PEMBANGUNAN PERANGKAT LUNAK SISTEM OPERASI DAN OFFICE BERBASISKAN TEKNOLOGI WEB 2.0

Azuwir

azuwir@gmail.com

Pembimbing I : Mira Kania Sabariah, S.T, M.T

Pembimbing II : Tati Harihayati M, S.T, M.T

Fakultas Teknik dan Ilmu Komputer, Jurusan Teknik Informatika

Universitas Komputer Indonesia

Abstrak

Sistem operasi saat ini sangat bervariasi. Sistem operasi modern harus sudah memiliki dukungan terhadap fitur multiuser, multithreading, multitasking, multiprogramming, dan remote desktop. Remote Desktop memberikan kemudahan bagi pengguna dalam mengerjakan dokumen kerja dari komputer manapun di sebuah perusahaan, dari rumah, luar kota, dan dari luar negeri. Persoalan mendasar sistem operasi lainnya terletak pada kompatibilitas aplikasi yang dapat berjalan pada sistem operasi yang berbeda, hal ini disebabkan perbedaan arsitektur kernel dan pemaketan aplikasi. Sebagai contoh aplikasi office, selain faktor kompatibilitas sistem operasi yang didukung, juga kompatibilitas format dokumen yang dihasilkan masing-masing aplikasi office masih menjadi persoalan yang belum ada solusinya.

Teknologi web 2.0 memungkinkan aplikasi desktop berjalan di atas aplikasi web. Hal ini memungkinkan aplikasi yang berjalan pada sistem operasi berbasis web 2.0 tidak mengalami persoalan kompatibilitas aplikasi, khususnya kompatibilitas format dokumen yang dihasilkan dari aplikasi perkantoran (office).

***Kata Kunci**: sistem operasi, remote desktop, perangkat lunak, office, kompatibilitas.*

1. PENDAHULUAN

Perangkat lunak sistem operasi yang ada saat ini sangatlah bervariasi. Kehadiran Microsoft Windows Vista, MacOS Tiger X Leopard, dan beberapa distribusi besar varian GNU/Linux seperti Mandriva, OpenSuSE, dan Ubuntu ikut menghadirkan banyak pilihan bagi pengguna komputer untuk menggunakan sistem operasi yang mereka inginkan pada *PC Desktop* dan *Notebook*. Perbedaan mendasar dari setiap sistem operasi terletak pada arsitektur kernel yang digunakan, dukungan ekstensi paket

aplikasi yang dapat dijalankan, struktur direktori, dan lisensi penggunaan yang menyertainya.

Sistem operasi modern harus sudah memiliki dukungan terhadap fitur *multiuser, multithreading,multitasking,multiprogramming,* dan *remote desktop. Remote Desktop* memberikan kemudahan bagi pengguna dalam mengerjakan dokumen kerja dari komputer manapun di sebuah perusahaan, dari rumah, luar kota, dan dari luar negeri. Fitur *remote desktop* telah diimplementasikan pada banyak sistem operasi modern, hanya saja masih dibutuhkan *resource bandwidth* internet yang besar jika kita menggunakannya tidak dalam ruang lingkup area internal perusahaan. Faktor kecepatan yang buruk juga masih merupakan persoalan yang krusial bagi pengguna yang memanfaatkan fitur *remote desktop*.

Perangkat lunak lainnya yang menjadi kebutuhan utama pengguna komputer adalah aplikasi perkantoran (*office*). Microsoft Office, OpenOffice[5], StarOffice, dan IBM Lotus Symphony merupakan beberapa nama produk yang dikhususkan untuk menunjang kebutuhan manipulasi dokumen seperti pengolah kata, *spreadsheet*, dan penunjang kebutuhan presentasi.

Persoalan kompatibilitas ekstensi dokumen yang dihasilkan pada masing-masing perangkat lunak office masih menjadi persoalan menarik. Seperti Openoffice.org saat ini mendukung semua format dokument standar yang digunakan oleh Microsoft Office secara *de facto* atau *de jure*, yaitu ekstensi dokumen berupa **.doc**, **.xls**, dan **.ppt**. Tetapi Microsoft Office sampai saat ini belum memberikan dukungan terhadap format dokumen .odt, .ods, dan .odp yang telah diakui sebagai standar dunia melalui ISO/IEC 26300.

