

PERANCANGAN SISTEM PERPUSTAKAAN MENGUNAKAN MODEL VIEW CONTROLLER (MVC) DENGAN METODE OBJECT MODELLING TECHNIQUE (OMT) PADA SEKOLAH MENENGAH PERTAMA NEGERI 44 PALEMBANG

Yudi Wiharto, Ari Irawan

visited.me123@gmail.com, ari.irawan@uinjkt.ac.id

*Teknik Informatika Fakultas Teknologi Informasi, Universitas Budi Luhur Jakarta
Sistem Informasi Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif
Hidayatullah Jakarta*

ABSTRAK

Perpustakaan adalah sebuah fasilitas atau tempat yang menyediakan sarana bahan bacaan. Tujuan adanya perpustakaan sendiri, khususnya pada perpustakaan sekolah adalah untuk memberikan sebuah layanan berupa informasi dalam kegiatan belajar dan mengajar, penggunaan teknologi informasi di perpustakaan bertujuan untuk meningkatkan efisiensi dalam melakukan pekerjaan dan kualitas pelayanan pada pengguna. Arsitektur MVC diharapkan mampu menghasilkan sebuah perangkat lunak yang lebih efisien. Sehingga dengan adanya perangkat lunak tersebut, pengguna nantinya dapat lebih mudah mengelola informasi perpustakaan dan pengembang dapat lebih mudah dalam pengembangan perangkat lunak. Perangkat lunak yang dibentuk menggunakan arsitektur MVC akan membentuk pola yang memiliki aturan, sehingga mempermudah dalam pengembangan perangkat lunak. Arsitektur MVC dibagi menjadi tiga bagian yaitu Model, View, dan Controller. Setiap bagian memiliki hubungan dan saling berkomunikasi antar bagian, sehingga membentuk perangkat lunak yang secara utuh. Bagian model dari sebuah arsitektur MVC bersifat independent, sifat independent model dari arsitektur MVC menjadikan perangkat lunak bersifat reuseability.

Kata Kunci: *Perpustakaan, Sistem, Perangkat lunak, Arsitektur, MVC, OMT*

PENDAHULUAN

Latar Belakang

Prinsipnya perpustakaan memiliki beberapa kegiatan-kegiatan pokok yaitu, mengumpulkan semua informasi yang berkaitan dengan kebutuhan pengguna, memelihara, melestarikan dan merawat seluruh koleksi perpustakaan, serta menyediakan bahan-bahan perpustakaan agar dapat dimanfaatkan dengan baik oleh semua pengguna. Saat ini pengguna perpustakaan menginginkan perpustakaan untuk memberikan layanan informasi yang tepat dan cepat.

Hal ini akan dapat terlaksana dengan baik apabila sebuah perpustakaan dapat menghadirkan dan memanfaatkan perkembangan teknologi informasi dalam pengelolaan perpustakaannya. Dengan pengelolaan perpustakaan seperti itu maka akan mempermudah dan mengefisienkan pekerjaan pengelolaan perpustakaan, memberikan layanan yang lebih baik pada pengguna, serta meningkatkan citra perpustakaan dan pustakawan (Ishak, 2008).

Namun dengan jumlah pustakawan yang terbatas, dan proses pengelolaan

informasi perpustakaan pada Sekolah Menengah Pertama Negeri 44 yang masih manual dalam hal manajemen informasi perpustakaan dan dirasakan kurang efisien, baik dalam keamanan data maupun efektifitas waktu. Maka pada penelitian ini penulis mencoba merancang sebuah model sistem perangkat lunak perpustakaan menggunakan arsitektur *Model View Controller* (MVC) yang dikombinasikan dengan metode *Object Modelling Technique* (OMT). Arsitektur MVC diharapkan mampu menghasilkan sebuah perangkat lunak yang lebih efisien. Sehingga dengan adanya perangkat lunak tersebut pustakawan nantinya dapat lebih mudah mengelola informasi perpustakaan dan pengembang dapat lebih mudah dalam pengembangan perangkat lunak.

Model View Controller (MVC) merupakan paradigma pengembangan sistem yang memisahkan komponen data (*Object Model*) dengan komponen desain (*Object View*) dalam sebuah komponen kontrol (*Object Controller*) masing-masing komponen MVC bersifat *independen*, namun saling berkomunikasi satu sama lain (Chico, 2009).

