

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan analisis, desain dan implementasi perangkat lunak dan pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan yaitu:

1. Sistem Informasi Geografis Pemerataan Penyebaran Guru di Yogyakarta telah berhasil memberikan informasi pemerataan penyebaran guru.
2. Sistem Informasi Geografis Pemerataan Penyebaran Guru di Yogyakarta mempermudah penyampaian dan berbagi informasi pengguna mengenai pemerataan penyebaran guru karena menggunakan dapat menampilkan grafik statistik pemerataan guru di D.I Yogyakarta.

6.2 Saran

Saran yang dapat diambil dari proses analisis sampai pada pembuatan SIGPEGY di tugas akhir ini adalah sebagai berikut:

1. Sistem diharapkan bisa dikembangkan lebih lanjut dengan fungsionalitas yang baik yang dapat membuat sistem menjadi lebih baik, misalnya mampu menampilkan grafik pemerataan ketika sebuah item pada peta Google Maps di-klik.
2. Sistem diharapkan dapat dikembangkan lagi bukan hanya pada guru di provinsi D.I Yogyakarta. Hal ini dapat dilakukan dengan memanipulasi bagian Google Maps sehingga dapat memperluas cakupan daerah pemerataan penyebaran guru

3. Sistem diharapkan bisa dikembangkan lebih lanjut agar bisa membedakan status guru (honorer / pegawai negeri / swasta).
4. Untuk pengembangan aplikasi SIGPEGY selanjutnya disarankan menggunakan AJAX untuk membuat aplikasi semakin efisien, misalnya ketika sebuah *combobox* dipilih maka tidak perlu melakukan muat ulang halaman.
5. Sistem diharapkan dapat mengambil data kabupaten dan kecamatan sebuah sekolah secara otomatis dengan menggunakan *code* Google Maps.

DAFTAR PUSTAKA

- Aini, Anisah., 2007, *Sistem Informasi Geografis Pengertian Dan Aplikasinya*.
- Amarullz, 2010, *Mengenal Google Map API*, <http://amarullz.blog.unikom.ac.id/mengenal-google-map.b1>, Juli, 2011
- Avison D., Fitzgerald G., 2006, *Information Systems Development: Methodologies, Techniques and Tools 4th edition*, McGraw-Hill, New York
- Fanani, Herjuno Hamzah., Basofi, Arif., Arifin, Firman. 2009, *Sistem Informasi Geografis Untuk Pemetaan Alumni Pens-Its*.
- Paryono, Petrus, 1994, *Sistem Informasi Geografis*, Penerbit Andi, Yogyakarta
- Prahasta, Eddy, 2002, *Sistem Informasi Geografis Konsep-konsep Dasar*, Penerbit Informatika, Bandung
- Purnomo, Agung ., Andreas, 2009, *Pengembangan Sistem Informasi Geografis Posyandu Bebrbasis Web*, Jurusan Teknik Informatika Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta, Yogyakarta
- Robert A. LEITCH, K. Roscoe DAVIS, 1992, *Accounting Information Systems : Theory and Practice / 2nd .ed*, New Jersey : Prentice-Hall, New Jersey.
- Saor, Marganda Romel., 2006, *Peta Zonasi Longsoran dengan Metode Anbalagan dan Sistem Informasi Geografi; Studi Kasus di Sub Daerah Aliran Sungai Citarum, Kecamatan Cipongkor, Kabupaten Bandung, Jawa Barat*, Laboratory of Engineering Geology

Department of Geological Engineering Faculty of Earth Sciences and Technology (FITB).

Susilana, Rudi., Yani, Ahmad., 2006, *Model Pengembangan Program Pendidikan Berdasarkan Kajian Potensi Daerah dalam Rangka Peningkatan Mutu Pendidikan Dasar*, Universitas Pendidikan Indonesia.

Suwarna, Ketut, 1997, *Peningkatan Efisiensi Pendidikan Ditinjau dari Distribusi dan Mutu Guru Sekolah Dasar di Daerah Tingkat I Propinsi Bali*, Aneka Widya STKIP Singaraja, No. 4.

Thuan L. Thai, Hoang Q. Lam, *.NET Framework Essentials*, O'Reilly & Associates, 2003.

Winarto, Ary Bima., Fariza, Arna., Setiowati, Yuliana., 2007, *Investigasi Daerah Rawan Banjir di Kota Surabaya dengan Menggunakan Metode Fuzzy*.

Yani, Ahmad., M. Nur, Djakaria., 2009, *Pengembangan Model Sistem Informasi Geografis (SIG) untuk Pengelolaan Pendidikan dalam Era Otonomi Daerah (Studi) Pengembangan di Kabupaten Sukabumi*, Universitas Pendidikan Indonesia.

Yaqin, 2010, *Sistem Informasi Berbasis Web*, <http://yaqinov.wordpress.com/2011/01/10/sistem-informasi-berbasis-web/>, 19 Juni 2011.

SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

SIGPEGY

(Sistem Informasi Geografis Pemerataan
Penyebaran Guru di Yogyakarta)

Untuk :

Department Pendidikan Nasional

Dipersiapkan oleh:

Sabrina Editha Putri / 07 07 05439

**Program Studi Teknik Informatika – Fakultas Teknologi
Industri
Universitas Atma Jaya Yogyakarta**

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL - SIGPEGY		1/38
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperik sa oleh								
Disetuj ui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan.....	6
1.1	Tujuan.....	6
1.2	Lingkup Masalah.....	6
1.3	Definisi, Akronim dan Singkatan.....	7
1.4	Referensi.....	7
1.5	Deskripsi umum (Overview).....	8
2	Deskripsi Kebutuhan.....	9
2.1	Perspektif produk.....	9
2.2	Fungsi Produk.....	10
2.3	Karakteristik Pengguna.....	14
2.4	Batasan-batasan.....	15
2.5	Asumsi dan Ketergantungan.....	15
3	Kebutuhan khusus.....	15
3.1	Kebutuhan antarmuka eksternal.....	15
3.2	Kebutuhan fungsionalitas Perangkat Lunak.....	17
4	Spesifikasi Rinci Kebutuhan.....	18
4.1	Spesifikasi Kebutuhan Fungsionalitas.....	18
5	Entity Relationship Diagram (ERD).....	36

Daftar Gambar

1. Arsitektur Perangkat Lunak SIGPEGY	10
2. Use Case Diagram	17
3. ERD	38



Pendahuluan

Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak Sistem Informasi Geografis Pemerataan Penyebaran Guru di Yogyakarta (SIGPEGY) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna), performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (feature-feature tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SIGPEGY ini juga mendefinisikan batasan perancangan perangkat lunak.

Dokumen SKPL ini menjadi dasar kesepakatan antara pihak pengembang mengenai perangkat lunak yang akan dikembangkan.

Lingkup Masalah

Perangkat Lunak SIGPEGY dikembangkan dengan tujuan untuk :

4. Menampilkan informasi peta pemerataan penyebaran guru di wilayah Yogyakarta.
5. Menampilkan informasi detail pemerataan penyebaran guru bagi admin dan pengguna.
6. Mempermudah proses pengelolaan data guru oleh admin.

Dan berjalan pada lingkungan dengan *platform* Windows XP.

Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SIGPEGY	Sistem Informasi Geografis Pemerataan Penyebaran Guru di Yogyakarta.
SKPL-SIGPEGY-Y-XXX-ZZ	Kode yang merepresentasikan kebutuhan pada SIGPEGY (Sistem Informasi Geografis Penyebaran Guru di Yogyakarta) dimana Y merupakan nomor fungsi produk dan XXX merupakan nomor sub fungsi produk dan ZZ merupakan nomor sub-sub fungsi produk
Admin	Orang yang bertanggung jawab untuk mengatur pengelolaan data dalam suatu sistem.
LAN	Local Area Network

Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

2. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
3. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.
4. Hartanto, Budi, *Memahami Visual C#.Net Secara Mudah*, Andi Offset, 2008.
5. Deitel, C# How to Program, Prentice-Hall Inc, 2002.
6. MSDN Library-October 2005, Microsoft, 2005.

7. Johannis, Yohanni Eveline, 2008, *Pembangunan Sistem Informasi Geografis Daerah Rawan Banjir Berbasis Web*, Program Studi Teknik Informatika, Universitas Atma Jaya, Yogyakarta.

Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 5 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SIGPEGY yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SIGPEGY tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SIGPEGY yang akan dikembangkan, yang mencakup kebutuhan antarmuka eksternal dan kebutuhan fungsionalitas perangkat lunak beserta use case diagram dari kebutuhan fungsionalitas perangkat lunak ini.

Bagian keempat berisi penjelasan mengenai spesifikasi rinci kebutuhan, mencakup spesifikasi kebutuhan fungsionalitas yang terbagi menjadi beberapa use case specification sesuai dengan use case diagram yang ada pada bagian ketiga.

Bagian kelima berisi ERD (Entitas relationship Diagram) yang menggambarkan hubungan antar entitas.

Deskripsi Kebutuhan

Perspektif produk

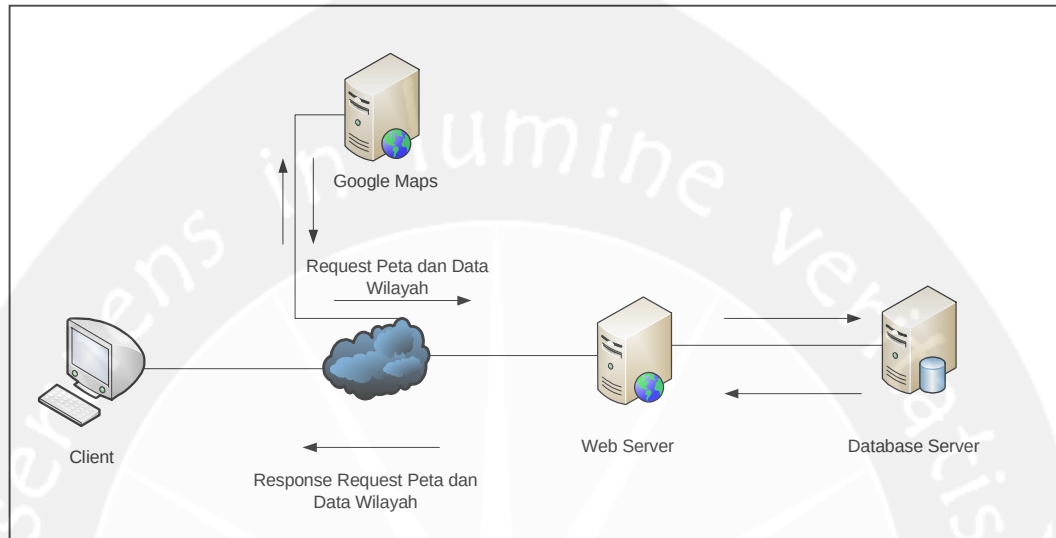
Berikut ini akan dijelaskan mengenai perspektif produk, yaitu kegunaan dari perangkat lunak SIGPEGY yang dibuat, cara kerja, serta arsitektur yang digunakan.

SIGPEGY adalah perangkat lunak yang menyediakan informasi mengenai pemerataan penyebaran guru yang berada di wilayah Yogyakarta. Informasi yang disajikan dalam bentuk peta daerah pemerataan penyebaran guru serta dalam bentuk tabel informasi penyebaran guru. Selain itu SIGPEGY juga mempunyai fasilitas untuk melakukan pengelolaan user serta peng-*edit*-an penyebaran guru.

Perangkat lunak SIGPEGY ini berjalan pada *platform* Windows XP untuk komputer PC atau notebook, yang memiliki *internet browser* dan terhubung secara *on-line* ke *internet*. Bahasa pemrograman yang digunakan adalah ASP.NET. Sedangkan untuk lingkungan pemrogramannya menggunakan Microsoft Visual Studio. NET 2005, peta menggunakan *google maps* API dan menggunakan *database* SQL Server 2005.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*). Pada sistem ini, seperti terlihat pada gambar 1, arsitektur perangkat lunak yang digunakan berupa aplikasi *client server*, di mana semua data disimpan di *server*.

Inputan data yang dimasukkan akan disimpan dalam database server, sehingga jika ada pencarian data, maka data yang diinginkan akan dicari ke database server yang selanjutnya dikirimkan ke client yang merequest.



Gambar 1. Arsitektur Perangkat lunak SIGPEGY

Fungsi Produk

Perangkat lunak SIGPEGY memiliki beberapa fungsi dalam aplikasinya guna untuk kelancaran jalannya perangkat lunak. Fungsi produk perangkat lunak SIGPEGY adalah sebagai berikut :

Pengguna Admin dan Sekolah :

1. Fungsi Login (*SKPL-SIGPEGY-1-001*)

Merupakan fungsi untuk login ke dalam sistem, dimana admin memiliki hak akses yang berbeda dengan pengguna biasa.

3. Fungsi Kelola Guru(*SKPL-SIGPEGY-1-002*)

Merupakan fungsi yang digunakan untuk mengelola data guru.

Fungsi pengelolaan data sekolah meliputi:

1. Fungsi *Entry Data Guru* (**SKPL-SIGPEGY-1-002-01**).

Merupakan fungsi yang digunakan untuk menambahkan data guru baru.

2. Fungsi *Edit Data Guru* (**SKPL-SIGPEGY-1-002-02**).

Merupakan fungsi yang digunakan untuk mengubah data guru.

3. Fungsi *Search Data Guru* (**SKPL-SIGPEGY-1-002-03**).

Merupakan fungsi untuk mencari data guru berdasarkan kata kunci kemudian menampilkannya.

4. Fungsi *Delete Data Guru* (**SKPL-SIGPEGY-002-1-04**)

Merupakan fungsi yang digunakan untuk menghapus data guru.

Pengguna Admin :

4. Fungsi *Kelola Anggota* (**SKPL-SIGPEGY-2-001**)

Merupakan fungsi yang digunakan untuk melakukan pengelolaan terhadap data anggota.

Fungsi Pengelolaan Data Anggota meliputi:

e. Fungsi *Entry Data Anggota* (**SKPL-SIGPEGY-2-001-01**).

Merupakan fungsi yang digunakan untuk menambahkan data anggota baru.

f. Fungsi *Edit Data Anggota* (**SKPL-SIGPEGY-2-001-02**).

Merupakan fungsi yang digunakan untuk mengubah data anggota.

- g. Fungsi *Search Data Anggota* (**SKPL-SIGPEGY-2-001-03**).

Merupakan fungsi untuk mencari data anggota berdasarkan kata kunci kemudian menampilkannya.

- h. Fungsi *Delete Data Anggota* (**SKPL-SIGPEGY-2-001-04**)

Merupakan fungsi yang digunakan untuk menghapus data anggota.

5. Fungsi *Display Request Guru* (**SKPL-SIGPEGY-2-002**)

Merupakan fungsi yang digunakan untuk menampilkan request terhadap guru.

6. Fungsi *Kelola Sekolah* (**SKPL-SIGPEGY-2-003**)

Merupakan fungsi yang digunakan untuk mengelola data sekolah.

Fungsi pengelolaan data sekolah meliputi:

- e. Fungsi *Entry Data Sekolah* (**SKPL-SIGPEGY-2-003-01**).

Merupakan fungsi yang digunakan untuk menambahkan data sekolah baru.

- f. Fungsi *Edit Data Sekolah* (**SKPL-SIGPEGY-2-003-02**).

Merupakan fungsi yang digunakan untuk mengubah data sekolah.

- g. Fungsi *Search Data Sekolah* (**SKPL-SIGPEGY-2-003-03**).

Merupakan fungsi untuk mencari data sekolah berdasarkan kata kunci kemudian menampilkannya.

h. Fungsi *Delete Data Sekolah* (**SKPL-SIGPEGY-2-003-04**)

Merupakan fungsi yang digunakan untuk menghapus data sekolah.

