

BAB I PENDAHULUAN

1. Latar Belakang

PT. PLN (Persero) merupakan sebuah Badan Usaha Milik Negara (BUMN) yang menangani segala aspek yang berkaitan dengan ketenagalistrikan yang ada di Indonesia. PT. PLN. (Persero) memerlukan teknologi informasi sebagai pendukung tugas utama mereka yaitu menjamin tersedianya tenaga listrik di Indonesia. Dalam usaha mencapai tujuan dan tanggung jawabnya dalam memastikan ketersediaan tenaga listrik di Indonesia, PT. PLN. (Persero) terbagi menjadi beberapa wilayah distribusi yang tersebar diseluruh daerah di Indonesia. Distribusi tersebut membawahi banyak Area Pelayanan Jaringan (APJ) yang mencakup daerah-daerah yang termasuk wilayah kerja distribusi tersebut. Kota Malang termasuk pada Area Pelayanan Jaringan (APJ) Malang yang dinaungi oleh distribusi Jawa Timur. Suatu Area Pelayanan Jaringan (APJ) dipimpin oleh seorang manajer APJ.

Kesehatan keamanan dan keselamatan kerja bagi para karyawan merupakan salah satu tanggung jawab yang dijalankan oleh bidang distribusi. Dimana standarisasi mengenai kesehatan, keselamatan, dan keamanan kerja sudah diterapkan pada PT. PLN (Persero) area pelayanan dan jaringan Malang.

Dalam kaitannya dengan kesehatan, keselamatan dan keamanan kerja, PT. PLN (Persero) Area Pelayanan dan Jaringan Malang memiliki salah satu bagian yang memiliki keadaan yang beresiko, yaitu bagian Pekerjaan Dalam Keadaan Bertegangan (PDKB). Dimana Dalam fungsinya memiliki procedure pekerjaan standar yang telah ditentukan oleh PT. PLN (Persero) Area Pelayanan dan Jaringan Malang

Dalam menerapkan kegiatan kesehatan, keselamatan dan keamanan kerja dibidang Pekerjaan Dalam Keadaan Bertegangan (PDKB), bidang

distribusi akan selalu berkoordinasi dengan bidang lain untuk mendapatkan informasi mengenai kelayakan suatu pekerjaan yang akan dikerjakan oleh pelaksana pekerjaan. Dan bisa mengetahui standard kelayakan pekerjaan apa saja yang harus diterapkan, dan siapa yang harus mengerjakan berdasarkan informasi dari bidang-bidang yang bersangkutan.

Begitu juga pada pihak pegawai (baik yang langsung terjun ke lapangan ataupun tidak), akan memakan waktu yang lama untuk menerima surat perintah kerja apabila koordinasi pekerjaan terlalu lama, sehingga mengurangi efisiensi kerja. Dan juga prosedur kelayakan kerja harus dengan pendokumentasian yang terlalu banyak dan memungkinkan ketidakteelitian dalam mengikuti prosedur kerja yang telah ditetapkan.

Koordinasi yang dilakukan oleh bidang distribusi sering kali dilaksanakan dengan komunikasi lisan, dengan pencatatan data berdasarkan laporan-laporan lisan maupun tertulis dari bidang-bidang yang bersangkutan sehingga pendokumentasian harus tertata dengan baik, agar laporan informasi sebelumnya tidak hilang pada saat pemeriksaan ulang. Masalah yang muncul adalah sulitnya melacak dan memeriksa data kelayakan kerja yang telah disusun sebelumnya. Asisten Manajer (ASMAN) akan kesulitan menentukan siapa saja yang akan dikirim untuk melaksanakan pekerjaan, standard kelayakan apa saja yang harus diterapkan, dan apa saja yang harus dikerjakan dalam pekerjaan tersebut. Sehingga ASMAN harus meminta informasi data ulang ke bagian bidang-bidang yang terkait.

Sistem informasi ini sangat mendukung bila dibangun berbasis web sehingga dengan kepadatan jadwal dan tugas para ASMAN, dimanapun mereka berada dapat memperoleh informasi mengenai diklat. Untuk menjamin bahwa data yang ada hanya boleh diubah oleh mereka yang berhak maka diberi pembatasan-pembatasan kewenangan yang dikelola oleh seorang administrator. Yang berhak memiliki akses pada aplikasi ini hanya para Asman bidang dan administrator itu sendiri.

Sistem informasi ini akan dibangun dengan teknologi basis data MySQL 5.0.51a, dan bahasa pemrograman pada sisi *server* yang digunakan adalah PHP 5.

PHP 5 merupakan teknologi disisi *server* yang digunakan untuk pengembangan web dinamis dan merupakan bahasa pemrograman berbasis object. Hal ini dapat dibuktikan dengan dapat ditemukannya empat prinsip dasar pemrograman berbasis object yaitu enkapsulasi, pewarisan sifat, penyembunyian informasi, dan polimorfisme.

Pemodelan UML (*Unified Modelling Language*) merupakan suatu standart dalam industri untuk visualisasi, merancang, dan mendokumentasikan rancangan sistem perangkat lunak dengan bahasa pemrograman berbasis object. Atas pertimbangan tersebut maka pemodelan UML dipilih sebagai metode pemodelan dalam melakukan pengembangan aplikasi ini terkait dengan teknologi PHP 5 yang digunakan.

2. Rumusan Masalah

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang, maka rumusan masalah yang diberikan adalah:

1. Merancang dan membuat sebuah aplikasi sistem informasi manajemen kesehatan, keselamatan dan keamanan kerja yang berbasis web dengan menggunakan bahasa pemrograman server-side PHP 5 yang berbasis object dan Basis data MySQL 5.0.51a.
2. Mengimplementasikan sistem informasi secara *online* ke dalam jaringan intranet.

3. Batasan Masalah

Perencanaan dan pembuatan skripsi ini hanya dibatasi oleh hal – hal sebagai berikut:

1. Sistem informasi dibuat dengan menggunakan Sistem Operasi Microsoft Windows XP Professional Service Pack 2, XAMPP for Windows Version 1.6.7 yang mencakup program aplikasi, Web Server Apache 2.2.4 (Win32), database MySQL 5.0.51a dan bahasa pemrograman PHP 5.
2. Pembahasan difokuskan pada perancangan dan pembuatan sistem informasi yang berbasis *web* dan berkomunikasi dengan sistem database MySQL.
3. Data yang digunakan adalah data pada perusahaan PT. PLN (Persero) area pelayanan dan jaringan malang.
4. Pembuatan sistem informasi difokuskan pada kesehatan, keselamatan, dan keamanan kerja, dibidang Pekerjaan Dalam Keadaan Bertegangan (PDKB).

4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk membuat sistem informasi Kesehatan, keselamatan, dan keamanan kerja yang berbasis *web* dengan menggunakan bahasa pemrograman PHP dan sistem basis data MySQL yang dapat digunakan oleh perusahaan tersebut untuk meningkatkan kinerja system dalam perusahaan.

5. Manfaat

Manfaat yang diharapkan dari penyusunan skripsi ini adalah dapat membantu dan memudahkan bagi bidang-bidang yang terkait dalam divisi Pekerjaan Dalam Keadaan Bertegangan (PDKB) pada PT. PLN (Persero) Area Pelayanan dan Jaringan Malang dalam menerapkan kesehatan, keselamatan, dan keamanan kerja.

Manfaat bagi penulis adalah mendapatkan kesempatan untuk mempraktekan proses rekayasa perangkat lunak dan perancangan basis data yang sebelumnya telah dipelajari, mempelajari lebih dalam mengenai UML (*Unified Modelling Language*), memahami lebih dalam bahasa pemrograman PHP.

1.6. Sistematika Penulisan

Sistematika Penulisan dalam skripsi ini sebagai berikut :

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi pembahasan, dan sistematika pembahasan.

BAB II Dasar Teori

Membahas teori-teori yang mendukung dalam pengembangan sistem informasi berbasis web.

BAB III Metode Penelitian

Berisi tentang metode penelitian yang digunakan dalam perancangan dan pengujian sistem.

BAB IV Perancangan

Membahas tentang pelaksanaan analisis kebutuhan dan perancangan perangkat lunak.

BAB V Implementasi

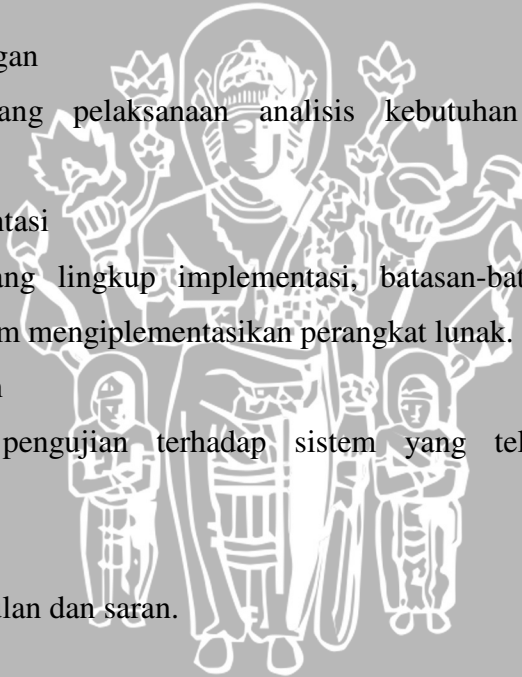
Membahas tentang lingkup implementasi, batasan-batasan implementasi, kendala serta solusi dalam mengimplementasikan perangkat lunak.

BAB VI Pengujian

Memuat hasil pengujian terhadap sistem yang telah dirancang dan diimplementasikan.

BAB VII Penutup

Memuat kesimpulan dan saran.



BAB II DASAR TEORI

Bab ini menjelaskan mengenai teori dasar yang digunakan untuk menunjang penulisan skripsi ini. Teori dasar yang digunakan yaitu, konsep dasar Sistem Informasi Manajemen (SIM), perancangan sistem informasi berbasis web ,basis data, serta teori dasar *web*.

2.1 Konsep Dasar Sistem Informasi Manajemen Kesehatan Keselamatan, dan Keamanan Kerja

2.1.1 Sistem Kesehatan, keselamatan, dan keamanan kerja

Sistem kesehatan, keselamatan, dan keamanan kerja adalah bagian dari sistem manajemen perusahaan secara keseluruhan yang dibutuhkan bagi pengembangan, penerapan, pencapaian, pengkajian dan pemeliharaan kewajiban K3, dalam rangka pengendalian resiko yang berkaitan dengan kegiatan kerja guna terciptanya tempat kerja yang aman, efisien dan produktif.

2.1.2 Pengertian Kesehatan, keselamatan, dan keamanan kerja

2.1.2.1 Kesehatan

Upaya Kesehatan Kerja adalah upaya penyesuaian antara kapasitas kerja, beban kerja dan lingkungan kerja agar setiap pekerja dapat bekerja secara sehat tanpa membahayakan dirinya sendiri maupun masyarakat di sekelilingnya, agar diperoleh produktivitas kerja yang optimal (UU Kesehatan Tahun 1992 Pasal 23).

Derajat/tingkat keadaan fisik dan psikologi individu (the degree of physiological and psychological well being of the individual) [ANO-06].

Ruang Lingkup Kesehatan Kerja

Kesehatan Kerja meliputi berbagai upaya penyesuaian antara pekerja dengan pekerjaan dan lingkungan kerjanya baik fisik maupun psikis dalam hal cara/metode kerja, proses kerja dan kondisi yang bertujuan untuk :

1. Memelihara dan meningkatkan derajat kesehatan kerja masyarakat pekerja di semua lapangan kerja setinggi-tingginya baik fisik, mental maupun kesejahteraan sosialnya.
2. Mencegah timbulnya gangguan kesehatan pada masyarakat pekerja yang diakibatkan oleh keadaan/kondisi lingkungan kerjanya.
3. Memberikan pekerjaan dan perlindungan bagi pekerja di dalam pekerjaannya dari kemungkinan bahaya yang disebabkan oleh faktor-faktor yang membahayakan kesehatan.
4. Menempatkan dan memelihara pekerja disuatu lingkungan pekerjaan yang sesuai dengan kemampuan fisik dan psikis pekerjanya [ANO-06].

2.1.2.2 Keamanan

Aman (safe) adalah suatu kondisi dimana atau kapan munculnya sumber bahaya telah dapat dikendalikan ke tingkat yang memadai, dan ini adalah lawan dari bahaya (danger).

Keamanan kerja berhubungan erat dengan kenyamanan ataupun kepuasan pegawai dalam melakukan pekerjaannya. Sehingga faktor keamanan kerja tidak hanya bisa mempengaruhi kenyamanan karyawan dalam bekerja, juga bisa memberikan efek yang cukup baik perusahaan tempat karyawan bekerja, Sehingga kedua pihak tidak ada yang dirugikan.

Dalam kaitan kewanja, perusahaan memiliki standar keamanan tertentu dalam tiap-tiap bidang pekerjaan dalam perusahaan. Tidak lain adalah untuk meningkatkan efisien kerja, yang berkaitan dengan timbulnya rasa nyaman, meningkatnya kehandalan kerja, dan meningkatnya produktifitas kerja [ANO-06].

2.1.2.3 Keselamatan :

- Mengendalikan kerugian dari kecelakaan (control of accident loss)
- Kemampuan untuk mengidentifikasi dan menghilangkan (mengontrol) resiko yang tidak bisa diterima (the ability to identify and eliminate unacceptable risks)

Manajemen keselamatan kerja adalah suatu upaya untuk mengelola penerapan aspek keselamatan kerja berkesinambungan sehingga kegiatan operasi perusahaan dapat berjalan dengan aman, andal, efisien dan berwawasan lingkungan [ANO-06].

Bagian manajemen akan menyadari keselamatan pekerja pada saat bekerja sangat penting untuk dikemukakan. Ditandai dengan banyaknya biaya tidak langsung yang baru dibahas dari hasil kecelakaan. Hasil efisiensi dari pergantian sementara dan turunnya efektivitas dari pekerja yang menjadi korban kecelakaan tidak tergantung pada dimana atau kapan penderitaan yang dialaminya. Walaupun pekerja tidak di gaji karena absen disebabkan sakit, ketidakhadiran pekerja dapat menyebabkan berkurangnya pendapatan perusahaan [ANO-06].

Sebab-sebab Kecelakaan

Kecelakaan tidak terjadi begitu saja, kecelakaan terjadi karena tindakan yang salah atau kondisi yang tidak aman. Kelalaian sebagai sebab kecelakaan merupakan nilai tersendiri dari teknik keselamatan. Ada pepatah yang mengungkapkan tindakan yang lalai seperti kegagalan dalam melihat atau berjalan mencapai suatu yang jauh diatas sebuah tangga. Hal tersebut

menunjukkan cara yang lebih baik selamat untuk menghilangkan kondisi kelalaian dan memperbaiki kesadaran mengenai keselamatan setiap karyawan [ANO-06].

Diantara kondisi yang kurang aman salah satunya adalah pencahayaan, ventilasi yang memasukkan debu dan gas, layout yang berbahaya ditempatkan dekat dengan pekerja, pelindung mesin yang tak sebanding, peralatan yang rusak, peralatan pelindung yang tak mencukupi, seperti helm dan gudang yang kurang baik [ANO-06].

Diantara tindakan yang kurang aman salah satunya diclassifikasikan seperti latihan sebagai kegagalan menggunakan peralatan keselamatan, mengoperasikan pelindung mesin mengoperasikan tanpa izin atasan, memakai kecepatan penuh, menambah daya dan lain-lain. Dari hasil analisa kebanyakan kecelakaan biasanya terjadi karena mereka lalai ataupun kondisi kerja yang kurang aman, tidak hanya satu saja. Keselamatan dapat dilaksanakan sedini mungkin, tetapi untuk tingkat efektivitas maksimum, pekerja harus dilatih, menggunakan peralatan keselamatan [ANO-06].

2.1.3 Sistem Kesehatan, keselamatan, dan keamanan kerja pada PT.PLN (Persero) Area pelayanan Dan Jaringan Malang

Dalam pemrosesan penerapan Kesehatan, keselamatan, dan keamanan kerja Asisten Manager(ASMAN) selaku pengambil keputusan akan memproses kegiatan tersebut berdasarkan data yang didapatkan dari bagian bidang-bidang yang terkait. Dengan sistem yang efektif akan mendukung keputusan manajemen, pendokumentasian laporan yang terstruktur, penentuan penugasan kerja karyawan, dan efisiensi waktu. Tujuan Spesifik dari Sistem Kesehatan, keselamatan, dan keamanan kerja adalah:

- a. Memproses data kelayakan kerja karyawan berdasarkan data dari tiap-tiap bidang yang terkait

- b. Menerapkan Prosedure kelayakan kerja karyawan dilihat dari data karyawan yang telah tercantum didalam sistem
- c. Penentuan keputusan kerja oleh Asisten Manajer(ASMAN) kepada karyawan dalam bagian-bagian kerja yang sesuai dengan kondisi karyawan
- d. Penggunaan waktu yang efisien, sehingga bisa meningkatkan kinerja perusahaan
- e. Pencatatan laporan kerja yang cepat dan terstruktur dengan baik

2.1.3.1 Pekerjaan Dalam Keadaan Bertegangan (PDKB) pada PT.PLN (Persero) Area pelayanan Dan Jaringan Malang

Dalam kaitannya dengan Kesehatan, keselamatan dan keamanan kerja, PT. PLN (Persero) Area Pelayanan dan Jaringan Malang memiliki salah satu bagian yang memiliki keadaan yang beresiko, yaitu bagian Pekerjaan Dalam Keadaan Bertegangan (PDKB). Dimana Dalam fungsinya memiliki procedure pekerjaan standar yang telah ditentukan oleh PT. PLN (Persero) Area pelayanan Dan Jaringan Malang, PDKB dalam pekerjaannya memiliki Buku laporan, Buku Data Karyawan, Surat pengerjaan.

Dalam PDKB tidak hanya difokuskan pada keamanan dan keselamatan kerja saja, namun juga diperhatikan mengenai kesehatan pekerja. Dan ini merupakan syarat awal pemeriksaan dalam suatu pengerjaan PDKB. Dan apabila syarat awal tidak terpenuhi, maka pekerjaan akan ditangguhkan sementara, hingga mendapatkan karyawan pengganti.

Dari Pemeriksaan kesehatan, berlanjut masuk ke dalam procedure keamanan dan keselamatan yang telah ditetapkan sebelumnya. Prosedure kerja ini juga memiliki tahapan-tahapan proses yang harus dijalankan dan dipatuhi, agar Keamanan dan Keselamatan pekerjaan dapat terpenuhi dengan baik.

Jika ketiga persyaratan sudah dipenuhi, maka proses pengerjaan PDKB dapat dilaksanakan, dengan resiko yang sangat kecil, bahkan mendekati resiko 0%. Karena dengan prosedur – prosedur standart yang telah ditetapkan, dapat menekan angka resiko terjadinya kecelakaan.

2.2 Teori dasar perancangan sistem informasi berbasis web

2.2.1 Object Oriented Development

Adalah suatu cara pengembangan perangkat lunak dan system informasi berdasarkan *abstraksi* objek-objek yang berada di dunia nyata. *Abstraksi* adalah menemukan serta memodelkan fakta-fakta dari suatu objek yang penting bagi suatu aplikasi. Dengan metode pemrograman yang disebut dengan Object Oriented Programming, Object Oriented programming adalah suatu cara baru untuk berfikir dan berlogika dalam menghadapi masalah-masalah yang akan dicoba atasi dengan bantuan komputer.

Object Oriented programming memiliki Standard bahasa pemodelan yang disebut dengan UML (Unified Modelling Language). UML menyediakan bermacam teknik dengan notasi grafik (diagram) memungkinkan developers untuk menyajikan system model software dalam bentuk yang berbeda dan dari sudut pandang yang berbeda. Meskipun UML terdiri dari 12 teknik, tetapi tidak seluruhnya diaplikasikan didalam project sistem, penggunaannya hanya sebagian saja terkait dengan kebutuhan system. Berikut akan dijelaskan beberapa teknik, terkait dengan tugas akhir ini.

Perancangan Berbasis Object (UML)

Unified Modelling Language merupakan suatu standart dalam industri untuk visualisasi, merancang, dan mendokumentasikan rancangan sistem perangkat lunak. UML sangat cocok untuk perancangan piranti lunak dengan bahasa pemrograman yang berbasis object karena memanfaatkan class dan operasi pada konsep dasarnya.

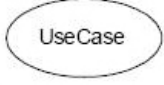


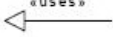
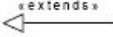
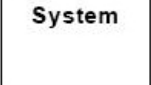
Beberapa kelebihan pengembangan perangkat lunak berbasis object antara lain [SHO-07]:

1. Relatif lebih sederhana karena antara data dan fungsi dapat dikemas menjadi satu object yang utuh. Dalam object tersebut terdapat data yang ingin dikelola dan juga metode yang melakukan pengelolaan, Jika ada object lain yang ingin menggunakan data dari object tersebut tidak perlu tahu bagaimana data tersebut dikelola melainkan hanya langsung berkomunikasi dengan metode pengakses datanya saja.
2. Pendekatan berorientasi object mencakup beberapa fase dalam pengembangan sistem. Sehingga aplikasi dapat dirancang dengan lebih baik karena melalui fase-fase yang telah ditentukan.
3. Pendekatan berorientasi object yang baik memungkinkan penggunaan ulang kode-kode dari class yang telah dibuat sehingga waktu yang dibutuhkan untuk pengembangan perangkat lunak menjadi relatif lebih singkat dan menjadi lebih handal.

2.2.1.1 Use-case Diagram

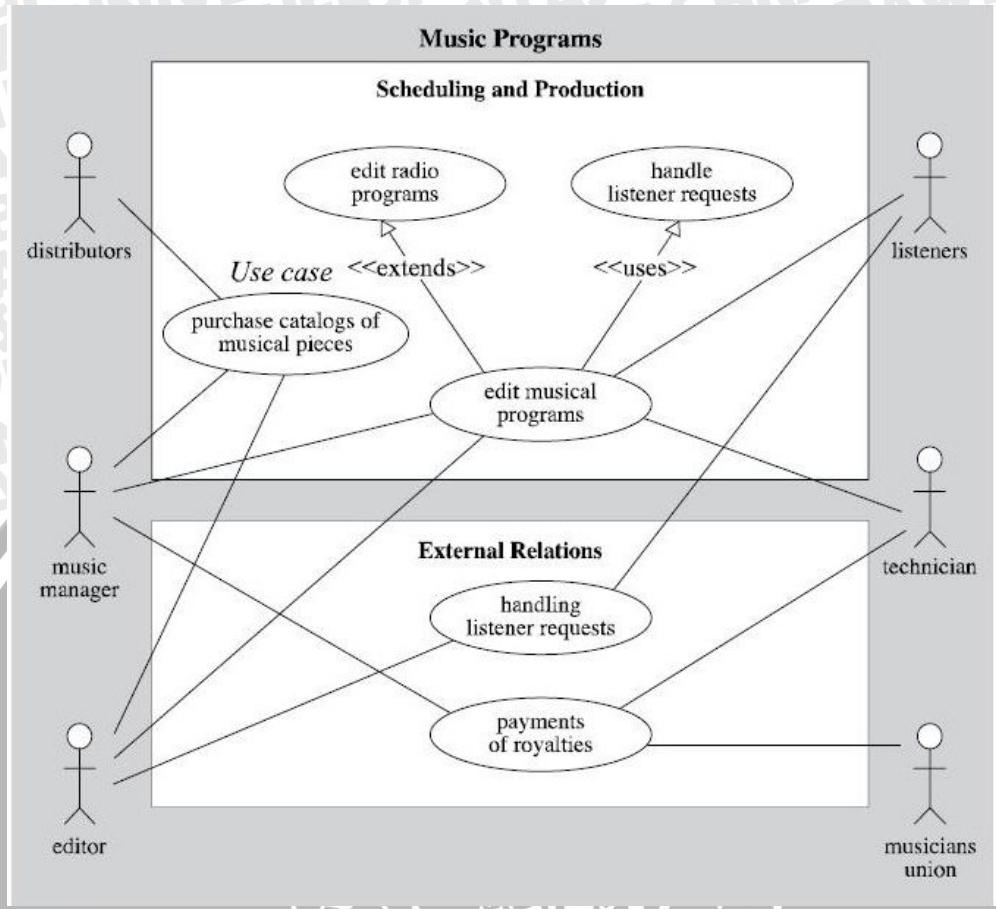
Diagram *Use-case* menunjukkan interaksi antara system satu dengan system lainnya maupun dengan user system, yang disebut dengan aktor. Diagram ini menggambarkan siapa yang akan menggunakan system dan bagaimana user berinteraksi dengan system untuk mencapai tujuan yang diinginkan. Biasanya, diagram disertai dengan deskripsi narasi yang dijelaskan secara detil dengan cara yang terstruktur untuk langkah-langkah setiap interaksi. Teknik ini memungkinkan menciptakan/membuat deskripsi inisial kebutuhan user, sehingga nantinya perilaku system bisa ditetapkan dengan menggunakan arti yang lain. Contohnya, sequence

atau collaboration diagram. Tabel 2.1 Menunjukkan komponen-komponen di dalam *use-case*

Notation	Meaning
	Use-case : Gambar ellipse menunjukkan single <i>use-case</i> . Nama <i>use-case</i> dituliskan di dalam ellipse
	Aktor : menunjukkan setiap element yang berhubungan dengan system. Bisa digambarkan User atau system lain yang berhubungan dengan penjelasan system didalam <i>use-case</i>
	Ordinary Relationship : Menunjukkan koneksi, sebagai saluran transfer informasi antara actor dengan <i>use-case</i> , ato <i>use-case</i> dengan <i>use-case</i>
	Uses Relationship : menunjukkan ketergantungan <i>use-case</i> , yaitu <i>use-case</i> satu menggunakan <i>use-case</i> yang lain.
	Extends Relationship : <i>Use-case</i> complex, yang banyak memiliki aktifitas yang sulit dipahami, dan bisa berubah menjadi siple <i>use-case</i>
	System Boundary : Kotak menunjukkan batas system. Didalam kotak adalah <i>use-case</i> , dan bagian luar adalah actor yang berhubungan dengan system.

Tabel 2.1 Komponen Use-case Diagram

Gambar 2.1 memperlihatkan diagram *use-case* untuk contoh program music. Diagram tersebut terdiri dari 6 *use-case*, yang dibagi dalam 2 subsistem, yang masing-masing subsistemnya berisi : **Assignment and Productions** dan **External Relations**. Pembagian ini memungkinkan 2 team mengembangkan system. Seperti yang bisa di lihat subsistem “**Assignment and Productions**“ terdiri atas 4 *use-case*, termasuk “**Edit Musical Program**” *use-case* ini melanjutkan *use-case* “**Edit Radio Program**” dan menggunakan “Handle Listener Request” Subsistem “External Relations” terdiri atas 2 *use-case*, Sistem secara keseluruhan memiliki 6 aktor.






Gambar 2.1 Use-case diagram untuk contoh music programs

2.2.1.2 Class Diagram (statis)

Class Diagram digunakan untuk menguraikan system struktur static. Yang menunjukkan class object, atribut, methods, dan bermacam tipe hubungan. Class diagram digunakan dalam seluruh metodologi OO Development.

Class Dibagi dalam tiga bagian : Bagian atas – Nama Class, Bagian Tengah – Atribut Class, Bagian Bawah – Fungsi Class. Hubungan antar class ditunjukkan dengan garis penghubung. Tabel 2.2 dan 2.3 menunjukkan notasi UML

Notation	Meaning						
<table border="1"> <tr><td>ClassName</td></tr> <tr><td>-attribute : String</td></tr> <tr><td>+doOperation(in Parameter : 1 Double ; Integer)</td></tr> </table>	ClassName	-attribute : String	+doOperation(in Parameter : 1 Double ; Integer)	Kelas : Dibagi dalam 3 bagian : (1) Nama Kelas, (2) Atribut: Setiap atribut Memiliki nama dan tipe data. Dan (3) Fungsi : Setiap fungsi memiliki nama , parameter, dan tipe data			
ClassName							
-attribute : String							
+doOperation(in Parameter : 1 Double ; Integer)							
<table border="1"> <tr> <td>-Role1</td> <td>-Role2</td> </tr> <tr> <td>_____</td> <td>_____</td> </tr> <tr> <td>*..1</td> <td>5..2</td> </tr> </table>	-Role1	-Role2	_____	_____	*..1	5..2	Ordinary Relationship : Hubungan ini ditunjukkan dengan garis; multiplisitas ditunjukkan diakhir garis
-Role1	-Role2						
_____	_____						
*..1	5..2						
	Inheritance : Simbol ini menunjukkan hubungan superkelas dan subkelas. arah panah menunjukkan superkelas						
	Composite Aggregation : Simbol ini menunjukkan hubungan "Seluruh - bagian". Bentuk kepala garis panah menuju "seluruh" kelas bentuk panah berwarna hitam menunjukkan satu unit dan tidak bisa dipisahkan (ex: badan manusia).						
	Shared Aggregation : Simbol ini menunjukkan hubungan "Seluruh Bagian". Perbedaannya adalah bagian dari shared agregation : bisa menunjuk ke "wholes" yang berbeda.						

Tabel 2.2 Komponen Class Diagram

Type	Notation	Example	Meaning																		
Exactly 1	1 or nothing	<table border="1"> <tr><td>Teacher</td><td>teaches</td><td>Course</td></tr> <tr><td>_____</td><td></td><td>_____</td></tr> <tr><td></td><td></td><td>1</td></tr> </table>	Teacher	teaches	Course	_____		_____			1	Guru hanya mengajar satu kursus - tidak ada lagi.									
Teacher	teaches	Course																			
_____		_____																			
		1																			
0 or 1	0..1	<table border="1"> <tr><td>Teacher</td><td>teaches</td><td>Course</td></tr> <tr><td>_____</td><td></td><td>_____</td></tr> <tr><td></td><td></td><td>0..1</td></tr> </table>	Teacher	teaches	Course	_____		_____			0..1	Guru hanya mengajar satu kursus atau tidak sama sekali									
Teacher	teaches	Course																			
_____		_____																			
		0..1																			
0 or more	0..* or *	<table border="1"> <tr><td>Teacher</td><td>teaches</td><td>Course</td></tr> <tr><td>_____</td><td></td><td>_____</td></tr> <tr><td></td><td></td><td>0..*</td></tr> </table> <table border="1"> <tr><td>Teacher</td><td>teaches</td><td>Course</td></tr> <tr><td>_____</td><td></td><td>_____</td></tr> <tr><td></td><td></td><td>*</td></tr> </table>	Teacher	teaches	Course	_____		_____			0..*	Teacher	teaches	Course	_____		_____			*	Guru mengajar banyak kursus atau tidak sama sekali.
Teacher	teaches	Course																			
_____		_____																			
		0..*																			
Teacher	teaches	Course																			
_____		_____																			
		*																			
1 or more	1..*	<table border="1"> <tr><td>Teacher</td><td>teaches</td><td>Course</td></tr> <tr><td>_____</td><td></td><td>_____</td></tr> <tr><td></td><td></td><td>1..*</td></tr> </table>	Teacher	teaches	Course	_____		_____			1..*	Guru mengajar banyak kursus tetatapi minimal 1.									
Teacher	teaches	Course																			
_____		_____																			
		1..*																			
A defined range of numbers	3..5	<table border="1"> <tr><td>Teacher</td><td>teaches</td><td>Course</td></tr> <tr><td>_____</td><td></td><td>_____</td></tr> <tr><td></td><td></td><td>3..5</td></tr> </table>	Teacher	teaches	Course	_____		_____			3..5	Guru hanya mengajar antara 3 sampai 5 kursus									
Teacher	teaches	Course																			
_____		_____																			
		3..5																			

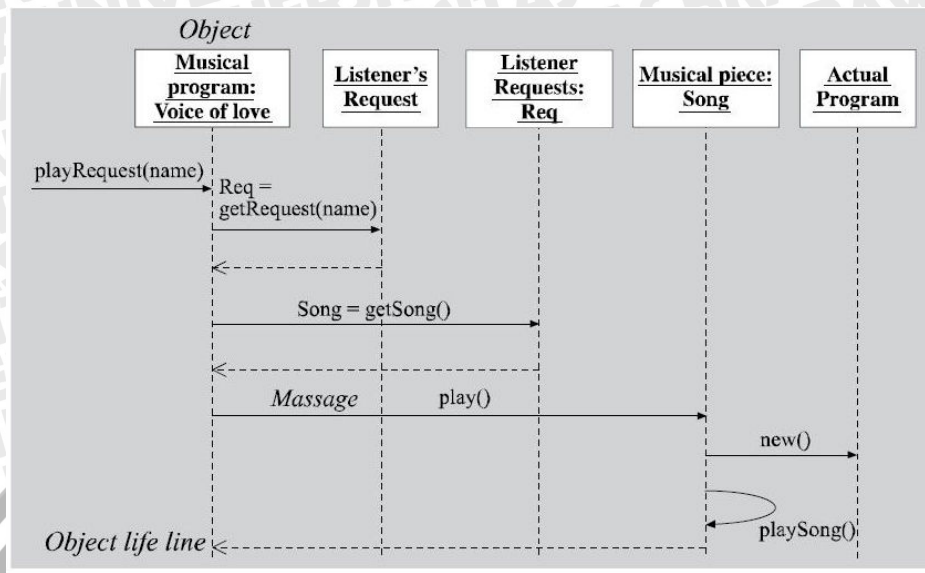
Tabel 2.3 Multiplicity Constraints dalam UML

2.2.1.3 Sequence Diagram (dinamis)

Sequence diagram menggambarkan sebuah skenario interaksi antar object dalam *use-case*. Interaksinya digambarkan oleh message yang terkirim dari 1 object ke object lainnya. Sequence diagram bisa digunakan sebagai cara untuk mengetahui metode apa yang perlu dilampirkan class-class object.

Sequence diagram bisa digunakan untuk menggambarkan interaksi antara system dan element external dalam kasus ini disebut dengan “Black Box Diagram”, dan ini melengkapi diagram *use-case*. Diagram ini menggambarkan, setiap *use-case*-nya, subevent yang dibuat, permintaan event dan reaksi system yang mungkin terjadi. Satusatu nya komponen dalam diagram adalah elemen external dan system.

Penggunaan sequence diagram yang lain adalah untuk menggambarkan interaksi antara objek system dan message yang ditukar. Message-message tersebut diminta secara kronologis. Setiap object yang ada memiliki “Life line” (ditunjukkan dengan garis putus-putus dibawah kotak yang menandakan object) Menandakan bahwa objek sedang aktif selama event berjalan. Message sebelumnya ditunjukkan dengan tanda panah, dimana terhubung dengan garis objek lain. Apabila message sudah terkirim, berarti objek yang sudah di terima memiliki fungsi yang penting untuk melakukan layanan kebutuhan.



Gambar 2.2 Sequence diagram dari penanganan request Music

Object Oriented Programming memiliki 4 macam karakteristik :

2.2.1.4 Pembungkusan (Encapsulation)

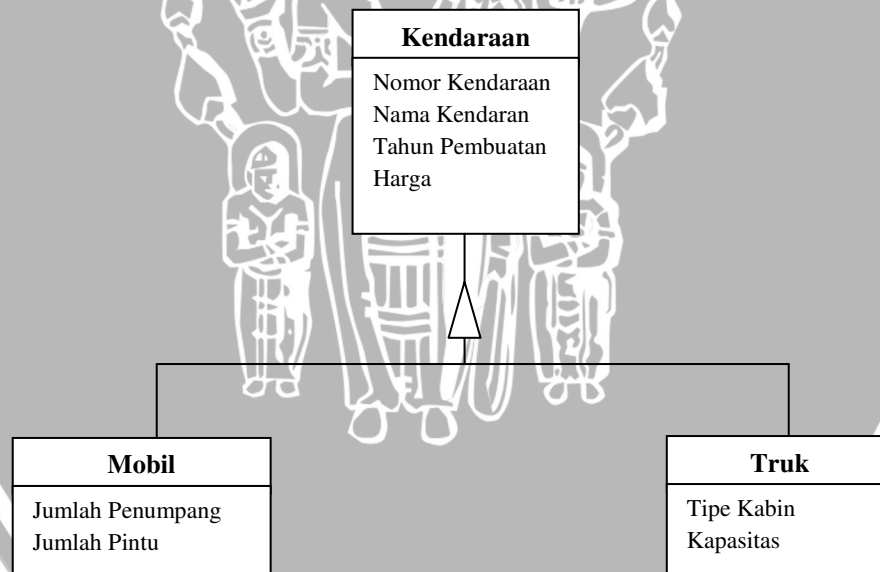
Pembungkusan (penyembunyian Informasi) berarti meninggalkan aspek eksternal dari objek yang dapat dimasuk (diakses) oleh objek lain dan memfokuskan diri pada implementasi internal suatu objek. Rincian implementasi internal dari suatu objek tersembunyi dari objek-objek lain dan terpisah dari implementasi eksternal, yaitu antarmuka (interface) satu objek dengan objek lainnya. Oleh karena itu, implementasi internal suatu objek dapat dirubah tanpa mempengaruhi aplikasi yang menggunakannya. Pembungkusan sebenarnya tidak unik pada pemrograman berorientasi objek tapi kemampuannya untuk menggabungkan struktur data dan perilaku (baca: fungsi atau prosedur) dalam suatu entitas tunggal membuat bahasa berorientasi objek lebih berdaya guna dibandingkan bahasa konvensional yang memisahkan struktur data dan perilaku.

2.2.1.5 Penyembunyian Informasi (Information Hiding)

Penyembunyian implementasi mengacu perlindungan implementasi internal objek. Objek disusun dari antarmuka public dan bagian private yang merupakan kombinasi data dan method internal. Manfaat utama adalah bagian internal dapat berubah tanpa mempengaruhi bagian-bagian program yang lain.

2.2.1.6 Generalisasi dan Pewarisan (Inheritance)

Generalisasi serta Pewarisan adalah suatu cara yang sangat berdayaguna untuk berbagi apa yang dimiliki suatu class (atau objek) bagi class-class (atau objek-objek) yang lain. Misalkan, kita ambil contoh class kendaraan bermotor. Mobil, truk, dan lain-lain bisa berbagi atribut yang sama, misalnya : atribut model, tahun pembuatan, jumlah gigi transmisi, dan sebagainya.



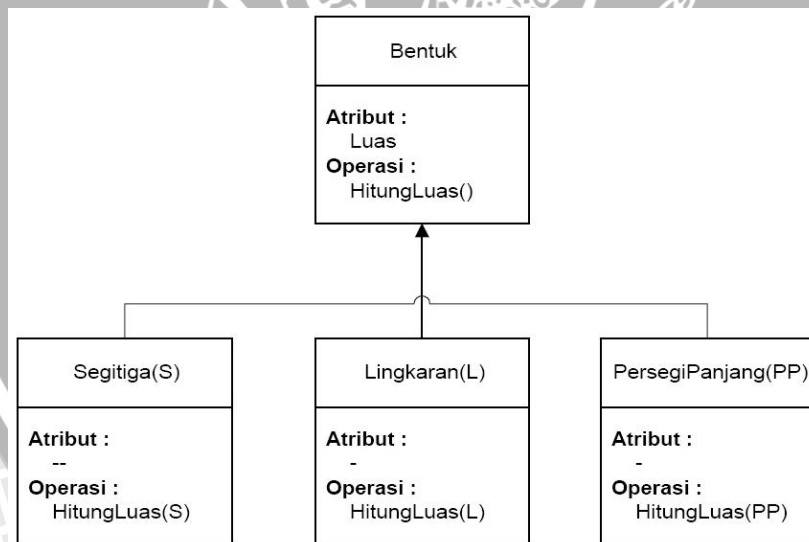
Gambar 2.3 Pewarisan Atribut dari Class Pendahulu ke Class Turunan

Dari keterangan gambar diatas dapat dijelaskan bahwa generalisasi adalah proses “bottom-up” (“bawah ke atas”) sedangkan spesialisasi adalah

proses “*top-bottom*” (“atas ke bawah”) dimana pewarisan adalah berbagi atribut yang sama diantara dua class dibawahnya (mobil dan truk) dalam class yang lebih atas (kendaraan). Perlu diperhatikan bahwa pewarisan memungkinkan atribut-atribut yang cukup dituliskan sekali saja pada superclass dan tidak perlu ditulis ulang pada subclass yang mewarisi atribut-atribut yang sama itu.

2.2.1.7 Polimorfisme (Polymorphism)

Polymorphism berasal dari bahasa Yunani yang berarti banyak bentuk. Dalam PBO, konsep ini memungkinkan digunakannya suatu interface yang sama untuk memerintah objek agar melakukan aksi atau tindakan yang mungkin secara prinsip sama namun secara proses berbeda. Dalam konsep yang lebih umum sering kali polymorphism disebut dalam istilah satu interface banyak aksi.



Gambar 2.4 Pewarisan dari Atribut yang memiliki Class berbeda

Pada contoh diatas clas dasar adalah class bentuk yang memiliki atribut berupa Luas dan operasi hitung luas, class tersebut dapat diturunkan kedalam berbagai macam class bentuk seperti Segitiga, Lingkaran, Persegi panjang. Class Segitiga, Lingkaran, Persegi panjang memiliki atribut “Luas” dari hasil penurunan class bentuk, akan tetapi operasi hitung luas pada masing masing class akan berbeda-beda, inilah yang disebut sebagai polymorphism

2.2.2 Entity-Relationship Diagram (Diagram E-R)

Diagram *Entity-Relationship* menjelaskan data-data yang ada pada dunia nyata. Diagram *Entity-Relationship* menterjemahkan atau mentransformasikan data dengan memanfaatkan sejumlah model konseptual [POH-97]. Komponen-komponen pembentuk diagram *Entity-Relationship*, antara lain:

1. Entitas dan Himpunan Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah himpunan entitas. Entitas menunjuk pada individu suatu objek, sedangkan himpunan entitas menunjuk pada rumpun dari individu tersebut. Himpunan entitas digambarkan dengan persegi panjang dalam diagram *Entity-Relationship*.

2. Atribut

Entitas pasti memiliki atribut yang mendeskripsikan karakteristik dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal penting lain dalam pembentukan model data. Atribut digambarkan dengan elips dalam diagram *Entity-Relationship*.

3. Relasi dan Himpunan Relasi

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Kumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas-himpunan entitas tersebut

membentuk suatu himpunan relasi. Himpunan relasi digambarkan dengan belah ketupat dalam diagram *Entity-Relationship*.

4. Kardinalitas atau Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain. Kardinalitas relasi yang terjadi di antara dua himpunan entitas (misal A dan B) dapat berupa:

- Satu ke Satu (*One to One*)

Entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B dan sebaliknya.

- Satu ke Banyak (*One to Many*)

Entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B tetapi tidak sebaliknya. Setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

- Banyak ke Satu (*Many to One*)

Entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B tetapi tidak sebaliknya. Setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

- Banyak ke Banyak (*Many to Many*)

Entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B dan sebaliknya.

Diagram E-R selalu dibuat bertahap. Tahapan pertama yang dapat dilakukan adalah dengan membuat diagram E-R awal. Tujuan dari pentahapan ini adalah untuk mendapatkan sebuah rancangan basis data minimal yang dapat mengakomodasi kebutuhan penyimpanan data terhadap sistem yang sedang ditinjau. Langkah-langkah yang harus dilakukan antara lain:

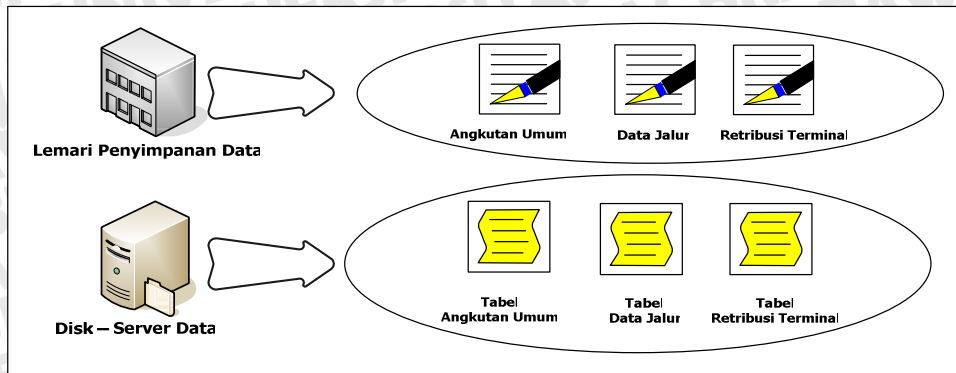
1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut key dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas-himpunan entitas yang ada beserta *foreign key*-nya.
4. Menentukan derajat atau kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non-key*).

2.3 Basis data

Basis data merupakan tempat penyimpanan kumpulan data secara elektronik. Data tersebut dapat disimpan menggunakan media penyimpanan seperti *diskette*, *harddisk*, atau *CD (Compact Disk)*. Basis data sendiri dapat didefinisikan sebagai [FAT-02] :

- 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-sama tanpa adanya duplikasi data yang tidak perlu, untuk memenuhi berbagai kebutuhan;
- Kumpulan *file*, tabel, dan arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Sebuah basis data dibentuk dari susunan beberapa komponen-komponen penting pembuatnya. Dan untuk membedakan antara basis data yang satu dengan yang lainnya, diperlukan suatu key dan atribut. *Key* adalah satu atau gabungan dari beberapa atribut yang membedakan semua baris data (*record*) dalam tabel secara unik.



Gambar 2.5 Lemari Arsip dan Basis data

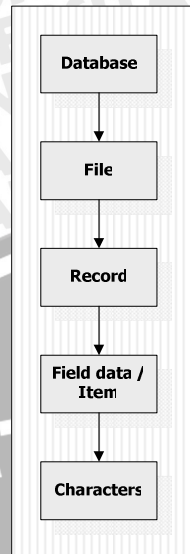
Sumber : [FAT-02:3]

Bahasa basis data standar yang sering digunakan adalah SQL. Penggunaan SQL dapat mengurangi beban kinerja jaringan. Komputer *client* yang melakukan permintaan dalam perintah SQL akan dikirimkan mengirimkan data yang dibutuhkan dalam jaringan oleh komputer *server*.