Object Modelling Technique (OMT) sebuah metode *Object Oriented* untuk membangun *software* lewat koleksi objek-objek yang didalamnya mengandung data dan *behavior*. Karakteristik dari OMT adalah menemukan hal mendasar dari *Object Oriented* berupa *Identity*, *Classification* (*class & Object*), *polymorphism*, *Inheritance*.

Rumusan Masalah

Adapun rumusan masalah dari uraian latar belakang diatas adalah:

1. Proses manajemen informasi perpustakaan yang selama ini dilakukan dengan manual dan kurang efisien dapat digantikan

menggunakan sebuah perangkat lunak.

2. Perlunya sebuah model perangkat lunak yang bersifat independen, yang tidak dipengaruhi oleh tampilan perangkat lunak yang akan dibangun, data yang akan diolah, dan proses-proses yang akan dilakukan.

Batasan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka batasan masalah dalam penelitian ini adalah sebagai berikut :

“Bagaimana merancang sistem perangkat lunak perpustakaan menggunakan konsep *Model View Controller*”

Tujuan Penelitian

Ada pun tujuan penelitian yang ingin dicapai yaitu:

1. Untuk mengetahui bagaimana merancang sistem perangkat lunak perpustakaan menggunakan konsep *Model View Controller*.
2. Menghasilkan perangkat lunak yang mampu memenuhi kebutuhan manajemen informasi perpustakaan.

Manfaat Penelitian

Hasil penelitian ini diharapkan dapat:

1. Memberikan informasi yang berguna dalam hal perancangan sistem perpustakaan menggunakan konsep *Model View Controller* (MVC) yang dikombinasikan dengan metode *Object Modelling Technique* (OMT).
2. Membantu pengembangan perangkat lunak dan bisa dikembangkan oleh para pengembang yang berbeda yang mengacu pada aturan model yang dihasilkan.
3. Menghasilkan perangkat lunak yang dapat dikembangkan dan mampu memenuhi kebutuhan manajemen informasi perpustakaan.

Sistem

Istilah sistem sering kita jumpai, baik dalam media grafik, seperti surat kabar dan majalah, maupun media elektronik, seperti radio dan televisi. (Lucas,1987) mengartikan sebuah sistem sebagai suatu himpunan atau kumpulan dari unsur, komponen, atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain. Sedangkan (Indrajit, 2000) mendefinisikan sebuah sistem sebagai kumpulan dari komponen-komponen yang memiliki unsur keterkaitan antara satu dan lainnya. Sedangkan (Zwass, 1997) menyatakan sistem adalah “*Set of components (subsystems or elementary parts) that operate together to achieve a common objective (or multiple objectives).*” Sehingga dapat diartikan bahwa sistem adalah merupakan suatu hal yang saling terkait satu sama lain demi mencapai sebuah tujuan yang sama.

Dari definisi sistem diatas maka dapat di definisikan istilah sistem perpustakaan. Pada kebanyakan literature yang ada, sistem perpustakaan termasuk di dalam kajian sistem informasi manajemen (SIM). (Oetomo, 2002) memasukkan sistem informasi perpustakaan di dalam sistem informasi manajemen berdasarkan bidang minat perusahaan atau organisasi. Sehingga dengan memodifikasi yang telah disampaikan (Davis, 1988) tentang definisi dari Sistem Informasi Manajemen, maka Sistem Informasi Manajemen Perpustakaan dapat didefinisikan sebagai sistem manusia dan mesin yang terpadu atau terintegrasi, untuk memberikan informasi guna mendukung fungsi manajemen, operasional, dan pengambilan keputusan dalam sebuah perpustakaan. (Rowley, 1998) yang menyatakan bahwa fokus pada sistem informasi (manajemen) perpustakaan adalah untuk mendukung pelayanan secara efektif bagi pengguna,

manajemen pengadaannya, dan secara umum manajemen layanan-layanan yang diberikan oleh perpustakaan dan badan-badan lainnya yang menyelenggarakan akses terhadap koleksi-koleksi dokumen.

