

## PENERAPAN ARSITEKTUR THREE-TIER DENGAN COM+ DALAM PORTAL JURNAL

**Dimara Kusuma Hakim, Fathul Wahid**

Laboratorium Sistem Informasi dan Rekayasa Perangkat Lunak (SIRKEL)  
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia  
Jl. Kaliurang Km. 14 Yogyakarta 55501  
Telp. (0274) 895287 ext. 122, Faks. (0274) 895007 ext. 148  
E-mail: dimarahin@yahoo.com, fathulwahid@fti.uii.ac.id

### Abstrak

Penerapan arsitektur three-tier digunakan untuk mengatasi kelemahan-kelemahan model two-tier dalam hal penggunaan kembali komponen, skalabilitas, dan perawatan sistem. COM+ adalah salah satu platform yang digunakan untuk memecahkan masalah-masalah tersebut karena kemampuannya untuk (a) memusatkan logika aplikasi pada middleware, (b) mendistribusikan beban proses data pada beberapa mesin, (c) meningkatkan keamanan dengan kemampuan menyembunyikan data, dan (d) memungkinkan perawatan yang mudah dengan tanpa instalasi database driver pada setiap client. Dalam model three-tier, aplikasi dibuat menjadi tiga partisi: (1) aplikasi client yang menyediakan user service, (2) aplikasi middleware yang menyediakan business service, dan (3) aplikasi database menyediakan data service. Paper ini mempresentasikan studi kasus penggunaan model three-tier untuk aplikasi Portal Jurnal.

**Kata kunci:** three-tier, portal jurnal, COM+, middleware, client server

### 1. Pendahuluan

Bentuk Sistem Informasi dengan teknologi *client/server two-tier* memiliki kelemahan-kelemahan dalam hal penggunaan kembali komponen, skalabilitas, dan perawatan sistem. Kelemahan-kelemahan model *client/server 2-tier* akan terasa pada suatu sistem yang kompleks dan melibatkan banyak transaksi. Oleh karena itu, beberapa perusahaan mulai berpindah dari teknologi *client/server two-tier* ke teknologi *client/server three-tier* yang menyediakan kemampuan dalam hal perluasan sistem, kemudahan perawatan sistem, kemudahan dalam penggunaan kembali komponen, dan pendistribusian beban.

Arsitektur *three-tier* dengan teknologi COM+ dipilih dalam penelitian ini karena memiliki banyak keuntungan seperti memusatkan logika aplikasi pada *middle-tier*, dapat mendistribusikan beban proses data pada beberapa mesin, meningkatkan keamanan dengan kemampuan menyembunyikan data, cepatnya perawatan dengan tanpa instalasi database driver pada setiap client, dan yang paling utama adalah banyaknya dukungan dari berbagai vendor *programming tools*, sistem operasi, dan database server terhadap teknologi ini, sehingga memudahkan dalam hal *developing*.

Kemudian, studi kasus yang diambil adalah Portal Jurnal. Portal Jurnal adalah suatu aplikasi web yang bertujuan untuk mengembangkan dan menyebarkan ilmu pengetahuan dan teknologi dalam bentuk jurnal. Aplikasi web ini disediakan sebagai wadah bagi dan pembaca karya-karya ilmiah.

### 2. Landasan Teori

Arsitektur three-tier merupakan bagian dari client server, yang merupakan salah satu arsitektur pendukung sistem terdistribusi.

#### 2.1 Sistem Terdistribusi

Sistem terdistribusi merupakan suatu bentuk arsitektur sistem dimana komponen-komponen di dalamnya dapat berjalan di beberapa komputer yang berbeda secara bersamaan. Komponen-komponen dari sistem tersebut dapat berkomunikasi satu sama lain dan dirancang agar dapat beroperasi secara terpisah.

