

PEMODELAN BASIS DATA DENGAN PENDEKATAN MODEL DATA BERORIENTASI OBJEK PADA NATIVE APPS LOKASI TEMPAT IBADAH DI KOTA PALANGKA RAYA

Rio Irawan¹, Rommi Kaestria²

Manajemen Pendidikan Islam - Institut Agama Islam Negeri (IAIN) Palangka Raya, Jl. G. Obos Kompleks Islamik Center, Palangka Raya, Kalimantan Tengah, 73112¹
Sistem Informasi - STMIK Palangkaraya, Jl. G. Obos No. 114, Palangka Raya, Kalimantan Tengah²

Email : rioirawan@iain-palangkaraya.ac.id¹

Email : rokaforddev@gmail.com²

ABSTRAK

Pemodelan basis dengan pendekatan model data berorientasi objek dibuat untuk menghasilkan sebuah desain basis data berorientasi objek dalam pengembangan *native apps* lokasi tempat ibadah di Kota Palangka Raya. Perkembangan *native apps* yang begitu pesat baik pada platform android ataupun iOS akan memberikan tantangan tersendiri bagi pengembang untuk menghasilkan sebuah aplikasi yang mendekati atau semaksimal mungkin sesuai dengan kondisi *real*. Dengan model data berorientasi objek ini, mempermudah dalam melakukan analisis permasalahan sebab akan analogikan menjadi sebuah objek, kemudian objek tersebut akan dibuat yang mempunyai sifat atau ciri khas yang berbeda antar kelas. Pemodelan model data berorientasi objek menggunakan *Unified Modelling Language* (UML) dengan menggunakan *Class Diagram*. Hasil dari penelitian ini berupa desain pemodelan basis data agar mempermudah pengembang aplikasi pada saat pengembangan *native apps* yang umumnya dikembangkan dengan pemrograman berorientasi objek.

Kata kunci : *Pemodelan Basis Data, Model Data Berorientasi Objek, Native Apps*

ABSTRACT

Base modeling with an object-oriented data model approach is made to produce an object-oriented database design in developing native apps for places of worship in Palangka Raya City. The rapid development of native apps on both the Android and iOS platforms will provide a challenge for developers to produce an application that is as close or as close as possible to real conditions. With this object-oriented data model, it is easier to do problem analysis because it will be analogous to an object, then the object will be created that has different characteristics or characteristics between classes. Object-oriented data modeling using Unified Modeling Language (UML) using Class Diagrams. The results of this study are in the form of database modeling design to facilitate application developers when developing native apps which are generally developed with object-oriented programming

Keywords : *Database Modeling, Object-Oriented Data Model, Native Apps*

1. PENDAHULUAN

Teknologi informasi memiliki peran yang sangat besar terhadap arus informasi sekarang ini, dalam segala aspek bidang, maupun dalam kegiatan rutin yang dilakukan sehari-hari. Teknologi informasi erat kaitannya dengan data, dari yang awalnya berupa *raw data* atau data mentah kemudian diolah menjadi sebuah informasi akhir yang bermanfaat bagi pengguna. Semakin banyak data yang tersimpan dalam sebuah sistem tentunya akan semakin banyak pula informasi yang dapat diolah dan dihasilkan yang disesuaikan dengan kebutuhan informasi yang diinginkan.

Pengolahan *raw data* dilakukan dengan memanfaatkan *Database Management System* (DBMS) yang dibuat agar memudahkan dan mengoptimalkan pemasukan, pengeditan, penghapusan dan pengambilan informasi terhadap *database* [4]. Manfaat dan kelebihan melakukan penyimpanan data dalam DBMS yaitu dalam segi *performance*, integritas, independensi, sentralisasi, dan *security*. [4]

Dengan perkembangan teknologi yang semakin pesat mendorong serta kebutuhan akan informasi yang ceoat dan tepat pula, untuk itu diperlukan desain basis data yang mampu menunjang kebutuhan informasi yang sering berubah-ubah setiap waktu. Dengan memanfaatkan pendekatan berbasis objek, sebuah kebutuhan data akan dianalisis dan dibagi

kedalam objek-objek tertentu yang terpisah dari objek lainnya, sehingga mudah dalam proses pemodelannya sesuai dengan permasalahan yang ditemukan yang kemudian akan diterjemahkan ke dalam sebuah *database*.