Perpustakaan

Dalam bahasa Indonesia dikenal istilah “perpustakaan”(berasal dari kata Sanskerta pustaka) artinya kitab, buku. Dalam bahasa Inggris, tentu mengenal istilah *library* (berasal dari kata Latin *liber* atau *libri*) artinya buku. Dari kata Latin terbentuklah istilah *librarian* yang artinya tentang buku. Dari istilah diatas dapat diperoleh batasan dari perpustakaan yaitu merupakan kumpulan buku dan bahan pustaka lainnya yang dapat digunakan untuk keperluan studi atau bacaan, serta kenyamanan atau kesenangan (*Webster's Third Edition International Dictionary* ,1961).

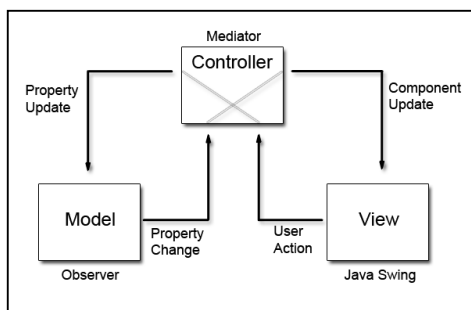
Definisi dari Berbagai Sumber :

1. Perpustakaan adalah kumpulan materi tercetak dan media noncetak atau sumber informasi dalam komputer yang disusun secara sistematis untuk digunakan pemakai (*International Federation of Library Association and Institutions*).
2. Perpustakaan diartikan sebuah ruangan atau gedung yang digunakan untuk menyimpan buku dan terbitan lainnya yang biasanya disimpan menurut tata susunan tertentu yang digunakan pembaca bukan untuk dijual (Sulisty, Basuki ; 1991).
3. Perpustakaan adalah fasilitas atau tempat menyediakan sarana bahan bacaan. Tujuan dari perpustakaan sendiri, khususnya perpustakaan perguruan tinggi adalah memberikan layanan informasi untuk kegiatan belajar, penelitian, dan pengabdian masyarakat dalam rangka melaksanakan Tri Dharma

Perguruan Tinggi (Wiranto dkk,1997).

Model View Controller (MVC)

Konsep *Model View Controller* menyediakan sebuah solusi terhadap permasalahan dengan membagi aplikasi menjadi bagian – bagian tersendiri yaitu, *Model*, *View* dan *Controller*, yang memisahkan antar bagian - bagian tersebut dan membuat interaksi diantaranya.



Gambar 1. Arsitektur *Model View Controller* (MVC)

Gambar di atas menunjukkan 3 komponen yang terdapat dalam konsep MVC dan interaksi yang terjadi.

1. *Model*, pada bagian ini biasanya berhubungan langsung dengan *database* untuk dapat memanipulasi data, menangani validasi dari bagian *controller*, namun tidak bisa berhubungan secara langsung dengan bagian *view*.
2. *View*, adalah merupakan salah satu bagian yang menangani *presentation logic*. Pada suatu aplikasi pada bagian ini biasanya, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Pada bagian ini tidak memiliki akses langsung terhadap bagian *model*.
3. *Controller*, merupakan bagian yang dapat mengatur hubungan antara bagian *model* dan bagian *view*, *controller* sendiri dapat berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Kelebihan dalam penggunaan *layer Controller* secara terpisah: Pertama, dengan menggunakan komponen terpisah untuk menampung secara detail dari transisi *layer*, komponen *view* dapat didesain tanpa harus memperhatikan bagian lain secara berlebihan. Hal ini akan memudahkan *team* pengembang *multiple interface* dapat bekerja secara terpisah dari yang lain secara simultan. Interaksi antar komponen *View* akan terabstraksi dalam *Controller*. Kedua, dengan menggunakan *layer* terpisah yang melakukan *update* terhadap komponen *Model*, detail tersebut dapat dihapus dari *layer presentasi*. *Layer presentasi* kembali pada fungsi utamanya untuk menampilkan data kepada *user*. Detail tentang bagaimana data dari *user* dapat mengubah ketetapan aplikasi akan disembunyikan oleh *Controller*. Hal ini memisahkan dengan jelas antara *presentation logic* dengan *business logic*.

