

# ANALISIS DAN PERANCANGAN *APPLICANT TRACKING SYSTEM*

Soni<sup>1</sup>; Hapsoro Renaldy N.<sup>2</sup>; Yogi Sujiwo<sup>3</sup>

<sup>1, 2, 3</sup>Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bina Nusantara,  
Jl. K.H. Syahdan No. 9, Kemanggisian/Palmerah, Jakarta Barat 11480  
<sup>1</sup>sony@binus.ac.id

## ABSTRACT

*The aim of the research was to design a web-based applicant tracking system to support Human Resources Management system. Research methods were library study and system design. It can be concluded that the applicant tracking system is able to help HRD staff to chose staff candidates, manage recruitment time, create an applicant database, and it can be function as a web bursa kerja site (EE).*

**Keywords:** *applicant tracking system, candidate selection, employee recruitment*

## ABSTRAK

*Tujuan penelitian adalah merancang applicant tracking system berbasis web guna mendukung sistem Human Resources Management. Metode penelitian menggunakan studi perpustakaan dan perancangan sistem. Disimpulkan bahwa Applicant Tracking System memiliki manfaat staf HRD, memiliki calon pegawai, mengatur jadwal perkantoran, menjadi basis data pelamar, dan dapat berfungsi sebagai situs web bursa kerja (EE).*

**Kata kunci:** *applicant tracking system, penyaringan kandidat, perekrutan pegawai*

---

## PENDAHULUAN

Era perdagangan bebas adalah sebuah era ketika batasan regional dan politik diantara negara yang menyangkut urusan perdagangan dihapuskan atau pun diminimalisasikan. Hal itu menyebabkan investor asing dapat masuk ke Indonesia serta menanamkan modalnya disini. Dengan masuknya para investor asing tersebut maka perusahaan lokal akan berusaha untuk lebih memajukan dirinya sehingga dapat ikut berkompetisi dengan perusahaan asing. Dampak dari persaingan tersebut serta masuknya para investor asing menyebabkan kebutuhan akan tenaga kerja yang berkualitas meningkat. Banyak calon tenaga kerja yang akan mengambil peluang ini untuk mencari pekerjaan dengan bidang yang mereka kuasai dan inginkan. Dengan banyaknya calon pelamar yang ingin melamar pekerjaan di perusahaan maka akan banyak pula data calon pekerja yang harus ditangani oleh bagian HRD (*Human Resource Department*) untuk

kemudian diolah menjadi data yang berguna bagi perusahaan. Data para pelamar yang telah diolah tersebut dapat dimanfaatkan sebagai bahan pertimbangan untuk memilih pelamar mana saja yang akan dipanggil untuk wawancara.

Untuk melakukan penyaringan data tersebut, dibutuhkan sebuah sistem penyaringan dan pelacakan. Sistem penyaring akan memilah data penting sedangkan sistem pelacakan akan selalu mencari data baru dan meng-*update* data yang lama. Pada sistem lama, data para pelamar disimpan dalam bentuk berkas kertas yang seiring dengan berjalannya waktu dapat menumpuk dan memenuhi gudang arsip. Dengan sistem lama tersebut, ada kemungkinan juga berkas para pelamar yang akhirnya tidak diterima bekerja di perusahaan tersebut, dibuang ke tempat sampah ataupun hilang. Sebenarnya dapat saja dari sekian pelamar yang tidak diterima tersebut, berpotensi untuk mengisi kekosongan lowongan pekerjaan di perusahaan tersebut untuk beberapa waktu kemudian.

Dengan adanya kecanggihan teknologi informasi yang ada sekarang ini, sistem lama dapat diperbaharui dan lebih dipermudah serta lebih ringkas dalam penyimpanan data. *Applicant Tracking System* (ATS) atau yang disebut juga *candidate management system* adalah sebuah *software* yang didesain untuk membantu perusahaan dalam melakukan perekrutan pegawai secara efisien. ATS dapat digunakan untuk mengirim pengumuman lowongan pekerjaan ke *website* milik perusahaan maupun tempat pengumuman lowongan pekerjaan lainnya di dunia maya. Sistem ini memungkinkan melakukan pemeliharaan terhadap *database* para pelamar dengan lebih baik dan lebih hemat tempat penyimpanan daripada sistem konvensional yang lama. ATS juga memungkinkan pencarian pelamar yang cocok untuk menempati lowongan yang tersedia sesuai dengan kriteria yang dipilih. ATS juga memiliki keuntungan bagi pelamar, mereka hanya cukup memasukkan data mereka sekali saja dan kemudian sistem akan mengurus hal lainnya. Pelamar dapat mencari lowongan yang tersedia dan melamar, balasan dari perusahaan akan dikirimkan langsung ke-*email* sang pelamar.

Tujuan pengembangan *software* adalah merancang dan membuat sebuah *Applicant Tracking System* berbasis *web* untuk digunakan sebagai sistem pendukung dalam sistem *Human Resource Management* sebuah perusahaan. Untuk penelitian ini, PT Maestro Nusantara Prima bersedia menjadi tempat pengaplikasian sistem ini. Adapun manfaat yang akan didapat dari penerapan sistem ini adalah memudahkan penyimpanan data dari para pelamar serta penyaringan pelamar; Data dari pelamar yang belum diterima dapat dijadikan sebagai bahan pertimbangan untuk posisi lowong berikutnya; Sistem yang berbasis *web* dapat diintegrasikan ke dalam *website* perusahaan sehingga dapat meminimalisasikan biaya pemasangan lowongan kerja; Staf HRD dapat menggunakan sistem tanpa harus terlalu bergantung kepada staf IT; Eksekutif HRD dapat menerima laporan terperinci mengenai data yang ada secara lebih *user friendly*; Perusahaan mendapatkan staf yang andal.