Admin, Sekolah dan Pengunjung :

2. Fungsi *Display Data Pemerataan Penyebaran Guru* (**SKPL-SIGPEGY-3-001**)

Merupakan fungsi yang digunakan untuk mengelola seluruh data pemerataan penyebaran guru.

Fungsi pengelolaan data pemerataan penyebaran guru meliputi:

e. Fungsi *Display Peta Pemerataan Penyebaran Guru* (**SKPL-SIGPEGY-3-001-01**)

Merupakan fungsi yang digunakan untuk menampilkan informasi penyebaran guru melalui peta Google Maps API.

f. Fungsi *Display Grafik Pemerataan Penyebaran Guru* (**SKPL-SIGPEGY-3-001-02**)

Merupakan fungsi yang digunakan untuk menampilkan informasi pemerataan penyebaran guru melalui grafik.

g. Fungsi *Display Sekolah* (**SKPL-SIGPEGY-3-001-03**)

Merupakan fungsi yang digunakan untuk menampilkan data dan informasi sekolah.

h. Fungsi *Display Guru* (**SKPL-SIGPEGY-3-001-04**)

Merupakan fungsi yang digunakan untuk menampilkan data dan informasi guru.

Sekolah :

2. Fungsi Kelola Request Guru (**SKPL-SIGPEGY-4-001**)

Merupakan fungsi yang digunakan untuk mengelola request guru.

Fungsi Pengelolaan Request Guru meliputi:

e. Fungsi *Entry Request* (**SKPL-SIGPEGY-4-001-01**)

Merupakan fungsi yang digunakan untuk menambah request guru terhadap sebuah sekolah.

f. Fungsi *Display Request* (**SKPL-SIGPEGY-4-001-02**)

Merupakan fungsi yang digunakan untuk menampilkan request.

g. Fungsi *Edit Request* (**SKPL-SIGPEGY-4-001-03**)

Merupakan fungsi yang digunakan untuk mengubah request.

h. Fungsi *Delete Request* (**SKPL-SIGPEGY-4-001-04**)

Merupakan fungsi yang digunakan untuk menghapus data request.

Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SIGPEGY adalah sebagai berikut :

1. Admin

a. Memahami pengoperasian sistem operasi windows serta pernah menggunakan aplikasi yang berbasis web.

b. Mengerti dan memahami cara kerja perangkat lunak yang digunakan.

c. Mengerti tentang internet.

d. Memahami proses pengelolaan data perangkat lunak.

2. User dan Sekolah

- a. Mengerti pengoperasian komputer.
- b. Biasa menggunakan aplikasi berbasis web (internet).

Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SIGPEGY tersebut adalah :

1. Kebijaksanaan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SIGPEGY.

2. Keterbatasan perangkat keras

Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

Asumsi dan Ketergantungan

Kebutuhan dari SIGPEGY ini yaitu yang sifatnya non-fungsionalitas adalah :

- a. Tersedia perangkat lunak yang sesuai dengan kebutuhan untuk mengoperasikan produk perangkat lunak SIGDarBan.
- b. Tersedianya web browser untuk akses internet

Kebutuhan khusus

Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak SIGPEGY meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk antarmuka berupa halaman web.

Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SIGPEGY adalah:

3. Perangkat desktop (monitor, motherboard, keyboard, mouse, dll).
4. Modem atau LAN Card.

Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SIGPEGY adalah sebagai berikut :

5. Nama : Windows XP
Sumber : Microsoft

Sebagai sistem operasi untuk perangkat komputer.

6. Nama : Microsoft SQL Server
Nomor Versi : 2005
Sumber : Microsoft

Sebagai database management system (DBMS) yang dibutuhkan untuk menyimpan data SIGPEGY di sisi server.

7. Nama : Google Chrome
Sumber : Google

Sebagai internet browser dimana perangkat lunak SIGPEGY dijalankan.

8. Nama : IIS
Sumber : Microsoft

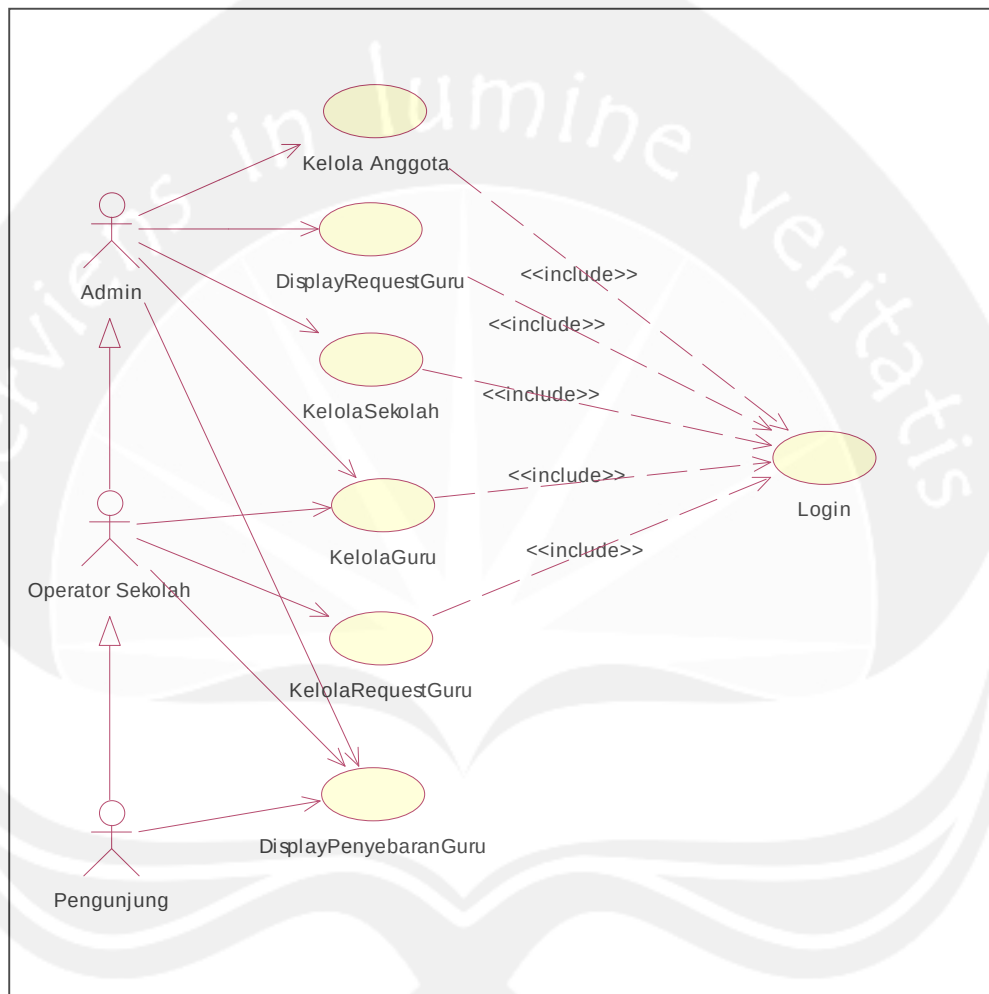
Sebagai *web server*.

Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SIGPEGY menggunakan localhost.

Kebutuhan fungsionalitas Perangkat Lunak

Use Case Diagram



Gambar 2. Use Case Diagram

Spesifikasi Rinci Kebutuhan

Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Spesification : Login

9. Brief Description

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada username dari user dan password yang berupa rangkaian karakter.

10. Primary Actor

1. Admin
2. Sekolah

11. Supporting Actor

none

12. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.
3. Aktor memasukkan username/NUPTK dan password.
4. Sistem memeriksa username dan password yang diinputkan aktor.

E-1 Password atau username/NUPTK tidak sesuai

5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

13. Alternative Flow

none

14. Error Flow

E-1 Password atau username tidak sesuai

1. Sistem menampilkan peringatan bahwa username atau password tidak sesuai.
2. Kembali ke Basic Flow langkah ke 3.

15. PreConditions

none

16. PostConditions

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.

4.1.2 Use case Spesification : Kelola Guru

7. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data guru. Aktor dapat melakukan entry data user, edit data user, delete data guru dan search data guru.

8. Primary Actor

1. Admin

9. Supporting Actor

none

10. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola data guru.
2. Sistem memberikan pilihan untuk melakukan entry data guru, edit data guru, delete data guru atau search data guru.
3. Aktor memilih untuk melakukan entry data guru.
 - A-1 Aktor memilih untuk melakukan edit data guru
 - A-2 Aktor memilih untuk melakukan delete data guru
 - A-3 Aktor memilih untuk melakukan search data guru
4. Aktor menginputkan data guru.

5. Aktor meminta sistem untuk menyimpan data guru yang telah diinputkan.

6. Sistem mengecek data guru yang telah diinputkan.

E-1 Data guru yang diinputkan aktor tidak lengkap

E-2 Data guru yang diinputkan aktor telah ada di database

7. Sistem menyimpan data guru ke database.

8. Use Case selesai.

11. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data guru

8. Sistem memasukkan kata kunci data guru yang ingin di-edit.

9. Sistem menampilkan data guru yang mengandung kata kunci.

10. Aktor mengedit data guru yang sudah ditampilkan.

11. Aktor meminta sistem untuk menyimpan data guru yang telah di-edit.

12. Sistem melakukan pengecekan terhadap data guru yang telah diedit.

E-3 Data guru yang telah diedit belum lengkap

E-4 Data guru yang telah diedit sudah ada di database

13. Sistem menyimpan data guru yang telah diedit ke database.

14. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan delete data guru

8. Aktor memasukkan kata kunci data guru yang ingin dihapus.

9. Sistem menampilkan data guru yang mengandung kata kunci.

10. Aktor menghapus data guru yang sudah ditampilkan.

11. Sistem meminta konfirmasi penghapusan data guru.

12. Aktor menyetujui konfirmasi penghapusan data guru.

A-4 Aktor menolak konfirmasi penghapusan data guru.

13. Sistem melakukan penghapusan data guru yang telah disetujui dari database.

14. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan search data guru

1. Aktor memasukkan kata kunci data guru yang ingin ditampilkan.

6. Sistem melakukan pencarian terhadap data guru.

7. Sistem melakukan pengecekan terhadap data guru yang dicari dalam database.

E-5 Data guru yang dicari tidak ditemukan

8. Sistem menampilkan data guru yang dicari pada kotak display yang ada.

9. Berlanjut ke Basic Flow langkah ke 8.

A-4 Aktor menolak konfirmasi penghapusan data guru

1. Berlanjut ke Basic Flow langkah ke 8.

12. Error Flow

E-1 Data guru yang diinputkan aktor tidak lengkap

1. Sistem memberikan pesan peringatan untuk melengkapi data terlebih dahulu.

2. Kembali ke Basic Flow Langkah ke 4.

E-2 Data guru yang diinputkan aktor telah ada di database

3. Sistem memberikan pesan peringatan bahwa data telah ada di database.

4. Kembali ke Basic Flow Langkah ke 4.

E-3 Data guru yang diedit aktor tidak lengkap

3. Sistem memberikan pesan peringatan bahwa silahkan lengkapi dahulu.

4. Kembali ke Alternative Flow A-1 Langkah ke 2.

E-4 Data guru yang diedit aktor telah ada di database

3. Sistem memberikan pesan peringatan bahwa data telah ada di database.

4. Kembali ke Alternative Flow A-1 Langkah ke 2.

E-5 Data guru yang dicari tidak ditemukan

3. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.

4. Kembali ke Alternative Flow A-3 Langkah ke 2.

9. PreConditions

9. Use Case Login telah dilakukan.

10. Aktor telah memasuki sistem dengan role sebagai admin.

10. PostConditions

1. Data penelitian di database telah terupdate.

4.1.3 Use case Spesification : Kelola Anggota

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data anggota. Aktor dapat melakukan entry data anggota, edit data anggota, delete data anggota dan search data anggota.

2. Primary Actor

1. Admin

11. Supporting Actor

none

12. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola data anggota.
2. Sistem memberikan pilihan untuk melakukan entry data anggota, edit data anggota, delete data anggota atau search data anggota.
3. Aktor memilih untuk melakukan entry data anggota.
 - A-1 Aktor memilih untuk melakukan edit data anggota
 - A-2 Aktor memilih untuk melakukan delete data anggota
 - A-3 Aktor memilih untuk melakukan search data anggota
4. Aktor menginputkan data anggota.
5. Aktor meminta sistem untuk menyimpan data anggota yang telah diinputkan.
6. Sistem mengecek data anggota yang telah diinputkan.
 - E-1 Data anggota yang diinputkan aktor tidak lengkap
 - E-2 Data anggota yang diinputkan aktor telah ada di database
7. Sistem menyimpan data anggota ke database.
8. Use Case selesai.

13. Alternative Flow

- A-1 Aktor memilih untuk melakukan edit data anggota

8. Sistem memasukkan kata kunci data anggota yang ingin di-edit.

9. Sistem menampilkan data anggota yang mengandung kata kunci.

10. Aktor mengedit data anggota yang sudah ditampilkan.

11. Aktor meminta sistem untuk menyimpan data anggota yang telah di-edit.

12. Sistem melakukan pengecekan terhadap data anggota yang telah diedit.

E-3 Data anggota yang telah diedit belum lengkap

E-4 Data anggota yang telah diedit sudah ada di database

13. Sistem menyimpan data anggota yang telah diedit ke database.

14. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan delete data anggota

8. Aktor memasukkan kata kunci data anggota yang ingin dihapus.

9. Sistem menampilkan data anggota yang mengandung kata kunci.

10. Aktor menghapus data anggota yang sudah ditampilkan.

11. Sistem meminta konfirmasi penghapusan data anggota.

12. Aktor menyetujui konfirmasi penghapusan data anggota.

A-4 Aktor menolak konfirmasi penghapusan data anggota.

13. Sistem melakukan penghapusan data anggota yang telah disetujui dari database.

14. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan search data anggota

1. Aktor memasukkan kata kunci data anggota yang ingin ditampilkan.

2. Sistem melakukan pencarian terhadap data anggota.

3. Sistem melakukan pengecekan terhadap data anggota yang dicari dalam database.

E-5 Data anggota yang dicari tidak ditemukan

4. Sistem menampilkan data anggota yang dicari pada kotak display yang ada.

5. Berlanjut ke Basic Flow langkah ke 8.

A-4 Aktor menolak konfirmasi penghapusan data anggota

1. Berlanjut ke Basic Flow langkah ke 8.

14. Error Flow

E-1 Data anggota yang diinputkan aktor tidak lengkap

3. Sistem memberikan pesan peringatan untuk melengkapi data terlebih dahulu.

4. Kembali ke Basic Flow Langkah ke 4.

E-2 Data anggota yang diinputkan aktor telah ada di database

3. Sistem memberikan pesan peringatan bahwa data telah ada di database.

4. Kembali ke Basic Flow Langkah ke 4.

E-3 Data anggota yang diedit aktor tidak lengkap

3. Sistem memberikan pesan peringatan bahwa silahkan lengkapi dahulu.

4. Kembali ke Alternative Flow A-1 Langkah ke 2.

E-4 Data anggota yang diedit aktor telah ada di database

3. Sistem memberikan pesan peringatan bahwa data telah ada di database.

4. Kembali ke Alternative Flow A-1 Langkah ke 2.

E-5 Data anggota yang dicari tidak ditemukan

3. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.