Bahasa SQL dapat membuat sebuah basis data melalui sebuah aplikasi MySQL. Perancangan serta pengujian sebuah basis data memerlukan aplikasi Sybase Power Designer 12.5.

2.3.1 Komponen Basis data

Basis data terdiri dari beberapa komponen untuk membentuk struktur basis data yang diperlukan [DHA-02]. Tingkatan dari komponen basis data ditunjukkan dalam Gambar 2.6.



Gambar 2.6 Tingkatan Komponen Basis data

Sumber : [DHA-02:102]

Tingkatan dari komponen basis data terdiri dari :

- *Characters*/Karakter
Karakter merupakan bagian data terkecil, dapat berupa karakter numerik, huruf, ataupun karakter-karakter khusus yang membentuk suatu *field*.
- *Field*
Field merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti nama, alamat, dan lain sebagainya. Kumpulan dari *field* akan membentuk suatu *record*.
- *Record*
Record menggambarkan suatu unit data tertentu, dimana kumpulan dari *record* akan membentuk suatu *file*/tabel. Misalnya kumpulan *record* data mahasiswa yang membentuk tabel mahasiswa.

- *File*

File adalah kumpulan *record-record* yang menggambarkan kesatuan data yang sama. Misalnya *file* mata pelajaran berisi tentang semua mata pelajaran yang ada.

- *Basis data*

Basis data adalah kumpulan dari *file* atau tabel.

2.3.2 Key dan Atribut Deskriptif

Key adalah satu atau gabungan dari beberapa atribut yang membedakan semua baris data (*record*) dalam tabel secara unik. Jika suatu atribut dijadikan *key*, maka tidak boleh terdapat dua data dengan nilai yang sama pada atribut tersebut. *Key* dibagi menjadi 3 macam, yaitu [FAT-02]:

- *Super Key*

Super key merupakan satu atau kumpulan dari atribut yang dapat digunakan untuk membedakan baris data dalam sebuah tabel secara unik. Misalnya pada tabel mahasiswa terdapat atribut/*field* nim, nama, alamat, dan telephone, maka kemungkinan atribut yang dapat digunakan sebagai *super key* adalah:

- nim, nama, dan alamat;
- nim, nama;
- nim;
- nama (jika dipastikan tidak ada nama yang sama).

- *Candidate Key*

Candidate key merupakan kumpulan atribut minimal yang dapat membedakan setiap data dalam sebuah tabel secara unik. *Candidate key* tidak boleh berisi atribut atau kumpulan atribut yang telah menjadi *super key*. Pada contoh tabel mahasiswa, atribut yang dapat dijadikan *candidate key* adalah:

- nim;
- nama (jika dipastikan tidak ada nama yang sama).
- **Primary Key**

Primary key merupakan atribut yang datanya dibedakan secara unik. *Primary key* hanya dapat digunakan oleh satu atribut dalam satu tabel. *Candidate key* dapat digunakan sebagai *primary key*. Pemilihan *primary key* dari sejumlah *candidate key* umumnya didasari oleh:

- *Key* tersebut lebih sering (*natural*) untuk dijadikan acuan;
- *Key* tersebut lebih ringkas;
- Jaminan keunikan *key* tersebut lebih baik.

Candidate key dari tabel mahasiswa yaitu nim dan nama yang dapat dijadikan sebagai *primary key* adalah nim.

Atribut deskriptif adalah atribut-atribut yang tidak menjadi atau merupakan anggota dari *primary key*. Atribut deskriptif pada tabel mahasiswa adalah nama, alamat, dan telephone. Atribut nim digunakan sebagai *primary key*

2.3.3 SQL (*Structured Query Language*)

SQL adalah sebuah bahasa basis data standar untuk membuat, meng-*update*, dan melakukan *query* pada sistem manajemen basis data relasional.

SQL mengalami beberapa perubahan. SQL mula-mula dibuat pada tahun 1970 dengan Sequel. Standarisasi yang pertama dibuat pada tahun 1986 oleh ANSI (*The American National Standard Institute*) dan ISO (*International Standard Organization*), yang disebut sebagai SQL-86. SQL-86 diperbarui pada tahun 1989 menjadi SQL-89. Standar terakhir yang dibuat dan digunakan hingga saat ini adalah SQL-92 yang dikeluarkan pada tahun 1992. [FAT-02]

SQL dapat digunakan untuk membuat sebuah *query* dan digabungkan dalam sebuah aplikasi sebagai instruksi-instruksi yang digunakan untuk menangani data di dalam basis data [PIN-05]

2.3.4 Basis data MySQL

MySQL merupakan salah satu RDBMS (*Relational Database Management System*) yang dapat digunakan untuk membangun aplikasi basis data berbasis *web*. MySQL memiliki beberapa keistimewaan antara lain [PUT-06]:

- *Portability*
MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Solaris* dan lain sebagainya.
- *Open Source*
MySQL didistribusikan secara *open source* sehingga dapat digunakan secara gratis.
- *Multi-User*
MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- *Performance Tuning*
MySQL memiliki kecepatan yang tinggi dalam menangani sebuah *Query*.
- *Column Types*
MySQL mendukung tipe kolom yang banyak di antaranya *signed/unsigned integer*, *float*, *double*, *char*, *varchar*.
- *Command dan Functions*
MySQL mendukung penuh perintah *select* dan *where* dalam *query*.
- *Scalability dan Limits*
MySQL mampu menangani basis data dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

- *Interface*
MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- *Connectivity*
Client dapat terhubung dengan MySQL Server menggunakan *socket* TCP/IP dari *platform* apapun. Pada sistem operasi Microsoft Windows keluarga NT (NT, 2000, XP, atau 2003), *client* dapat terhubung dengan *pipe*. Pada sistem Unix, *client* dapat terhubung dengan menggunakan *file-file socket domain* Unix.
- Antarmuka *Connector / ODBC* (MyODBC)
MyODBC memberikan dukungan MySQL untuk program-program *client* yang menggunakan koneksi ODBC (*Open Database Connectivity*).

2.3.4.1 Jenis-jenis Kolom Pada MySQL

Jenis kolom merupakan rincian dari jenis data yang diterapkan pada setiap kolom pada saat membuat sebuah tabel dengan perintah CREATE TABLE.

Jenis kolom tanggal dan waktu pada basis data MySQL ditunjukkan dalam Tabel 2.4.

Tabel 2.4 Jenis Kolom Tanggal dan Waktu

Jenis Kolom	Keterangan	Ukuran
DATE	Digunakan untuk tanggal dengan format YYYY-MM-DD.	3 byte
DATETIME	Digunakan untuk tanggal dan waktu dengan format YYYY-MM-DD hh:mm:ss.	8 byte
TIME	Digunakan untuk waktu dengan format hh:mm:ss.	3 byte
TIMESTAMP	Digunakan untuk penulisan waktu dan tanggal, dengan format YYYYMMDDhhmmss.	4 byte

YEAR	Digunakan untuk penulisan tahun dengan format YYYY.	1 byte
------	---	--------

Sumber : [ARB-04:80]

Jenis kolom Karakter pada basis data MySQL ditunjukkan dalam Tabel 2.5.

Tabel 2.5 Jenis Kolom Karakter

Jenis Kolom	Bentuk Dasar Penulisan	Keterangan	Ukuran
CHAR	Char (M)	Data string dengan lebar data yang tetap (M).	M byte
VARCHAR	Varchar (M)	Data string dengan lebar data bervariasi (M).	M byte

Sumber : [ARB-04:75]

Jenis kolom numerik pada basis data MySQL ditunjukkan dalam Tabel 2.6.

Tabel 2.6 Jenis Kolom Numerik

Jenis Kolom	Bentuk Dasar Penulisan	Keterangan	Ukuran
TINYINT	TINYINT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed</i> : -128 s/d 127 <i>Unsigned</i> : 0 s/d 255	1 byte
SMALLINT	SMALLINT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed</i> : -32768 s/d 32767 <i>Unsigned</i> : 0 s/d 65535	2 byte
MEDIUMINT	MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed</i> : -8388608 s/d 8388607	3 byte

		<i>Unsigned: 0 s/d 1677215</i>	
INT	INT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed: -128 s/d 127</i> <i>Unsigned: 0 s/d 255</i>	4 byte
BIGINT	BIGINT[(M)] [UNSIGNED] [ZEROFILL]	Bilangan bulat dengan kisaran nilai: <i>Signed: -9223372036854775808 s/d 9223372036854775807</i> <i>Unsigned: 0 s/d 18446744073709551615</i>	8 byte
FLOAT	FLOAT[(M,D)] [UNSIGNED] [ZEROFILL] atau FLOAT(4)	Bilangan pecahan presisi tunggal dengan kisaran nilai: Nilai minimum (tanpa nilai nol): $\pm 1.175494351E-38$ Nilai maksimum (tanpa nilai nol): $\pm 6.402823466E+38$	4 byte
DOUBLE	DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL] atau FLOAT(8)	Bilangan pecahan presisi ganda dengan kisaran nilai: Nilai minimum (tanpa nilai nol): $\pm 2.2250738580720E-308$ Nilai maksimum (tanpa nilai nol): $\pm 1.7976931348623157E+308$	8 byte
DECIMAL	DECIMAL[(M,D)]	Bilangan desimal, dengan nilai tergantung pada besaran M dan D.	M byte

Sumber : [ARB-04:71-72]

Jenis kolom dapat dijelaskan sebagai berikut (Kode M merupakan nilai maksimum yang diperkenankan (1-255) dan D merupakan jumlah digit setelah tanda titik pada bilangan pecahan (kisaran nilai 0-30)) [ARB-04].

2.4 Teori Dasar Web

Aplikasi berbasis *web* membutuhkan bahasa pemrograman dan program editor HTML agar *web* lebih dinamis dan interaktif. Bahasa pemrograman yang digunakan adalah HTML (*Hyper Text Markup Language*), PHP (*PHP Hypertext Preprocessor*) dan program editor Macromedia Dreamweaver MX 2004.

2.4.1 HTML (Hyper Text Markup Language)

Suatu style tata penulisan khusus yang digunakan dalam penyusunan dokumen web di *World Wide Web*, dengan tata penulisan hiperteks, dokumen web dapat berisikan baik informasi format font, ukuran, layout gambar, suara, maupun *link* hubungan kesumber-sumber lain di internet. *Link* dapat merupakan hubungan ke dokumen web lain, maupun ke site lain di internet [MUL-07].

XHTML

XHTML merupakan pengembangan dari HTML. Struktur XHTML pada prinsipnya hampir sama dengan HTML, hanya lebih terstruktur dan konsisten. Berikut ini beberapa perbedaan antara HTML dan XHTML:

- Dalam HTML, penggunaan tag `<html>`, `<head>`, `<body>`, DOCTYPE tidak ada pengaruhnya. Artinya dalam HTML tag-tag tersebut sifatnya optional (boleh ada, boleh tidak) dan tidak berpengaruh pada tampilan di browser. Namun dalam XHTML tag-tag tersebut harus ada. Apabila ketentuan tersebut dilanggar, maka akan terjadi efek tertentu pada tampilan di browser.
- Dalam HTML, tag-tag penutup misalnya `</p>` atau `<i/>` dll, Sifatnya optional. Namun pada XHTML tag penutup harus ada.
- Dalam HTML tag-tag yang berdiri sendiri (tanpa tag penutup), misalnya `` tidak ada slash penutup. Sedangkan dalam XHTML slash penutup harus ada missal : ``.

- Dalam HTML, nilai semua atribut dalam tag tidak harus diapit dengan tanda petik ganda (double quote), misal penulisan `` diperbolehkan. Namun dalam XHTML double quote harus mengapit nilai atribut, misal ``.
- Dalam HTML, penulisan semua tag boleh dalam huruf besar atau huruf kecil, misal `<HEAD>`. Namun dalam XHTML semua tag harus ditulis dalam huruf kecil, misal `<head>`.

Dalam HTML, nilai yang sama dengan atribut akan diabaikan, misal `<hr width=70% noshade>`. Artinya dalam HTML untuk nama atribut yang sama dengan nilainya dapat di tulis dalam satu nama saja. Namun dalam XHTML, semua atribut dan nilai harus dinyatakan secara eksplisit, misal `<hr width="70%" noshade="noshade">`.

2.4.2 PHP (*PHP Hypertext Preprocessor*)

Aplikasi berbasis *web* dibangun tidak cukup hanya menggunakan HTML. HTML memiliki kelemahan yaitu tidak dapat mengakses dan memanipulasi data di dalam *server* basis data. Kelemahan HTML dapat diatasi dengan bahasa pemrograman yang mempunyai kemampuan untuk mengakses dan memanipulasi data di dalam *server* basis data [KRI-03].

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada *server*, sedangkan yang dikirimkan ke *browser* hanya hasilnya saja [WID-03].

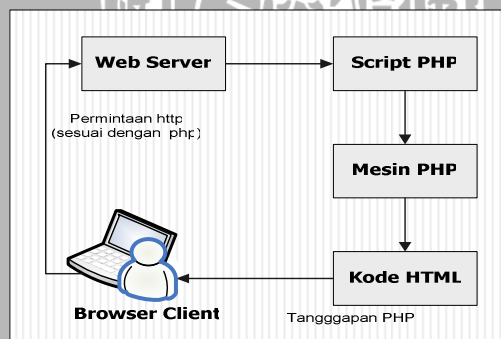
PHP dirancang secara khusus untuk membentuk *web* dinamis. PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini. PHP mempunyai fungsi yang sama dengan *script-script* seperti ASP (*Active Server Page*), Cold Fusion ataupun Perl.

PHP dirancang pada awalnya untuk diintegrasikan dengan *web server* Apache. Namun belakangan, PHP juga dapat bekerja dengan *web server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Services*) dan Xitami [KAD-02].

Untuk dapat lebih memahami tentang PHP, maka akan dibahas lebih lanjut mengenai konsep kerja PHP, variabel pada PHP, *Script* PHP, *tag* PHP, *cookie*, *session* dan keunggulan dari PHP itu sendiri.

2.4.2.1 Konsep Kerja PHP

Model kerja PHP diawali dengan permintaan suatu halaman *web* oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencari berkas yang diminta dan apabila sudah didapatkan maka *web server* akan mengirimkannya ke mesin PHP [KAD-02]. Mesin inilah yang kemudian akan memproses dan memberikan hasilnya (berupa kode HTML) ke *web server*. Selanjutnya *web server* akan menyampaikan hasil tersebut ke *client*. Skema konsep kerja PHP ditunjukkan dalam Gambar 2.8.



Gambar 2.7 Skema konsep kerja PHP Sumber : [KAD-02:6]

2.4.2.2 Variabel Pada PHP

Setiap variabel dalam PHP selalu dimulai dengan tanda dolar (\$) dan harus dimulai dengan huruf atau garis bawah (_) dan kemudian dapat diikuti oleh huruf,

angka ataupun garis bawah. `$warpspeed`, `$impuls_speed`, `$LCAR` dan `$Dilithium1` adalah contoh penamaan variabel PHP yang benar.

Jenis suatu variabel pada PHP ditentukan pada saat jalannya program (*runtime*). PHP tidak memerlukan pendeklarasian variabel terlebih dahulu karena PHP mempunyai kemampuan untuk membedakan jenis variabel secara otomatis berdasarkan konteks yang sedang berlaku bagi variabel tersebut dan tipe datanya bisa diubah sesuai dengan keinginan [IRA-03].

PHP mendukung beberapa jenis variabel sebagai berikut:

1. *Integer*

Variabel berjenis *integer* bertujuan untuk menyimpan bilangan bulat.

2. *Double*

Untuk menyimpan bilangan bernilai pecahan dan juga bilangan pemangkatan.

3. *String*

String merupakan jenis data karakter yang disimpan sebagai nomor pada memori komputer. Nilai yang disimpan adalah nilai ASCII karakter *string* tersebut.

4. *Array*

Array adalah sebuah set variabel yang mempunyai jenis data sama. *Array* mengandung komponen yang disebut elemen dan disimpan pada lokasi tertentu pada memori.

5. *Object*

Jenis variabel objek adalah berdasarkan gambaran objek pada dunia nyata yang mempunyai status dan tingkah laku. Sebuah variabel objek menyimpan statusnya dalam bentuk variabel dan tingkah lakunya dalam bentuk parameter.

2.4.2.3 Script PHP

Menurut Prasetyo (2003:4-5), terdapat dua cara yang sering digunakan untuk menulis *script* PHP, yaitu:

1. *Embedded Script*

Cara ini dilakukan dengan meletakkan *script* PHP di antara *tag-tag* HTML.

Berikut contoh penggunaannya:

```
<html>
<head>
<title> embedded script </title>
</head>
<body>
<? echo "Ini adalah contoh embedded script"; ?>
</body>
</html>
```

2. *Non-Embedded Script*

Non-embedded script merupakan pembuatan program murni PHP, dimana *tag-tag* HTML yang diletakkan didalamnya. Berikut contoh penggunaannya:

```
<?
echo"<html>";
echo"<head>";
echo"<title> non embedded script </title>";
echo"</head>";
echo"<body>";
echo"Ini adalah contoh non embedded script";
echo"</body>";
echo"</html>";
?>
```

2.4.2.4 *Tag PHP*

Penulisan program PHP dapat dilakukan dengan menggunakan berbagai *tag*. Hal ini tidak akan mempengaruhi hasil program yang dibuat [PUT-06]. *Tag* yang dapat digunakan adalah sebagai berikut:

1. *Style Standar*

Style standar PHP sangat mirip dengan penulisan program XML, yakni diawali dengan **<?php** dan diakhiri dengan **?>**. Contoh penulisannya adalah

```
<html>
<head>
<title> style standar </title>
</head>
<body>
<?php echo "Ini adalah contoh style standar PHP"; ?>
</body>
</html>
```

2. *Short Style*

Penulisan *script* pada *style* ini cukup praktis bila dibandingkan dengan *style* sebelumnya karena hanya diawali dengan **<?** dan diakhiri dengan **?>**. Contoh penulisannya adalah:

```
<html>
<head>
<title> short style </title>
</head>
<body>
<? echo "Ini adalah contoh short style PHP"; ?>
</body>
</html>
```

3. *Style JavaScript*

Style ini digunakan apabila telah terbiasa menggunakan pemrograman JavaScript. Penulisan programnya diawali dengan **<SCRIPT LANGUAGE = 'PHP'>** dan diakhiri dengan **</SCRIPT>**. Contoh penulisannya adalah:

```
<html>
<head>
```



```
<title> style JavaScript </title>
</head>
<body>
<SCRIPT LANGUAGE = 'PHP'>
    echo "Ini adalah contoh style Javascript";
</SCRIPT>
</body>
</html>
```

4. Style ASP

Diawali dengan `<%` dan diakhiri dengan `%>`. Contoh penulisannya adalah:

```
<html>
<head>
<title> style ASP </title>
</head>
<body>
<% echo "Ini adalah contoh style ASP"; %>
</body>
</html>
```

2.4.2.5 Cookie

Cookie adalah sepotong data yang disimpan pada *harddisk* lokal milik pengunjung (*client*) dan digunakan oleh halaman *web* dalam mengingat sesuatu informasi [KAD-02]. Data pada *harddisk* lokal ini dikenali oleh *server* untuk mendapatkan kembali informasi yang pernah dikirimkan ke *client*.

Cookie bersifat sementara, artinya ketika *browser* ditutup atau waktu penyimpanan *cookie* tersebut telah habis. *Cookie* juga akan secara otomatis dihapus. Dengan demikian, ketika *browser* kembali dibuka atau melewati jangka waktu tertentu, maka nilai *cookie* tersebut akan kosong.

Sebuah *client* hanya dapat memegang 300 *cookie* pada satu saat dan sebuah *server* hanya dapat mengirim 20 *cookie* ke sebuah *client*. Sebuah *cookie* hanya bisa berukuran sampai dengan 4 kilobyte, sehingga total memori *harddisk* yang digunakan *cookie* hanya mencapai 1,2 megabyte.

Web server mengirim *cookie* ke *client* melalui judul HTTP, yang dikirim sebelum teks HTML. Melalui judul pada HTTP inilah *client* bisa mengetahui apakah ia perlu mengirimkan *cookie* yang ada pada *harddisk* lokal ke *web server* atau tidak. Seperti halnya dengan *web server*, *client* juga mengirimkan kembali *cookie* ke *web server* melalui judul HTTP. Di dalam judul HTTP, *cookie* diatur melalui set-cookie. set-cookie mempunyai informasi nama *cookie* dan nilainya, batas kadaluarsa, *path*, *domain* dan parameter keamanan. Informasi nama variabel dan nilai dari *cookie* ditunjukkan dalam Tabel 2.7.

Informasi	Keterangan
<i>Path</i>	Digabungkan dengan <i>domain</i> , nilai <i>path</i> menentukan direktori pada <i>web server</i> yang dapat menggunakan <i>cookie</i> . Jika informasi <i>path</i> dan URL yang diminta tidak cocok, maka <i>client</i> tidak akan mengirimkan <i>cookie</i> . Nilai bawaan pada <i>path</i> berupa “/” yang berarti bahwa <i>cookie</i> valid untuk semua direktori pada <i>web server</i> .
<i>Domain</i>	Menyatakan <i>domain</i> (alamat) <i>server</i> yang mendefinisikan <i>cookie</i> . <i>Client</i> tidak akan mengirimkan <i>cookie</i> apabila <i>path</i> dan URL yang diminta tidak cocok dengan <i>domain</i> ini. Salah satu pemanfaatannya adalah untuk berbagi <i>cookie</i> pada beberapa <i>server</i> .
<i>Expires</i>	Menyatakan batas waktu kadaluarsa. Bawaannya, <i>cookie</i> hanya berlaku sampai <i>browser</i> ditutup.
<i>Secure</i>	Untuk menentukan pengiriman <i>cookie</i> hanya apabila protokol HTTPS (HTTP <i>Secure</i>) digunakan.

Tabel 2.7 Parameter *cookie*

Sumber : [KAD-02:408-409]

2.4.2.6 Session

Session dapat digunakan untuk membentuk interaksi antara sebuah *client* dengan *web server* dalam selang waktu tertentu. PHP menyediakan pustaka yang berguna untuk membentuk sebuah *session* dengan menggunakan *session*. Berbeda dengan *cookie* yang menyimpan data pada *client*, *session* diimplementasikan dengan menyimpan data pada *server*. Dengan demikian tidak perlu ada komunikasi bolak-balik antara *web server* dan *client* ketika *web server* membutuhkan data tersebut.

Session akan dimulai ketika *client* mulai masuk sebuah situs dan akan berakhir begitu *client* tersebut menutup halaman situs yang telah dibukanya. *Client* akan mendapat variabel yang terus ada selama ia melakukan kunjungannya tersebut. Setiap kali suatu *session* dibentuk, maka akan terdapat referensi yang menunjuk ke *session* yang bersangkutan. Referensi ini dikenal dengan SID (*Session Identifier*). SID ini akan disimpan sebagai *cookie* apabila fasilitas *cookie* dalam keadaan dihidupkan pada *browser*, sedangkan data *session*-nya akan disimpan di *server* [KAD-02].

2.4.2.7 Keunggulan PHP

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya. Kelebihan PHP yaitu [SYU-99]:

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* bisa berkonsentrasi langsung pada penampilan dokumen *web*-nya.
2. Tidak ada proses *compiling* dan *linking*.
3. Berorientasi obyek (*object oriented*).
4. Sintaks pemrogramannya mudah dipelajari, sangat menyerupai C dan Perl.
5. Integrasi yang sangat luas ke berbagai *server* basis data. Menulis *web* yang terhubung ke basis data menjadi sangat sederhana. Basis data yang didukung oleh PHP antara lain Oracle, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, dBase, UNIX dbm dan sebagainya.

2.4.3 WeBuilder 2008

WeBuilder 2008 merupakan web code editor untuk segala macam editing web dokumen yang dikembangkan oleh Blumentals software. Memiliki tampilan yang bersih, mudah dipahami, flexible, dan start awal yang cepat.

Kelebihan dari WeBuilder 2008 ini adalah kemampuannya mendukung pemrograman *script server side* seperti ASP (*Active Server Page*). ASP.Net, ColdFusion, JSP (*Java Server Pages*) dan PHP (*PHP Hypertext Preprocessor*). WeBuilder juga mendukung pemrograman *client side* yang sangat terkenal dan banyak dipakai, yaitu HTML dan JavaScript. Beberapa keunggulan lain dari WeBuilder 2008 adalah sebagai berikut:

- Dilengkapi dengan sql browser untuk mempermudah membuat aplikasi berbasis database
- Memiliki inspeksi *run-time* PHP (tidak harus selalu mencompile untuk melihat kesalahan / error pada php)
- Mengedit HTML, CSS, JavaScript, PHP, ASP, Ruby menggunakan satu program
- Memiliki masa tunggu yang relative cepat.

2.4.3.1 Area kerja WeBuilder 2008

Area kerja WeBuilder 2008 terdiri dari:

- *Insert Bar*

Pada bagian ini, berisi tombol-tombol untuk memasukkan berbagai jenis objek seperti gambar, *form*, tabel, *layer* dan lainnya ke dalam dokumen *website* yang sedang dibuat. *Insert bar* dapat diatur agar ditampilkan dalam bentuk menu sehingga untuk dapat berpindah ke menu yang lain cukup menekan tombol menu yang sedang digunakan.

- *Format bar*

Pada bagian ini , berisi tombol-tombol untuk mengatur teks,dan paragraph seperti mengatur, *font, paragraph, numbered list, buletted list, heading,underline, italic*, dan lain sebagainya.

- *CSS*

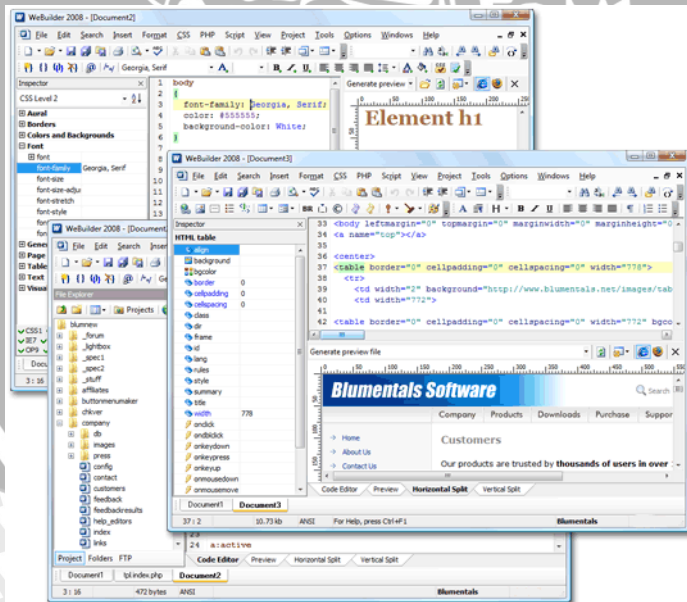
Bagian ini dipergunakan untuk mengatur property pada CSS (Cascading Style Sheet). Selain mengatur property pada CSS, pada menu ini juga dilengkapi dengan CSS check untuk memeriksa kode CSS apabila terdapat kesalahan, selain itu juga disediakan link untuk memeriksa kode CSS pada W3C CSS validator.

- *PHP*

Pada bagian ini disediakan menu *php* untuk memasukan *syntax php* seperti *if, if..else; if..else if..else; switch; while*. Dilengkapi juga dengan *debug* untuk memeriksa kesalahan kode *php*.

- *Script*

Menu berikut menyediakan beberapa *syntax* untuk pemrograman script, seperti : ASP, Javascript, PHP, SSI, Ruby.



Gambar 2.8 Area kerja WeBuilder 2008

Sumber : [BLU-09]

BAB III

METODE PENELITIAN

Metodologi penelitian menjelaskan mengenai langkah-langkah yang dilakukan untuk merealisasikan aplikasi perangkat lunak yang akan dibuat. Langkah-langkah yang diperlukan antara lain studi literatur, studi lapangan, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan analisis perangkat lunak serta pengambilan kesimpulan dan saran.

3.1 Studi Literatur

Studi literatur yang dilakukan bertujuan untuk mengkaji hal-hal yang berhubungan dengan teori-teori yang mendukung dalam perancangan dan perealisasiannya aplikasi yaitu:

- a. Perancangan sistem informasi berbasis *web* meliputi :
 - *Use-Case Diagram*.
 - *Class Diagram*.
 - *Sequence Diagram*
 - *Statechart Diagram*.
 - *Entity-Relationship Diagram (ER Diagram)*.
- b. Basis data meliputi :
 - Bahasa basis data SQL (*Structured Query Language*).
 - Basis data MySQL.
- c. Teori dasar *web* meliputi :
 - Struktur dan bahasa pemrograman HTML (*Hyper Text Markup Language*).
 - Struktur dan bahasa pemrograman PHP (*PHP Hypertext Preprocessor*).
 - Aplikasi *web editor* WeBuilder 2008 yang akan digunakan untuk menulis kode-kode PHP

3.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri dari perancangan sistem, perancangan dan pembuatan website, perancangan dan pembuatan basis data. Perancangan didasarkan pada teori-teori yang nantinya akan diaplikasikan pada sistem. Sistem yang dibuat adalah Sistem Informasi Manajemen Kesehatan, Keselamatan, Keamanan Kerja pada PT. PLN persero - APJ Malang.

Perancangan dan Pembuatan Perangkat Lunak:

A. Perancangan Sistem

- Menentukan Aktor
- Merancang daftar kebutuhan Fungsional Sistem Informasi
- Pemodelan dalam *Use-case diagram*
- Spesifikasi *Use-case*
- Class diagram
- Sequence diagram

B. Perancangan dan Pembuatan Website

- Penentuan kontent website
- Perancangan Design
- Penyusunan coding

C. Perancangan dan Pembuatan Basis Data

- Perencanaan tabel awal
- Diagram ER
- Normalisasi

Perancangan aplikasi dilakukan untuk mempermudah implementasi, analisis algoritma dan pengujian. Perancangan aplikasi berdasarkan *Object Oriented Analysis* dan *Object Oriented Design* dengan menggunakan bahasa UML (*Unified Modeling Language*). Pada tahap ini dibuat suatu diagram pemodelan sistem aplikasi secara keseluruhan yang sesuai dengan analisis kebutuhan sistem yang digambarkan dalam diagram *use-case*. Perancangan dilakukan dengan membuat diagram *sequence* yang memperlihatkan interaksi pengguna dengan objek-objek dalam aplikasi dan diagram

class untuk melihat hubungan antar class, objek dan *interface* yang ada di dalam aplikasi

3.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak terdiri dari implementasi basis data dan implementasi antarmuka perangkat lunak.

Implementasi basis data dilakukan pada basis data MySQL. Implementasi basis data MySQL terdiri dari konfigurasi basis data MySQL dan implementasi perancangan basis data menggunakan *query*. Implementasi antarmuka aplikasi terdiri dari implementasi perangkat lunak untuk *login*, perangkat lunak untuk operator dan perangkat lunak untuk *guest*

3.4 Pengujian Aplikasi

Pengujian aplikasi yaitu untuk mengetahui kesesuaian analisa kebutuhan yang dibuat dengan implementasi aplikasi. Analisis sistem dilakukan dengan membandingkan sistem aplikasi yang dengan teori yang ada sehingga didapatkan suatu kesimpulan.

- Pengujian software aplikasi web menggunakan PHP untuk mengetahui apakah bahasa PHP yang telah dimuat pada aplikasi web dapat berjalan dengan benar.
- Pengujian software aplikasi web terhadap database MySQL untuk mengetahui apakah pemrograman database pada aplikasi web tersebut dapat berjalan sesuai dengan fungsinya.

3.5 Pengambilan Kesimpulan dan Saran

Setelah mendapatkan hasil dan analisis dari pengujian software, maka langkah berikutnya adalah penarikan kesimpulan dari data-data yang didapat dan memberi saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penulisan

BAB IV PERANCANGAN

Bab ini membahas analisis kebutuhan dan perancangan aplikasi pada Sistem Informasi Manajemen Kesehatan, Keselamatan, dan Keamanan Kerja pada PT.PLN persero. Tahap analisis kebutuhan menggunakan pemodelan *use case diagram*. Tahap perancangan aplikasi terdiri dari 5 tahap, yang berisi perancangan sistem, *class diagram*, *sequence diagram*, perancangan antarmuka, dan perancangan basis data.

4.1 Analisis Kebutuhan

Analisis kebutuhan perangkat lunak adalah aktifitas rekayasa perangkat lunak yang menjembatani antara kebutuhan ditingkat sistem dan perancangan perangkat lunak. Digunakan untuk mendapatkan, menganalisis, dan memvalidasi kebutuhan-kebutuhan sistem.

4.1.1 Analisis Sistem Saat Ini

Proses pelaksanaan kesehatan, keselamatan, dan keamanan (K3) kerja yang akan dibahas pada skripsi ini adalah pada bagian Pekerjaan Dalam Keadaan Bertegangan (PDKB) di PT. PLN Persero APJ Malang sebagaimana pelaksanaan kegiatannya adalah sebagai berikut.

Setiap 1 bulan sekali karyawan PT.PLN Persero APJ Malang melakukan pemeriksaan kesehatan di klinik kesehatan PT.PLN Persero untuk checkup kondisi badan, dan untuk didata kesehatan tubuh karyawan, yang kemudian akan dicatat dalam *medical record* tiap karyawan.

Dari bagian kesehatan data *medical record* akan dikirim ke bagian SDM-Administrasi untuk didata kesehatan personal sebagai pertimbangan pengusulan nama dalam pelaksanaan PDKB. Bagian SDM-Administrasi juga akan menerima masukan data lulus uji kompetensi dari bagian training.

ASMAN akan membuat jadwal rutin dalam perawatan dan juga setiap saat ASMAN akan memperoleh laporan kerusakan dari koordinator lokasi dalam lingkup APJ malang. Setelah Laporan diterima, ASMAN akan berkoordinasi dengan bagian PDKB untuk mengetahui jenis kegiatan PDKB apa yang akan diterapkan untuk perbaikan ataupun perawatan berkala. Setelah jenis pekerjaan ditentukan, bagian PDKB akan berkoordinasi dengan bagian SDM-Administrasi untuk memilih nama-nama yang sesuai dengan jenis pekerjaan PDKB yang akan dilaksanakan. Dengan mempertimbangkan *medical Record* dan kompetensi dari nama-nama yang akan dipilih.

Usulan nama yang telah sesuai akan dilaporkan kepada bagian coordinator & preparatory, ASMAN bidang SDM-Administrasi, dan manajer APJ malang untuk dilakukan persetujuan dan pemeriksaan. Setelah persetujuan diberikan, maka usulan nama yang telah disetujui akan dikirim ke bagian PDKB sebagai penyelenggara kegiatan, diteruskan kepada bagian SDM-Administrasi untuk pemanggilan nama yang telah diusulkan. Koordinator bidang bertugas untuk membuat surat perintah bagi nama-nama dipanggil tersebut, dan menunjuk untuk bertugas sebagai pengawas dan pelaksana pekerjaan.

Setelah surat perintah diserahkan kepada pengawas dan pelaksana pekerjaan, coordinator akan memberikan Surat “Analisis Pekerjaan dan Pembagian Tugas” kepada pengawas pekerjaan, untuk memeberikan job desc yang akan dilaksanakan, beserta daftar anggota pelaksana.

Dari job desc yang telah diberikan, koordinator akan menyerahkan surat “Pemberitahuan Penyelesaian Pekerjaan ” yang disetujui dan disepakati oleh pengawas.

Pengawas akan memantau pekerjaan dan akan dibuat laporan pekerjaan beserta data keselamatan pekerja selama pekerjaan berlangsung, yang akan diserahkan ke bagian SDM-Administrasi, yang sebelumnya telah disetujui oleh Manajer APJ-malang, Koordinator, dan ASMAN bidang SDM –Adminstrasi.

Kelebihan yang ada pada system yang berjalan selama ini adalah:

1. Usulan nama pekerja telah dipantau mulai awal, sejak check kesehatan hingga akhir pekerjaan.
2. Seluruh pekerja akan bekerja sesuai bidang pekerjaan yang sesuai, dengan kesehatan yang terpantau, serta dengan penerapan prosedur keselamatan dan keamanan kerja yang sesuai dengan jenis pekerjaan.

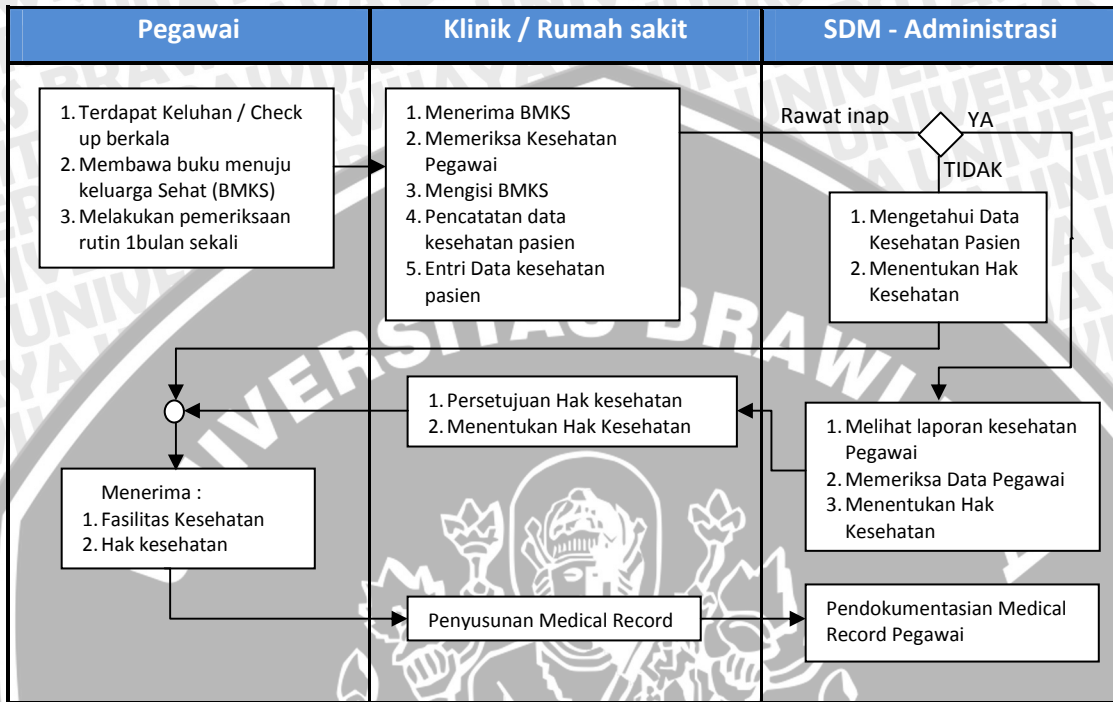
Kekurangan yang ada pada system yang berjalan selama ini adalah:

1. Asman akan sering berkoordinasi dengan banyak bidang untuk mendapatkan informasi, terutama dengan SDM-Administrasi, Dari banyaknya bidang yang berkoordinasi dengan SDM-Administrasi membuat informasi yang diminta disajikan beberapa waktu kemudian.
2. Dari koordinasi dengan banyak bidang, membuat waktu yang digunakan menjadi lama, dan tidak efisien.

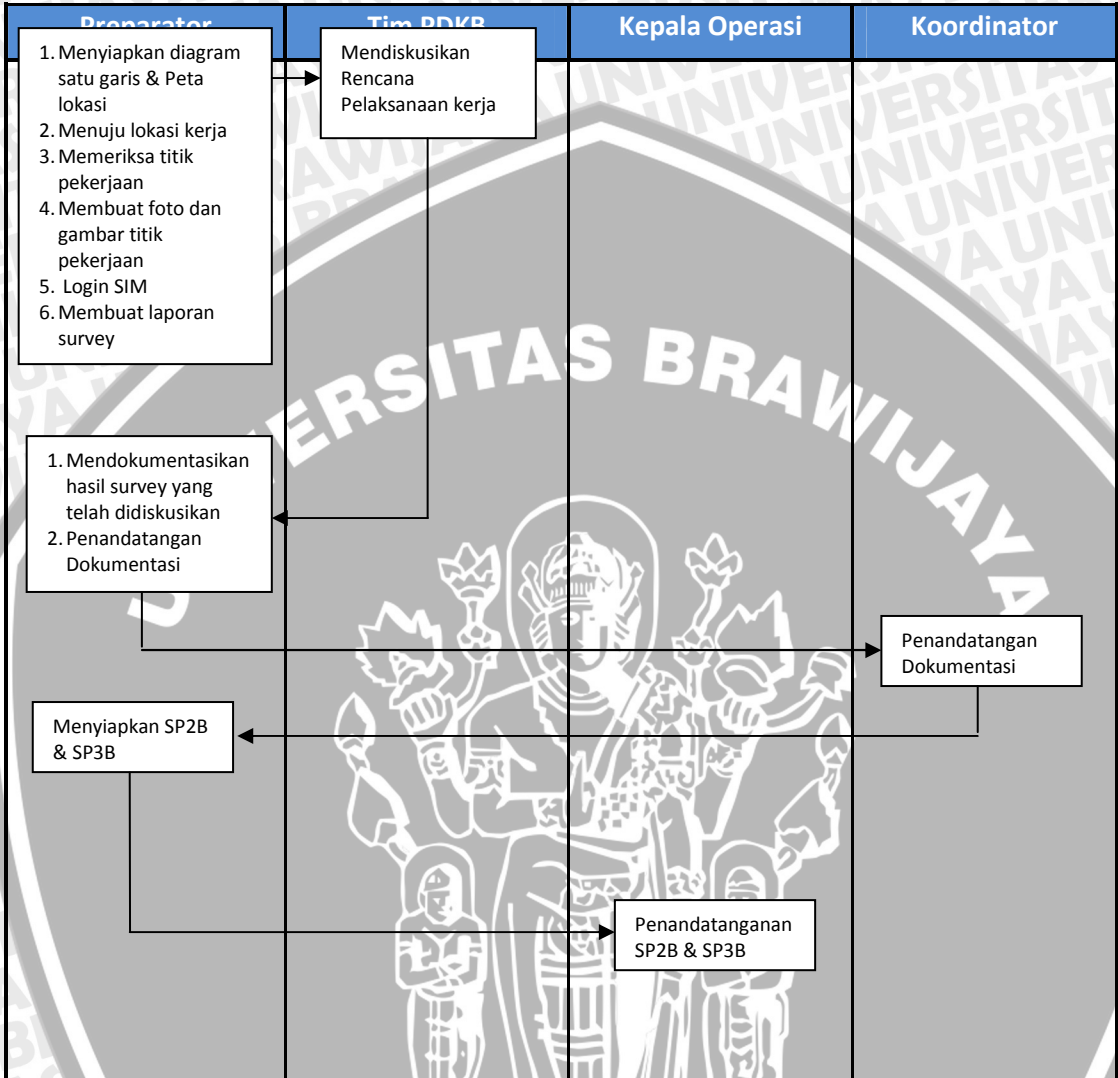
Kekurangan dari sistem yang berjalan selama ini dapat diatasi dengan membuat sebuah aplikasi yang memenuhi kriteria :

1. Dapat menyajikan Informasi yang sesuai secara Up to Date
2. Dapat mendukung koordinasi yang dibutuhkan terkait masalah K3 dalam bidang PDKB

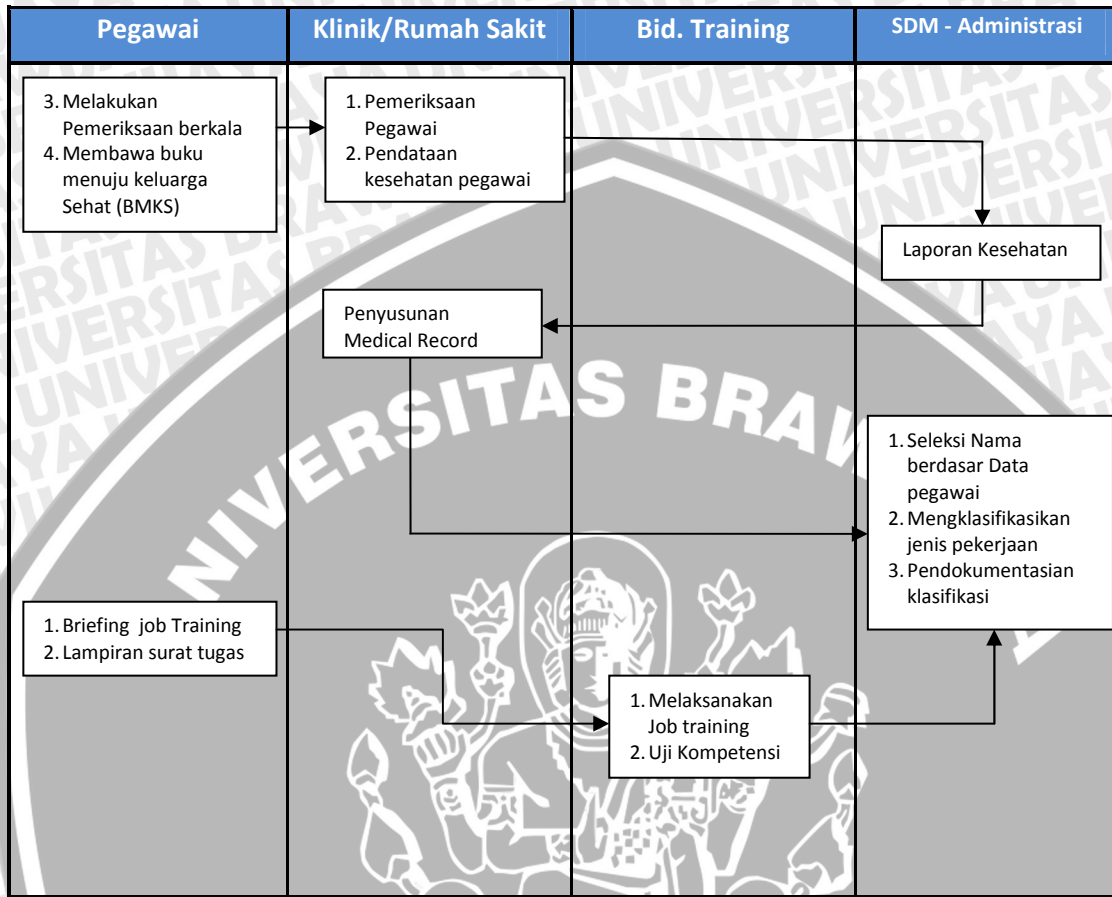
Berikut digambarkan dalam Diagram Prosedur Operasi :



Tabel 4.1 Pendatan dan Pemberian Hak Kesehatan Pegawai



Tabel 4.2 Laporan Lokasi Kerusakan/perawatan



Tabel 4.4 Proses Pengusulan Nama

4.1.2 Analisis Sistem Yang Akan Disusun

Tahap analisis kebutuhan dilakukan dengan memodelkan kebutuhan ke dalam *use case diagram*. *Use case diagram* bertujuan untuk menggambarkan kebutuhan-kebutuhan fungsional yang harus disediakan oleh Sistem Informasi Manajemen Kesehatan Keselamatan dan Keamanan kerja agar dapat memecahkan permasalahan yang dihadapi oleh pengguna saat ini.