Beberapa arsitektur yang digunakan oleh sistem terdistribusi diantaranya:

- Client-server
- Tightly coupled (clustered)

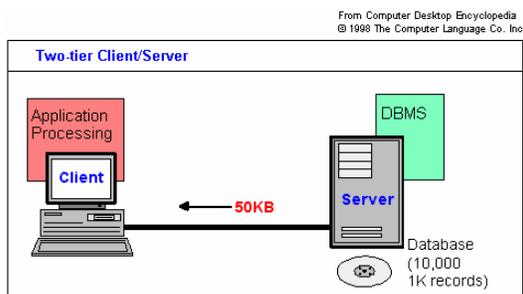
#### 2.2 Client-server

Prinsip utama dari model *client/server* adalah suatu proses dapat dipisahkan menjadi 2, yaitu pada proses client dan proses server, dimana kedua proses tersebut menggunakan suatu protokol tertentu agar dapat saling berkomunikasi.

#### 2.3 Two-tier Client/Server

Dalam model *client/server*, sebelumnya dikenal arsitektur *monolithic*, contoh yang sederhana adalah pembangunan suatu aplikasi Sistem Informasi yang *stand alone*, dimana model *file server* menjadi dominan. Padahal, sebenarnya *file server* lebih mengarah pada suatu layanan yang disediakan pada level sistem operasi bukan pada level aplikasi yang dikembangkan itu sendiri.

Berangkat dari itu, muncul model *client/server two-tier*. Contohnya adalah adanya penggunaan database server (seperti SQL Server, Oracle Server, Sybase, MySQL, ProgresSQL, Interbase, FireBird). Dengan penerapan database server ini, proses akan terbagi menjadi dua, yaitu untuk pemrosesan data dilakukan oleh *database server*, sedangkan untuk antar muka pemakainya diproses pada *client* (lihat Gambar 1).

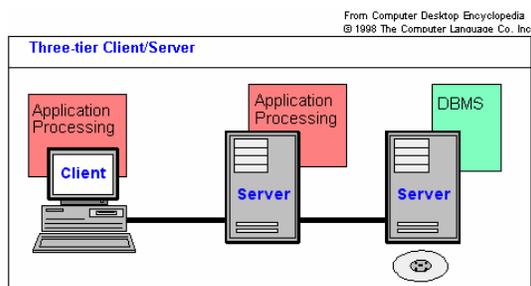


Gambar 1. Model *two-tier*

## 2.4 Three-tier Client/Server

Pada model *two-tier*, terjadi penggunaan teknologi ADO, RDO atau ODBC untuk melakukan akses terhadap database. Komponen tersebut harus melekat pada aplikasi client dan jika suatu permintaan dikirimkan ke database server, kemudian semua hasil akan dikirimkan ke client, lalu client akan melakukan proses terhadap data tersebut dengan *business logic*-nya (lihat Gambar 2). Di sini, dengan *model client/server two-tier*, bottleneck terjadi pada *traffic* jaringannya, walaupun *traffic*-nya lebih rendah dibandingkan dengan file server.

Dengan acuan tersebut, muncul model *client/server three-tier* dan *multi-tier*, dimana *business logic* dipisahkan ke *server*. Oleh karena *business logic* dikerjakan oleh *server*, maka data yang terkirimkan hanyalah data hasil pengolahan *business logic*. Tentu saja dengan demikian akan mengurangi padatannya lalu lintas jalur network yang ada. Teknologi yang dapat digunakan untuk membangun model ini antara lain COM+, CORBA, CGI, ASP, dan sebagainya.



Gambar 2. Model *three-tier*

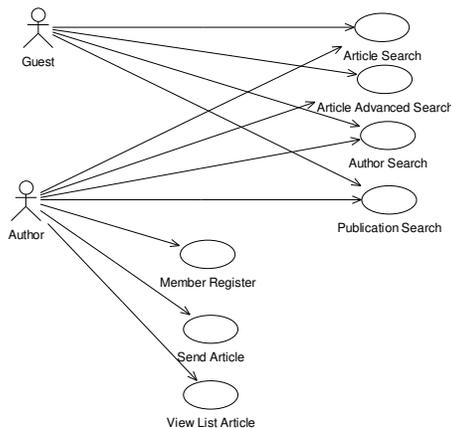
## 2.5 COM+

COM+ adalah suatu teknologi *middleware* yang merupakan solusi Microsoft untuk membuat application server atau yang sering disebut sebagai *middle-tier*. Pada *middle-tier* inilah, beberapa *business object* disimpan.