Object Modelling Technique (OMT)

Object Modelling Technique (OMT) sebuah metode *Object Oriented* untuk membangun *software* lewat koleksi objek-objek yang didalamnya mengandung data dan *behavior*. Karakteristik dari OMT adalah menemukan hal mendasar dari *Object Oriented* berupa *Identity*, *Classification* (*class* & *Object*), *polymorphism*, *Inheritance*. Esensi dari *Object Oriented Development* OMT adalah mengidentifikasi dan mengorganisasikan *application-domain concept* sampai pada *implementation-domain concept*.

Metode *Object Modelling Technique* terdiri dari analisis, design dan implementasi. *Object Modelling Technique* membagi analisis & design menjadi 3 bagian, yaitu:

1. *Analysis*: membuat model dari dunia nyata berdasarkan problem domain
2. *System Design*: membangun arsitektur sistem secara keseluruhan

3. *Object Design*: menemukan struktur dari Objek sehingga memudahkan dalam proses implementasi

Langkah diatas bukanlah suatu urutan, namun perlu pengulangan sehingga menemukan bentuk terbaik, sistem *Object Modelling Technique* memberikan 3 model view:

1. *Object View (The information view)*
2. *Dynamic View (Then process computation)*
3. *Fuctional View (Control flow)*

Ruang Lingkup

Ruang lingkup penelitian ini adalah mengenai perancangan sistem perpustakaan menggunakan konsep *model view controller* pada Sekolah Menengah Pertama Negeri 44 Palembang. Perancangan sistem perpustakaan menggunakan konsep *model view controller* ini dilakukan dengan menggunakan metode pengembangan *Object Modelling Technique (OMT)*. Hasil perancangan ini kemudian diimplementasikan ke dalam kode-kode program dengan menerapkan konsep berorientasi objek.

METODOLOGI PENELITIAN

Lokasi Penelitian

Penulis melakukan penelitian pada Perpustakaan Sekolah menengah Pertama Negeri 44 Palembang.

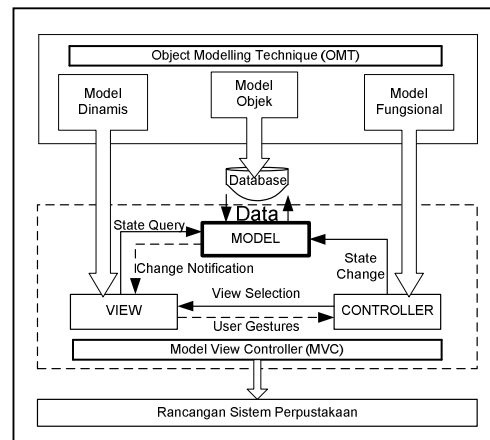
Kerangka Pemikiran

Setelah melihat dari penelitian-penelitian sebelumnya, maka kerangka pemikiran untuk penelitian ini adalah seperti gambar 2.

Penelitian ini bertujuan untuk menghasilkan sebuah desain model dari hasil kombinasi antara OMT dan MVC, dimana metode analisis objeknya merupakan hasil dari OMT dan hasil modelnya dari arsitektur MVC.

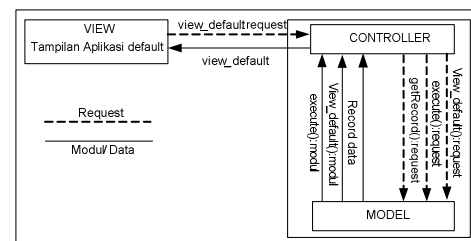
Object Modelling Technique (OMT) sebuah metode *Object Oriented* untuk membangun *software* lewat koleksi

objek-objek yang didalamnya mengandung data dan *behavior*. Karakteristik dari OMT adalah menemukan hal mendasar dari *Object Oriented* berupa *Identity, Classification (class & Object), polymorphism, Inheritance*. Konsep MVC menyediakan sebuah solusi terhadap permasalahan dengan membagi aplikasi menjadi bagian – bagian tersendiri, *Model, View* dan *Controller*, memisahkan antar bagian tersebut dan membuat tata interaksi diantaranya.