2. LANDASAN TEORI

2.1 Rekayasa Perangkat Lunak

Menurut Fritz Bauer, rekayasa perangkat lunak (*Software Engineering*) adalah penggunaan prinsip-prinsip rekayasa untuk mendapatkan perangkat lunak yang ekonomis dan dapat bekerja secara efisien pada mesin nyata. Dari sudut pandang lain, rekayasa perangkat lunak adalah pendekatan sistematis untuk merekayasa perangkat lunak yang handal atau bermutu, tepat waktu dan dengan biaya yang optimal.

2.2 Sistem Operasi

Sistem operasi adalah suatu program yang mengontrol eksekusi program-program aplikasi dan berfungsi sebagai antarmuka (*interface*) antara pengguna komputer dengan perangkat keras (*hardware*) komputer. Sistem operasi dapat dianggap memiliki tiga tujuan:

1. Kemudahan: Sistem operasi membuat komputer menjadi lebih mudah dipakai.
2. Efisiensi: Sistem operasi memungkinkan sumber daya sistem komputer untuk digunakan dengan cara yang efisien.
3. Kemampuan Berkembang: Sistem operasi harus disusun untuk

memungkinkan pengembangan yang efektif di masa yang akan datang secara cepat dan efisien.

2.3 Komponen Sistem Operasi

Menurut Avi Silberschatz, Peter Galvin, dan Greg Gagne, umumnya sebuah sistem operasi modern mempunyai komponen sebagai berikut:

1. Manajemen proses
2. Manajemen memori utama
3. Manajemen berkas
4. Menajemen sistem masukan/keluaran
5. Manajemen penyimpanan sekunder
6. Sistem proteksi
7. Jaringan
8. *Command-Interpreter system*

Sedangkan menurut A.S. Tanenbaum, sistem operasi mempunyai empat komponen utama, yaitu:

1. Manajemen proses
2. Manajemen masukan/keluaran
3. Manajemen memori
4. Manajemen berkas

2.4 Pemrograman Berorientasi Objek

Tiga aspek penting dalam pemrograman berorientasi objek:

1. Menggunakan objek-objek bukan algoritma-algoritma sebagai blok-blok bangunan logik dasar.
2. Masing–masing objek adalah instan suatu kelas.
3. Kelas-kelas saling berhubungan lewat keterhubungan pewarisan.

Pemrograman berorientasi objek adalah metode implementasi dimana program diorganisasikan sebagai komunitas objek yang bekerja sama berinteraksi. Masing–masing merepresentasikan instan suatu kelas yang menyediakan sejumlah layanan yang relative sederhana. Objek-objek saling berkomunikasi untuk mempertukarkan informasi. Perilaku komplek diperoleh dengan kerjasama antar objek-objek. Kelas–kelas dapat merupakan anggota hirarki yang disatukan lewat hubungan pewarisan.

2.5 Definisi UML

UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas–kakas yang diintegrasikan lewat XML.

Standar UML dikelola oleh OMG (*Object Management Group*).

2.6 AJAX

Ajax adalah suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web interaktif. Tujuannya adalah untuk memindahkan sebagian besar interaksi pada

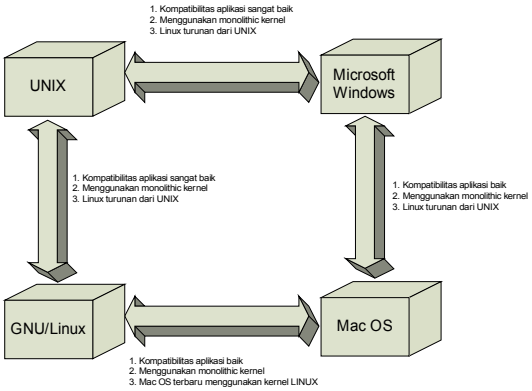
komputer *web surfer*, melakukan pertukaran data dengan *server* di belakang layar, sehingga halaman web tidak harus dibaca ulang secara keseluruhan setiap kali pengguna melakukan perubahan. Hal ini akan meningkatkan efektivitas, kecepatan, dan *usability*.