4. Kembali ke Alternative Flow A-3 Langkah ke 2.

15. PreConditions

3. Use Case Login telah dilakukan.

4. Aktor telah memasuki sistem dengan role sebagai admin.

16. PostConditions

1. Data penelitian di database telah terupdate.

4.1.4 Use case Spesification : Display Request Guru

7. Brief Description

Use Case ini digunakan oleh aktor untuk kelola data request guru. Aktor dapat melakukan display data request guru.

8. Primary Actor

1. Admin

9. Supporting Actor

none

10. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan kelola request guru.

2. Sistem memberikan pilihan untuk melakukan display request guru.

3. Aktor memilih untuk melakukan display request guru.

4. Sistem menampilkan data request guru.

5. Use Case selesai.

11. Alternative Flow

-

12. Error Flow

-

9. PreConditions

3. Use Case Login telah dilakukan.

4. Aktor telah memasuki sistem dengan role sebagai Admin.

10. PostConditions

-

4.1.5 Use case Spesification : Kelola Sekolah

1. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data sekolah. Aktor dapat melakukan entry data sekolah, edit data sekolah, delete data sekolah dan search data sekolah.

2. Primary Actor

1. Admin

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data sekolah.

2. Sistem memberikan pilihan untuk melakukan entry data sekolah, edit data sekolah, delete data sekolah atau search data sekolah.

3. Aktor memilih untuk melakukan entry data sekolah.

A-1 Aktor memilih untuk melakukan edit data sekolah

A-2 Aktor memilih untuk melakukan delete data sekolah

A-3 Aktor memilih untuk melakukan search data sekolah

4. Aktor menginputkan data sekolah.

5. Aktor meminta sistem untuk menyimpan data sekolah yang telah diinputkan.

6. Sistem mengecek data sekolah yang telah diinputkan.

E-1 Data sekolah yang diinputkan aktor tidak lengkap

E-2 Data sekolah yang diinputkan aktor telah ada di database

7. Sistem menyimpan data sekolah ke database.

8. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan edit data sekolah

8. Sistem memasukkan kata kunci data sekolah yang ingin di-edit.

9. Sistem menampilkan data sekolah yang mengandung kata kunci.

10. Aktor mengedit data sekolah yang sudah ditampilkan.

11. Aktor meminta sistem untuk menyimpan data sekolah yang telah di-edit.

12. Sistem melakukan pengecekan terhadap data sekolah yang telah diedit.

E-3 Data sekolah yang telah diedit belum lengkap

E-4 Data sekolah yang telah diedit sudah ada di database

13.Sistem menyimpan data sekolah yang telah diedit ke database.

14.Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan delete data sekolah

9. Aktor memasukkan kata kunci data sekolah yang ingin dihapus.

10.Sistem menampilkan data sekolah yang mengandung kata kunci.

11.Aktor menghapus data sekolah yang sudah ditampilkan.

12.Sistem meminta konfirmasi penghapusan data sekolah.

13.Aktor menyetujui konfirmasi penghapusan data sekolah.

A-4 Aktor menolak konfirmasi penghapusan data sekolah.

14.Sistem melakukan penghapusan data sekolah yang telah disetujui dari database.

15.Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan search data sekolah

1. Aktor memasukkan kata kunci data sekolah yang ingin ditampilkan.

2. Sistem melakukan pencarian terhadap data sekolah.

3. Sistem melakukan pengecekan terhadap data sekolah yang dicari dalam database.

E-5 Data sekolah yang dicari tidak ditemukan

4. Sistem menampilkan data sekolah yang dicari pada kotak display yang ada.

5. Berlanjut ke Basic Flow langkah ke 8.

A-4 Aktor menolak konfirmasi penghapusan data sekolah

1. Berlanjut ke Basic Flow langkah ke 8.

6. Error Flow

E-1 Data sekolah yang diinputkan aktor tidak lengkap

3. Sistem memberikan pesan peringatan untuk melengkapi data terlebih dahulu.

4. Kembali ke Basic Flow Langkah ke 4.

E-2 Data sekolah yang diinputkan aktor telah ada di database

3. Sistem memberikan pesan peringatan bahwa data telah ada di database.

4. Kembali ke Basic Flow Langkah ke 4.

E-3 Data sekolah yang diedit aktor tidak lengkap

3. Sistem memberikan pesan peringatan bahwa silahkan lengkapi dahulu.

4. Kembali ke Alternative Flow A-1 Langkah ke 2.

E-4 Data sekolah yang diedit aktor telah ada di database

3. Sistem memberikan pesan peringatan bahwa data telah ada di database.

4. Kembali ke Alternative Flow A-1 Langkah ke 2.

E-5 Data sekolah yang dicari tidak ditemukan

3. Sistem memberikan pesan peringatan bahwa data tidak ditemukan.

4. Kembali ke Alternative Flow A-3 Langkah ke 2.

7. PreConditions

1. Use Case Login telah dilakukan.

3. Aktor telah memasuki sistem dengan role sebagai admin.

16. PostConditions

1. Data sekolah di database telah terupdate.

4.1.6 Use case Spesification : Display Data Pemerataan Penyebaran Guru

7. Brief Description

Use Case ini digunakan oleh aktor untuk menampilkan peta dan informasi detail penyebaran guru di Yogyakarta.

8. Primary Actor

1. Admin
2. Sekolah
3. Pengunjung

9. Supporting Actor

none

10. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan display pemerataan penyebaran guru.
2. Sistem menampilkan antarmuka display penyebaran guru.
3. Sistem menampilkan antarmuka untuk memilih display peta data (sekolah & guru) atau grafik pemerataan penyebaran guru.
4. Aktor memilih untuk menampilkan peta penyebaran guru.
A-1 Aktor memilih untuk menampilkan grafik pemerataan penyebaran guru
5. Aktor memilih salah satu sekolah pada peta.

6. Sistem akan menampilkan informasi detail sekolah tersebut dalam bentuk tabel beserta informasi guru.

7. Use Case selesai.

11. Alternative Flow

A-1 Aktor memilih untuk menampilkan grafik pemerataan penyebaran guru.

4. Aktor memilih wilayah atau kabupaten untuk identifikasi pemerataan penyebaran guru melalui grafik.

5. Aktor memilih salah satu sekolah pada peta.

6. Sistem akan menampilkan informasi detail pemerataan penyebaran guru di wilayah atau kabupaten tersebut dalam bentuk grafik.

12. Error Flow

-

9. PreConditions

-

10. PostConditions

2. Peta serta informasi detail pemerataan penyebaran guru.

4.1.7 Use case Spesification : Pengelolaan Request Guru

7. Brief Description

Use Case ini digunakan oleh aktor untuk mengelola data request guru. Aktor dapat melakukan add, edit, dan delete request guru.

8. Primary Actor

1. Sekolah

9. Supporting Actor

none

10. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan request guru.
2. Sistem memberikan pilihan untuk melakukan add, edit, dan delete request guru.
3. Aktor memilih untuk melakukan add request guru
 - A-1 Aktor memilih untuk melakukan edit request guru.
 - A-2 Aktor memilih untuk melakukan delete data request guru.
4. Aktor menginputkan data request guru.
5. Aktor meminta sistem untuk menyimpan data sekolah yang telah diinputkan.
6. Sistem mengecek data request guru yang telah diinputkan.
 - E-1 Data request guru yang diinputkan aktor tidak lengkap
7. Sistem menyimpan data request guru ke database.
8. Use case selesai.

11. Alternative Flow

- A-1 Aktor memilih untuk melakukan edit request guru
 8. Sistem memasukkan kata kunci data request guru yang ingin di-edit.
 9. Sistem menampilkan data request guru yang mengandung kata kunci.
 10. Aktor mengedit data request guru yang sudah ditampilkan.
 11. Aktor meminta sistem untuk menyimpan data request guru yang telah di-edit.

12. Sistem melakukan pengecekan terhadap data request guru yang telah diedit.

E-2 Data request yang telah diedit belum lengkap

E-3 Data request yang telah diedit sudah ada di database

13. Sistem menyimpan data request guru yang telah diedit ke database.

14. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan delete data request guru

8. Aktor memasukkan kata kunci data request guru yang ingin dihapus.

9. Sistem menampilkan data request guru yang mengandung kata kunci.

10. Aktor menghapus data request guru yang sudah ditampilkan.

11. Sistem meminta konfirmasi penghapusan data request guru.

12. Aktor menyetujui konfirmasi penghapusan data request guru.

A-3 Aktor menolak konfirmasi penghapusan data request.

13. Sistem melakukan penghapusan data request guru yang telah disetujui dari database.

14. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor menolak konfirmasi penghapusan data request guru.

2. Berlanjut ke Basic Flow langkah ke 8.

12. Error Flow

E-1 Data request guru yang diinputkan belum lengkap

1. Sistem memberikan pesan peringatan bahwa silahkan lengkapi dahulu.

2. Kembali ke Basic Flow Langkah ke 4.

E-2 Data request guru yang telah diedit belum lengkap

3. Sistem memberikan pesan peringatan bahwa silahkan lengkapi dahulu.

4. Kembali ke Basic Flow Langkah ke 3.

E-3 Data request guru yang telah diedit sudah ada di database

3. Sistem memberikan pesan peringatan bahwa data sudah ada di database.

4. Kembali ke Basic Flow Langkah ke 3.

9. PreConditions

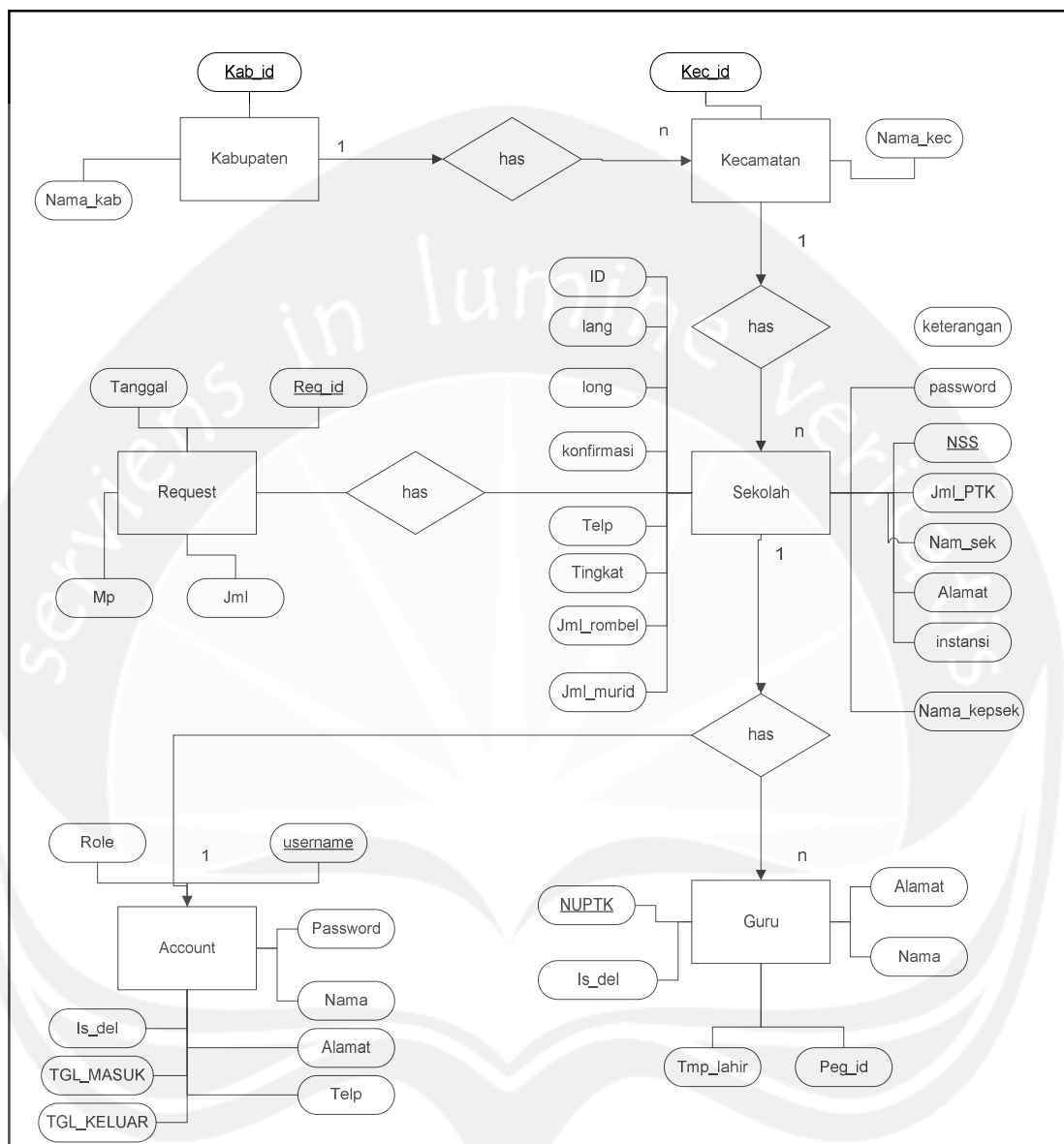
3. Use Case Login telah dilakukan.

4. Aktor telah memasuki sistem dengan role sebagai sekolah.

10. PostConditions

2. Data request guru di database telah terupdate.

Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram SIGPEGY

Gambar 3 memperlihatkan tentang relasi antara tabel - tabel pada database yang digunakan pada perangkat lunak "SIGPEGY".

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SIGPEGY

(Sistem Informasi Geografis Pemerataan
Penyebaran Guru di Yogyakarta)

Untuk :

Department Pendidikan Nasional

Dipersiapkan oleh:

Sabrina Editha Putri / 07 07 05439

Program Studi Teknik Informatika – Fakultas Teknologi
Industri
Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL - SIGPEGY		1/48
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX	-	A	B	C	D	E	F	G
TGL								
Ditulis oleh								
Diperik sa oleh								
Disetuj ui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

DAFTAR ISI

1	Pendahuluan.....	6
1.1	Tujuan	6
1.2	Ruang Lingkup	6
1.3	Definisi dan Akronim	8
1.4	Referensi	8
2	Perancangan Sistem.....	10
2.1	Perancangan Arsitektur.....	10
2.2	Perancangan Detail.....	11
2.2.1	Sequence Diagram.....	11
2.2.1.1	Login.....	11
2.2.1.2	Kelola Data Guru.....	11
2.2.1.2.1	Entry Data Guru.....	11
2.2.1.2.2	Edit Data Guru.....	12
2.2.1.2.3	Search Data Guru.....	12
2.2.1.2.4	Delete Data Guru.....	13
2.2.1.3	Kelola Data Anggota.....	13
2.2.1.3.1	Entry Data Anggota.....	13
2.2.1.3.2	Edit Data Anggota.....	14
2.2.1.3.3	Delete Data Anggota.....	14
2.2.1.3.4	Search Data Anggota.....	15
2.2.1.4	Display Request Guru.....	15
2.2.1.5	Kelola Sekolah.....	16
2.2.1.5.1	Entry Data Sekolah.....	16
2.2.1.5.2	Edit Data Sekolah.....	16
2.2.1.5.3	Search Data Sekolah.....	17
2.2.1.5.4	Delete Data Sekolah.....	17
2.2.1.6	Display Peta Penyebaran Guru.....	18
2.2.1.7	Display Sekolah.....	18
2.2.1.7	Display Guru.....	19
2.2.1.5	Kelola Data Request Guru.....	19
2.2.1.5.1	Entry Data Request Guru.....	19
2.2.1.5.2	Edit Data Request Guru.....	20
2.2.1.5.3	Delete Data Request Guru.....	20
2.2.1.5.4	Display Data Request Guru.....	21
2.3	Class Diagram.....	22
2.4	Classes Diagram Spesific Description.....	23
2.4.1	Spesific Design Class Login.....	23
2.4.2	Spesific Design Class KelolaAnggota.....	23
2.4.3	Spesific Design Class Tampil Kabupaten.....	24
2.4.4	Spesific Design Class Tampil Kecamatan.....	24
2.4.5	Spesific Design Class Kelola Sekolah.....	25
2.4.6	Spesific Design Class Kelola Guru.....	25
2.4.7	Spesific Design Class Kelola Request guru.....	26
2.4.8	Spesific Design Class Tampil Peta.....	26
2.4.9	Spesific Design Class Login Manager.....	26
2.4.10	Spesific Design Class Anggota Manager.....	27
2.4.11	Spesific Design Class Kabupaten Manger.....	28
2.4.12	Spesific Design Class Kecamatan Manager.....	28
2.4.13	Spesific Design Class Sekolah Manager.....	29
2.4.14	Spesific Design Class Guru Manager.....	29
2.4.15	Spesific Design Class Request guru Manager.....	30
2.4.16	Spesific Design Class Peta Manager.....	31