4.1.2.1 Daftar Kebutuhan

Daftar kebutuhan merupakan daftar yang menguraikan kebutuhan-kebutuhan pengguna yang harus disediakan oleh perangkat lunak baik kebutuhan fungsional maupun non fungsional. Sebelum dapat membuat daftar kebutuhan, perlu dirumuskan terlebih dahulu aktor-aktor yang menggunakan sistem ini.

Tabel 4.5 dibawah ini akan memperlihatkan aktor-aktor beserta penjelasannya yang merupakan hasil dari proses identifikasi aktor.

No.	Aktor	Penjelasan
1.	User	Aktor yang telah melakukan login pada sistem. Setelah aktor melakukan login, maka aktor memiliki hak kerja masing-masing sesuai dengan peran.
2.	Perancang	Aktor yang bertugas merancang jenis pekerjaan yang akan dilaksanakan
3.	Pengelola	Aktor yang bertugas untuk mengelola pelaksanaan kerja. Yang dimaksud dengan mengelola adalah menyiapkan data kesehatan pegawai, pengusulan calon pekerja, menyusun jadwal pekerjaan, membuat surat persetujuan, membuat surat tugas bagi pekerja.
4.	Pengawas	Aktor yang bertugas memantau keselamatan, keamanan, dan kesehatan pegawai;
5.	Pegawai	Aktor yang bertugas memantau data kepegawaian yang bersifat personal.
6.	Administrator	Aktor yang bertugas untuk membuat Account baru yang memiliki hak-hak tertentu sesuai kapasitas dan keperluannya masing-masing. Termasuk perbaikan system apabila terjadi

		gangguan pada system
7.	Pengurus	Aktor yang berkaitan dengan memproses data, termasuk didalamnya menghapus, menambah, mengubah, melihat data.

Tabel 4.5 Deskripsi Aktor [Analisis]

Daftar kebutuhan ini terdiri dari sebuah kolom yang menguraikan kebutuhan yang harus disediakan oleh sistem, dan pada kolom yang lain akan menunjukkan nama *use case* yang akan menyediakan fungsionalitas masing-masing kebutuhan tersebut. Daftar kebutuhan fungsional dan non fungsional sistem yang dikembangkan pada tugas akhir ini ditunjukkan pada Tabel 4.6.

No.	<i>Use Case</i>	Kebutuhan
1.	Login	Sistem harus memberikan fasilitas untuk login, sehingga hanya pengguna tertentu yang dapat mengakses informasi.
2.	Logout	Sistem harus menyediakan fasilitas untuk logout agar User dapat keluar dari sistem.
3.	Menambahkan <i>Account</i>	Sistem harus menyediakan fasilitas bagi administrator untuk menambah <i>account</i> baru yang dapat melakukan akses terhadap sistem ini dengan hak-hak tertentu sesuai kapasitas dan keperluannya terhadap sistem ini.
4.	Melihat <i>Account</i>	Sistem harus menyediakan fasilitas bagi administrator untuk melihat <i>account-account</i> yang ada.
5.	Mengubah <i>Account</i>	Sistem harus menyediakan fasilitas bagi administrator untuk mengubah <i>account-account</i>

		yang ada
6.	Menghapus <i>Account</i>	Sistem harus menyediakan fasilitas bagi administrator untuk menghapus <i>account-account</i> yang ada
7.	Menambah Data Kesehatan	Sistem harus menyediakan fasilitas bagi Pengurus untuk menambah data kesehatan.
8.	Melihat Data Kesehatan	Sistem harus menyediakan fasilitas bagi Pengurus untuk melihat data kesehatan.
9.	Mengganti Data Kesehatan	Sistem harus menyediakan fasilitas bagi Pengurus untuk mengubah informasi terkait dengan data kesehatan yang ada.
10.	Menghapus Data Kesehatan	Sistem harus menyediakan fasilitas bagi Pengurus untuk menghapus data kesehatan yang sudah tidak ada.
11.	Menambah Data Pegawai	Sistem harus menyediakan fasilitas bagi Pengurus untuk menambah data pegawai.
12.	Melihat Data Pegawai	Sistem harus menyediakan fasilitas bagi Pengurus untuk melihat data pegawai yang ada.
13.	Mengganti Data Pegawai	Sistem harus menyediakan fasilitas bagi Pengurus untuk mengubah data pegawai.
14.	Menghapus Data Pegawai	Sistem harus menyediakan fasilitas bagi Pengurus untuk menghapus data pegawai.
15.	Menambah Kegiatan Kerja	Sistem harus menyediakan fasilitas bagi Pengurus untuk menambah kegiatan kerja.

	PDKB	
16.	Melihat Kegiatan Kerja PDKB	Sistem harus menyediakan fasilitas bagi Pengurus untuk melihat kegiatan kerja.
17.	Mengganti Kegiatan Kerja PDKB	Sistem harus menyediakan fasilitas bagi Pengurus untuk mengubah kegiatan kerja.
18.	Menghapus Kegiatan Kerja PDKB	non-sistem Sistem harus menyediakan fasilitas bagi administrator untuk menghapus kegiatan kerja.
19.	Melihat Medical Record	Sistem harus menyediakan fasilitas bagi Pengelola dan pegawai untuk melihat track record kesehatan
20.	Menyusun Jadwal Operasi Perbaikan & Perawatan	Sistem harus menyediakan fasilitas bagi Pengelola untuk menyusun Jadwal Operasi
21.	Menambah Jadwal Operasi Perbaikan & Perawatan	Sistem harus menyediakan fasilitas bagi Pengelola untuk menambah Jadwal Operasi
22.	Melihat Jadwal Operasi Perbaikan & Perawatan	Sistem harus menyediakan fasilitas bagi Pengelola dan pegawai untuk melihat Jadwal Operasi yang ada.
23.	Mengganti	Sistem harus menyediakan fasilitas bagi Pengelola

	Jadwal Operasi Perbaikan & Perawatan	untuk mengubah Jadwal Operasi.
24.	Menghapus Jadwal Operasi Perbaikan & Perawatan	Sistem harus menyediakan fasilitas bagi Pengelola untuk menghapus Jadwal Operasi.
25.	Menyusun Analisis Pekerjaan Dan Pembagian Tugas	Sistem harus menyediakan fasilitas bagi Perancang untuk menyusun Analisis Pekerjaan Dan Pembagian Tugas sebagai pemantapan job description dan penentuan pembagian tugas
26.	Melihat Analisis Pekerjaan Dan Pembagian Tugas	Sistem harus menyediakan fasilitas bagi Pengelola dan Pegawai untuk melihat Analisis Pekerjaan Dan Pembagian Tugas
27.	Mengubah Analisis Pekerjaan Dan Pembagian Tugas	Sistem harus menyediakan fasilitas bagi Pengelola untuk mengubah data mengenai Analisis Pekerjaan Dan Pembagian Tugas
28.	Menghapus Analisis Pekerjaan Dan Pembagian	Sistem harus menyediakan fasilitas bagi Pengelola untuk menghapus data mengenai Analisis Pekerjaan Dan Pembagian Tugas

	Tugas	
29.	Menyusun pemberitahuan penyelesaian pekerjaan	Sistem harus menyediakan fasilitas bagi pengawas untuk menentukan jadwal penyelesaian pekerjaan.
30.	Melihat pemberitahuan penyelesaian pekerjaan	Sistem harus menyediakan fasilitas bagi pengawas untuk melihat jadwal penyelesaian pekerjaan
31.	Mengubah pemberitahuan penyelesaian pekerjaan	Sistem harus menyediakan fasilitas bagi pengawas untuk mengubah jadwal penyelesaian pekerjaan
32.	Menghapus pemberitahuan penyelesaian pekerjaan	Sistem harus menyediakan fasilitas bagi pengawas untuk menambah jadwal penyelesaian pekerjaan
33.	Merancang Nama Calon Pekerja	Sistem harus menyediakan fasilitas bagi pengelola dan perancang untuk dapat membuat usulan calon pekerja yang sesuai dengan jenis pekerjaan yang akan dikerjakan dengan pertimbangan kesehatan pekerja.
34.	Melihat Nama Calon Pekerja	Sistem harus menyediakan fasilitas bagi pengelola Pegawai, dan pengawas untuk dapat melihat nama-nama calon pekerja yang diusulkan.
35.	Mengubah	Sistem harus menyediakan fasilitas bagi pengelola

	Nama Calon Pekerja	dan pengawas untuk dapat menghapus nama-nama calon pekerja yang diusulkan.
36.	Menambah Usulan Calon Pekerja	Sistem harus menyediakan fasilitas bagi pengelola dan pengawas untuk dapat menambah nama-nama calon pekerja yang diusulkan.
37.	Menghapus Nama Calon Pekerja	Sistem harus menyediakan fasilitas bagi pengelola dan pengawas untuk dapat mengubah nama-nama calon pekerja yang diusulkan.
38.	Menyetujui Nama Calon Pekerja	Sistem harus menyediakan fasilitas bagi pengawas untuk menyetujui nama-nama calon pekerja yang diusulkan. Setelah usulan nama-nama calon pekerja disetujui, maka tidak bisa lagi dilakukan perubahan terhadap nama-nama calon pekerja.
39.	Membuat Surat Tugas	Sistem harus menyediakan fasilitas bagi pengelola untuk membuat surat penugasan bagi pegawai yang diberangkatkan mengikuti pekerjaan dalam PDKB.
40.	Melihat Surat Tugas	Sistem harus menyediakan fasilitas bagi pengelola dan Pegawai untuk melihat surat penugasan bagi pegawai yang diberangkatkan mengikuti pekerjaan dalam PDKB.
41.	Mengubah Surat Tugas	Sistem harus menyediakan fasilitas bagi pengelola untuk mengubah surat penugasan bagi pegawai yang diberangkatkan mengikuti pekerjaan dalam PDKB.
42.	Menghapus	Sistem harus menyediakan fasilitas bagi pengelola untuk menghapus informasi terkait penugasan

	Surat Tugas	pegawai yang diberangkatkan mengikuti pekerjaan dalam PDKB.
43.	Mengawasi Kegiatan Pekerjaan	Sistem harus menyediakan fasilitas bagi pengawas untuk melihat nama-nama yang diusulkan, diberangkatkan, dan dapat melihat hasil evaluasi pekerjaan dan keselamatan pekerjaan.
44.	Mencetak Rencana Kerja	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak rencana kerja periodik.
45.	Mencetak Usulan Nama	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak daftar usulan nama.
46.	Mencetak Surat Tugas	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak surat tugas.
47.	Mencetak Surat Analisis Pekerjaan dan Pembagian Tugas	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak surat Analisis Pekerjaan dan Pembagian Tugas
48.	Mencetak Surat Pemberitahuan Penyelesaian Pekerjaan	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak Surat Pemberitahuan Penyelesaian Pekerjaan
49.	Mencetak Hasil Evaluasi Dampak Pekerjaan	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak hasil evaluasi Pekerjaan & Keselamatan Pekerjaan.

Tabel 4.6. Daftar Kebutuhan Fungsional Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja [Analisis]

Kebutuhan fungsionalitas di atas dapat dikelompokkan menjadi 3 kelompok untuk lebih mempermudah pemahaman dan pendisainan sistem. Ketiga kelompok tersebut adalah sebagai berikut :

1. Pendukung Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja
2. Penyedia data pendukung Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja
3. Pengelolaan informasi dalam Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja

Tabel 4.7 berikut menyatakan pembagian kelompok dan kebutuhan fungsionalitas yang termasuk dalam kelompok tersebut

No.	Kelompok	Kebutuhan Fungsionalitas
1.	Pendukung Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja	1. Login
		2. Logout
		3. Menambahkan <i>Account</i>
		4. Melihat <i>Account</i>
		5. Mengubah <i>Account</i>
		6. Menghapus <i>Account</i>
2.	Penyedia data pendukung Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja	7. Menambah Data Kesehatan
		8. Melihat Data Kesehatan
		9. Mengganti Data Kesehatan
		10. Menghapus Data Kesehatan
		11. Menambah Data Pegawai
		12. Melihat Data Pegawai
		13. Mengganti Data Pegawai
		14. Menghapus Data Pegawai

		15. Menambah kegiatan kerja PDKB
		16. Melihat kegiatan kerja PDKB
		17. Mengganti kegiatan kerja PDKB
		18. Menghapus kegiatan kerja PDKB
3.	Pengelolaan informasi dalam Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja	19. Melihat Medical Record
		20. Menyusun Jadwal Operasi Perbaikan & Perawatan
		21. Menambah Jadwal Operasi Perbaikan dan Perawatan
		22. Melihat Jadwal Operasi Perbaikan dan Perawatan
		23. Mengganti Jadwal Operasi Perbaikan dan Perawatan
		24. Menghapus Jadwal Operasi Perbaikan dan Perawatan
		25. Menyusun Analisis Pekerjaan dan Pembagian Tugas
		26. Melihat Analisis Pekerjaan dan Pembagian Tugas
		27. Mengubah Analisis Pekerjaan dan Pembagian Tugas
		28. Menghapus Analisis Pekerjaan dan Pembagian Tugas
		29. Menyusun Pemberitahuan Penyelesaian Pekerjaan
		30. Melihat Pemberitahuan Penyelesaian Pekerjaan
		31. Mengubah Pemberitahuan Penyelesaian Pekerjaan

	32. Menghapus Pemberitahuan Penyelesaian Pekerjaan
	33. Merancang Nama Calon Pekerja
	34. Melihat Nama Calon Pekerja
	35. Mengubah Nama Calon Pekerja
	36. Menambah Nama Calon Pekerja
	37. Menghapus Nama Calon Pekerja
	38. Menyetujui Nama Calon Pekerja
	39. Membuat Surat Tugas
	40. Melihat Surat Tugas
	41. Mengubah Surat Tugas
	42. Menghapus Surat Tugas
	43. Mengawasi Kegiatan Pekerjaan
	44. Mencetak Rencana Kerja
	45. Mencetak Usulan Nama
	46. Mencetak Surat Tugas
	47. Mencetak Surat Analisis Pekerjaan dan Pembagian Tugas
	48. Mencetak Surat Pemberitahuan Penyelesaian Pekerjaan
	49. Mencetak Hasil Evaluasi Dampak Pekerjaan

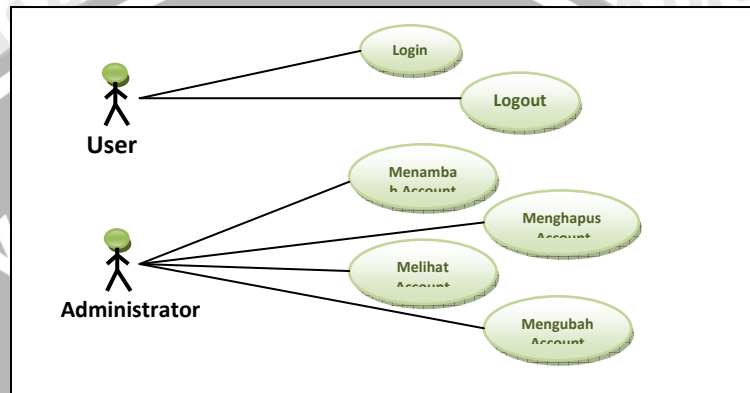
Tabel 4.7. pembagian kelompok dan kebutuhan fungsionalitas

4.1.2.2 Use Case Diagram

Pemodelan dalam use case diagram yang menggambarkan fungsionalitas yang disediakan oleh Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja di bagi menjadi tiga buah diagram yang bersesuaian dengan kelompok kebutuhan.

4.1.2.2.1 Pendukung Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja

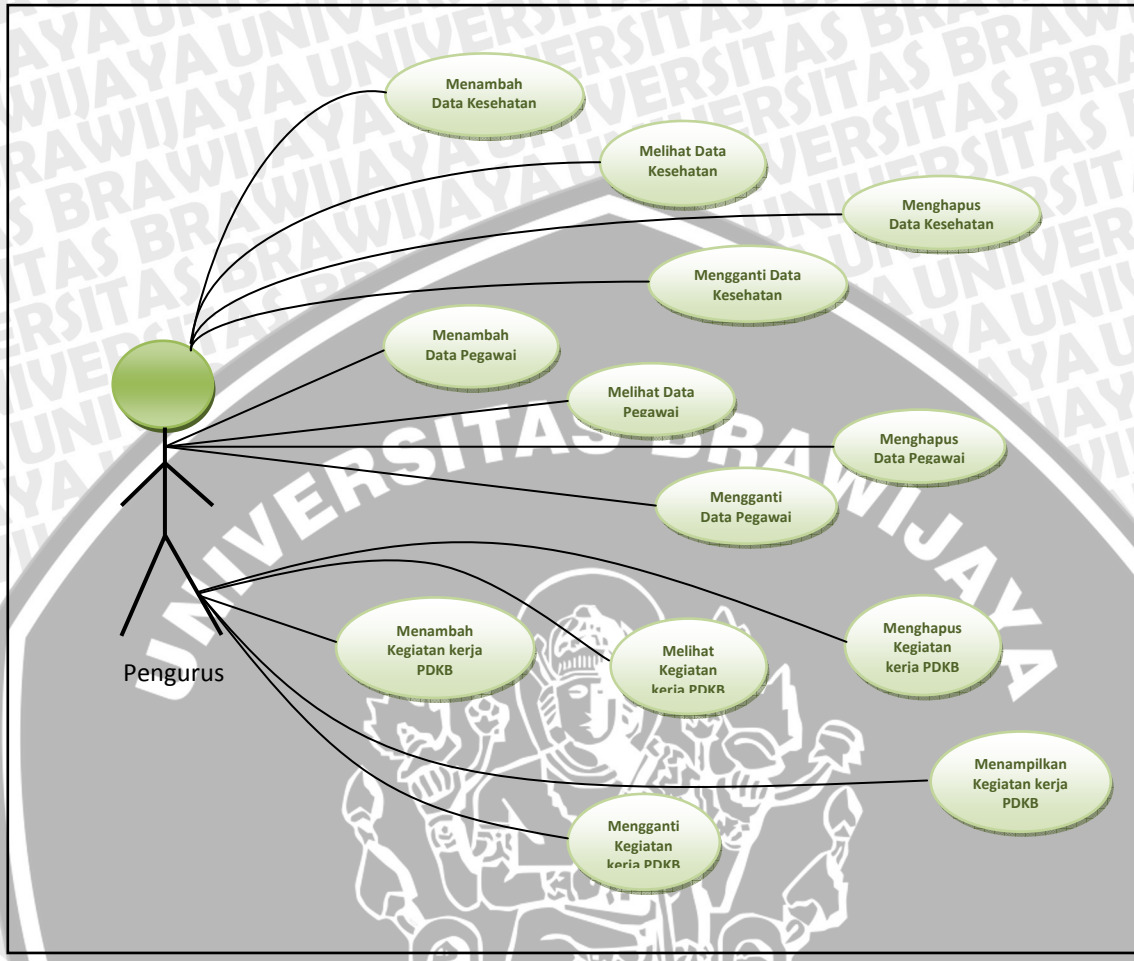
Use Case Diagram untuk kebutuhan fungsional yang termasuk dalam pendukung sistem informasi Kesehatan, Keselamatan, dan Keamanan Kerja melibatkan dua aktor yaitu *User*, dan *Administrator*.



Gambar 4.1 Use Case Diagram Pendukung Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja [Analisis]

4.1.2.2.2 Penyedia data pendukung Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja

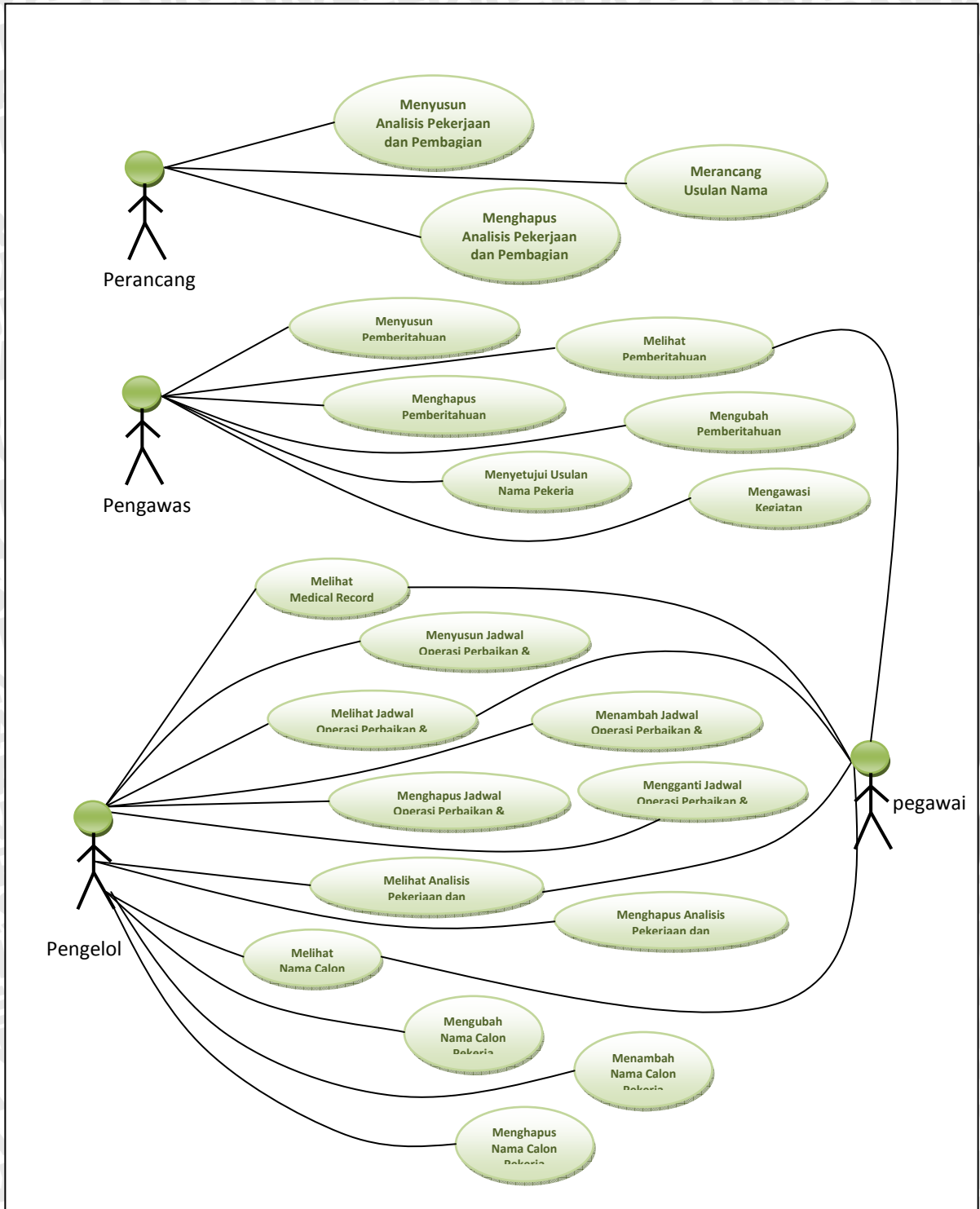
Use Case Diagram untuk kebutuhan fungsional yang termasuk dalam penyedia data pendukung melibatkan hanya satu aktor saja yaitu *Pengurus*.



Gambar 4.2 Use Case Diagram Penyedia Data Pendukung Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja [Analisis]

4.1.2.2.3 Pengelolaan informasi dalam Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja

Use Case Diagram untuk kebutuhan fungsional yang termasuk dalam kelompok pengelolaan informasi dalam Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja melibatkan tiga aktor yaitu Perancang, Pengelola, dan Pengawas. Pada Gambar 4.3 berikut ini akan digambarkan *Use Case Diagram* dari aktor tersebut dan perannya dalam use case.



Gambar 4.3 Use Case Diagram Pengelolaan Informasi dalam Sistem Informasi Kesehatan, Keselamatan, dan Keamanan Kerja [Analisis]

4.1.2.3 Spesifikasi Use Case

Use Case Diagram memberikan gambaran mengenai hal – hal yang dapat dilakukan oleh sistem. Tetapi diagram – diagram tersebut hanya menyajikan ”siapa dapat melakukan apa”, dalam spesifikasi *Use Case* ini akan coba dijelaskan secara tekstual mengenai *Use Case* – *Use Case* yang telah digambarkan sebelumnya. Hal ini perlu dilakukan untuk membawa pembaca pada tingkat pemahaman yang lebih dalam lagi.

4.1.2.3.1 Spesifikasi Use Case Pendukung Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja

- *Spesifikasi Use Case Login*

Nama Use Case	Login	
Aktor	User	
Deskripsi	Sistem harus memberikan fasilitas untuk login, sehingga masing- masing aktor memiliki hak kerja masing-masing sesuai dengan peran.	
Kondisi Awal	Aktor menjalankan sistem sampai pada halaman pertama	
Kondisi Akhir	Aktor memasuki sistem dengan hak – hak tertentu sesuai kewenangannya.	
	Aksi dari Aktor	Tanggapan dari Sistem
	1. Aktor memasukkan username dan password, kemudian tekan tombol ”login”	2. Sistem memvalidasi pasangan username dan password yang dimasukan. 3. Sistem Menampilkan halaman, sesuai peran dari masing-masing aktor.
	Pengecualian I :	
	Pasangan username dan password tidak ada dalam sistem	
		a. Sistem menampilkan pesan bahwa pasangan username dan password yang dimasukan tidak sesuai
		b. Sistem kembali ke langkah no 1

• **Spesifikasi Use Case Logout**

Nama Use Case	Logout
Aktor	User
Deskripsi	Sistem harus menyediakan fasilitas agar User dapat keluar dari sistem.
Kondisi Awal	User telah berada dalam sistem
Kondisi Akhir	User keluar dari sistem
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor menekan tombol "Logout"	2. Sistem melakukan Logout sehingga Aktor keluar dari sistem, dan sistem menampilkan Halaman Log in.

• **Spesifikasi Use Case Menambahkan Account**

Nama Use Case	Menambahkan <i>Account</i>
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas bagi administrator untuk menambah <i>account</i> baru yang dapat melakukan akses terhadap sistem ini dengan hak-hak tertentu sesuai kapasitas dan keperluannya terhadap sistem ini.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Administrator
Kondisi Akhir	Terdapat <i>account</i> baru dalam sistem
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih menu "Tambah Account"	2. Sistem menampilkan form yang berfungsi untuk memasukkan informasi nama, username, password, dan divisi.
3. Aktor memasukkan informasi nama, username, password, dan divisi. Kemudian actor menekan tombol "Simpan"	4. Sistem menyimpan data (nama, username, password, dan divisi) dalam basis data. 5. Sistem Menampilkan pesan "Tambah Data"

	Berhasil".
Pengecualian I :	
Username telah ada dalam sistem	
	a.Sistem tidak menyimpan informasi ke dalam sistem
	b.Sistem menampilkan pesan bahwa Username telah ada dalam sistem
	c. Sistem kembali menuju langkah no2.
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• Spesifikasi Use Case Melihat Account

Nama Use Case	Melihat <i>Account</i>
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas bagi administrator untuk melihat <i>account-account</i> yang ada.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Administrator
Kondisi Akhir	Seluruh data <i>account</i> ditampilkan oleh sistem
Aksi dari Aktor	Tanggapan dari Sistem
1.Aktor memilih menu " <i>Lihat Account</i> "	2.Sistem menampilkan seluruh informasi <i>account</i> yang tersimpan dalam basis data.

• Spesifikasi Use Case Mengubah Account

Nama Use Case	Mengubah <i>Account</i>
Aktor	Administrator

Deskripsi	Sistem harus menyediakan fasilitas bagi administrator untuk mengubah <i>account-account</i> yang ada								
Kondisi Awal	Aktor telah menjalankan <i>use case</i> Melihat <i>Account</i>								
Kondisi Akhir	Informasi <i>account</i> yang diubah tersimpan								
<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%; text-align: center;">Aksi dari Aktor</th> <th style="width: 50%; text-align: center;">Tanggapan dari Sistem</th> </tr> </thead> <tbody> <tr> <td>1. Aktor memilih salah satu baris <i>account</i></td> <td>2. Sistem menyajikan form berisi informasi <i>account</i> tersebut.</td> </tr> <tr> <td>3. Aktor melakukan perubahan pada tempat yang tersedia, kemudian tekan tombol "<i>simpan</i>".</td> <td>4. Sistem menyimpan informasi form ke dalam basis data.</td> </tr> <tr> <td></td> <td>5. Sistem menjalankan <i>use case</i> Melihat <i>Account</i></td> </tr> </tbody> </table>		Aksi dari Aktor	Tanggapan dari Sistem	1. Aktor memilih salah satu baris <i>account</i>	2. Sistem menyajikan form berisi informasi <i>account</i> tersebut.	3. Aktor melakukan perubahan pada tempat yang tersedia, kemudian tekan tombol " <i>simpan</i> ".	4. Sistem menyimpan informasi form ke dalam basis data.		5. Sistem menjalankan <i>use case</i> Melihat <i>Account</i>
Aksi dari Aktor	Tanggapan dari Sistem								
1. Aktor memilih salah satu baris <i>account</i>	2. Sistem menyajikan form berisi informasi <i>account</i> tersebut.								
3. Aktor melakukan perubahan pada tempat yang tersedia, kemudian tekan tombol " <i>simpan</i> ".	4. Sistem menyimpan informasi form ke dalam basis data.								
	5. Sistem menjalankan <i>use case</i> Melihat <i>Account</i>								
Pengecualian I :									
Perubahan menyebabkan adanya username ganda dalam sistem									
	a. Sistem tidak menyimpan perubahan yang dilakukan								
	b. Sistem menampilkan pesan bahwa username tersebut telah ada dalam sistem								
	c. Sistem menjalankan <i>use case</i> Melihat <i>Account</i>								
Aturan Khusus									
- Tidak boleh ada kolom isian yang tidak terisi									

• **Spesifikasi Use Case Menghapus Account**

Nama Use Case	Menghapus <i>Account</i>
Aktor	Administrator
Deskripsi	Sistem harus menyediakan fasilitas bagi administrator untuk menghapus

Aturan Khusus

- Tidak boleh ada kolom isian yang tidak terisi

• Spesifikasi Use Case Melihat Data Kesehatan

Nama Use Case	Melihat Data Kesehatan
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk melihat data kesehatan yang ada.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengurus
Kondisi Akhir	Seluruh data kesehatan ditampilkan oleh sistem
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih menu Data Kesehatan, kemudian tekan tombol "Lihat Data"	2. Sistem menampilkan seluruh informasi Kesehatan yang tersimpan dalam basis data.

• Spesifikasi Use Case Mengganti Data Kesehatan

Nama Use Case	Mengganti Data Kesehatan
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk mengubah informasi terkait dengan data kesehatan yang ada.
Kondisi Awal	Aktor telah menjalankan <i>use case</i> Melihat Data Kesehatan
Kondisi Akhir	Informasi data kesehatan yang diubah tersimpan
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih salah satu baris kesehatan .	2. Sistem menyajikan informasi tersebut dalam form.

3. Aktor melakukan perubahan pada tempat yang tersedia. Kemudian tekan tombol "Simpan".	4. Sistem menyimpan informasi dari form ke dalam basis data.
	5. Sistem menjalankan <i>use case</i> Melihat Data Kesehatan
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• **Spesifikasi Use Case Menghapus Data Kesehatan**

Nama Use Case	Menghapus <i>Data Kesehatan</i>
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk menghapus data Kesehatan yang sudah tidak ada.
Kondisi Awal	Aktor telah menjalankan <i>use case</i> Melihat Data Kesehatan
Kondisi Akhir	<i>Data</i> yang dipilih terhapus dari basis data sistem.
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih salah satu baris data kesehatan. Kemudian aktor menekan tombol " <i>Hapus</i> ".	2. Sistem menghapus data Kesehatan telah dipilih.
	3. Sistem menjalankan <i>use case</i> Melihat Data Kesehatan

• **Spesifikasi Use Case Menambah Data Pegawai**

Nama Use Case	Menambahkan Data Pegawai
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk menambah data pegawai.

Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengurus
Kondisi Akhir	Terdapat data pegawai baru dalam basis data sistem
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu Data Pegawai, Kemudian menekan tombol "Tambah Data"	2. Sistem menampilkan form yang berfungsi untuk memasukan Nama, NIP, Keterangan, serta pilihan daftar divisi.
3. Aktor memasukan informasi Nama, NIP, dan Keterangan, dilanjutkan dengan memilih divisi tempat Pegawai tersebut bergabung . Kemudian aktor menekan tombol "Simpan"	4. Sistem menyimpan data dalam basis data 5. Sistem menampilkan <i>Use Case</i> Melihat Data Pegawai
Pengecualian I :	
Pasangan Nama – NIP telah ada dalam sistem	
	a. Sistem tidak menyimpan data tersebut kedalam sistem
	b. Sistem menampilkan pesan bahwa pasangan Nama- NIP telah ada dalam sistem
	c. Sistem kembali ke langkah no 2.
Pengecualian II :	
NIP telah ada dalam sistem	
	a. Sistem tidak menyimpan data tersebut kedalam sistem
	b. Sistem menampilkan pesan bahwa NIP tersebut telah ada dalam sistem
	c. Sistem kembali ke langkah no 2.
Aturan Khusus	

- Tidak boleh ada kolom isian yang tidak terisi

• Spesifikasi Use Case Melihat Data Pegawai

Nama Use Case	Melihat Data Pegawai
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk melihat data pegawai yang ada.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengurus
Kondisi Akhir	Informasi mengenai pegawai tertentu ditampilkan
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih menu Data Pegawai, kemudian menekan tombol "Lihat Data"	2. Sistem menampilkan daftar nama seluruh pegawai yang ada.
3. Aktor memilih salah satu baris pegawai.	4. Sistem akan menampilkan informasi pegawai tersebut dan informasi kesehatan pegawai tersebut

• Spesifikasi Use Case Mengubah Data Pegawai

Nama Use Case	Mengubah Data Pegawai
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk mengubah data pegawai.
Kondisi Awal	Aktor telah menjalankan use case Melihat Data Pegawai
Kondisi Akhir	Informasi data pegawai yang diubah tersimpan
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih salah satu baris pegawai.	2. Sistem menyajikan informasi tersebut dalam

	form.
3. Aktor melakukan perubahan pada tempat yang tersedia. Kemudian aktor menekan tombol "Simpan"	4. Sistem menyimpan informasi form ke dalam basis data.
	5. Sistem menjalankan <i>use case</i> Melihat Data Pegawai
Pengecualian I :	
Perubahan menyebabkan adanya pasangan Nama – NIP ganda dalam sistem	
	a. Sistem tidak menyimpan data tersebut kedalam sistem
	b. Sistem menampilkan pesan bahwa pasangan Nama- NIP telah ada dalam sistem
	c. Sistem menjalankan <i>use case</i> Melihat Data Pegawai
Pengecualian II :	
Perubahan menyebabkan adanya NIP ganda dalam sistem	
	a. Sistem tidak menyimpan data tersebut kedalam sistem
	b. Sistem menampilkan pesan bahwa NIP telah ada dalam sistem
	c. Sistem menjalankan <i>use case</i> Melihat Data Pegawai
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• Spesifikasi Use Case Menghapus data pegawai

Nama Use Case	Menghapus Data Pegawai
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk menghapus data pegawai.
Kondisi Awal	Aktor telah menjalankan <i>use case</i> Melihat Data Pegawai
Kondisi Akhir	Informasi Data Pegawai yang dipilih terhapus dari basis data sistem.
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih salah satu baris pegawai . Kemudian aktor menekan tombol " <i>Hapus</i> "	2. Sistem menghapus data pegawai telah dipilih.
	3. Sistem menjalankan <i>use case</i> Melihat Data Pegawai

• Spesifikasi Use Case Menambah kegiatan kerja PDKB

Nama Use Case	Menambahkan Kegiatan Kerja PDKB
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk menambah kegiatan kerja.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengurus
Kondisi Akhir	Terdapat data pegawai baru dalam basis data sistem
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih menu Kerja PDKB, kemudian menekan tombol "Tambah Data"	2. Sistem menampilkan form yang berfungsi untuk memasukan Langkah kerja, Peralatan, Syarat kerja, Pelaksana, dan keterangan
3. Aktor memasukan informasi Langkah kerja,	4. Sistem menyimpan data dalam basis data

peralatan, Syarat kerja, Pelaksana, dan keterangan. Kemudian menekan tombol "Simpan"
--

Pengecualian I :

Kegiatan kerja telah ada dalam sistem

	a.Sistem tidak menyimpan data tersebut kedalam sistem
	b.Sistem menampilkan pesan bahwa kegiatan kerja telah ada dalam sistem
	c. Sistem kembali ke langkah no 2.

Aturan Khusus

- Tidak boleh ada kolom isian yang tidak terisi

• Spesifikasi Use Case Melihat kegiatan kerja PDKB

Nama Use Case	Melihat Kegiatan Kerja PDKB
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk melihat Kegiatan Kerja PDKB yang ada.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengurus
Kondisi Akhir	Informasi mengenai kegiatan kerja ditampilkan
Aksi dari Aktor	Tanggapan dari Sistem
1.Aktor memilih menu Kerja PDKB, kemudian menekan tombol "Lihat Data".	2.Sistem menampilkan daftar kerja seluruh Kegiatan yang ada.
3.Aktor memilih salah satu baris pegawai.	4.Sistem akan menampilkan informasi kegiatan kerja tersebut

• Spesifikasi Use Case Mengganti kegiatan kerja PDKB

Nama Use Case	Mengubah Kegiatan kerja
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk mengubah Kegiatan kerja.
Kondisi Awal	Aktor telah menjalankan <i>use case</i> Melihat Kegiatan kerja
Kondisi Akhir	Informasi data pegawai yang diubah tersimpan
Aksi dari Aktor	Tanggapan dari Sistem
1.Aktor memilih salah satu Jenis Kegiatan Kerja.	2.Sistem menyajikan informasi tersebut dalam form.
3.Aktor melakukan perubahan pada tempat yang tersedia. Kemudian aktor menekan tombol "Simpan"	4.Sistem menyimpan informasi form ke dalam basis data.
	5.Sistem menjalankan <i>use case</i> Melihat Kegiatan Kerja
Pengecualian I :	
Perubahan menyebabkan adanya Kegiatan Kerja ganda dalam sistem	
	a.Sistem tidak menyimpan data tersebut kedalam sistem
	b.Sistem menampilkan pesan bahwa Kegiatan kerja telah ada dalam sistem
	c. Sistem menjalankan <i>use case</i> Melihat Kegiatan Kerja
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• Spesifikasi Use Case Menghapus kegiatan kerja PDKB

Nama Use Case	Menghapus Kegiatan Kerja
Aktor	Pengurus
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengurus untuk menghapus Kegiatan Kerja.
Kondisi Awal	Aktor telah menjalankan <i>use case</i> Melihat Kegiatan Kerja
Kondisi Akhir	Informasi yang dipilih terhapus dari basis data sistem.
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih salah satu baris Kegiatan Kerja. Kemudian aktor menekan tombol " <i>Hapus</i> "	2. Sistem menghapus data Kegiatan Kerja telah dipilih.
	3. Sistem menjalankan <i>use case</i> Melihat Kegiatan Kerja

4.1.2.3.3 Spesifikasi Use Case Pengelolaan Informasi

• Spesifikasi Use Case Melihat Medical Record

Nama Use Case	Melihat Medical Record
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk melihat track data kesehatan yang ada.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola
Kondisi Akhir	Seluruh track data kesehatan ditampilkan oleh sistem
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih menu "Lihat Medical Record"	2. Sistem menampilkan seluruh informasi Melihat Data Kesehatan yang tersimpan dalam basis data.

• Spesifikasi *Use Case* Menyusun Jadwal Baru Operasi Perbaikan dan Perawatan

Nama Use Case	Menyusun Jadwal Baru Operasi Perbaikan dan Perawatan	
Aktor	Pengelola	
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk Menyusun Jadwal Baru Operasi Perbaikan dan Perawatan.	
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola	
Kondisi Akhir	Terdapat Jadwal baru dalam basis data sistem	
	Aksi dari Aktor	Tanggapan dari Sistem
	1. Aktor memilih menu "Susun Jadwal"	2. Sistem menampilkan form yang berfungsi untuk memasukkan Jenis Kegiatan, Pelaksana, Waktu, Persyaratan, dan keterangan.
	3. Aktor memasukkan informasi Jenis Kegiatan, Pelaksana, Waktu, Persyaratan, dan keterangan. Kemudian aktor menekan tombol "Simpan"	4. Sistem menyimpan data dalam basis data 5. Sistem Menampilkan <i>Use Case</i> melihat jadwal operasi
	Pengecualian I :	
	Jenis kegiatan telah ada dalam sistem	
		a. Sistem tidak menyimpan data tersebut kedalam sistem
		b. Sistem menampilkan pesan bahwa Jenis kegiatan kerja telah ada dalam sistem
		c. Sistem kembali ke langkah no 2.
	Pengecualian II :	
	Waktu telah ada dalam sistem	
		a. Sistem tidak menyimpan data tersebut kedalam

	sistem
	b.Sistem menampilkan pesan bahwa Waktu berbenturan dengan kegiatan lain
	c. Sistem kembali ke langkah no 2.
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• Spesifikasi *Use Case* Menambah Jadwal Operasi Perbaikan dan Perawatan

Nama Use Case	Menambah Jadwal Operasi Perbaikan dan Perawatan	
Aktor	Pengelola	
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk Menambah Jadwal Operasi Perbaikan dan Perawatan.	
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola	
Kondisi Akhir	Terdapat Tambahan susunan data baru di Jadwal dalam basis data sistem	
	Aksi dari Aktor	Tanggapan dari Sistem
	1.Aktor memilih menu "Tambah Susunan Jadwal"	2.Sistem menampilkan form yang berfungsi untuk memasukan Jenis Kegiatan, Pelaksana, Waktu, Persyaratan, dan keterangan.
	3.Aktor memasukan informasi Jenis Kegiatan, Pelaksana, Waktu, Persyaratan, dan keterangan. Kemudian aktor menekan tombol " <i>Simpan</i> "	4.Sistem menyimpan data dalam basis data 5. Sistem menampilkan <i>Use Case</i> melihat Jadwal Operasi Perbaikan & Perawatan
Pengecualian I :		
Jenis kegiatan telah ada dalam sistem		
	a.Sistem tidak menyimpan data tersebut kedalam sistem	

	b.Sistem menampilkan pesan bahwa Jenis kegiatan kerja telah ada dalam sistem
	c. Sistem kembali ke langkah no 2.
Pengecualian II :	
Waktu telah ada dalam sistem	
	a.Sistem tidak menyimpan data tersebut ke dalam sistem
	b.Sistem menampilkan pesan bahwa Waktu berbenturan dengan kegiatan lain
	c. Sistem kembali ke langkah no 2.
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• Spesifikasi Use Case Melihat Jadwal Operasi Perbaikan dan Perawatan

Nama Use Case	Melihat Jadwal Operasi Perbaikan dan Perawatan
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk Menampilkan Jadwal Operasi Perbaikan dan Perawatan.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola
Kondisi Akhir	Seluruh jadwal ditampilkan oleh sistem
Aksi dari Aktor	
Tanggapan dari Sistem	
1.Aktor memilih menu "Lihat Jadwal"	2. Sistem menampilkan informasi Susunan Jadwal yang tersimpan dalam basis data.

