BAB II

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

2.1.1 Sejarah Instansi

Pesawat terbang adalah tranportasi yang mempunyai peran penting dalam ekonomi maupun pertahanan suatu negara,apalagi mengingat bahwa Indonesia merupakan negara kepulauan dengan kondisi geografis yang sulit ditembus dengan alat transportasi biasa. Berdasarkan kondisi tersebut, mucul pemikiran bahwa sebagai negara kepulauan, Indonesia harus mengembangkan industri maritim dan penerbangan. Kondisi inilah yang memicu lahirnya industri penerbangan di Indonesia.

Segera setelah kemerdekaan Indonesia diproklamasikan tahun 1945, kesempatan Indonesia untuk mewujudkan impian membuat pesawat terbang mereka sendiri terbuka lebar. Sejak saat itu, Indonesia telah mulai sadar bahwa sebagai Negara kepulauan, Indonesia akan selalu membutuhkan transportasi udara untuk memperlancar usaha pemerintahah, pengembangan ekonomi dan pertahanan nasional.

Tahun 1946, biro perencanaan dan konstruksi dibangun di TRI-Udara atau TNI-AU. Disponsori oleh Wiweko Supono, Nurtanio Pringgadisurjo, dan Sumarsono, tempat perakitan dibangun di Magetan, dekat Madiun, Jawa Timur. Berasal dari

material sederhana, sebuah pesawat layang (NWG-1) berhasil dibuat. Pembuatan pesawat ini juga melibatkan Tossin, didukung oleh Ahmad, dkk. Dengan 6 pesawat, pesawat-pesawat tersebut dimanfaatkan untuk mengembangkan ketertarikan Indonesia dan pada saat yang sama memperkenalkan dunia penerbangan kepada calon pilot yang telah bersiap untuk mengikuti latihan penerbangan di India.

Sejalan dengan penghargaan yang telah diperoleh dan dalam rangka menyediakan pengembangan yang lebih cepat, dengan Surat Ketetapan Kepala Staff Angkatan Udara no.488 Agustus 1960, Lembaga Persiapan Industri Penerbangan (LAPIP) dibentuk. Diresmikan 16 Desember 1961, badan ini mempunyai fungsi untuk menyiapkan pembentukan industri penerbangan dengan kemampuan untuk menyokong aktivitas penerbangan nasional di Indonesia.

Dengan Peraturan Pemerintah No.12 5 April 1976, persiapan industri penerbangan telah dibuat. Melalui registrasi ini, semua aset, fasilitas, dan potensi yang tersedia diakumulasi meliputi aset Pertamina, Divisi ATTP yang telah bersiap untuk berdirinya sebuah industri penerbangan dengan asset-aset LIPNUR, Angkatan Udara Indonesia, sebagai modal dasar untuk industri penerbangan. Modal dasar ini diharapkan mampu menyumbangkan pertumbuhan industri penerbangan yang mampu menjawab semua tantangan.

Pada 26 april 1976, berdasarkan Akte Notaris no 15, di Jakarta, PT. Industri Pesawat Terbang Nurtanio secara resmi didirikan oleh Dr. BJ. Habibie sebagai

Presiden Direktur. Ketika fasilitas fisik industri ini lengkap, pada Agustus 1976 Presiden Suharto meresmikan industri penerbangan ini.

Pada 11 Oktober 1985, PT. Industri Pesawar Terbang Nurtanio dihapus dan diganti dengan PT. Industri Pesawat Terbang Nusantara atau IPTN.

Selama 24 tahun melayani, IPTN dengan sukses mengembangkan teknologi penerbangan yang canggih dan mutakhir, terutama mengadopsi teknologi barat ke Indonesia. IPTN khususnya telah menguasai desain pesawat terbang, pengembangan, dan usaha manufaktur komuter daerah kecil dan menengah.

PT. IPTN sekarang melayani dengan kelebihannya dalam bidang teknik, dengan menawarkan desain untuk menguji aktifitas layanan, pembuatan komponen pesawat dan bukan pesawat, serta layanan purna jual.