Gambar 2. Kerangka pemikiran untuk pembangunan perangkat lunak

Penggunaan arsitektur MVC yang dikombinasikan dengan OMT, mampu menghasilkan sebuah model independen yang memisahkan tampilan, dan jenis-jenis proses pada sebuah perangkat lunak. Sedangkan metode *Object Modelling Technique (OMT)* membentuk dalam 3 model, yaitu: model objek, model dinamis, dan model fungsional.

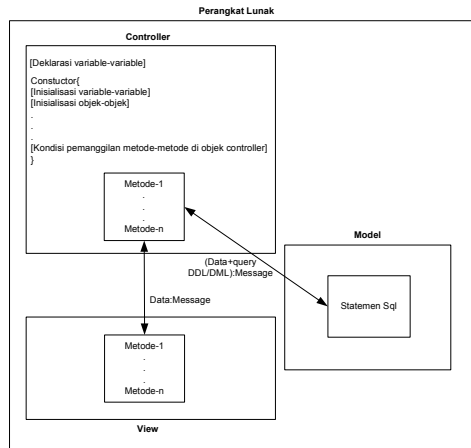


Gambar 3. Pemodelan MVC

Desain Penelitian

Pemodelan perangkat lunak dengan arsitektur *Model View Controller* (MVC) yang dihasilkan dapat dilihat pada gambar 3.

Jika ditinjau dari pemodelan yang dihasilkan, maka model diatas dapat dibentuk dalam sebuah perangkat lunak seperti pada gambar 4.



Gambar 4. Pemodelan perangkat lunak dengan arsitektur MVC

Pemodelan yang dihasilkan membentuk sebuah pola bagi perangkat lunak. Dimana *View* dan *Model* dihubungkan dengan bagian *Controller* dengan mengikuti aturan dan pola tertentu, dimana bagian *Model* melakukan manajemen *database* yang meliputi konektivitas, manajemen DDL dan DML.

Manajemen DDL

Prosedure manajemen DDL dalam pemodelan perangkat lunak sebagai berikut:

1. Melakukan inialisasi *object controller* sebagai objek *instance*.
2. Memanggil metode *execute()* pada *object controller instance*.
3. Pada *object controller* dilakukan inialisasi objek model sebagai *object instance*.

Komunikasi yang terjadi antar *object controller* dan model, dengan

melakukan pengiriman *message* berupa *query DDL* yang akan dijalankan.

Manajemen DML

Dalam sebuah perangkat lunak, implementasi manajemen DML biasanya membutuhkan *interface* yang digunakan si pengguna untuk melihat hasil proses manajemen DML atau sarana bantu interaktif si pengguna yang dibutuhkan perangkat lunak untuk melakukan manajemen DML, misal: menampilkan data, menghapus data, menambah data, mengubah data, dan pencarian data pada sebuah table atau beberapa table.

Desain Proses Analisis- Analisis Objek

Domain Masalah

Domain masalah yang merupakan pokok bahasan dalam penelitian adalah sebagai berikut:

1. Rancangan Sistem.
2. Model Perangkat Lunak.

Rancangan Sistem

Rancangan sistem adalah spesifikasi atau konstruksi dari sebuah teknik, solusi berbasis komputer untuk identifikasi kebutuhan-kebutuhan bisnis dalam sebuah analisis sistem, dapat dengan menambahkan, menghapus, mengganti bagian-bagian yang berhubungan dengan sistem original (Witten, Bentley & Dittman, 2004).

Sistem yang dibuat, bekerja dengan memberikan kemudahan dalam proses perpustakaan. Dimana dalam prosesnya sistem mengatur data keanggotaan, data buku dan juga memproses peminjaman dan pengembalian buku, selain itu sistem juga memberikan kemudahan dalam pencatatan data buku-buku perpustakaan, kemudian sistem juga bekerja dalam proses mengatur laporan seperti laporan anggota, dan laporan peminjaman.