Native apps merupakan sebuah aplikasi yang umumnya diperuntukkan bagi pengguna *smartphone*, yang dikembangkan dengan menggunakan Bahasa pemrograman tertentu yang spesifik untuk platform tertentu, misalnya *smartphone* berbasis Android atau IOS. Perkembangan dan pengembangan *native apps* yang dilakukan oleh para *developer* yang banyak dikembangkan adalah aplikasi berbasis lokasi yang berguna sebagai petunjuk arah kesuatu daerah atau objek tertentu dengan memanfaatkan layanan GPS dan peta digital yang dimiliki oleh beberapa penyedia layanan, misalnya *Google Maps*, *Here Maps* dan lain sebagainya dengan cara melakukan sinkronisasi data antara sistem yang dikembangkan dengan *Application Programming Interface* (API) yang dimiliki oleh penyedia layanan peta digital.

Detail lokasi, terutama dalam hal ini *longitude* dan *latitude* merupakan indikator yang sangat berpengaruh terhadap aplikasi berbasis lokasi. Dengan data *longitude* dan *latitude* yang tepat maka sistem akan menampilkan data lokasi yang sesuai, petunjuk arah yang tepat dan detail-detail informasi lainnya. Untuk itu, diperlukannya sebuah pemodelan *database* yang baik sehingga menghasilkan sistem baik pula, yaitu sistem yang tepat dalam memberikan informasi lokasi tempat ibadah yang di-*request* oleh pengguna melalui aplikasi.

Fokus pada penelitian ini akan memberikan gambaran mengenai basis data yang disusun atau dirancang dengan melakukan pendekatan berbasis objek dengan studi kasus lokasi tempat ibadah yang terdapat di Kota Palangka Raya, Provinsi Kalimantan Tengah.

2. TINJAUAN PUSATAKA

Beberapa rujukan penelitian terdahulu yang dijadikan dasar pada penelitian ini yaitu sebagai berikut:

- Penelitian dengan judul Implementasi Location Based Services (lbs) Kantor Pemerintahan Kota (studi Kasus : Kota Palangka Raya, Provinsi Kalimantan Tengah, Indonesia). Pada penelitian ini menjelaskan mengenai implementasi aplikasi berbasis lokasi yang mampu memberikan detail informasi mulai dari jarak tempuh, estimasi waktu, petunjuk arah, rute dan lain sebagainya.
- Penelitian dengan judul Pemodelan Basis Data Akademik Universitas XYZ Menggunakan Pendekatan Objek.. penelitian ini menghasilkan desain basis data akademik untuk menghubungkan antara spesifikasi kebutuhan dan implementasi yang akan membantu pihak pengembang sistem.
- Penelitian dengan judul Perancangan Database Point of Sales Apotek Dengan Menerapkan Model Data Relasional. Pada penelitian ini desain database dengan menggunakan metode *Database Life Cycle*

(DBLC) yaitu desain konseptual, desain logikal dan desain fisikal.

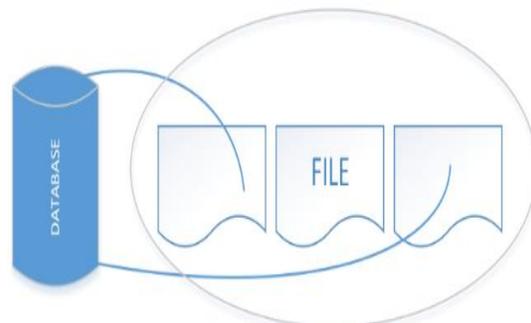
2.1 Landasan Teori

Teori pendukung dalam penelitian ini yaitu sebagai berikut :

2.1.1 Basis Data

Basis data terdiri dari 2 kata, yaitu basis dan data. Basis dapat diartikan sebagai markas, Gudang, tempat berkumpul. Sedangkan data adalah fakta yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks gambar, bunyi atau kombinasinya. Basis data sendiri dapat di definisikan dalam sejumlah sudut pandang seperti : [5]

- Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi), untuk memenuhi berbagai kebutuhan.
- Kumpulan *file* yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.



