

## ANALISA DAN RANCANGAN SISTEM INFORMASI ANTRIAN MENGUNAKAN *BISKITZ CMS*

Rizky Tahara Shita <sup>1)</sup>, Gandung Triyono <sup>2)</sup>

Fakultas Teknologi Informasi, Universitas Budi Luhur  
Jl.Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260, Indonesia  
E-mail : <sup>1)</sup> [rizky.tahara@gmail.com](mailto:rizky.tahara@gmail.com), <sup>2)</sup> [gandung.triyono@budiluhur.ac.id](mailto:gandung.triyono@budiluhur.ac.id)

**Abstract** -- *Queuing is a less pleasant for many people, especially if the queue is in progress does not irregular. To overcome this things, then it needs a system that helps to be organized and more fun. At some companies, this is already implemented by conducting a computerized queuing system. Thus, the person will get a queue number and wait to be called, to be served by the customer service. Biskitz CMS is a content management system that can be embedded for the purposes of systems that are customizable and this time biskitz CMS will be implemented to be able to perform the integration with the queue system that will be applied to the Directorate of Finance Bureau of the University of Budi Luhur.*

**Keywords:** *content management system, queueing systems, biskitz*

### 1. PENDAHULUAN

Antrian sering menjadi sebuah kendala yang dihadapi pada bagian *customer service* dalam memberikan pelayanan kepada *customer*. Kendala yang dihadapi dapat kurang teraturnya pelayanan yang diberikan, sehingga akan berdampak pada kurang nyamannya pelayanan yang diterima oleh *customer*.

Dengan membuat sebuah sistem untuk membantu agar antrian lebih rapih dan teratur, sehingga pelayanan yang diberikan juga dapat memuaskan *customer*; maka kenyamanan pastinya juga akan dirasakan oleh *customer* tersebut.

Sistem antrian yang dikembangkan merupakan sebuah modul yang akan di-embed pada *biskitz CMS*, sehingga untuk keperluan akademik lainnya dapat saling

Terintegrasi dengan mudah dan dapat lebih cepat dalam melakukan perbaikan dan perubahan.

Adapun sistem antrian yang dikembangkan juga akan dapat menampung data pelayanan yang dilakukan, sehingga dapat menunjang bagi pimpinan untuk dapat memutuskan standarisasi lama pelayanan yang dilakukan terhadap satu *customer*. Juga didukung dengan adanya laporan – laporan bagi pihak pimpinan yang berguna dalam memutuskan suatu hal (*decision*) yang berhubungan dengan pelayanan kepada *customer* dan sistem antrian yang ada.

### 2. LANDASAN TEORI

#### 2.1. Antrian

Pelayanan kepada *customer* dapat diberikan setelah *customer* tersebut mengantri. Adapun metode yang dapat digunakan untuk membuat sebuah sistem antrian antara lain adalah:

1. *First In First Out* (FIFO)  
Pada metode ini, *customer* akan dilayani sesuai dengan urutan kedatangan. Yang lebih dahulu datang yang akan lebih dahulu dilayani.
2. *Last In Last Out* (LIFO)  
Sedangkan pada metode LIFO, *customer* yang paling terakhir datanglah yang akan dilayani terlebih dahulu.
3. *Processor Sharing* (PS)  
Untuk metode *processor sharing*, maka *customer* akan dilayani secara bersamaan.

Dari metode antrian tersebut, tidak semua metode dapat diterapkan pada sistem

antrian untuk melayani *customer*. Sistem antrian untuk memberikan pelayanan pelanggan akan sangat tepat menggunakan metode FIFO, sedangkan metode antrian yang lainnya dapat digunakan pada model antrian yang berbeda (misalnya: untuk melayani penyambungan nomor telepon, dapat digunakan metode *processor sharing*).

Metode FIFO sangat tepat digunakan pada sistem antrian pelayanan pelanggan; karena secara logika, *customer* yang lebih dahulu datang yang akan mendapatkan pelayanan lebih awal dan *customer* yang datang berikutnya akan kemudian dilayani oleh *customer service*.

