

PENERAPAN COMPONENT BASED SOFTWARE ENGINEERING DALAM PENGEMBANGAN WEBSITE SISTEM INFORMASI SEKOLAH (STUDI KASUS : SMA SANTO PAULUS PONTIANAK)

SANDY KOSASI¹, DAVID²

¹Program Studi Sistem Informasi, ²Program Studi Teknik Informatika

^{1,2}Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak

Jln. Merdeka No. 372 Pontianak, Kalimantan Barat

¹Sandykosasi@yahoo.co.id, ²David_Liau@yahoo.com

Abstract

The authors conducted a study on St. Paul's High School Pontianak due to lack of necessary information for example is seeing the value, the latest news about the school, the school agenda, and other general information about Saint Paul's High School Pontianak. The purpose of research conducted by the authors is to develop information systems that exist in high schools of St. Paul by applying Component-Based Software Engineering. This research will generate a website using COM technology (Component Object Model) which all database queries generated by using Visual C++ which will be build into a file with the extension DLL. DLL files generated will be called using COM technology in PHP. COM is a specification developed includes implementation of Microsoft Corporation, which provides a framework for the integration of components. This framework supports the interoperability and reusability of distributed objects that enable developers to build systems by performing assembly components from various vendors. Those components will communicate via COM. The conclusion that the authors can provide through this study is the use of COM is applied to the PHP can be applied also to other programming languages.

Keyword : *Component Based Software Engineering, Website, PHP, Visual C++, COM, DLL, Microsoft Corporation*

Abstrak

Penulis melakukan penelitian pada SMA Santo Paulus Pontianak dikarenakan kurangnya informasi yang diperlukan misalnya adalah melihat nilai, berita terkini tentang sekolah, agenda kegiatan sekolah, dan informasi umum lainnya mengenai SMA Santo Paulus Pontianak. Tujuan penelitian yang dilakukan oleh penulis adalah untuk mengembangkan sistem informasi sekolah yang ada pada SMA Santo Paulus dengan menerapkan *Component Based Software Engineering*. Penelitian ini akan menghasilkan website dengan menggunakan teknologi COM (*Component Object Model*) dimana semua *query* database dihasilkan dengan menggunakan Visual C++ yang nantinya akan di-build menjadi file dengan ekstensi DLL. File DLL yang dihasilkan akan dipanggil dengan menggunakan teknologi COM pada PHP. COM adalah spesifikasi mencakup implementasi yang dikembangkan *Microsoft Corporation*, yang menyediakan *framework* untuk integrasi komponen. *Framework* ini mendukung *interoperability* dan *reusability* objek-objek terdistribusi yang memungkinkan *developer* membangun sistem dengan melakukan *assembly* berbagai komponen dari berbagai *vendor*. Komponen-komponen tersebut akan berkomunikasi melalui COM. Kesimpulan yang dapat penulis berikan melalui hasil penelitian ini adalah penggunaan COM yang diterapkan pada PHP dapat diterapkan juga pada bahasa pemrograman lainnya.

Kata Kunci : *Component Based Software Engineering, Website, PHP, Visual C++, COM, DLL, Microsoft Corporation*

1. Pendahuluan

SMA Santo Paulus Pontianak merupakan salah satu sekolah swasta unggulan di kota Pontianak sehingga menjadikan sekolah ini banyak dijadikan pilihan oleh orang tua calon siswa saat ini karena sekolah ini cukup

maju. Pada tahun 2011 ini jumlah peserta didik di SMA Santo Paulus Pontianak memiliki jumlah total sebanyak 907 siswa, dan memiliki 42 tenaga pengajar. Akan tetapi sistem informasi di sekolah ini masih belum tertata dengan begitu baik, seperti pengolahan dan

penyimpanan data juga penyediaan informasi yang sering terkendala karena proses *retrieving data* yang memakan waktu.

Website SMA Santo Paulus Pontianak juga tidak memiliki fitur-fitur yang menarik seperti tidak adanya fitur yang memungkinkan siswa dapat melihat nilai secara *online*, melihat pengumuman secara online sehingga website pada SMA Santo Paulus Pontianak ini tidak *up to date* dan tidak banyak menyediakan informasi yang dibutuhkan oleh para siswa, guru, dan masyarakat.