• Spesifikasi Use Case Mengganti Jadwal Operasi Perbaikan dan Perawatan

Nama Use Case	Mengganti Jadwal Operasi Perbaikan dan Perawatan
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk Mengganti Jadwal Operasi Perbaikan dan Perawatan.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola
Kondisi Akhir	Terdapat susunan data baru di Jadwal dalam basis data sistem
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih menu "Ganti Susunan Jadwal"	2. Sistem menampilkan form yang berfungsi untuk memasukkan Jenis Kegiatan, Pelaksana, Waktu, Persyaratan, dan keterangan.
3. Aktor memasukkan informasi Jenis Kegiatan, Pelaksana, Waktu, Persyaratan, dan keterangan. Kemudian aktor menekan tombol "Simpan"	4. Sistem menyimpan data dalam basis data 5. Sistem menampilkan Use Case melihat Jadwal Operasi Perbaikan & Perawatan
Pengecualian I :	
Jenis kegiatan telah ada dalam sistem	
	a. Sistem tidak menyimpan data tersebut kedalam sistem
	b. Sistem menampilkan pesan bahwa Jenis kegiatan kerja telah ada dalam sistem
	c. Sistem kembali ke langkah no 2.
Pengecualian II :	
Waktu telah ada dalam sistem	
	a. Sistem tidak menyimpan data tersebut kedalam sistem

	b.Sistem menampilkan pesan bahwa Waktu berbenturan dengan kegiatan lain
	c. Sistem kembali ke langkah no 2.
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• **Spesifikasi Use Case Menghapus Jadwal Operasi Perbaikan dan Perawatan**

Nama Use Case	Menghapus Jadwal Operasi Perbaikan dan Perawatan
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk Menghapus Jadwal Operasi Perbaikan dan Perawatan.
Kondisi Awal	Aktor telah menjalankan Use Case Melihat Jadwal Operasi Perbaikan dan Perawatan
Kondisi Akhir	Susunan Jadwal yang dipilih terhapus dari basis data
Aksi dari Aktor	Tanggapan dari Sistem
1.Aktor memilih menu "Hapus Jadwal"	2. Sistem Menghapus Susunan Jadwal yang tersimpan dalam basis data.
	3.Sistem menjalankan use case Melihat Susunan Jadwal

• **Spesifikasi Use Case Menyusun Analisis Pekerjaan dan Pembagian Tugas**

Nama Use Case	Menyusun Analisis Pekerjaan dan Pembagian Tugas
Aktor	Perancang
Deskripsi	Sistem harus menyediakan fasilitas bagi Perancang untuk Menyusun Analisis Pekerjaan dan Pembagian Tugas
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Perancang

Kondisi Akhir	Terdapat Susunan Analisis Pekerjaan dan Pembagian Tugas dalam basis data sistem	
	Aksi dari Aktor	Tanggapan dari Sistem
	1. Aktor memilih menu "Susun Analisis Pekerjaan dan Pembagian Tugas"	2. Sistem menampilkan form yang berfungsi untuk memasukan Jenis Kegiatan, Pelaksana, Anggota tim, job Description, dan Identifikasi Hazard
	3. Aktor memasukan informasi Jenis Kegiatan, Pelaksana, Anggota tim, job Description, dan Identifikasi Hazard. Kemudian aktor menekan tombol "Simpan"	4. Sistem menyimpan data dalam basis data 5. Sistem menampilkan <i>Use Case</i> Melihat Analisis Pekerjaan dan Pembagian Tugas
Pengecualian I :		
Jenis pekerjaan telah ada dalam sistem		
		a. Sistem tidak menyimpan data tersebut kedalam sistem
		b. Sistem menampilkan pesan bahwa Jenis kegiatan kerja telah ada dalam sistem
		c. Sistem kembali ke langkah no 2.
Aturan Khusus		
- Tidak boleh ada kolom isian yang tidak terisi		

• **Spesifikasi Use Case Melihat Analisis Pekerjaan dan Pembagian Tugas**

Nama Use Case	Melihat Analisis Pekerjaan dan Pembagian Tugas
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk Menampilkan Analisis Pekerjaan dan Pembagian Tugas.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola

Kondisi Akhir	Seluruh Daftar ditampilkan oleh sistem	
Aksi dari Aktor		Tanggapan dari Sistem
1. Aktor memilih menu "Lihat Daftar Analisis Pekerjaan dan Pembagian Tugas"		2. Sistem menampilkan informasi Susunan Analisis Pekerjaan dan Pembagian Tugas dalam basis data.

• **Spesifikasi Use Case Mengubah Analisis Pekerjaan dan Pembagian Tugas**

Nama Use Case	Mengubah Analisis Pekerjaan dan Pembagian Tugas	
Aktor	Pengelola	
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk Mengubah Analisis Pekerjaan dan Pembagian Tugas.	
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola dan telah menjalankan Use Case Melihat Analisis Pekerjaan dan Pembagian Tugas	
Kondisi Akhir	Terdapat susunan data baru dalam basis data sistem	
Aksi dari Aktor		Tanggapan dari Sistem
1. Aktor memilih menu "Susun Analisis Pekerjaan dan Pembagian Tugas"		2. Sistem menampilkan form yang berfungsi untuk memasukan Jenis Kegiatan, Pelaksana, Anggota tim, job Description, dan Identifikasi Hazard
3. Aktor memasukan informasi Jenis Kegiatan, Pelaksana, Anggota tim, job Description, dan Identifikasi Hazard. Kemudian aktor menekan tombol "Simpan"		4. Sistem menyimpan data dalam basis data 5. Sistem menampilkan Use Case melihat Analisis Pekerjaan dan Pembagian Tugas
Pengecualian I :		
Jenis pekerjaan telah ada dalam sistem		
		a. Sistem tidak menyimpan data tersebut kedalam sistem

	b.Sistem menampilkan pesan bahwa Jenis kegiatan kerja telah ada dalam sistem
	c. Sistem kembali ke langkah no 2.
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• Spesifikasi *Use Case* Menghapus Analisis Pekerjaan dan Pembagian Tugas

Nama Use Case	Menghapus Analisis Pekerjaan dan Pembagian Tugas
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengelola untuk Menghapus Analisis Pekerjaan dan Pembagian Tugas
Kondisi Awal	Aktor telah menjalankan <i>Use Case</i> Melihat Analisis Pekerjaan dan Pembagian Tugas
Kondisi Akhir	Susunan data yang dipilih terhapus dari basis data
Aksi dari Aktor	
Tanggapan dari Sistem	
1.Aktor memilih menu "Hapus Analisis Pekerjaan dan Pembagian Tugas".	2. Sistem Menghapus Susunan data yang tersimpan dalam basis data.
	3.Sistem menjalankan <i>use case</i> Melihat Analisis Pekerjaan dan Pembagian Tugas

• Spesifikasi *Use Case* Menyusun Pemberitahuan Penyelesaian Pekerjaan

Nama Use Case	Menyusun Pemberitahuan Penyelesaian Pekerjaan
Aktor	Pengawas
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengawas untuk Menyusun Pemberitahuan Penyelesaian Pekerjaan
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengawas

Kondisi Akhir	Terdapat Susunan Pemberitahuan Penyelesaian Pekerjaan dalam basis data sistem	
	Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Pemberitahuan Penyelesaian Pekerjaan". Kemudian aktor menekan tombol "Susun"		2. Sistem menampilkan form yang berfungsi untuk memasukan no pekerjaan, tanggal, tempat, waktu selesai pekerjaan
3. Aktor memasukan no pekerjaan, tanggal, tempat, waktu selesai pekerjaan. Kemudian aktor menekan tombol "Simpan"		4. Sistem menyimpan data dalam basis data 5. Sistem menampilkan Use Case melihat Pemberitahuan Pekerjaan
Aturan Khusus		
- Tidak boleh ada kolom isian yang tidak terisi		

• Spesifikasi Use Case Melihat Pemberitahuan Penyelesaian Pekerjaan

Nama Use Case	Melihat Pemberitahuan Penyelesaian Pekerjaan	
Aktor	Pengawas	
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengawas untuk Menampilkan Pemberitahuan Penyelesaian Pekerjaan.	
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengawas	
Kondisi Akhir	Jadwal Penyelesaian ditampilkan oleh sistem	
	Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Pemberitahuan Penyelesaian Pekerjaan". Kemudian aktor menekan tombol "Lihat"		2. Sistem menampilkan informasi Susunan Pemberitahuan Penyelesaian Pekerjaan.

• Spesifikasi Use Case Mengubah Pemberitahuan Penyelesaian Pekerjaan

Nama Use Case	Mengubah Pemberitahuan Penyelesaian Pekerjaan
Aktor	Pengawas
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengawas untuk Mengubah Pemberitahuan Penyelesaian Pekerjaan
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengawas dan telah menjalankan Use Case Melihat Pemberitahuan Penyelesaian Pekerjaan
Kondisi Akhir	Terdapat Susunan data baru di “Pemberitahuan Penyelesaian Pekerjaan” dalam basis data sistem
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu ” Pemberitahuan Penyelesaian Pekerjaan”. Kemudian aktor menekan tombol “Ubah”	2. Sistem menampilkan form yang berfungsi untuk memasukan no pekerjaan, tanggal, tempat, waktu selesai pekerjaan
3. Aktor memasukan no pekerjaan, tanggal, tempat, waktu selesai pekerjaan. Kemudian aktor menekan tombol “Simpan”	4. Sistem menyimpan data dalam basis data 5. Sistem menampilkan Use Case melihat Pemberitahuan Penyelesaian Pekerjaan
Aturan Khusus	
- Tidak boleh ada kolom isian yang tidak terisi	

• Spesifikasi Use Case Menghapus Pemberitahuan Penyelesaian Pekerjaan

Nama Use Case	Menghapus Pemberitahuan Penyelesaian Pekerjaan
Aktor	Pengawas
Deskripsi	Sistem harus menyediakan fasilitas bagi Pengawas untuk Menghapus daftar Pemberitahuan Penyelesaian Pekerjaan
Kondisi Awal	Aktor telah menjalankan Use Case Melihat Pemberitahuan Penyelesaian Pekerjaan

Kondisi Akhir	Susunan data yang dipilih terhapus dari basis data	
Aksi dari Aktor		Tanggapan dari Sistem
1. Aktor memilih menu "Pemberitahuan Penyelesaian Pekerjaan". Kemudian aktor menekan tombol "Hapus"		2. Sistem Menghapus Susunan data yang tersimpan dalam basis data.
		3. Sistem menjalankan <i>use case</i> Melihat Pemberitahuan Penyelesaian Pekerjaan

• Spesifikasi Use Case Merancang Usulan Nama Pekerja

Nama Use Case	Merancang Usulan Nama Pekerja	
Aktor	Perancang	
Deskripsi	Sistem harus menyediakan fasilitas bagi Perancang untuk dapat membuat usulan calon pekerja berdasarkan Jenis Pekerjaan yang dibutuhkan dalam PDKB	
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Perancang	
Kondisi Akhir	Rancangan usulan calon pekerja tersimpan dalam basis data	
Aksi dari Aktor		Tanggapan dari Sistem
1. Aktor memilih menu "Rancang Usulan Pekerja"		2. Sistem akan menampilkan daftar Pekerjaan yang ada.
3. Aktor memilih salah satu baris dari daftar Pekerjaan.		4. Sistem akan menampilkan daftar Pekerjaan yang telah di simpan.
5. Aktor memilih salah satu dari daftar Pekerjaan yang diterima.		6. Sistem akan menampilkan daftar nama yang memenuhi syarat untuk Pekerjaan yang tertera dalam daftar, dimasing-masing baris terdapat sebuah tempat sebagai penanda.
7. Aktor menandai nama-nama yang diusulkan. Kemudian aktor menekan tombol "Simpan"		8. Sistem menyimpan nama-nama yang telah dipilih.

Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Ubah Rancangan Usulan Pekerja"	2. Sistem akan menampilkan daftar Pekerjaan yang ada.
3. Aktor memilih salah satu baris dari daftar pekerjaan.	4. Sistem akan menampilkan Jenis Pekerjaan dan nama-nama yang diusulkan.
5. Aktor memilih salah satu baris dari daftar nama Pekerja dan nama-nama yang diusulkan.	6. Sistem akan menampilkan daftar nama yang memenuhi syarat untuk Melaksanakan pekerjaan dan belum diusulkan, dimasing-masing baris terdapat tempat penanda
7. Aktor menandai nama-nama yang diusulkan. . Kemudian aktor menekan tombol "Simpan"	8. Sistem akan menyimpan perubahan usulan nama. 9. Sistem akan menampilkan <i>Use Case</i> melihat Usulan Nama Calon Pekerja

• Spesifikasi *Use Case* Menambah Usulan Nama Calon Pekerja

Nama Use Case	Menambah Usulan Nama Calon Pekerja
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk dapat menambah nama-nama calon pekerja yang akan diusulkan.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola
Kondisi Akhir	Usulan calon pekerja yang baru tersimpan dalam basis data ditampilkan oleh sistem

Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Tambah Usulan Nama Pekerja"	2. Sistem akan menampilkan jenis Pekerjaan dan nama yang diusulkan.
3. Aktor memilih salah satu baris dari daftar pekerjaan.	4. Sistem akan menampilkan Jenis Pekerjaan dan nama-nama yang diusulkan.
5. Aktor memilih salah satu baris dari daftar nama	6. Sistem akan menampilkan form untuk pengisian

Pekerja dan nama-nama yang diusulkan.	daftar nama baru
7.Aktor mengisi nama calon pekerja baru. . Kemudian aktor menekan tombol " <i>Simpan</i> "	8.Sistem akan menyimpan penambahan usulan nama. 9.Sistem menampilkan <i>Use Case</i> melihat Usulan Nama Calon Pekerja

• Spesifikasi *Use Case* Menghapus Usulan Nama Calon Pekerja

Nama Use Case	Menghapus Nama Calon Pekerja
Aktor	Pengelola, Pengawas
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola dan pengawas untuk dapat menghapus nama-nama calon pekerja yang diusulkan.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola atau Pengawas
Kondisi Akhir	Usulan calon pekerja yang telah diubah tersimpan dalam basis data ditampilkan oleh sistem
Aksi dari Aktor	Tanggapan dari Sistem
1.Aktor memilih menu "Hapus Rancang Usulan Pekerja"	2.Sistem akan menampilkan Jenis Kegiatan yang ada.
3.Aktor memilih salah satu baris dari Jenis Kegiatan	4.Sistem akan menampilkan Jenis Kegiatan dan nama-nama yang diusulkan.
5.Aktor memilih salah satu baris dari daftar Jenis Kegiatan dan nama-nama yang diusulkan . Kemudian aktor menekan tombol " <i>Hapus</i> "	6.Sistem akan menghapus nama yang dipilih tersebut. 7.sistem akan menampilkan <i>Use Case</i> melihat Usulan Nama Calon Pekerja

• Spesifikasi Use Case Menyetujui Usulan Nama Pekerja

Nama Use Case	Menyetujui Usulan Pekerja	
Aktor	Pengawas	
Deskripsi	Sistem harus menyediakan fasilitas bagi pengawas untuk menyetujui nama-nama calon pekerja yang diusulkan. Setelah usulan nama-nama calon pekerja disetujui, maka tidak bisa lagi dilakukan perubahan terhadap nama-nama calon pekerja.	
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengawas	
Kondisi Akhir	Usulan calon pekerja yang telah disetujui akan tersimpan dalam basis data	
	Aksi dari Aktor	Tanggapan dari Sistem
	1. Aktor memilih menu "Persetujuan Rancang Usulan Pekerja"	2. Sistem akan menampilkan Jenis Pekerjaan yang ada.
	3. Aktor memilih salah satu baris dari Jenis Pekerjaan	4. Sistem akan menampilkan daftar nama-nama Pekerja yang diterima dari jenis pekerjaan tersebut.
	5. Aktor memilih salah satu baris dari daftar nama pekerja tersebut.	6. Sistem akan menampilkan daftar nama-nama pekerja yang diusulkan untuk pekerjaan tersebut
	7. Aktor menekan tombol "Disetujui"	8. Informasi nama-nama diusulkan untuk Pekerjaan tersebut tercatat dalam basis data 9. Sistem menampilkan daftar nama disetujui

• Spesifikasi Use Case Membuat Surat Tugas

Nama Use Case	Membuat Surat Tugas
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk membuat surat

	penugasan bagi pegawai yang diberangkatkan melaksanakan kegiatan kerja
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola
Kondisi Akhir	Terbentuk suatu surat penugasan bagi nama-nama yang akan diberangkatkan
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Buat Surat Tugas"	2. Sistem akan menampilkan tempat digunakan untuk memuat informasi mengenai nomor surat undangan, nomor surat penugasan, tanggal penugasan dan tanggal kembali.
3. Aktor mengisi informasi mengenai nomor surat undangan, nomor surat penugasan, tanggal penugasan, dan tanggal kembali.	4. Sistem akan menampilkan daftar usulan nama.
5. Aktor memilih salah satu baris dari daftar usulan nama.	6. Sistem akan menampilkan daftar nama-nama peserta yang diusulkan untuk kegiatan kerja tersebut disertai penanda dimasing masing baris.
7. Aktor menandai nama-nama yang akan ditugaskan. . Kemudian aktor menekan tombol "Simpan"	8. Sistem akan menyimpan informasi penugasan dalam basis data 9. Sistem akan menampilkan <i>Use Case</i> melihat Surat Tugas

• Spesifikasi *Use Case* Melihat Surat Tugas

Nama Use Case	Melihat Surat Tugas
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk melihat surat penugasan bagi pegawai yang diberangkatkan mengikuti pekerjaan PDKB.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola

Kondisi Akhir	Informasi surat penugasan bagi nama-nama yang akan diberangkatkan ditampilkan oleh sistem
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih menu "Lihat Surat Tugas"	2. Sistem akan menampilkan daftar surat tugas yang ada.
3. Aktor memilih salah satu baris dari daftar surat tugas.	4. Sistem akan menampilkan informasi surat tugas yang dipilih

• Spesifikasi *Use Case* Mengubah Surat Tugas

Nama Use Case	Mengubah Surat Tugas
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk mengubah surat penugasan bagi pegawai yang diberangkatkan mengikuti pekerjaan PDKB.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola
Kondisi Akhir	Perubahan informasi surat tugas tersimpan dalam basis data
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor memilih menu "Ubah Surat Tugas"	2. Sistem akan menampilkan sub menu "Ubah Informasi Surat Tugas", sub menu "Ubah Nama dalam Surat Tugas", dan sub menu "Tambah Nama dalam Surat Tugas".
Alternatif 1 :	
Aktor memilih sub menu " Ubah Informasi Surat Tugas"	
	3. Sistem akan menampilkan daftar surat tugas yang ada.
4. Aktor memilih salah satu baris dari daftar surat tugas.	5. Sistem akan menampilkan tempat yang berisi informasi nomor surat undangan, nomor surat

	penugasan, tanggal pemberangkatan dan tanggal kembali.
6. Aktor melakukan perubahan terhadap informasi terkait surat tugas. . Kemudian aktor menekan tombol " <i>Simpan</i> "	7. Sistem akan menyimpan perubahan yang telah dilakukan
Alternatif 2 :	
Aktor memilih sub menu " Ubah Nama dalam Surat Tugas"	
	8. Sistem akan menampilkan daftar surat tugas yang ada.
9. Aktor memilih salah satu baris dari daftar surat tugas.	10. Sistem akan menampilkan nama-nama yang ditugaskan melalui surat tersebut
11. Aktor memilih salah satu baris dari daftar nama .	12. Sistem akan menampilkan daftar pilihan yang berisi informasi nama yang belum ditugaskan untuk pekerjaan tersebut.
13. Aktor memilih salah satu nama dari daftar nama. Kemudian aktor menekan tombol " <i>Ubah</i> "	14. Sistem mengganti nama yang ditugaskan
Alternatif 4 :	
Aktor memilih sub menu " Tambah Nama Ditugaskan"	
	15. Sistem akan menampilkan daftar surat tugas yang ada.
16. Aktor memilih salah satu baris dari daftar surat tugas.	17. Sistem akan menampilkan daftar nama yang belum ditugaskan untuk Pekerjaan tersebut disertai dengan penanda.
18. Aktor menandai <i>checkbox</i> dan kemudian menekan tombol "Tambah"	19. Sistem menambahkan nama-nama tersebut dalam surat tugas.
	20. Sistem menampilkan <i>Use Case</i> melihat Surat Tugas

• Spesifikasi Use Case Menghapus Surat Tugas

Nama Use Case	Menghapus Surat Tugas
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk menghapus informasi terkait penugasan pegawai yang diberangkatkan mengikuti pekerjaan PDKB.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola
Kondisi Akhir	Informasi terkait surat tugas terhapus dari basis data
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Hapus Surat Tugas"	2. Sistem akan menampilkan sub menu "Hapus Informasi Surat Tugas" dan "Hapus Nama dalam Surat Tugas".
Alternatif 1 :	
Aktor memilih sub menu " Hapus Informasi Surat Tugas"	
	3. Sistem akan menampilkan daftar surat tugas yang ada.
4. Aktor memilih salah satu baris dari daftar surat tugas.	5. Sistem akan menghapus informasi terkait surat tugas tersebut
Alternatif 4 :	
Aktor memilih sub menu " Hapus Nama Ditugaskan"	
	6. Sistem akan menampilkan daftar nama yang belum ditugaskan untuk Pekerjaan tersebut disertai dengan penanda.
7. Aktor menandai nama yang tang tidak jadi diusulkan. Kemudian aktor menekan tombol "Hapus"	8. Sistem menghapus nama-nama yang ditandai tersebut dari surat tugas.

• Spesifikasi Use Case Mengawasi Kegiatan Pekerjaan

Nama Use Case	Mengawasi Kegiatan Pekerjaan
Aktor	Pengawas
Deskripsi	Sistem harus menyediakan fasilitas bagi pengawas untuk melihat nama-nama yang diusulkan, diberangkatkan, dan dapat melihat hasil evaluasi pekerjaan.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengawas
Kondisi Akhir	Informasi yang dibutuhkan dapat disajikan
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor memilih menu "Pengawasan Kegiatan Pekerjaan"	2. Sistem menyajikan sub menu "Melihat Usulan Nama", "Melihat Penugasan Pegawai", dan "Melihat Hasil Evaluasi"
Alternatif 1 :	
Aktor memilih sub menu "Melihat Usulan Nama"	
	3. Sistem menjalankan Use Case Melihat Nama Calon Peserta
Alternatif 2 :	
Aktor memilih sub menu "Melihat Penugasan Pegawai"	
	4. Sistem menjalankan Use Case Melihat Surat Tugas
Alternatif 3 :	
Aktor memilih sub menu "Melihat Hasil Evaluasi"	
	5. Sistem menjalankan Use Case Melihat Dampak pekerjaan

• Spesifikasi Use Case Mencetak Rencana Kerja

Nama Use Case	Mencetak Rencana Kerja
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak rencana Kerja periodik.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola dan telah menjalankan Use Case Melihat Rencana Kerja
Kondisi Akhir	Informasi Rencana Kerja Periodik dapat tercetak
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor menekan tombol "Print"	2. Sistem akan menyajikan informasi dalam sebuah halaman dengan format seperti pada hasil pencetakan dan dilanjutkan dengan mencetak halaman tersebut

• Spesifikasi Use Case Mencetak Usulan Nama

Nama Use Case	Mencetak Usulan Nama
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak daftar usulan nama.
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola dan telah menjalankan Use Case Melihat Nama Calon Pekerja
Kondisi Akhir	Informasi usulan namaPekerja dapat tercetak
Aksi dari Aktor	
Tanggapan dari Sistem	
1. Aktor menekan tombol "Print"	2. Sistem akan menyajikan informasi dalam sebuah halaman dengan format seperti pada hasil pencetakan dan dilanjutkan dengan mencetak

1. Aktor menekan tombol "Print"	2. Sistem akan menyajikan informasi dalam sebuah halaman dengan format seperti pada hasil pencetakan dan dilanjutkan dengan mencetak halaman tersebut
---------------------------------	---

• Spesifikasi *Use Case* Mencetak Surat Pemberitahuan Penyelesaian Pekerjaan

Nama Use Case	Mencetak Surat Pemberitahuan Penyelesaian Pekerjaan	
Aktor	Pengelola	
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak Surat Pemberitahuan Penyelesaian Pekerjaan	
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola dan telah menjalankan <i>Use Case</i> Melihat Surat	
Kondisi Akhir	Surat Surat Pemberitahuan Penyelesaian Pekerjaan bagi pekerja dapat tercetak	
	Aksi dari Aktor	Tanggapan dari Sistem
	1. Aktor menekan tombol "Print"	2. Sistem akan menyajikan informasi dalam sebuah halaman dengan format seperti pada hasil pencetakan dan dilanjutkan dengan mencetak halaman tersebut

• Spesifikasi *Use Case* Mencetak Hasil Evaluasi Dampak Pekerjaan

Nama Use Case	Mencetak Surat Hasil Evaluasi Dampak Pekerjaan
Aktor	Pengelola
Deskripsi	Sistem harus menyediakan fasilitas bagi pengelola untuk mencetak hasil evaluasi dampak Pekerjaan
Kondisi Awal	Aktor telah Login sebagai User yang berperan sebagai Pengelola dan telah menjalankan <i>Use Case</i> Melihat Penilaian Dampak Pekerjaan

Kondisi Akhir	Hasil evaluasi para pegawai dapat tercetak
Aksi dari Aktor	Tanggapan dari Sistem
1. Aktor menekan tombol "Print"	2. Sistem akan menyajikan informasi dalam sebuah halaman dengan format seperti pada hasil pencetakan dan dilanjutkan dengan mencetak halaman tersebut

4.2 Perancangan

Use case diagram dan *use case specification* yang telah dimodelkan pada tahap analisis kebutuhan memberikan sebuah pandangan terhadap kebutuhan fungsional dari *user* yang harus disediakan oleh sistem perangkat lunak.

Perancangan pada tugas akhir ini akan dimodelkan dengan beberapa diagram yaitu *class diagram*, dan *sequence diagram*.

Diagram klas memberikan gambaran pemodelan elemen-elemen klas yang membentuk sebuah perangkat lunak. Klas bisa didapatkan dengan menganalisis secara lebih detail terhadap *use case* yang telah dimodelkan dan merealisasikannya

Sequence diagram digunakan untuk memodelkan aliran jalannya proses yang ditunjukkan dengan interaksi antar objek atau klas, dan disusun berdasarkan urutan waktu.

4.2.1 Class Diagram

Diagram Klas merupakan diagram yang paling umum dijumpai dalam pemodelan berbasis object. [BOO-98] Diagram Klas menampilkan sederet penjelasan mengenai klas yang digunakan dalam implementasi.

Kandidat klas diperoleh melalui kosa kata yang merupakan jenis kata benda yang dapat kita temukan dalam penjelasan sistem yang akan kita bangun atau dapat juga melalui *use case* yang telah dibuat sebelumnya [BOO-98]. Setiap kata benda yang ditemukan dapat dikenali karakteristiknya, karakteristik ini yang menjadi atribut bagi klas yang dirancang. Masing-masing kata benda yang telah ditemukan tentunya

memiliki peran yang dapat dikenali dalam kaitannya dengan sistem yang akan dibangun, peran inilah yang nantinya akan menjadi operasi atau *method* dalam kelas yang dirancang.

Pada tabel dibawah ini yaitu tabel 4.8 akan ditampilkan *use case – use case* yang ada dalam sistem yang akan dibangun. Dari daftar *use case* tersebut akan didapatkan kata benda – kata benda yang menjadi kandidat kelas.

Use Case	Kandidat Class
1. Login	▪ User
2. Logout	▪ User
3. Menambahkan <i>Account</i>	▪ <i>Administrator</i> ▪ <i>Account</i>
4. Melihat <i>Account</i>	▪ <i>Administrator</i> ▪ <i>Account</i>
5. Mengubah <i>Account</i>	▪ <i>Administrator</i> ▪ <i>Account</i>
6. Menghapus <i>Account</i>	▪ <i>Administrator</i> ▪ <i>Account</i>
7. Menambah Data Kesehatan	▪ <i>Pengurus</i> ▪ <i>Kesehatan</i>
8. Melihat Data Kesehatan	▪ <i>Pengurus</i> ▪ <i>Kesehatan</i>
9. Mengganti Data Kesehatan	▪ <i>Pengurus</i> ▪ <i>Kesehatan</i>
10. Menghapus Data Kesehatan	▪ <i>Pengurus</i> ▪ <i>Kesehatan</i>
11. Menambah Data Pegawai	▪ <i>Pengurus</i> ▪ <i>Pegawai</i> ▪ <i>Kesehatan</i> ▪ <i>Kegiatan Kerja</i>
12. Melihat Data Pegawai	▪ <i>Pengurus</i> ▪ <i>Pegawai</i> ▪ <i>Kesehatan</i> ▪ <i>Kegiatan Kerja</i>
13. Mengganti Data Pegawai	▪ <i>Pengurus</i> ▪ <i>Pegawai</i> ▪ <i>Kesehatan</i> ▪ <i>Kegiatan Kerja</i>

14. Menghapus Data Pegawai	<ul style="list-style-type: none"> ▪ Pengurus ▪ Pegawai ▪ Kesehatan
15. Menambah kegiatan kerja PDKB	<ul style="list-style-type: none"> ▪ Kegiatan Kerja ▪ Pengurus ▪ Kegiatan Kerja ▪ Divisi
16. Melihat kegiatan kerja PDKB	<ul style="list-style-type: none"> ▪ Pengurus ▪ Kegiatan Kerja ▪ Divisi
17. Mengganti kegiatan kerja PDKB	<ul style="list-style-type: none"> ▪ Pengurus ▪ Kegiatan Kerja ▪ Divisi
18. Menghapus kegiatan kerja PDKB	<ul style="list-style-type: none"> ▪ Pengurus ▪ Kegiatan Kerja ▪ Divisi
19. Melihat Medical Record	<ul style="list-style-type: none"> ▪ Pengurus ▪ Pegawai ▪ Kesehatan ▪ Kegiatan Kerja
20. Menyusun Jadwal Operasi Perbaikan & Perawatan	<ul style="list-style-type: none"> ▪ Pengelola ▪ Jadwal ▪ Kegiatan kerja
21. Menambah Jadwal Operasi Perbaikan dan Perawatan	<ul style="list-style-type: none"> ▪ Pengelola ▪ Jadwal ▪ Kegiatan kerja
22. Melihat Jadwal Operasi Perbaikan dan Perawatan	<ul style="list-style-type: none"> ▪ Pengelola ▪ Jadwal ▪ Pegawai ▪ Kegiatan kerja
23. Mengganti Jadwal Operasi Perbaikan dan Perawatan	<ul style="list-style-type: none"> ▪ Pengelola ▪ Jadwal ▪ Kegiatan kerja
24. Menghapus Jadwal Operasi Perbaikan dan Perawatan	<ul style="list-style-type: none"> ▪ Pengelola ▪ Jadwal ▪ Kegiatan kerja
25. Menyusun Analisis Pekerjaan dan Pembagian Tugas	<ul style="list-style-type: none"> ▪ Pengelola ▪ Data_laporan ▪ Perancang
26. Melihat Analisis Pekerjaan dan Pembagian Tugas	<ul style="list-style-type: none"> ▪ Pengelola ▪ Data_laporan

27. Mengubah Analisis Pekerjaan dan Pembagian Tugas	<ul style="list-style-type: none"> ▪ Pengelola ▪ Data_laporan
28. Menghapus Analisis Pekerjaan dan Pembagian Tugas	<ul style="list-style-type: none"> ▪ Pengelola ▪ Data_laporan ▪ Perancang
29. Menyusun Pemberitahuan Penyelesaian Pekerjaan	<ul style="list-style-type: none"> ▪ Pengawas ▪ Data_laporan
30. Melihat Pemberitahuan Penyelesaian Pekerjaan	<ul style="list-style-type: none"> ▪ Pengawas ▪ Data_laporan
31. Mengubah Pemberitahuan Penyelesaian Pekerjaan	<ul style="list-style-type: none"> ▪ Pengawas ▪ Data_laporan
32. Menghapus Pemberitahuan Penyelesaian Pekerjaan	<ul style="list-style-type: none"> ▪ Pengawas ▪ Data_laporan
33. Merancang Usulan Nama Pekerja	<ul style="list-style-type: none"> ▪ Pengelola ▪ kandidat ▪ perancang
34. Melihat Nama Calon Pekerja	<ul style="list-style-type: none"> ▪ Pengelola ▪ kandidat
35. Mengubah Nama Calon Pekerja	<ul style="list-style-type: none"> ▪ Pengelola ▪ kandidat
36. Menambah Usulan Calon Pekerja	<ul style="list-style-type: none"> ▪ Pengelola ▪ kandidat
37. Menghapus Nama Calon Pekerja	<ul style="list-style-type: none"> ▪ Pengelola ▪ kandidat
38. Menyetujui Usulan Nama Pekerja	<ul style="list-style-type: none"> ▪ Pengawas ▪ kandidat
39. Membuat Surat Tugas	<ul style="list-style-type: none"> ▪ Data_Laporan ▪ pengelola
40. Melihat Surat Tugas	<ul style="list-style-type: none"> ▪ Data_Laporan ▪ pengelola
41. Mengubah Surat Tugas	<ul style="list-style-type: none"> ▪ Data_Laporan ▪ pengelola
42. Menghapus Surat Tugas	<ul style="list-style-type: none"> ▪ Data_Laporan ▪ pengelola
43. Mengawasi Kegiatan Pekerjaan	<ul style="list-style-type: none"> ▪ pengawas ▪ Kegiatan Kerja
44. Mencetak Rencana Kerja	<ul style="list-style-type: none"> ▪ cetak

45. Mencetak Usulan Nama	<ul style="list-style-type: none"> ▪ Kegiatan Kerja ▪ Cetak
46. Mencetak Surat Tugas	<ul style="list-style-type: none"> ▪ kandidat ▪ cetak ▪ Data_Laporan
47. Mencetak Surat Analisis Pekerjaan dan Pembagian Tugas	<ul style="list-style-type: none"> ▪ cetak ▪ Data_Laporan
48. Mencetak Surat Pemberitahuan Penyelesaian Pekerjaan	<ul style="list-style-type: none"> ▪ cetak ▪ Data_Laporan
49. Mencetak Hasil Evaluasi Dampak Pekerjaan	<ul style="list-style-type: none"> ▪ cetak ▪ evaluasi

Tabel 4.8 Daftar Use Case dan Kandidat Klas [Analisis]

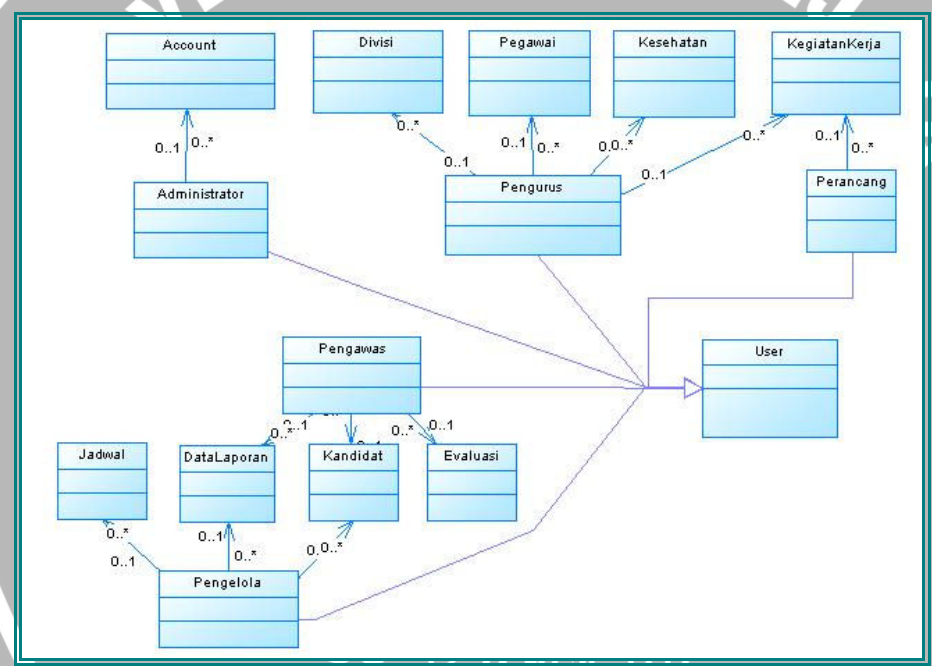
Dari tabel diatas yang memuat kandidat-kandidat klas bagi sistem yang akan dibangun, maka dapat diperoleh daftar klas yang diperlukan untuk sistem yang akan dibangun. Klas-klas tersebut dapat ditampilkan pada daftar berikut ini.

No.	Klas yang dibutuhkan sistem
1.	User
2.	Administrator
3.	Pengurus
4.	Account
5.	Kesehatan
6.	Pegawai
7.	Kegiatan Kerja
8.	Pengelola
9.	Divisi
10.	Pengawas
11.	Data Laporan
12.	Kandidat
13.	Perancang

14.	Jadwal
15.	Evaluasi

Tabel 4.9 Daftar Kelas yang dibutuhkan oleh sistem [Perancangan]

Relasi antar kelas tersebut dapat dilihat pada diagram kelas dibawah ini. Atribut dan operasi masing-masing kelas akan disampaikan pada penjelasan masing-masing kelas, hal ini dilakukan agar dapat menampilkan gambar secara sederhana yang mudah dimengerti.



Gambar 4.4 Diagram Kelas [Perancangan]

4.2.1.1 Penjelasan *Class diagram* untuk Klas *Account*

Klas ini berguna untuk memfasilitasi pencatatan user yang diizinkan untuk menggunakan sistem kedalam basis data.

<i>Account</i>
<ul style="list-style-type: none"> - <i>IdPengguna</i> : <i>Int</i> - <i>IdPegawai</i> : <i>Int</i> - <i>Username</i> : <i>Str</i> - <i>Password</i> : <i>Str</i> - <i>Level</i> : <i>Str</i>
<ul style="list-style-type: none"> + <i>memasukkan_account</i> () + <i>mengubah_account</i> () + <i>menghapus_account</i> () + <i>melihat_account</i> ()

4.2.1.2 Penjelasan *Class diagram* untuk Klas *Data_Laporan*

Klas *Data_laporan* berfungsi untuk Mengolah data Analisis pekerjaan, pemberitahuan penyelesaian pekerjaan, dan untuk mengetahui jenis tugas yang akan dilaksanakan.

<i>Data_Laporan</i>
<ul style="list-style-type: none"> - <i>IdData_Laporan</i> : <i>Int</i> - <i>AnalisisPekerjaan</i> : <i>Str</i> - <i>IdPemberitahuanKerja</i> : <i>Int</i> - <i>IdSurat_tugas</i> : <i>Int</i> - <i>Laporan</i>: <i>Str</i> - <i>keterangan</i>: <i>Str</i> - <i>jenis_kerja</i>: <i>Str</i> - <i>hazard</i>: <i>Str</i>

- + *memasukkan _PemberitahuanKerja* ()
- + *mengubah _PemberitahuanKerja* ()
- + *menghapus _PemberitahuanKerja* ()
- + *melihat _PemberitahuanKerja* ()
- + *memasukkan _Surat_tugas* ()
- + *mengubah _Surat_tugas* ()
- + *menghapus _Surat_tugas* ()
- + *melihat _Surat_tugas* ()

4.2.1.3 Penjelasan *Class diagram* untuk Klas Jadwal

Klas Jadwal berfungsi untuk Mengolah jadwal kerja yang akan dilaksanakan pegawai yang telah ditugaskan dalam system informasi.

<i>Jadwal</i>
- <i>protected IdJadwal : Int</i>
- <i>protected Data : Int</i>
+ <i>create_jadwalBaru</i> ()
+ <i>memasukkan _Jadwal</i> ()
+ <i>mengubah _Jadwal</i> ()
+ <i>menghapus _Jadwal</i> ()
+ <i>melihat _Jadwal</i> ()

4.2.1.4 Penjelasan *Class diagram* untuk Klas Divisi

Klas Divisi berfungsi untuk Mengolah Divisi yang akan dilaksanakan pegawai dalam system informasi.

<i>Divisi</i>
- <i>protected IdDivisi : Int</i>

- | |
|---------------------------------|
| - <i>protected Data : Int</i> |
| + <i>memasukkan _Divisi ()</i> |
| + <i>mengubah _Divisi ()</i> |
| + <i>menghapus _Divisi ()</i> |
| + <i>melihat _Divisi ()</i> |

4.2.1.5 Penjelasan *Class diagram* untuk Klas Kesehatan

Klas Kesehatan berfungsi untuk Mengolah Data pegawai berkaitan dengan kesehatan pegawai.

<i>Kesehatan</i>
- <i>protected IdKesehatan : Int</i>
- <i>protected Data : Int</i>
+ <i>memasukkan _dataKesehatan()</i>
+ <i>mengubah _dataKesehatan ()</i>
+ <i>menghapus _dataKesehatan ()</i>
+ <i>melihat _dataKesehatan ()</i>
+ <i>melihat _MedicalRecord ()</i>

4.2.1.6 Penjelasan *Class diagram* untuk Klas Kegiatan Kerja

Klas Kegiatan Kerja berfungsi untuk menangani jenis kegiatan kerja yang akan diberikan kepada pegawai yang bertugas, serta kegiatan kerja yang ditawarkan dan tempat kegiatan kerja dilaksanakan

<i>Kegiatan Kerja</i>
- <i>protected IdKegiatanKerja : Int</i>
- <i>protected Data : Int</i>
+ <i>memasukkan _kegiatanKerja()</i>

- + *mengubah _ kegiatanKerja ()*
- + *menghapus _ kegiatanKerja ()*
- + *melihat _ kegiatanKerja ()*
- + *mengawasi _ kegiatanKerja ()*

4.2.1.7 Penjelasan *Class diagram* untuk Klas Evaluasi

Klas Evaluasi berfungsi untuk melakukan penilaian bagi pegawai yang telah ditugaskan untuk mengikuti kegiatan kerja.

<i>Evaluasi</i>
<ul style="list-style-type: none"> - <i>protected IdEvaluasi : Int</i> - <i>protected Laporan:Str</i> - <i>protected keterangan: Str</i> - <i>protected Jenis_kerja: Str</i> - <i>protected Hazard: Str</i> - <i>protected Data : Str</i>
<ul style="list-style-type: none"> + <i>memasukkan _ Evaluasi ()</i> + <i>mengubah _ Evaluasi ()</i> + <i>menghapus _ Evaluasi ()</i> + <i>melihat _ Evaluasi ()</i>

4.2.1.8 Penjelasan *Class diagram* untuk Klas Kandidat

Klas Kandidat berfungsi untuk menangani penentuan usulan nama untuk penentuan jenis pekerjaan.

<i>kandidat</i>
<ul style="list-style-type: none"> - <i>public IdKandidat :Int</i> - <i>public Data : Int</i>

- + *memasukkan _ Kandidat ()*
- + *mengubah _ Kandidat ()*
- + *menghapus _ Kandidat ()*
- + *menyetujui _ Kandidat ()*
- + *melihat _ Kandidat ()*
- + *merancang _ Kandidat ()*

4.2.1.9 Penjelasan *Class diagram* untuk Klas Pegawai

Klas ini berfungsi untuk menangani informasi pribadi pegawai yang dibutuhkan terkait dengan pelaksanaan kegiatan diklat.

<i>pegawai</i>
- <i>protected IdPegawai : Int</i>
- <i>protected Data : Int</i>
+ <i>memasukkan _ DataPegawai ()</i>
+ <i>mengubah _ DataPegawai ()</i>
+ <i>menghapus _ DataPegawai ()</i>
+ <i>melihat _ DataPegawai ()</i>

4.2.1.10 Penjelasan *Class Diagram* untuk klas *Pengelola*

Klas ini berguna untuk menginstansi klas Jadwal, Kandidat, dan Data Laporan yang akan memanfaatkan operasi-operasi yang dimilikinya

<i>Pengelola</i>
- <i>protected isi : Str</i>
- <i>protected Judul :Str</i>
- <i>protected Data : Int</i>
+ <i>tampilstyle2 ()</i>
+ <i>tampil _ Menuatas _ pengurus()</i>

```

+ tampilAtas()
+ tampilBawah ()
+ create_jadwalBaru ()
+ memasukkan _Jadwal ()
+ mengubah _Jadwal ()
+ menghapus _Jadwal ()
+ melihat _Jadwal ()
+ memasukkan _Kandidat ()
+ mengubah _Kandidat ()
+ menghapus _Kandidat ()
+ menyetujui _Kandidat ()
+ melihat _Kandidat ()
+ memasukkan _PemberitahuanKerja ()
+ mengubah _PemberitahuanKerja ()
+ menghapus _PemberitahuanKerja ()
+ melihat _PemberitahuanKerja ()
+ memasukkan _Surat_tugas ()
+ mengubah _Surat_tugas ()
+ menghapus _Surat_tugas ()
+ melihat _Surat_tugas ()
+ Lihat_detail ()

```

4.2.1.11 Penjelasan *Class Diagram* untuk klas *Pengurus*

Klas ini berguna untuk menginstansi klas Kesehatan, Kegiatan Kerja, dan Pegawai, yang akan memanfaatkan operasi-operasi yang dimilikinya

Pengurus

- *protected isi Str:*
- *protected Judul : Str*


```

- protected Data : Int
+ tampilstyle2 ( )
+ tampil_Menuatas_pengurus( )
+ tampilAtas( )
+ tampilBawah ( )
+ tambahData_Kesehatan ( )
+ melihatData_Kesehatan ( )
+ ubahData_Kesehatan ( )
+ hapusData_Kesehatan ( )
+ tambahData_Pegawai ( )
+ lihatData_Pegawai ( )
+ ubahData_Pegawai ( )
+ hapusData_Pegawai ( )
+ tambahData_PDKB ( )
+ lihatData_PDKB ( )
+ ubahData_PDKB ( )
+ hapusData_PDKB ( )
    
```

4.2.1.12 Penjelasan Class Diagram untuk klas Perancang

Klas ini berguna untuk merancang nama pegawai yang akan ditugaskan.

```

Perancang
- protected isi :Str
- protected Judul : Str
- protected Data : Int
+ tampilstyle2 ( )
+ tampil_Menuatas_pengurus( )
+ tampilAtas( )
+ tampilBawah ( )
    
```

```

+ tambahData_AnalisisPekerjaan( )
+ lihatData_AnalisisPekerjaan ( )
+ ubahData_AnalisisPekerjaan ( )
+ hapusData_AnalisisPekerjaan ( )
+ merancang_Kandidat ( )

```

4.2.1.13 Penjelasan *Class Diagram* untuk klas *Pengawas*

Klas ini berguna untuk menginstansi klas Data Laporan, Kegiatan kerja, dan Kandidat, yang akan memanfaatkan operasi-operasi yang dimilikinya

<i>Pengawas</i>
<ul style="list-style-type: none"> - <i>protected</i> isi :Str - <i>protected</i> Judul : Str - <i>protected</i> Data : Int
<ul style="list-style-type: none"> + tampilstyle2 () + tampil_Menuatas_pengurus() + tampilAtas() + tampilBawah () + memasukkan_PemberitahuanKerja () + mengubah_PemberitahuanKerja () + menghapus_PemberitahuanKerja () + melihat_PemberitahuanKerja () + menyetujui_Kandidat () + mengawasi_kegiatanKerja ()

4.2.1.4 Penjelasan *Class diagram* untuk Klas User

Klas User akan menginstansiasi Klas Account dan akan memanfaatkan operasi-operasi yang dimilikinya.