## 2.2. DBMS

*Database Management System* atau yang disingkat dengan DBMS, merupakan sebuah sistem yang dapat menyimpan data agar tidak bersifat *volatile* dengan analisa – analisa yang telah dilakukan baik menggunakan *Entity Relation Diagram* maupun menggunakan *Class Diagram* dan berisi koleksi data operasional yang digunakan oleh sistem – sistem aplikasi suatu perusahaan / organisasi, sehingga dapat mendefinisikan, menciptakan, menyimpan serta menyediakan akses ke *database* dan *repository*-nya.

*Repository* yang terdapat dalam DBMS juga dikenal dengan istilah *Data Dictionary* yang berguna untuk menyimpan semua *metadata* untuk suatu *database* termasuk informasi tentang hubungan antar file – file atau tabel – tabel dalam *database* tersebut.

Sedangkan *metadata* yang terdapat dalam *repository* merupakan informasi tentang data yang disimpan dalam *database*. Informasi ini berupa deskripsi atribut – atribut dari suatu entitas (nama atribut, tipe data, besarnya nilai atribut, formatnya, serta atribut lainnya) dan juga termasuk didalamnya informasi tentang hubungan antar entitas.

## 2.3. UML

UML yang merupakan kependekan dari *Unified Modeling Language* adalah bahasa

yang digunakan untuk mem-visualisasikan, mendefinisikan, membangun dan membuat dokumen dari arsitektur perangkat lunak. UML dapat digunakan pada semua proses melalui metodologi pengembangan perangkat lunak dan melakukan implementasinya pada teknologi yang berbeda.

Dengan UML, maka dapat digunakan untuk menggambarkan proses bisnis dengan *actor* dan *use case*, menggambarkan interaksi dengan *interaction diagram*, menggambarkan struktur statis dari sistem yang dibangun dengan menggunakan *class diagram* dan beberapa diagram lainnya yang dapat membantu menggambarkan pengembangan sistem secara *object-oriented*.

*Object-oriented* dapat dengan mudah dipahami dengan bantuan UML, sehingga pada tim *development* dapat saling mengerti dengan mudah dan cepat; dan hal ini juga berdampak pada proses analisa yang dilakukan dan proses pengerjaan pengembang perangkat lunak tersebut.

## 2.4. Content Management System

*Content Management System* atau yang biasa disingkat dengan CMS merupakan sebuah sistem yang dapat membantu pengguna dalam mengembangkan sistem yang dibutuhkannya. Dengan adanya CMS, maka pengguna dapat memilih CMS yang sesuai dengan kebutuhannya. Hal ini tidak terelakan karena dengan adanya CMS, maka pengguna yang minim pengetahuan tentang pembuatan sistem dapat dengan mudah mendapatkan sebuah sistem yang sesuai dengan proses bisnis yang dijalani.

Banyak sekali jenis CMS yang dapat digolongkan menjadi beberapa kategori, antara lain:

1. CMS Portal
2. CMS Blog
3. CMS e-Commerce
4. CMS Forum
5. CMS e-Learning
6. CMS Image Galleries
7. CMS Wiki

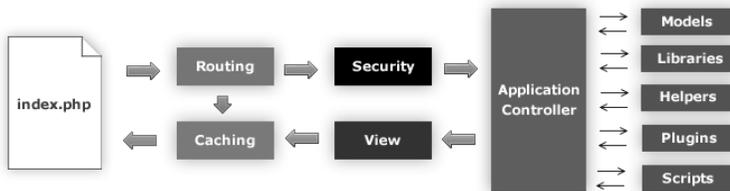
Dalam kategori CMS tersebut, terdapat beberapa nama CMS yang sudah populer dikenal dan digunakan; antara lain adalah Wordpress, Joomla, phpBB, Moodle dan masih banyak lagi untuk tiap kategori. Dengan adanya kategori CMS ini, maka semakin memudahkan bagi para pengguna dalam memilih CMS yang sesuai dengan kebutuhan.