Oleh karena itu, nama IPTN telah berubah menjadi PT. DIRGANTARA INDONESIA atau *Indonesian Aerospace* atau disingkat IAe dan secara sah telah diresmikan oleh Presiden Republik Indonesia, KH. Abdurrahman Wahid, di Bandung pada tanggal 24 Agustus 2009.

2.1.2 Logo Instansi



Gambar II.1 Logo PT. Dirgantara Indonesia

I. Bentuk

Bentuk logo PT DIRGANTARA INDONESIA terdiri dari:

1. Lingkaran

   Menggambarkan lingkaran dunia, memberikan makna aktifitas usaha yang mencakup pasar global.

2. Sayap

   Berjumlah 3 (tiga) buah dengan ukuran yang berbeda. Menggambarkan kekuatan usaha untuk mencapai tujuan yang lebih tinggi:

   a. Sayap besar, menggambarkan Bisnis Inti (*Core Business*)

b. Sayap Sedang, menggambarkan Bisnis Plasma (*Non – Core Business*)

c. Sayap kecil, menggambarkan Korporasi (*Corporate*)

Ketiganya menjalin persatuan dan kesatuan menuju ke atas dalam sudut kecondongan / elavasi 45^{o} yang berarti arah yang seimbang dan optimal dalam pencapaian target.

II. Warna

Warna Logo PT DIRGANTARA INDONESIA adalah Biru (*cyan* 100% dan *magenta* 100%) yang memiliki makna warna dirgantara, kemantapan dan kekuatan. Ini mencerminkan tekad untuk berusaha semaksimal mungkin sesuai kompetensi dan etika usaha.

III. Tulisan

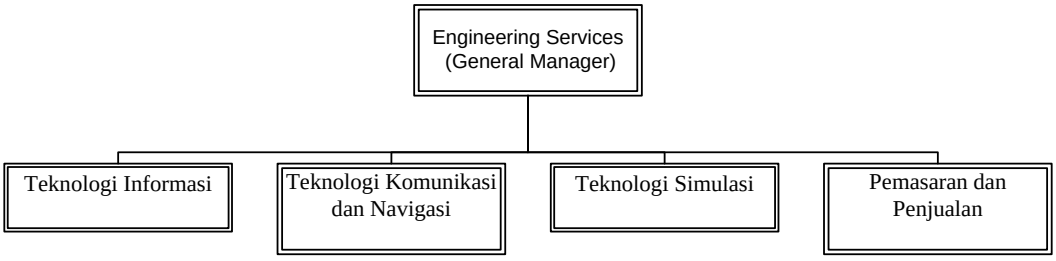
- Tulisan Logo PT DIRGANTARA INDONESIA adalah dalam huruf kapital Arial Narrow Bold, berwarna Biru (*cyan* 100% dan *magenta* 100%).
- Tulisan INDONESIA AEROSPACE (IAe) adalah dalam huruf capital Arial Narrow Bold, berwarna Biru (*cyan* 100% dan *magenta* 100%) merupakan nama dalam korespondensi internasional.

2.1.3 Badan Hukum Instansi

Status PT. Dirgantara Indonesia (Persero) tidak dapat dikategorikan sebagai BUMN yang bergerak dalam bidang publik sebagaimana yang dimaksud oleh pasal 1 angka 4 UU BUMN. Karena dalam PT. Dirgantara Indonesia (Persero) seluruh modalnya terbagi dalam saham.

Mengenai status hukum PT. Dirgantara Indonesia (Persero) dapat dilihat dalam Berita Negara mengenai persetujuan akte perubahan anggaran dasar Perseroan Terbatas Tanggal 25 Oktober 2005 No. 85 oleh Depkum dan HAM RI. Sesuai dengan Keputusan Menkum dan HAM RI No.C-04670.HT.01.04 tahun 2005 dalam Pasal 1 angka 1 disebutkan Perseroan Terbatas ini bernama Perusahaan Perseroan (Persero) PT. Dirgantara Indonesia disingkat PT. DI (Persero). Kemudian dalam Pasal 4 Ayat (2) dan Ayat (3) disebutkan pemegang saham dan PT. Dirgantara Indonesia (Persero) adalah Menteri Negara Badan Usaha Milik Negara RI cq. Negara Republik Indonesia dan Menteri Keuangan RI cq. Negara Republik Indonesia.