<i>User</i>
<ul style="list-style-type: none"> - <i>Protected user_id : Int</i> - <i>Protected user_level : Str</i> - <i>Protected password: Str</i>
<ul style="list-style-type: none"> + <i>login ()</i> + <i>logout ()</i>

4.2.1.15 Penjelasan *Class Diagram* untuk kelas *Administrator*

Klas ini berguna untuk menginstansiasi klas *Account* dan akan memanfaatkan operasi-operasi yang dimilikinya, serta memfasilitasi pencatatan user yang diizinkan untuk menggunakan sistem kedalam basis data.

<i>Administrator</i>
<ul style="list-style-type: none"> - <i>protected isi :Str</i> - <i>protected Judul : Str</i> - <i>protected Data : Int</i>
<ul style="list-style-type: none"> + <i>tampilstyle2 ()</i> + <i>tampil_Menuatas_pengurus()</i> + <i>tampilAtas()</i> + <i>tampilBawah ()</i> + <i>memasukkan _Account ()</i> + <i>mengubah _Account ()</i> + <i>menghapus Account ()</i> + <i>melihat_ Account ()</i>

4.2.2 Sequence Diagram

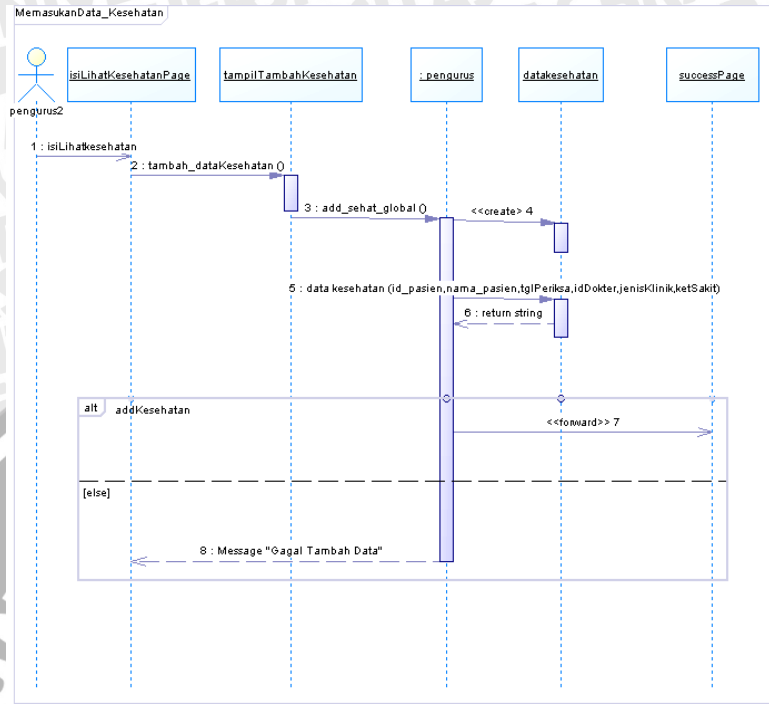
Hubungan antar kelas yang digambarkan pada diagram kelas termasuk dalam pemodelan statis, tanpa disertai dengan gambaran aliran proses atau data antarklas yang satu dengan kelas yang lain. *Sequence diagram* berguna untuk menampilkan aliran jalannya proses yang ditunjukkan dengan interaksi antar objek atau kelas, dan disusun berdasarkan urutan waktu. *Sequence diagram* dirancang dengan mengambil acuan pada *use case* serta operasi – operasi dalam kelas yang menjadi implementasi dari fungsionalitas yang digambarkan pada *use case* tersebut.

Sub bab 4.2.2 ini tidak memodelkan keseluruhan *sequence diagram* untuk tiap *use case*, akan tetapi diambil beberapa contoh *sequence diagram* untuk *use case* tertentu. Sub bab 4.2.2 ini memodelkan *sequence diagram* untuk *use case* berikut :

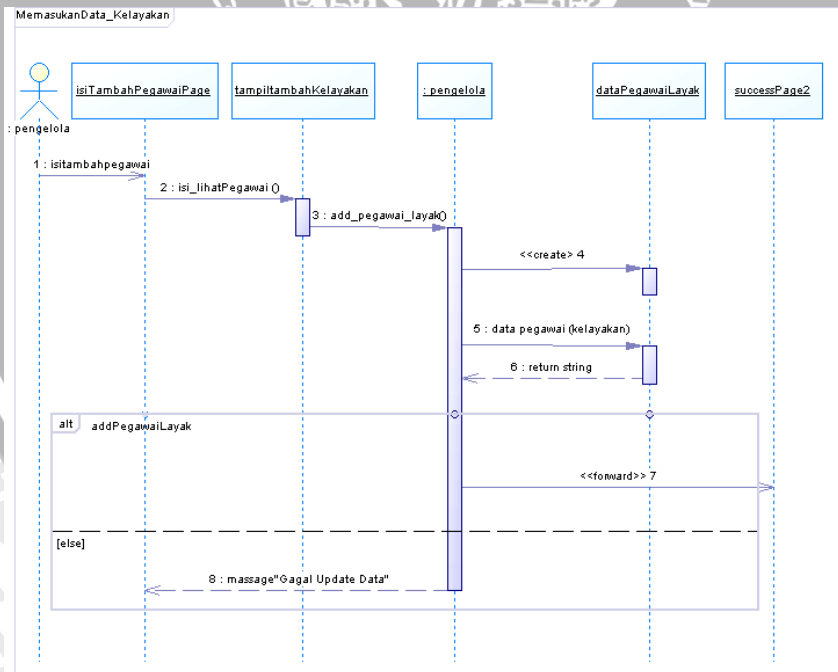
1. *Use case* Menyusun Data Pegawai
2. *Use case* Penanganan Kegiatan Kerja PDKB
3. *Use case* Menyusun Analisis Pekerjaan dan Pembagian Tugas
4. *Use case* Membuat surat tugas

4.2.2.1 Sequence Diagram untuk Use case menyusun Data Pegawai

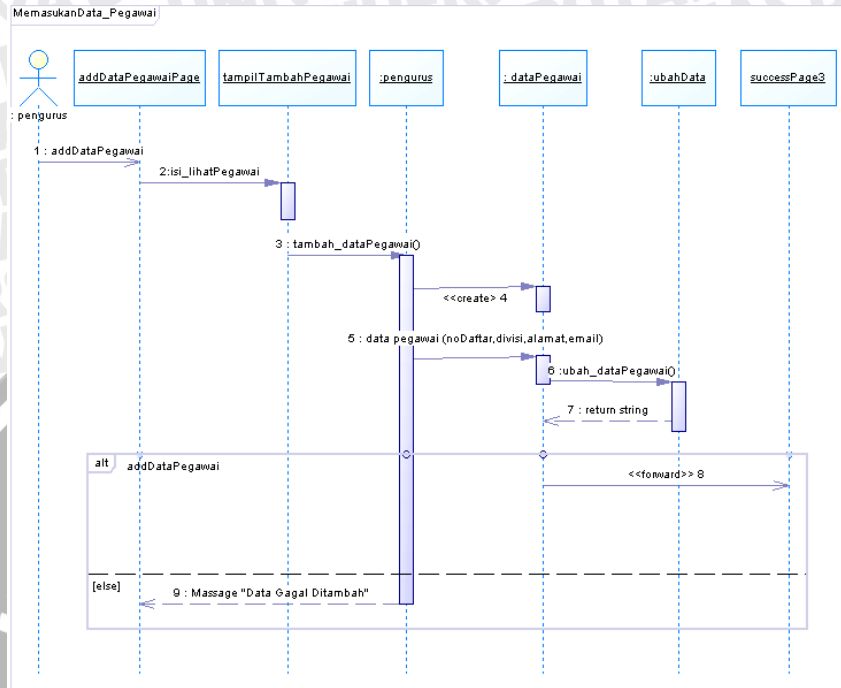
Gambar – gambar dibawah ini adalah *Sequence Diagram* untuk *Use case* Menyusun Data Pegawai. Klas yang menjawab kebutuhan dari *Use case* Menyusun Data Pegawai ini adalah klas pengurus dan pengelola, operasi yang terkait adalah menambahData_Kesehatan, menambahData_Kelayakan, dan menambahData_Pegawai.



Gambar 4.5 Sequence Diagram untuk Use case Menyusun Data Pegawai pada klas Pengurus dengan operasi menambahData_Kesehatan [Pengurus]



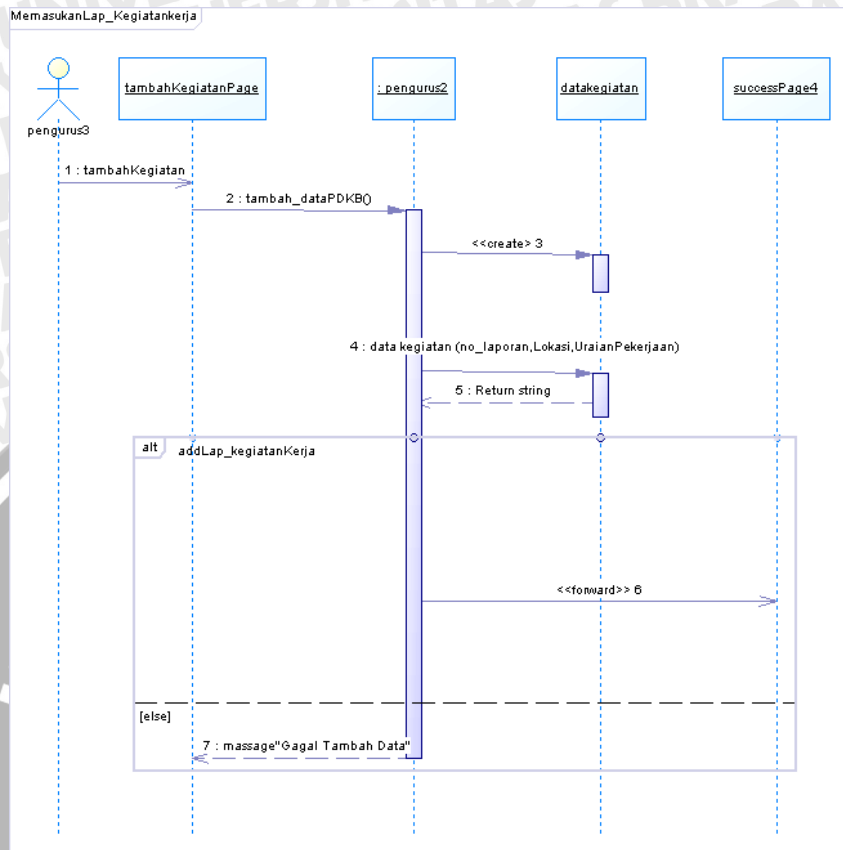
Gambar 4.6 Sequence Diagram untuk Use case Menyusun Data Pegawai pada klas Pengelola dengan operasi menambahData_Kelayakan [Pengelola]



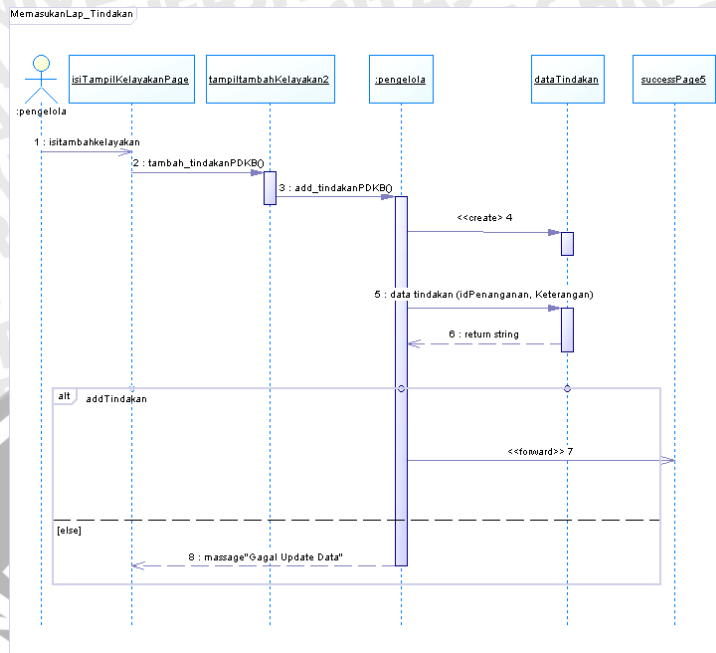
Gambar 4.7 *Sequence Diagram* untuk *Use case* Menyusun Data Pegawai pada kelas Pengurus dengan operasi menambahData_Pegawai [Pengurus]

4.2.2.2 *Sequence Diagram* untuk *Use case* Penanganan Kegiatan Kerja PDKB

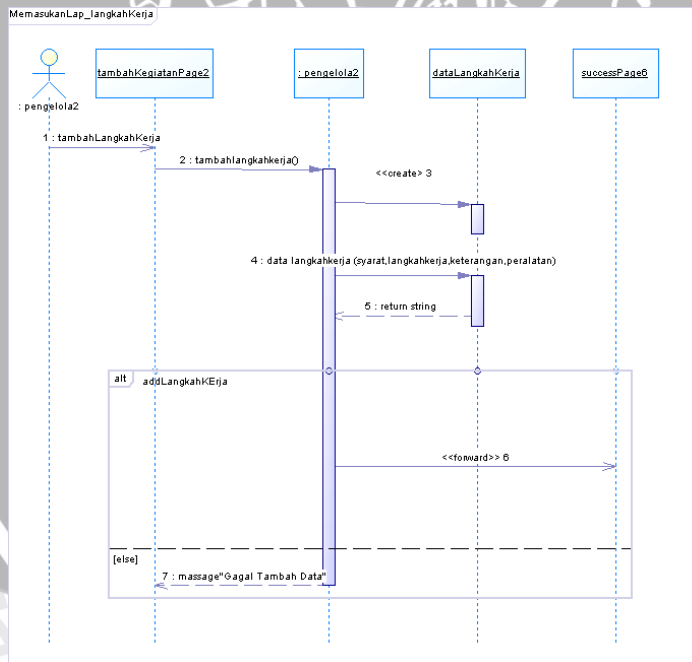
Gambar – gambar dibawah ini adalah *Sequence Diagram* untuk *Use case* Penanganan Kegiatan Kerja PDKB. Klas yang menjawab kebutuhan dari *Use case* Penanganan Kegiatan Kerja PDKB adalah klas pengelola dan Pengurus, operasi yang terkait adalah menambahLap_KegiatanKerja, menambahLap_Tindakan, menambahLap_LangkahKerja.



Gambar 4.8 Sequence Diagram untuk Use case Penanganan Kegiatan Kerja PDKB pada klas Pengurus dengan operasi menambahLap_KegiatanKerja. [Pengurus]



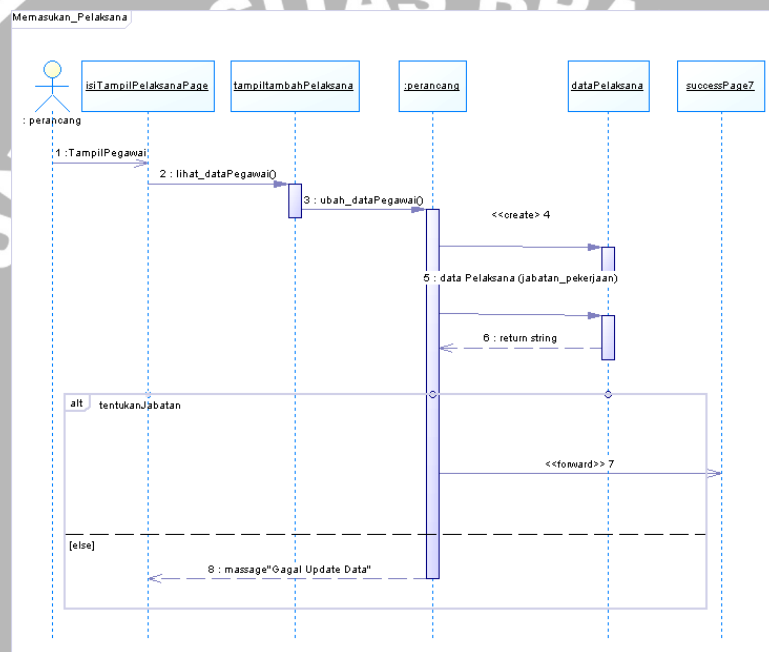
Gambar 4.9 Sequence Diagram untuk Use case Penanganan Kegiatan Kerja PDKB pada klas Pengelola dengan operasi menambahLap_Tindakan. [Pengelola]



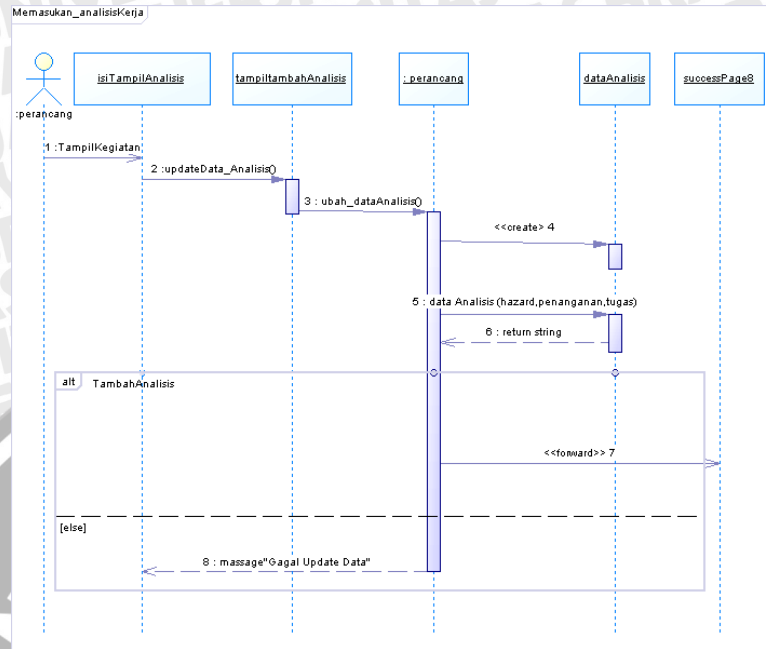
Gambar 4.10 Sequence Diagram untuk Use case Penanganan Kegiatan Kerja PDKB pada klas Pengelola dengan operasi menambahLap_LangkahKerja. [Pengelola]

4.2.2.3 Sequence Diagram untuk Use case Menyusun Analisis Pekerjaan dan Pembagian Tugas

Gambar – gambar dibawah ini adalah *Sequence Diagram* untuk *Use case* Menyusun Analisis Pekerjaan dan Pembagian Tugas. Klas yang menjawab kebutuhan dari *Use case* Menyusun Analisis Pekerjaan dan Pembagian Tugas ini adalah klas perancang, operasi yang terkait adalah menambah_Pelaksana, menambah_analisisKerja.



Gambar 4.11 *Sequence Diagram* untuk *Use case* Menyusun Analisis Pekerjaan dan Pembagian Tugas pada class Perancang dengan operasi menambah_Pelaksana. [Perancang]

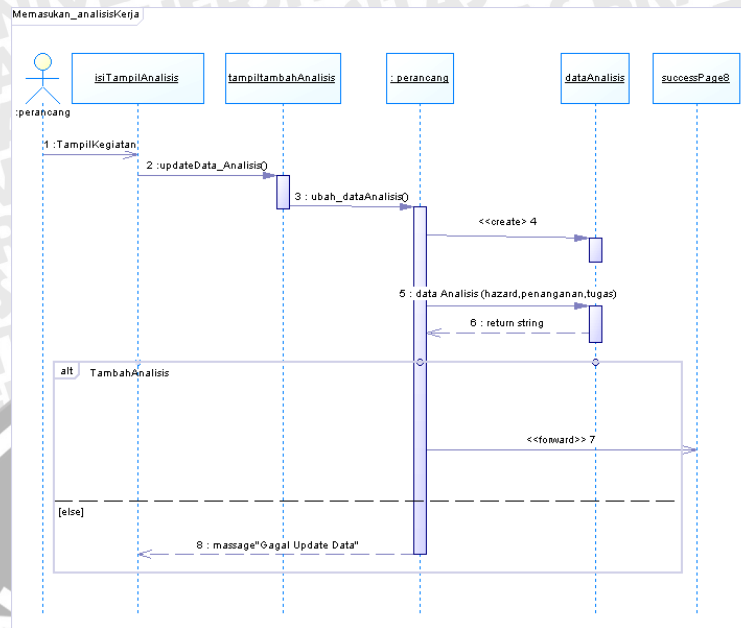


Gambar 4.12 *Sequence Diagram* untuk *Use case* Menyusun Analisis Pekerjaan dan Pembagian Tugas pada class Perancang dengan operasi menambah_Analisis.

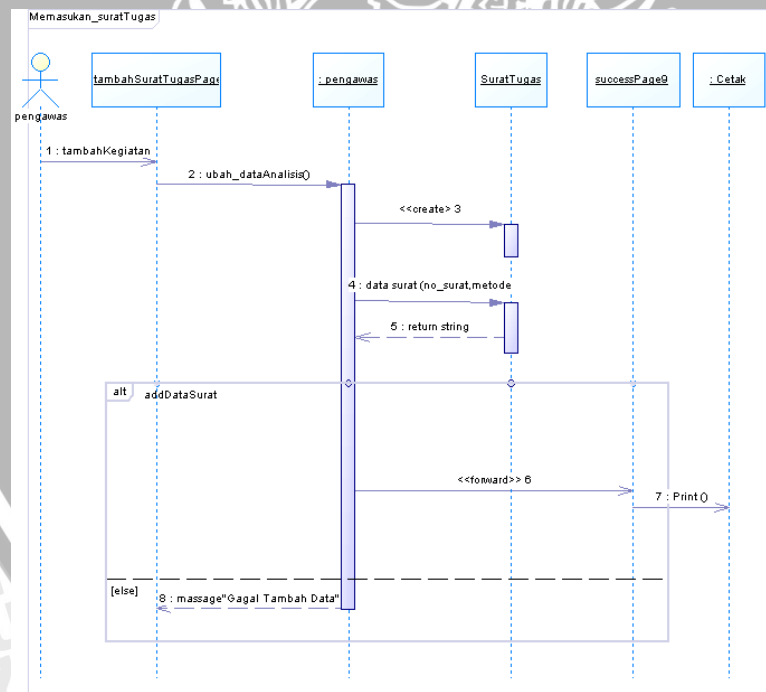
[Perancang]

4.2.2.4 *Sequence Diagram* untuk *Use case* membuat Surat Tugas

Gambar – gambar dibawah ini adalah *Sequence Diagram* untuk *Use case* membuat Surat Tugas. Klas yang menjawab kebutuhan dari *Use case* membuat Surat Tugas ini adalah klas perancang dan pengawas, operasi yang terkait adalah menambah_Analisis dan menambah_SuratTugas.



Gambar 4.13 Sequence Diagram untuk Use case Membuat Surat Tugas pada class Perancang dengan operasi menambah_Analisis. [Perancang]



Gambar 4.14 Sequence Diagram untuk Use case Membuat Surat Tugas pada class Pengawas dengan operasi menambah_SuratTugas. [Pengawas]

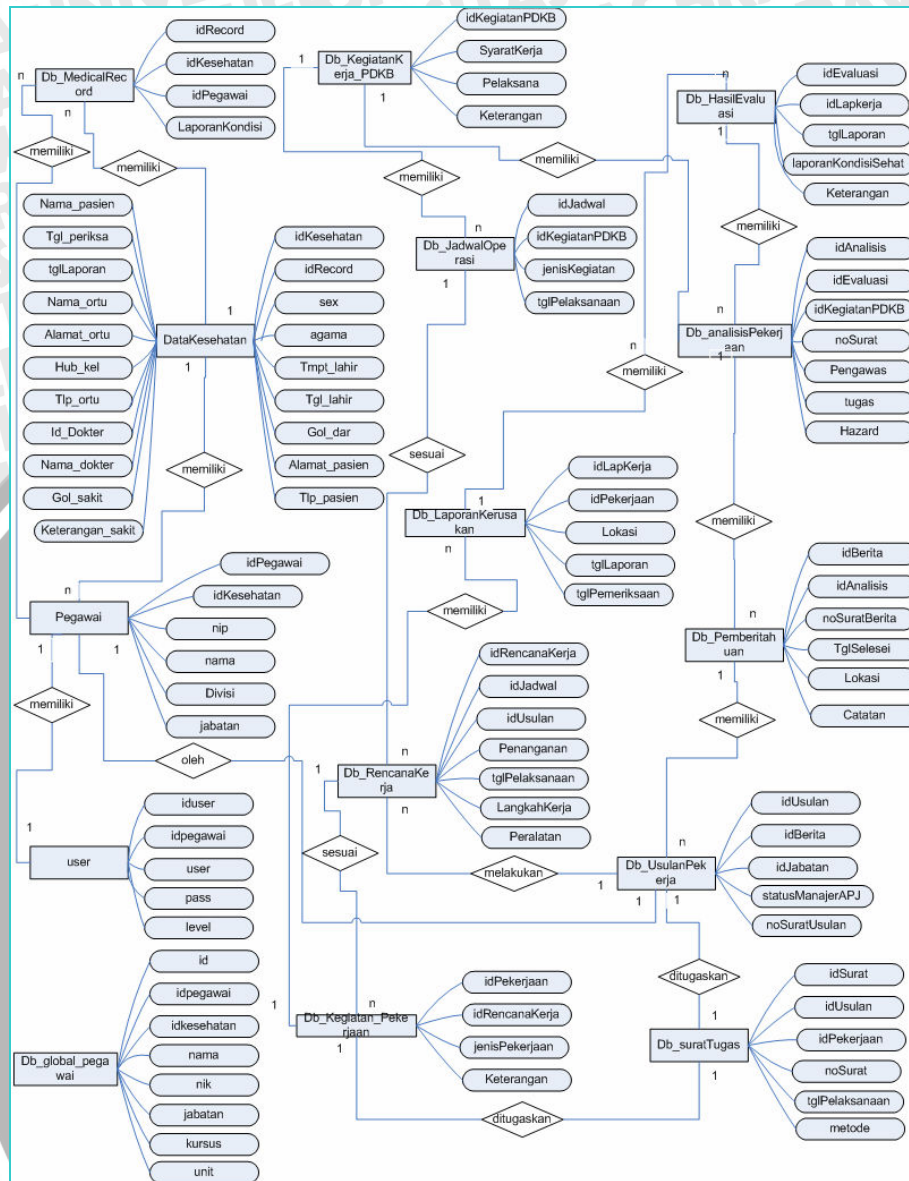
4.3 Basis Data

Sistem Informasi yang akan dikembangkan ini memerlukan suatu basis data yang berfungsi untuk menampung data-data yang terkait. Perancangan basis data dilakukan agar basis data dapat efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan, dan mudah dalam manipulasi data. Perancangan basis data dapat dilakukan dengan menggambarkan diagram hubungan antar entitas atau biasa dikenal dengan *Entity –Relationship Diagram* (ER-Diagram), dan normalisasi. Normalisasi merupakan teknik dalam mengelompokkan atribut dari suatu relasi sehingga terbentuk struktur yang baik. Dalam ER-Diagram akan tampak relasi antar entitas yang saling terkait.

4.3.1 Diagram Entitas Relasional

Diagram Entitas Relasional ini digunakan untuk menggambarkan hubungan entitas satu dengan lainnya dengan memperlihatkan hubungan antar key untuk berelasi antar tabel. Diagram Entitas Relasional dari aplikasi Sistem Informasi Manajemen Kesehatan Keselamatan dan Keamanan Kerja digambarkan pada gambar 4.14.

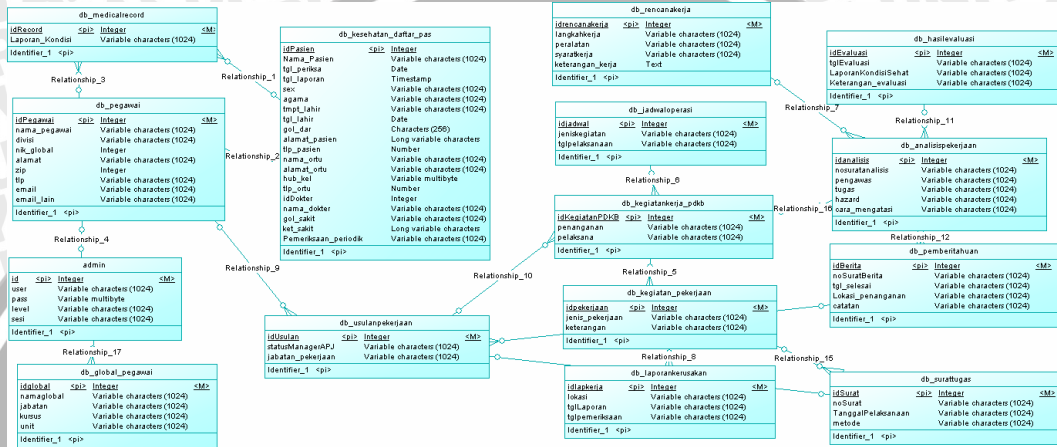
Diagram Relasi Antar Tabel disajikan untuk menampilkan relasi tabel-tabel yang digunakan dalam sistem ini. Diagram Relasi Antar Tabel digambarkan melalui *Conceptual Data Model* pada gambar 4.15 dan *Physical Data Model* pada gambar 4.16



Gambar 4.15 Diagram Entitas Relasional dari Sistem Informasi Manajemen Kesehatan Keselamatan dan Keamanan Kerja [Perancangan]

4.3.2 Normalisasi Data

Normalisasi data adalah bagian penting dalam perancangan basis data untuk menjamin atribut-atribut dalam tabel hanya bergantung pada *primary key*. Proses normalisasi data membuat basis data lebih mudah untuk dipelihara dan memiliki laporan yang konsisten. Rancangan normalisasi data ditunjukkan pada Gambar 4.18.



Gambar 4.18 Normalisasi data

Sumber: Perancangan

4.3.2.1 Normalisasi Pertama

Secara umum tujuan dilakukannya normalisasi pertama adalah sebagai berikut :

- Mendefinisikan atribut yang menjadi kunci bagi tabel tersebut
- Tidak adanya perulangan kolom dalam suatu tabel
- Semua atribut selain kunci, bergantung pada atribut kunci

Dari hasil perancangan pada tahap normalisasi pertama atribut sebagai kunci telah didefinisikan, tidak terdapat perulangan kolom, semua atribut selain kunci bergantung pada kunci .

4.3.2.2 Normalisasi Kedua

Secara umum tujuan dilakukannya normalisasi kedua adalah sebagai berikut :

- Memastikan bentuk normal pertama telah terpenuhi

- Memastikan tidak ada ketergantungan parsial

Dari hasil perancangan pada tahap normalisasi kedua diperoleh bentuk normal pertama telah dipenuhi, dan tidak terjadi ketergantungan parsial.

4.3.2.3 Normalisasi Ketiga

Secara umum tujuan dilakukannya normalisasi ketiga adalah sebagai berikut :

- Memastikan bentuk normal kedua telah terpenuhi
- Memastikan tidak ada ketergantungan transitif

Dari hasil perancangan pada tahap normalisasi ketiga diperoleh bentuk normal kedua telah dipenuhi, tidak terjadi ketergantungan transitif.

Dari hasil perancangan Diagram ER diperoleh beberapa tabel-tabel.

Tabel : db_MedicalRecord		
idRecord	Integer	<i>Primary Key</i>
idPasien	Integer	<i>Foreign Key</i>
Idpegawai	Integer	<i>Foreign Key</i>
LaporanKondisi	String	-

[sumber : Perancangan]

Table : db_Kesehatan_daftar_pas		
idpasien	Integer	<i>Primary Key</i>
Nama_pasien	String	-
tglPeriksa	String	-
tglLaporan	String	-
sex	String	-
agama	String	-
tmpt_lahir	String	-
tgl_lahir	Date	-
gol_dar	String	-
alamat_pasien	String	-
tlp_pasien	String	-
nama_ortu	String	-

alamat_oru	String	-
hub_kel	String	-
tlp_oru`	String	-
id_dokter	Integer	-
nama_dokter`	String	-
gol_sakit	String	-
Keterangan_sakit	String	-

[sumber : Perancangan]

Tabel : db_Pegawai

idPegawai	Integer	<i>Primary Key</i>
idpasien	Integer	<i>Foreign Key</i>
id	Integer	<i>Foreign Key</i>
nama_pegawai	String	-
divisi	String	-
nik	String	-
Alamat	String	-
Zip	String	-
tlp	String	-
email	String	-

[sumber : Perancangan]

Tabel : admin

id	Integer	<i>Primary Key</i>
idPegawai	Integer	<i>Foreign Key</i>
user	String	-
pass	String	-
level	String	-
sesi	String	-

[sumber : Perancangan]

Tabel : db_kegiatankerja_pdkb

idKegiatanPDKB	Integer	<i>Primary Key</i>
idjadwal	Integer	<i>Foreign Key</i>
idUsulan	Integer	<i>Foreign Key</i>
Penanganan	String	-
Pelaksana	String	-

[sumber : Perancangan]

Tabel : db_jadwalOperasi

idJadwal	Integer	<i>Primary Key</i>
jeniskegiatan	String	-
tglPelaksanaan	Date	-

[sumber : Perancangan]

Tabel : db_RencanaKerja

idRencanaKerja	Integer	<i>Primary Key</i>
LangkahKerja	String	-
Peralatan	String	-
SyaratKerja	String	-
keterangan_kerja	String	-

[sumber : Perancangan]

Tabel : db_KegiatanPekerjaan

idPekerjaan	Integer	<i>Primary Key</i>
idkegiatanpdkb	Integer	<i>Foreign Key</i>
Jenis_pekerjaan	String	-
keterangan	String	-

[sumber : Perancangan]

Tabel : db_LaporanKerusakan

idlapKerja	Integer	<i>Primary Key</i>
idPekerjaan	Integer	<i>Foreign Key</i>
Lokasi	String	-
tglLaporan	date	-
tglPemeriksaan	date	-

[sumber : Perancangan]

Tabel : db_HasilEvaluasi

idEvaluasi	Integer	<i>Primary Key</i>
tglEvaluasi	Date	-
LaporanKondisiSehat	String	-
keterangan_evaluasi	String	-

[sumber : Perancangan]

Tabel : db_analisisPekerjaan

idAnalisis	Integer	<i>Primary Key</i>
idrencanakanerja	Integer	<i>Foreign Key</i>
idkegiatanpdkb	Integer	<i>Foreign Key</i>
idEvaluasi	Integer	<i>Foreign Key</i>
nosuratanalisis	String	-
pengawas	String	-
Tugas	String	-
Hazard	String	-
Cara_mengatasi	String	-

[sumber : Perancangan]

Tabel : db_Pegawai

id	Integer	<i>Primary Key</i>
idpasien	Integer	<i>Foreign Key</i>
idPegawai	Integer	<i>Foreign Key</i>
nama	String	-

nik	String	-
jabatan	String	-
kursus	String	-
unit	String	-

[sumber : Perancangan]

Tabel : db_pemberitahuan		
idBerita	Integer	<i>Primary Key</i>
idAnalisis	Integer	<i>Foreign Key</i>
noSuratBerita	String	-
tgl_selesai	Date	-
Lokasi_Penanganan	String	-
catatan	String	-

[sumber : Perancangan]

Tabel : db_UsulanPekerja		
idUsulan	Integer	<i>Primary Key</i>
idPegawai	Integer	<i>Foreign Key</i>
idBerita	Integer	<i>Foreign Key</i>
idsurat	Integer	<i>Foreign Key</i>
statusManagerAPJ	Integer	-
jabatan_pekerjaan	String	-

[sumber : Perancangan]

Tabel : db_SuratTugas		
idsurat	Integer	<i>Primary Key</i>
idUsulan	Integer	<i>Foreign Key</i>
idpekerjaan	Integer	<i>Foreign Key</i>
noSurat	String	-
TanggalPelaksanaan	Date	-
metode	String	-

[sumber : Perancangan]

BAB V

IMPLEMENTASI PERANGKAT LUNAK

Bab ini membahas tentang implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak sebelumnya. Pembahasan terdiri dari penjelasan implementasi perancangan basis data, dan implementasi perancangan perangkat lunak Sistem Informasi Manajemen Kesehatan, Keselamatan, dan Keamanan kerja.

5.1 Implementasi Perancangan Basis Data

Aplikasi ini dirancang untuk dapat terhubung ke *server* basis data MySQL. Implementasi perancangan basis data skripsi dilakukan sesuai dengan *Entity Relationship Diagram*. Implementasi perancangan basis data menggunakan *query* SQL. *Query* SQL digunakan untuk mengimplementasikan rancangan basis data ke dalam sistem basis data MySQL.

Query SQL yang digunakan dalam membentuk basis data `simk3` ditunjukkan pada Gambar 5.1.

```
CREATE DATABASE simk3;
```

Gambar 5.1 *Query* untuk membuat basis data `simk3`
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_pegawai ditunjukkan pada Gambar 5.2.

```
CREATE TABLE db_pegawai (
  idpegawai int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idpasien int(10) UNSIGNED NOT NULL,
  id int(10) UNSIGNED NOT NULL,
  nama_pegawai varchar(100) NOT NULL,
  divisi varchar(20) NOT NULL,
  nik varchar (20) NOT NULL,
  alamat varchar(255) NOT NULL,
  zip varchar(10) NOT NULL,
  tlp varchar(10) NOT NULL,
  email varchar(50) NOT NULL,
  email_lain varchar(50) NOT NULL,
  PRIMARY KEY(idpegawai),
  FOREIGN KEY(idpasien) REFERENCES db_datakesehatan(idpasien),
  FOREIGN KEY(id) REFERENCES admin(id)
) ENGINE=MyISAM;
```

Gambar 5.2 *Query* untuk membuat tabel db_pegawai
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_kegiatankerja_pdkb ditunjukkan pada Gambar 5.3.

```
CREATE TABLE db_kegiatankerja_pdkb (
  idkegiatanpdkb int(10) NOT NULL,
  idjadwal int(10) UNSIGNED NOT NULL,
  idusulan int(10) UNSIGNED NOT NULL,
  penanganan varchar(20) NOT NULL,
  pelaksana varchar(20) NOT NULL,
  PRIMARY KEY(idkegiatanpdkb)
  FOREIGN KEY(idjadwal) REFERENCES db_jadwaloperasi(idjadwal),
  FOREIGN KEY(idusulan) REFERENCES db_usulanpekerjaan(idusulan)
) ENGINE=MyISAM;
```

Gambar 5.3 *Query* untuk membuat tabel db_kegiatankerja_pdkb
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_kesehatan_daftar_pas ditunjukkan pada Gambar 5.4.


```
CREATE TABLE db_kesehatan_daftar_pas (
  idpasien int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  nama_pasien varchar(20) NOT NULL,
  tgl_periksa varchar(50) NOT NULL,
  sex varchar(10) NOT NULL,
  agama varchar(10) NOT NULL,
  tmpt_lahir varchar(30) NOT NULL,
  tgl_lahir varchar(50) NOT NULL,
  gol_dar varchar(6) NOT NULL,
  alamat_pasien varchar(255) NOT NULL,
  tlp_pasien varchar(16) NOT NULL,
  nama_ortu varchar(20) NOT NULL,
  alamat_ortu varchar(255) NOT NULL,
  hub_kel varchar(50) NOT NULL,
  tlp_ortu varchar(16) NOT NULL,
  id_dokter varchar(6) NOT NULL,
  nama_dokter varchar(20) NOT NULL,
  gol_sakit varchar(50) NOT NULL,
  Keterangan_sakit text NOT NULL,
  PRIMARY KEY(idkesehatan)
) ENGINE=MyISAM;
```

Gambar 5.4 Query untuk membuat tabel db_kesehatan_daftar_pas

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_medicalRecord ditunjukkan pada Gambar 5.5.

```
CREATE TABLE db_medicalRecord (
  idrecord int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idpasien int(10) UNSIGNED NOT NULL,
  idpegawai int(10) UNSIGNED NOT NULL,
  laporankondisi VARCHAR(255) NOT NULL,
  PRIMARY KEY(idrecord),
  FOREIGN KEY(idpasien) REFERENCES db_kesehatan_daftar_pas(idpasien),
  FOREIGN KEY(idpegawai) REFERENCES db_pegawai(idpegawai)
) ENGINE=MyISAM;
```

Gambar 5.5 Query untuk membuat tabel db_medicalRecord

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_laporankerusakan ditunjukkan pada Gambar 5.6.

```
CREATE TABLE db_laporankerusakan (
  idlapkerja int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idpekerjaan int(10) UNSIGNED NOT NULL,
  lokasi VARCHAR (100) NOT NULL,
  tglLaporan VARCHAR (10) NOT NULL,
  tglpemeriksaan VARCHAR (10) NOT NULL,
  PRIMARY KEY(idlapkerja),
  FOREIGN KEY(idpekerjaan) REFERENCES db_kegiatan_pekerjaan(idpekerjaan)
) ENGINE=MyISAM;
```

Gambar 5.6 Query untuk membuat tabel db_laporanKerusakan

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_analisispekerjaan ditunjukkan pada Gambar 5.7.

```
CREATE TABLE db_analisispekerjaan (
  idanalisis int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idrencanakanerja int(10) UNSIGNED NOT NULL,
  idevaluasi int(10) UNSIGNED NOT NULL,
  idkegiatanpdkb int(10) UNSIGNED NOT NULL,
  nosurat VARCHAR (100) NOT NULL,
  pengawas VARCHAR (20) NOT NULL,
  tugas VARCHAR (255) NOT NULL,
  hazard VARCHAR (255) NOT NULL,
  PRIMARY KEY(idanalisis),
  FOREIGN KEY(idevaluasi) REFERENCES db_hasilevaluasi(idevaluasi),
  FOREIGN KEY(idrencanakanerja) REFERENCES db_rencanakanerja(idrencanakanerja),
  FOREIGN KEY(idkegiatanpdkb) REFERENCES db_kegiatanpekerjaan(idkegiatanpdkb)
) ENGINE=MyISAM;
```

Gambar 5.7 *Query* untuk membuat tabel db_analisispekerjaan

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_jadwaloperasi ditunjukkan pada Gambar 5.8.

```
CREATE TABLE db_jadwaloperasi (
  idjadwal int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  jeniskegiatan VARCHAR (100) NOT NULL,
  tglpelaksanaan VARCHAR (20) NOT NULL,
  PRIMARY KEY(idjadwal),
) ENGINE=MyISAM;
```

Gambar 5.8 *Query* untuk membuat tabel db_jadwaloperasi

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_kegiatan_pekerjaan ditunjukkan pada Gambar 5.9.

```
CREATE TABLE db_kegiatan_pekerjaan (
  idpekerjaan int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idkegiatanpdkb int(10) UNSIGNED NOT NULL,
  jenis_pekerjaan VARCHAR (100) NOT NULL,
  keterangan VARCHAR (255) NOT NULL,
  PRIMARY KEY(idpekerjaan),
  FOREIGN KEY(idkegiatanpdkb) REFERENCES db_kegiatanpekerjaan(idkegiatanpdkb)
) ENGINE=MyISAM;
```

Gambar 5.9 *Query* untuk membuat tabel db_kegiatan_pekerjaan

Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_usulanpekerjaan ditunjukkan pada Gambar 5.10.

```
CREATE TABLE db_usulanpekerjaan (
  idusulan int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idberita int(10) UNSIGNED NOT NULL,
  idpegawai int(10) UNSIGNED NOT NULL,
  statusmanajerapj VARCHAR (100) NOT NULL,
  jabatan_pekerjaan VARCHAR (100) NOT NULL,
  PRIMARY KEY(idusulan),
  FOREIGN KEY(idberita) REFERENCES db_pemberitahuan(idberita),
  FOREIGN KEY(idpegawai) REFERENCES db_pegawai(idpegawai)
) ENGINE=MyISAM;
```

Gambar 5.10 *Query untuk membuat tabel db_usulanpekerjaan*
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_pemberitahuan ditunjukkan pada Gambar 5.11.

```
CREATE TABLE db_pemberitahuan (
  idberita int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idanalisis int(10) UNSIGNED NOT NULL,
  nosuratberita VARCHAR (100) NOT NULL,
  tglselesai VARCHAR (10) NOT NULL,
  lokasi_penanganan VARCHAR (100) NOT NULL,
  catatan VARCHAR (255) NOT NULL,
  PRIMARY KEY(idusulan),
  FOREIGN KEY(idanalisis) REFERENCES db_analisispekerjaan(idanalisis)
) ENGINE=MyISAM;
```

Gambar 5.11 *Query untuk membuat tabel db_pemberitahuan*
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_global_pegawai ditunjukkan pada Gambar 5.12.

```
CREATE TABLE db_pegawai (
  id int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idpasien int(10) UNSIGNED NOT NULL,
  idpegawai int(10) UNSIGNED NOT NULL,
  nama varchar(100) NOT NULL,
  nik varchar (20) NOT NULL,
  jabatan varchar(10) NOT NULL,
  kursus varchar(50) NOT NULL,
  unit varchar(20) NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY(idpasien) REFERENCES db_datakesehatan(idpasien),
  FOREIGN KEY(idpegawai) REFERENCES db_pegawai(idpegawai)
) ENGINE=MyISAM;
```

Gambar 5.12 *Query untuk membuat tabel db_global_pegawai*
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_rencanakerja ditunjukkan pada Gambar 5.13.

```
CREATE TABLE db_rencanakerja (
  idrencanakerja int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  langkahkerja VARCHAR(100) NOT NULL,
  peralatan VARCHAR(100) NOT NULL,
  Syaratkerja VARCHAR(100) NOT NULL,
  Keterangan_kerja VARCHAR(100) NOT NULL,
  PRIMARY KEY(idrencanakerja),
) ENGINE=MyISAM;
```

Gambar 5.13 *Query untuk membuat tabel db_rencanakerja*
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_surattugas ditunjukkan pada Gambar 5.14.

```
CREATE TABLE db_surattugas (
  idsurat int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  idpekerjaan int(10) UNSIGNED NOT NULL,
  idusulan int(10) UNSIGNED NOT NULL,
  nosurat VARCHAR(100) NOT NULL,
  Tglpelaksanaan VARCHAR(100) NOT NULL,
  metode VARCHAR(255) NOT NULL,
  PRIMARY KEY(idsurat),
  FOREIGN KEY(idpekerjaan) REFERENCES db_kegiatan_pekerjaan(idpekerjaan),
  FOREIGN KEY(idusulan) REFERENCES db_usulanpekerja(idusulan)
) ENGINE=MyISAM;
```

Gambar 5.14 *Query untuk membuat tabel db_surattugas*
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel db_hasilevaluasi ditunjukkan pada Gambar 5.15.

```
CREATE TABLE db_hasilevaluasi (
  idevaluasi int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  tglevaluasi VARCHAR(10) NOT NULL,
  laporanKondisisehat VARCHAR(255) NOT NULL,
  keturanganevaluasi TEXT NOT NULL,
  PRIMARY KEY(idevaluasi),
) ENGINE=MyISAM;
```

Gambar 5.15 *Query untuk membuat tabel db_hasilevaluasi*
Sumber: Implementasi

Query SQL yang digunakan dalam membentuk tabel admin ditunjukkan pada Gambar 5.16.

```
CREATE TABLE admin(
  id INT(15) UNSIGNED NOT NULL AUTO_INCREMENT,
  idpegawai INT(10) UNSIGNED NOT NULL,
  user varchar(10) NOT NULL,
  pass varchar(100) NOT NULL,
  level varchar(30) NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY(idpegawai) REFERENCES db_pegawai(idpegawai)
) ENGINE=MyISAM;
```

Gambar 5.16 Query untuk membuat table user
Sumber: Implementasi

5.2 Implementasi Perangkat Lunak

Pada Implementasi Perangkat Lunak aplikasi dilakukan dalam beberapa tahap yaitu Implementasi Pendukung Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja, Implementasi Penyedia Data Pendukung Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja, Implementasi Pengelolaan Informasi Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja

5.2.1 Implementasi Pendukung Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja

Implementasi Pendukung Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja meliputi beberapa kebutuhan fungsional yaitu : Login, Logout, Menambah Account, Melihat Account, Mengubah Account, Menghapus Account.