Arsitektur jaringan pada SMA Santo Paulus Pontianak juga masih menggunakan arsitektur *One-Tier* dimana semua kode aplikasi, data, pemrosesan, dan semua komponen sistem dilakukan pada satu komputer yang digunakan oleh bagian tata usaha. Cukup banyak kelemahan dalam arsitektur *One-Tier* ini, salah satunya adalah jaringan yang berskala kecil, tingkat keamanan yang rendah, dan jika terjadi perubahan salah satu komponen maka dapat menyebabkan semua bagian berubah. Informasi yang ada pada bagian tata usaha tidak saling terintegrasi ke bagian kepala sekolah sehingga jika kepala sekolah memerlukan informasi tentang siswa ataupun guru maka kepala sekolah harus pergi ke bagian tata usaha untuk meminta informasi yang diinginkan.

Untuk mengatasi masalah yang dihadapi SMA Santo Paulus Pontianak tersebut maka penulis akan mencoba untuk mengembangkan sebuah website yang dapat mendukung jalannya sistem informasi pada sekolah dengan menerapkan *Component Based Software Engineering (CBSE)*. *Component Based Software Engineering (CBSE)* adalah pendekatan rekayasa perangkat lunak dengan penggabungan komponen. Yang ditawarkan oleh *CBSE* adalah dalam membangun sebuah aplikasi, penulis tidak perlu membangun semua kebutuhan aplikasi tersebut dari awal, melainkan hanya tinggal menggabungkan bagian-bagian yang tersedia menjadi satu aplikasi yang diinginkan.

Adapun arsitektur jaringan yang akan penulis gunakan untuk jaringan lokal pada SMA Santo Paulus Pontianak adalah arsitektur *Three-Tier*. Arsitektur *Three-Tier* adalah model yang membagi fungsionalitas ke dalam lapisan-lapisan, aplikasi-aplikasi mendapatkan skalabilitas, keterbaharuan, dan keamanan.

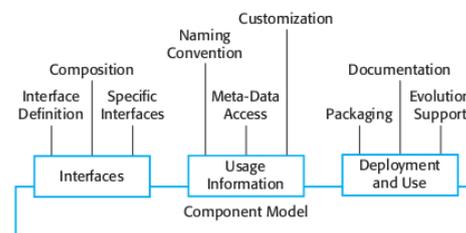
2. Tinjauan Pustaka

2.1 Component Based Software Engineering

Component Based Software Engineering merupakan salah satu paradigma pemrograman baru yang bertujuan untuk menyempurnakan teknik-teknik pengembangan perangkat lunak yang telah ada. *CBSE*

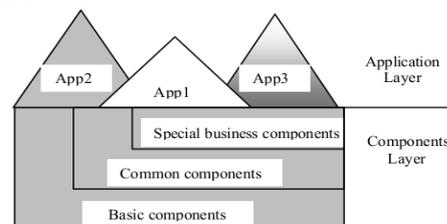
menawarkan pendekatan perakitan dan penggunaan kembali komponen-komponen perangkat lunak untuk menggantikan teknik pengembangan dari awal seperti yang dipakai selama ini.

Secara konseptual, komponen adalah sebuah bagian dari sistem yang dapat tergantung, selama masih merealisasikan interface-interface yang sama. Interface pada komponen hanya menjelaskan layanan yang disediakan oleh komponen tersebut, bukan implementasinya. Implementasi fisik sebuah komponen tersembunyi bagi para penggunanya, karena komponen hanya dapat diakses melalui interfacenya, karenanya tingkah laku yang direpresentasikan oleh interface-interface sebuah komponen harus didefinisikan dengan jelas.