Ajax bukanlah teknologi spesifik, tetapi merupakan gabungan dari teknologi yang dipakai bersamaan. Bahkan teknologi turunan/komposit yang berdasarkan Ajax, seperti AFLAX sudah mulai bermunculan.

3. ANALISIS MASALAH

Kendala-kendala yang muncul berdasarkan kondisi yang ada di negara Indonesia saat ini yaitu (Gambar 3.1):

1. Kompatibilitas paket aplikasi di setiap sistem operasi. Paket aplikasi yang dapat diinstal dan dijalankan pada satu sistem operasi belum tentu dapat dijalankan pada sistem operasi lainnya.
2. Harga lisensi
   Microsoft Windows dan Microsoft Office menggunakan lisensi komersial, dimana pengguna akan dibebankan biaya untuk setiap satu PC atau *Notebook* yang menggunakannya. Mac OS juga menggunakan lisensi yang sama. UNIX menggunakan lisensi BSD, dimana pengguna boleh mendapatkannya secara gratis tetapi tidak memiliki akses terhadap *source code* kernelnya, kecuali untuk kebutuhan non-komersial dan riset untuk pendidikan. GNU/Linux menggunakan lisensi GNU GPL yang membebaskan penggunanya untuk menggunakan, memodifikasi, merilis ulang dengan nama yang berbeda selama lisensi yang digunakan tetap sama, yaitu GNU GPL.
3. Kesulitan migrasi aplikasi dan dokumen. Proses migrasi dapat terjadi disebabkan berbagai faktor diantaranya kemampuan keuangan perusahaan dalam membayar lisensi perangkat lunak, kebutuhan aplikasi yang lebih baik layanan dan dukungan teknisnya, dan migrasi dari dokumen yang terkena virus. Kebiasaan menggunakan satu aplikasi tertentu dalam waktu yang lama, format ekstensi dokumen yang digunakan, dan fitur yang digunakan pada aplikasi sebelumnya akan tetapi tidak ditemukan di aplikasi yang baru, mengakibatkan proses migrasi aplikasi dan dokumen menjadi lebih sulit.
4. Dukungan teknis (*technical support*).
   Pengguna biasanya lebih menyukai aplikasi dengan dukungan teknis yang baik.



Gambar 3.1 Kompatibilitas aplikasi

4. PERANCANGAN

4.1 Pemodelan *Use Case*

Pemodelan *use case* adalah pemodelan sistem dari perspektif pandangan pemakai akhir (*end user*). Sasaran pemodelan *use case* sebagai berikut:

1. Mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan yang disepakati antara pemakai/pengguna dan pengembang (*developer)*.
2. Menyediakan deskripsi yang jelas dan tidak ambigu mengenai cara pengguna dan sistem saling berinteraksi.
3. Menyediakan basis untuk pengujian validasi.

4.2 Identifikasi Actor

Aktor yang terlibat dalam sistem yaitu pemakai (*user*). Pemakai dikhususkan pada pegawai/karyawan dari sebuah instansi atau perusahaan, karena nantinya sistem ini digunakan pada lingkungan khusus instansi dan perusahaan.

4.3 Diagram *Use Case*

Diagram *use case* memperlihatkan hubungan-hubungan yang terjadi antara aktor-aktor dengan *use case-use case* dalam sistem. Salah satu manfaat dari diagram *use case* adalah untuk komunikasi. Calon pengguna sistem/perangkat lunak dapat mengamati diagram *use case* untuk mendapatkan pemahaman yang utuh tentang sistem yang akan dikembangkan oleh pengembang sistem.

4.4 Diagram Sekuen

Kebutuhan interaksi objek yang disusun dalam suatu urutan waktu dimodelkan dalam diagram sekuen. Diagram ini memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *use case*.

