

## IMPLEMENTASI ARSITEKTUR *THREE TIER* PADA PELAYANAN JASA KALIBRASI (STUDI KASUS DI P2KIM LIPI)

Wawan Wardiana

Pusat Penelitian Informatika, Lembaga Ilmu Pengetahuan Indonesia  
Komplek LIPI Gd 20 Lt 3, Jl. Sanguriang Bandung, 40135

### Abstrak

Pusat Penelitian Kalibrasi Instrumentasi dan Metrologi yang berada di lingkungan Lembaga Ilmu Pengetahuan Indonesia (LIPI) memiliki banyak laboratorium-laboratorium yang digunakan sebagai sarana penelitian, selain itu laboratorium-laboratorium tersebut juga seringkali digunakan untuk melayani jasa kalibrasi bagi instansi / perusahaan yang memerlukannya. Sistem sekarang dilakukan secara konvensional, pengguna jasa akan datang dengan membawa peralatan yang akan dikalibrasi dan melakukan negosiasi dan konfirmasi, dan bila memungkinkan maka peralatan tersebut dapat dikalibrasi. Adakalanya pengguna jasa datang dari jauh ternyata alat tersebut tidak bisa diterima, dikarenakan tidak adanya peralatan yang memiliki fungsi untuk itu. Untuk selanjutnya pengguna menunggu hasilnya, atau dengan cara melakukan komunikasi telepon untuk mengetahui sampai sejauh mana pelaksanaan kalibrasinya tersebut.. Tetapi saat menelepon, bisa saja tidak ada petugas yang mengangkat telepon tersebut. Dalam tulisan ini akan dibahas bagaimana sistem konvensional tersebut dirancang menjadi sistem yang berbasis web dengan mengimplementasikan arsitektur *three tier*, sehingga pengguna dapat melakukan pengecekan terlebih dahulu dengan peralatan yang memiliki kemampuan untuk alat tersebut, melakukan negosiasi dan konfirmasi secara online, serta dapat menelusuri histori dari peralatan yang dikalibrasi secara online. Dalam mengembangkan sistem ini perangkat lunak yang digunakan adalah perangkat lunak yang *open source*, seperti Apache Web Server, PHP.

**Kata kunci:** Pelayanan Kalibrasi, Arsitektur *Three Tier*, Sistem Terdistribusi

### 1. Pendahuluan

Dalam melakukan tugasnya, komponen-komponen yang berada dalam suatu sistem terdistribusi seringkali harus melakukan komunikasi dengan komponen yang lain melalui antarmuka dari komponen. Biasanya dalam melakukan komunikasi ini digunakan suatu *middleware* yang memang memiliki *socket* yang siap dihubungkan dengan komponen.

*Middleware* merupakan sebuah kelas teknologi perangkat lunak yang didesain untuk membantu mengatur kompleksitas dan heterogenitas yang terdapat dalam sistem terdistribusi. Ia ditentukan sebagai lapisan perangkat lunak di atas sistem operasi tapi di bawah program aplikasi yang menyediakan abstraksi pemrograman untuk melewati sistem terdistribusi. Dalam melaksanakan tugasnya, ia menyediakan suatu blok pada tingkat yang lebih tinggi ke pemrogram dibanding *Application Program Interface (API)*. Ini akan sangat mengurangi beban pemrogram aplikasi dengan membebaskan mereka dari kesalahan pemrograman yang cenderung terjadi dan membosankan. *Middleware* seringkali disebut sebagai *plumbing* karena ia menghubungkan aplikasi terdistribusi dengan pipa data dan melewatkan data antara aplikasi melalui pipa tersebut.(1)

Sistem pelayanan jasa kalibrasi yang dilakukan oleh P2KIM-LIPI khususnya pada laboratorium suhu dilakukan dengan menggunakan beberapa prosedur yang sudah baku, walaupun ada kalanya prosedur tersebut dilewati karena satu dan

lain hal. Namun secara keseluruhan sistem pelayanan yang telah dilakukan sudah baik.