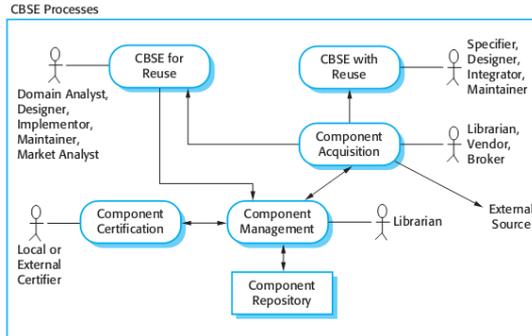


Gambar 1. Basic Element of Component Model (Sommerville, 2011:459)

Untuk memastikan bahwa sistem dengan *Component Based Software Engineering* dapat berjalan dengan baik dan efektif, maka diperlukan pertimbangan arsitektur sistem sebagai faktor yang paling penting. Sistem arsitektur *Component Based Software Engineering* harus menjadi arsitektur berlapis dan modular. Arsitektur ini dapat dilihat pada Gambar berikut. Lapisan aplikasi atas adalah sistem aplikasi yang mendukung bisnis. Lapisan kedua terdiri dari komponen yang terlibat dalam hanya bisnis tertentu atau domain aplikasi, termasuk komponen dapat digunakan di lebih dari satu aplikasi. Lapisan ketiga adalah *cross-bisnis* komponen *middleware* terdiri dari perangkat lunak umum dan interface untuk entitas lainnya. Lapisan terendah dari komponen perangkat lunak sistem mencakup komponen dasar yang antarmuka dengan sistem operasi yang mendasari dan hardware.



Gambar 2. System architecture of component-based software systems(Cai, et.al. 2000)

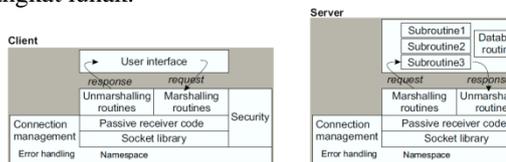


Gambar 3. CBSE Process (Sommerville, 2011:462)

2.2 COM (Component Object Model)

Teknologi *client-server* muncul untuk menjawab semakin besarnya jumlah masalah dan data yang harus diselesaikan. *Client-server* mengoptimalkan jaringan dan *resource* komputer yang ada. Teknologi *client-server* membagi layer menjadi tiga yaitu; *Front-End Component*, *Back-End Component* dan *Database Component*. Masing-masing komponen memiliki tugas dan hak yang berbeda-beda. Pembagian komponen ini juga dapat menambah keamanan terhadap data *user*, karena *user* tidak dapat langsung berhubungan langsung ke komponen database.

Secara teknis, COM adalah sebuah objek berbasis model pemrograman yang dirancang untuk mempromosikan interoperabilitas perangkat lunak. Itu berarti COM memungkinkan dua atau lebih komponen untuk bekerja sama dengan satu sama lain dengan mudah, bahkan jika komponen yang dihasilkan oleh vendor yang berbeda pada waktu yang berbeda dan dalam bahasa pemrograman yang berbeda, atau jika program berjalan pada mesin yang berbeda dengan sistem operasi yang berbeda pula. Untuk mendukung fitur interoperabilitas, COM mendefinisikan dan menerapkan mekanisme yang memungkinkan aplikasi untuk berkomunikasi satu sama lain sebagai objek perangkat lunak.



Gambar 4. Client/server application architecture (Sherlock & Cronin, 2000:4)

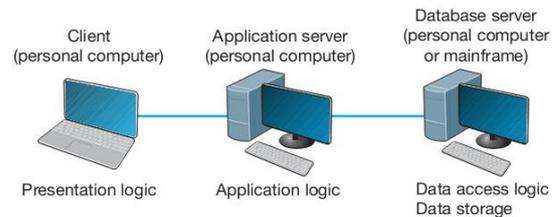
2.3 Arsitektur 3-tier (Three-tier)

Istilah arsitektur mengacu pada desain sebuah aplikasi, atau dimana komponen yang membentuk suatu sistem ditempatkan dan bagaimana mereka berkomunikasi. *Client* merupakan sembarang sistem atau proses yang

melakukan suatu permintaan data atau layanan ke server sedangkan server ialah sistem atau proses yang menyediakan data atau layanan yang diminta oleh *client*.

Client-Server adalah pembagian kerja antara server dan *client* yg mengakses server dalam suatu jaringan. Jadi arsitektur *client-server* adalah desain sebuah aplikasi terdiri dari *client* dan server yang saling berkomunikasi ketika mengakses server dalam suatu jaringan.

Sebuah arsitektur *Three-tier* menggunakan tiga set komputer, seperti yang ditunjukkan pada Gambar 4. Dalam kasus ini, *software* pada komputer *client* bertanggung jawab atas logika presentasi, sebuah *Application server* bertanggung jawab atas logika aplikasi, dan *Database server* yang terpisah bertanggung jawab atas logika akses data dan penyimpanan data.