## TINJAUAN PUSTAKA

Rekayasa perangkat lunak didefinisikan oleh Fritz Bauer sebagai penetapan dan penggunaan prinsip rekayasa dalam rangka mendapatkan perangkat lunak yang ekonomis, terpercaya, serta bekerja efisien pada komputer. Siklus rekayasa

perangkat lunak terdiri atas: Rekayasa Sistem, Analisis, Perancangan, Pengkodean, Pengetesan, dan Pemeliharaan.

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 besar yang berhubungan dengannya. Untuk merancang sebuah antarmuka sistem, perlu diperhatikan beberapa aspek penting. Aspek penting tersebut terangkum dalam delapan aturan emas, yakni: Berusaha untuk konsisten; Memungkinkan pengguna menggunakan *shortcut*; Memberikan umpan balik yang informatif; Merancang dialog untuk menghasilkan keadaan akhir yang baik; Memberikan pencegahan kesalahan dan penanganan kesalahan yang sederhana; Memungkinkan pembalikan aksi yang mudah; Pengguna dapat menguasai sistem; Mengurangi beban ingatan jangka pendek.

*Unified Modelling Language* (UML) adalah sebuah bahasa untuk menspesifikasikan dan memodelkan sebuah objek yang digunakan dalam rekayasa perangkat lunak. UML juga menyertakan notasi grafis standar yang dapat digunakan untuk menunjukkan abstraksi model dari sebuah sistem.

*Applicant Tracking System* adalah sebuah aplikasi yang berfungsi untuk menangani proses *recruitment* perusahaan. Sistem ini terkadang juga dimasukkan ke dalam *situs web* milik perusahaan sehingga memungkinkan perusahaan untuk memasukkan iklan lowongan pekerjaan pada *situs web* mereka dan menarik minat para pelamar pekerjaan.

*Human Resource Management* (HRM) (Dessler, 1999:2) adalah sebuah teori akademis namun juga merupakan sebuah praktek bisnis yang berlandaskan pada fakta bahwa pekerja adalah manusia dan bukanlah sumber daya perusahaan yang berupa barang. HRM dapat dilihat sebagai sebuah pemahaman perusahaan atas aspek manusia dan sumber daya bisnis yang terkandung di dalamnya. HRM harus dianggap sebagai sebuah aspek mayoritas pada filosofi perusahaan dan semua manajer dan staf biasa adalah pilar penyokong perusahaan yang perlu diatur berdasarkan kebijakan tertentu sehingga dapat menghasilkan sesuatu secara maksimal namun dengan tetap memperhatikan aspek kemanusiaan.

*Model View Controller* (MVC, Anonymous, <http://en.wikipedia.org/wiki/Model-view-controller>) adalah sebuah arsitektur *software* yang memisahkan model data, antarmuka pemakai, serta kontrol logis dari sebuah aplikasi menjadi tiga

komponen berbeda. Hal itu bertujuan untuk meminimalkan kerusakan yang terjadi akibat modifikasi yang dilakukan kepada salah satu komponen. Sering kali, MVC dianggap sebagai *software design pattern*, hanya saja MVC menitikberatkan lebih kepada arsitektur *software* daripada *design pattern* lainnya. Oleh karena itu, lebih tepat disebut sebagai *architectural pattern* (Buschmann, 1996).

Rancangan *database* dalam bentuk *class model* dapat ditranslasikan dalam bentuk *relational model*. Untuk melakukan pemetaan ini, dapat digunakan pola berikut. Pertama, kelas yang memiliki struktur data sederhana dapat langsung menjadi tabel. Kedua, penanda objek (OID) setiap objek dapat dipastikan unik, karenanya penanda objek tersebut dapat digunakan sebagai *primary key* pada tabel relasional. Ketiga, untuk sebuah kelas yang memiliki *instance* kelas lain dalam dirinya sebagai sebuah *attribute* sehingga perlu dibuat sebuah tabel lain sebagai representasi kelas yang di-*embed*. Keempat, untuk hubungan *one-to-many*, dibuat sebuah tabel terpisah yang memiliki dua kolom. Kolom pertama digunakan untuk menampung penanda objek yang terhubung ke kelas lain sedangkan kolom kedua berisi penanda objek lain yang terhubung ke objek tersebut.

Kelima, hubungan *many-to-many* diimplementasikan dengan membentuk tabel terpisah dan tabel tersebut hanya terdiri dari dua kolom. Setiap baris merupakan pasangan objek yang saling berpasangan. Keenam, hubungan *one-to-one* diimplementasikan menggunakan *foreign key* sebagai atribut yang terdapat di masing-masing tabel. Ketujuh, untuk mengimplementasikan sebuah asosiasi, setiap *subclass* diimplementasikan sebagai tabel, setiap *attribut* dari *superclass* disimpan pada tabel *subclass*. Hal itu hanya dapat dilakukan jika *superclass* bersifat abstrak dan tidak akan dijadikan sebagai sebuah *instance*. Jika sebuah *superclass* juga ingin dijadikan sebuah *instance* maka setiap kelas yang ada (*superclass* dan *subclass*) harus dibuatkan tabel masing-masing.