Pengguna laboratorium ini sudah banyak, dan terdiri dari berbagai kalangan industri, instansi pemerintah, maupun kalangan swasta lainnya. Pelayanan yang dilakukan sekarang ini masih menggunakan cara-cara konvensional, sedangkan para pengguna jasa tersebut tidak sedikit yang bertempat tinggal jauh dari laboratorium tersebut, sehingga pengguna jasa sudah tiba di laboratorium tersebut, ternyata tidak memiliki peralatan yang dimaksud untuk melakukan kalibrasi terhadap peralatan tersebut.

Selain itu diperlukan juga informasi yang mutakhir oleh para pengguna tentang peralatan dan fasilitas lainnya yang dimiliki oleh laboratorium dimaksud, agar dengan mengetahui informasi tersebut para pengguna dapat mempertimbangkan pelaksanaan jasa kalibrasi pada laboratorium dimaksud.

Implementasi arsitektur *three-tier* pada sistem ini dimaksudkan agar dengan menggunakan arsitektur ini maka tingkat keamanan data, skalabilitas, dan fleksibilitas update data dapat terjamin dengan baik.

### 2. Konsep Arsitektur *Three Tier*

Pada aplikasi *two-tier* selalu membutuhkan penyimpanan data pada sebuah *server*. Biasanya, informasi disimpan dalam sistem *file*. Bagaimanapun hal integritas data meningkat saat banyak *client* secara simultan meminta pelayanan

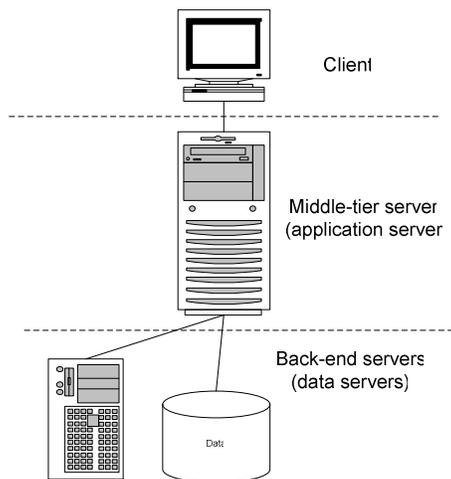
server untuk menyelesaikan pekerjaannya. Karena sistem *file* secara umum memiliki pengendalian konkurensi yang terbaik (penguncian *file* di beberapa *platform*, dan semuanya selalu tidak berhasil dengan baik), solusi yang paling umum adalah dengan menambahkan sebuah program ketiga (*the third program*), atau *database*.

*Database* dikhususkan dalam penyimpanan, temu kembali dan mengurutkan data. Tidak seperti arsitektur *two-tier* memisahkan GUI dan *business logic*, sebuah arsitektur *three-tier* mengijinkan kita memisahkan *business logic* dan pengaksesan data. Kita dapat juga melengkapi sepenuhnya optimisasi mengurut data dan metoda pengambilan data, dan dilengkapi juga untuk replikasi (*replication*), cadangan (*backup*), redundansi (*redundancy*), dan prosedur khusus untuk menjaga keseimbangan data yang dibutuhkan. Pemisahan kode *client* dan kode *server* menambah skalabilitas dari aplikasi tersebut, sehingga penempatan data pada suatu proses yang berdedikasi, *host* ataupun serangkaian *host*.