2.4.17	Spesific Design Class Login	32
2.4.18	Spesific Design Class Anggota	32
2.4.19	Spesific Design Class Kabupaten	33
2.4.20	Spesific Design Class Kecamatan	34
2.4.21	Spesific Design Class Sekolah	34
2.4.22	Spesific Design Class Guru	36
2.4.23	Spesific Design Class Request sekolah	37
2.4.24	Spesific Design Class Peta	38
3	Perancangan Data	39
3.1	Dekomposisi Data	39
3.1.1	Dekomposisi Entitas Data Anggota	39
3.1.2	Dekomposisi Entitas Data Kabupaten	39
3.1.3	Dekomposisi Entitas Data Kecamatan	39
3.1.4	Dekomposisi Entitas Data Sekolah	40
3.1.5	Dekomposisi Entitas Data Guru	40
3.1.6	Dekomposisi Entitas Data Request Guru	41
3.2	Physical Data Model	42
4	Perancangan Antarmuka	43
4.1	Deskripsi Perancangan Antarmuka	43
4.1	Antarmuka Beranda	43
4.2	Antarmuka kelola Guru	44
4.3	Antarmuka Tampil login bagi administrator	45
4.4	Antarmuka Kelola Anggota admin	46
4.5	Antarmuka Kelola Anggota Operator sekolah	47
4.6	Antarmuka Kelola Sekolah bagi administrator	48

DAFTAR GAMBAR

Gambar 2.1 - Arsitektur SIGPEGY	10
Gambar 2.2 - Sequence Diagram - Login	11
Gambar 2.3 - Sequence Diagram - Entry Data Guru	11
Gambar 2.4 - Sequence Diagram - Edit Data Guru	12
Gambar 2.5 - Sequence Diagram - Search Data Guru	12
Gambar 2.6 - Sequence Diagram - Delete Data Guru	13
Gambar 2.7 - Sequence Diagram - Entry Dtaa Anggota	13
Gambar 2.8 - Sequence Diagram - Edit Data Anggota	14
Gambar 2.9 - Sequence Diagram - Delete Data Anggota	14
Gambar 2.10 - Sequence Diagram - Search Data Anggota	15
Gambar 2.11 - Sequence Diagram - Display Request Guru	15
Gambar 2.12 - Sequence Diagram - Entry Data Sekolah	16
Gambar 2.13 - Sequence Diagram - Edit Data Sekolah	16
Gambar 2.14 - Sequence Diagram - Search Data Sekolah	17
Gambar 2.15 - Sequence Diagram - Delete Data Sekolah	17
Gambar 2.16 - Sequence Diagram - Display Peta penyebaran guru	18
Gambar 2.17 - Sequence Diagram - Display sekolah	18
Gambar 2.18 - Sequence Diagram - Display guru	19
Gambar 2.19 - Sequence Diagram - Entry Data request Guru	19
Gambar 2.20 - Sequence Diagram - Edit Data request Guru	20
Gambar 2.21 - Sequence Diagram - Delete Data request Guru	20
Gambar 2.22 - Sequence Diagram - Display Data request Guru	21
Gambar 2.23 - Class Diagram	22
Gambar 3.1 - Physical Data Model	42
Gambar 4.1 - Rancangan Antarmuka - Beranda	43
Gambar 4.1 - Rancangan Antarmuka - Kelola Guru	44
Gambar 4.1 - Rancangan Antarmuka - Login Administrator	45
Gambar 4.1 - Rancangan Antarmuka - Kelola Anggota	46
Gambar 4.1 - Rancangan Antarmuka - Kelola Anggota Sekolah	47
Gambar 4.1 - Rancangan Antarmuka - Kelola Sekolah (admin)	48

1. Pendahuluan

1.1. Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

1.2 Ruang Lingkup

Perangkat Lunak SIGPEGY dikembangkan dengan tujuan untuk :

7. Menampilkan informasi peta pemerataan penyebaran guru di wilayah Yogyakarta.
8. Menampilkan informasi detail pemerataan penyebaran guru bagi admin dan pengguna.
9. Mempermudah proses pengelolaan data guru oleh admin.

Dan dibangun dalam versi Web.

1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
SIGPEGY	Sistem Informasi Geografis Pemerataan Penyebaran Guru di Yogyakarta.

Upload	Istilah umum internet untuk mengirimkan suatu data/file dari klien ke server.
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk network global yang terdiri dari 164komputer dan layanannya dengan sekitar 30 sampai 50 juta pemakai 164komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
Database	Kumpulan data yang terkait yang diorganisasikan dalam struktur tertentu.
Website	Kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (hyperlink)
Framework	Kerangka kerja yang digunakan untuk membangun perangkat lunak.
ERD	Entity Relationship Diagram merupakan teknis grafis/diagram yang menggambarkan objek dan hubungan antar objek.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
Google Maps	Sebuah jasa <u>peta globe virtual</u> gratis dan online disediakan oleh <u>Google</u> dapat ditemukan di <u>http://maps.google.com</u>

1.4 Referensi

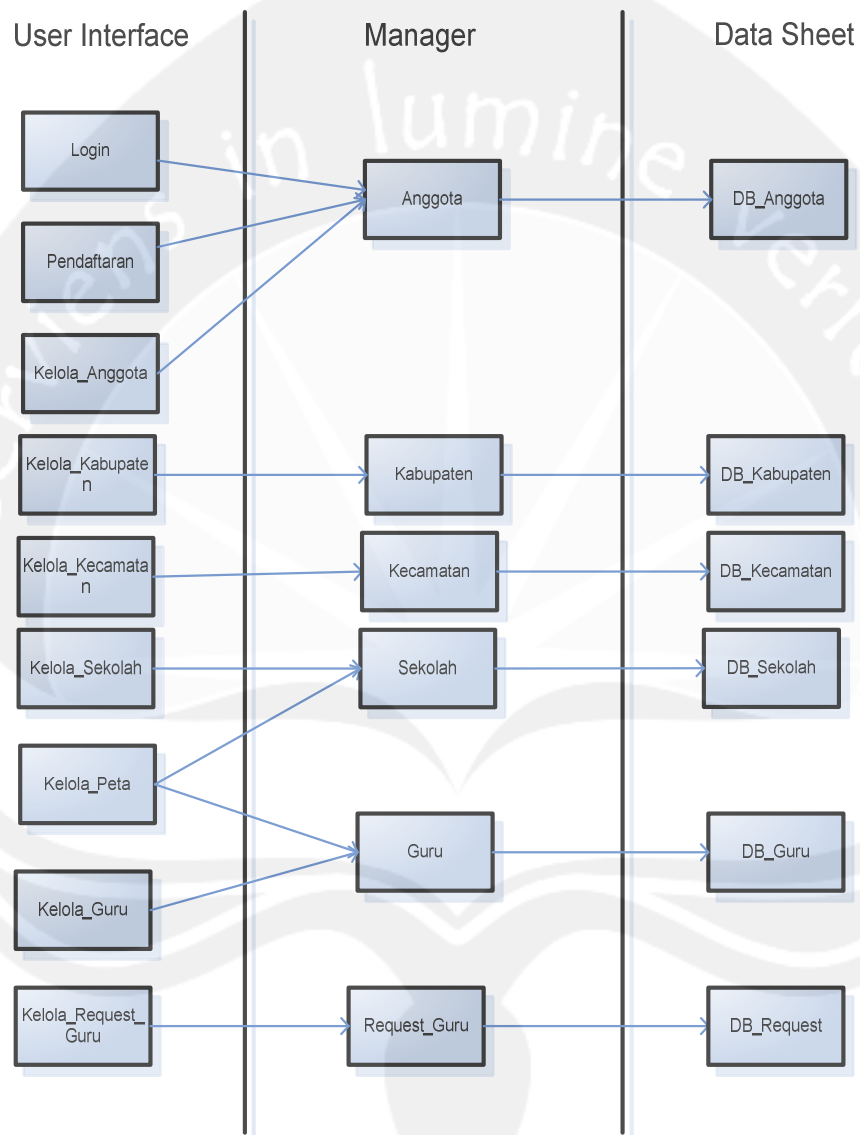
Referensi yang digunakan pada perangkat lunak tersebut adalah:

- a. Editha Putri, Sabrina, *Spesifikasi Kebutuhan Perangkat Lunak SIGPEGY (Sistem Informasi Geografis Pemerataan Penyebaran Guru di Yogyakarta)*, Jurusan Teknik Informatika – UAJY, 2010.
- b. Kadir, Abdul. *Dasar Pemrograman Web Dinamis Menggunakan PHP*, Penerbit Andi. 2008.



2 Perancangan Sistem

2.1 Perancangan Arsitektur

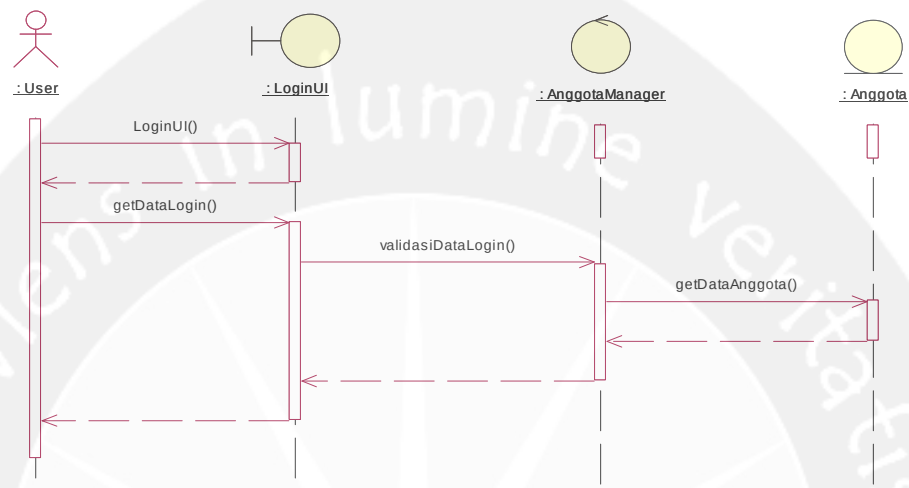


Gambar 2.1.1 Rancangan Arsitektur SIGPEGY

2.2 Perancangan Detail

2.2.1 Sequence Diagram

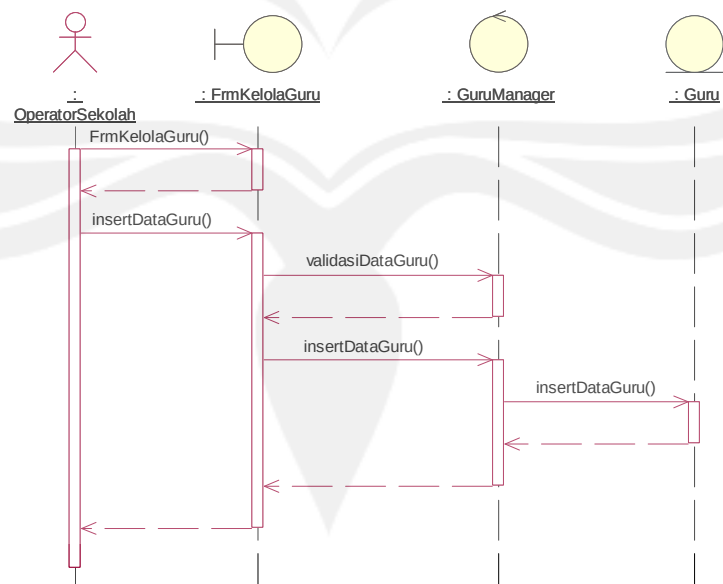
2.2.1.1 Login



Gambar 2.1.2 Sequence Diagram : Login

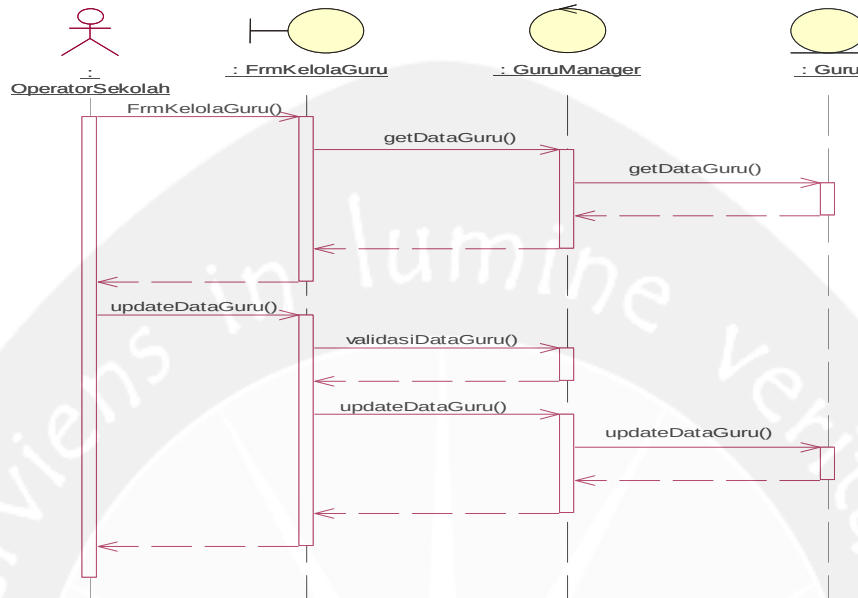
2.2.1.2 Kelola Data Guru

2.2.1.2.1 Entry Data Guru



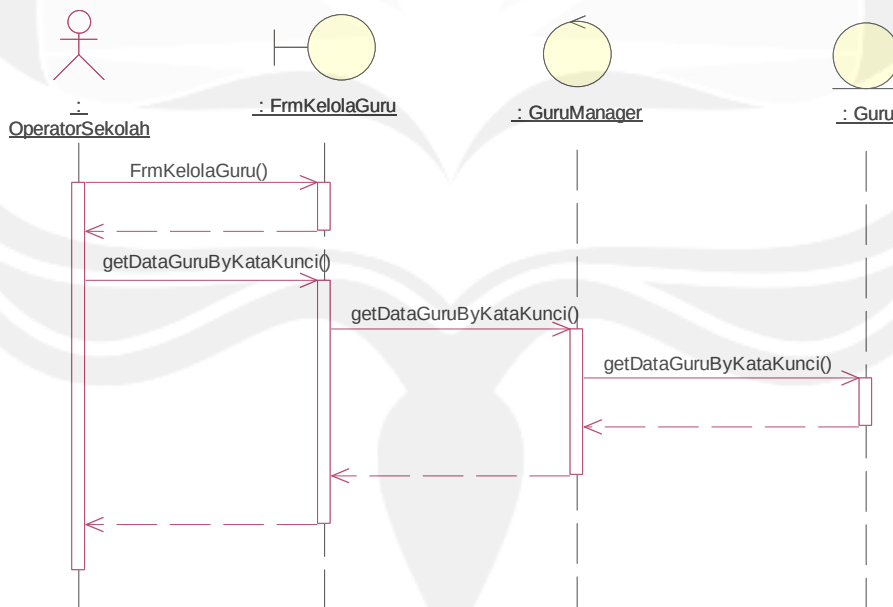
Gambar 2.1.3 Sequence Diagram : Kelola Data Guru - Entry Data Guru