2.1.4 Struktur Organisasi dan *Job Description*



Gambar II.2 Struktur organisasi

Tugas dan fungsi dari struktur organisasi *Engineering Services* PT Dirgantara Indonesia.

a. *General Manager*

   Bertugas untuk mengatur seluruh aktifitas satuan usaha engeneering services PT. Dirgantara Indonesia dan *General Manager* berfungsi sebagai pimpinan bagi manager-manager yang berada dibawah wewenangnya.

b. Pemasaran dan Penjualan, bertugas untuk memasarkan produk-produk yang telah diproduksi oleh Pusat Perancangan, Teknologi Komunikasi, Teknologi Informasi dan Teknologi Simulasi, dan bagian Pemasaran dan Penjualan berfungsi sebagai pimpinan dari pemasaran produk-produk tersebut.

c. Teknologi Komunikasi dan Navigasi, bertugas untuk memproduksi produk yang bersifat perangkat keras untuk menunjang teknologi komunikasi berdasarkan keinginan pasar, dan bagian Teknologi Komunikasi dan Navigasi berfungsi memimpin proses produksi produk-produk teknologi komunikasi.

d. Teknologi Informasi, bertugas untuk memproduksi produk yang bersifat perangkat lunak seperti membangun sebuah *database*, aplikasi perkantoran

dan lain sebagainya yang menyangkut hubungannya dengan teknologi informasi, dan bagian Teknologi Informasi berfungsi untuk memimpin proses produksi produk-produk teknologi informasi.

e. Teknologi Simulasi, bertugas untuk memproduksi *realtime simulator*, seperti *simulator* pesawat terbang, *simulator* terjun payung dan produk simulator lainnya, dan bagian Teknologi Simulasi berfungsi untuk memimpin proses produksi *realtime simulator*.

2.2 Landasan Teori

Selama pelaksanaan kerja praktek di PT. Dirgantara Indonesia, peserta kerja praktek menggunakan pengetahuan yang diperoleh selama masa perkuliahan sebagai landasan teori pengembangan aplikasi Office Automation. Pengetahuan dan teori yang digunakan antara lain:

2.2.1 Metode Pengembangan Perangkat Lunak Berbasis Objek

Teori tentang metode pengembangan perangkat lunak berbasis objek diperoleh di mata kuliah Rekayasa Perangkat Lunak.

2.2.2 Konsep RDBMS (*Relational Database Management System*)

Teori dan konsep RDMS yang diperoleh di mata kuliah Basis Data dan Sistem Basis Data.

2.2.3 Konsep Sistem Informasi

Teori tentang pembangunan sistem informasi yang baik diperoleh di mata kuliah Sistem Informasi.

2.2.4 Konsep PHP

PHP: ***Hypertext Preprocessor*** adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memrogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS.

Sejarah PHP

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 perilisan kode sumber ini menjadi sumber terbuka, maka banyakpemrogram 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 dan singkatan PHP dirubah menjadi akronim berulang *PHP: Hypertext Preprocessing*.

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.

2.2.5 Konsep Zend Framework (ZF)

Zend Framework ini disusun pada awal tahun 2005 sementara kerangka kerja baru, seperti Ruby on Rails dan Spring Framework, telah mendapatkan popularitas dalam pengembangan komunitas web. ZF itu diumumkan pada Konferensi Zend pertama di bulan Oktober 2005.

Zend Framework adalah salah satu framework bahasa pemrograman PHP yang berbasis OOP (*Object Oriented Programming*), sederhana, dan *open source*.

Zend Framework difokuskan untuk membangun aplikasi Web 2.0 dan *Web Service* yang lebih aman, reliabel dan modern. *Framework* ini juga telah mendukung API untuk beberapa vendor seperti e Google, Amazon, Yahoo!, Flickr.