Gambar 5. Three-tier client-server architecture (Fitzgerald, et.al., 2011:45)

3. Metodologi Penelitian

Penelitian ini bersifat penelitian terapan (*applied research*), dilakukan dengan tujuan menerapkan kemampuan suatu teori yang diterapkan dalam memecahkan masalah-masalah praktis. Penelitian yang dimaksud dalam penelitian ini adalah penelitian dan pengembangan (*research and development*), dimana penelitian ini bertujuan untuk mengembangkan produk atau suatu prosedur dan cara kerja baru. Agar produk atau prosedur dan cara kerja baru tersebut mempunyai kualitas yang lebih tinggi sehingga dapat memenuhi kebutuhan yang diharapkan. Di samping itu, penelitian ini hanya difokuskan kepada pengetahuan teoritis dan praktis dalam bidang tertentu, bukan pengetahuan yang bersifat universal.

Metode pengumpulan data bersumber dari jenis data primer dan data sekunder. Data primer dikumpulkan menggunakan teknik wawancara kepada pihak SMA Santo Paulus Pontianak. Penulis melakukan wawancara untuk mendapatkan informasi dengan mengajukan beberapa pertanyaan secara langsung. Penulis menggunakan teknik wawancara semi terstruktur untuk mengetahui

dan meneliti gambaran umum sekolah dan prosedur-prosedur yang berjalan pada SMA Santo Paulus Pontianak. Sedangkan data sekunder merupakan data yang sudah tersedia sehingga tinggal dicari dan dikumpulkan yang berhubungan dengan isi informasi yang akan ditampilkan dengan menggunakan teknik observasi dan studi dokumentasi. Pada metode pengumpulan observasi penulis melakukan pengamatan dengan meninjau secara langsung kegiatan-kegiatan yang sedang dilakukan.

4. Hasil Penelitian

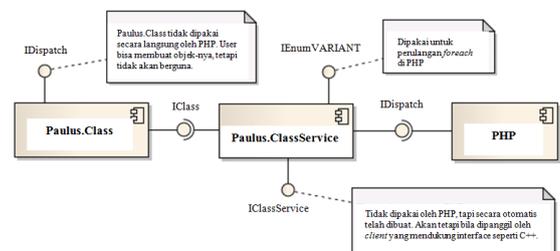
Dari hasil penelitian yang penulis lakukan, maka dihasilkan dua buah website yang berbeda dimana satu website menjadi aplikasi berbasis web yang dijalankan di dalam area lokal SMA Santo Paulus Pontianak dan satu website lagi merupakan website SMA Santo Paulus Pontianak yang dihosting dan dapat diakses oleh masyarakat luas. Aplikasi Berbasis Web menggunakan teknologi COM (*Component Object Model*) dengan arsitektur *Three-tier*. DLL yang dihasilkan menggunakan Visual C++ dari Visual Studio 2010. Terdapat tiga buah aplikasi pendukung yang digunakan oleh penulis untuk menghubungkan aplikasi C++ ke database MySQL. Aplikasi yang digunakan adalah *mysql-connector-c++-1.1.0*, *cmake-2.8.8*, dan *boost_1_50_0*.

Penulis menggunakan *mysql-connector-c++-1.1.0* untuk menghasilkan sebuah *library* dari *source code*. Kemudian menggunakan *cmake-2.8.8* untuk mengkompilasi proyek dan aktifitas serupa sehingga penulis tidak perlu beralih ke OS Linux untuk menggunakan *make*. Sementara itu penulis menggunakan *boost_1_50_0* karena Connector/C++ yang dihasilkan menggunakan Boost, Boost sendiri adalah kumpulan *library* untuk C++ yang sangat lengkap mulai dari fitur *foreach* hingga dukungan akses jaringan yang *portable*. Adapun hasil dari konversi file DLL dengan Visual C++ hingga cara pemanggilan COM (*Component Object Model*) pada PHP dapat dilihat pada lampiran.