Meski banyak CMS yang bersifat *open-source*, ada juga CMS yang komersil (berbayar) dan hal ini tidak menutup kemungkinan bagi para pengguna yang memerlukan CMS yang spesifik atau butuh dikustomisasi lebih lanjut.

aplikasi dengan menggunakan *framework Codeigniter* dapat dilihat pada gambar 1.

**2.6. biskitZ CMS**

Sebuah CMS yang dikembangkan oleh Direktorat Biro Teknologi Informasi Universitas Budi Luhur adalah *biskitZ* CMS. Dimana pada CMS ini merupakan CMS spesifik untuk kebutuhan tertentu, sehingga kustomisasi dapat dilakukan dengan lebih flexibel agar dapat disesuaikan dengan keinginan pengguna. Meskipun difokuskan pada CMS yang spesifik, *biskitZ* tidak menutup kemungkinan untuk dapat diintegrasikan



**Gambar 1: Codeigniter Application Flow Chart**

**2.5. Codeigniter PHP Framework**

*Framework* merupakan salah satu pendukung dalam pengembangan CMS, terutama jika CMS atau aplikasi yang akan dibuat berbasiskan pada *object oriented*. Dengan adanya *framework* yang tersedia, maka pengembangan aplikasi yang dibuat menjadi lebih mudah, cepat dan lebih *object oriented*. Sehingga analisa – analisa yang dilakukan secara *object oriented* dapat dengan mudah dan cepat diterapkan untuk dikembangkan.

Jenis – jenis dari *framework* sangat banyak, baik yang *open-source* maupun yang *non open-source*. Salah satunya adalah *Codeigniter PHP Framework* yang merupakan *framework PHP* yang bersifat *open-source*. *Codeigniter PHP Framework* mengusung konsep *object oriented* yang sangat baik dengan memisahkannya menggunakan konsep MVC (*Model, View* dan *Controller*). Konsep MVC merupakan konsep yang digunakan oleh banyak *framework* dalam melakukan implementasi dari konsep *object oriented*. Alur kerja

dengan modul – modul generik lainnya; seperti: blog, e-commerce maupun forum.

Dengan *biskitZ* CMS, maka pihak pengembang dapat lebih fokus pada pembuatan modul dari proses bisnis yang ingin dikustomisasi. Hal ini dapat dilakukan karena sifat dari CMS tersebut yang sudah dibuat agar modular dan dapat dengan mudah digabung dengan modul – modul yang dibutuhkan agar dapat memenuhi kebutuhan pengguna.

**2.7. Prototype Development Methodology**

Dalam mengembangkan *prototype*, metodologi yang dimanfaatkan adalah *Prototype Development Methodology*. Metodologi ini dipilih karena kelebihan yang diberikannya dalam menghasilkan *prototype* yang cepat, sehingga waktu yang dibutuhkan tidak lama dan pengguna dapat langsung berinteraksi untuk menggunakan *prototype* tersebut. Dan dengan metodologi ini, maka *prototype* dapat langsung digunakan oleh pengguna dan pengguna dapat langsung memberikan respon terhadap kesesuaian *prototype*.

### 3. METODOLOGI PENELITIAN

Pada penelitian ini, pengumpulan data berasal dari data primer dan data sekunder; dimana pada data primer dilakukan dengan wawancara, observasi lapangan, uji coba serta mendapatkan *feedback*. Sedangkan data sekunder didapat dari studi literatur, tulisan ilmiah tentang sistem antrian dan analisisnya serta pengembangan CMS.