Beberapa fitur terbaru Zend Framework antara lain :

- Mendukung AJAX melalui JSON
- Pencarian – Lucene
- Sindikasi
- Layanan web
- Kelas library berkualitas tinggi berorientasi objek untuk PHP 5

ZF adalah *use-at-will* framework. Tidak ada paradigma *single development* atau pola yang harus diikuti oleh pengguna Zend Framework, meskipun ZF menyediakan component untuk MVC, *Table Data Gateway*, dan pola desain *Row Data Gateway*. Zend framework menyediakan komponen individual untuk banyak penggunaan yang sering dibutuhkan dalam membangun aplikasi web.

Zend Framework membutuhkan PHP 5.2.4 keatas atau sejak versi 1.7.0. Versi sebelumnya membutuhkan PHP 5.1.4 keatas, meskipun *ZF Programmer's Reference Guide* merekomendasikan PHP 5.2.3 keatas untuk peningkatan keamanan dan performa termasuk versi PHP saat ini. PHPUnit 3.0 keatas membutuhkan sebuah unit percobaan yang disisipkan di Zend Framework. Banyak komponen juga membutuhkan ekstensi PHP.

Fitur

Fitur Zend Framework antara lain:

- Semua komponen sepenuhnya PHP 5 berorientasi objek dan mengikuti protocol E_STRICT.
- Arsitektur *Use-at-will* dengan penggunaan komponen yang mudah dan minimal interpendesi.
- Implementasi MVC ekstensible mendukung layout dan template berbasis PHP secara default.
- Mendukung banyak sistem database dan vendor, termasuk MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, dan Informix Dynamic Server.
- Susunan email dan pengiriman, penerimaan melalui mbox, Maildir, POP3 dan IMAP4
- Penyimpanan sub system yang fleksibel dengan dukungan banyak *backend*, seperti memori atau *system file*.

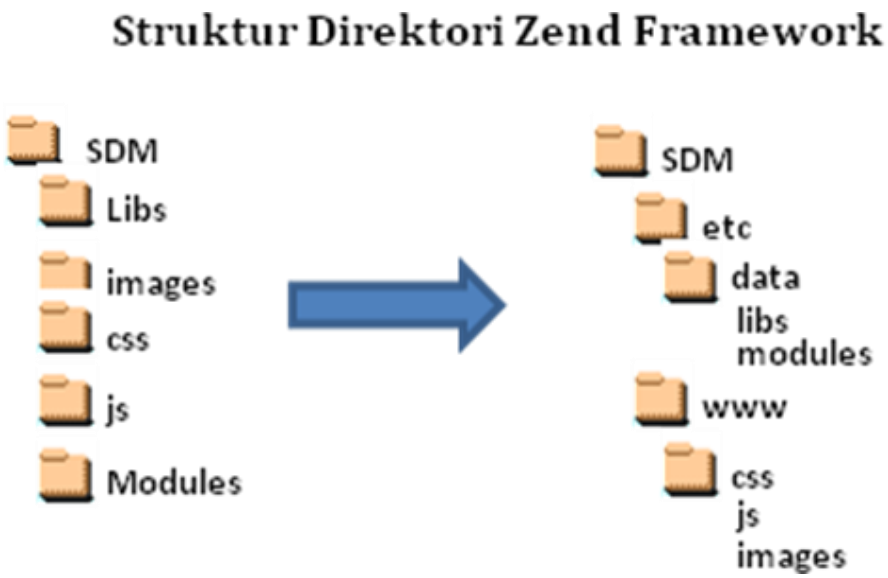
Membangun dan menjalankan aplikasi Zend Framework

Teknologi Zend menyediakan PHP stack, Zend Server (atau Zend Server Community Edition), yang telah dioptimalisasi untuk menjalankan aplikasi Zend Framework. Zend Server berisi Zend Framework dalam OS masing-masing, installer yang terintegrasi, bersama dengan PHP dan semua ekstensi yang diperlukan. Zend Server menyediakan performa yang telah ditingkatkan untuk

PHP dan khususnya aplikasi Zend Framework melalui percepatan *opcode* dan beberapa kemampuan *caching*. Dan menyertakan aplikasi *monitoring* dan fasilitas *diagnose*. Zend server tidak dibutuhkan untuk menjalankan aplikasi Zend Framework dan Zend Framework bisa di unduh dan dipasang secara terpisah dengan Zend Server.