Sebuah aplikasi *three-tier* menggabungkan sebuah program ketiga, biasanya *database*, tempat dimana *server* menyimpan datanya. Aplikasi *three-tier* merupakan sebuah perbaikan dari *two-tier architecture*. Aliran informasi pada dasarnya tetap linier, sebuah permintaan (*request*) datang dari *client* ke *server*, *server* tersebut mengambil atau menyimpan data dari/ke *database*, *database* tersebut mengembalikan informasi ke *server*, *server* tersebut mengembalikan informasi tersebut ke *client*.(2)

Pada arsitektur *three-tier* ada tiga bagian penting yang paling mendasar, adanya lapisan-lapisan yang menunjukkan bahwa arsitektur tersebut *three-tier*, yaitu:

- User tier (presentation layer, user services)*
- Business tier (business logic layer, business service)*
- Data tier (data layer, data service)*



Gambar 1. Arsitektur Three-tier Client/Server

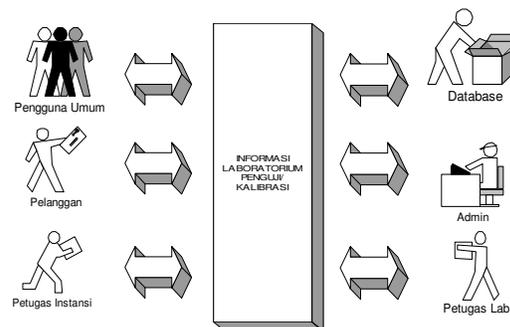
Keuntungan dari arsitektur *three-tier* ini adalah:

- Segala sesuatu mengenai *database* terinstalasikan pada sisi *server*, begitu pula dengan pengkonfigurasinya. Hal ini membuat harga yang harus dibayar lebih kecil.
- Adanya *middle-tier* untuk akses *database*, membuat penggantian pada skema *database* maupun *database* itu sendiri, tanpa harus penginstalasian ulang pada sisi *client*.
- Apabila terjadi kesalahan pada salah satu lapisan tidak akan menyebabkan lapisan lain ikut salah
- Perubahan pada salah satu lapisan tidak perlu menginstalasi ulang pada lapisan yang lainnya dalam hal ini sisi *server* ataupun sisi *client*.

### 3. Sistem Pelayanan Laboratorium Kalibrasi

Dalam sistem ini semua data laboratorium kalibrasi akan dikelola dan dipublikasikan oleh seorang pengelola atau lebih dikenal dengan sebutan *administrator*. Pengguna sistem ini adalah siapapun yang memerlukan informasi dan jasa kalibrasi laboratorium tersebut. Oleh karena itu diharapkan dengan sistem ini dapat mempercepat penyebaran informasi dan layanan jasa dari unit kerja yang ada di lingkungan P2KIM

Setiap pengguna yang menggunakan jasa laboratorium ini dapat pula melihat secara langsung informasi status peralatan atau bahan yang dikalibrasi, apakah sudah selesai atau belum, atau apakah masih berada di gudang, atau sudah dilaksanakan tetapi belum selesai, atau bahkan tidak bisa dilaksanakan secepatnya karena adanya sesuatu hal. Sistem pengelolaan informasi laboratorium kalibrasi ini dirancang sebagai sebuah aplikasi perangkat lunak yang interaktif yang dapat dijalankan pada semua *Platform browser* dan dikembangkan dengan metoda berorientasi objek.



Gambar 2. Gambaran umum sistem

*Administrator* akan melayani pendaftaran laboratorium tersebut dan membantu mempublikasikan informasi tersebut. Selain itu *administrator* juga akan melayani bila ada pengguna sistem ini menginginkan status penggunaannya diubah dari pengguna umum menjadi pengguna tercatat sehingga pengguna tersebut dapat

menggunakan haknya untuk melakukan pemesanan jasa kalibrasi atau melakukan pengecekan terhadap status peralatan yang dikalibrasi di laboratorium terkait.

Pengguna sistem lainnya adalah petugas laboratorium, yang mempunyai tugas dan wewenang untuk melayani pengguna tercatat yang akan menggunakan layanan jasa laboratorium terkait. Petugas ini akan melayani secara langsung menggunakan media *electronic mail* untuk berkomunikasi dengan pengguna jasa, dengan memberikan informasi sesuai permintaan pengguna jasa. Sebelumnya pengguna tercatat dapat melihat antrian yang ada dalam layanan laboratorium tersebut, sehingga bila antrian cukup banyak, maka pengguna tercatat dapat mencari laboratorium sejenis lainnya, karena di lingkungan LIPI laboratorium yang mempunyai fungsi yang sama bisa berada pada unit kerja yang berbeda.