Gambar 1. Ilustrasi Basis Data [5]

2.1.2 Model Data Berorientasi Objek

Menurut Connolly dan Begg dalam [2] Basis data berbasis objek terdiri dari *Object-Oriented Data Model* (OODM), *Object-Oriented Data Base* (OODB) dan *Object-Oriented DBMS* (OODBMS). OODM adalah suatu model data logika yang menangkap semantik dari objek yang didukung oleh pemrograman berbasis objek sedangkan OODB adalah sekelompok objek persistent dan sharable yang didefinisikan OODM, dan OODBMS merupakan manajer dari OODB.

Menurut [5], manifesto OODBMS yaitu mendukung objek objek kompleks, mendukung identitas objek, memungkinkan objek dikapsulkan, mendukung tipe atau kelas, mendukung pewarisan, menghindari teknik binding yang dini dan dapat mengingat lokasi data.

Kelemahan OODBMS:

- Kemungkinan korupsi basis data
- Kekurangan dan perluasan logik
- OODB lebih sulit dimengerti.
- Dibutuhkan orang yang benar-benar menguasai OODB.

Keunggulan OODBMS:

- Fitur fitur lanjut
- Sistem tipe yang seragam
- Dapat menyimpan kelas dengan jumlah banyak
- Dapat mengatasi data yang *interrelated* dan kompleks

2.1.3 Native Apps

Purely Native apps – these apps are written in a device-specific language, using a device-specific programming interface: objective-C for iPhone apps or Java for Android apps. Native apps can access all the capabilities of the device and can take many forms, from simple utility to advanced 3-D games. [6]

Aplikasi native adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk platform tertentu. Contoh populernya yakni penggunaan bahasa pemrograman Objective-C atau Swift untuk platform iOS (Apple). Adapun platform Android yang menggunakan bahasa pemrograman Java.[7]

2.1.4 Location Based Service

Sebuah layanan berbasis lokasi adalah layanan informasi atau hiburan yang dapat diakses dengan perangkat *mobile* melalui jaringan selular. Sistem Layanan Berbasis Lokasi, atau lebih dikenal dengan *Location-Based Services* (LBS), menggabungkan antara proses dari layanan *mobile* dengan posisi geografis dari penggunaanya. Posisi target, dimana sebuah target bisa jadi adalah pengguna *Location-Based Services* itu sendiri atau entitas lain yang tergabung dalam suatu layanan. *Location Provider* Menyediakan teknologi pencarian lokasi yang berasal dari perangkat. API Location berhubungan dengan data *Global Positioning System* (GPS) dan data lokasi *real-time*. [1]

Menurut Manav dan Shukla dalam [1] Unsur utama dari LBS yaitu *location manager* (*API Maps*) dan *location provider* (*API Location*). Sedangkan komponen LBS menurut Pankti, Jain dan Shakwala dalam [1] yaitu piranti *mobile*, jaringan komunikasi, komponen *positioning*, penyedia layanan dan aplikasi dan penyedia data dan konten

2.1.5 Unified Modeling Language (UML)

Unified Modeling Language (UML) digunakan untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. [1]

Unified Modeling Language selanjutnya disebut UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem.[8]

3. METODE PENELITIAN

Tahapan yang dilakukan dalam melakukan penelitian ini untuk menghasilkan desain atau model basis data lokasi tempat ibadah di Kota Palangka Raya dengan melakukan studi pustaka terhadap penelitian-penelitian yang relevan terhadap studi kasus yang diteliti serta melakukan observasi terhadap beberapa aplikasi *native* berbasis lokasi dalam proses pengumpulan data.