Untuk pengembang yang memilih IDE untuk membuat kode dan pencarian kesalahan, Zend Studio, yang berintegrasi erat dengan Zend Framework, menyediakan jendela MVC, pembangun kode MVC, pemformat kode, pendampingan kode, dan sebagainya.

Zend Server mungkin tidak cocok dengan alat pencari kesalahan seperti Xdebug yang disertakan dengan PHP IDE gratis seperti Netbeans. Untuk alasan ini, para pengembang yang ingin menggunakan PHP stack dengan lengkap, termasuk ODE, harus mempertimbangkan alternatif lain seperti Zend Server.



Gambar II.3 Struktur direktori Zend Framework

2.2.6 Konsep SSO (*Single Sign On*)

Single Sign-on (SSO) adalah properti kontrol akses dari berbagai macam sistem perangkat lunak dan terhubung. Dengan property dari satu login user akan memberikan akses ke semua sistem tanpa diingatkan untuk login lagi, *Single Sign-off* adalah kebalikannya, dimana dengan satu kali sign out akan menghentikan akses ke banyak sistem perangkat lunak.

Sebagaimana aplikasi dan sumber daya yang berbeda mendukung mekanisme autentikasi yang berbeda, *single sign-on* harus diterjemahkan secara internal untuk dan dari sistem pengenal yang digunakan untuk autentifikasinya.

Keuntungan

Keuntungan meliputi:

- Mengurangi *phising*, karena user tidak terlatih untuk memasukan password tanpa berpikir.
- Mengurangi kelelahan dari banyak kombinasi *username* dan *password* user.
- Mengurangi waktu yang dihabisakan untuk memasukkan *password* lagi untuk *username* yang sama.
- Dapat mendukung autentikasi konvensional seperti Windows Credentials
- Mengurangi biaya IT atas rendahnya panggilan IT help desk tentang password.
- Keamanan di semua level dari masuk / keluar / akses ke sistem tanpa perlu mengingatkan user.
- Laporan terpusat

SSO menggunakan server autentikasi terpusat dimana semua aplikasi dan sistem dimanfaatkan untuk tujuan autentikasi, dan dikombinasikan dengan teknik ini untuk menjamin user tidak memasukan pengenal mereka lagi lebih dari satu kali.

Kritik

Kata *enterprise reduced sign-on* digunakan oleh beberapa penulis yang yakin bahwa *single sign-on* tidak mungkin diaplikasikan dalam kehidupan nyata.

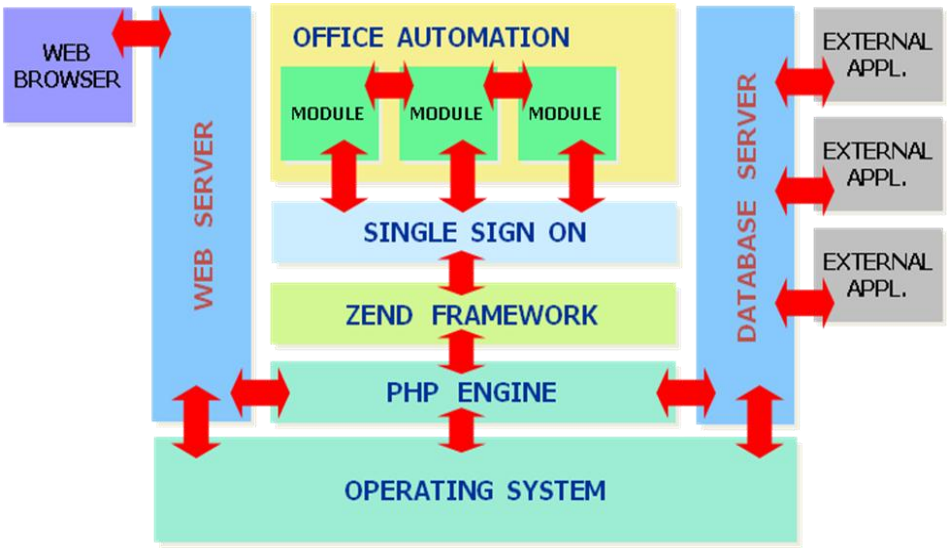
Sebagaimana *single sign-on* menyediakan akses ke banyak sumber begitu user telah di autentikasi, hal tersebut menimbulkan akibat negatif misalnya akun dipakai oleh orang lain dan disalahgunakan. Oleh karena itu, *single sign-on* membutuhkan perhatian lebih pada perlindungan akun user, dan harus

dikombinasikan dengan metode autentikasi seperti *smart cards* dan token *one-time password*.