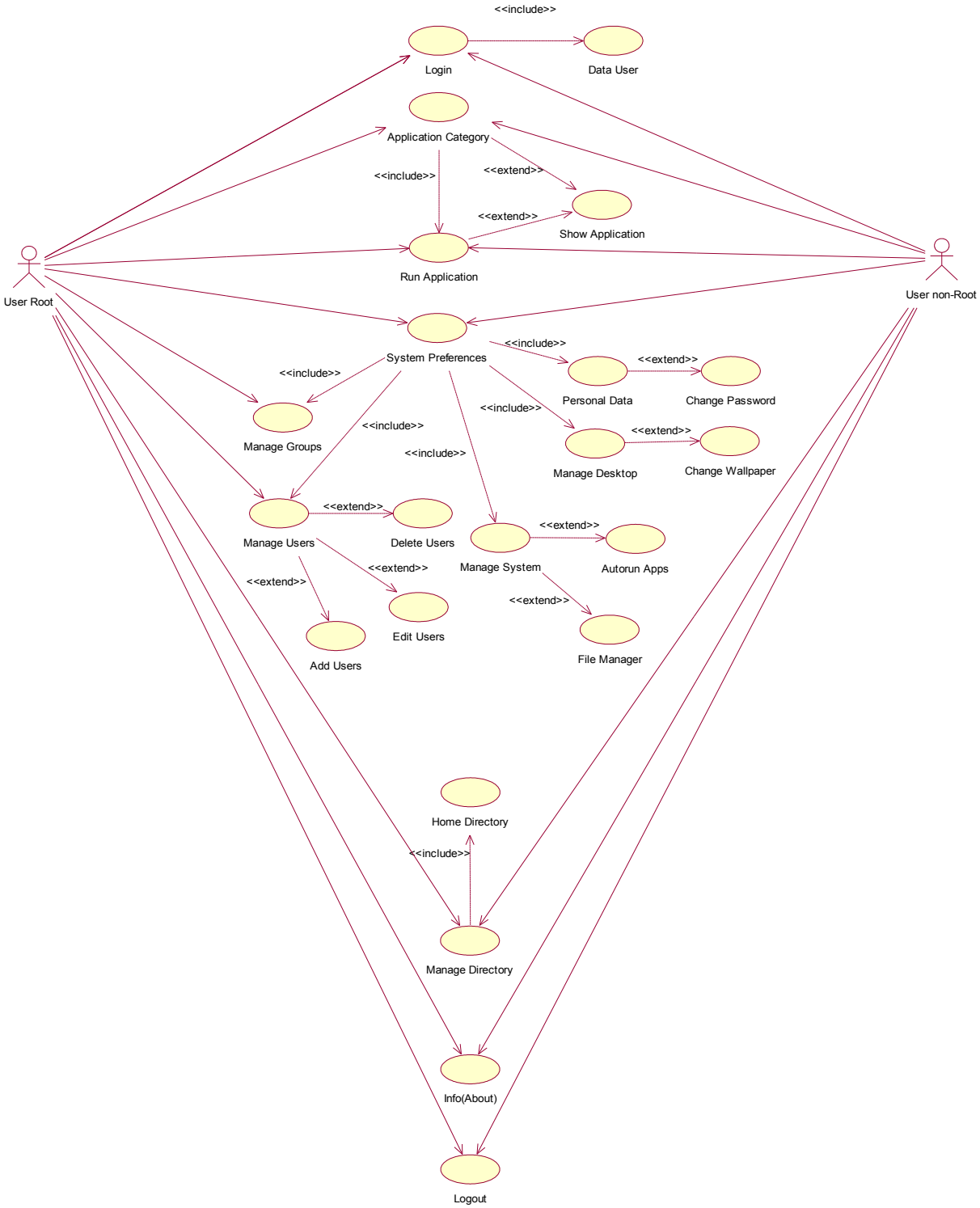
4.5 Diagram Sekuen Login

Diagram sekuen login terdiri dari beberapa tahap (Gambar 4.2), yaitu:

1. Pengguna (*user root* dan *non-root*) memasukkan *username* dan *password* ke dalam suatu form input (*text field*)
2. Sistem melakukan proses pengecekan terhadap hasil masukkan di *textfield*.
3. Pada tahap pengecekan, sistem memanggil fungsi *CekInput()* untuk mengecek validasi dengan mencocokkan data yang ada pada tabel *username* dan *password*.