COM+ memungkinkan penggunaan objek-objek yang ada lintas jaringan komputer oleh aplikasi client. Di balik layar, COM+ menggunakan sebuah *protocol* yang bernama Remote Procedure Call (RPC) untuk mengeksekusi *method-method* yang berada didalamnya lintas proses dan lintas batasan host.

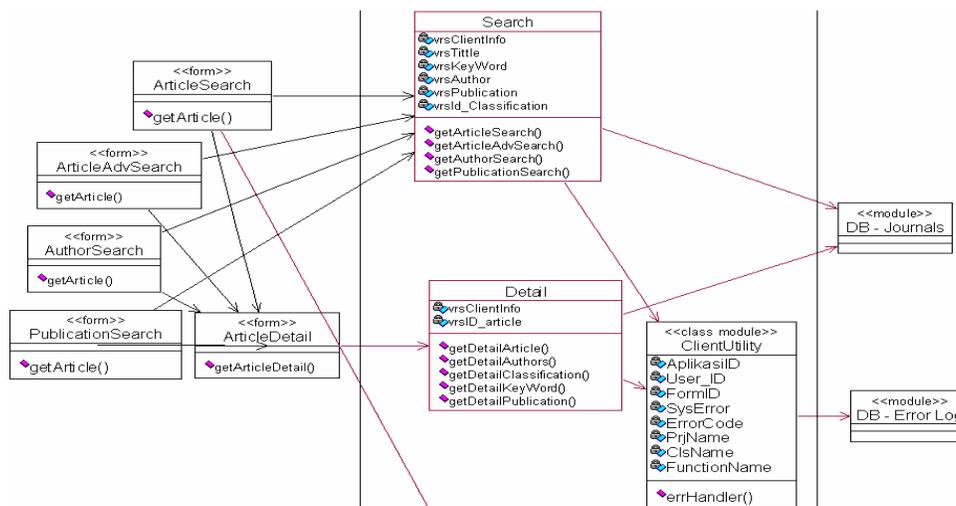
## 3. Perancangan

Untuk merancang sistem, digunakan metodologi berarah obyek agar permasalahan sistem dapat dipecah dalam obyek-obyek kecil. Bahasa pemodelan yang digunakan adalah Unified Modelling Language (UML).