5.2.1.1 Implementasi Kebutuhan Fungsional Login

Implementasi dilakukan dengan menyediakan dua buah *text field* untuk masing-masing username dan password, dan satu tombol login. Pengguna akan login sesuai peran yang dimiliki, jika pengguna berhasil melakukan login maka user akan memiliki hak tertentu sesuai peran yang dimiliki. Dan selanjutnya pengguna akan menuju halaman utama terkait dengan pengguna yang melakukan login.

Implementasi kebutuhan Fungsional *login* dapat dilihat pada gambar 5.17 dibawah ini.



Gambar 5.17 Halaman untuk melakukan *Login*
Sumber: Implementasi

Algoritma pada tombol login diatas ditunjukkan secara sederhana melalui *pseudocode* berikut ini :

```

Login
$login = Buka Koneksi dengan basis data
    if(koneksi dengan basis data berhasil) {echo "berhasil koneksi
database";} else{echo "gagal";}
$login->pilih database();
    if(berhasil memilih tabel admin dalam basis data){
        while($permintaan_kolom = mysql_fetch_row(data dalam tabel))
if($permintaan_kolom[1] == username && $permintaan_kolom[2] == password){
    if($permintaan_kolom[3]=='admin'){
        echo "memunculkan halaman admin'>";
    }elseif($permintaan_kolom[3]=='pengurus'){
        echo "memunculkan halaman pengurus'>";
    }elseif($permintaan_kolom[3]=='pengelola'){
        echo "memunculkan halaman pengelola'>";
    }elseif($permintaan_kolom[3]=='perancang'){
        echo "memunculkan halaman Perancang'>";
    }elseif($permintaan_kolom[3]=='pengawas'){
        echo "memunculkan halaman Pengawas'>";
    }elseif($permintaan_kolom[3]=='pegawai'){
        echo "memunculkan halaman Pegawai'>";
    }else{echo "gagal!!!";} }}}}
End login
    
```

Gambar 5.18 *Pseudocode* pada tombol login untuk Kebutuhan Fungsional *login*
Sumber: Implementasi

5.2.1.2 Implementasi Kebutuhan Fungsional Logout

Implementasi Kebutuhan Fungsional *logout* dilakukan dengan menyediakan satu buah tombol yang hanya muncul apabila telah melakukan login. Implementasi kebutuhan

Implementasi kebutuhan Fungsional *logout* dapat dilihat pada gambar 5.19 dibawah ini.



Anda Telah berhasil logout
isi username dan password dibawah ini untuk Log in kembali

Informasi

Silahkan masukan User dan Password Anda, Bila belum memiliki Account Klik disini.

Selamat Datang

Login details

Username
username

Password
password

login

Gambar 5.19 Halaman setelah melakukan untuk melakukan *Logout*
Sumber: Implementasi

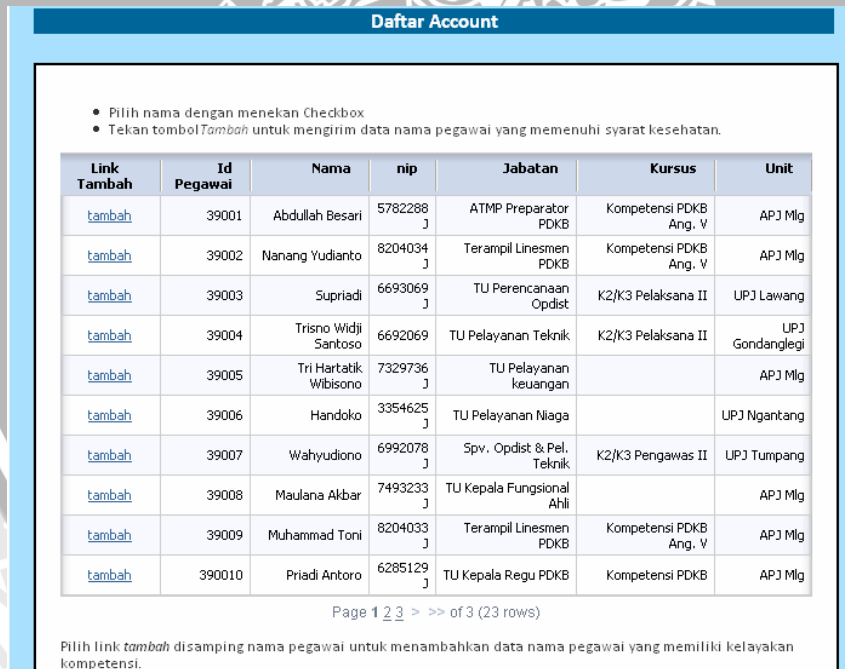
5.2.1.3 Implementasi Kebutuhan Fungsional Menambah Account

Untuk mengakses halaman ini pengguna masuk sebagai Admin. Kemudian pengguna akan menemukan beberapa sub menu, dimana salah satunya adalah sub menu Menambah Account.



Gambar 5.20 Submenu Menambah Account
Sumber: Implementasi

Setelah sub menu *tambah account* dipilih, maka akan muncul halaman seperti dibawah ini.



Gambar 5.21 Halaman yang menampilkan nama yang akan dijadikan account
Sumber: Implementasi

Pada halaman ini akan muncul daftar nama pegawai. Disamping masing-masing nama tersebut terdapat link *tambah*. Jika hendak menambahkan salah satu baris nama, maka kita tekan link *tambah* tersebut.

Apakah Anda Yakin akan menambahkan nama **Abdullah Besari** ke dalam Sistem Informasi Kesehatan Keselamatan Kerja ?

Personal information

ID	✓	<input type="text" value="11"/>
Username	*	<input type="text"/>
Password	*	<input type="text"/>
Nama	✓	<input type="text" value="Abdullah Besari"/>
NIP	✓	<input type="text" value="5782288 J"/>
Jabatan	*	<input type="text" value="ATMP Preparator PDKB"/>
Kursus	*	<input type="text" value="Kompetensi PDKB Ang. V"/>
Id Pegawai	✓	<input type="text" value="39001"/>
unit		<input type="text" value="APJ Mlg"/>
Level	*	<input type="text"/>

Pilih Link [Batal](#) berikut untuk kembali kehalaman daftar Account

Gambar 5.22 Halaman yang menampilkan form untuk menambah account
Sumber: Implementasi

Setelah kita menekan link *tambah*, maka akan tampil halaman seperti pada gambar 5.22 diatas. Nama pegawai akan muncul, diikuti informasi NIP, Jabatan, Kursus, ID Pegawai, dan unit. dan tiga buah *textbox* untuk Username, Password, dan Level. Beserta sebuah tombol simpan, untuk menyimpan data.

Algoritma pada tombol diatas, secara sederhana dapat ditunjukkan dalam *pseudocode* berikut ini :

```
add_account
$this->definisikan dengan data2[pilihan kolom] = $_POST[nama field input];
$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

$db3 = new pilih class database();
$db3->Buka Koneksi dengan basis data();
$db3->pilih databse();
$db3->query = "masukan data kedalam databse db_pegawai;";
$db3->jalankan query();

$db4 = new pilih class database();
$db4->Buka Koneksi dengan basis data();
$db4->pilih databse();
$db4->query = "masukan data2 kedalam databse;";
$db4->jalankan query();

if($db3->query && $db4->query){
    echo "<Tampilkan halaman index.php>";
}else{
    echo "GAGAL";
}

End add_account
```

Gambar 5.23 Pseudocode pada tombol login untuk Kebutuhan Fungsional Menambah Account

Sumber: Implementasi

5.2.1.4 Implementasi Kebutuhan Fungsional Melihat Account

Pengguna harus terlebih dahulu masuk sebagai admin. Selanjutnya Pengguna memilih sub menu *Melihat Account*.

Setelah memilih sub menu *Melihat Account* maka akan muncul gambar seperti gambar 5.24 dibawah ini.

Daftar Account							
<ul style="list-style-type: none"> Pilih nama dengan menekan Checkbox Tekan tombol <i>Tambah</i> untuk mengirim data nama pegawai yang memenuhi syarat kesehatan. 							
<input type="checkbox"/> Pilih Semua Baris <input checked="" type="checkbox"/> Pilih Semua Baris dalam halaman ini							
No	Id Pegawai	NIP	Nama	Level	Kompetensi	Unit	Edit - Hapus
1	390023	3339995	Hafid Ali	admin	own	own	Edit - Hapus
2	39007	699207E	Wahyudiono	pengawas	K2/K3 Pengawas II	UPJ Tumpang	Edit - Hapus
3	39002	8204034	Nanang Yudianto	pengurus	Kompetensi PDKB Ang. V	APJ Mlg	Edit - Hapus
4	390022	6383337	Sumarsono Tasmun	pelaksana	K2/K3 Pelaksana II	UPJ Lawang	Edit - Hapus
5	390010	6285125	Priadi Antoro	pengelola	Kompetensi PDKB	APJ Mlg	Edit - Hapus
6	390016	547909E	Setyo Wahono	perancang	Kompetensi PDKB Ang. V	APJ Mlg	Edit - Hapus
7	39003	6693065	Supriadi	pelaksana	K2/K3 Pelaksana II	UPJ Lawang	Edit - Hapus
8	39004	6692065	Trisno Widji Santoso	pelaksana	K2/K3 Pelaksana II	UPJ Gondanglegi	Edit - Hapus
10	390012	689107E	Wahyu Nagano	pelaksana	K2/K3 Pengawas II	UPJ Kepanjen	Edit - Hapus
Page 1 of 1 (9 rows)							
Pilih link disamping nama pegawai, <i>edit</i> untuk mengubah data account dan link <i>hapus</i> untuk menghapus account yang telah dipilih							

Gambar 5.24 Halaman Melihat Account
Sumber: Implementasi

Akan ditampilkan nama-nama yang telah memiliki account beserta informasi berkait dengan Level, Kompetensi, dan Unit Pegawai. Diatas form ditampilkan link *tambah* untuk menambah account baru.

Pada baris nama yang telah memiliki account akan muncul link *Edit - Hapus* yang berfungsi untuk mengubah data account dan menghapus data account yang akan dijelaskan lebih detail dalam Implementasi Kebutuhan Fungsional Mengubah Account dan Implementasi Kebutuhan Fungsional Menghapus Account

5.2.1.5 Implementasi Kebutuhan Fungsional Mengubah Account

Pengguna harus terlebih dahulu login sebagai admin. Untuk mengakses halaman ubah Account, user terlebih dahulu mengakses halaman *lihat Account*. Dari nama-nama yang muncul, disebelah paling kanan nama, terdapat link *Edit -Hapus*. Pilih link *Edit* untuk mengubah data Account. Setelah link *Edit* dipilih, maka akan muncul halaman sebagai berikut.

DATA DAFTAR ACCOUNT
Ubah Username Dan Password Account

Personal information

ID

Username

Password

Nama Pegawai

Id Pegawai

Level

Simpan Clear

Gambar 5.25 Halaman untuk Mengubah Account
Sumber: Implementasi

Pada halaman ini user dapat melakukan perubahan terhadap informasi Account, Data yang dapat diubah adalah Username, dan Password. Disertai tombol Simpan untuk menyimpan data yang telah di ubah, dan tombol Batal untuk membatalkan perubahan account.

Algoritma pada tombol simpan diatas, secara sederhana ditunjukkan dalam *pseudocode* berikut ini.

```

update_account
$this->definisikan dengan data2[pilihan kolom] = $_POST[nama field input];
$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

if($_POST['Simpan']){
    $update = new pilih class database();
    $update->Buka Koneksi dengan basis data();
    $update->pilih databse();
    $update->query = "masukan data kedalam databse;";
    $update->jalankan query();
    if($update->query){ echo "Data Berhasil di Update
    <a href='lihat_account.php'>lihat data</a> | <a href='../admin'>kembali
    ke halaman admin</a>";}
    else{ echo "gagal Update data";}
}include "forms_ubah.php";
End update_account

```

Gambar 5.26 Pseudocode pada tombol Simpan Untuk Kebutuhan Fungsional Mengubah Account

Sumber: Implementasi

5.2.1.6 Implementasi Kebutuhan Fungsional Menghapus Account

Pengguna Harus terlebih dahulu masuk sebagai admin. Untuk menghapus Data Account, User terlebih dahulu masuk ke halaman *lihat Account* yang kemudian akan muncul daftar nama, dengan link *Edit-Hapus* disebelah kanan nama. Untuk menghapus data Account, pilih link *hapus* tersebut, dan data account yang dipilih akan dihapus dari database.

Algoritma pada Link *hapus* diatas, secara sederhana ditunjukkan dalam *pseudocode* berikut ini.

```
hapus_account

$this->definisikan dengan data2[pilihan kolom] = $_POST[nama field input];
$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

    if($_GET['id']){
        $update = new pilih class database();
        $update->Buka Koneksi dengan basis data();
        $update->pilih databse();
        $update->query = "hapus data2 dari table admin dengan id =
        '".$_GET[id]."'";
        $update->jalankan query();

        $update2 = new pilih class database();
        $update2->Buka Koneksi dengan basis data();
        $update2->pilih databse();
        $update2->query = "hapus data dari table dengan id = '".$_GET[id]."'";

        $update2->jalankan query();
        if($update->query_result && $update2->query_result){
            echo "Data Berhasil di Hapus
            <a href='lihat_account.php'>lihat data</a> ";
        }else{
            echo "gagal Hapus data";
        }
    }

End hapus_account
```

Gambar 5.27 Pseudocode pada Link *hapus* Untuk Kebutuhan Fungsional Menghapus Account

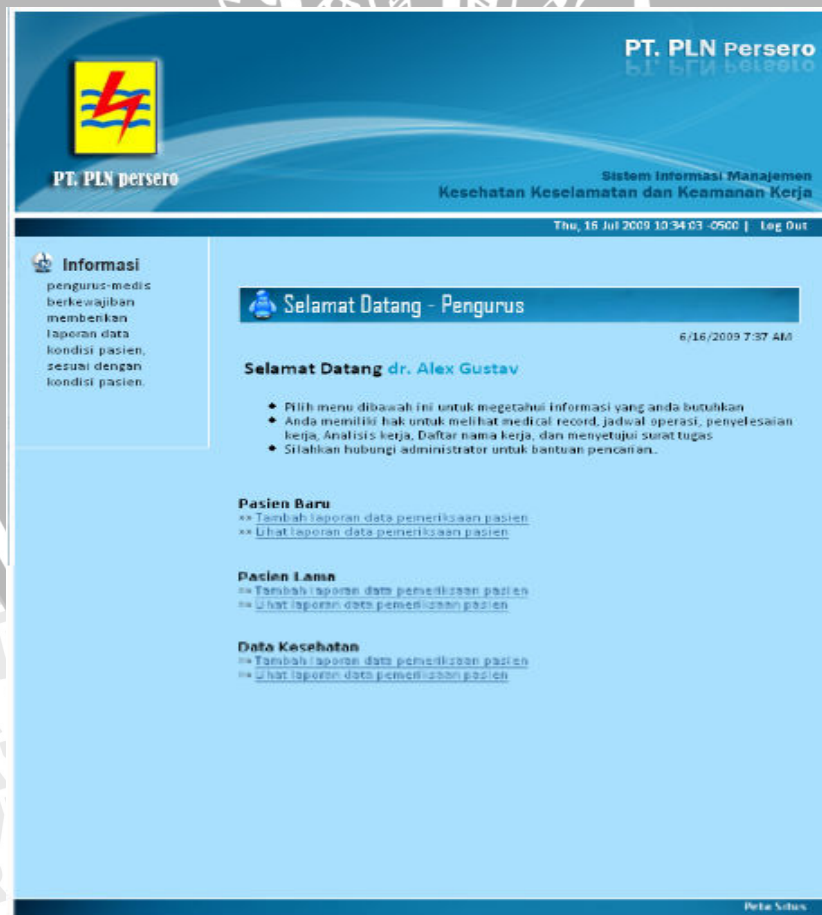
Sumber: Implementasi

5.2.2 Implementasi Penyedia Data Pendukung Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja

Implementasi Penyedia Data Pendukung Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja meliputi beberapa kebutuhan fungsional, yang sebagian dijelaskan yaitu : Menambah Data Kesehatan, Mengubah Data Kesehatan, Menambah Data Pegawai, Mengubah Data Pegawai, Menambah Kegiatan Kerja PDKB, Mengubah Kegiatan Kerja PDKB.

5.2.2.1 Implementasi Kebutuhan Fungsional Menambah Data Kesehatan

Sebelum user bisa mengakses halaman Menambah Data Kesehatan, terlebih dahulu user masuk sebagai *pengelola*. Setelah user masuk sebagai pengelola, maka akan muncul beberapa sub menu, yang ditunjukkan dalam gambar 5.28 berikut ini.



Gambar 5.28 Halaman untuk User Pengurus
Sumber: Implementasi

Dalam halaman pengurus, dibagian atas terlihat menu, yaitu : *Pasien Baru* untuk menambah informasi data pasien baru terdaftar kedalam sistem, *Pasien Lama* untuk menambah atau merubah data kesehatan bagi pasien yang sebelumnya sudah terdaftar, *Data Kesehatan* untuk melihat perkembangan kesehatan pasien, dan terakhir *logout* untuk keluar dari hak user yang telah diakses. untuk menambah Data Kesehatan, pilih link *Tambah laporan data pemeriksaan pasien*. Setelah link *Tambah Data* dipilih, maka akan muncul halaman *Tambah Data*, seperti gambar 5.29 berikut.

DATA ACCOUNT

Daftar Account

- Tekan tombol *Tambah* untuk mengirim data nama pegawai yang memenuhi syarat kesehatan.
- Pilih kolom edit - hapus untuk mengubah atau menghapus data pegawai

Link Tambah	Id Pegawai	Nama	nip	Jabatan	Kompetensi	Unit
tambah	390023	Hafid Ali	3339999	own	own	own
tambah	39007	Wahyudiono	699207	Spv. Opdist & Pel. Teknik	K2/K3 Pengawas II	UPJ Tumpang
tambah	39002	Nanang Yudianto	820403	Terampil Linesmen PDKB	Kompetensi PDKB Ang. V	APJ Mlg
tambah	390022	Sumarsono Tasmun	638333	TU Pemeliharaan Distribusi	K2/K3 Pelaksana II	UPJ Lawang
tambah	390010	Priadi Antoro	628512	TU Kepala Regu PDKB	Kompetensi PDKB	APJ Mlg
tambah	390016	Setyo Wahono	547909	TU Linesmen PDKB	Kompetensi PDKB Ang. V	APJ Mlg
tambah	39003	Supriadi	669306	TU Perencanaan Opdist	K2/K3 Pelaksana II	UPJ Lawang
tambah	39004	Trisno Widji Santoso	669206	TU Pelayanan Teknik	K2/K3 Pelaksana II	UPJ Gondanglegi
tambah	390012	Wahyu Nagano	689107	Spv. kantor Pel. Smbr Pcnng	K2/K3 Pengawas II	UPJ Kepanjen
tambah	390017	Wibisono	638522	Spv. Opdist & Pel. Teknik	K2/K3 Pengawas II	UPJ Gondanglegi

Page 1 of 1 (10 rows)

Pilih link *tambah* disamping nama pegawai untuk menambahkan data nama pegawai yang memiliki kelayakan kompetensi.

Gambar 5.29 Halaman Menampilkan Data Nama untuk Menambah Data Kesehatan
Sumber: Implementasi

Kemudian akan muncul daftar nama-nama pegawai yang telah memiliki Account, diikuti link *tambah* di sebelah kiri nama pegawai. Untuk menambah informasi data pegawai, pilih link *tambah*. yang kemudian akan muncul form untuk mengisi data kesehatan, seperti terlihat pada gambar 5.29 berikut.

Isi form Data Kesehatan di bawah ini

Personal information

No Penerimaan (Id Pasien) 39003

Nama Pasien **Muhammad Toni**

Keterangan Dokter

Id Dokter *

Nama Dokter *

Tanggal Periksa *
(yyyy mm.dd)

[masukan kondisi demam](#)

Keterangan Kesehatan

*Tentukan Rujukan:

**Jenis Rujukan:

* isi bila terdapat rujukan
** Pilih klinik bila terdapat rujukan

Gambar 5.30 Halaman yang menampilkan form untuk mengisi data kesehatan
Sumber: Implementasi

Terdapat beberapa field untuk diisi mengenai data kesehatan pegawai yang namanya dipilih, kemudian disediakan tombol *Simpan* untuk menyimpan informasi kesehatan ke dalam database, dan tombol *clear* untuk menghapus data dari field yang telah diisikan.

Algoritma pada tombol *Simpan* diatas, ditunjukkan secara sederhana pada *Pseudocode* berikut ini.


```

tambah_kesehatan
$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

$lapor = new pilih class database();
$lapor->Buka Koneksi dengan basis data();
$lapor->pilih databse();
$lapor->query = "masukan data kedalam database db_kesehatan_daftar_pas;";
$lapor->jalankan query();

$lapor = new pilih class database();
$lapor->Buka Koneksi dengan basis data();
$lapor->pilih databse();
$lapor->query = "update data2 kedalam database WHERE idpegawai;";
$lapor->jalankan query();

if($lapor->query && $lapor2->query){
echo "<tampilkan halaman='lihat_kesehatan.php'>";
}
End tambah_kesehatan
    
```

Gambar 5.31 Pseudocode pada Link Simpan Untuk Kebutuhan Fungsional Menambah Data Kesehatan
Sumber: Implementasi

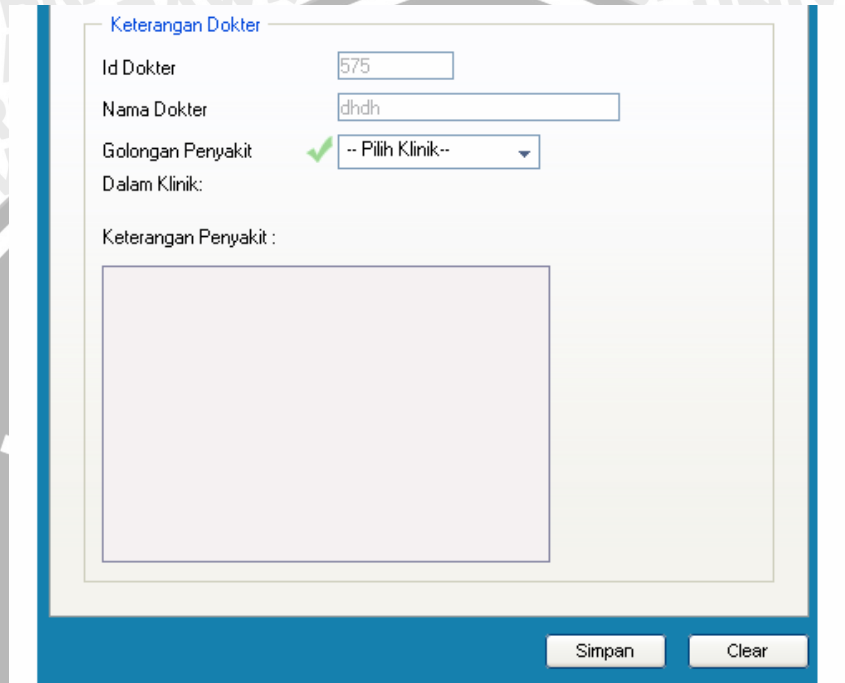
5.2.2.2 Implementasi Kebutuhan Fungsional Mengubah Data Kesehatan

Pengguna Sebelumnya telah login sebagai Pengurus, dibagian sebelah kiri akan muncul submenu-submenu, kemudian User memilih submenu Data Kesehatan dengan link *lihat Data*, yang ditunjukkan dalam gambar 5.31 berikut.

Id	Nama	Tanggal Lahir	Sex	Golongan Sakit	Keterangan	Edit - Hapus
10000	Setyo Wahono	2/3/67	Laki-laki	Gizi	dhdhd	Edit - Hapus
2	pratama	22072008	P	Penyakit Dalam	Tenaga dalam nya kuat sekali!	Edit - Hapus
3	sandra	12052009	P	Rawat Inap	Menginap 3 malam	Edit - Hapus
4	hera	12-12-2345	Perempu	Gizi	wokeh	Edit - Hapus
5	sammy	04-08-2009	Laki-Laki	Syaraf	syaraf ketawa rusak!	Edit - Hapus
6	erwin	12-06-2009	Laki-laki	Mata	sakit mata	Edit - Hapus
1	Hafid	09-09-1994	Laki-laki	Syaraf	sangat pintar	Edit - Hapus
10000	Nanang Yudianto	09-07-6667	Laki-laki	Jantung	dhghgdhd	Edit - Hapus
10000	Trisno Widji Santoso	05-05-7868	Laki-laki	Unit Gawat Darurat	gdgaga	Edit - Hapus
10000	Suyoko	22-12-3455	Laki-laki	Syaraf	sdfsdfsdsd	Edit - Hapus

Gambar 5.32 Halaman Menampilkan nama pegawai, untuk Mengubah *Data Kesehatan*
Sumber: Implementasi

Kemudian muncul Daftar nama yang telah ditambah Data Kesehatannya, disertai link *Edit - Hapus*, disebelah kanan nama. pilih link *Edit* untuk mengubah Data kesehatan pegawai. Setelah link *Edit* dipilih, akan muncul form seperti gambar dibawah ini.



The screenshot shows a web form titled "Keterangan Dokter" with the following elements:

- Id Dokter:** A text input field containing the value "575".
- Nama Dokter:** A text input field containing the value "dhdh".
- Golongan Penyakit:** A dropdown menu with a green checkmark icon and the text "-- Pilih Klinik--".
- Dalam Klinik:** A label positioned below the dropdown menu.
- Keterangan Penyakit :** A large, empty text area for providing details about the disease.
- Buttons:** Two buttons at the bottom right, labeled "Simpan" (Save) and "Clear".

Gambar 5.33 Halaman yang menampilkan form untuk mengubah data kesehatan

Sumber: Implementasi

Dalam Form tersebut disediakan tombol *Simpan* untuk menyimpan data yang telah diubah, dan tombol *Clear* untuk mengkosongkan field kembali.

Algoritma Simpan diatas secara sederhana ditunjukkan dalam *Pseudocode* berikut ini.


```
mengubah_kesehatan

$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

if($_POST['Simpan']){
    $update = new pilih class database();
    $update->Buka Koneksi dengan basis data();
    $update->pilih databse();
    $update->query = "Update data didalam database db_kesehatan_daftar_pas;";
    $update->jalankan query();

    if($update->query){
        echo "Data Berhasil di Update
        <a href='lihat_account.php'>lihat data</a> | <a
        href='../admin'>kembali ke halaman admin</a>";
    }else{
        echo "gagal Update data";}

    }include "forms_ubah.php";

End mengubah_kesehatan
```

Gambar 5.34 Pseudocode pada Link Simpan Untuk Kebutuhan Fungsional Mengubah Data Kesehatan

Sumber: Implementasi

5.2.2.3 Implementasi Kebutuhan Fungsional Menambah Data Pegawai

User harus terlebih dahulu masuk sebagai Admin. Setelah User masuk kehalaman Admin, maka akan tampil beberapa link, dimana untuk menambah datam pegawai terdapat dalam link *Masukan Nama Pegawai*. Setelah user memilih link *Masukan Nama Pegawai* maka akan muncul form untuk menambahkan keterangan pegawai, seperti terlihat dalam gambar 5.35 berikut.

Masukan Data Nama Pegawai

Personal information

ID Pegawai	✓	<input type="text" value="390025"/>
Nama	✓	<input type="text" value="Ramli Husada"/>
NIK	✓	<input type="text" value="556789 J"/>
Jabatan		<input type="text" value="TU Linesmen PDKB"/> ✓
Kursus		<input type="text" value="Perke"/>
unit		<input type="text" value="APJ Mlg"/>

Pilih Link [Batal](#) berikut untuk kembali kehalaman daftar Account

Gambar 5.35 Halaman Menampilkan Daftar nama untuk Menambah Data Pegawai

Sumber: Implementasi

Halaman Utama Tentang K3 Divisi PDKB Info Umum Log Out

Informasi
Admin bertindak dalam hal pembuatan Account baru

Download the sample html file and css file

Selamat Datang

Selamat Datang Hafid Ali

Tanggal 12 April 2009 19:21:151





Gambar 5.36 Halaman Menampilkan halaman utama admin

Sumber: Implementasi

Setelah setiap field telah diisi, disediakan tombol simpan dibagian paling bawah form, untuk menyimpan data ke dalam basis data.

Algoritma Simpan diatas secara sederhana ditunjukkan dalam *Pseudocode* berikut ini.

```
Menambah_datapegawai

$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

    $catat = new pilih class database();
    $catat->Buka Koneksi dengan basis data();
    $catat->pilih databse();
    $catat->query = "masukan data kedalam database db_pegawai;";
    $catat->jalankan query();

        if($catat->query){
            echo "<tampilkan halaman=lihat_datapegawai.php>";
        }else{
            echo "GAGAL";
        }

End Menambah_datapegawai
```

Gambar 5.37 *Pseudocode* pada Link Simpan Untuk Kebutuhan Fungsional Menambah Data Pegawai

Sumber: Implementasi

5.2.2.4 Implementasi Kebutuhan Fungsional Mengubah Data Pegawai

User harus terlebih dahulu masuk sebagai pengurus. Setelah User masuk kehalaman pengurus, maka akan muncul beberapa submenu yang salah satunya terdapat submenu Data Pegawai dengan link *Lihat Data*. Setelah user memilih link *Lihat Data*, maka akan muncul barisan nama-nama yang memenuhi syarat kelayakan. Seperti terlihat dalam gambar 5.38 berikut.

• Tekan tombol *edit* - *hapus* untuk menghapus atau mengubah nama pegawai.

Id Pegawai	Nama	Divisi	Nik	Golongan Sakit	Keterangan Sakit	Edit - Hapus
390023	Hafid Ali	own	own	Gizi	AAAA	Edit - Hapus
39004	Trisno Widji Santoso	K2/K3 Pelaksana II	UPJ Gondanglegi	Syaraf	BAek-baek saja...	Edit - Hapus
390012	Wahyu Nagano	K2/K3 Pengawas II	UPJ Kepanjen	Penyakit Dalam	sehat	Edit - Hapus

Page 1 of 1 (3 rows)

Gambar 5.38 Halaman Menampilkan Daftar nama untuk *Menambah Data Pegawai*
Sumber: Implementasi

Kemudian akan muncul barisan nama, dilengkapi dengan link *Edit - Hapus* ,
 untuk mengubah data pegawai, pilih link *Edit* dan akan muncul halaman berikut.



Personal information

No Daftar 39004

Nama Pegawai * Trisno Widji Santoso

Divisi *

NIK 6692069

Tanggal Lahir(d.m.y)

Alamat *

Zip 0 * Data tidak sesuai 00000

Jenis Kelamin * Laki - Laki Perempuan

Telpon

Online Information

Email

Email Lain *

Gambar 5.39 Halaman Menampilkan Daftar nama untuk Menambah Data Pegawai
Sumber: Implementasi

Disediakan beberapa field untuk mengubah informasi data pegawai, dengan tanda peringatan, apabila salah satu field yang harus diisi masih kosong. Setelah data di isi tekan tombol *Simpan* untuk mengubah data pegawai sebelumnya, dan untuk mengkosongkan field yang telah diisi, tekan tombol *Clear*.

Algoritma Simpan diatas secara sederhana ditunjukkan dalam *Pseudocode* berikut ini.

```
Mengubah_datapegawai

$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

$update = new pilih class database();
$update->Buka Koneksi dengan basis data();
$update->pilih database();
$update->query = "Update data didalam database db_pegawai WHERE
idpegawai;";
$update->jalankan query();

if($update->query){
echo "Data Berhasil di Update
<a href='lihat_account.php'>lihat data</a> | <a
href='../admin'>kembali ke halaman admin</a>";
}else{
echo "GAGAL UPDATE DATA";}

End Mengubah_datapegawai
```

Gambar 5.40 Pseudocode pada Link Simpan Untuk Kebutuhan Fungsional Mengubah Data Pegawai

Sumber: Implementasi

5.2.2.5 Implementasi Kebutuhan Fungsional Menambah Kegiatan Kerja PDKB

User harus terlebih dahulu masuk sebagai pengurus. Setelah User masuk kehalaman pengurus, maka akan muncul beberapa submenu yang salah satunya terdapat submenu Kerja PDKB dengan link *Tambah Data*. Setelah user memilih link *Tambah Data* maka akan muncul form tambah data kegiatan kerja PDKB, seperti terlihat dalam gambar 5.41 berikut.

Admin pengurus bertindak dalam hal kesehatan, kepegawaian, dan kegiatan kerja pada divisi PDKB.

Pilihan Data Kegiatan kerja:
Lihat Data
Ubah Data
Hapus Data

ISI DATA KEGIATAN KERJA
Isi form Laporan Kerusakan/Perbaikan

Data Laporan

No Laporan *

Lokasi *

Uraian Pekerjaan *

Keterangan Lain :

Tanggal Pemeriksaan (d.m.y)

Tanggal Laporan (d.m.y)

Simpan Hapus

Gambar 5.41 Halaman Menampilkan Daftar nama untuk *Menambah Kegiatan Krja PDKB*
Sumber: Implementasi

Akan muncul form seperti gambar diatas, disediakan field-field yang berkaitan dengan penambahan informasi data Kegiatan kerja. Disebelah kiri juga akan muncul submenu kegiatan kerja, untuk melihat, mengubah, dan menghapus data, berkaitan dengan kegiatan Kerja PDKB. Setelah Data diisi, pengurus menyimpan informasi dengan menekan tombol *Simpan* :

Algoritma Simpan diatas secara sederhana ditunjukkan dalam *Pseudocode* berikut ini.

Menambah_kegiatankerja

```
$this->definisikan dengan data2[pilihan kolom] = $_POST[nama field input];
$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];
```

```
$add_pdkb = new pilih class database();
$add_pdkb->Buka Koneksi dengan basis data();
$add_pdkb->pilih databse();
$add_pdkb->query = "masukan data kedalam database
db_kegiatankerja_pdkb;";
$add_pdkb->jalankan query();
```

```
$add_pdkb = new pilih class database();
$add_pdkb->Buka Koneksi dengan basis data();
$add_pdkb->pilih databse();
$add_pdkb->query = "masukan data2 kedalam databse;";
$add_pdkb->jalankan query();
```

```
if($add_pdkb->query){
    echo "<tampilkan halaman='lihat_datapdkb.php'>";
}else{
    echo "GAGAL";}
```

End Menambah_kegiatankerja

Gambar 5.42 Pseudocode pada Link Simpan Untuk Kebutuhan Fungsional Menambah Kegiatan Kerja

Sumber: Implementasi

5.2.2.6 Implementasi Kebutuhan Fungsional Mengubah Kegiatan Kerja PDKB

User harus terlebih dahulu masuk sebagai pengurus. Setelah User masuk kehalaman pengurus, maka akan muncul beberapa submenu yang salah satunya terdapat submenu Kerja PDKB dengan link *Lihat Data*. Setelah user memilih link *Lihat Data*, maka akan muncul barisan kegiatan kerja. Seperti terlihat dalam gambar 5.43 berikut.

DATA KEGIATAN KERJA PDKB
 Tambah Kegiatan Kerja

• Tekan tombol *edit* - *hapus* untuk menghapus atau mengubah nama pegawai.

No Lap	Lokasi	Uraian Pekerjaan	Keterangan	Tanggal Laporan	Edit - Hapus
1	UPJ Gondanglegi	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	01.11.2008	Edit - Hapus
2	UPJ Singosari	Pemas Arester untuk penangkalan petir	volume 6 set	11.11.2008	Edit - Hapus
3	APJ Malang	Penggantian Trafo Untuk Program Preman Trafo	Volume 31 Unit	22.05.2009	Edit - Hapus
4	UPJ Ngantang	Penambahan Grounding TR U/perbaikan teg ujung	Volume 22 Set	01.01.2009	Edit - Hapus
5	UPJ Kepanjen	Pengadaan Kwh meter 3phasa	Volume 50 buah	28.03.2009	Edit - Hapus

Page 1 of 1 (5 rows)

Gambar 5.43 Halaman Menampilkan Daftar Kegiatan Kerja untuk *Mengubah Data Kegiatan Kerja PDKB*
Sumber: Implementasi

Kemudian akan muncul data-data Kegiatan kerja, dilengkapi dengan link *Edit - Hapus* , untuk mengubah data kegiatan kerja, pilih link *Edit* untuk mengubah dan akan muncul halaman berikut.

Personal information

No Laporan

No Laporan

Nama Lokasi

Uraian Pekerjaan

Keterangan

Tanggal

Pemeriksaan(d.m.y) Tanggal

Laporan(d.m.y)

Gambar 5.44 Halaman Menampilkan Form untuk mengubah Data Kegiatan Kerja
Sumber: Implementasi

Disediakan beberapa field untuk mengubah informasi data kegiatan kerja. Setelah data di isi tekan tombol *Simpan* untuk mengubah informasi data Kegiatan Kerja sebelumnya, dan untuk mengkosongkan field yang telah diisi, tekan tombol *Clear*.

Algoritma Simpan diatas secara sederhana ditunjukkan dalam *Pseudocode* berikut ini.

```
Mengubah_kegiatankerja

$this->definisikan dengan data2[pilihan kolom] = $_POST[nama field input];
$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

    if($_POST['Simpan']){
        $update_pdkb = new pilih class database();
        $update_pdkb->Buka Koneksi dengan basis data();
        $update_pdkb->pilih databse();
        $update_pdkb->query = "Update data didalam databse WHERE idlapkerja";
        $update_pdkb->jalankan query();

        $update_pdkb2 = new pilih class database();
        $update_pdkb2->Buka Koneksi dengan basis data();
        $update_pdkb2->pilih databse();
        $update_pdkb2->query = "Update data didalam databse WHERE idpekerjaan";
        $update_pdkb2->jalankan query();

        if($update_pdkb->query_result && $update_pdkb2->query_result){
            echo "Data Berhasil di Update
            <a href='lihat_account.php'>lihat data</a> | <a
            href='../admin'>kembali ke halaman admin</a>";
        }else{
            echo "gagal Update data";
        }include "forms_ubah.php";
    }

End Mengubah_kegiatankerja
```

Gambar 5.45 Pseudocode pada Link *Simpan* Untuk Kebutuhan Fungsional Mengubah Data Kegiatan Kerja

Sumber: Implementasi

5.2.3 Implementasi Pengelolaan Informasi Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja

Implementasi Pengelolaan Informasi Sistem Informasi Kesehatan Keselamatan dan Keamanan Kerja meliputi beberapa kebutuhan fungsional, yang

sebagian dijelaskan yaitu : Menambah Jadwal Operasi Perbaikan dan Perawatan, Melihat Jadwal Operasi Perbaikan dan Perawatan, Menyusun Analisis Pekerjaan dan Pembagian Tugas, Melihat Analisis Pekerjaan dan Pembagian Tugas, Memeriksa Kelayakan Kerja Pegawai, Melihat Nama Calon Pekerja, Membuat Surat Tugas, Melihat Surat Tugas

5.2.3.1 Implementasi Kebutuhan Fungsional Menambah Jadwal Operasi Perbaikan dan Perawatan

Untuk mengakses halaman ini user harus masuk sebagai Pengelola. Kemudian akan muncul halaman, disertai beberapa sub menu, dimana salah satunya adalah sub menu Tambah data pada keterangan submenu Jadwal Operasi.

Informasi
Admin pengelola bertindak dalam hal penyusunan jadwal operasi, Penentuan Calon Pekerja, dan Penentuan tindakan kerja pdkb.

Pilih Menu

TINDAKAN KERJA PDKB
-> [Tambah Data](#)
-> [Lihat Data](#)

JADWAL OPERASI
-> [Tambah Data](#)
-> [Lihat Data](#)

ATUR KELAYAKAN PELAKSANA
-> [Tambah Data](#)
-> [Lihat Data](#)

[Download the sample html file and css file](#)

Selamat Datang - Pengelola

Selamat Datang Pengelola. Disini kami menyediakan Sistem Informasi Manajemen Kesehatan Keselamatan & Keamanan Kerja, Khususnya di Divisi PDKB (Pekerjaan Dalam Keadaan Berbahaya). Guna dibuatnya situs ini ialah untuk mempermudah Pihak Perusahaan, dalam hal ini PT. PLN persero Area Pelayanan dan Jaringan Malang, dalam pengelolaan Sistem Kesehatan Keselamatan dan Keamanan Kerja.

- [Link Jadwal](#) untuk menyusun jadwal operasi yang akan dilaksanakan.
- [Link Usulan Nama Pekerja](#)
- [Link Tindakan PDKB](#) digunakan dalam menentukan tindakan PDKB yang akan dilaksanakan dalam penanganan Laporan Kerusakan atau perbaikan.

Gambar 5.46 Submenu Tambah Data pada Jadwal Operasi

Sumber: Implementasi

Setelah sub menu *tambah Data* dipilih, maka akan muncul halaman seperti dibawah ini.

DATA TAMBAH JADWAL
[Lihat Jadwal](#)

- Tekan tomoledit - hapus untuk menghapus atau mengubah nama pegawai.

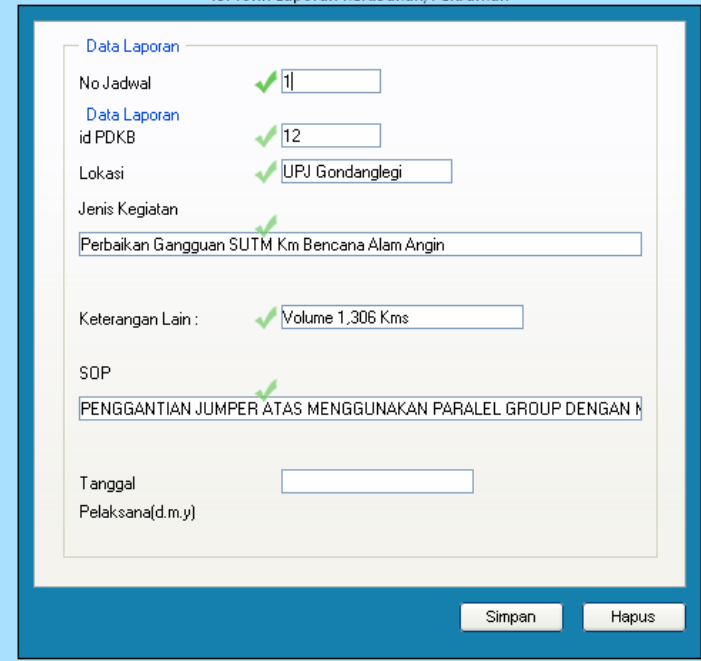
Susun	No	Lokasi	Uraian Pekerjaan	Keterang	SOP
Buat Jadwal	1	UPJ Gondangkl	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	PENGGANTIAN JUMPER ATAS MENGGUNAKAN PARALEL GROUP DENGAN METODE BERJARAK
Buat Jadwal	2	UPJ Singosari	Pemas Arester untuk penangkalan petir	volume 6 set	PEMASANGAN SAMBUNGAN BARU SKTM KE SUTM DENGAN METODE BERJARAK
Buat Jadwal	3	APJ Malang	Penggantian Trafo Untuk Program Preman Trafo	Volume 31 Unit	ssadasdsa
Buat Jadwal	4	UPJ Ngantang	Penambahan Grounding TR U/perbaikan teg ujung	Volume 22 Set	SAMBUNGAN BARU BAWAH LURUS DENGAN METODE BERJARAK
Buat Jadwal	5	UPJ Kepanjen	Pengadaan Kwh meter 3phasa	Volume 50 buah	

Page 1 of 1 (5 rows)

Gambar 5.47 Halaman yang menampilkan Laporan Kerusakan dan Perbaikan, yang akan ditentukan jadwal Penanganannya
Sumber: Implementasi

Pada halaman ini akan muncul Laporan Kerusakan atau Perawatan. Disamping masing-masing kegiatan tersebut terdapat link *Buat Jadwal*. Jika hendak menentukan jadwal pelaksanaannya, pilih link tersebut pada salah satu baris kegiatan tersebut.