4. Setelah masukan valid, proses akan menjalankan fungsi *setView()* untuk menampilkan halaman utama sistem operasi sesuai dengan hak akses dari pengguna yang bersangkutan.

Gambar 4.1. Diagram *use case* sistem operasi dan office



Gambar 4.2 Diagram sekuen proses login

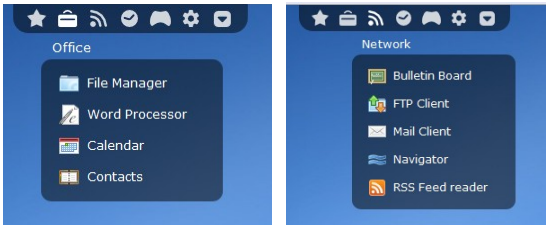
5. IMPLEMENTASI

Untuk masuk dalam web sistem operasi, pengguna harus memasukkan nama dan password pengguna pada halaman awal seperti yang terlihat pada gambar 5.1.



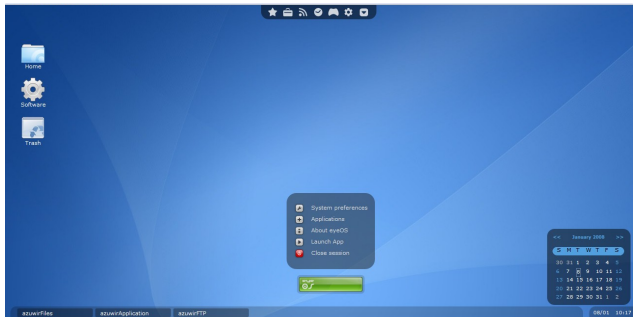
Gambar 5.1 Tampilan awal web sistem operasi

Setelah proses login berhasil, pengguna masuk ke halaman utama dari web sistem operasi. Halaman utama terdiri dari 4 bagian kolom, yaitu pada kolom teratas pengguna akan melihat pilihan menu kategori aplikasi, seperti terlihat pada gambar 5.2.



Gambar 5.2 Pilihan menu kategori

Pada kolom tengah terdapat *desktop* dan *icon/shortcut* untuk menjalankan aplikasi secara langsung. Pada kolom bagian bawah terdapat *Start Menu* dan *Bottom Bar* untuk meletakkan aplikasi yang diminimasi oleh pengguna.



Gambar 5.3 Halaman utama

Tampilan selengkapnya dapat dilihat di website http://azuwir.web.id dibagian link Tugas Akhir.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan :

- Kompatibilitas aplikasi pada setiap sistem operasi masih menjadi persoalan pelik.
- Sistem operasi harus mampu mengakomodasi kemampuan *multiuser, multitasking,* dan *remote desktop*.

6.2 Saran :

- Perancangan sistem menggunakan pemrograman berorientasi objek harus menggunakan pendekatan yang tepat agar tidak terjadi kesulitan pengkodean.
- Sistem operasi harus menyediakan kompatibilitas lintas *platform*.

7. DAFTAR PUSTAKA

[1] Adi Nugroho, *Rational Rose untuk Pemodelan Berorientasi Objek*, Penerbit Informatika, Bandung, 2005.

[2] Djon Irwanto, *Perancangan Object Oriented Software dengan UML*, Penerbit Andi, Jogjakarta, 2006.

[3] Babin, Lee, 2007, *Beginning Ajax with PHP From Novice to Professional*, Apress California.

[4] Pressman, Roger S., Ph. D., 2002, *Software Engineering*, Fourth Edition, McGraw-Hill.

[5] Stallings, William, 2005, *Operating Systems : Internal and Design Principles,* Fourth Edition, Prentice Hall.

[6] Openoffice.Org Team, *Getting Started with OpenOffice 2.x,* Third Edition, 2005-2007.

[7] Openoffice.Org Team, *Migration Guide,* Version 2.5, 2006.

[8] http://en.wikipedia.org/OperatingSystem, diakses tanggal 20 Desember 2007.

[9] http://en.wikipedia.org/AJAX, diakses tanggal 20 Desember 2007.