2.2.1.2.2 Edit Data Guru



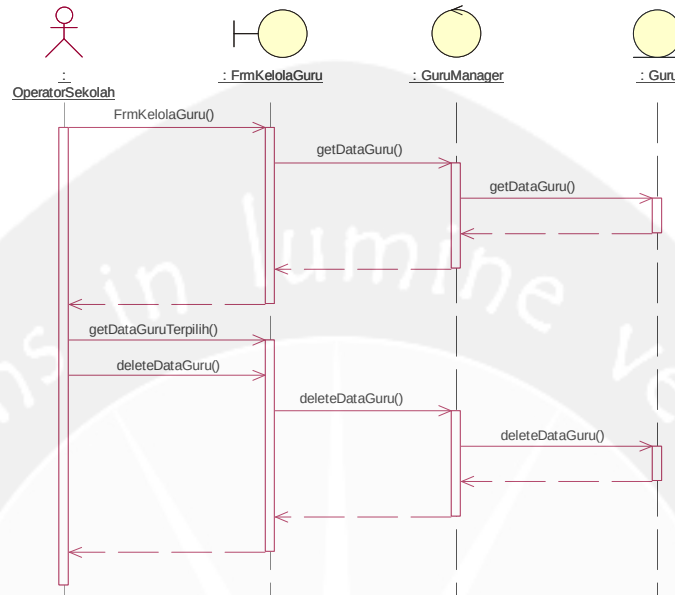
Gambar 2.1.4 Sequence Diagram : Kelola Data Guru - Edit Data Guru

2.2.1.2.3 Search Data Guru



Gambar 2.1.5 Sequence Diagram : Kelola Data Guru - Search Data Guru

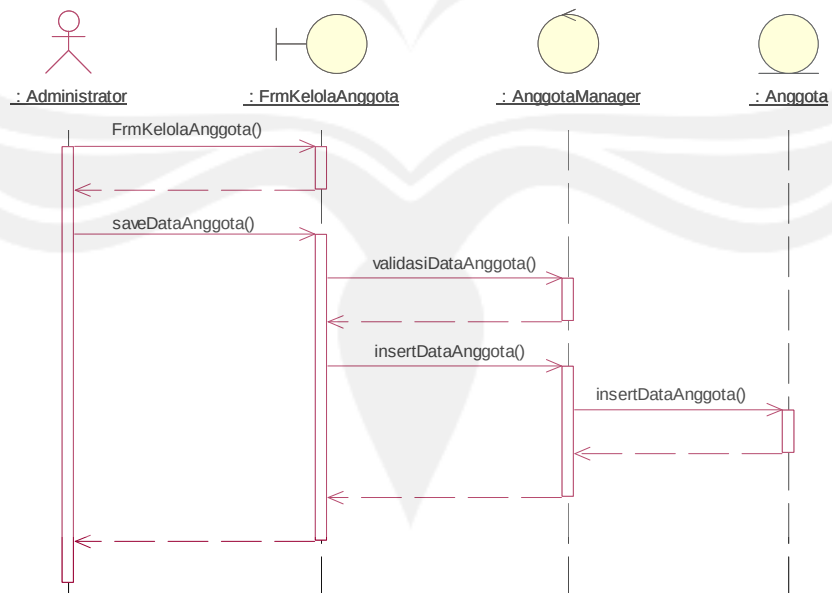
2.2.1.2.4 Delete Data Guru



Gambar 2.1.6 Sequence Diagram : Kelola Data Guru - Delete Data Guru

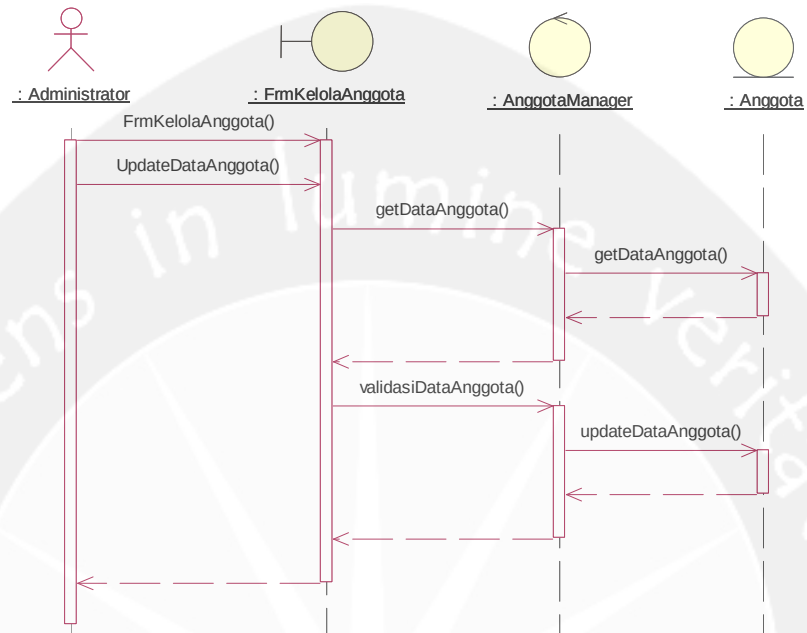
2.2.1.3 Kelola Data Anggota

2.2.1.3.1 Entry Data Anggota



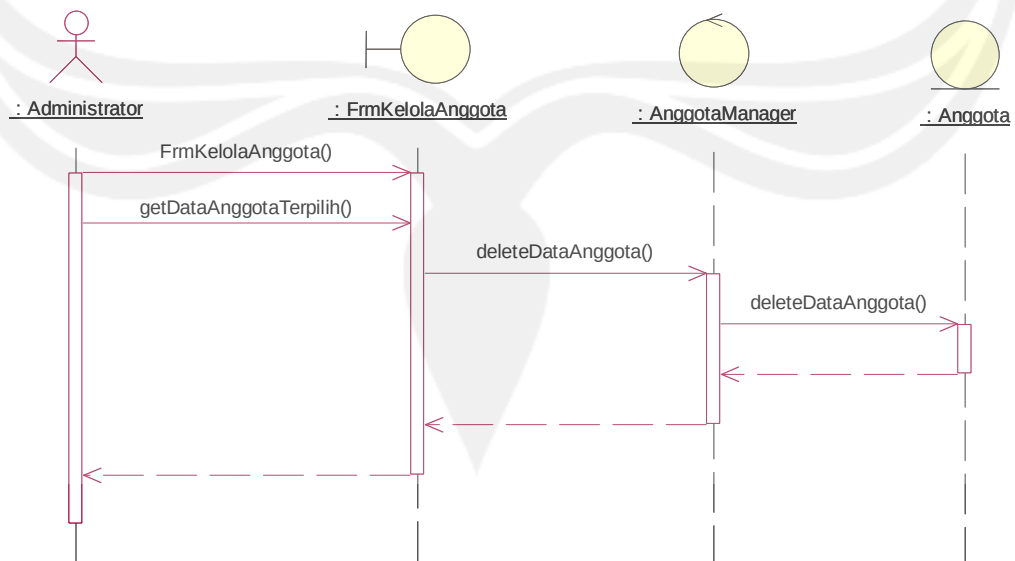
Gambar 2.1.7 Sequence Diagram : Kelola Data Anggota - Entry Data Anggota

2.2.1.3.2 Edit Data Anggota



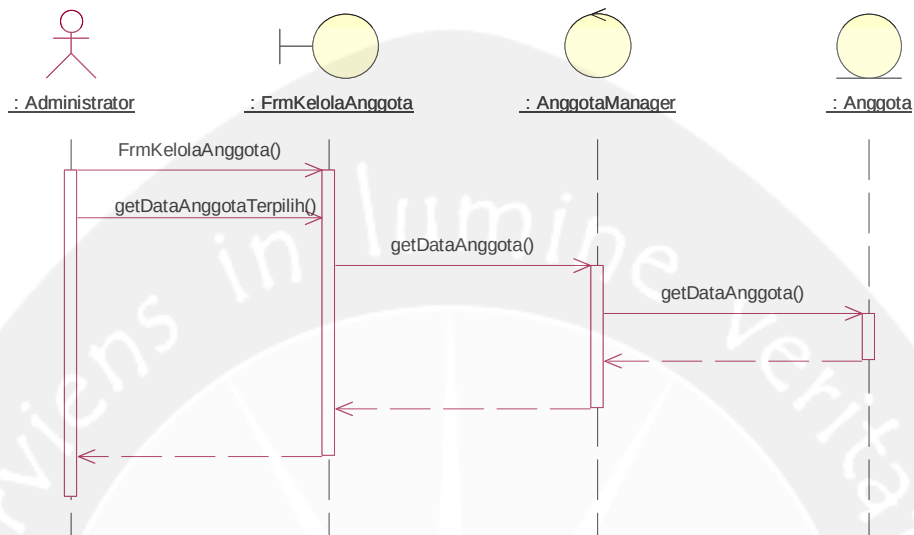
Gambar 2.1.8 Sequence Diagram : Kelola Data Anggota - Edit Data Anggota

2.2.1.3.3 Delete Data Anggota



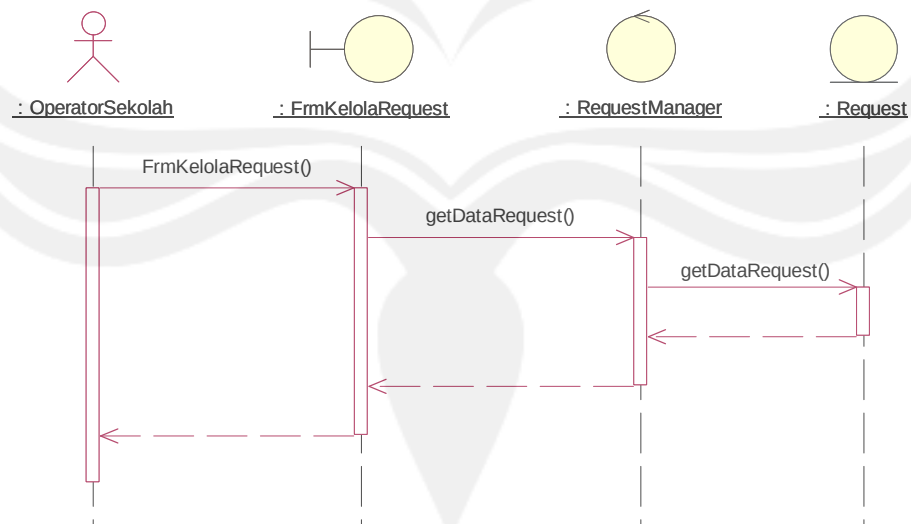
Gambar 2.1.9 Sequence Diagram : Kelola Data Anggota - Delete Data Anggota

2.2.1.3.4 Search Data Anggota



Gambar 2.1.10 Sequence Diagram : Kelola Data Anggota - Search Data Anggota

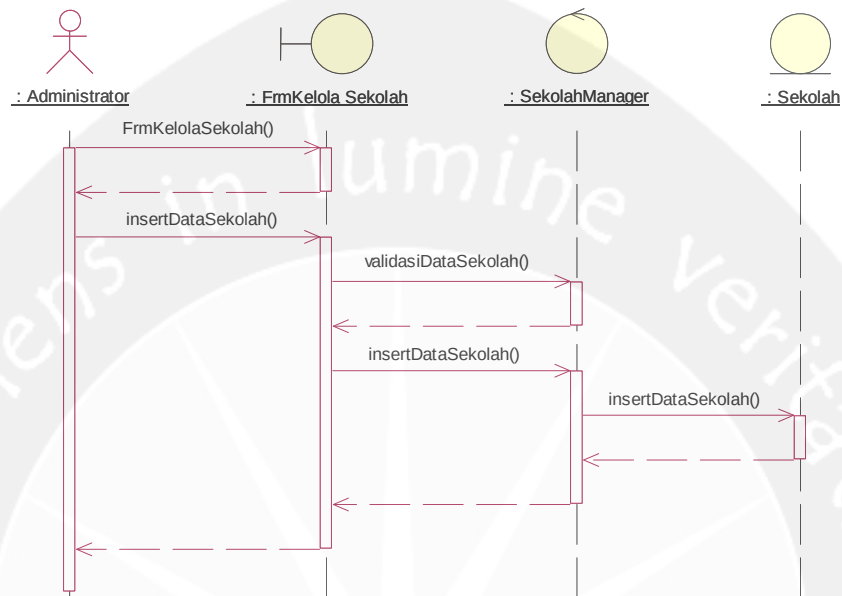
2.2.1.4 Display Request Guru



Gambar 2.1.11 Sequence Diagram : Display Data Request Guru

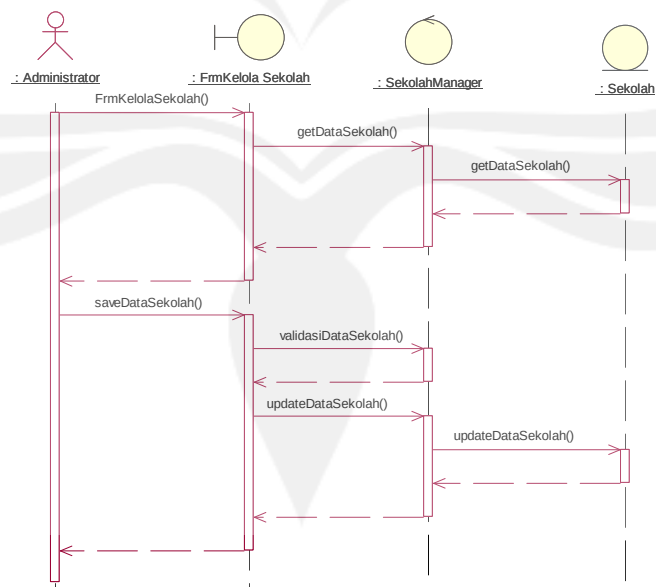
2.2.1.5 Kelola Sekolah

2.2.1.5.1 Entry Data Sekolah



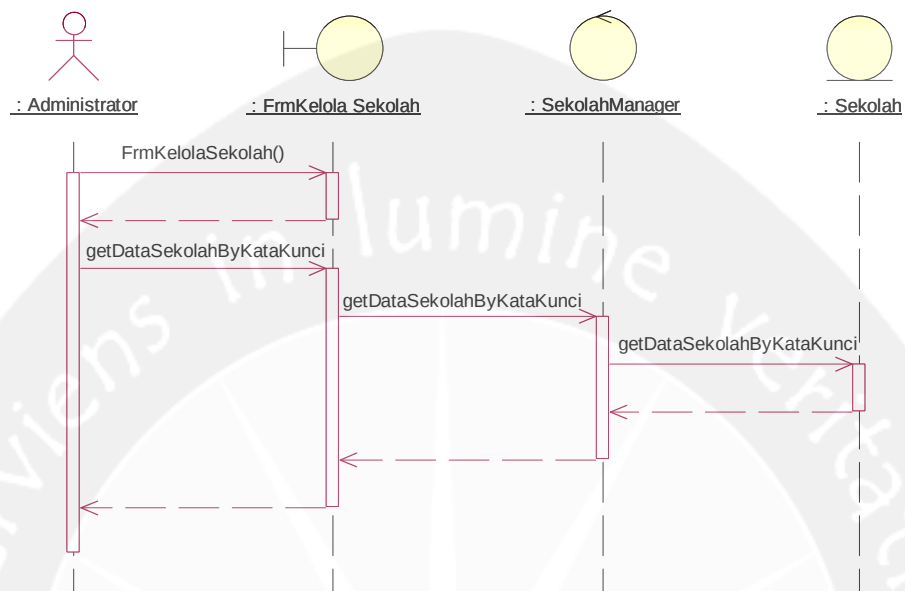
Gambar 2.1.12 Sequence Diagram : Kelola Sekolah - Entry Data Sekolah