## PEMBAHASAN

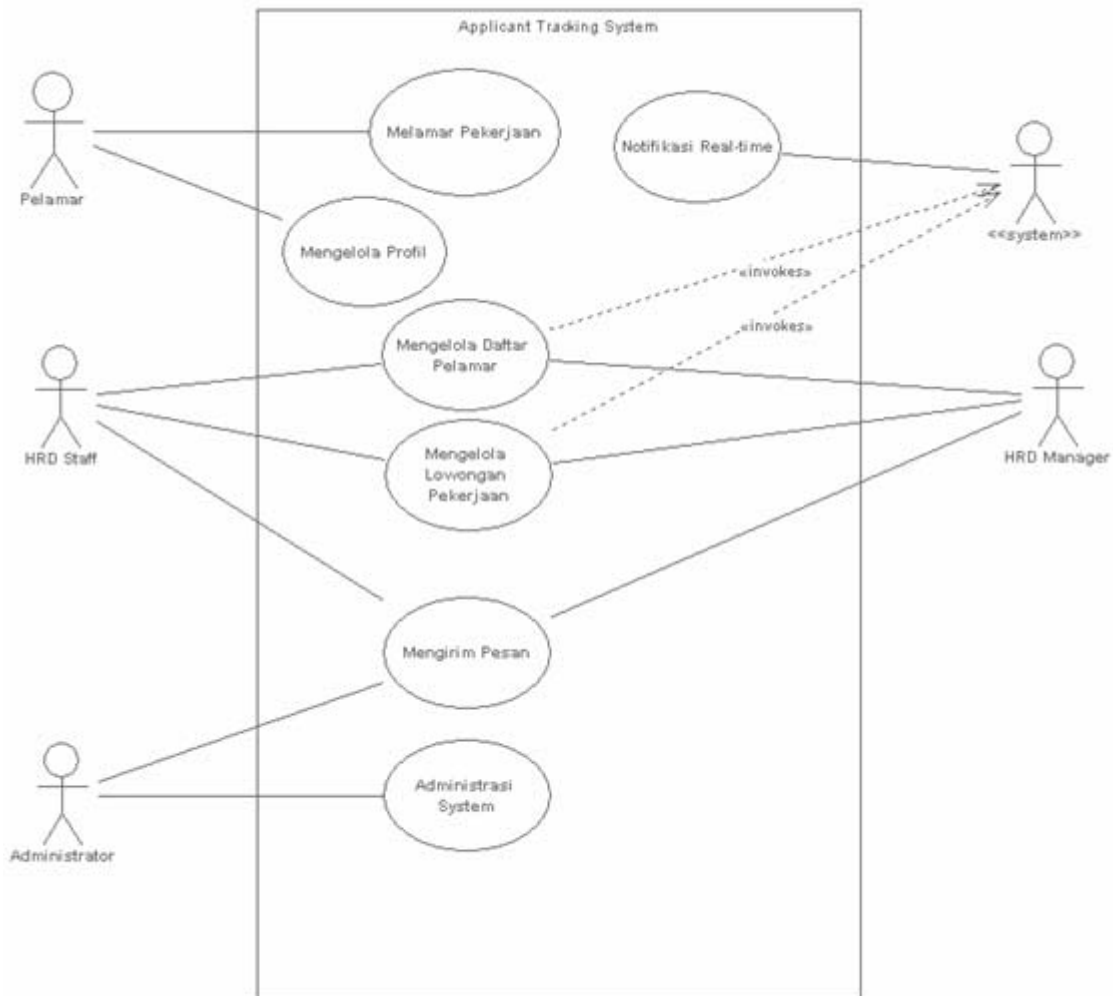
### Analisis dan Perancangan Sistem yang Diusulkan

Masalah yang ada adalah format data yang tidak sama atau standar; Tidak adanya pendataan yang terstruktur rapi atas data para pelamar;

Komunikasi antara staf HRD dan Manajer dalam melakukan penyeleksian pegawai maupun mengelola sebuah lowongan pekerjaan, harus dilakukan secara tatap muka dan terkadang ada beberapa hal yang dapat dilakukan tanpa perlu tatap muka; Sering kali terjadi masalah dalam jadwal wawancara pelamar, misalnya saja adanya jadwal yang bertabrakan; Setiap pelamar harus mengerjakan beberapa tes dasar yang terkadang dilakukan pada saat hari wawancara. Hal itu menghabiskan waktu yang cukup banyak, apalagi hal itu adalah sebuah kegiatan yang terus berulang; Terkadang data pelamar hilang dikarenakan kurangnya pengawasan; Pembuatan laporan kegiatan perekrutan susah dilakukan karena tidak adanya keseragaman format data; Pemanggilan wawancara kerja yang dilakukan *via email* terkadang mengalami kendala karena staf harus selalu membuka *inbox email* miliknya untuk memeriksa apakah pelamar sudah melakukan konfirmasi.

Dari sejumlah permasalahan yang ada dan dari hasil studi terhadap ATS lain yang telah ada, dapat disimpulkan usulan pemecahan masalah sebagai berikut. Pertama, data pelamar dimasukkan secara *online* menggunakan aplikasi berbasis web karena pelamar sendiri yang memasukkan data maka beban pekerjaan bagi staf HRD dapat sedikit berkurang. Hal itu juga mempermudah pelamar, jika ingin memperbaharui data pribadinya. Kedua, *database* pelamar yang terpusat memungkinkan dilakukannya pencarian kandidat berdasarkan kategori tertentu secara langsung. Ketiga, membuat sebuah kolaborasi *online* antarstaf maupun dengan manajer. Keempat, hal itu memungkinkan adanya komunikasi secara langsung dan juga meminimalkan waktu yang terbuang karena harus menunggu adanya laporan tertulis. Jadwal wawancara yang bertabrakan juga dapat dihindari karena setiap staf dapat saling berbagi informasi.

Kelima, pembuatan laporan secara otomatis yang diambil dari *database* ATS. Keenam, notifikasi secara langsung kepada pelamar melalui *email* dan adanya konfirmasi dari pelamar kepada sistem secara langsung. Ketujuh, dengan adanya hal semacam ini, staf tidak harus selalu memeriksa *inbox* untuk mengetahui adanya konfirmasi dari pelamar, sistem yang akan memberitahukan daftar para pelamar yang telah konfirmasi untuk mengikuti wawancara kerja kepada staf dan Manajer (Lihat gambar 1).