Single sign-on juga membuat sistem autentikasi sangat rawan, kegagalan atau ketidakmampuan untuk masuk bisa menghasilkan penolakan akses ke semua sistem yang disatukan oleh SSO. Hal ini bisa membuat SSO tidak dinginkan untuk sistem yang aksesnya telah terjamin sepanjang waktu, seperti keamanan atau *plant-floor systems*

Skema *shared authentication* yang bukan *single sign-on*

Single sign on membutuhkan user untuk *sign in* sekali untuk mempertahankan aksesnya. Sistem akan membutuhkan *user* untuk login berkali-kali dengan identitas yang sama bukanlah *single sign on*. Contohnya, sebuah lingkungan dimana user dipersilahkan untuk *log in* ke desktop mereka, lalu *log in* ke email mereka menggunakan pengenal yang sama, itu bukan *single sign on*.



Gambar II.4 Penerapan SSO pada *Office Automation* Kementerian Negara BUMN

2.2.7 Konsep JQuery

JQuery adalah sebuah *library javascript* lintas browser yang dibuat untuk menyederhanakan skrip HTML di sisi client. Dirilis di januari 2006 di BarCamp NYC oleh John Resig. Digunakan oleh 31% dari 10.000 situs web yang paling sering dikunjungi. jQuery adalah *library javascript* yang paling popular digunakan saat ini.

jQuery gratis, perangkat lunak *open source*, lisensi dibawah MIT License dan GNU General Public License, Version 2. Peinrtah jQuery dibuat agar mudah untuk menjelajahi dokumen, memilih elemen DOM, membuat animasi, menangani *event*, dan mengembangkan aplikasi ajax. jQuery juga menyediakan kapabilitas untuk para pengembang untuk membuat *plug-in* untuk Javasript Library. Dengan fasilitas ini, pengembang mampu membuat abstraksi untuk

interaksi dan animasi tingkat rendah, efek dan tema tingkat lanjut. Kontribusi ini semakin memajukan halaman web.

Microsoft dan nakia telah mengungumkan rencana untuk menyertakan jQuery di platform mereka. Microsoft mengadopsinya ke dalam Visual Studio untuk digunakan di ASP.NET AJAX framework kepunyaan Microsoft dan ASP.NET MVC frameworl sementara Nokia telah menggabungkannya dengan *platofrm* Run-Time Web mereka. JQuery telah digunakan di MEdiaWiki sejak versi 1.16.

2.2.8 Konsep AJAX (*Asynchronous JavaScript and XMLHTTP*)

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

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.

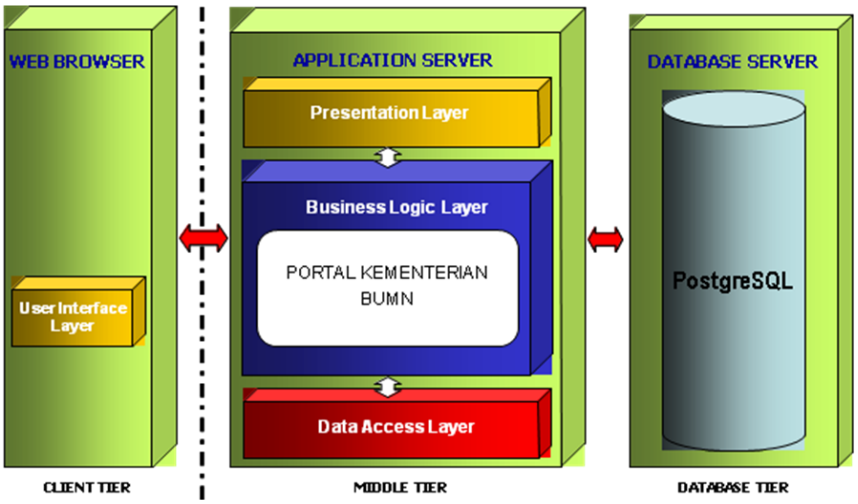
XML umumnya digunakan sebagai dokumen transfer, walaupun format lain juga memungkinkan, seperti HTML, plain text XML dianjurkan dalam pemakaian teknik AJaX karena kemudahan akses penanganannya dengan memakai DOM.