2.2.1.5.2 Edit Data Sekolah



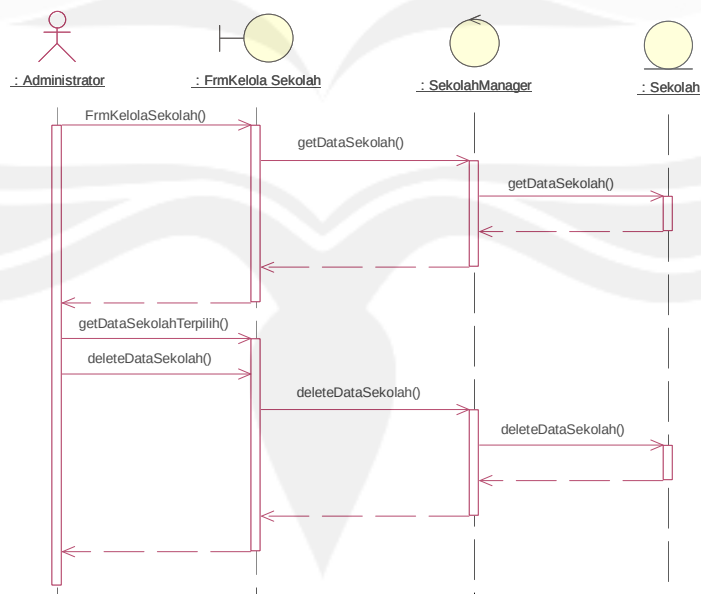
Gambar 2.1.13 Sequence Diagram : Kelola Sekolah - Edit Data Sekolah

2.2.1.5.3 Search Data Sekolah



Gambar 2.1.14 Sequence Diagram : Kelola Sekolah - Search Data Sekolah

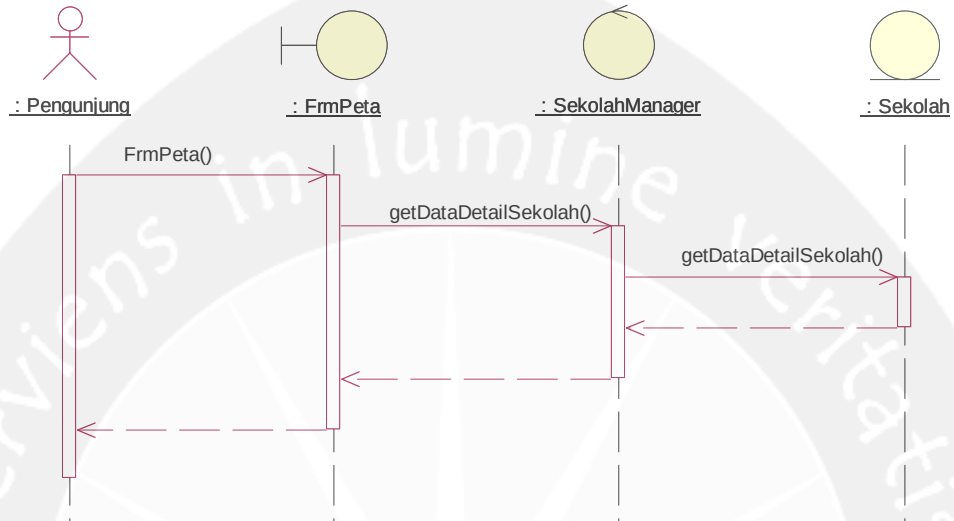
2.2.1.5.4 Delete Data Sekolah



Gambar 2.1.15 Sequence Diagram : Kelola Sekolah - Delete Data Sekolah

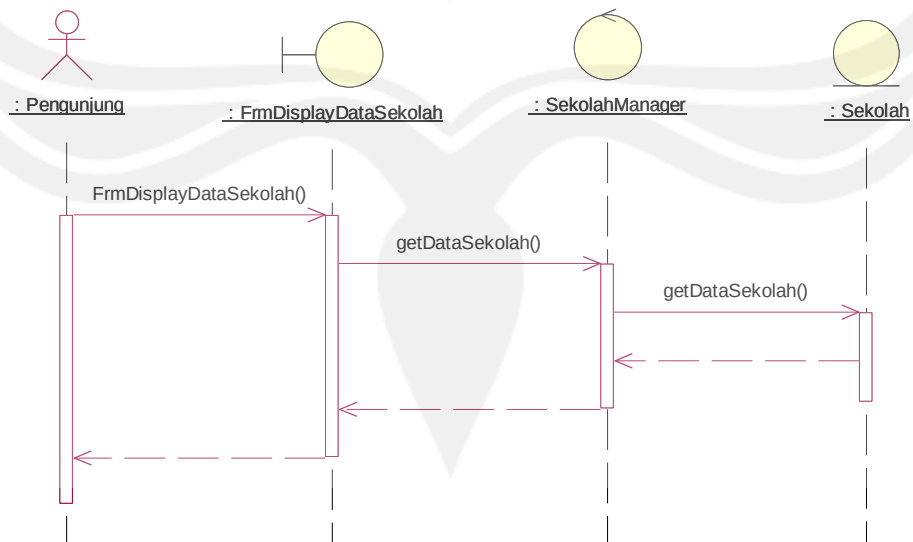
2.2.1.6 Display Data Pemerataan Penyebaran Guru

2.2.1.6.1 Display Peta Penyebaran Guru



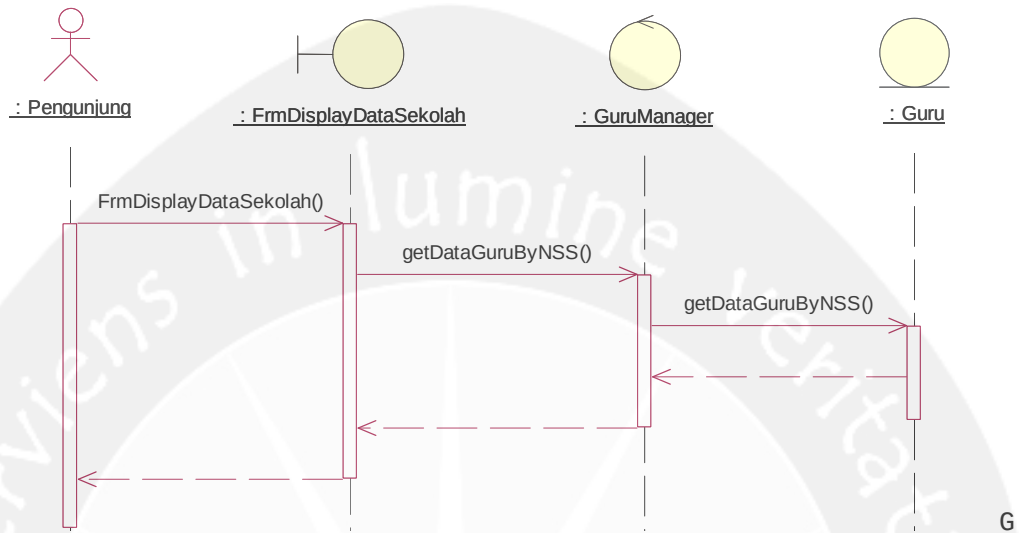
Gambar 2.1.16 Sequence Diagram : Display Peta Penyebaran Guru

2.2.1.6.2 Display Sekolah



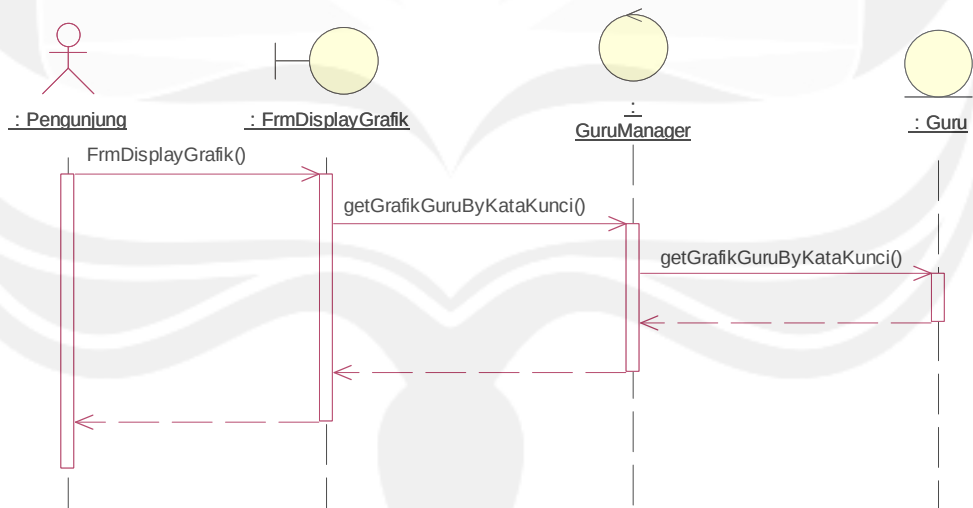
Gambar 2.1.17 Sequence Diagram : Display Data Sekolah

2.2.1.6.3 Display Guru



ambar 2.1.18 Sequence Diagram : Display Guru

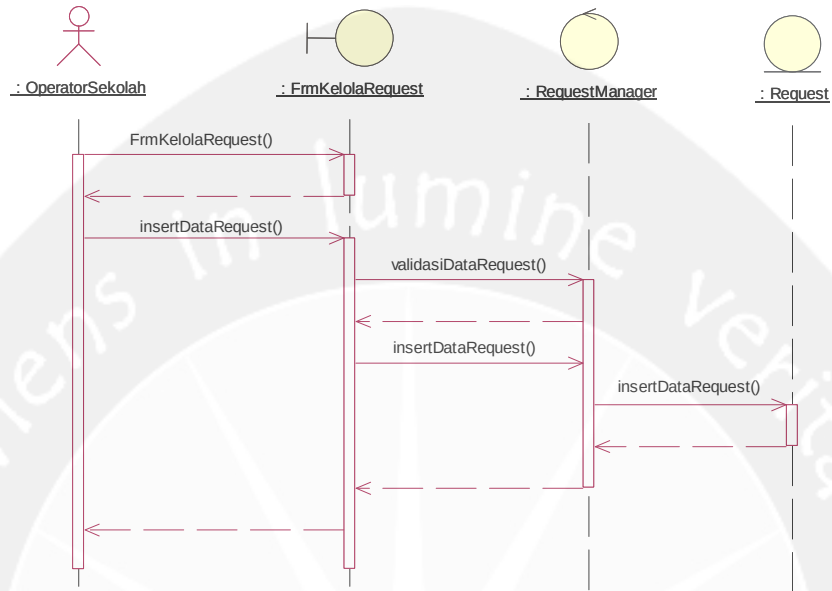
2.2.1.6.4 Display Grafik Pemerataan Penyebaran Guru



Gambar 2.1.19 Sequence Diagram : Display Grafik Pemerataan Penyebaran Guru

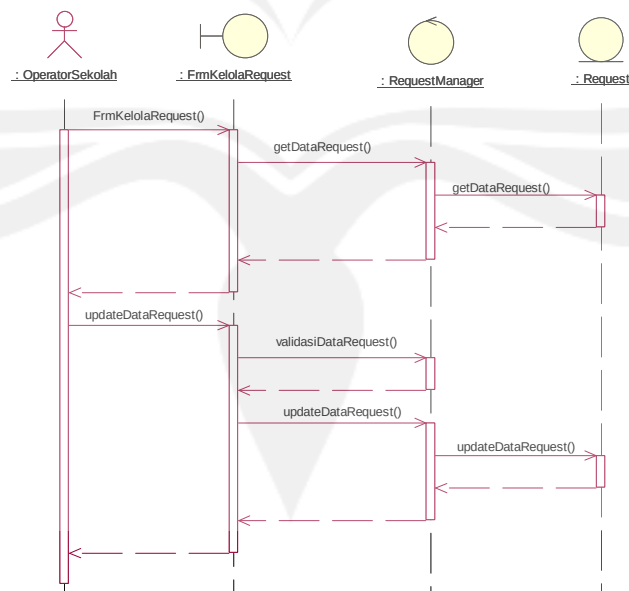
2.2.1.7 Kelola Data Request Guru

2.2.1.7.1 Entry Data Request Guru



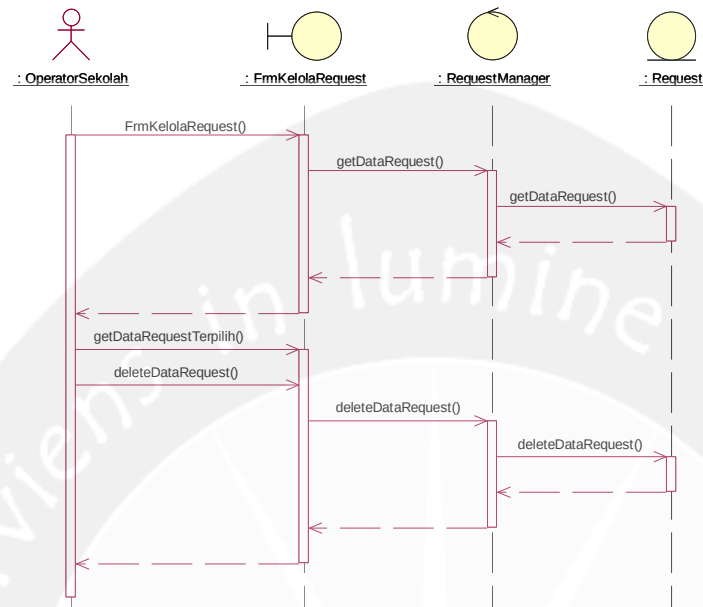
Gambar 2.1.20 Sequence Diagram : Kelola Data Request Guru - Entry

2.2.1.7.2 Edit Data Request Guru



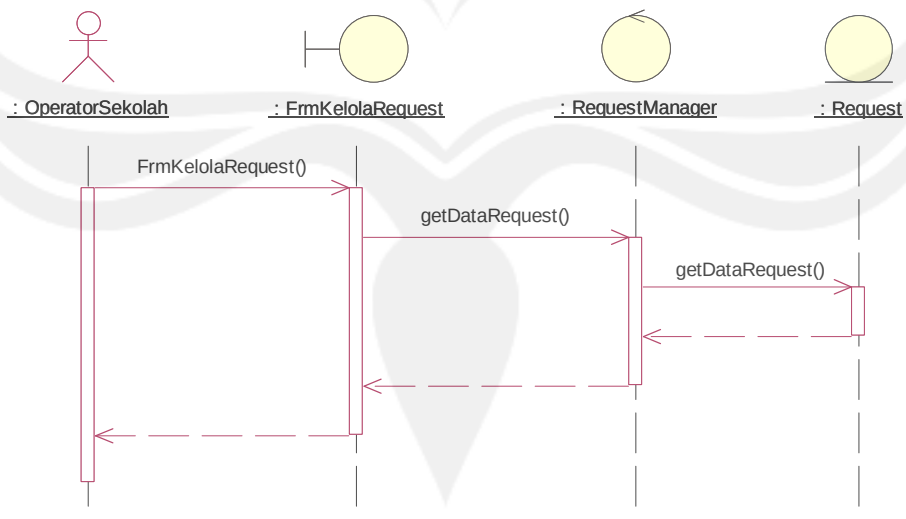
Gambar 2.1.21 Sequence Diagram : Kelola Data Request Guru - Edit
Data Request Guru