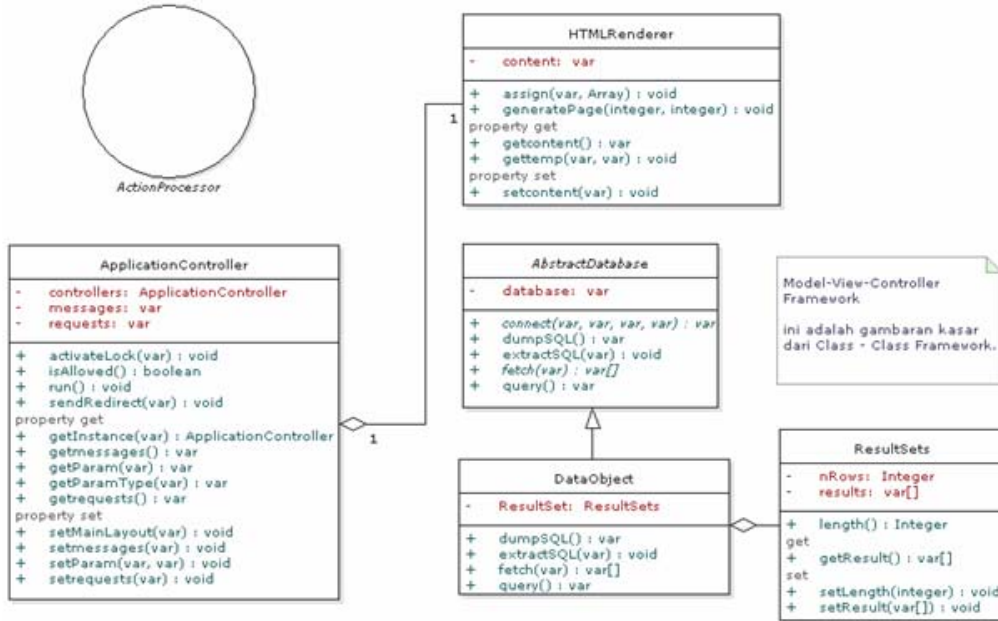


Gambar 1 Use Case Applicant Tracking System

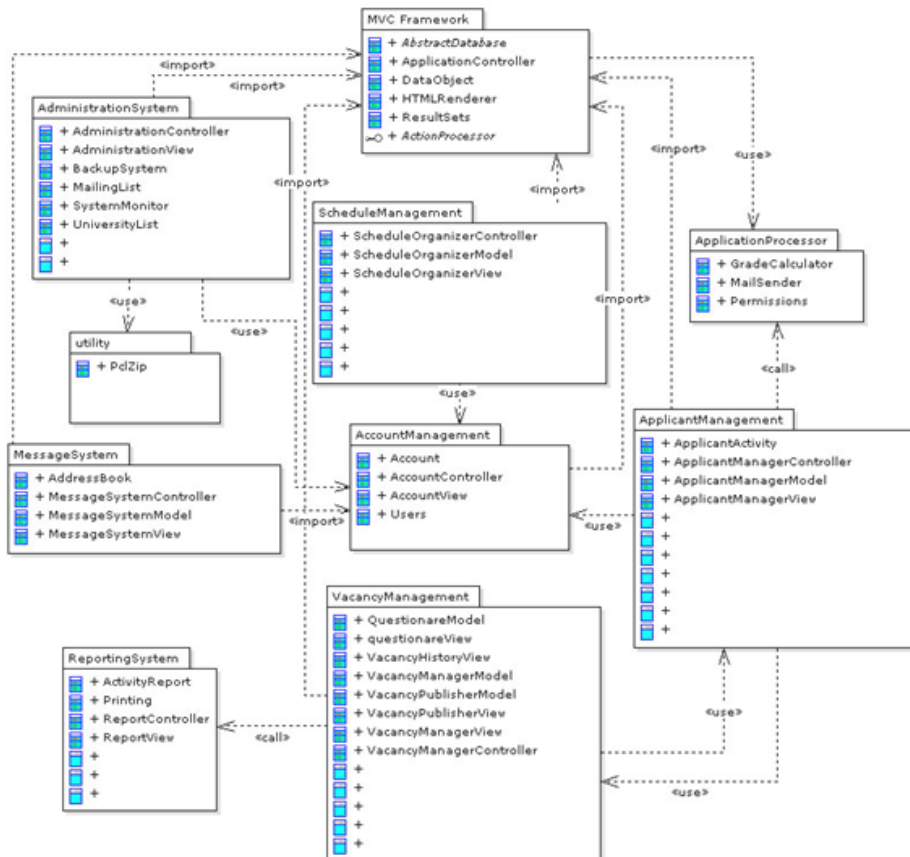
Use Case	Description
Melamar Pekerjaan	Pelamar melakukan beberapa kegiatan seperti memasukkan data pribadi, mengerjakan tes untuk mengetahui kompetensi dirinya, serta memberikan resume dirinya
Mengelola Daftar Pelamar	Data para pelamar akan bertambah seiring waktu sehingga harus dikelola dan diatur sehingga data tersebut nantinya dapat berguna bagi keseluruhan proses perekrutan
Mengelola Lowongan Pekerjaan	Ketika sebuah lowongan pekerjaan baru masuk, ia dapat dikelola dan diatur sehingga tidak akan mengganggu proses kegiatan penerimaan pelamar pada lowongan pekerjaan yang lain
Mengirim Pesan	Jika Staf, Manager, maupun Administrator ingin berbagi pendapat, mereka dapat saling mengirim pesan mengenai suatu topik. Staf juga dapat menggunakannya untuk memberi pemberitahuan kepada kandidat.
Notifikasi Realtime	Setiap kali terjadi suatu perubahan maupun hal baru yang perlu diberitahukan kepada Manager maupun Staf, sistem akan meng-generate sebuah pemberitahuan secara real time
Administrasi Sistem	Seiring berjalannya waktu, sebuah sistem membutuhkan perawatan untuk menjaga agar sistem dapat berjalan dengan normal. Perawatan sistem dilakukan dengan melakukan backup data penting juga perubahan beberapa data yang harus selalu up-to-date

Secara lengkap *Class Diagram* dapat dibuat menjadi 2 *package* utama, yaitu *Package Framework* yang merupakan *Package* yang berisi *Class* utama dari *MVC Framework* yang digunakan dalam

development ATS ini dan *Package Components* yang terdiri dari *sub-package class* yang membangun aplikasi ATS ini.