Aturan atau kebijakan pada sistem juga dibuat sesuai dengan aturan atau

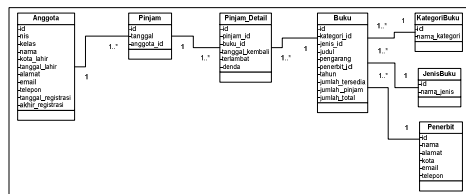
kebijakan pada sistem sebelumnya atau sesuai dengan permintaan dari instansi. Adapun aturan atau kebijakan itu antara lain :

1. Setiap siswa dapat menjadi anggota perpustakaan.
2. Setiap siswa wajib mendaftarkan diri sebagai anggota sebelum melakukan peminjaman buku.
3. Anggota hanya bisa melakukan peminjaman buku maksimal 2 buku dan maksimal peminjaman 3 hari.
4. Denda terlambat pengembalian buku sebesar Rp 1.000 / hari.

Model Objek

Berdasarkan proses diatas, didapatkan beberapa kelompok objek, yaitu:

1. Objek siswa yang mendaftar sebagai anggota diberi nama objek "anggota".
2. Objek yang melakukan peminjaman buku diberi nama objek "Pinjam", dan pengembalian buku beserta dendanya diberi nama objek "Pinjam_Detail".
3. Objek buku yang akan dipinjam anggota diberi nama objek "buku", dan buku mempunyai penerbit yang diberi nama objek "penerbit" kategori buku yang diberi nama objek "kategori buku" dan buku juga mempunyai jenis buku yang diberi nama objek "jenis buku".



Gambar 5. Class diagram bagian model pada tahap analisis

Model Dinamis

Event Flow Pada Model

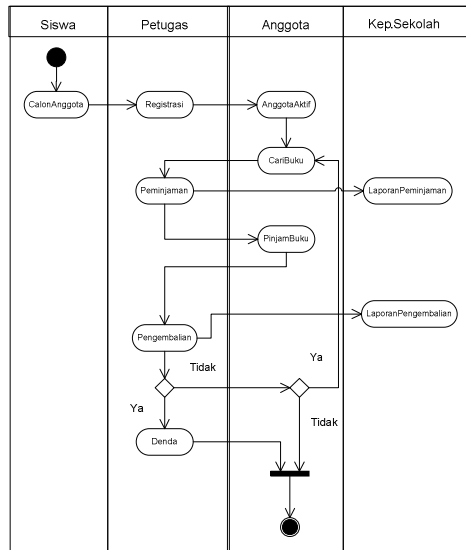
1. Pada objek model
 - a. Pertama kali akan menciptakan objek *instance*

dari objek Anggota, dengan menyertakan nilai parameter yang dibutuhkan, seperti: : id, nis, kelas, nama, kota_lahir, tanggal_lahir, alamat, email, telepon, tanggal_registrasi, akhir_registrasi.

- b. Objek *instance* dari objek Pinjam dengan menyertakan nilai parameter yang dibutuhkan, seperti: id, tanggal, anggota_id.
- c. Objek *instance* dari objek Pinjam_Detail dengan menyertakan nilai parameter yang dibutuhkan, seperti: id, pinjam_id, buku_id, tanggal_kembali, terlambat, denda.
- d. Objek *instance* dari objek Buku dengan menyertakan nilai parameter yang dibutuhkan, seperti: id, kategori_id, jenis_id, judul, pengarang, penerbit_id, tahun, jumlah_tersedia, jumlah_pinjam, jumlah_total.
- e. Objek *instance* dari objek KategoriBuku dengan menyertakan nilai parameter yang dibutuhkan, seperti: id, nama_kategori.
- f. Objek *instance* dari objek JenisBuku dengan menyertakan nilai parameter yang dibutuhkan, seperti: id, nama_jenis.
- g. Objek *instance* dari objek Penerbit dengan menyertakan nilai parameter yang dibutuhkan, seperti: id, nama, alamat, kota, email, telepon.

1. Pemanggilan pada objek model
 - a. Jika objek Anggota dipanggil, maka objek Anggota akan melakukan inialisasi data anggota.

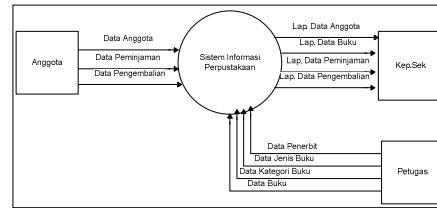
- b. Jika objek Pinjam dipanggil, maka objek Anggota juga akan dipanggil sebagai inialisasi anggota_id.
- c. Pada objek Pinjam_Detail akan memanggil objek Pinjam dan objek Buku sebagai inialisasi pinjam_id dan buku_id.
- d. Jika objek Buku dipanggil, maka objek KategoriBuku, JenisBuku, dan Penerbit juga akan dipanggil sebagai inialisasi kategori_id, jenis_id, dan penerbit_id.