Gambar 2. Rancangan Penelitian

Berdasarkan hasil pengumpulan data, dianalisis hingga menghasilkan atau mendefinisikan kelas-kelas yang akan digunakan untuk melakukan proses perancangan basis data hingga pada akhirnya dilakukan implementasi pada saat pengembangan perangkat lunak atau aplikasi *native*.

4. HASIL DAN PEMBAHASAN

Setelah data diperoleh dan telah dilakukan analisis kebutuhan, maka proses selanjutnya adalah proses desain atau pemodelan basis data dengan pendekatan berorientasi objek.

4.1 Pemetaan Kelas

Berdasarkan hasil analisis, dilakukan pemetaan kelas yang akan digunakan pada perancangan model objek. Daftar kelas yang dipetakan terlihat pada tabel 1.

Tabel 1. Pemetaan Kelas

No	Nama Kelas	Keterangan
1	Tempat_ibadah	menggambarkan <i>property</i> yang dimiliki oleh tempat ibadah
2	Kategori	Menggambarkan <i>property</i> dari kategori tempat ibadah
3	Detail	Informasi detail tempat ibadah

Berdasarkan pada tabel 1, maka kelas yang telah dipetakan akan didesain seperti berikut ini.



Gambar 3. Desain Pemetaan Kelas

4.2 Pemetaan Atribut

Dari hasil pemetaan kelas, selanjutnya masing-masing kelas akan diisi dengan atribut-atribut untuk menjelaskan lebih detail tentang sifat atau nilai dari kelas seperti pada tabel-tabel berikut ini.

Tabel 2. Pemetaan Atribut Kelas Tempat Ibadah

No	Nama Atribut	Nilai Atribut	Keterangan
1	Id	Int	Indikator tempat ibadah
2	Nama	Varchar	Nama tempat ibadah
3	Agama	Varchar	Agama pengunjung tempat ibadah
4	Latitude	Varchar	Nilai <i>latitude</i> tempat ibadah
5	Longitude	Varchar	Nilai <i>longitude</i> tempat ibadah

Hasil dari pemetaan atribut pada kelas Tempat_ibadah dapat digambarkan seperti berikut.



Gambar 4. Pemberikan Atribut Pada Kelas Tempat_ibadah

Tabel 3. Pemetaan Atribut Kelas Katergori

No	Nama Atribut	Nilai	Keterangan
1	Id	Int	Indikator kategori tempat ibadah
2	Nama	Varchar	Nama kategori tempat ibadah

		Atribut	
1	Id	Int	Indikator kategori tempat ibadah
2	Nama	Varchar	Nama kategori tempat ibadah

Hasil dari pemetaan atribut pada kelas Kategori dapat digambarkan seperti berikut.



Gambar 5. Pemberikan Atribut Pada Kelas Kategori

Tabel 4. Pemetaan Atribut Kelas Detail

No	Nama Atribut	Nilai Atribut	Keterangan
1	Id	Int	Indikator detail tempat ibadah
2	Alamat	Text	Alamat tempat ibadah
3	Kelurahan	Varchar	Nama kelurahan tempat ibadah
4	Kecamatan	Varchar	Nama kecamatan tempat ibadah
5	Fasilitas	Text	Vasilitas yang ada ditempat ibadah
6	Keterangan	Text	Informasi tempat ibadah lainnya
7	Foto1	Varchar	Foto tempat ibadah pertama
8	Foto2	Varchar	Foto tempat ibadah kedua

Hasil dari pemetaan atribut pada kelas Detail dapat digambarkan seperti berikut.



Gambar 6. Pemberikan Atribut Pada Kelas Detail

4.3 Pemetaan Metode/operasi

Setelah kelas yang telah ditentukan diberikan atribut, berikutnya kelas akan diberikan metode berikut dengan sifatnya seperti pada tabel berikut ini.