Jika terjadi pemesanan layanan jasa laboratorium terkait, maka sistem akan mencatat dan menambahkan dalam daftar antrian pada layanan jasa laboratorium tersebut, begitu juga sebaliknya bila sudah selesai maka sistem akan mengurangi antrian pada laboratorium terkait. Sistem memperlihatkan juga status bahan atau peralatan pengguna tercatat yang sedang diuji atau dikalibrasi.

Untuk menjaga keamanan data dan informasi yang ada dalam direktori tersebut, maka setiap pengguna sistem kecuali pengguna umum harus memasukkan *login\_name* terlebih dahulu untuk dicek status dan kebutuhannya serta pembagian akses datanya oleh sistem.

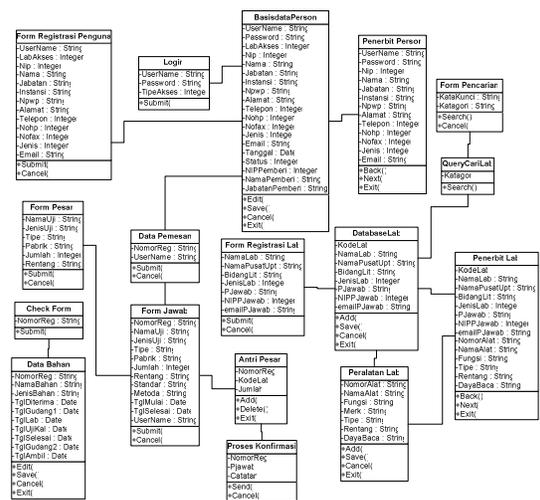
Seorang *administrator* selain mengatur dan mengendalikan sistem, juga melakukan penyeleksian terhadap informasi yang akan ditampilkan dalam direktori ini, sehingga informasi yang lulus seleksi akan disimpan datanya dalam database yang dikendalikannya.

Jika terjadi pemesanan jasa, sistem akan menyeleksi petugas laboratorium unit kerja mana yang harus melayani transaksi pemesanan tersebut, kemudian bila sudah terjadi kesepakatan kedua belah pihak, maka pengguna tercatat tersebut akan mendapatkan nomor registrasi untuk setiap produk, bahan atau peralatan yang akan diuji atau dikalibrasi pada unit kerja terkait. Nomor registrasi yang diterima dapat digunakan oleh pengguna tercatat untuk mengecek status pengujian atau kalibrasi yang dilakukan oleh laboratorium terkait, setelah sebelumnya *login* sebagai pengguna tercatat.

#### 4. Perancangan Class/Object

Berdasarkan hasil analisis di dapatkan kelas-kelas utama yang akan dikembangkan dalam pembuatan aplikasi. Gambar 2. berikut menggambarkan diagram kelas yang didapat dari pemetaan hasil analisis ke dalam perancangan termasuk didalamnya terdapat identifikasi relasi

antar *class*, identifikasi *attribute*, serta identifikasi *method*.



Gambar 3. Identifikasi Diagram Kelas

#### 5. Implementasi Sistem

Implementasi yang dilakukan adalah dengan membentuk prototipe perangkat lunak berupa suatu situs yang berisi informasi laboratorium kalibrasi. Dalam implementasi ini mengacu pada hasil analisis dan perancangan yang telah dibuat sebelumnya. Pada implementasi ini dibutuhkan dukungan perangkat lunak maupun perangkat keras.