Gambar 2 Package Framework



Gambar 3 Package Component

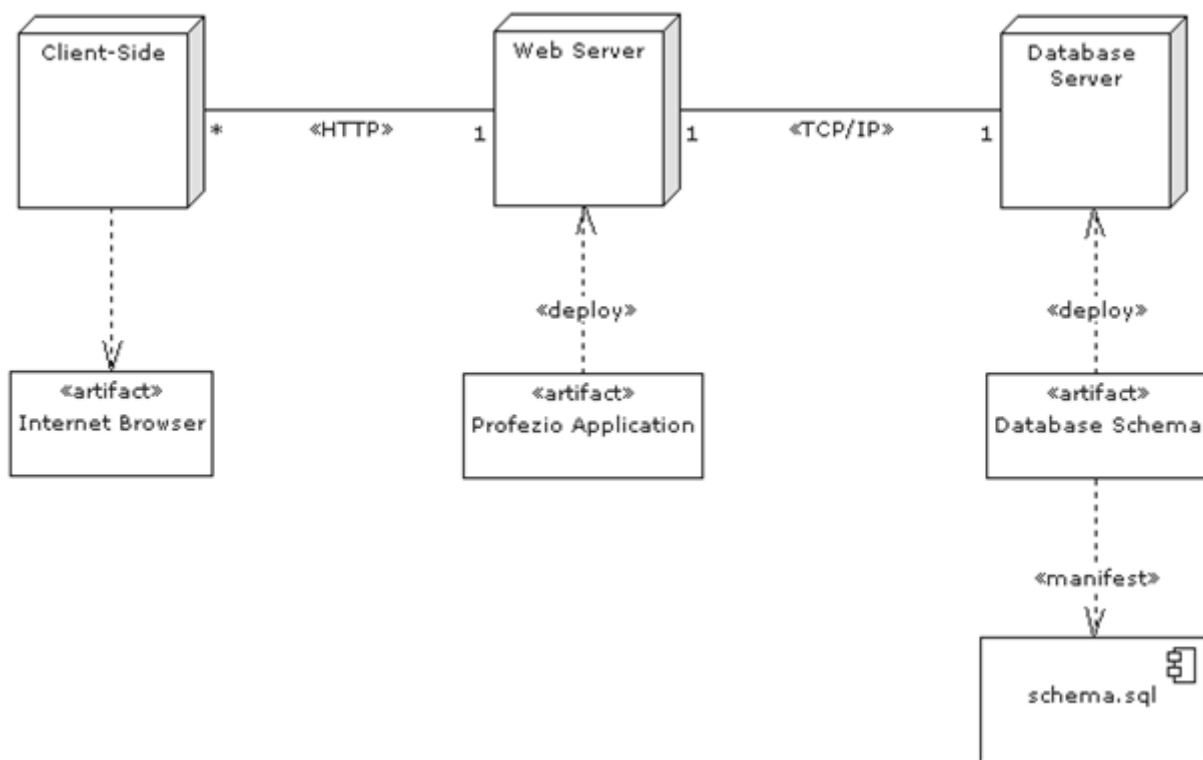
## Implementasi Sistem

Spesifikasi perangkat yang dibutuhkan untuk komputer server: Prosesor Intel Pentium Dual Xeon 3Ghz; RAM 2 Gb; Harddisk Space 300 MB; *Operating System Linux* (pada *server hosting* menggunakan CentOS Linux 3.0). *Server* menggunakan server *shared-hosting* yang di *host* oleh singcat.com. Kebutuhan tersebut dibuat sebagai antisipasi, jika suatu saat komputer *server* dikunjungi oleh ratusan pengunjung *web* secara serempak. Sistem operasi berbasis Linux dipilih karena lisensinya yang bebas dan dapat dimodifikasi sendiri sesuai dengan kebutuhan. Perangkat yang digunakan untuk komputer klien memiliki spesifikasi yang disarankan sebagai berikut: Prosesor Pentium 4 1,6 GHz; RAM 256 MB; Operating System : Microsoft Window 98, XP, Linux, MacOSX; Internet Browser : Internet Explorer 6.0 ke atas, Firefox 1.03 ke atas dengan javascript enabled.

Kebutuhan tersebut didasarkan kepada spesifikasi minimal komputer pengguna rumahan terbaru. Untuk sistem operasi yang digunakan, dapat bermacam-macam selama di sistem operasi tersebut ter-*install* aplikasi *web browser* yang mendukung *javascript* serta HTML versi terakhir (Lihat gambar 4).

Dalam merancang dan mengimplementasikan aplikasi *Applicant Tracking System*, digunakan perangkat lunak sebagai berikut: Sistem operasi Microsoft Windows XP; Zend Studio Client 5.10, digunakan untuk melakukan pemrograman PHP; Enterprise Architect 6.0, digunakan sebagai *CASE Tool*; Macromedia Dreamweaver MX, digunakan untuk merancang tampilan halaman; PHPMyAdmin, digunakan untuk mengkonfigurasi *database* sistem; Adobe Photoshop 7, digunakan untuk membuat gambar pendukung; Apache Web Server versi 2.0, digunakan sebagai *Web Server* untuk PHP; PHP versi 5.0, digunakan untuk menjalankan aplikasi *web*; MySQL 4.1 dengan dukungan InnoDB, digunakan untuk sebagai server *database*.

Gambar 5 merupakan layar yang nantinya *user* sistem dapat masuk. *Role* dirinya akan teridentifikasi dari *username* dan *password*-nya. Fungsi *sign-up* hanya dapat digunakan untuk membuat *account* pengguna baru khusus untuk tipe *applicant* saja. Untuk membuat *user* baru tipe lain, harus melalui aplikasi admin.