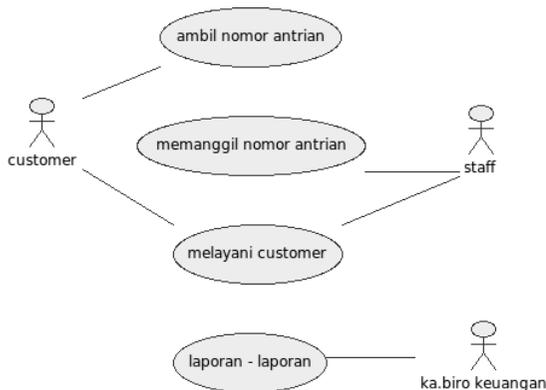
#### 3.1. Pengumpulan Data

Pengumpulan data didapatkan dengan melakukan wawancara terhadap pihak *executive* dari Direktorat Biro Keuangan Universitas Budi Luhur dan pihak Direktorat Biro Teknologi Informasi serta mendapatkan *feedback* dari mereka.

#### 3.2. Observasi Lapangan

Melakukan pengamatan pada lapangan dilakukan untuk menghimpun pengumpulan data, sehingga hal ini dapat membantu melihat hal yang terjadi sebenarnya di lapangan.

#### 3.3. Studi Literatur



**Gambar 2: Use Case Diagram Sistem Antrian**

Mempelajari dari berbagai referensi yang terkait dengan sistem antrian dan analisisnya serta pengembangan CMS menggunakan *Codeigniter PHP Framework*.

### 4. RANCANG BANGUN DAN UJI MODEL

#### 4.1. Analisa Kebutuhan

Untuk mengetahui kebutuhan pembuatan sistem antrian, maka dilakukan analisa kebutuhan dengan mengumpulkan data. Berikut ini adalah hasil analisa kebutuhannya:

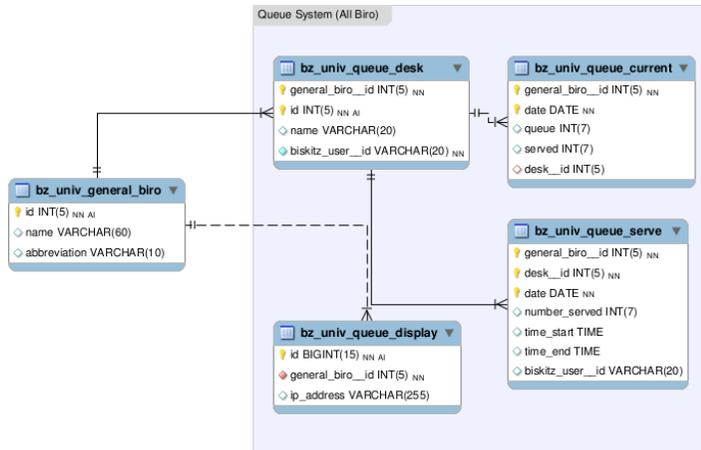
1. Dibutuhkan sebuah sistem untuk dapat melakukan kontrol terhadap antrian yang terjadi pada Direktorat Biro Keuangan Universitas Budi Luhur agar menjadi lebih teratur, rapih dan sistematis.
2. Dibutuhkan sebuah sistem yang dapat memberikan laporan – laporan terhadap sistem antrian yang sudah berjalan agar dapat digunakan oleh pihak *executive* Direktorat Biro Keuangan Universitas Budi Luhur untuk dapat mengambil keputusan.

#### 4.2. Pengembangan Implementasi Sistem Antrian

Dalam mengembangkan sistem antrian, diperlukan analisa yang cukup baik; tidak saja dari sisi analisa basis data maupun dari sisi pengembangan modul yang akan diintegrasikan pada *biskitZ CMS*. Analisa basis data diperlukan agar penyimpanan data yang akan dilakukan oleh CMS dapat disimpan dengan baik dan dapat memberikan laporan yang sesuai dengan kebutuhan pihak *executive* Direktorat Biro Keuangan Universitas Budi Luhur.

Pengembangan yang dilakukan dicanangkan untuk dapat menangani banyak biro yang ada pada Universitas Budi Luhur. Oleh karena itu, pada saat melakukan analisa basis data, diperlukan entitas untuk menampung biro – biro yang terdapat pada Universitas Budi Luhur.