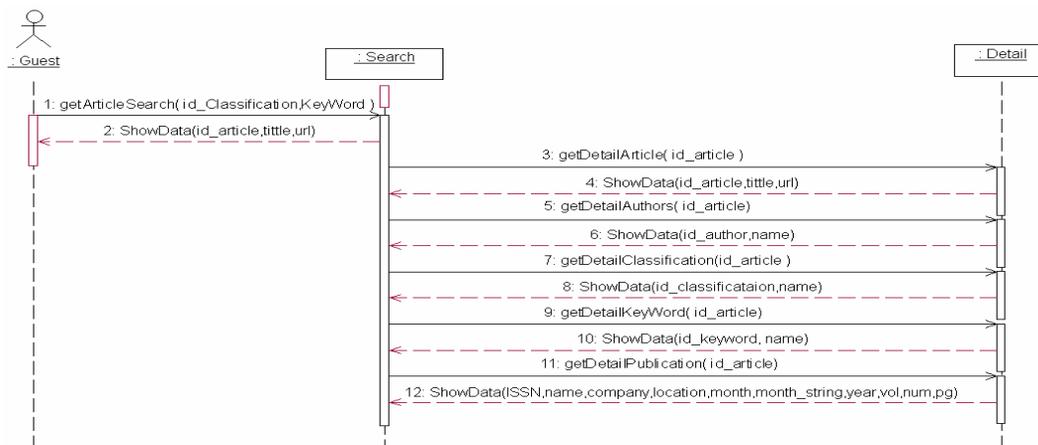


Gambar 3. Use case Portal Jurnal

Gambar 3 menunjukkan use case diagram aplikasi yang terdiri dari dua aktor (yaitu Guest dan Author) dan tujuh use case. Gambar 4 menunjukkan class diagram aplikasi, sedang Gambar 5 mengilustrasikan salah satu sequence diagram untuk use case pencarian.



Gambar 4. Class diagram



Gambar 5. Sequence diagram use case Search

## 4. Implementasi

### 4.1 Batasan Implementasi

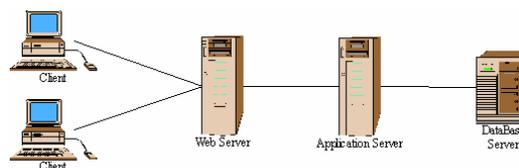
Pada aplikasi portal jurnal, yang pada akhirnya disebut portal jurnal InHamony, diasumsikan bahwa pengguna aplikasi baik itu sebagai pengunjung ataupun penulis artikel dapat melakukan akses ke sistem secara gratis, tidak ada penanganan pembayaran untuk download artikel, pengiriman artikel, dan sebagainya.

### 4.2 Implementasi Sistem

Implementasi Arsitektur Multi-tier pada Portal Jurnal membutuhkan 2 buah dua buah business object:

1. Business Object Portal Jurnal, untuk memproses operasi yang berkaitan dengan jurnal
2. Business Object Author Gateway, untuk memproses operasi yang berkaitan dengan aplikasi Pojok Pengarang.

Arsitektur sistem yang digunakan, mencakup penggunaan database server, application server dan web server diilustrasikan pada Gambar 6.

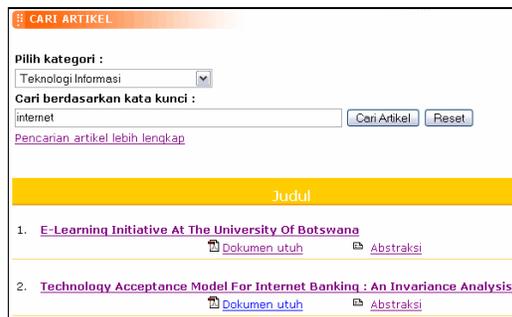


Gambar 6. Arsitektur sistem dimana semua business object diletakkan pada application server

Tampilan dari halaman awal portal jurnal ditunjukkan pada Gambar 7. Gambar 8 menunjukkan salah satu tampilan untuk pencarian artikel.

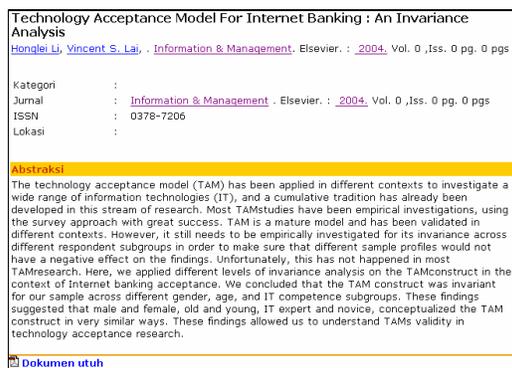


Gambar 7. Halaman awal Portal Jurnal



Gambar 8. Halaman pencarian artikel

Jika tampilan rinci artikel yang dipilih, maka akan ditampilkan informasi detail terkait artikel dan abstrak yang disertai dengan link ke dokumen utuh (lihat Gambar 9).