Sementara itu website yang dihosting dihasilkan dengan menggunakan *framework MVC (Model View Controller)*. *Framework* yang penulis gunakan adalah *framework CodeIgniter*. Penulis memilih menggunakan *framework MVC* karena website yang dihasilkan adalah berbasis OOP (*Object Oriented Programming*) dan *framework MVC* adalah salah satu *framework* yang mendukung OOP. Pada *framework MVC*, *source code* program telah terpisah menjadi beberapa bagian yaitu *Model*, *View*, dan *Controller*. Konsep yang ada pada model MVC sendiri sudah menyerupai konsep arsitektur *Three-tier*

dimana bagian *View* yang *handle* tampilan/*user interface* ataupun sebagai penghubung antara bagian *Controller* dan *Model*, bagian *Controller* yang *handle* semua fungsi yang akan dijalankan pada *View*, sementara itu bagian *Model* yang *handle* semua *query* database yang nantinya akan dipanggil ketika *View* melakukan aksi terhadap *Controller* dan kemudian *Controller* akan melakukan *request* terhadap *Model*.

Berdasarkan referensi dari www.php.net/manual/en/book.com.php bahwa PHP mendukung *Component Object Model (COM)* dan *.NET*. Keterkaitan ataupun hubungan *DLL* dan *COM* yaitu *DLL (Dynamic-link Library)* adalah implementasi *shared-library* di platform Windows. Programmer membuat fungsi yang dapat dipanggil ulang di program berbeda. *COM (Component Object Model)* adalah sebuah metodologi yang mengatur bagaimana menerapkan komponen program yang dapat dipakai ulang di program berbeda. *COM* tidak mengatur struktur bahasa yang dipakai (*MSDN*). Implementasi *COM* umumnya dalam bentuk file *DLL*. Tidak semua *DLL* dibuat berdasarkan aturan *COM*, atau dengan kata lain tidak semua *DLL* adalah *COM*. Apabila *DLL* dibuat dengan menggunakan teknologi *COM*, maka *DLL* tersebut dapat diakses di *PHP*. Jika *DLL* tersebut adalah *DLL* sederhana, maka *DLL* tersebut tidak dapat diakses di *PHP*.



Gambar 6. Rancangan COM

Pada gambar 6, terlihat bahwa DLL yang penulis hasilkan mengandung dua *class*, masing-masing dengan ProgID Paulus.Class dan Paulus.ClassService. PHP akan berinteraksi dengan Paulus.ClassService melalui *interface* IDispatch. Class Paulus.ClassService akan memakai *class* lain yaitu Paulus.Class. Walaupun *class* ini menyediakan IDispatch, *user* tidak seharusnya membuat *class* ini langsung dari PHP, karena tujuan utama *class* ini adalah dipakai oleh Paulus.ClassService. Paulus.ClassService menyediakan *provided interface* bernama IClassService yang tidak pernah dipakai. Pada komunikasi yang ideal, penulis akan memakai *interface* ini yang sudah disediakan oleh Visual

C++ ATL karena program akan membuatkan dua interface ini secara otomatis bagi kita.

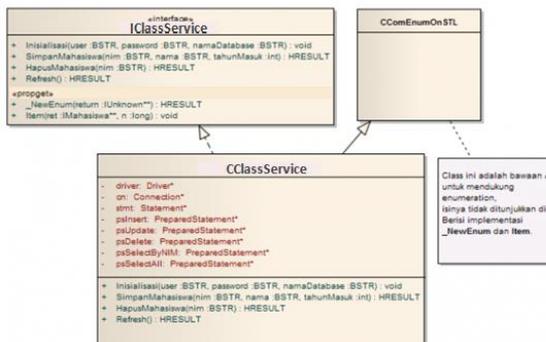
Kembali ke diagram di gambar 6, class *Paulus.ClassService* akan memakai class lain yaitu *Paulus.Class*. Walaupun class ini menyediakan *IDispatch*, pengguna tidak seharusnya membuat class ini langsung dari *PHP*, karena tujuan utama class ini adalah dipakai oleh *Paulus.ClassService*. Hal ini berkaitan dengan bagaimana mengembalikan isi tabel ke klien *PHP*. Saat *PHP* memanggil *getAllClass* cukup mengembalikan class *Paulus.Class*.