Proses sistem antrian yang dikembangkan dapat dilihat melalui *use case diagram* berikut ini: *Customer* datang untuk mengambil nomor antrian dan menunggu pada tempat yang disediakan sampai salah satu staf (*desk*) memanggil nomor antriannya menggunakan CMS yang



**Gambar 3: Logical Record Structure Sistem Antrian**

sudah disediakan dan secara otomatis akan melakukan *trigger* pada sistem untuk menambahkan nomor antrian yang berada pada *display* dan memanggil nomor antrian serta memberikan informasi nomor meja yang melayani nomor antrian tersebut. Untuk proses laporan – laporan yang dibutuhkan, maka kepala biro keuangan dapat langsung mengakses CMS agar dapat melihat dan mencetaknya untuk kepentingan pengambilan keputusan.

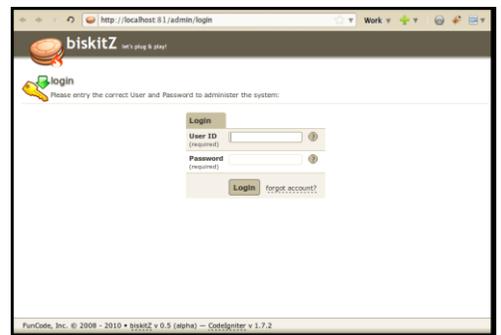
Sedangkan untuk *logical record structure* untuk sistem antrian dapat dilihat pada gambar 4.

Sistem antrian yang dikembangkan harus dapat menyimpan data nomor meja yang melayani. Nomor meja yang melayani juga harus menyimpan informasi waktu mulai melayani dan waktu selesai melayani. Karena dengan menyimpan informasi tersebut, maka nantinya dapat dibuat laporan yang dibutuhkan oleh pihak *executive*.

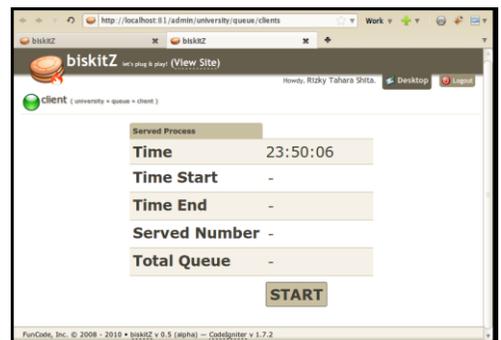
Dari sisi *customer*, sistem harus dapat menampung informasi nomor antrian *customer* tersebut dan harus dapat menampung informasi nomor antrian yang sedang ditangani. Selain itu, sistem juga harus dapat memanggil nomor antrian yang dipanggil oleh *customer services* dari meja mereka secara otomatis.

Implementasi dari sistem antrian yang

digunakan sebagai modul pada *biskitz* CMS dapat dilihat pada gambar – gambar dibawah ini:



**Gambar 4: biskitz CMS Login Area**



**Gambar 5: Aplikasi pemanggil nomor antrian yang terdapat pada meja customer service**



**Gambar 6: Display nomor antrian untuk memanggil customer**

#### 4.3. Pengujian

Pengujian dilakukan terlebih dahulu pada komputer lokal sebelum diterapkan pada lapangan. Beberapa komputer digunakan untuk melakukan simulasi pada keadaan yang sesungguhnya dengan jumlah meja dan *customer* yang cukup untuk memenuhi kebutuhan pelayanan kepada *customer*.

Setelah simulasi dilakukan dengan baik, maka implementasi dari sistem antrian untuk Direktorat Biro Keuangan Universitas Budi Luhur dapat diterapkan untuk uji coba tahap pertama. Pada tahap pertama ini, uji coba dilakukan pada konfigurasi *printer* untuk mencetak struk nomor antrian yang akan diterima oleh *customer* untuk mendapatkan nomor antriannya. Tahap kedua dilakukan dengan memanggil nomor antrian yang tersedia dari bagian CMS pada tiap staf. Pengecekan juga dilakukan terhadap suara pemanggil otomatis yang keluar dari *display* besar pada layar televisi.