2.2.1.7.3 Delete Data Request Guru



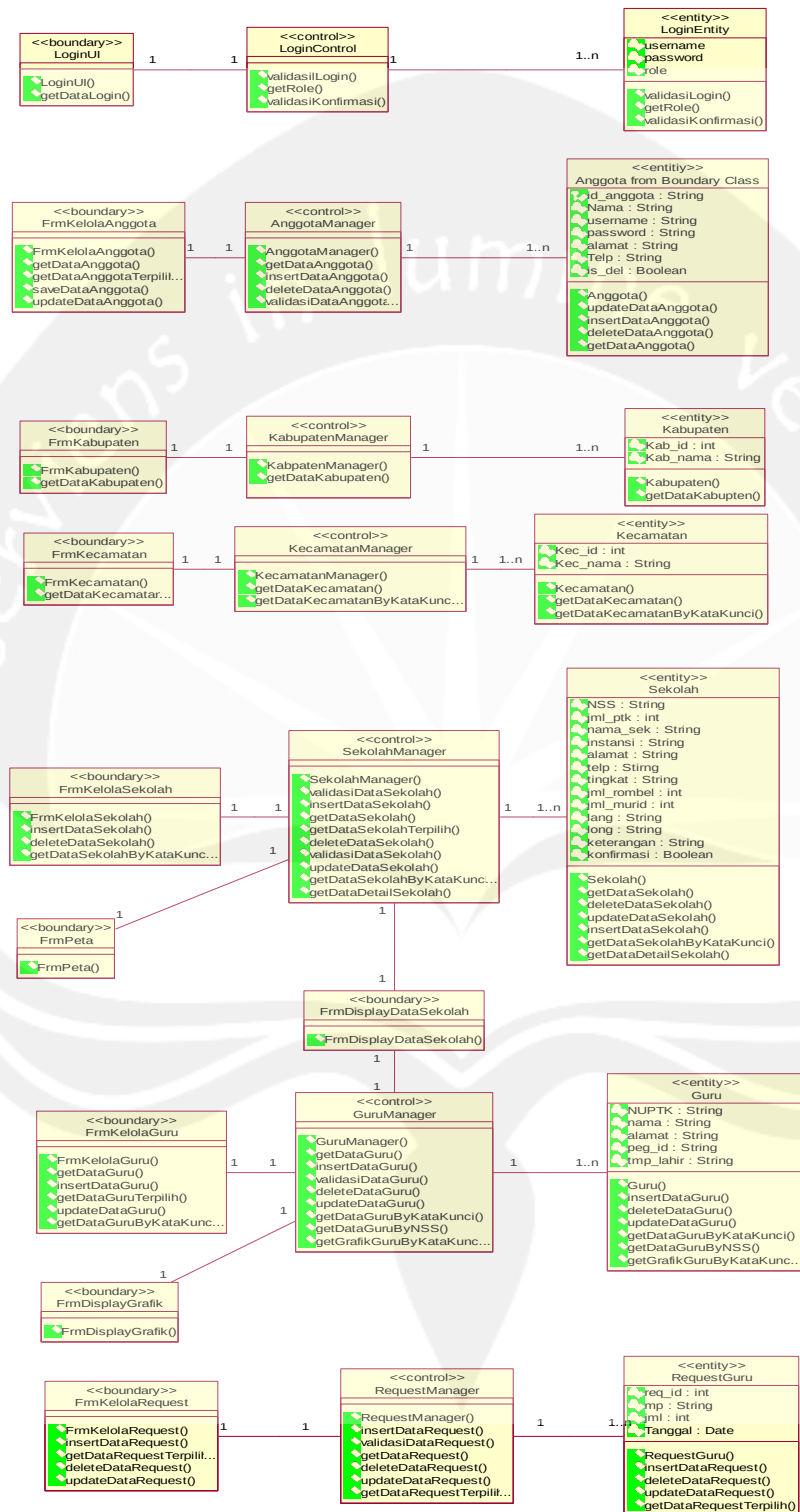
Gambar 2.1.22 Sequence Diagram : Kelola Data Request Guru - Delete Data Request Guru

2.2.1.7.4 Display Data Request Guru



Gambar 2.1.23 Sequence Diagram : Kelola Data Request Guru - Display Data Request Guru

2.3. Class Diagram



Gambar 2.9 Class Diagram SIGPEGY

2.4. Class Diagram Spes Description

2.4.1. Spesific Design Class Login

Login	<<boundary>>
+LoginUI() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getDataLogin() Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.	

2.4.2. Spesific Design Class KelolaAnggota

KelolaAnggota	<<boundary>>
+FrmKelolaAnggota () Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getDataAnggota() Operasi digunakan untuk menampilkan data anggota yang telah diinputkan ke dalam database.	
+getDataAnggotaTerpilih(Anggota) : Boolean Operasi ini digunakan untuk menampilkan data yang dicari pada grid view.	
+saveDataAnggota (Anggota): Boolean Operasi ini digunakan untuk menyimpan data anggota yang telah diinputkan ke dalam database.	
+updateDataAnggota(Anggota): Boolean Operasi ini digunakan untuk mengubah data anggota yang telah diinputkan ke dalam database.	

2.4.3. Spesific Design Class TampilKabupaten

TampilKabupaten	<<boundary>>
<pre>+FrmKabupaten() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataKabupaten() : DataTable Operasi ini digunakan untuk menampilkan seluruh data kabupaten di database.</pre>	

2.4.4. Spesific Design Class TampilKecamatan

TampilKecamatan	<<boundary>>
<pre>+FrmKecamatan() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataKecamatan() : DataTable Operasi ini digunakan untuk menampilkan seluruh data kecamatan di database.</pre>	

2.4.5. Spesific Design Class KelolaSekolah

KelolaSekolah	<<boundary>>
<pre>+KelolaSekolah() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataSekolah(Sekolah) : Boolean</pre>	

Operasi ini digunakan untuk menyimpan data sekolah ke database.

+deleteDataSekolah(Sekolah) : Boolean

Operasi ini digunakan untuk menghapus data sekolah di database.

+getDataSekolahByKataKunci() : DataTable

Operasi ini digunakan untuk menampilkan data sekolah yang terpilih.

2.4.6. Spesific Design Class KelolaGuru

KelolaGuru	<<boundary>>
+FrmKelolaGuru() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataGuru() : Boolean Operasi ini digunakan untuk menampilkan seluruh data guru di database ke datagrid. +insertDataGuru(Guru) : Boolean Operasi ini digunakan untuk menambah data guru ke dalam database. +getDataGuruTerpilih() Operasi ini digunakan untuk menampilkan data guru yang terpilih. +updateDataGuru(Guru) : Boolean Operasi ini digunakan untuk mengubah data guru yang berada dalam database. +getDataGuruByKataKunci Operasi ini digunakan untuk menampilkan data guru yang terpilih berdasarkan beberapa kata kunci.	

2.4.7. Spesific Design Class KelolaRequestGuru

KelolaRequestGuru	<<boundary>>
<pre>+FrmKelolaRequestGuru() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataRequestTerpilih() Operasi ini digunakan untuk menampilkan data request guru terpilih di database ke datagrid. +deleteDataRequest() Operasi ini digunakan untuk menghapus data request guru di database. +updateDataRequest() Operasi ini digunakan untuk mengubah data request guru di database.</pre>	

2.4.8. Spesific Design Class LoginManager

LoginControl	<<Control>>
<pre>+LoginManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +validasiLogin(username,password) : Boolean Operasi ini digunakan untuk mengecek data login yang diinputkan. Data login yang diinputkan user akan dibandingkan dengan data yang sudah tersimpan di database, apabila data login yang diinputkan benar maka akan direturnkan nilai True, jika sebaliknya akan direturnkan nilai False.</pre>	

+getRole(username,password) : String

Operasi ini digunakan untuk mengambil data role berdasarkan username dan password yang diinputkan oleh user.

+validasiKonfirmasi(username, password) : Boolean

Operasi ini digunakan untuk mengecek apakah data akun yang melakukan login telah dikonfirmasi atau belum. Jika belum maka proses login tidak terjadi.

2.4.9. Spesific Design Class AnggotaManager

LoginControl	<<Control>>
--------------	-------------

+LoginManager()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

+validasiLogin(username,password) : Boolean

Operasi ini digunakan untuk mengecek data login yang diinputkan. Data login yang diinputkan user akan dibandingkan dengan data yang sudah tersimpan di database, apabila data login yang diinputkan benar maka akan direturnkan nilai True, jika sebaliknya akan direturnkan nilai False.

+getRole(username,password) : String

Operasi ini digunakan untuk mengambil data role berdasarkan username dan password yang diinputkan oleh user.

+validasiKonfirmasi(username, password) : Boolean

Operasi ini digunakan untuk mengecek apakah data akun yang melakukan login telah dikonfirmasi atau belum. Jika belum maka proses login tidak terjadi.

2.4.10. Spesific Design Class KabupatenManager

KabupatenManager	<<Control>>
<pre>+ KabupatenManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataKabupaten() Operasi ini digunakan untuk menampilkan semua data kabupaten.</pre>	

2.4.11. Spesific Design Class KecamatanManager

KecamatanManager	<<Control>>
<pre>+ KecamatanManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataKecamatan() Operasi ini digunakan untuk menampilkan semua data kecamatan. +getDataKecamatanByKataKunci() : DataTable Operasi ini digunakan untuk menampilkan semua data kecamatan berdasarkan kata kunci tertentu.</pre>	

2.4.12. Spesific Design Class SekolahManager

KecamatanManager	<<Control>>
<pre>+ KecamatanManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</pre>	

+getDataKecamatan()

Operasi ini digunakan untuk menampilkan semua data kecamatan.

+getDataKecamatanByKataKunci() : DataTable

Operasi ini digunakan untuk menampilkan semua data kecamatan berdasarkan kata kunci tertentu.

2.4.13. Spesific Design Class GuruManager

GuruManager	<<Control>>
+GuruManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.	
+getDataGuru() Operasi ini digunakan untuk menampilkan semua data guru.	
+insertDataGuru (Guru) : Boolean Operasi ini digunakan untuk menyimpan data guru yang telah diinputkan ke dalam database.	
+validasiDataGuru(nuptk) : Boolean Operasi ini digunakan untuk mengecek validasi seorang guru.	
+updateDataGuru(Guru): Boolean Operasi ini digunakan untuk mengubah data guru yang telah diinputkan ke dalam database.	
+getDataGuruByKataKunci() : DataTable Operasi ini digunakan untuk menampilkan data guru berdasarkan kondisi tertentu.	
+deleteDataGuru(nuptk): Boolean Operasi ini digunakan untuk menghapus guru dari	

database.

+getDataGuruByKataKunci() : DataTable

Operasi ini digunakan untuk menampilkan data guru berdasarkan kondisi tertentu.

+getDataGuruByNSS(NSS) : DataTable

Operasi ini digunakan untuk menampilkan data guru berdasarkan nss tertentu.

+getGrafikGuruByKataKunci()

Operasi ini digunakan untuk menampilkan data grafik guru berdasarkan kondisi tertentu.

2.4.14. Spesific Design Class RequestGuruManager

RequestManager	<<Control>>
+RequestManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataRequest () Operasi ini digunakan untuk menampilkan semua data request guru. +insertDataRequest (RequestGuru) : Boolean Operasi ini digunakan untuk menyimpan data request guru yang telah diinputkan ke dalam database. +updateDataRequest(Guru): Boolean Operasi ini digunakan untuk mengubah data request guru yang telah diinputkan ke dalam database. +deleteDataRequest(req_id): Boolean Operasi ini digunakan untuk menghapus request guru dari database. +getDataRequestTerpilih(nss) : DataTable Operasi ini digunakan untuk menampilkan data request	

guru berdasarkan sekolah tertentu.

2.4.15. Spesific Design Class Login

Login	<<entity>>
-username : String Atribut ini digunakan untuk menyimpan data username. -password : String Atribut ini digunakan untuk menyimpan data password. -role : String Atribut ini digunakan untuk menyimpan role.	
+validasiLogin(username,password) : Boolean Operasi ini digunakan untuk mengecek data login yang diinputkan. Data login yang diinputkan user akan dibandingkan dengan data yang sudah tersimpan di database, apabila data login yang diinputkan benar maka akan direturnkan nilai True, jika sebaliknya akan direturnkan nilai False. +getRole(username,password) : String Operasi ini digunakan untuk mengambil data role berdasarkan username dan password yang diinputkan oleh user. +validasiKonfirmasi(username, password) : Boolean Operasi ini digunakan untuk mengecek apakah data akun yang melakukan login telah dikonfirmasi atau belum. Jika belum maka proses login tidak terjadi.	

2.4.16. Spesific Design Class Anggota

Anggota	<<entity>>
-username : String Atribut ini digunakan untuk menyimpan data username.	

-password : String

Atribut ini digunakan untuk menyimpan data password.

-role : String

Atribut ini digunakan untuk menyimpan role.

-alamat : String

Atribut ini digunakan untuk menyimpan alamat.

-telp : String

Atribut ini digunakan untuk menyimpan nomor telepon.

-is_del : Boolean

Atribut ini digunakan sebagai penanda penghapusan.

+Anggota()

Operasi ini digunakan untuk mendapatkan username.

+getDataAnggota()

Operasi ini digunakan untuk menampilkan semua data anggota.

+insertDataAnggota(Anggota) : Boolean

Operasi ini digunakan untuk menyimpan data anggota yang telah diinputkan ke dalam database.

+updateAnggota(Anggota): Boolean

Operasi ini digunakan untuk mengubah data anggota yang telah diinputkan ke dalam database.

+deleteAnggota(username): Boolean

Operasi ini digunakan untuk menghapus anggota dari database.

2.4.17. Spesific Design Class Kabupaten

Kabupaten	<<Entity>>
-Kab_id	
Atribut ini digunakan untuk menyimpan data kabupaten.	
-Kab_nama	

Atribut ini digunakan untuk menyimpan data kabupaten.

+ Kabupaten()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

+getDataKabupaten()

Operasi ini digunakan untuk menampilkan semua data kabupaten.

2.4.18. Spesific Design Class Kecamatan

Kecamatan	<<Entity>>
-Kec_id Atribut ini digunakan untuk menyimpan ID kecamatan. -Kec_nama Atribut ini digunakan untuk menyimpan nama kecamatan.	
+ Kecamatan() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getDataKecamatan() Operasi ini digunakan untuk menampilkan semua data kecamatan. +getDataKecamatanByKataKunci(Kecamatan) : DataTable Operasi ini digunakan untuk menampilkan semua data kecamatan berdasarkan kondisi tertentu.	

2.4.19. Spesific Design Class Sekolah

Sekolah	<<Entity>>
-NSS Atribut ini digunakan untuk menyimpan NSS sekolah. -jml_ptk Atribut ini digunakan untuk menyimpan jumlah guru	

sekolah.

-nama_sek

Atribut ini digunakan untuk menyimpan nama sekolah.

-instansi

Atribut ini digunakan untuk menyimpan nama instansi sekolah.

-alamat

Atribut ini digunakan untuk menyimpan alamat sekolah.

-telp

Atribut ini digunakan untuk menyimpan telepon sekolah.

-tingkat

Atribut ini digunakan untuk menyimpan tingkat sekolah.

-jml_rombel

Atribut ini digunakan untuk menyimpan jumlah ruang belajar sekolah.

-jml_murid

Atribut ini digunakan untuk menyimpan jumlah murid sekolah.

