Berikut ini adalah beberapa keuntungan dari aplikasi web berbasis AJAX:

- Memungkinkan untuk membuat *website* dan aplikasi web yang lebih baik dan lebih responsif.
- Kepopularitasannya mendorong perkembangan pengkodean yang membantu developer untuk menghindari pembuatan ulang dalam melakukan pekerjaan yang sudah umum.
- Mendayagunakan teknologi yang telah ada.
- Mendayagunakan skill developer yang telah ada.
- Fitur-fitur dari AJAX menyatu dengan baik dengan fungsionalitas yang telah disediakan oleh browser web.

Skenario umum dimana AJAX dapat digunakan antara lain:

- Validasi form server-side yang dilakukan dengan segera, sangat berguna di kondisi yang tidak memungkinkan pengiriman keseluruhan data ke client yang dibutuhkan untuk melakukan validasi ketika halaman di-load di awal mula.
- Membuat *chat online* sederhana yang tidak membutuhkan library eksternal seperti *Java Runtime Machine* atau Flash
- Membuat aplikasi dengan fungsionalitas seperti Google Suggest.
- Mendayagunakan teknologi yang telah ada secara lebih efektif. Misalnya pembuatan grafik secara real time menggunakan Scalable Vector Graphics (SVG), atau membuat list *drag-and-drop* sederhana.
- Pembuatan data grid yang responsif yang mengupdate database *server-side* secara *on-the-fly*.

- Membuat aplikasi yang membutuhkan update secara real time dari berbagai sumber eksternal, misalnya RSS.

Beberapa kemungkinan permasalahan dengan AJAX antara lain:

- Karena alamat halaman tidak berubah ketika proses bekerja, maka tidak mudah untuk membuat bookmark suatu halaman yang berbasis AJAX.
- Search Engine mungkin tidak dapat meng-indexing seluruh bagian dari situs dengan aplikasi AJAX.
- Tombol Back di browser, tidak memberikan hasil yang sama dengan aplikasi web klasik, karena seluruh aksi terjadi di halaman yang sama.
- JavaScript dapat di-*disable* di sisi client, sehingga aplikasi AJAX tidak berfungsi.

Membuat request asynchronous dengan AJAX:

Sebuah request asynchronous pada AJAX dilakukan dengan menggunakan objek XMLHttpRequest dan fungsi ataupun script yang digunakan untuk menangani respon dari request asynchronous yang dilakukan dengan menggunakan XMLHttpRequest tersebut. Untuk melakukan request secara asynchronous kita memerlukan :

1. Dokumen HTML sebagai halaman web akan ditampilkan pada web browser.
2. File JavaScript untuk meng-*instance* objek XMLHttpRequest untuk mengkonstruksi request asynchronous dan mengirimkan request asynchronous ke server.
3. File, yang dapat berupa file teks, dokumen HTML, dokumen PHP ataupun dokumen XML yang memberikan nilai balikan informasi yang diperlukan.
4. Fungsi callback yang digunakan untuk menangani respon dari server dari request asynchronous yang diminta dengan menggunakan data hasil request tersebut pada aplikasi dan menempatkannya pada element HTML yang diinginkan.

Seperti halnya DHTML, LAMP, atau SPA, AJAX bukanlah teknologi spesifik, melainkan merupakan gabungan dari teknologi yang dipakai bersamaan. Bahkan, teknologi turunan/komposit yang berdasarkan AJAX, seperti AFLAX sudah mulai bermunculan.

2.1.4 My SQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai

perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

2.1.5 CSS (Cascading Style Sheets)

Cascading Style Sheets (CSS) adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh *World Wide Web Consortium (W3C)*.

CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada struktur isi.

CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis-suara atau pembaca layar), dan juga alat pembaca braille. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

BAB III ANALISA DAN PEMODELAN

3.1 Pendahuluan

Pada bab ini akan dijelaskan bagaimana pengembangan produk berdasarkan ruang lingkup proyek serta proses-proses yang ada.

3.1.1 Identifikasi

Aplikasi ticket box ini dibuat untuk mempermudah dan membantu sebuah *event organizer* untuk memasarkan event yang dibuatnya. *Event organizer* dapat memberikan info-info yang penting dalam web tersebut, kemudian *event organizer* tersebut dapat dengan mudah membuat form yang wajib diisi oleh pengunjung(guest) yang hendak melakukan pembelian tiket. Dalam hal ini, pengunjung(guest) dalam acara yang diselenggarakan tersebut dapat memperoleh tiket dengan mudah, tanpa harus pergi ke counter tiket.

3.1.2 Overview Sistem

Aplikasi dengan judul Pemanfaatan Page Builder dan Form Builder untuk Implementasi Aplikasi Tiket Box menggunakan Teknologi Ajax ini memiliki fitur-fitur berikut:

➤ Admin

- Sign Up

Fitur ini digunakan untuk calon admin yang hendak melakukan pendaftaran sebagai admin. Dalam hal ini, calon admin tersebut harus mendapatkan hak akses dari admin lain yang telah terdaftar.

- Login

Fitur login ini digunakan untuk admin masuk ke dalam menu, sehingga memiliki hak akses menggunakan fitur-fitur lain yang ada pada web ini.

- Mengatur Event Info

Fitur ini digunakan admin untuk mengatur event info yang dimiliki oleh *event organizer* (user). Admin dapat menambah, mengubah, dan menghapus event yang dibuat oleh EO, jika event tersebut dianggap melanggar aturan yang berlaku.

Admin juga dapat membuat event yang dibuat oleh admin sendiri.

- Mengatur Form Builder

Fitur ini digunakan admin untuk mengatur form yang dimiliki oleh *event organizer* (user). Admin dapat menambah, mengubah, dan menghapus form yang dimiliki user.

Admin dapat membuat form, untuk event yang dibuat oleh admin.

- Mengatur *event organizer* (user)
Fitur ini digunakan admin untuk mengatur *event organizer* atau user. Admin dapat menambah dan menghapus EO yang sudah terdaftar.
- *Event organizer* (user)
 - Sign up
Calon user akan menjadi user jika telah melakukan proses sign up dengan mengisi data-data yang valid.
 - Login
User dapat mengakses aplikasi web ini dengan melakukan login.
 - Mengatur *event info*
Pada fitur ini, user akan menjelaskan tentang acara apa yang akan diselenggarakan. User memasukkan nama acara, waktu, tempat, tanggal, dan sponsor pada fitur ini.
 - Mengatur Form Builder
Fitur ini digunakan untuk membuat form pembelian yang akan digunakan guest membeli tiket acara. Form builder ini memiliki system drag and drop yang memudahkan user dalam membuat form.
- Guest
 - View event info
Guest melihat detail event info yang telah dipublikasikan oleh *event organizer*
 - Submit form
Guest melakukan pembelian tiket acara untuk menghadiri event yang diselenggarakan oleh EO.

3.2 Keputusan Desain Perangkat Lunak Secara Keseluruhan

Sub bab ini akan memperlihatkan analisa pembuatan aplikasi *website* ini secara keseluruhan. Analisa tersebut menggunakan UML (Unified Modeling Language) antara lain, metode *Usecase*, metode *Activity diagram*, metode Class Diagram, dan membahas tentang Entity Relationship Diagram (ERD).

3.2.1 Usecase

Sebuah *usecase* menggambarkan suatu urutan interaksi antara satu atau lebih actor dan system. Dalam fase requirements, model *usecase* menggambarkan system sebagai sebuah kotak hitam dan interaksi antara actor dan system dalam suatu bentuk naratif, yang terdiri dari input user dan respon-respon system.