Tabel 5. Pemetaan Metode Kelas Tempat Ibadah

No	Nama Metode	Sifat	Keterangan
1	GetKategori()	+ (Public)	Memanggil atau menampilkan data kategori pada kelas kategori
2	AddData()	# (Protected)	Melakukan penambahan data tempat ibadah
3	UpdateData()	# (Protected)	Melakukan proses perubahan data tempat ibadah
4	DeleteData()	# (Protected)	Menghapus data tempat ibadah
5	GetData ()	+ (Public)	Memanggil atau menampilkan data tempat ibadah
6	GetDataSearch()	# (Protected)	Menampilkan data tempat ibadah berdasarkan hasil pencarian nama tempat ibadah
7	GetDataFilter()	# (Protected)	Menampilkan data tempat ibadah berdasarkan filter data
8	GetDetail()	+ (Public)	Memanggil atau menampilkan detail tempat ibadah pada kelas detail

Hasil pemetaan metode terhadap kelas Tempat_ibadah terlihat seperti pada gambar berikut ini.

Tempat_ibadah

Id : Int
Nama : Varchar
Agama : Varchar
Latitude : Varchar
Longitude : Varchar

+ GetKategori()
#AddData()
#UpdateData()
#DeleteData()
+GetData()
#GetDataSearch()
#GetDataFilter()
+GetDetail()

Gambar 7. Pemberikan Metode Pada Kelas Tempat_ibadah

Tabel 6. Pemetaan Metode Kelas Kategori

No	Nama Metode	Sifat	Keterangan
1	GetKategori()	+ (Public)	Memanggil / menampilkan data kategori
2	AddData()	# (Protected)	Melakukan penambahan data kategori tempat ibadah
3	UpdateData()	# (Protected)	Melakukan proses perubahan data kategori tempat ibadah
4	DeleteData()	# (Protected)	Menghapus data kategori tempat ibadah

Hasil pemetaan metode terhadap kelas Kategori terlihat seperti pada gambar berikut ini.

Kategori

Id : Int
Nama : Varchar

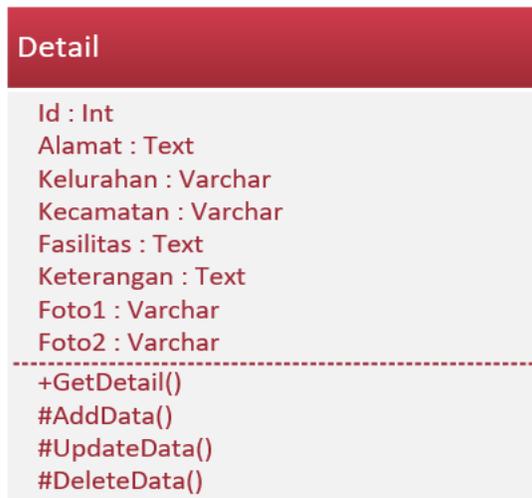
+ GetKategori()
AddData()
#UpadateData()
#DeleteData()

Gambar 8. Pemberikan Metode Pada Kelas Kategori

Tabel 7. Pemetaan Metode Kelas Detail

No	Nama Metode	Sifat	Keterangan
1	GetDetail ()	+ (Public)	Memanggil atau menampilkan data detail tepat ibadah
2	AddData()	# (Protected)	Melakukan penambahan data detail tempat ibadah
3	UpdateData()	# (Protected)	Melakukan proses perubahan data detail tempat ibadah
4	DeleteData()	# (Protected)	Menghapus data detail tempat ibadah