JSON dapat menjadi pilihan alternatif sebagai transfer dokumen, mengingat JSON adalah JavaScript itu sendiri sehingga penanganannya lebih mudah

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.

2.2.9 Konsep *Application Framework*

Arsitektur aplikasi Portal yang telah dibangun berbasis Web dengan penerapan teknologi *Three Tier*.



Gambar II.5 Penerapan *Three-Tier* pada Portal Kementerian Negara BUMN

Pada arsitektur *tree-tier* ada tiga bagian penting yang paling mendasar, seperti terlihat pada Gambar diatas dan gambar dibawah ini, terdapat lapisan-lapisan yang menunjukkan bahwa arsitektur tersebut *tree-tier*, yaitu :

1 *Client tier (presentation layer, user services).*

2 *Business tier (business logic layer, business services).*

3 *Data tier (data layer, data services).*

Arsitektur *tree-tier* memiliki komponen terdepan yang bertanggung jawab dalam penyediaan antarmuka ke pengguna, dan komponen paling belakang adalah server basisdata itu sendiri. Komponen *middle tier* memperbolehkan pengguna untuk berbagi dan mengontrol *business logic layer* dengan mengisolasi komponen-komponen yang ada didalam *middle tier* tersebut dari aplikasi yang sesungguhnya dihadapi pengguna. Dengan *middle tier* tersebut maka *business logic* dan *rules* dieksekusi dan dapat mencakup hingga banyak user, dengan menyediakan fungsi-

fungsi untuk antrian, eksekusi aplikasi, maupun akses basisdata. Sistem pada client berinteraksi dengan *middle tier* melalui sebuah protokol yang sudah menjadi standar seperti misalnya HTTP, RPC, ataupun *middleware* seperti COM+, ASP.NET, CORBA, DCOM, atau JavaBean. Dan *middle tier* tersebut berinteraksi dengan *database server* melalui protokol database standar seperti ADO, SQL, ODBC, dan JDBC. *Middle tier* merupakan lapisan yang paling penting dan menentukan adanya komunikasi antara client dengan server. Pada *middle tier* ini segala fungsi-fungsi yang dibutuhkan untuk komunikasi antara client dan server muncul. Adanya istilah *fat server* maupun *fat client* merupakan pengertian bahwa fungsi-fungsi tersebut cenderung lebih banyak menempel pada client atau pada server.

Keuntungan dari arsitektur *tree-tier* yaitu :

1. Perubahan pada antarmuka pengguna atau *application logic* tidak saling mempengaruhi satu sama lain, membuat aplikasi tersebut mudah berevolusi untuk memenuhi kebutuhan baru.
2. *Network bottleneck* diminimalkan karena *application layer* tidak mentransmisikan data ekstra ke client, hanya data yang diperlukan untuk menangani suatu *task*.
3. Ketika *business logic* perlu diubah, hanya server yang perlu diubah.
4. Client terisolasi dari basisdata dan operasi jaringan. Client tersebut dapat mengakses data dengan mudah dan cepat tanpa harus mengetahui lokasi penyimpanan data atau jumlah server yang ada dalam sistem.

5. Organisasi memiliki basisdata yang independen, karena data layer ditulis menggunakan SQL standar yang memang bebas platform. Perusahaan tersebut tidak terikat pada *vendor-specific stored procedures*.
6. *Application layer* ditulis dalam bahasa generasi ketiga atau keempat standar, yang telah biasa digunakan para pemrogram.