##### 5.1 Perangkat Lunak Pendukung

Perangkat lunak pendukung yang digunakan pada saat implementasi perangkat lunak ini terdiri dari : Sistem Operasi Windows XP Professional, Standar sistem operasi yang dapat mendukung pengelolaan data dengan 32 bit dan antarmuka yang dinamis; Macromedia Dreamweaver MX, digunakan untuk mempermudah perancangan antarmuka; Visio Technical, sebagai kakas untuk membuat gambar-gambar yang diperlukan pada pembuatan dokumentasi; EditPlus2, untuk mempermudah penulisan program; Web Server Apache, standar web server yang cukup handal; PHP, perangkat lunak yang mendukung pemrograman web yang dinamis; MySQL, menyediakan fasilitas untuk pengelolaan basis data yang cukup optimal

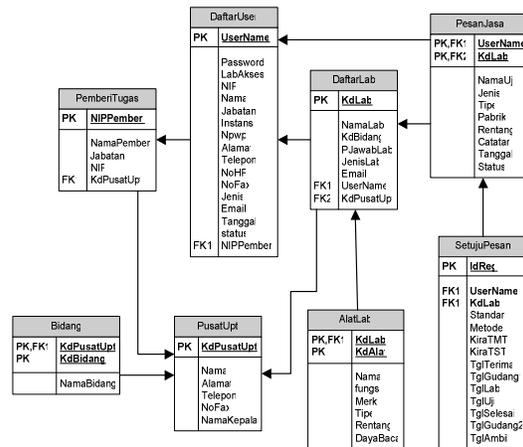
##### 5.2 Perangkat Keras Pendukung

Perangkat pendukung lainnya adalah perangkat keras, untuk mengimplementasikan kegiatan ini digunakan perangkat keras berupa : Komputer Personal (Kompatibel), Processor Pentium 4 dengan kecepatan 2.8 GHz, dengan RAM 256 Mbyte; Notebook IBM Think Pad P4 - 2.8 GHz yang digunakan sebagai *workstation / client*; Monitor LG F700P yang mendukung resolusi berbasis windows; Harddisk 40 Gbyte, cukup untuk

penyimpanan sistem operasi, perangkat lunak aplikasi pendukung, perangkat lunak yang dibangun dan data; dan Printer Laserjet HP 2200 D yang lebih dari cukup dalam pencapaian tingkat kejelasan pencetakan dokumen.

### 5.3 Basis Data

Hasil perancangan diagram kelas yang ada pada bahasan sebelumnya dipetakan menjadi basis data dalam bentuk *Entity Relational Diagram* (3). Untuk implementasi menggunakan sistem pengelolaan basis data MySQL.



Gambar 4. Entity Relational Diagram

## 6. Kesimpulan

- Sistem pengelolaan informasi jasa laboratorium kalibrasi dapat menjadi alternatif untuk dijadikan media penyebaran informasi, inventarisasi dan terintegrasinya informasi peralatan-peralatan laboratorium secara efektif.
- Dengan mengembangkan sistem arsitektur *three-tier* yang berbasis web, maka pengelolaan sistem dapat dilakukan dimana saja, kapan saja, oleh pengguna yang berwenang sebagai administrator, sistem juga dapat memisahkan *user interface* (UI) dari *business logic* yang ada.
- Sarana komunikasi antara pengelola laboratorium dengan pengguna jasa laboratorium yang dibuat relatif lebih efektif dan efisien. Karena pengguna maupun petugas laboratorium dapat mengakses data dan informasi yang diinginkan dimana saja dan kapan saja tanpa terhalang oleh waktu dan tempat.

### Daftar pustaka

- [1] Coulouris, George, *Distributed Systems: Concept and Design 3<sup>rd</sup> edition*, McGraw Hill, New York, 2002
- [2] Chaffee, Alex, *One, Two, Three or n tiers, Should you hold back the tiers of your application?*, url: <http://www.javaworld.com/javaworld/jw-01-2000/jw-01-ssj-tiers.html>.

downloaded by Wawan Wardiana, date : 18-12-2003, time: 14.30

- [3] Korth, Silberschatz, *Database System Concept 4<sup>th</sup> edition*, McGraw Hill, New York, 2002.
- [4] Wardiana, Wawan, *Pengelolaan Informassi Laboratorium Penguji/Kalibrasi Berbasis Web*, Tesis Magister, ITB, 2004.