-lang

Atribut ini digunakan untuk menyimpan data latitude sekolah.

-long

Atribut ini digunakan untuk menyimpan data longitude sekolah.

-keterangan

Atribut ini digunakan untuk menyimpan keterangan sekolah.

-konfirmasi

Atribut ini digunakan untuk menyimpan konfirmasi sekolah.

+ Sekolah()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

+getDataSekolah()

Operasi ini digunakan untuk menampilkan semua data sekolah.

+insertDataSekolah(Sekolah) : Boolean

Operasi ini digunakan untuk menyimpan data sekolah yang telah diinputkan ke dalam database.

+updateDataSekolah(Sekolah): Boolean

Operasi ini digunakan untuk mengubah data sekolah yang telah diinputkan ke dalam database.

+deleteDataSekolah(username): Boolean

Operasi ini digunakan untuk menghapus sekolah dari database.

+getDataSekolahByKataKunci() : Boolean

Operasi ini digunakan untuk menampilkan data sekolah berdasarkan suatu kondisi (kata kunci) tertentu.

+getDataDetailSekolah() : Boolean

Operasi ini digunakan untuk menampilkan seluruh data detail suatu sekolah.

2.4.20. Spesific Design Class Guru

Guru	<<Entity>>
-NUPTK Atribut ini digunakan untuk menyimpan NUPTK guru. -nama Atribut ini digunakan untuk menyimpan nama guru. -alamat Atribut ini digunakan untuk menyimpan alamat guru.	

-peg_id

Atribut ini digunakan untuk menyimpan id pegawai guru di sekolah.

-tmp_lahir

Atribut ini digunakan untuk menyimpan tempat lahir guru.

+Guru()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

+insertDataGuru (Guru) : Boolean

Operasi ini digunakan untuk menyimpan data guru yang telah diinputkan ke dalam database.

+updateDataGuru(Guru): Boolean

Operasi ini digunakan untuk mengubah data guru yang telah diinputkan ke dalam database.

+getDataGuruByKataKunci() : DataTable

Operasi ini digunakan untuk menampilkan data guru berdasarkan kondisi tertentu.

+deleteDataGuru(nuptk): Boolean

Operasi ini digunakan untuk menghapus guru dari database.

+getDataGuruByKataKunci() : DataTable

Operasi ini digunakan untuk menampilkan data guru berdasarkan kondisi tertentu.

+getDataGuruByNSS(NSS) : DataTable

Operasi ini digunakan untuk menampilkan data guru berdasarkan nss tertentu.

+getGrafikGuruByKataKunci()

Operasi ini digunakan untuk menampilkan data grafik guru berdasarkan kondisi tertentu.

2.4.21. Spesific Design Class RequestSekolah

RequestGuru	<<Entity>>
-req_id Atribut ini digunakan untuk menyimpan id request guru. -mp Atribut ini digunakan untuk menyimpan mata pelajaran request guru. -jml Atribut ini digunakan untuk menyimpan data jumlah request guru -tanggal Atribut ini digunakan untuk menyimpan data tanggal request guru.	
+RequestGuru() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +insertDataRequest (RequestGuru) : Boolean Operasi ini digunakan untuk menyimpan data request guru yang telah diinputkan ke dalam database. +updateDataRequest(Guru): Boolean Operasi ini digunakan untuk mengubah data request guru yang telah diinputkan ke dalam database. +deleteDataRequest(req_id): Boolean Operasi ini digunakan untuk menghapus request guru dari database. +getDataRequestTerpilih(nss) : DataTable Operasi ini digunakan untuk menampilkan data request guru berdasarkan sekolah tertentu.	

3 Perancangan Data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Data Anggota

Nama	Tipe	Panjang	Keterangan
username	Character	20	username anggota, Primary key
password	Character	20	password anggota
nama	Character	50	nama dari anggota
alamat	Character	100	alamat dari anggota
telp	Character	20	telepon dari anggota
TANGGAL_MASUK	Date	-	Tanggal anggota dipekerjakan
TANGGAL_KELUAR	Date	-	Tanggal administrator diberhentikan.
Is_del	Boolean	-	Penanda hapus anggota

3.1.2 Deskripsi Entitas Data Kabupaten

Nama	Tipe	Panjang	Keterangan
Kab_id	Integer	3	id kabupaten
Kab_nama	String	30	Nama dari kabupaten

3.1.3 Deskripsi Entitas Data Kecamatan

Nama	Tipe	Panjang	Keterangan
Kec_id	Integer	3	Id kecamatan, Primary Key
Kec_nama	Character	50	nama dari kecamatan
Jml_guru_kec	Integer	9	Jumlah seluruh guru dari kecamatan
Jml_murid_kec	Integer	9	Jumlah seluruh murid dari kecamatan

3.1.4 Deskripsi Entitas Data Sekolah

Nama	Tipe	Panjang	Keterangan
NSS	Character	30	Nomor Statistik Sekolah, Primary key

password	Character	30	password dari sekolah
namaSekolah	Character	50	Nama dari sekolah
instansi	Character	30	Instansi dari sekolah
tingkat	Character	30	Tingkat pendidikan sekolah
telp	Character	15	Telepon dari sekolah
jmlRombel	Integer	3	Jumlah ruang belajar
jmlMurid	Integer	4	Jumlah murid
jmlPTK	Integer	4	Jumlah tenaga pendidik
lat	Float	10,6	Koordinat latitude marker peta
long	Float	10,6	Koordinat longitude marker peta
Nama_kepsek	Character	50	Nama kepala sekolah
Keterangan	Character	500	Keterangan sekolah
Konfirmasi	Boolean	-	Penanda konfirmasi sekolah

3.1.5 Deskripsi Entitas Data Guru

Nama	Tipe	Panjang	Keterangan
NUPTK	Character	50	Nomor Unik Pendidikan dan Tenaga Kependidikan, Primary key
statusNUPTK	Character	50	status NUPTK dari guru
nama	Character	50	nama dari guru
alamat	Character	50	alamat dari guru
tmp_lahir	Character	20	tempat lahir dari guru
peg_id	Character	50	id pegawai dari guru di sekolah tempat guru mengajar

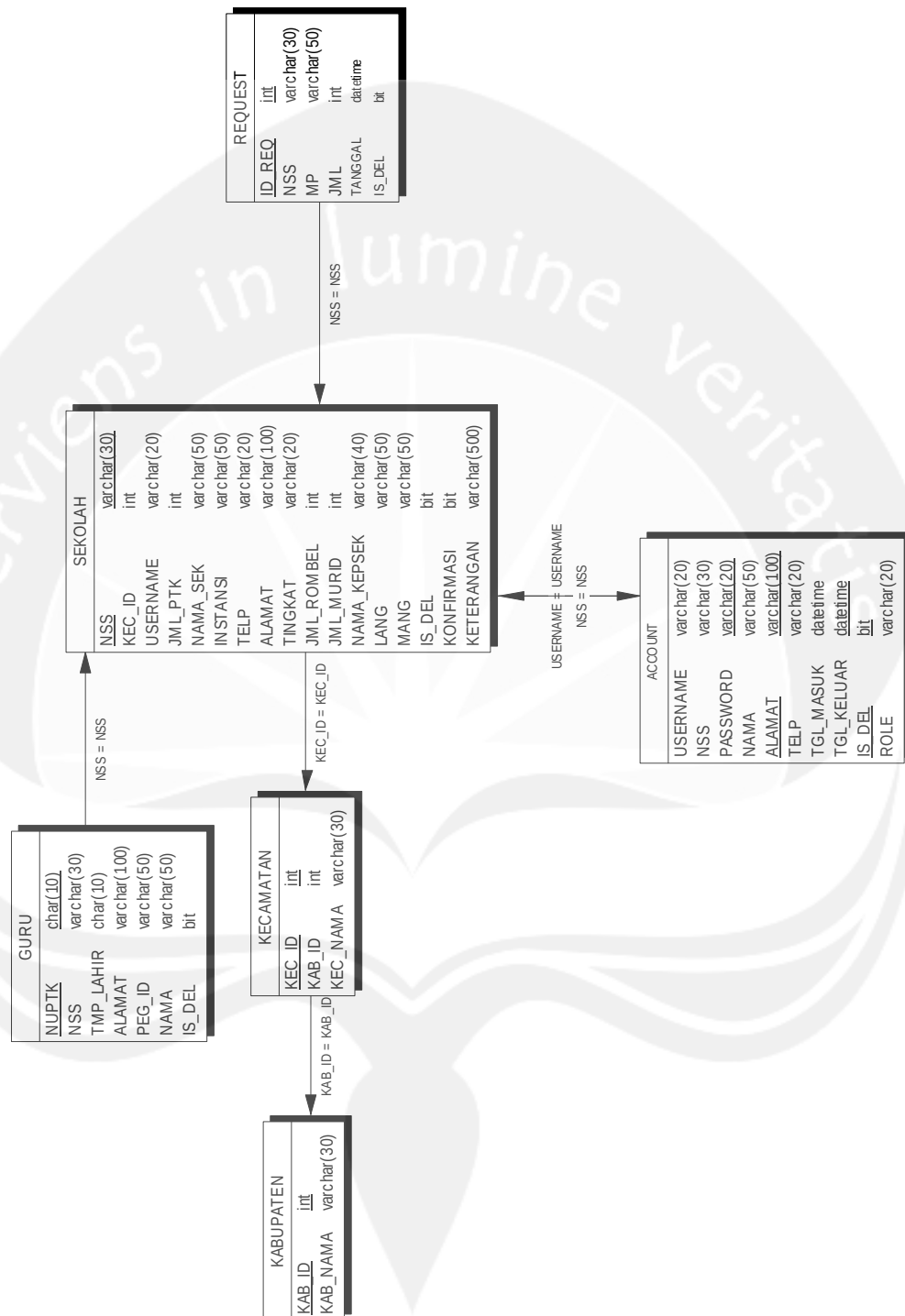
3.1.6 Deskripsi Entitas Data Request Guru

Nama	Tipe	Panjang	Keterangan
NUPTK	Character	50	Nomor Unik Pendidikan dan Tenaga Kependidikan, Primary key
statusNUPTK	Character	50	status NUPTK dari guru

nama	Character	50	nama dari guru
alamat	Character	50	alamat dari guru
tmp_lahir	Character	20	tempat lahir dari guru
peg_id	Character	50	id pegawai dari guru di sekolah



3.2 Physical Data Model



Gambar 3.1 Physical Data Model SIGPEGY

4 Perancangan Antarmuka

4.1 Deskripsi Perancangan AntarMuka

4.1.1 Beranda

The wireframe shows a web interface for 'SIGPEGY'. At the top, there is a navigation bar with five links: 'Beranda', 'Peta', 'Tentang Kami', 'Yogyakarta', and 'kontak'. Below the navigation bar, the main content area is divided into two columns. The left column contains the text 'SELAMAT DATANG DI SIPEGY' followed by a large rectangular placeholder labeled 'Peta Pesebaran Guru Di Yogyakarta'. The right column contains a login section with a 'Nama Pengguna' label, a password field (indicated by asterisks), a 'LOGIN' button, and a 'Daftar!' link. A footer bar is located at the bottom of the page.

Gambar 4.1.1 Rancangan Antarmuka Beranda

Pada antarmuka gambar 4.1.1 menunjukan rancangan antar muka beranda saat pengguna pertama kali menggunakan sistem, pada antar muka ini pengguna sistem dapat melakukan proses login dan dapat melakukan pendaftaran untuk pengguna yang ingin menggunakan sistem. Untuk mendapat akses masuk ke dalam sistem, pengguna harus memasukan nama pengguna dan password

dengan benar pada textbox yang telah disediakan. Jika proses login benar maka pengguna akan memasuki sistem.

4.1.2 Kelola Guru

Beranda Peta Tentang Kami Yogyakarta kontak

SIGPEGY

Kelola Guru

Cari Berdasarkan : NSS CARI

Kolom	Kolom	Kolom	Kolom

REFERSH HAPUS

Nama

NUPTK

NSS Sekolah

ID Pegawai

Tempat Lahir

Alamat

SIMPAN BERSIHKAN

Nama Pengguna

***** LOGIN Daftar!

Gambar 4.1.2 Rancangan Antarmuka Kelola Guru

Antarmuka pada gambar 4.1.2 menunjukkan halaman untuk melakukan kelola guru pengguna yang dengan role sebagai admin akan melakukan pengelolaan guru, terdapat menu untuk pencarian guru yang kemudian data guru akan

ditampilkan dalam bentuk tabel. Terdapat juga tombol hapus untuk melakukan penghapusan data guru.

4.1.3 Tampil Login bagi Administrator

Beranda Peta Tentang Kami Yogyakarta kontak	
SIGPEGY	
SELAMAT DATANG DI SIPEGY	Nama Pengguna ***** LOGIN Daftar!

Gambar 4.1.3 Rancangan Antarmuka Tampil Login bagi Administrator

Antarmuka gambar 4.1.3 menunjukkan rancangan antarmuka pada saat Administrator hendak melakukan login. Halaman login Administrator berbeda dengan Operator Sekolah.

4.1.4 Kelola Anggota

Beranda Peta Tentang Kami Yogyakarta kontak

SIGPEGY

Kelola Anggota

Pilih kelola akun berdasarkan : Admin ▼ TAMPIL

Cari Berdasarkan : Nama Pengguna ▼ CARI

Kolom	Kolom	Kolom	Kolom
-------	-------	-------	-------

REFRESH HAPUS

Nama Pengguna :

Nama Asli :

Telepon :

Alamat :

Role : Administrator ▼

NSS :

VALIDASI

SIMPAN BATIL

**Gambar 4.1.4 Rancangan Antarmuka Kelola Anggota
(Administrator)**

Antarmuka pada gambar 4.1.4 menunjukkan halaman untuk melakukan kelola pengguna yang menggunakan sistem. Terdapat menu untuk pencarian anggota, melakukan edit pengguna dan juga terdapat menu untuk menambah pengguna yang dengan role sebagai administrator.

4.1.5 Kelola Anggota (Operator sekolah)

Beranda Peta Tentang Kami Yogyakarta kontak

SIGPEGY

Kelola Anggota

Pilih kelola akun berdasarkan : Operator Sekolah ▼ TAMPIL

Cari Berdasarkan : Nama Pengguna ▼ CARI

Kolom	Kolom	Kolom	Kolom
-------	-------	-------	-------

REFRESH HAPUS

Nama Pengguna :

Nama Asli :

Telepon :

Alamat :

Role : Administrator ▼

NSS :

VALIDASI

SIMPAN BATAL

Gambar 4.1.5 Rancangan Antarmuka Kelola Anggota (Operator Sekolah)

Antarmuka pada gambar 4.1.5 menunjukkan halaman untuk melakukan kelola pengguna dengan role operator sekolah yang menggunakan sistem. Terdapat menu untuk pencarian anggota, melakukan edit pengguna dan juga terdapat menu untuk menambah pengguna yang dengan role sebagai operator sekolah.

4.1.6 Kelola Sekolah bagi Administrator

Beranda Peta Tentang Kami Yogyakarta kontak

SIGPEGY

Kelola Sekolah

Cari Berdasarkan : NSS Sekolah

Kolom	Kolom	Kolom	Kolom

Nama Sekolah :

Tingkat

Alamat

Telpon

Guru

Murid

Kepala Sekolah

Gambar 4.1.6 Rancangan Antarmuka Kelola Sekolah

Antarmuka pada gambar 4.1.6 menunjukkan halaman untuk melakukan kelola sekolah yang sudah terdaftar kedalam sistem, pengelolaan sekolah akan dilakukan pengguna yang mempunyai role admin atau operator sekolah.