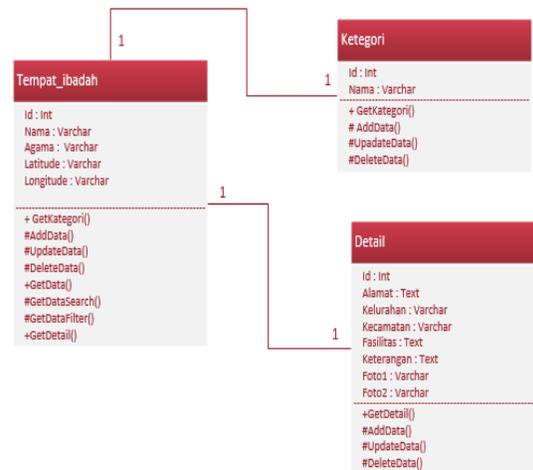
Hasil pemetaan metode terhadap kelas Detail terlihat seperti pada gambar berikut ini.



Gambar 9. Pemberikan Metode Pada Kelas Detail

4.4 Model Objek

Setelah kelas diberikan atribut dan metode, kemudian dilakukan integrasi dan implementasi dalam pembuatan basis data yang disesuaikan dengan kelas-kelas yang telah ditentukan. Maka akan terlihat gambaran hubungan antar kelas yang terjadi seperti gambar berikut ini.



Gambar 10. Model Objek (Class Diagram)

5. KESIMPULAN

Pemodelan basis data dengan pendekatan model data berorientasi objek pada *native apps* lokasi tempat ibadah di Kota Palangka Raya dilakukan dengan melakukan analisa masalah kedalam sebuah objek, kemudian objek-objek tersebut akan dipecah menjadi beberapa kelas yang menunjukkan karakteristik kelas-kelas masing-masing.

Pemodelan basis data dengan pendekatan berbasis objek dibuat dengan tujuan memudahkan pada saat pengembangan perangkat lunak, yang dikhususkan pula berorientasi objek. Selain itu, dengan menerapkan pemodelan ini memungkinkan memudahkan dalam tahap analisis, desain, implementasi dan pengujian perangkat lunak dalam hal ini *native apps*.

Desain pemodelan basis data ini dibuat untuk mempermudah dalam menganalisis kebutuhan fungsional dari *native apps* lokasi tempat ibadah di Kota Palangka Raya. Sehingga, diharapkan dengan hasil pemodelan ini *native apps* yang dikembangkan mampu memberikan informasi yang akurat dan mempunyai kualitas informasi yang baik pula agar bermanfaat bagi pengguna.

DAFTAR PUSTAKA

- [1] Irawan, Rio. "Implementasi Location Based Services (lbs) Kantor Pemerintahan Kota (studi Kasus : Kota Palangka Raya, Provinsi Kalimantan Tengah, Indonesia)", FAHMA Journal Teknologi Informasi dan Ilmu Komputer, vol. 14, no. 2, 2016, pp. 35-56.
- [2] Wulandari, Fera Tri. "Pemodelan Basis Data Akademik Universitas XYZ Menggunakan Pendekatan Objek". JITU : Journal Informatic Technology And Communication, vol. 3, no. 1, 2019, pp. 52-57.

- [3] Umar, Rusydi, dkk, "Perancangan Databae Point of Sales Apotek Dengan Menerapkan Model Data Relasional". Query: Jurnal Sistem Informasi, vol. 03. no. 2, 2019, pp. 33-41.
- [4] Yanto, Robi, 2016. "Manajemen Basis Data Menggunakan MySQL". Yogyakarta: Deepublish.
- [5] Hariyanto, B., 2004, "Sistem Manajemen Basisdata". Bandung: Informatika.
- [6] Rodger, Richard, 2011. "Beginning Mobile Application Development in the Cloud". New Jersey, United States: Wiley.
- [7] Pernama, Yana. 2016, "Ini Dia Perbedaan Aplikasi Native, Hybrid atau Web". <https://www.codepolitan.com/apa-bedanya-aplikasi-native-hybrid-dan-web>, on 3 April 2020.
- [8] Mulyani, Sri. 2016. "Metode Analisis dan Perancangan Sistem". Bandung: Abdi Sistematika