#### 4.4. Implikasi Penelitian

##### a) Aspek Manajerial

Implikasi penelitian terhadap aspek manajerial adalah:

1. Implementasi sistem antrian dapat dimanfaatkan untuk memberikan laporan – laporan yang dibutuhkan oleh pihak *executive* agar dapat mengambil keputusan.
2. Pihak *executive* juga dapat memantau lamanya proses pelayanan yang

dilakukan oleh tiap staf, sehingga untuk kedepannya dapat memutuskan lamanya pelayanan yang harus diberikan kepada *customer*.

##### b) Aspek Sistem

Implikasi pada aspek sistem yang terjadi adalah:

Dengan membuat CMS yang dapat diintegrasikan dengan banyak biro yang terdapat pada Universitas Budi Luhur, maka hal ini memudahkan dalam melakukan *maintenance* jika terjadi hal – hal yang tidak sesuai. Direktorat Biro Teknologi Informasi dapat langsung berfokus pada aplikasi yang mengalami kendala, sehingga respon perbaikan dapat lebih cepat dirasakan oleh pengguna.

##### c) Aspek Penelitian Lanjut

Aspek penelitian lanjut yang dapat dilakukan yaitu:

1. Dengan *biskitZ*, maka modul – modul untuk keperluan sistem lainnya dapat dengan mudah dibuat dan diintegrasikan.
2. Sistem antrian yang berjalan dapat dikembangkan lagi untuk dapat menampilkan informasi yang lebih banyak, misalnya untuk menampilkan data nomor antrian yang sedang dilayani pada sebuah loket.

## 5. KESIMPULAN

Implementasi sistem antrian dengan menggunakan CMS yang dikembangkan untuk kebutuhan spesifik dapat memberikan kelebihan kepada Direktorat Biro Keuangan dalam memberikan pelayanan kepada *customer*. Penggunaan CMS yang telah dikembangkan memberikan keuntungan dari sisi pengembangan sistem antrian, dimana sistem antrian yang dibuat menjadi lebih cepat tidak saja dibantu dengan *prototype development methodology* tapi juga dengan *component of the shelf* yang sudah tersedia dan dapat digunakan dengan sedikit penyesuaian.

Sistem antrian yang dikembangkan menggunakan konsep n-tier, sehingga pengguna dapat langsung merasakan perubahannya jika ada penyesuaian atau perbaikan. Konsep n-tier ini juga sangat berperan dalam terintegrasinya modul – modul yang digunakan, sehingga dibutuhkan koneksi jaringan yang memadai. *Maintenance* pada jaringan lokal sangat diperlukan untuk dapat menjalankan sistem antrian dengan baik. Sedangkan untuk keperluan penampilan informasi yang didapat dari internet (seperti informasi berita atau mungkin kedepannya dengan menambahkan informasi kurs), dapat diperbaiki dari sisi penggunaan *bandwith* internet agar menjadi lebih baik.

Kedepannya, sistem antrian ini dapat dikembangkan untuk menampilkan informasi nomor loket yang sedang dilayani maupun informasi lainnya; juga dapat dengan mudah untuk mengganti tampilan pada sisi *display*, dikarenakan *biskitZ* sudah mendukung sistem *template* (*templating system*).

## DAFTAR PUSTAKA

1. "Codeigniter User Guide", [http://www.codeigniter.com/user\\_guide](http://www.codeigniter.com/user_guide), diakses pada 17 Oktober 2010 pukul 10:30:00.
2. Dennis, Alan, et.al, "Systems Analysis and Design with UML", 3<sup>rd</sup> ed., John Wiley & Sons, Ltd, Indianapolis, 2009.
3. Griffiths, Adam, "CodeIgniter 1.7 Professional Development", Packt Publishing Ltd, Birmingham, 2010.
4. Hoffer, Jeffrey A., et.al, "Modern Database Management", 8<sup>th</sup> ed., Pearson Prentice Hall, New Jersey, 2007.
5. O'Docherty, Mike, "Object-Oriented Analysis and Design Understanding System Development with UML 2.0", John Wiley & Sons, Ltd, England, 2005.