Gambar 4 Deployment Diagram



Gambar 5 Layar *Login*

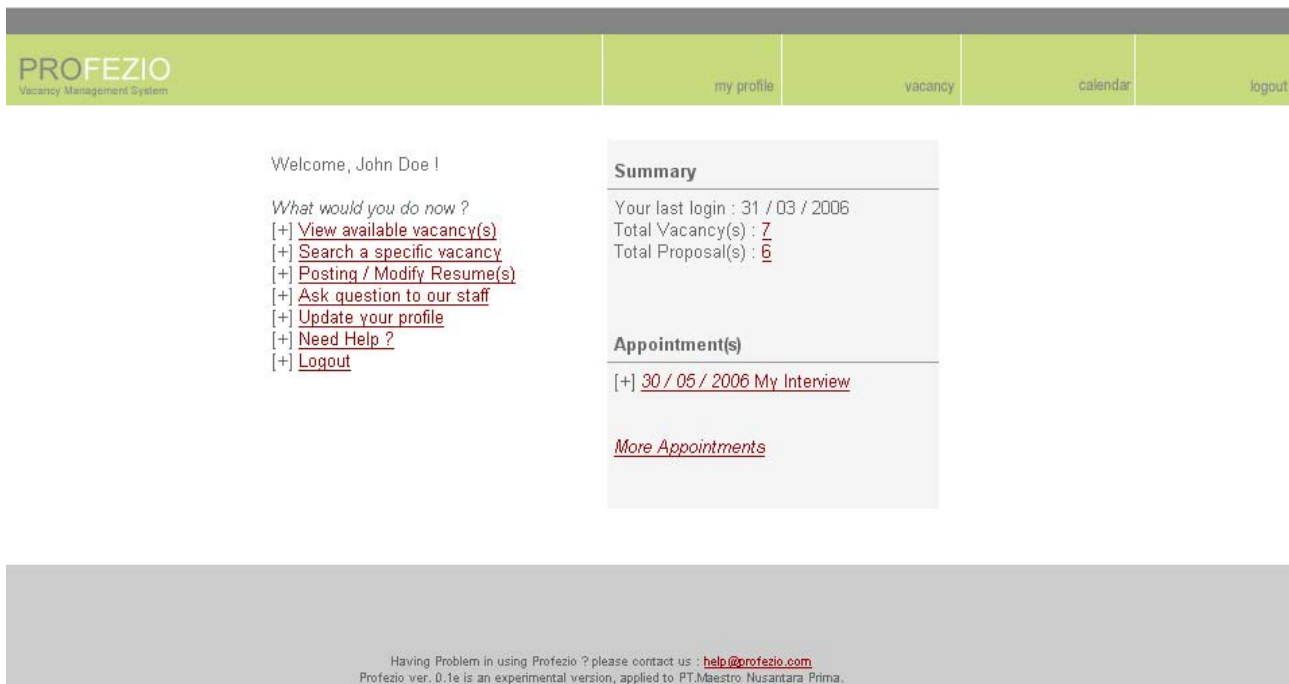
Gambar 6 merupakan layar awal milik *user* tipe *Applicant*. Disini mengakses ke semua fungsi vital dari seorang *user* tipe *Applicant*, misalnya melihat lowongan yang tersedia, melihat pesan baru yang masuk sehubungan dengan lowongan yang pernah dikirim, mengirim pertanyaan kepada staf, mengirim maupun memodifikasi *resume*, melihat jadwal wawancara dan tes.

Gambar 7 merupakan layar awal milik *user* tipe *Staff*. Disini dia dapat mengakses ke semua fungsi vital dari seorang *user* tipe *Staff*, misalnya membuat lowongan pekerjaan yang baru, melihat pesan baru yang masuk sehubungan dengan lowongan yang pernah dibuat, mencari pelamar dengan kriteria tertentu, mengirim pesan kepada manajer maupun pelamar, serta melihat jadwal kegiatan yang berhubungan dengan kegiatan pelamaran kerja.

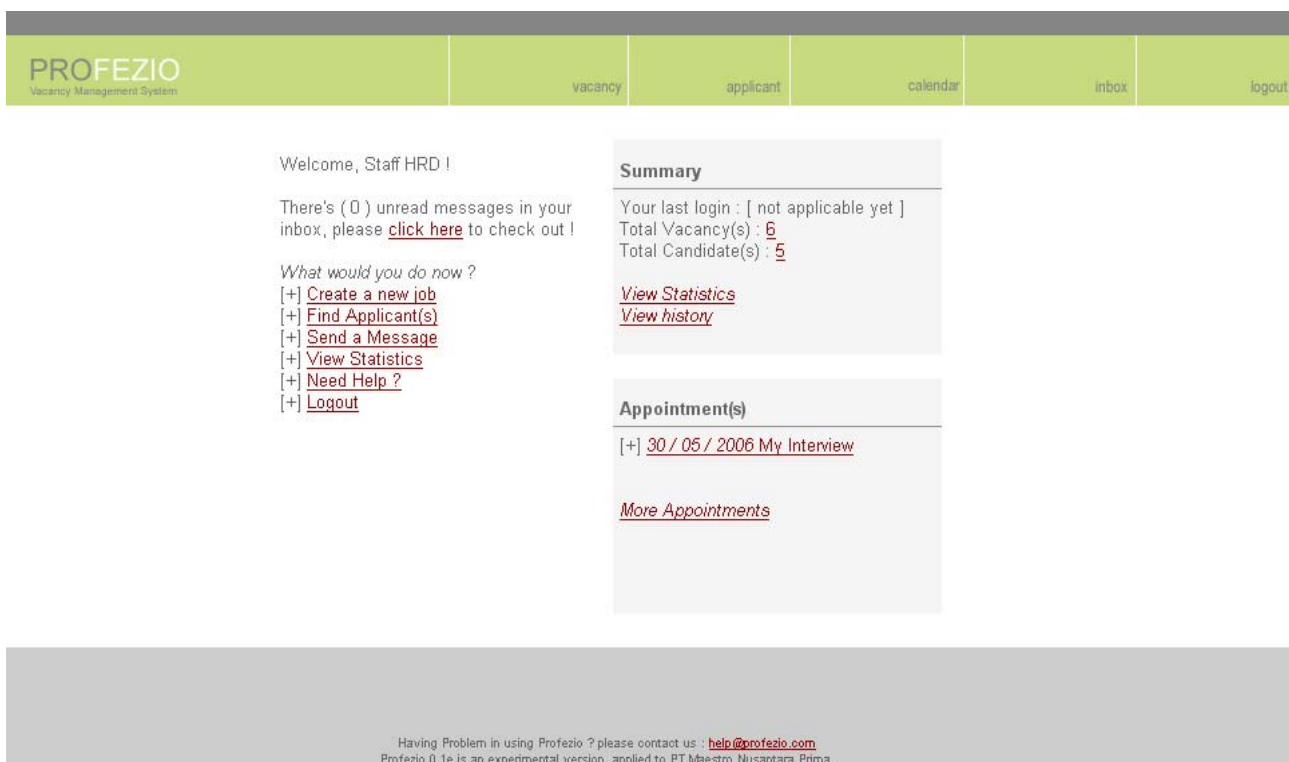
Gambar 8 merupakan layar awal milik *user* tipe *Manager*. Disini dia dapat mengakses ke semua fungsi vital dari seorang *user* tipe *Manager*, misalnya membuat lowongan pekerjaan yang baru,