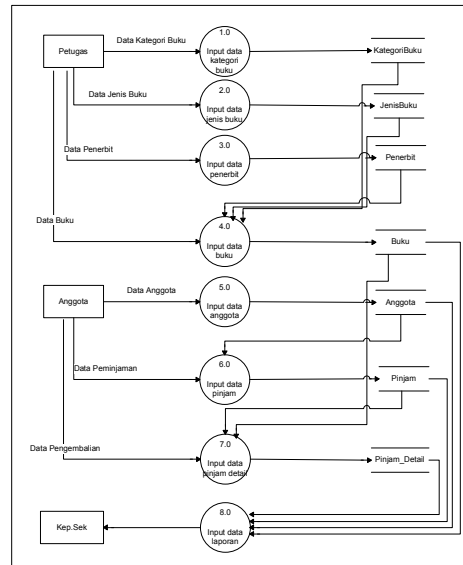
Gambar 6. Activity diagram bagian model dalam tahap analisis

Model Fungsional

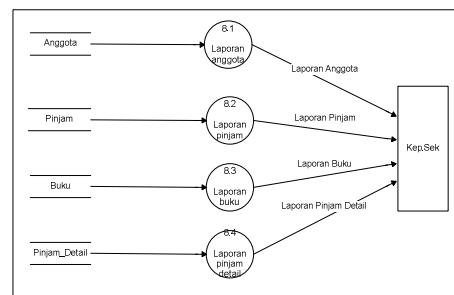
Model fungsional menggambarkan fungsi yang meminta operasi pada model objek dan aksi pada model dinamis. Berdasarkan model objek dan model dinamis diatas, maka model fungsionalnya dapat digambarkan menggunakan diagram aliran data.



Gambar 7. Diagram konteks untuk aliran data.



Gambar 8. Diagram level 0 untuk aliran data objek



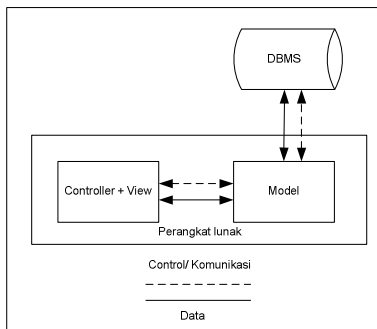
Gambar 9. Diagram level 1 untuk proses objek

Analisis Model

Model Perangkat Lunak

Model perangkat lunak yang akan dibangun adalah sebuah model prototype yang bertugas untuk

memanajemen data dari sebuah *DBMS*. Mulai dari proses koneksi ke *DBMS*, membuat objek yang dibutuhkan dalam *DBMS* menyimpan data ke dalam *DBMS*, mengambil data dari *DBMS*. Sehingga bisa disimpulkan, terdapat dua jenis aliran antara *DBMS* yang digunakan dengan model perangkat lunak, yaitu aliran komunikasi antara model dengan *DBMS*, dan aliran data dari model ke *DBMS* atau sebaliknya.



Gambar 10. Aliran data dan komunikasi dalam pemodelan

Adapun permasalahan yang harus diperhatikan dalam membuat model adalah sebagai berikut:

1. Bagaimana model mampu melakukan konektivitas ke sebuah *DBMS* yang akan digunakan.
2. Bagaimana model mampu dikontrol atau berkomunikasi dengan bagian *Controller* dan *View* dalam perangkat lunak.

Permasalahan-permasalahan diatas bisa diselesaikan dengan cara sebagai berikut:

1. Setiap *DBMS* memiliki *class* tersendiri. Dikarenakan metode konektivitas *DBMS* yang ada, memiliki perbedaan satu sama lain, baik dilihat dari struktur perintah, maupun parameter konektivitas yang dibutuhkan.
2. *Class-class* yang melayani kebutuhan *DBMS*, memiliki hubungan (asosiasi) dengan sebuah *class* utama. *Class* utama inilah

yang nantinya akan melayani *class-class* dibagian *Controller* dan *View* untuk berkomunikasi dengan *DBMS*.