Berikut ini contoh potongan script php yang memanggil *COM*.

```
<?php
    if (isset($_GET['nama_kelas'])
    &&isset($_GET['tahun_ajaran']))
    {
        $com = new
    COM("Paulus.Kelas");
        $com->DBUser = "root";
        $com->DBPassword = "";
        $com->DBNama
    = "db_adminpaulus";
        $com->Inisialisasi();

        $nama_kelas =
    $_GET['nama_kelas'];
        $tahun_ajaran =
    $_GET['tahun_ajaran'];
        $com-
    >TambahKelas($nama_kelas, $tahun_ajaran);
    }
?>
```

Kode program PHP di atas merupakan kode program yang menangani Insert data ke database dimana query INSERT didapat melalui pemanggilan *COM Paulus.Kelas*. *Paulus.Kelas* sendiri merupakan *ProgID* yang ditentukan saat pembuatan *ATL Simple Object* Kelas. Saat *ProgID Paulus.Kelas* dipanggil, maka *AdminPaulus.dll* akan digunakan dan akan menjalankan method *TambahKelas*.



Gambar 7. Rancangan Class Diagram ClassService

CClassService adalah class yang mengimplementasikan *IClassService*. Hierarki ini secara otomatis sudah dibuat oleh Visual C++ pada saat melakukan wizard new *ATL SimpleObject*. Method Inisialisasi() akan dipakai untuk melakukan koneksi database. Pengguna harus menyertakan user database, password, serta nama database yang dipakai saat memanggil method ini. Bila method lain dipanggil sebelum method ini, maka method lain akan mengembalikan nilai kesalahan. Pada perancangan yang lebih baik, kamu bisa menyediakan nilai default untuk user, password, dan nama db. Lalu setiap kali method lain dipanggil dan ternyata koneksi belum dibuat, maka method *Inisialisasi()* akan dikerjakan dengan nilai default tersebut.

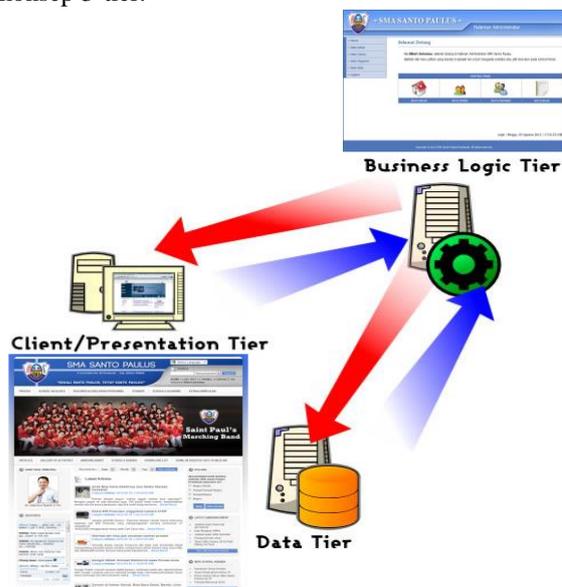
Selain melakukan koneksi database, method Inisialisasi juga akan menyiapkan *PreparedStatement* yang dibutuhkan. *PreparedStatement* adalah class dari *Connector/ODBC* untuk men-cache statement SQL sehingga pemanggilan query yang sama akan lebih cepat nantinya. Selain itu, *PreparedStatement* akan mencegah terjadinya SQL Injection. Pada diagram di atas, saya membuat beberapa *PreparedStatement* seperti *psInsert*, *psUpdate*, *psDelete*, *psSelectByNIS*, dan *psSelectAll*.

Setelah melakukan inisialisasi *PreparedStatement*, method *Inisialisasi()* juga akan melakukan query untuk mendapatkan seluruh isi tabel Class melalui method *Refresh()* guna dipakai oleh *_NewEnum()* dan *Item()* nantinya. Pada kasus yang lebih realistis dimana isi tabel Class sangat banyak sekali, user seharusnya hanya perlu men-select hanya bagian tertentu saja pada satu waktu tertentu, tidak semuanya sekaligus.

Setelah select seluruh data dilakukan pertama kali saat method *Inisialisasi()* dipanggil, mungkin saja user akan melakukan perubahan, seperti setelah pemanggilan *HapusClass()* atau *SimpanClass()*. Hal ini menyebabkan hasil query pertama kali menjadi tidak valid lagi. Oleh sebab itu, user wajib memanggil method *Refresh()* untuk memperbaharui query. Pada design yang lebih realistis, user perlu melakukan caching dan melakukan refresh secara otomatis sehingga user tidak perlu repot.