Gambar 9. Halaman penampilan artikel secara rinci

## 5. Analisis Kinerja

Dari sisi struktur *programming code*, ada kelebihan yang bisa dipetik dalam model three-tier dengan COM+, yaitu kesederhanaan baris program ketika memanggil data. Segmen program berikut ditujukan untuk pengambilan data pada model three-tier dengan COM+.

```
Set COM_Search = server.CreateObject("Jurnal.Search")
set rsClassification =
COM_List.getListClassification(rsClientInfo)

set rsArticles =
COM_Search.getArticleSearch
(rsClientInfo, Keyword, menuClassification)
```

```
set COM_Search = nothing
```

Bandingkan dengan struktur penulisan model *two-tier* berikut untuk digunakan untuk tujuan yang sama.

```
set cn = Server.CreateObject("ADODB.Connection")
```

```
openStr =
"Provider=MSDORA.1;Password=journal;User
ID=journal;Data Source=Harmony"
```

```
cn.open openStr, "", ""
```

```
sql = " SELECT DISTINCT a.title,
a.id_article, a.url " &
" FROM td_articles a, td_key_word b,
td_classification c " &
" WHERE ((a.id_article =
b.id_article(+) AND (a.id_article =
c.id_article(+))) " &
" AND ( lower(a.title) LIKE
lower('%" & Keyword1 & "%') " &
" OR lower(b.name) LIKE lower('%"
& Keyword2 & "%') ) " &
" AND c.id_classification LIKE '%" &
Classification & "%' "
```

```
set rsArticles =
Server.CreateObject("ADODB.Recordset")
```

```
rsArticles.open sql , cn, 3,3,1
```

Perbandingan jumlah koneksi ke database yang dibutuhkan dalam waktu 5 menit untuk menangani request user secara terus-menerus pada model 2-tier dan 3-tier dengan jumlah user yang berbeda dirangkum pada Tabel 1. pengujian dilakukan dengan bantuan Web Application Stress Tool.

Tabel 1. Perbandingan jumlah koneksi ke database pada model 2-tier dan 3-tier

Jumlah user	Jumlah koneksi pada model 2-tier	Jumlah koneksi pada model 3-tier	Penghematan (%)
100	354186	221113	37,57
200	163529	152192	6,93
300	129837	20365	84,31
400	111954	56246	49,76
500	111831	45274	59,52
600	125583	37519	70,12
700	141193	34694	75,43
800	136019	42963	68,41
900	125178	44109	64,76
1000	117065	47534	59,40
Rata-rata penghematan			57,62

Dari Tabel 1 di atas, terlihat bahwa penggunaan model 3-tier mengurangi *traffic* ke server rata-rata sebesar 57,62%.

## 6. Kesimpulan

Teknologi arsitektur *three-tier client/server* yang menyediakan kemampuan dalam hal perluasan sistem, kemudahan perawatan sistem, kemudahan dalam penggunaan kembali komponen, dan pendistribusian beban. *Traffic* ke server dengan model 3-tier jika dibandingkan dengan model 2-tier berkurang rata-rata sebesar 57,62%.

## Pustaka

- Ching, A. .Silva , P. Dan Wagner, A. *Performance Testing with the Web Application Stress Tool*. Microsoft Developer Network (MSDN), 2001.
- Computer Language Company. *The Computer Desktop Encyclopedia*. Computer Language Company, 2005
- Ciang, D. *Cara Mudah Pemrograman Database Delphi 7 Menggunakan Class Generator*. Jakarta: Elex Media Komputindo, 2004
- Meier, J.D. *I Can't Stress It Enough Load Test Your ASP Application*. Microsoft Developer Network (MSDN), 2001.
- Matthews, M. *Writing MTS and COM+ Components in Delphi*. Online: <http://consulting.dthomas.co.uk>, 2002
- Pattison, T. *COM+ Overview for Microsoft Visual Basic Programmers*. Microsoft Developer Network (MSDN), 2001.
- Wikimedia Foundation. *Wikipedia*. Wikimedia Foundation, 2005.