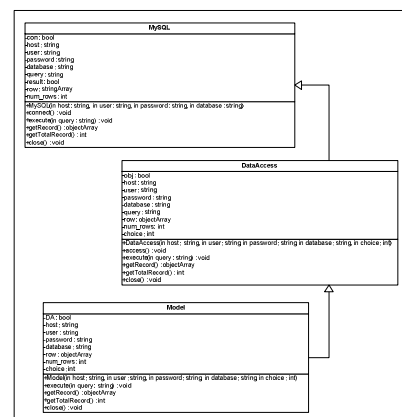
Adapun yang menjadi kunci dari model perangkat lunak dan sekaligus untuk menerangkan *class* dalam analisis perangkat lunak, yaitu:

1. *DataAccess*: Objek yang melakukan komunikasi dengan Objek *DBMS* (*MySQL*).
2. *Model*: Objek yang melakukan komunikasi dengan Objek *DataAccess* dan Objek-objek yang ada di bagian *Controller* dan *View*.

Konektivitas *DBMS*

Berdasarkan model perangkat lunak, didapatkan beberapa kelompok objek, yaitu:

1. Objek yang melakukan komunikasi dengan *DBMS*, yaitu: objek yang berkomunikasi dengan *MySQL* (diberi nama objek “*MySQL*”).
2. Objek yang melakukan komunikasi dengan objek-objek eksternal (diluar model, yaitu dengan *Controller*, dan *View*. Diberi nama objek “*Model*”) dan internal (didalam model, yaitu dengan objek yang berkomunikasi dengan *DBMS*. Diberi nama objek “*DataAccess*”).



Gambar 11. Class diagram bagian model pada tahap analisis

Controller) bisa saling berinteraksi di dalam sebuah perangkat lunak yang akan dikembangkan.

DAFTAR PUSTAKA

- [1] Jogyanto. (2008), *Metodologi Penelitian Sistem Informasi*, Andi, Yogyakarta.
- [2] Kadir, Abdul (2009), *Dasar Perancangan Dan Implementasi Database Relasional*, Andi, Yogyakarta.
- [3] Kadir, Abdul (2010), *Mudah Mempelajari Database MySQL*, Andi, Yogyakarta.
- [4] Nugraha, Bunafit (2005), *Database Relasional Dengan MySQL*, Andi, Yogyakarta.
- [5] Firdausy. (2008), *Sistem Informasi Perpustakaan Berbasis Web Dengan Php Dan MySQL*, Vol. 6, No. 2, h. 109-114
- [6] Gupta, P & Govil, M.C. (2010), 'MVC Design Pattern for the multi framework distributed applications using XML, spring and struts framework', *Jurnal international computer science and engineering*, Vol. 02, No. 04, h. 1047-1051.
- [7] Mamluati, Heni. (2006), *Pengembangan Antarmuka Sistem Katalogisasi Berbasis Web Menggunakan Konsep Model-View-Controller (MVC) Studi Kasus: Perpustakaan IPB [skripsi]*. Institut Pertanian Bogor. Fakultas Matematika dan Ilmu Pengetahuan Alam.
- [8] Yanto. (2009), 'Penerapan Java Server Faces Untuk Design Pattern Web', *Jurnal Informatika*, Vol. 5, No. 1, h. 38-49.
- [9] Pressman, Roger S. (2002), *Rekayasa Perangkat Lunak Pendekatan Praktisi*, Andi, Yogyakarta.
- [10] Prasetyo, Dwi, Didik. (2007), *150 Rahasia Pemrograman Java*, Elex Media Komputindo, Jakarta.
- [11] Hariyanto, Bambang. (2005), *Esensi-Esensi Bahasa Pemrograman Java*, Informatika, Bandung.
- [12] Munawar. (2005), *Pemodelan Visual Dengan UML*, Graha Ilmu, Yogyakarta.
- [13] Badriyah, T. (2007), *UML Class Diagram*, dari <http://lecturer.eepis-its.edu/~tessy>, diakses 2 April 2012.