melihat pesan baru yang masuk sehubungan dengan lowongan yang pernah dibuat, mencari pelamar dengan kriteria tertentu, mengirim pesan kepada Staf maupun pelamar, mendelegasikan seorang staf kepada salah satu lowongan pekerjaan yang baru saja dibuat, serta melihat jadwal kegiatan yang berhubungan dengan kegiatan pelamaran kerja.

Gambar 9 merupakan layar awal milik *user* tipe *Administrator*. Disini dia dapat mengakses ke semua fungsi vital dari seorang *user* tipe *Administrator*, misalnya melakukan manajemen *user* seperti membuat atau memodifikasi maupun menghapus *account user*, melakukan *backup* data *database*, maupun data yang di-*upload* oleh pelamar, dan memodifikasi daftar *mailing list*.

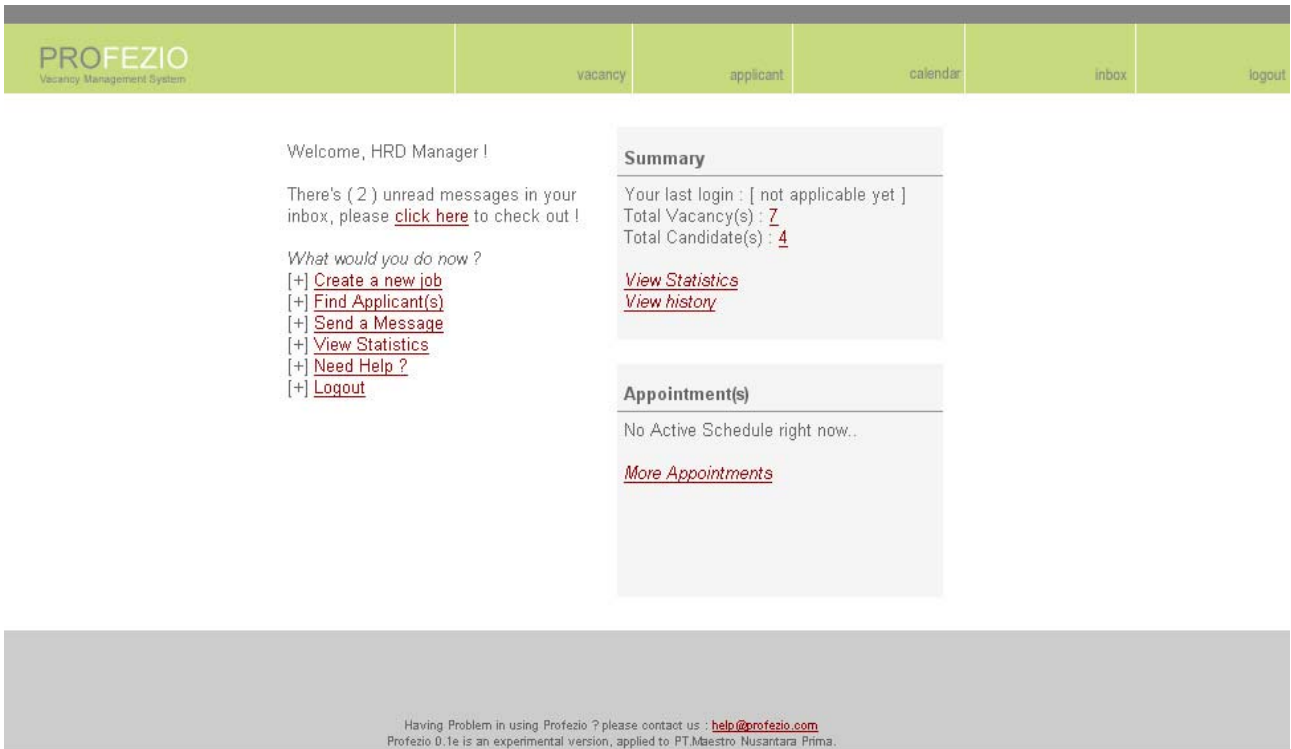


Gambar 6 Layar Awal Milik Pelamar

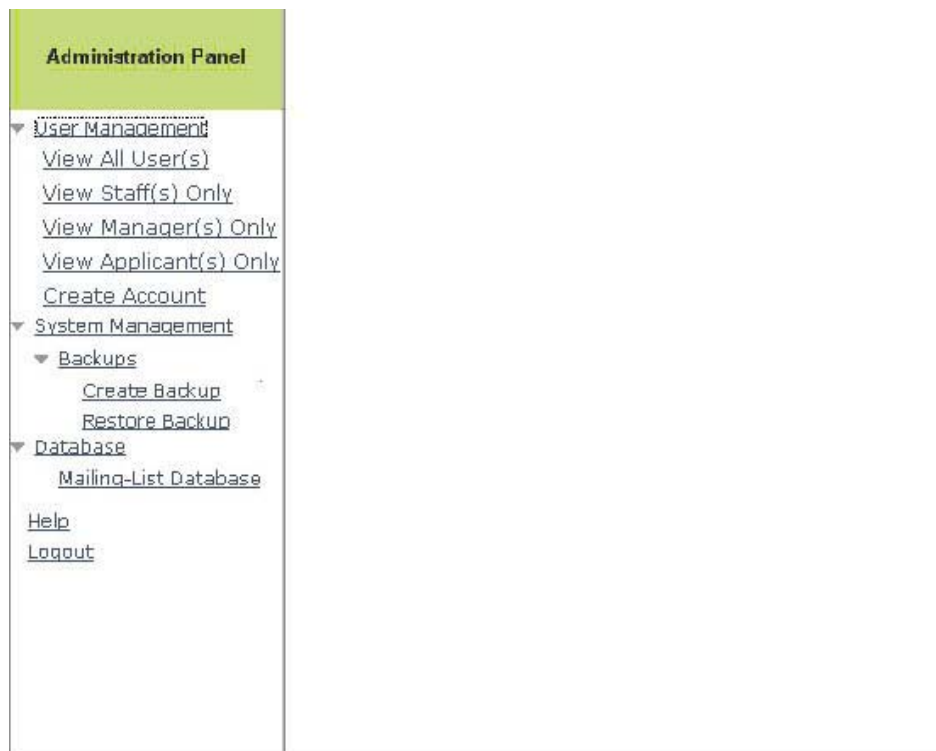


Gambar 7 Layar Awal Milik Staff





Gambar 8 Layar Awal Milik Manager



Gambar 9 Layar Awal Milik Administrator

## Evaluasi Sistem

Dari hasil kuesioner dan wawancara yang telah disebar, dapat disimpulkan: *Applicant Tracking System* ini dapat membantu kegiatan masing-masing kelompok pengguna (Pelamar, Staff HRD, Manager); Sebagian besar *tester* tidak mempermasalahkan fungsionalitas dari aplikasi tetapi lebih cenderung ke segi IMK-nya. Fungsionalitas yang diberikan sudah sesuai dengan kebutuhan pemakai secara keseluruhan. Tambahan fungsionalitas yang mungkin perlu diberikan lebih bersifat personalisasi pribadi pemakai.

## PENUTUP

Aplikasi *Applicant Tracking System* ini memiliki beberapa manfaat diantaranya: Membantu Staff HRD untuk memilih Calon Kandidat yang sesuai; Membantu Staff HRD maupun Manajer HRD untuk mengatur Jadwal Aktivitas perekrutan seperti jadwal wawancara, jadwal test, dan sebagainya; Menjadi Basis Data untuk menampung daftar pelamar serta *history* aktivitas perekrutan sebagai alternatif untuk solusi *paper-less*; Sebagai media kolaborasi antara staf dan Manajer dalam proses perekrutan; Sebagai *tools* untuk pembuatan laporan perekrutan.

Aplikasi *Applicant Tracking System* ini dapat dimanfaatkan untuk: Situs Web Bursa Kerja; Sebagai *Tools* Internal HRD sebuah perusahaan yang terintegrasi di dalam *website* perusahaan.