Tampak gambar dibawah ini, apabila ditekan link *Buat Jadwal* tersebut.



Data Laporan

No Jadwal ✓ 11

Data Laporan
id PDKB ✓ 12

Lokasi ✓ UPJ Gondanglegi

Jenis Kegiatan ✓
Perbaikan Gangguan SUTM Km Bencana Alam Angin

Keterangan Lain : ✓
Volume 1,306 Kms

SOP ✓
PENGANTIAN JUMPER ATAS MENGGUNAKAN PARALEL GROUP DENGAN M

Tanggal

Pelaksana(d.m.y)

Simpan Hapus

Gambar 5.48 Halaman yang menampilkan form untuk menambah Jadwal
Sumber: Implementasi

Jenis Kegiatan akan muncul, diikuti informasi keterangan lain dan keterangan SOP yang digunakan, dan sebuah *textbox* tanggal pelaksanaan. Beserta sebuah tombol simpan, untuk menyimpan data, dan tombol hapus untuk mengosongkan isi form.

Algoritma pada tombol diatas, secara sederhana dapat ditunjukkan dalam *pseudocode* berikut ini :

```

add_lap_jadwal

$this->definisikan dengan data2[pilihan kolom] = $_POST[nama field input];
$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

    $add_jadwal = new pilih class database();
    $add_jadwal->Buka Koneksi dengan basis data();
    $add_jadwal->pilih databse();
    $add_jadwal->query = "masukan data kedalam database db_jadwaloperasi;";
    $add_jadwal->jalankan query();

    $add_jadwal3 = new pilih class database();
    $add_jadwal3->Buka Koneksi dengan basis data();
    $add_jadwal3->pilih databse();
    $add_jadwal3->query = "update data2 dalam database db_kegiatankerja_pdkb
    WHERE idkegiatanpdkb;";
    $add_jadwal3->jalankan query();

    if($add_jadwal->query){
    echo "<tampilkan halaman='lihat_jadwal.php'>";
    }else{
    echo "GAGAL";}}

End add_lap_jadwal

```

Gambar 5.49 Pseudocode pada tombol Simpan untuk Kebutuhan Fungsional Menambah Jadwal Operasi Perbaikan Dan Perawatan
Sumber: Implementasi

5.2.3.2 Implementasi Kebutuhan Fungsional Mengubah Jadwal Operasi Perbaikan dan Perawatan

Untuk mengakses halaman ini user harus masuk sebagai Pengelola. Kemudian akan muncul halaman utama pengelola seperti gambar 5.50, disertai beberapa sub menu, dimana salah satunya adalah sub menu Lihat data pada keterangan submenu Jadwal Operasi.

Setelah sub menu *Lihat Data* dipilih, maka akan muncul halaman seperti dibawah ini.

DATA LIHAT JADWAL
TAMPIL JADWAL

- Tekan tombol edit - hapus untuk menghapus atau mengubah nama pegawai.

Nc	Lokasi	Uraian Pekerjaan	Keterangan	SOP	Lihat - Edit - Hapus
1	UPJ Gondang	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	PENGGANTIAN JUMPER ATAS MENGGUNAKAN PARALEL GROUP DENGAN METODE BERJARAK	Tampil - Edit - Hapus
2	UPJ Singosari	Pemas Aresrer untuk penangkalan petir	volume 6 set	PEMASANGAN SAMBUNGAN BARU SKTM KE SUTM DENGAN METODE BERJARAK	Tampil - Edit - Hapus

Page 1 of 1 (2 rows)

Gambar 5.50 Halaman yang menampilkan Laporan Kerusakan dan Perbaikan yang telah ditentukan tanggal penanganannya
Sumber: Implementasi

Pada halaman ini akan muncul Laporan Kerusakan atau Perawatan yang telah ditentukan tanggal penanganannya. Disamping masing-masing kegiatan tersebut terdapat link *Tampil - Edit - Hapus*, yang berfungsi untuk menampilkan data secara detail, mengubah data jadwal, dan menghapus kegiatan. Jika hendak merubah data jadwal, pilih link *Edit* pada salah satu baris kegiatan tersebut.

Data Laporan

No Jadwal

Lokasi

Jenis Kegiatan

Keterangan Lain :

SOP

Tanggal Pelaksanaan(d.m.y)

Gambar 5.51 Halaman yang menampilkan form untuk mengubah jadwal
Sumber: Implementasi

Jenis Kegiatan akan muncul, diikuti informasi keterangan lain dan keteranga SOP yang digunakan, dan sebuah *textbox* tanggal pelaksanaan. Beserta sebuah tombol Ubah, untuk mengubah data, dan tombol hapus untuk mengosongkan isi form.

Algoritma pada tombol diatas, secara sederhana dapat ditunjukkan dalam *pseudocode* berikut ini :

```

update_lap_jadwal

$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

    if(kirim variable 'Ubah'){

        $update_jadwal = new pilih class database();
        $update_jadwal->Buka Koneksi dengan basis data();
        $update_jadwal->pilih databse();
        $update_jadwal->query = "update data2 dalam database
db_jadwaloperasi WHERE idjadwal;";
        $update_jadwal->jalankan query();

        if($update_jadwal->query_result){
echo "</br><b> Data Berhasil di Update</b> </br></br>
<a href='lihat_jadwal_fix.php'>lihat data</a> | <a
href='../..//pengelola'>kembali ke halaman pengurus</a>";
        }else{
echo "gagal Update data";
        }include "form_ubah_jadwal.php";
        }

End update_lap_jadwal

```

Gambar 5.52 *Pseudocode* pada tombol Ubah untuk Kebutuhan Fungsional Mengubah Jadwal Operasi Perbaikan Dan Perawatan
Sumber: Implementasi

5.2.3.3 Implementasi Kebutuhan Fungsional Memeriksa Kelayakan Kerja Pegawai

Untuk menambah data pegawai, terlebih dahulu harus di periksa kelayakan kerjanya, dengan cara User harus terlebih dahulu masuk sebagai pengurus. Setelah User masuk kehalaman pengurus, maka akan muncul beberapa submenu yang salah satunya terdapat submenu Atur Kelayakan Pelaksana dengan link *Periksa Kelayakan*. Setelah user memilih link *Periksa Kelayakan* maka akan muncul barisan nama-nama

yang sebelumnya telah melakukan pemeriksaan kesehatan, seperti terlihat dalam gambar 5.35 berikut.

◆ Tekan tombol *Tambah* untuk mengirim data nama pegawai yang memenuhi syarat kesehatan.
 ◆ Pilih kolom edit - hapus untuk mengubah atau menghapus data pegawai

Link Detil	Id Pegawai	Nama	NIK	Jabatan	Kelayakan
Check	39001	Abdullah Besari	5782288	ATMP Preparator PDKB	Layak
Check	39002	Nanang Yudianto	8204034	Terampil Linesmen PDKB	
Check	39003	Supriadi	6693069	TU Perencanaan Opdist	
Check	39004	Trisno Widji Santoso	6692069	TU Pelayanan Teknik	
Check	39005	Tri Hartatik Wibison	7329736	TU Pelayanan keuangan	tidak memenuhi persyaratan jabatan
Check	39007	Wahyudiono	6992078	Spv. Opdist & Pel. Teknik	Layak
Check	39009	Muhammad Toni	8204033	Terampil Linesmen PDKB	Layak
Check	390010	Priadi Antoro	6285129	TU Kepala Regu PDKB	tidak memenuhi syarat kompetensi
Check	390012	Wahyu Nagano	6891076	Spv. kantor Pel. Smbr Pcngr	Layak
Check	390013	Acmad Susanto	5481248	TU Kepala Regu PDKB	

Page 1 [2](#) > >> of 2 (11 rows)

Pilih link *tambah* disamping nama pegawai untuk menambahkan data nama pegawai yang memiliki kelayakan kompetensi.

Gambar 5.53 Halaman Menampilkan Daftar nama untuk *Memeriksa kelayakan kerja pegawai*

Sumber: Implementasi

Nama-nama yang telah memeriksa kesehatan, akan ditampilkan dalam daftar nama. untuk memeriksa kelayakan kerja masing-masing nama diatas, berdasarkan kelayakan kompetensi, dan data kesehatan, disediakan link *Check* di kolom pertama dari daftar nama, dimana pegawai yang belum diperiksa kelayakannya dapat dilihat dari laporan kelayakan pada kolom *kelayakan* masih belum terisi. Sedangkan untuk pegawai yang telah diperiksa maka hasil laporan akan langsung ditampilkan di kolom *kelayakan*. Ketika link dipilih, maka akan muncul form sebagai berikut.

Personal information

Id Pegawai 390014

Nama Pegawai Rudy Hartono

id 20

Syarat Kondisi Kesehatan

1. Masukan Batasan Tanggal Pemeriksaan :

Batasan Tanggal

Periksa

Masukan Batasan Tanggal Kelayakan yang dikehendaki (yyyy-mm-dd)

2. Batasan Suhu Tubuh Normal :

Suhu tubuh normal

Batasan kelayakan suhu tubuh dalam derajat celsius (°C)

3. Pilih Rujukan untuk mengetahui pegawai dalam proses Rujukan atau tidak:

Pilih Rujukan none

Syarat kelayakan pelatihan

1. Jabatan pegawai :

Pertimbangan jabatan pegawai

2. Pelatihan yang telah ditempuh :

syarat pelatihan yang telah ditempuh

Kelayakan Kerja

Silahkan masukan persyaratan, dan klik tombol check.

[Result](#)

Periksa kelayakan pegawai lain [Keluar](#)

Gambar 5.54 Halaman Menampilkan Form untuk *Memeriksa kelayakan kerja pegawai*
Sumber: Implementasi

Akan muncul form seperti gambar diatas, dimana seleksi dibagi menjadi dua tahap, tahap yang pertama adalah memeriksa syarat kondisi kesehatan, dengan berdasar tanggal pemeriksaan, suhu tubuh, dan rujukan. Tahapan kedua adalah syarat kompetensi yang diseleksi melalui syarat kelayakan pelatihan, dengan pertimbangan jabatan, dan pelatihan yang ditempuh

Setelah seleksi kelayakan telah ditentukan, pengurus memeriksa kelayakan pegawai dengan menekan tombol *Check*.

Algoritma Simpan diatas secara sederhana ditunjukkan dalam *Pseudocode* berikut ini.

```

Memeriksa_kelayakanPegawai
if($login->query_result)
{
  while($result_row = mysql_fetch_row($login->query_result))
  {
    if ($result_row[0] == $this->idpegawai)
    {
      if($result_row[6] >= $this->tgl_periksa)
      {
        if(($result_row[14] == $this->suhu) or ($this->suhu == 38) or ($this->
suhu == 36))
        {
          if(($result_row[12] == 'Gigi') or ($result_row[12] == 'Spesial')
or ($result_row[12] == 'Paramedis') or ($result_row[12] ==
'Puskesmas') or ($result_row[12] == 'RSU'))
          {
            echo "Terdapat Rujukan yang tidak memenuhi syarat bagi
pegawai untuk bekerja";

            echo '<meta http-equiv="refresh"
content=0;url=help_lihat_tambah.php?id='.$this->id.'>';

            $update_layak = new database();
            $update_layak->connect_db();
            $update_layak->select_db();

            $update_layak->query="update db_kesehatan_daftar_pas set
kelayakan='".$this->rujukan.'" WHERE idpasien = '".$this->
>idpegawai."";

            $update_layak->run_query();

            $update_layak = new database();
            $update_layak->connect_db();
            $update_layak->select_db();

            $update_layak->query="update db_usulanpekerjaan set
statusmanajerapj='".$this->rujukan.'" WHERE idpegawai = '".$this->
>idpegawai."";

            $update_layak->run_query();
          }else{
            $update_layak = new database();
            $update_layak->connect_db();
          }
        }
      }
    }
  }
}

```

```

$update_layak->select_db();

$update_layak->query="update db_kesehatan_daftar_pas
set kelayakan='Layak' WHERE idpasien = ".$this->idpegawai."";

$update_layak->run_query();

$update_layak2 = new database();

$update_layak2->connect_db();

$update_layak2->select_db();

$update_layak2->query = "insert into
db_usulanpekerjaan (idpegawai) VALUES('".$this->idpegawai."')";

$update_layak2->run_query();

echo '<div class="zpFormContent"><fieldset>

<legend>Personal information</legend>

<label class="zpFormLabel">Id Pegawai :</label>

<b>'.$this->idpegawai.'</b><br /><br />

<label class="zpFormLabel">Nama Pegawai :</label>

<b>'.$this->nama.'</b><br /></fieldset>';

echo '<fieldset>

<legend>Tanggal Kelayakan Periksa
Kesehatan</legend>

<font size="3" color="#33CC00" ><b> Tanggal
Masih Dalam Batas Layak. Silahkan kehalaman Syarat Kondisi
sehat<b></font>';

echo '</fieldset>'; echo '</div>';

if(($result_row[3] == 'ATMP Preparator PDKB') or ($result_row[3]
== 'TU Kepala Regu PDKB') or ($result_row[3] == 'TU Linesmen PDKB') or
($result_row[3] == 'Terampil Linesmen PDKB') or ($result_row[3] ==
'Spv. kantor Pel. Smbr Pcng') or ($result_row[3] == 'Spv.kantor
Pel.Smbr Pcng') or ($result_row[3] == 'TU Perencanaan Opdist') or
($result_row[3] == 'TU Pemeliharaan Distribusi') or ($result_row[3] ==
'TU Perencanaan Pemdist') or ($result_row[3] == 'TU Pelayanan Teknik')
or ($result_row[3] == 'SPV.Opdist & Pel.Teknik') or ($result_row[3] ==
'Spv. Opdist & Pel. Teknik'))

{ if(($result_row[4] == 'Kompetensi PDKB') or ($result_row[4] ==
'Kompetensi PDKB Ang.V') or ($result_row[4] == 'Kompetensi PDKB
Ang.IV') or ($result_row[4] == 'Kompetensi PDKB Ang.III') or
($result_row[4] == 'Kompetensi PDKB Ang.II') or ($result_row[4] ==
'Kompetensi PDKB Ang.I') or ($result_row[4] == 'K2/K3') or
($result_row[4] == 'K2/K3 Pelaksana') or ($result_row[4] == 'K2/K3
Pelaksana I') or ($result_row[4] == 'K2/K3 Pelaksana II') or
($result_row[4] == 'K2/K3 Pelaksana III') or ($result_row[4] == 'K2/K3
Pengawas') or ($result_row[4] == 'K2/K3 Pengawas I') or
($result_row[4] == 'K2/K3 Pengawas II') or ($result_row[4] == 'K2/K3
Pengawas III'))

{echo '<meta http-equiv="refresh" content=0;

```



```
url=help_lihat_tambah.php?id='.$this->id.'>;

$update_layak = new database();
$update_layak->connect_db();
$update_layak->select_db();

$update_layak->query="update db_usulanpekerjaan set
kelayakan_kompetensi = 'Layak' WHERE idpegawai = ".$this-
>idpegawai."";

$update_layak->run_query();

$update_layak4 = new database();
$update_layak4->connect_db();
$update_layak4->select_db();

$update_layak4->query="update db_usulanpekerjaan set
statusmanajerapj = 'Layak' WHERE idpegawai = ".$this->idpegawai."";

$update_layak4->run_query();

echo '<div class="zpFormContent"><fieldset>
<legend>Personal information</legend>
<label class="zpFormLabel">Id Pegawai :</label>
<b>'.$this->idpegawai.'</b><br /><br />
<label class="zpFormLabel">Nama Pegawai :</label>
<b>'.$this->nama.'</b><br />
</fieldset>';

echo '<fieldset>
<legend>Pegawai Layak dalam kompetensi</legend>
<font size="3" color="#33CC00" ><b> Pegawai memenuhi
syarat kelayakan pelatihan dan memenuhi syarat jabatan.<b></font>';

echo '</fieldset>'; echo '</div>';

}else{

echo '<meta http-equiv="refresh" content=0; url=
help_lihat_tambah.php?id='.$this->id.'>;

$update_layak = new database();
$update_layak->connect_db();
$update_layak->select_db();
```

```
$update_layak->query="update db_usulanpekerjaan set
set kelayakan_kompetensi='Tidak memenuhi syarat pelatihan' WHERE
idpegawai = ".$this->idpegawai.";";

$update_layak->run_query();

$update_layak4 = new database();
$update_layak4->connect_db();
$update_layak4->select_db();

$update_layak4->query="update db_usulanpekerjaan set
statusmanajerapj='tidak memenuhi syarat kompetensi' WHERE idpegawai =
".$this->idpegawai.";";

$update_layak4->run_query(); }

}else{

echo '<meta http-equiv="refresh"
content=0;url=help_lihat_tambah.php?id='.$this->id.';>';

$update_layak = new database();
$update_layak->connect_db();
$update_layak->select_db();

$update_layak->query="update db_usulanpekerjaan
set set kelayakan_kompetensi='Tidak memenuhi persyaratan jabatan'
WHERE idpegawai = ".$this->idpegawai.";";

$update_layak->run_query();

$update_layak4 = new database();
$update_layak4->connect_db();
$update_layak4->select_db();

$update_layak4->query="update db_usulanpekerjaan
set statusmanajerapj='tidak memenuhi persyaratan jabatan' WHERE
idpegawai = ".$this->idpegawai.";";

$update_layak4->run_query(); } }

}else{

echo '<meta http-equiv="refresh"
content=0;url=help_lihat_tambah.php?id='.$this->id.';>';

$update_layak = new database();

$update_layak->connect_db();

$update_layak->select_db();

$update_layak->query="update db_kesehatan_daftar_pas
set kelayakan='kondisi kurang baik' WHERE idpasien = ".$this->
```


menu *Lihat data* pada keterangan submenu Atur Kelayakan Pelaksana. Seperti terlihat pada gambar 5.51.

Setelah sub menu *Lihat Data* dipilih, maka akan muncul halaman seperti dibawah ini.

DATA PEGAWAI

Data Pegawai

• Tekan toboedit - hapus untuk menghapus atau mengubah nama pegawai.

Id Pegawai	Nama	Divisi	Nik	Golongan Sakit	Keterangan Sakit	Edit - Hapus
390023	Hafid Ali	own	own	Khusus	alhamdulillah,SEHA1	Edit - Hapus
39007	Wahyudiono	K2/K3 Pengawas II	UPJ Tumpang	Anak	SEhat wal afiat	Edit - Hapus
390022	Sumarsono Tasmun	K2/K3 Pelaksana II	UPJ Lawang	Khusus	Sehat walafiat	Edit - Hapus

Page 1 of 1 (3 rows)

Gambar 5.56 Halaman yang menampilkan Daftar Nama, yang akan di ubah kelayakannya
Sumber: Implementasi

Pada halaman ini akan muncul Daftar nama yang sebelumnya telah lulus uji kesehatan. Disamping masing-masing Nama kegiatan tersebut terdapat link *Edit - Hapus*. Jika hendak mengubah Kelayakan pegawai, pilih link *Edit* pada salah satu baris nama tersebut.

Tampak gambar dibawah ini, apabila ditekan link *Edit* tersebut.

Personal information

Id Pegawai 390022

Nama Pegawai Sumarsono Tasmun

NIP 6383337J

Jabatan TU Pemeliharaan Distribusi

Kursus K2/K3 Pelaksana II

Unit UPJ Lawang

Kesehatan

Kondisi Kesehatan Saudara/i **Sumarsono Tasmun** adalah sebagai berikut:

Golongan : Khusus

Keterangan Kesehatan : Sehat walafiat

Tanggal Periksa : 17-06-2007

Kelayakan

Saudara/i **Sumarsono Tasmun** Dinyatakan:

Kelayakan Saudara/i Layak

Sumarsono Tasmun Periksa Ulang

dalam Pekerjaan Tidak Layak

PDKB adalah

Gambar 5.57 Halaman yang menampilkan form untuk mengubah Kelayakan pegawai

Sumber: Implementasi

Akan muncul *fieldset* personal information, kesehatan, dan kelayakan. Dimana untuk menentukan kelayakan personal, pilih salah satu *option* didalam *fieldset* Kelayakan dan sebuah *textbox* tanggal pelaksanaan. Beserta sebuah tombol simpan, untuk menyimpan data, dan tombol hapus untuk mengosongkan isi form.

Algoritma pada tombol diatas, secara sederhana dapat ditunjukkan dalam *pseudocode* berikut ini :

Mengubah_usulanpekerja

```

$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

    if(kirim variable 'Ubah'){
        $update = new pilih class database();
        $update->Buka Koneksi dengan basis data();
        $update->pilih databse();
        $update->query = "update data2 dalam database db_usulanpekerjaan WHERE idpegawai";
        $update->jalankan query();

        if($update->query_result){
            echo "<br><b> Data Berhasil di Update</b> </br></br>";
            <a href='lihat_datapegawai.php'>lihat data</a> | <a href='../..>kembali ke halaman pengurus</a>;
        }else{
            echo "gagal Update data";
        }
    }

    include "forms_ubah.php";

End Mengubah_usulanpekerja
    
```

Gambar 5.58 Pseudocode pada tombol Simpan untuk Kebutuhan Fungsional Mengubah Nama Calon Pekerja
Sumber: Implementasi

5.2.3.5 Implementasi Kebutuhan Fungsional Menyusun Analisis Pekerjaan dan Pembagian Tugas

Untuk mengakses halaman ini user harus masuk sebagai Perancang. Kemudian akan muncul halaman, disertai beberapa sub menu, dimana salah satunya adalah sub menu Tambah Analisis pada keterangan submenu Analisis Pekerjaan.

Informasi
 Admin perancang bertindak dalam hal penyusunan jadwal operasi, Penentuan Calon Pekerja, dan Penentuan tindakan kerja pdkb.

Pilih Menu

ATUR PELAKSANA KERJA
 Tentukan Pelaksana

ANALISIS PEKERJAAN
 Tambah Analisis

Download the sample html file and css file

Selamat Datang - Perancang

Selamat Datang Perancang. Disini kami menyediakan Sistem Informasi Manajemen Kesehatan Keselamatan & Keamanan Kerja, khususnya di Divisi PDKB (Pekerjaan Dalam Keadaan Berbahaya). Guna dibuatnya situs ini ialah untuk mempermudah Pihak Perusahaan, dalam hal ini PT. PLN persero Area Pelayanan dan Jaringan Malang, dalam pengelolaan Sistem Kesehatan Keselamatan dan Keamanan Kerja.

- Link Jadwal untuk menyusun jadwal operasi yang akan dilaksanakan.
- Link Usulan Nama Pekerja
- Link Tindakan PDKB digunakan dalam menentukan tindakan PDKB yang akan dilaksanakan dalam penanganan Laporan Kerusakan atau perbaikan.

Gambar 5.59 Submenu Tambah Analisis pada Analisis Pekerjaan
Sumber: Implementasi

Setelah sub menu *tambah Analisis* dipilih, maka akan muncul halaman seperti gambar 5.61 dibawah ini.

ISI DATA KEGIATAN KERJA

- Tekan tombol *edit* - *hapus* untuk menghapus atau mengubah nama pegawai.

No Lap	Uraian Pekerjaan	Keterangan	Tindakan Penanganan
1	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	Tambah - Ubah - Hapus
2	Pemas Aresteser untuk penangkalan petir	volume 6 set	Tambah - Ubah - Hapus

Page 1 of 1 (2 rows)

Gambar 5.60 Halaman yang menampilkan Kegiatan, yang akan ditambahkan Analisis Pekerjaan.
Sumber: Implementasi

Pada halaman ini akan muncul Daftar Kegiatan. Disamping masing-masing Nama kegiatan tersebut terdapat link *Tambah - Ubah - Hapus*, yang masing-masing link berfungsi untuk menambah data Analisis, Mengubah Data Analisis, dan Menghapus Daftar Kegiatan. Jika hendak menambah Analisis Pekerjaan, pilih link tersebut pada salah satu baris kegiatan tersebut.

Tampak gambar dibawah ini, apabila ditekan link *tambah* tersebut.

SOP PDKB - TM NOMOR 06
ANALISIS PEKERJAAN DAN PEMBAGIAN TUGAS
(TAILGATE SESSION)

Personal information

ID Analisis

Nama Pengawas

Id Kegiatan PDKB

Id Pekerjaan

Jenis Pekerjaan : Perbaikan Gangguan SUTM Km Bencana Alam Angin

SOP No : 12

Metode : PENGGANTIAN JUMPER ATAS MENGGUNAKAN PARALEL GROUP DENGAN METODE BERJARAK

Nama Lokasi : UPJ Gondanglegi

ID Usulan

Identifikasi Hazard

Hazard *

Cara Mengatasi (Eliminate, Isolate, & minimise)

Nama Anggota : Sumarsono Tasmun

Tugas

klik tombol untuk menambahkan Data
 klik tombol untuk mengubah Data yang dipilih

Gambar 5.61 Halaman yang menampilkan form untuk menambah Kelayakan pegawai
Sumber: Implementasi

Akan muncul *fieldset* personal information, Identifikasi Hazard, dan Daftar Anggota. Dimana untuk menambah Analisis, Lengkapi *fieldset* Identifikasi Hazard, dan *textbox* tugas. Beserta sebuah tombol simpan, untuk menyimpan data, dan tombol hapus untuk mengosongkan isi form.

Algoritma pada tombol diatas, secara sederhana dapat ditunjukkan dalam *pseudocode* berikut ini :

Menambah_analisis

```

$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];
$this2->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

    if(kirim variable 'Simpan'){

$update_pdkb2 = new pilih class database();
$update_pdkb2->Buka Koneksi dengan basis data();
$update_pdkb2->pilih databse();
$update_pdkb2->query = "masukan data kedalam database db_pemberitahuan;";
$update_pdkb2->jalankan query();

$update_pdkb = new pilih class database();
$update_pdkb->Buka Koneksi dengan basis data();
$update_pdkb->pilih databse();
$update_pdkb->query = "masukan data kedalam database
db_analisispekerjaan;";
$update_pdkb->jalankan query();

$update_pdkb3 = new pilih class database();
$update_pdkb3->Buka Koneksi dengan basis data();
$update_pdkb3->pilih databse();
$update_pdkb3->query = "update data2 dalam database db_usulanpekerjaan
WHERE idusulan;";
$update_pdkb3->jalankan query();

if($update_pdkb->query_result){
echo "</br><b> Data Berhasil di Update</b> </br></br><a
href='lihat_datapdkb.php'>lihat data</a> | <a
href='.././perancang'>kembali ke halaman pengurus</a>";
}else{
echo "Id Analisis Sudah terpakai";}}

End Menambah_analisis

```

Gambar 5.62 Pseudocode pada tombol Simpan untuk Kebutuhan Fungsional Menambah Analisis Pekerjaan Dan Pembagian Tugas
Sumber: Implementasi

5.2.3.6 Implementasi Kebutuhan Fungsional Mengubah Analisis Pekerjaan dan Pembagian Tugas

Untuk mengakses halaman ini user harus masuk sebagai Perancang. Kemudian akan muncul halaman, disertai beberapa sub menu, dimana salah satunya adalah sub menu Tambah Analisis pada keterangan submenu Analisis Pekerjaan. Seperti terlihat pada gambar 5.58

Setelah sub menu *tambah Analisis* dipilih, maka akan muncul halaman seperti gambar 5.63 dibawah ini.

ISI DATA KEGIATAN KERJA

- Tekan *tomboledit - hapus* untuk menghapus atau mengubah nama pegawai.

No Lap	Uraian Pekerjaan	Keterangan	Tindakan Penanganan
1	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	Tambah - Ubah - Hapus
2	Pemas Arester untuk penangkaln petir	volume 6 set	Tambah - Ubah - Hapus

Page 1 of 1 (2 rows)

Gambar 5.63 Halaman yang menampilkan Kegiatan, yang akan diubah Analisis Pekerjaan.
Sumber: Implementasi

Pada halaman ini akan muncul Daftar Kegiatan. Disamping masing-masing Nama kegiatan tersebut terdapat link *Tambah - Ubah - Hapus*, yang masing-masing link berfungsi untuk menambah data Analisis, Mengubah Data Analisis, dan Menghapus Daftar Kegiatan. Jika hendak mengubah Analisis Pekerjaan, pilih link *Ubah* pada salah satu baris kegiatan tersebut.

Akan muncul halaman, seperti terlihat dalam gambar 5.64, disertai *fieldset* personal information, Identifikasi Hazard, dan Daftar Anggota. Dimana untuk mengubah Analisis, Lengkapi *fieldset* Identifikasi Hazard, dan *textbox* tugas. Beserta sebuah tombol ubah, untuk menyimpan data yang telah diubah, dan tombol hapus untuk mengosongkan isi form.

Algoritma pada tombol diatas, secara sederhana dapat ditunjukkan dalam *pseudocode* berikut ini :


```
Mengubah_analisis

$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];
$this2->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

if(kirim variable 'Ubah'){

$update_pdkb = new pilih class database();
$update_pdkb->Buka Koneksi dengan basis data();
$update_pdkb->pilih databse();
$update_pdkb->query = "update data2 dalam database db_analisispekerjaan WHERE
idkegiatanpdkb";
$update_pdkb->jalankan query();

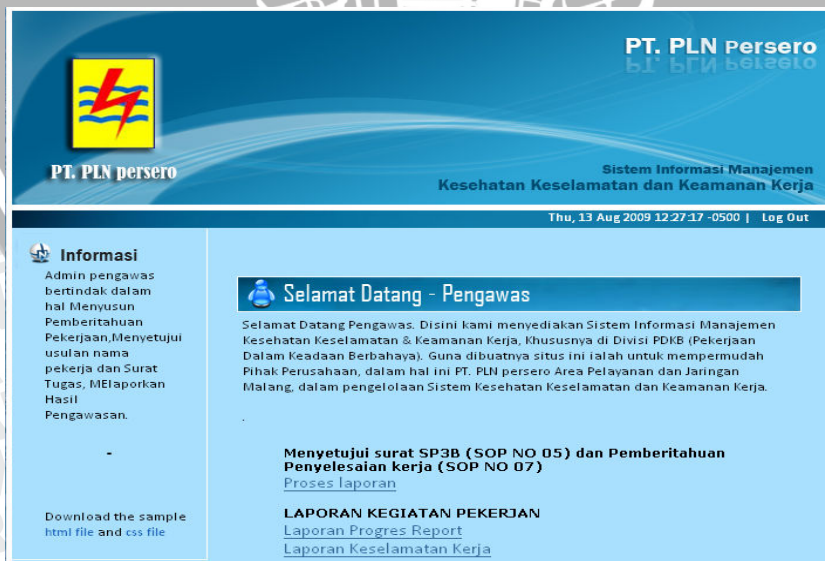
if($update_pdkb->query_result){
echo "</br><b> Data Berhasil di Update</b> </br></br>";
<a href='lihat_datapdkb.php'>lihat data</a> | <a
href='.././perancang'>kembali ke halaman pengurus</a>";
}else{
echo "Id Analisis Sudah terpakai";
}include "forms_ubah.php";}

End Mengubah_analisis
```

Gambar 5.64 Pseudocode pada tombol Ubah untuk Kebutuhan Fungsional Mengubah Analisis Pekerjaan Dan Pembagian Tugas
Sumber: Implementasi

5.2.3.7 Implementasi Kebutuhan Fungsional Membuat Surat Tugas

Untuk mengakses halaman ini user harus masuk sebagai Pengawas. Kemudian akan muncul halaman, disertai beberapa sub menu, dimana salah satunya adalah sub menu Proses Laporan.



Gambar 5.65 Submenu Menyetujui Surat Tugas pada Keterangan Submenu Proses Laporan

Sumber: Implementasi

Setelah sub menu *Proses Laporan* dipilih, maka akan muncul halaman seperti gambar 5.66 dibawah ini

◆ Tekan tombol *Menyetujui Surat Tugas* untuk melihat surat tugas dan mencetak


No Lap	Uraian Pekerjaan	Keterangan	Pengawas	Tindakan Penanganan	Penyelesaian Kerja
4	Penggantian trafo untuk program preman trafo	volume 31 unit	Abdullah Besari	Menyetujui Surat Tugas	Penyelesaian kerja
1	Perbaikan Gangg SUTM Km bencana alam angin	Volume 1,306 Kms	Abdullah Besari	Menyetujui Surat Tugas	Penyelesaian kerja

Page 1 of 1 (2 rows)

Gambar 5.66 Halaman yang menampilkan Kegiatan Pekerjaan, yang akan dicetak Surat tugasnya.

Sumber: Implementasi

Pada halaman ini akan muncul Daftar Kegiatan. Disamping masing-masing Nama kegiatan tersebut terdapat link *Menyetujui Surat Tugas*. Jika hendak menambah Data Surat Tugas, pilih link *Menyetujui Surat Tugas* pada salah satu baris kegiatan tersebut.



PT. PLN Persero
PLN PERSERO

PT. PLN Persero
PLN PERSERO

[Halaman Utama](#) [Halaman Pengurus](#) [Log Out](#)

DATA ANALISIS PEKERJAAN

**SOP PDKB - TM NOMOR 05
SURAT PENUNJUKAN PENGAWASAN PEKERJAAN
BERTEGANGAN (SP3B)**

No. 0003/PDKB-TM/2009 Tanggal : 28.03.2009

Kepala Operasi atau Wakilnya,
Memberikan Kewenangan Kepada Pengawas Pekerjaan : Hafid Ali
Pemegang Sertifikat Kewenangan :

UNTUK MELAKSANAKAN PENGAWASAN PEKERJAAN DALAM KEADAAN BERTEGANGAN PADA
INSTALASI BERIKUT INI :

JENIS PEKERJAAN YANG AKAN DILAKSANAKAN :
Perbaikan Gangguan SUTM Km Bencana Alam Angin

CARA OPERASI YANG DIPILIH OLEH PENGAWAS PEKERJAAN :
Lintaskan Kabel Grounding di Zsisi bangunan dengan jarak <300M

SYARAT OPERASI KHUSUS (Untuk Kategori kedua dan Ketiga)

HUBUNGAN KOMUNIKASI DENGAN LOKASI : Pesawat HT

KETERANGAN TAMBAHAN : Jarak EP Harap Diperhatikan


PT.PLN (PERSERO)
CABANG / APJ. UPJ Gondanglegi

KEPALA OPERASI

()

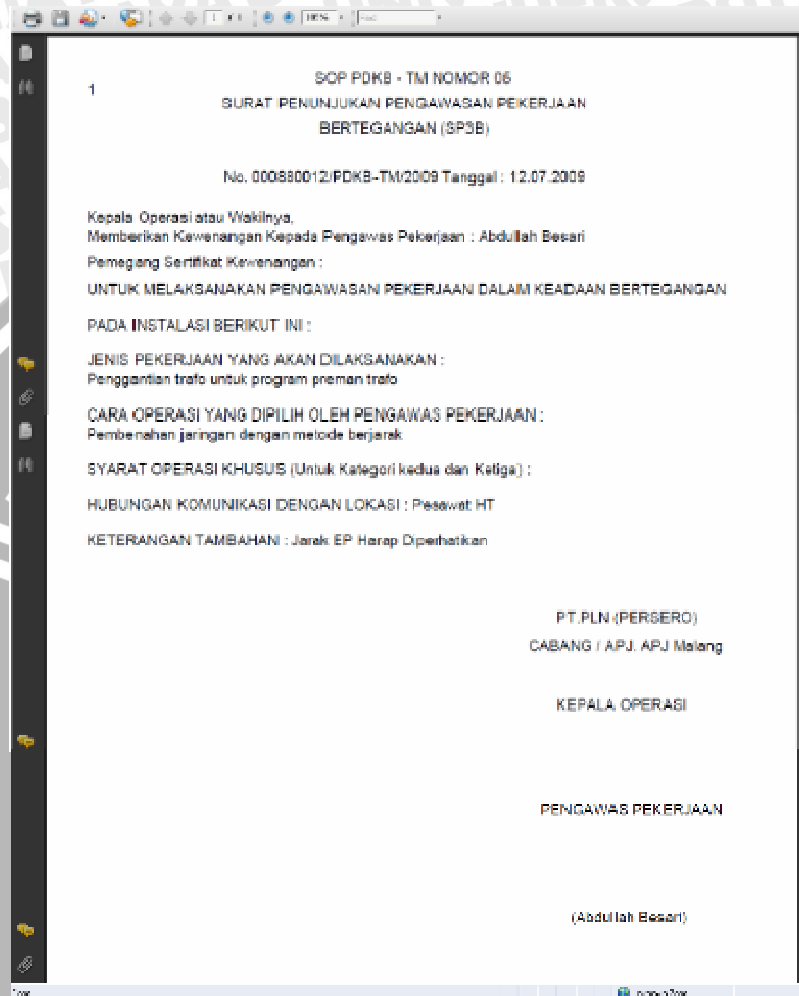
PENGAWAS PEKERJAAN

(Hafid Ali)

Klik tombol  untuk Mencetak dan menandatangani SP3B

Gambar 5.67 Halaman yang menampilkan Surat Tugas
Sumber: Implementasi

Akan muncul Surat Tugas yang telah disusun sebelumnya. Dibagian paling bawah, terdapat sebuah tombol Print, dimana fasilitas print menggunakan modul pdf.



Gambar 5.68 Halaman yang menampilkan surat tugas
Sumber: Implementasi

Sebelum muncul halaman ini, surat tugas terlebih dahulu telah disimpan kedalam basis data..

Algoritma pada tombol simpan, secara sederhana dapat ditunjukkan dalam *pseudocode* berikut ini :

Menambah_jadwal

```
$this->definisikan dengan data[pilihan kolom] = $_POST[nama field input];

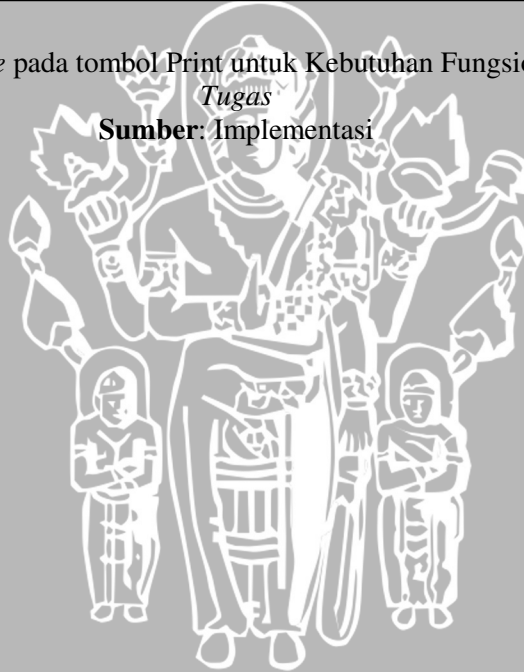
    if(kirim variable 'Simpan'){
        $update_suratTugas = new pilih class database();
        $update_suratTugas->Buka Koneksi dengan basis data();
        $update_suratTugas->pilih databse();
        $update_suratTugas->query = "masukan data kedalam database
        db_surattugas;";
        $update_suratTugas->jalankan query();

        if($update_suratTugas->query_result){
            echo "</br><b> Data Berhasil di Tambah</b> </br></br>";
            <a href='lihat_datapdkb.php'>lihat data</a> | <a
            href='../perancang'>kembali ke halaman pengurus</a>";
        }else{
            echo "Id Analisis Sudah terpakai";
        }
    }
}
```

End Menambah_jadwal

Gambar 5.69 Pseudocode pada tombol Print untuk Kebutuhan Fungsional Menambah Surat Tugas

Sumber: Implementasi



BAB VI PENGUJIAN

Pada Bab ini dilakukan proses pengujian pada Sistem Informasi Manajemen Kesehatan Keselamatan dan Keamanan Kerja. Proses pengujian yang akan dilakukan terbagi menjadi dua bagian, yaitu pengujian perangkat lunak, dan pengujian basis data.

Pengujian basis data meliputi pengujian rancangan basis data dan koneksi basis data. Pengujian rancangan basis data bertujuan untuk menguji apakah implementasi perancangan basis data telah sesuai dengan *Entity Relational Diagram*. Pengujian koneksi basis data dilakukan untuk memastikan bahwa komputer *client* dapat melakukan koneksi dengan *database* MySQL yang berada pada komputer *server*.

Pengujian Perangkat lunak terbagi menjadi dua tahap, yaitu pengujian unit, dan pengujian integrasi. Pada pengujian unit akan digunakan teknik pengujian *White Box (White Box Testing)*. Pada pengujian integrasi akan digunakan teknik pengujian *Black Box (Black Box Testing)*.

6.1 Pengujian Basis Data

Pengujian basis data meliputi dua macam pengujian rancangan basis data dan koneksi basis data.

6.1.1 Pengujian Rancangan Basis Data

Pengujian Rancangan basis data bertujuan untuk mengetahui kebenaran rancangan basis data yang telah dibuat. Pengujian perancangan basis data dilakukan dengan menggunakan perangkat lunak Sybase PowerDesigner 12.5

Nama Kasus Uji : Rancangan Basis Data
Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa tabel pada basis data sim_k3, hasil perancangan sesuai dengan tabel hasil implementasi.

Spesifikasi *Hardware* :

- Prosesor : Intel PentiumIV CPU 2.66GHz
- Memori (RAM) : 512 MB
- Hardisk : Maxtor 2F0L0F0, kapasitas 40 GB
- VGA Card : ATI Radeon X1050

Spesifikasi *Software* :

- Windows XP Profesional Service Pack 2
- MySQL 5.0.51b
- Sybase PowerDesigner 12.5

Prosedur Uji :

1. Sebuah *window* Command Prompt dijalankan dari:
Start | Run... | Open: cmd.exe
2. *Server* basis data MySQL dijalankan sebagai *service* dengan memberikan perintah:
C:\Documents and Settings\Administrator>net start mysql
3. Memasuki *SQL Shell* dengan perintah berikut:
C:\xampp\mysql\bin>mysql -u root -p sim_k3

Enter password:
4. Tabel-tabel yang terdapat pada basis data sim_k3 ditampilkan dengan menggunakan perintah SQL berikut:

```
mysql>show tables;
```

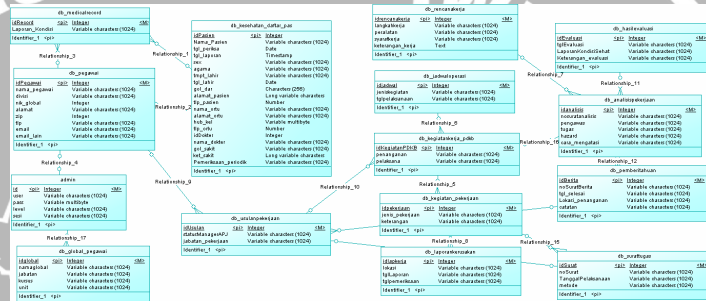
5. Membuka *software* Sybase PowerDesigner 12.5

dengan cara berikut:

Start | All Programs | Sybase | PowerDesigner 12.5 | PowerDesigner

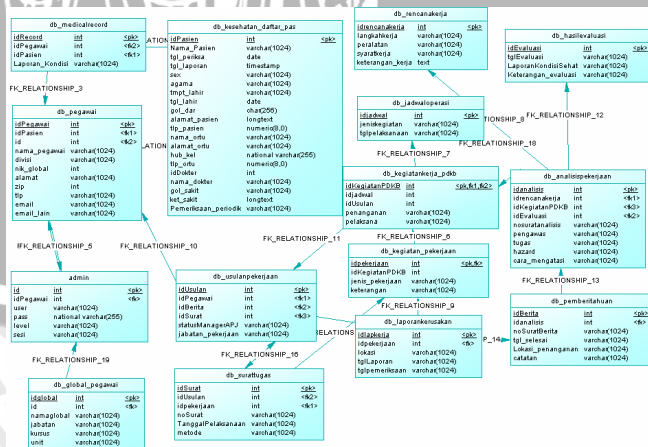
6. Menggambarkan *Entity Relationship Diagram* pada area kerja *Conceptual Data Model (CDM)*.

7. Memeriksa diagram ER tersebut dengan cara menekan tombol *Check Model* pada *toolbar*. Hasil pemeriksaan ini disebut dengan *CDM Object* yang ditunjukkan dalam Gambar 6.1.



Gambar 5.1 Conceptual Data Model Object
Sumber: Pengujian

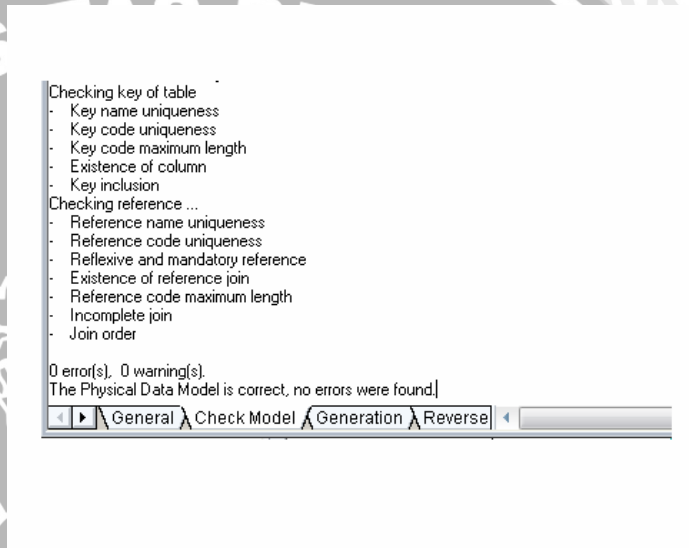
8. Untuk mengubah diagram ER dari *CDM Object* menjadi *Physical Data Model (PDM) Object*, dilakukan proses *generate* dengan menekan link *Generate Physical Data Model* pada *toolbar Tools*. *PDM Object* ditunjukkan dalam Gambar 6.2.



Gambar 5.2 *Physical Data Model*

Sumber: Pengujian

- Setelah berubah menjadi PDM *Object*, dilakukan pemeriksaan kembali dengan cara menekan tombol *Check Model* pada *toolbar*. Hasil Pemeriksaan ditunjukkan pada gambar 6.3.



Gambar 5.3 *Laporan Check Model Physical Data Model*

Sumber: Pengujian

- Setelah berubah menjadi PDM *Object*, dilakukan *generate* ke *database* MySQL dengan cara menekan *link Generate Database* pada *toolbar Tools*.