Method *SimpanClass()* akan dipakai untuk insert dan update. Bila NIS Class belum ada di tabel, maka program harus menambah (insert) record baru. Tetapi bila NIS sudah ada di tabel, maka program tinggal mengubah (update) record yang sudah ada.

Berikut ini hasil penelitian yang penulis lakukan, dihasilkan dua buah website yang berbeda dimana satu website menjadi aplikasi berbasis web yang dijalankan di dalam area lokal SMA Santo Paulus Pontianak dan satu website lagi merupakan website SMA Santo Paulus Pontianak yang dihosting dan dapat diakses oleh masyarakat luas. Gambar 8 berikut menunjukkan hasil website dengan konsep 3-tier.



Gambar 8. Hasil Penelitian Website dengan teknologi 3-tier

5. Kesimpulan

Kesimpulan yang dapat penulis berikan melalui hasil penelitian ini adalah penggunaan COM yang diterapkan pada PHP dapat diterapkan juga pada bahasa pemrograman lainnya karena COM yang dihasilkan dengan struktur biner yang sesuai dengan spesifikasi *Microsoft* dapat dibangun dengan bahasa pemrograman yang berbeda dan akan dapat saling berinteraksi. Namun untuk dapat membangun arsitektur COM secara online dibutuhkan sebuah *server dedicated*. Website yang dikembangkan dengan menggunakan teknologi *MVC (Model View Controller)* dimana tiap fungsi dan bagiannya saling terpisah sehingga menjadikan website murni menggunakan *OOP (Object Oriented Programming)* dan lebih mudah dalam melakukan maintenance. Website sistem informasi sekolah menggunakan sistem modul dimana pemanggilan fungsi-fungsinya menggunakan teknologi *COM (Component Object Model)* berupa penggunaan file *DLL*. Fungsi *COM* yang dihasilkan tidak hanya dapat digunakan pada PHP, tetapi juga dapat digunakan pada bahasa pemrograman lainnya. Pada bagian aplikasi berbasis web

yang menggunakan teknologi *COM*, penulis hanya menghasilkan sebuah sistem sederhana karena keterbatasan waktu. Namun sistem ini dapat lebih dikembangkan lagi sehingga tidak menutup kemungkinan dapat digunakan oleh semua sekolah yang ada. Bagian aplikasi berbasis web yang menggunakan teknologi *COM*, penulis hanya menggunakan fungsi query *INSERT* dan *UPDATE* saja karena keterbatasan kemampuan. Sementara itu fungsi query *SELECT* dan *DELETE* masih menggunakan kode program dari PHP.

Daftar Pustaka

- [1]. Cai, Xia., Lyu, Michael R., Wong, Kam-Fai., Ko, Roy., "Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes", *Proceeding APSEC '00 Proceedings of the Seventh Asia-Pacific Software Engineering Conference*, IEEE Computer Society Washington, DC, USA ©2000, pp. 372
- [2]. Sommerville, Ian., "Software Engineering", 9th edition, Addison-Wesley, 2011
- [3]. W. Kozaczynski, G. Booch, "Component-Based Software Engineering," *IEEE Software*, Volume: 155, Sept.-Oct. 1998, pp. 34-36
- [4]. Pressman, Roger., "Software Engineering: A Practitioner's Approach", 7th edition, McGraw-Hill, 2009
- [5]. Sherlock, Terence., Cronin, Gene., "COM Beyond Microsoft: Designing and Implementing COM Servers on Compaq Platforms (HP Technologies)", Digital Press, 2000
- [6]. Oestereich, Bernd, "Developing Software with UML : Object Oriented Analysis and Design in Practice (2nd Edition)", Addison-Wesley, London, 2002
- [7]. Wall, David, "Multi-Tier Application Programming With PHP, Practical Guide for Architects and Programmers", Elsevier, Inc, United States of America, 2004
- [8]. Gupta, Ratneshwer., Tripathi, Anil., "Component Based Software Engineering Dependability and Software Process Issues", LAP LAMBERT Academic Publishing, 2011