Saran yang dapat diberikan, yaitu aplikasi ini akan berjalan *optimal* di sistem yang mendukung PHP 5.0 atau yang terbaru dengan *database server* MySQL versi 5.0 atau yang terbaru; Untuk menjalankan aplikasi ini, pastikan fitur Javascript di *internet browser* yang digunakan dalam keadaan aktif karena beberapa fitur dalam aplikasi menggunakan javascript untuk menjalankan prosesnya; Aplikasi ini memiliki potensi untuk dikembangkan menjadi sebuah Aplikasi *Hiring Management System*, dengan melengkapi beberapa fitur tambahan, seperti fitur kalender yang lebih luas penggunaannya, statistik perekrutan yang lebih lengkap, dan beberapa fitur interaktif, seperti Wawancara Online, Kolaborasi Interaktif, sistem filterisasi yang lebih sempurna, serta Sistem *Reporting* yang lebih lengkap.

## DAFTAR PUSTAKA

- Anonymous. "Model View Controller." Diakses dari <http://en.wikipedia.org/wiki/Model-view-controller.html>
- \_\_\_\_\_. "Human Resource Management." Diakses dari [http://en.wikipedia.org/wiki/Human\\_resource\\_management.htm](http://en.wikipedia.org/wiki/Human_resource_management.htm)
- \_\_\_\_\_. "Job Analysis." Diakses dari [http://en.wikipedia.org/wiki/Job\\_analysis.htm](http://en.wikipedia.org/wiki/Job_analysis.htm)
- \_\_\_\_\_. "Personel Analysis." Diakses dari [http://en.wikipedia.org/wiki/Personel\\_analysis.htm](http://en.wikipedia.org/wiki/Personel_analysis.htm)
- Bennet, Steve McRobb dan Ray Farmer. 2002. *Object Oriented System Analysis and Design*. Singapore: McGraw-Hill.
- Dessler, Garry. 2004. *A Framework for Human Resource Management*. Singapore: McGraw-Hill.
- EagleStones, Barry dan Mick Ridley. 2001. *Web Database System*. McGraw-Hill Book Company.
- Levkow, Dave. "The World's Very Best Employment Websites." Diakses dari <http://www.ereexchange.com/articles/5921C81.asp>
- Nielsen, Jakob. 2000. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing.
- Nielsen, Jakob dan Marie Tahir. 2001. *Homepage Usability: 40 Web Sites Deconstructed*. New Riders Publishing.
- Shneiderman, Ben. 1998. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Third Edition. Addison-Wesley.
- Singh, Raghav. "Building a Better ATS." Diakses dari <http://www.ereexchange.com/articles/774.asp>
- Sparks, Geoffrey. "Database Modelling in UML." Diakses dari <http://www.methodsandtools.com/archive/archive.php?id=9p1>
- Whittne, Lonnie D. Bentley dan Kevin C. Ditman. 2004. *System Analysis and Design Method*. Singapore: McGraww Hill.