Gambar 5.4 *Laporan Check Model database MySQL*

Sumber: Pengujian

Hasil Pengujian :

Tabel pada basis data `sim_k3` berhasil di generate, dan sesuai dengan tabel hasil implementasi. Hasil generate bisa dilihat dengan perintah
`C:\xampp\mysql\bin>mysql -u root -p sim_k3` sesuai dengan prosedur uji, point 1 sampai dengan point 4.

6.1.2 Pengujian Koneksi Basis Data

Nama Kasus Uji : Kasus Uji Koneksi *Database*

Tujuan Pengujian : Pengujian ini dilakukan untuk memastikan bahwa komputer *client* dapat melakukan koneksi dengan *database* MySQL yang berada pada komputer *server*.

Spesifikasi *Hardware* : PC *Server* Aplikasi:

- Prosesor Intel ® Pentium 4 CPU 2,66 GHz, 256 MB RAM
- Sistem operasi Microsoft Windows XP Professional *Version* 2002 SP2
- Alamat IP pada perangkat Ethernet: 172.17.8.4

PC *Client*:

- Prosesor Intel ® Pentium 4 CPU 2,66 GHz, 256 MB RAM
- Sistem operasi Microsoft Windows XP Professional *Version* 2002 SP2
- Alamat IP pada perangkat Ethernet: 172.17.8.5
-

Prosedur Uji : PC *Server* Aplikasi:

Sebuah *window* Command Prompt dijalankan dari:

Start | Run... | Open: `cmd.exe`

Server database MySQL dijalankan sebagai *service* dengan memberikan perintah:

```
C:\>net start mysql
```

Aplikasi yang sedang berjalan dan koneksi yang sedang aktif ditampilkan dengan memberikan perintah:

```
C:\>netstat -an
```

PC *Client*:

1. Membuka aplikasi *web* sistem Informasi Manajemen Kesehatan keselamatan keamanan kerja <http://172.17.8.4:8080/smk3>
2. Melakukan proses registrasi, *login* sebagai *Member*, dan *login* sebagai *Administrator*.
3. Aplikasi yang sedang berjalan dan koneksi yang sedang aktif ditampilkan dengan memberikan perintah:

```
C:\>netstat -an
```

PC *Server* Aplikasi:

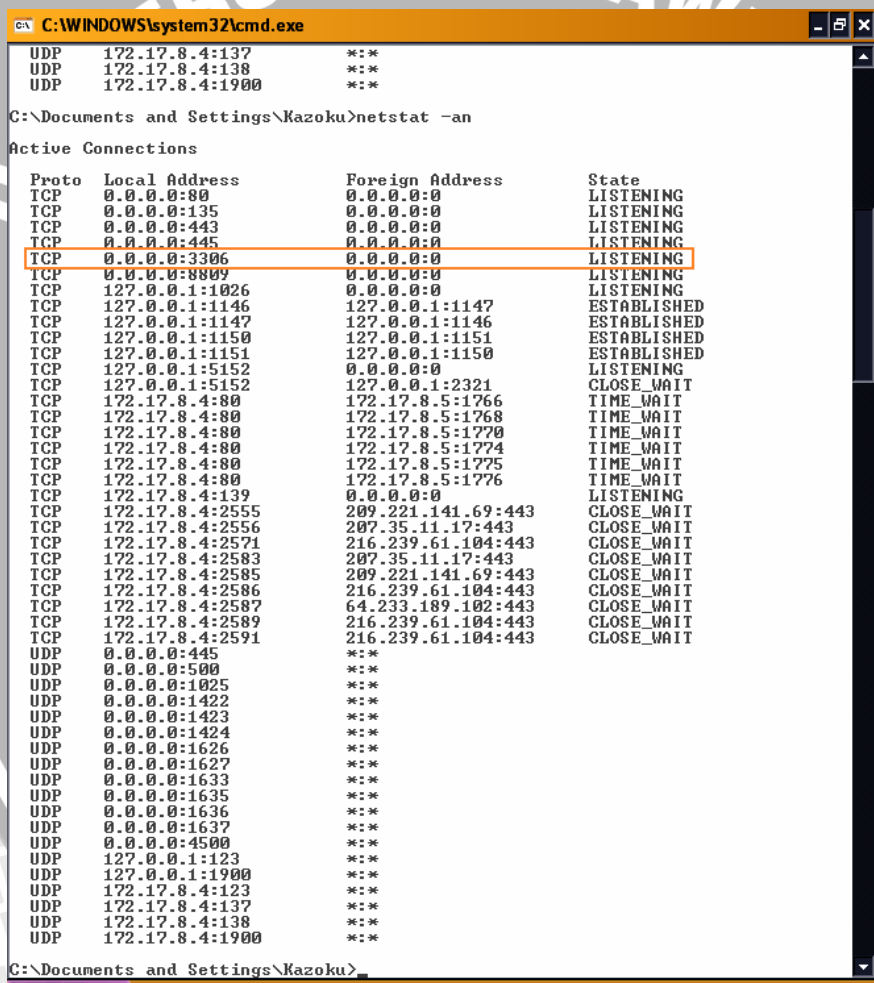
1. Aplikasi yang sedang berjalan dan koneksi yang sedang aktif ditampilkan kembali dengan memberikan perintah:

```
C:\>netstat -an
```

Hasil yang diharapkan : Komputer *client* dapat melakukan koneksi dengan *database* MySQL yang berada pada komputer *server*.

Hasil Pengujian : Komputer *client* dapat melakukan koneksi dengan *database* MySQL yang berada pada komputer *server*.

Hasil dari penggunaan perintah netstat -an pada komputer *server* aplikasi sebelum ada koneksi dengan komputer *client* ditunjukkan dalam Gambar 6.6. Perintah tersebut digunakan untuk menampilkan koneksi yang sedang aktif. Dari gambar tersebut terlihat bahwa *database server* MySQL (*mysqld-nt.exe*) memiliki kondisi (*state*) LISTENING pada alamat lokal 0.0.0.0:3306. Hal tersebut berarti bahwa *database server* MySQL telah siap untuk menerima sebuah koneksi *database* pada *port* TCP 3306.



Gambar 5.5 Koneksi yang sedang aktif pada komputer *server* sebelum aplikasi dijalankan pada komputer *client*

Sumber: Pengujian

perintah `netstat` –an pada komputer *client* setelah membuka aplikasi ini digunakan untuk menampilkan koneksi yang sedang aktif. Terdapat koneksi antara komputer *server* (alamat IP: 172.17.8.4) dengan komputer *client* (alamat IP: 172.17.8.5) ini ditunjukkan dengan adanya koneksi antara alamat IP 172.17.8.5:195 ke alamat IP 172.17.8.4:8080 dengan kondisi (*state*): ESTABLISHED.

6.1.3 Pengujian Waktu Akses Query

A. Tujuan

- Pengujian dilakukan untuk mengetahui waktu yang dibutuhkan untuk melakukan sebuah *query* pada basis data *sim_k3*.
- Pengujian dilakukan untuk mendapatkan perbandingan waktu *query* yang dilakukan terhadap jumlah data yang berbeda pada basis data *sim_k3*.

B. Prosedur Pengujian

PC *Client*:

- Mengakses `query_tes.php`. Halaman ini dibuat khusus untuk melakukan pengujian waktu akses *query* dengan menggunakan fungsi `microtime` pada `php`.
- Pengujian dilakukan pada tabel-tabel yang terdapat pada basis data *sim_k3*.
- Melakukan perhitungan untuk mendapatkan nilai rata-rata waktu akses *query* yang dilakukan terhadap jumlah data yang berbeda pada basis data *sim_k3*.

C. Hasil Pengujian dan Analisis

- Hasil pengujian waktu akses *query* terhadap tabel admin dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.6.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.538 ms	Query 1 = 0.652 ms
Query 2 = 0.291 ms	Query 2 = 0.293 ms
Query 3 = 0.301 ms	Query 3 = 0.269 ms
Query 4 = 0.265 ms	Query 4 = 0.263 ms
Query 5 = 0.268 ms	Query 5 = 0.261 ms
Query 6 = 0.291 ms	Query 6 = 0.259 ms
Query 7 = 0.267 ms	Query 7 = 0.259 ms
Query 8 = 0.271 ms	Query 8 = 0.259 ms
Query 9 = 0.268 ms	Query 9 = 0.261 ms
Query 10 = 0.263 ms	Query 10 = 0.261 ms
Rata - rata waktu query = 0.3023 ms	Rata - rata waktu query = 0.3037 ms

Gambar 5.6 Hasil pengujian waktu akses *query* terhadap tabel admin

Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel db_analisispekerjaan dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.7.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.521 ms	Query 1 = 0.664 ms
Query 2 = 0.294 ms	Query 2 = 0.295 ms
Query 3 = 0.271 ms	Query 3 = 0.27 ms
Query 4 = 0.265 ms	Query 4 = 0.265 ms
Query 5 = 0.264 ms	Query 5 = 0.271 ms
Query 6 = 0.291 ms	Query 6 = 0.262 ms
Query 7 = 0.268 ms	Query 7 = 0.262 ms
Query 8 = 0.265 ms	Query 8 = 0.262 ms
Query 9 = 0.263 ms	Query 9 = 0.26 ms
Query 10 = 0.261 ms	Query 10 = 0.26 ms
Rata - rata waktu query = 0.2963 ms	Rata - rata waktu query = 0.3071 ms

Gambar 5.7 Hasil pengujian waktu akses *query* terhadap tabel db_analisispekerjaan

Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel `db_global_pegawai` dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.8.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.57 ms	Query 1 = 0.569 ms
Query 2 = 0.348 ms	Query 2 = 0.351 ms
Query 3 = 0.323 ms	Query 3 = 0.326 ms
Query 4 = 0.321 ms	Query 4 = 0.322 ms
Query 5 = 0.325 ms	Query 5 = 0.327 ms
Query 6 = 0.319 ms	Query 6 = 0.318 ms
Query 7 = 0.319 ms	Query 7 = 0.321 ms
Query 8 = 0.321 ms	Query 8 = 0.32 ms
Query 9 = 0.319 ms	Query 9 = 0.32 ms
Query 10 = 0.319 ms	Query 10 = 0.32 ms
Rata - rata waktu query = 0.3484 ms	Rata - rata waktu query = 0.3494 ms

Gambar 5.8 Hasil pengujian waktu akses *query* terhadap tabel `db_global_pegawai`
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel `db_kegiatan_pekerjaan` dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.9.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.466 ms	Query 1 = 0.649 ms
Query 2 = 0.25 ms	Query 2 = 0.326 ms
Query 3 = 0.229 ms	Query 3 = 0.282 ms
Query 4 = 0.224 ms	Query 4 = 0.235 ms
Query 5 = 0.222 ms	Query 5 = 0.223 ms
Query 6 = 0.221 ms	Query 6 = 0.223 ms
Query 7 = 0.274 ms	Query 7 = 0.221 ms
Query 8 = 0.231 ms	Query 8 = 0.221 ms
Query 9 = 0.224 ms	Query 9 = 0.221 ms
Query 10 = 0.223 ms	Query 10 = 0.221 ms
Rata - rata waktu query = 0.2564 ms	Rata - rata waktu query = 0.2822 ms

Gambar 5.9 Hasil pengujian waktu akses *query* terhadap tabel `db_kegiatan_pekerjaan`
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel `db_kegiatankerja_pdkb` dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.10.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.466 ms	Query 1 = 0.649 ms
Query 2 = 0.25 ms	Query 2 = 0.326 ms
Query 3 = 0.229 ms	Query 3 = 0.282 ms
Query 4 = 0.224 ms	Query 4 = 0.235 ms
Query 5 = 0.222 ms	Query 5 = 0.223 ms
Query 6 = 0.221 ms	Query 6 = 0.223 ms
Query 7 = 0.274 ms	Query 7 = 0.221 ms
Query 8 = 0.231 ms	Query 8 = 0.221 ms
Query 9 = 0.224 ms	Query 9 = 0.221 ms
Query 10 = 0.223 ms	Query 10 = 0.221 ms
Rata - rata waktu query = 0.2564 ms	Rata - rata waktu query = 0.2822 ms

Gambar 5.10 Hasil pengujian waktu akses *query* terhadap tabel `db_kegiatankerja_pdkb`
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel `db_kesehatan_daftar_pas` dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.11.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.466 ms	Query 1 = 0.649 ms
Query 2 = 0.25 ms	Query 2 = 0.326 ms
Query 3 = 0.229 ms	Query 3 = 0.282 ms
Query 4 = 0.224 ms	Query 4 = 0.235 ms
Query 5 = 0.222 ms	Query 5 = 0.223 ms
Query 6 = 0.221 ms	Query 6 = 0.223 ms
Query 7 = 0.274 ms	Query 7 = 0.221 ms
Query 8 = 0.231 ms	Query 8 = 0.221 ms
Query 9 = 0.224 ms	Query 9 = 0.221 ms
Query 10 = 0.223 ms	Query 10 = 0.221 ms
Rata - rata waktu query = 0.2564 ms	Rata - rata waktu query = 0.2822 ms

Gambar 5.11 Hasil pengujian waktu akses *query* terhadap tabel `db_kesehatan_daftar_pas`
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel `db_laporankerusakan` dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.12.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.599 ms	Query 1 = 0.488 ms
Query 2 = 0.254 ms	Query 2 = 0.258 ms
Query 3 = 0.228 ms	Query 3 = 0.238 ms
Query 4 = 0.226 ms	Query 4 = 0.226 ms
Query 5 = 0.225 ms	Query 5 = 0.226 ms
Query 6 = 0.229 ms	Query 6 = 0.222 ms
Query 7 = 0.223 ms	Query 7 = 0.228 ms
Query 8 = 0.222 ms	Query 8 = 0.22 ms
Query 9 = 0.221 ms	Query 9 = 0.225 ms
Query 10 = 0.224 ms	Query 10 = 0.228 ms
Rata - rata waktu query = 0.2651 ms	Rata - rata waktu query = 0.2559 ms

Gambar 5.12 Hasil pengujian waktu akses *query* terhadap tabel `db_laporankerusakan`
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel `db_medicalrecord` dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.13.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.682 ms	Query 1 = 0.552 ms
Query 2 = 0.255 ms	Query 2 = 0.256 ms
Query 3 = 0.231 ms	Query 3 = 0.226 ms
Query 4 = 0.226 ms	Query 4 = 0.222 ms
Query 5 = 0.226 ms	Query 5 = 0.223 ms
Query 6 = 0.232 ms	Query 6 = 0.219 ms
Query 7 = 0.225 ms	Query 7 = 0.252 ms
Query 8 = 0.223 ms	Query 8 = 0.222 ms
Query 9 = 0.228 ms	Query 9 = 0.25 ms
Query 10 = 0.224 ms	Query 10 = 0.22 ms
Rata - rata waktu query = 0.2752 ms	Rata - rata waktu query = 0.2642 ms

Gambar 5.13 Hasil pengujian waktu akses *query* terhadap tabel `db_medicalrecord`
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel `db_pegawai` dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.14.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.601 ms	Query 1 = 0.579 ms
Query 2 = 0.347 ms	Query 2 = 0.343 ms
Query 3 = 0.326 ms	Query 3 = 0.327 ms
Query 4 = 0.319 ms	Query 4 = 0.314 ms
Query 5 = 0.326 ms	Query 5 = 0.321 ms
Query 6 = 0.316 ms	Query 6 = 0.313 ms
Query 7 = 0.317 ms	Query 7 = 0.314 ms
Query 8 = 0.318 ms	Query 8 = 0.312 ms
Query 9 = 0.317 ms	Query 9 = 0.312 ms
Query 10 = 0.316 ms	Query 10 = 0.313 ms
Rata - rata waktu query = 0.3503 ms	Rata - rata waktu query = 0.3448 ms

Gambar 5.14 Hasil pengujian waktu akses *query* terhadap tabel `db_pegawai`
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel `db_pemberitahuan` dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.15.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.481 ms	Query 1 = 0.617 ms
Query 2 = 0.258 ms	Query 2 = 0.258 ms
Query 3 = 0.234 ms	Query 3 = 0.238 ms
Query 4 = 0.228 ms	Query 4 = 0.232 ms
Query 5 = 0.225 ms	Query 5 = 0.242 ms
Query 6 = 0.225 ms	Query 6 = 0.27 ms
Query 7 = 0.254 ms	Query 7 = 0.232 ms
Query 8 = 0.233 ms	Query 8 = 0.229 ms
Query 9 = 0.227 ms	Query 9 = 0.232 ms
Query 10 = 0.225 ms	Query 10 = 0.226 ms
Rata - rata waktu query = 0.259 ms	Rata - rata waktu query = 0.2776 ms

Gambar 5.15 Hasil pengujian waktu akses *query* terhadap tabel `db_pemberitahuan`
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel *db_rencanakerja* dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.16.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.62 ms	Query 1 = 0.494 ms
Query 2 = 0.274 ms	Query 2 = 0.279 ms
Query 3 = 0.258 ms	Query 3 = 0.257 ms
Query 4 = 0.249 ms	Query 4 = 0.247 ms
Query 5 = 0.246 ms	Query 5 = 0.248 ms
Query 6 = 0.255 ms	Query 6 = 0.271 ms
Query 7 = 0.247 ms	Query 7 = 0.254 ms
Query 8 = 0.247 ms	Query 8 = 0.249 ms
Query 9 = 0.247 ms	Query 9 = 0.246 ms
Query 10 = 0.247 ms	Query 10 = 0.247 ms
Rata - rata waktu query = 0.289 ms	Rata - rata waktu query = 0.2792 ms

Gambar 5.16 Hasil pengujian waktu akses *query* terhadap tabel *db_rencanakerja*
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel *db_surattugas* dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.17.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.405 ms	Query 1 = 0.506 ms
Query 2 = 0.2 ms	Query 2 = 0.205 ms
Query 3 = 0.181 ms	Query 3 = 0.187 ms
Query 4 = 0.176 ms	Query 4 = 0.178 ms
Query 5 = 0.176 ms	Query 5 = 0.176 ms
Query 6 = 0.176 ms	Query 6 = 0.176 ms
Query 7 = 0.177 ms	Query 7 = 0.176 ms
Query 8 = 0.177 ms	Query 8 = 0.197 ms
Query 9 = 0.185 ms	Query 9 = 0.178 ms
Query 10 = 0.177 ms	Query 10 = 0.175 ms
Rata - rata waktu query = 0.203 ms	Rata - rata waktu query = 0.2154 ms

Gambar 5.17 Hasil pengujian waktu akses *query* terhadap tabel *db_surattugas*
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel *db_usulanpekerjaan* dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.18.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.609 ms	Query 1 = 0.478 ms
Query 2 = 0.265 ms	Query 2 = 0.262 ms
Query 3 = 0.243 ms	Query 3 = 0.242 ms
Query 4 = 0.238 ms	Query 4 = 0.237 ms
Query 5 = 0.236 ms	Query 5 = 0.238 ms
Query 6 = 0.263 ms	Query 6 = 0.236 ms
Query 7 = 0.239 ms	Query 7 = 0.265 ms
Query 8 = 0.235 ms	Query 8 = 0.237 ms
Query 9 = 0.235 ms	Query 9 = 0.235 ms
Query 10 = 0.235 ms	Query 10 = 0.236 ms
Rata - rata waktu query = 0.2798 ms	Rata - rata waktu query = 0.2666 ms

Gambar 5.18 Hasil pengujian waktu akses *query* terhadap tabel *db_usulanpekerjaan*
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel *hasilvaluasi* dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.19.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.987 ms	Query 1 = 0.486 ms
Query 2 = 0.221 ms	Query 2 = 0.197 ms
Query 3 = 0.186 ms	Query 3 = 0.181 ms
Query 4 = 0.193 ms	Query 4 = 0.178 ms
Query 5 = 0.179 ms	Query 5 = 0.178 ms
Query 6 = 0.18 ms	Query 6 = 0.176 ms
Query 7 = 0.176 ms	Query 7 = 0.176 ms
Query 8 = 0.177 ms	Query 8 = 0.2 ms
Query 9 = 0.177 ms	Query 9 = 0.18 ms
Query 10 = 0.175 ms	Query 10 = 0.178 ms
Rata - rata waktu query = 0.2651 ms	Rata - rata waktu query = 0.213 ms

Gambar 5.19 Hasil pengujian waktu akses *query* terhadap tabel *hasilvaluasi*
Sumber: Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel jadwaloperasi dengan jumlah data sebanyak 500 dan 1000 data *entry* ditunjukkan dalam Gambar 6.20.

Loop = 10	Loop = 10
Jumlah Data = 500	Jumlah Data = 1000
Query 1 = 0.584 ms	Query 1 = 0.474 ms
Query 2 = 0.238 ms	Query 2 = 0.241 ms
Query 3 = 0.215 ms	Query 3 = 0.219 ms
Query 4 = 0.209 ms	Query 4 = 0.215 ms
Query 5 = 0.207 ms	Query 5 = 0.211 ms
Query 6 = 0.209 ms	Query 6 = 0.216 ms
Query 7 = 0.22 ms	Query 7 = 0.239 ms
Query 8 = 0.209 ms	Query 8 = 0.217 ms
Query 9 = 0.208 ms	Query 9 = 0.219 ms
Query 10 = 0.208 ms	Query 10 = 0.211 ms
Rata - rata waktu query = 0.2507 ms	Rata - rata waktu query = 0.2462 ms

Gambar 5.20 Hasil pengujian waktu akses *query* terhadap tabel jadwaloperasi
Sumber: Pengujian

6.2 Pengujian Perangkat Lunak

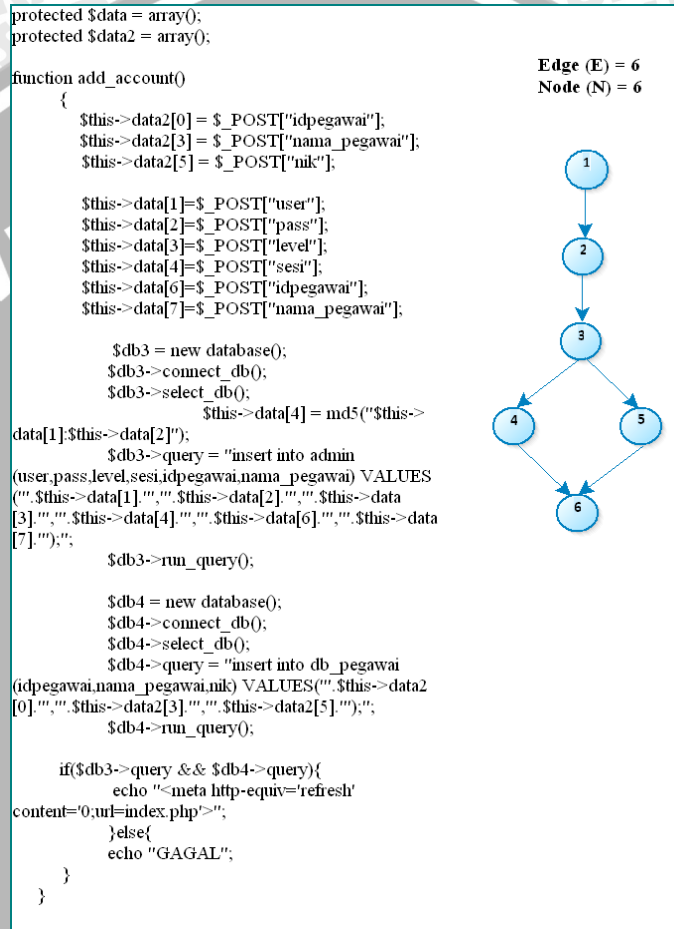
Pengujian perangkat lunak dibagi menjadi 2 tahap, yaitu pengujian unit dan pengujian integrasi.

6.2.1 Pengujian Unit

Dalam sistem yang dibangun dengan pemrograman berorientasi objek, pengujian unit diterapkan untuk suatu metode (operasi) dari suatu *class*. Pada pengujian unit ini, digunakan teknik pengujian *white box* (*white box testing*) dengan teknik *basis path testing*. Pada teknik *basis path testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan. Di dalam penulisan laporan skripsi ini, hanya dicantumkan hasil pengujian unit untuk algoritma dari beberapa metode (operasi) saja (tidak untuk keseluruhan metode).

6.2.1.1 Pengujian Unit untuk Operasi menambahAccount

Operasi menambahAccount merupakan salah satu metode yang terdapat dalam class admin. Pada Gambar 6.5 memperlihatkan proses pemodelan dalam *flow graph* pada operasi menambahAccount pada class admin.



Gambar 5.21 Flowgraph pada Operasi Menambah Account

Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.5 yang telah dilakukan terhadap operasi menambahAccount, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-4-6

Jalur 2 : 1-2-3-5-6

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.1 *Test case* untuk pengujian unit operasi manambahAccount

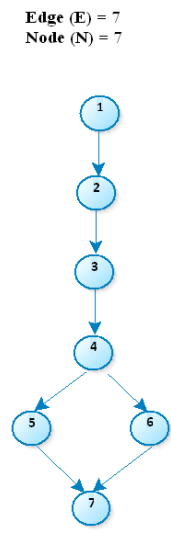
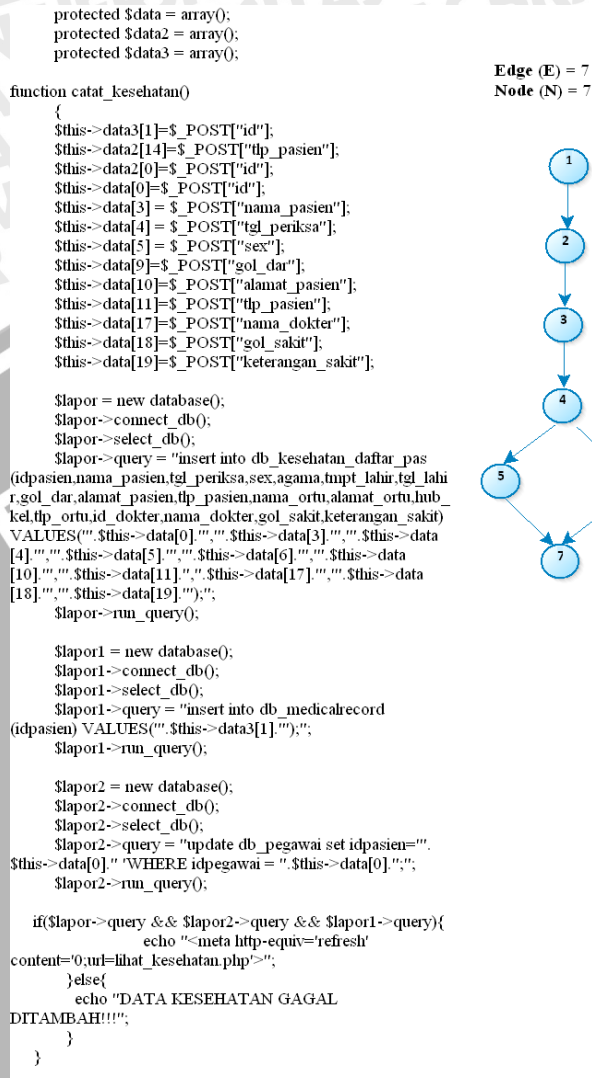
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil Admin.memasukanAccount (user,pass,level,sesi,idpegawai, nama_pegawai)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class</i> admin	dikembalikan ke <i>class</i> admin

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit operasi menambahAccount dari kelas admin, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.2 Pengujian Unit untuk Operasi menambahDataKesehatan

Operasi menambahDataKesehatan merupakan salah satu metode yang terdapat dalam class kesehatan. Pada Gambar 6.6 memperlihatkan proses pemodelan dalam *flow graph* pada operasi menambahDataKesehatan pada class kesehatan.



Gambar 5.22 Flowgraph pada Operasi menambahDataKesehatan
Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.6 yang telah dilakukan terhadap operasi menambahDataKesehatan, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 7 - 7 + 2 \\
 &= 2
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-4-5-7

Jalur 2 : 1-2-3-4-6-7

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.2 *Test case* untuk pengujian unit operasi `manambahDataKesehatan`

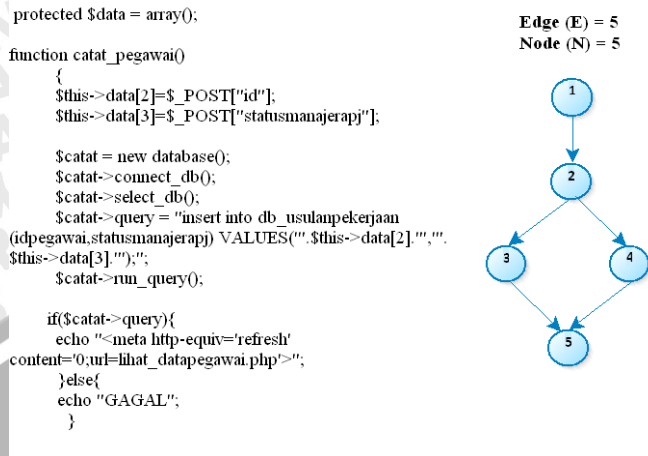
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil <code>Kesehatan.memasukanDataKesehatan(id,nama_pasien,tgl_periksa,sex,gol_dar, alamat_pasien,tlp_pasien, id_dokter, nama_dokter, gol_sakit, ket_sakit)</code>	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class</i> kesehatan	dikembalikan ke <i>class</i> kesehatan

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit operasi `menambahDataKesehatan` dari kelas kesehatan, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.3 Pengujian Unit untuk Operasi `menambahData_Kelayakan`

Operasi `Operasi menambahData_Kelayakan` merupakan salah satu metode yang terdapat dalam class pengelola. Pada Gambar 6.7 memperlihatkan proses pemodelan dalam *flow graph* pada operasi `menambahData_Kelayakan` pada class `Pengelola`.



Gambar 5.23 Flowgraph pada Operasi menambahData_Kelayakan
Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.7 yang telah dilakukan terhadap operasi menambahData_Kelayakan, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 5 - 5 + 2 \\
 &= 2
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-5

Jalur 2 : 1-2-4-5

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.3 Test case untuk pengujian unit operasi menambahData_Kelayakan

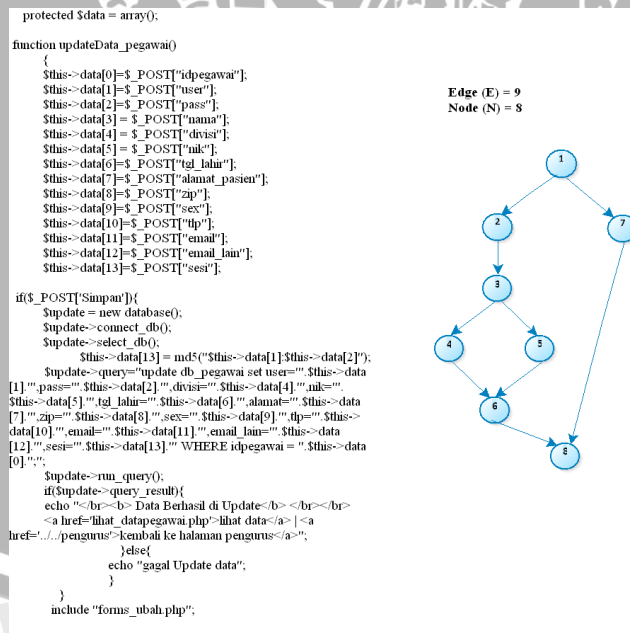
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil Pengelola. menambahData_Kelayakan (id,statusmanajerapj)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class</i> pengelola	dikembalikan ke <i>class</i> pengelola

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit operasi menambahData_Kelayakan dari kelas pengelola, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.4 Pengujian Unit untuk Operasi menambahData_Pegawai

Operasi Operasi menambahData_Pegawai merupakan salah satu metode yang terdapat dalam class pegawai. Pada Gambar 6.8 memperlihatkan proses pemodelan dalam *flow graph* pada operasi menambahData_Pegawai pada class Pegawai.



Gambar 5.24 Flowgraph pada Operasi menambahData_Pegawai

Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.8 yang telah dilakukan terhadap operasi menambahData_Pegawai, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 9 - 8 + 2 \\
 &= 3
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-4-6-8

Jalur 2 : 1-2-3-5-6-8

Jalur 3 : 1-7-8

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.4 *Test case* untuk pengujian unit operasi menambahData_Pegawai

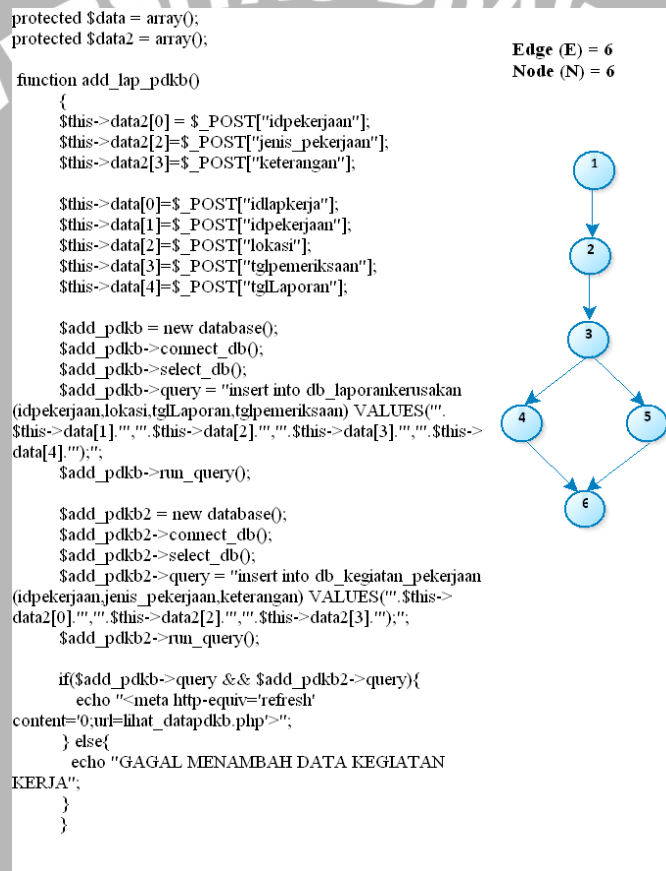
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil Pegawai. menambahData_Pegawai (idpegawai,nama,divisi,nik,tgl_lahir, alamat_pasien,zip,sex,tlp,email)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class</i> pegawai	dikembalikan ke <i>class</i> pegawai
3	Prosedure menyimpan tidak ditemukan	dikembalikan ke <i>form</i> tambah data	dikembalikan ke <i>form</i> tambah data

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit operasi menambahData_Pegawai dari kelas pegawai, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.5 Pengujian Unit untuk Operasi menambahLap_KegiatanKerja

Operasi menambahLap_KegiatanKerja merupakan salah satu metode yang terdapat dalam class pdkb. Pada Gambar 6.9 memperlihatkan proses pemodelan dalam *flow graph* pada operasi menambahLap_KegiatanKerja pada class pdkb.



Gambar 5.25 Flowgraph pada Operasi menambahLap_KegiatanKerja

Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.9 yang telah dilakukan terhadap operasi menambahLap_KegiatanKerja, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 1$

2, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 6 - 6 + 2 \\ &= 2 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-4-6

Jalur 2 : 1-2-3-5-6

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.5 *Test case* untuk pengujian unit operasi menambahLap_KegiatanKerja

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil <i>pdcb</i> . menambahLap_KegiatanKerja (<i>idLapkerja</i> , <i>idPekerjaan</i> , <i>Lokasi</i> , <i>tglLaporan</i> , <i>tglPemeriksaan</i>)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class pdcb</i>	dikembalikan ke <i>class pdcb</i>

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit *menambahLap_KegiatanKerja* dari kelas *pdcb*, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.6 Pengujian Unit untuk Operasi menambahJadwalOperasi

Operasi *menambahJadwalOperasi* merupakan salah satu metode yang terdapat dalam class *jadwal*. Pada Gambar 6.10 memperlihatkan proses pemodelan dalam *flow graph* pada operasi *menambahJadwalOperasi* pada class *jadwal*.


```

protected $data = array();
protected $data2 = array();

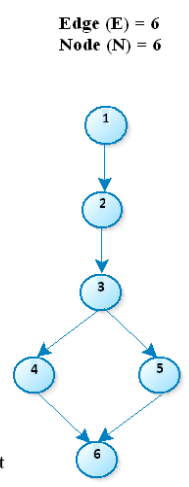
function add_lap_jadwal()
{
    $this->data2[1]=$_POST["idjadwal"];
    $this->data2[0]=$_POST["idkegiatanpdkb"];

    $this->data[0]=$_POST["idjadwal"];
    $this->data[1]=$_POST["idkegiatanpdkb"];
    $this->data[2]=$_POST["jeniskegiatan"];
    $this->data[3]=$_POST["tglpelaksanaan"];

    $add_jadwal = new database();
    $add_jadwal->connect_db();
    $add_jadwal->select_db();
    $add_jadwal->query = "insert into db_jadwaloperasi
(jeniskegiatan,tglpelaksanaan) VALUES('".$this->data[2]."', '".
$this->data[3]."',";
    $add_jadwal->run_query();

    $add_jadwal3 = new database();
    $add_jadwal3->connect_db();
    $add_jadwal3->select_db();
    $add_jadwal3->query="update db_kegiatankerja_pdkb set
idjadwal='".$this->data2[1]."' WHERE idkegiatanpdkb = ".
$this->data2[0]."',";
    $add_jadwal3->run_query();

    if($add_jadwal->query){
        echo "<meta http-equiv='refresh'
content=0,url=lihat_jadwal.php>";
    }else{
        echo "GAGAL";
    }
}
    
```



Gambar 5.26 Flowgraph pada Operasi menambahJadwalOperasi
Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.10 yang telah dilakukan terhadap operasi menambahJadwalOperasi, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-4-6

Jalur 2 : 1-2-3-5-6

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.6 *Test case* untuk pengujian unit operasi menambahJadwalOperasi

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil jadwal. menambahJadwalOperasi (idjadwal,idKegiatanpdkb, jenis_kegiatan,tglPelaksanaan)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class jadwal</i>	dikembalikan ke <i>class jadwal</i>

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit menambahJadwalOperasi dari kelas jadwal, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.7 Pengujian Unit untuk Operasi menambahLap_Tindakan

Operasi menambahLap_Tindakan merupakan salah satu metode yang terdapat dalam class pdkb. Pada Gambar 6.11 memperlihatkan proses pemodelan dalam *flow graph* pada operasi menambahLap_Tindakan pada class pengelola.


```
protected $data = array();
protected $data2 = array();

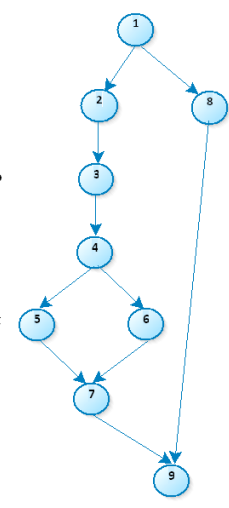
function add_lap_pdkb()
{
    $this->data[0] = $_POST["id"];
    $this->data[1] = $_POST["idkegiatanpdkb"];
    $this->data[0]=$_POST["idkegiatanpdkb"];
    $this->data[1]=$_POST["penanganan"];

    if($_POST['Simpan'])
    {
        $add_pdkb = new database();
        $add_pdkb->connect_db();
        $add_pdkb->select_db();
        $add_pdkb->query = "insert into db_kegiatankerja_pdkb
(idkegiatanpdkb,penanganan) VALUES('".$this->data[0]."', '".
$this->data[1]."',";
        $add_pdkb->run_query();

        $add_pdkb2 = new database();
        $add_pdkb2->connect_db();
        $add_pdkb2->select_db();
        $add_pdkb2->query="update db_kegiatan_pekerjaan set
idkegiatanpdkb='".$this->data[1]."' WHERE idpekerjaan = '".
$this->data[0]."',";
        $add_pdkb2->run_query();

        if($add_pdkb2->query_result){
            echo "<meta http-equiv='refresh'
content='0;url=help_langkahkerja.php'>";
        }else{
            echo "gagal Tambah data";
        }
    }else{
        echo "SIMPAN TIDAK BERHASIL";
    }
}
```

Edge (E) = 10
Node (N) = 9



Gambar 5.27 Flowgraph pada Operasi menambahLap_Tindakan
Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.11 yang telah dilakukan terhadap operasi menambahLap_Tindakan, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 10 - 9 + 2 \\
 &= 3
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

- Jalur 1 : 1-2-3-4-5-7-9
- Jalur 2 : 1-2-3-4-6-7-9

Jalur 3 : 1-8-9

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.7 *Test case* untuk pengujian unit operasi menambahLap_Tindakan

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil pengelola. menambahLap_Tindakan (idKegiatanpdkb, penanganan)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class</i> pengelola	dikembalikan ke <i>class</i> pengelola
3	Prosedure menyimpan tidak ditemukan	dikembalikan ke <i>form</i> tambah data	dikembalikan ke <i>form</i> tambah data

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit menambahLap_Tindakan dari kelas pengelola, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.8 Pengujian Unit untuk Operasi menambah_analisisKerja

Operasi menambah_analisisKerja merupakan salah satu metode yang terdapat dalam class analisis. Pada Gambar 6.12 memperlihatkan proses pemodelan dalam *flow graph* pada operasi menambah_analisisKerja pada class analisis.


```

protected $data = array();
protected $data2 = array();
protected $data3 = array();

function updateData_Analisis()
{
    $this->data[0]=$_POST["idanalisis"];
    $this->data[3]=$_POST["idkegiatanpdbk"];
    $this->data[5]=$_POST["pengawas"];
    $this->data[6]=$_POST["tugas"];
    $this->data[7]=$_POST["hazard"];
    $this->data[8]=$_POST["cara_mengatasi"];

    $this->data2[1]=$_POST["idanalisis"];
    $this->data3[4]=$_POST["idanalisis"];
    $this->data3[0]=$_POST["idusulan"];
}

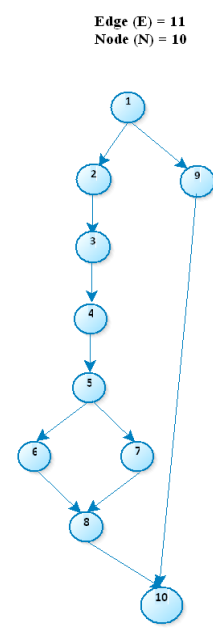
if($_POST["Simpan"]){
    $update_pdkb3 = new database();
    $update_pdkb3->connect_db();
    $update_pdkb3->select_db();
    $update_pdkb3->query="update db_usulanpekerjaan set
    idanalisis='".$this->data3[4]."' WHERE idusulan = '".$this->data3
    [0]."'";
    $update_pdkb3->run_query();

    $update_pdkb2 = new database();
    $update_pdkb2->connect_db();
    $update_pdkb2->select_db();
    $update_pdkb2->query="insert into db_pembentahan (idanalisis)
    VALUES('".$this->data2[1]."' )";
    $update_pdkb2->run_query();

    $update_pdkb = new database();
    $update_pdkb->connect_db();
    $update_pdkb->select_db();
    $update_pdkb->query="insert into db_analisispekerjaan
    (idanalisis,idkegiatanpdbk,pengawas,tugas,hazard,cara_mengatasi)
    VALUES('".$this->data[0]."' , '".$this->data[3]."' , '".$this->data
    [5]."' , '".$this->data[6]."' , '".$this->data[7]."' , '".$this->data[8]."' )";
    $update_pdkb->run_query();

    if($update_pdkb->query_result){
        echo "<br><b> Data Berhasil di Update</b><br><br>";
        <a href="lihat_datapdkb.php?lihat data"><a | <a
        href="...perancang">kembali ke halaman pengurus</a>";
    }else{
        echo "Id Analisis Sudah terpakai";
    }
    }
    include "forms_ubah.php";
}

```



Gambar 5.28 Flowgraph pada Operasi menambah_analisisKerja
Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.12 yang telah dilakukan terhadap operasi menambah_analisisKerja, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 11 - 10 + 2 \\
 &= 3
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-4-5-6-8-10

Jalur 2 : 1-2-3-4-5-7-8-10

Jalur 3 : 1-9-10

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.8 *Test case* untuk pengujian unit operasi menambah_analisisKerja

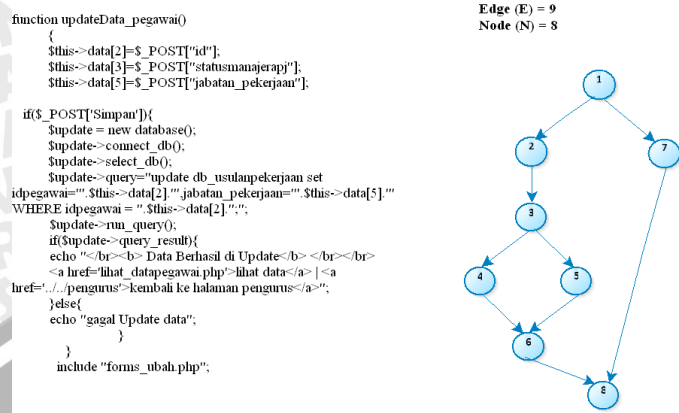
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil analisis. menambah_analisisKerja (idanalisis, idKegiatanpdkb, pengawas,tugas,hazard, cara_mengatasi)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class</i> analisis	dikembalikan ke <i>class</i> analisis
3	Prosedure menyimpan tidak ditemukan	dikembalikan ke <i>form</i> tambah data	dikembalikan ke <i>form</i> tambah data

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit menambah_analisisKerja dari kelas analisis, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.9 Pengujian Unit untuk Operasi menambah_Pelaksana

Operasi menambah_Pelaksana merupakan salah satu metode yang terdapat dalam class Perancang. Pada Gambar 6.13 memperlihatkan proses pemodelan dalam *flow graph* pada operasi menambah_analisisKerja pada class perancang.



Gambar 5.29 Flowgraph pada Operasi menambah_Pelaksana
Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.13 yang telah dilakukan terhadap operasi menambah_Pelaksana, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 9 - 8 + 2 \\
 &= 3
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

- Jalur 1 : 1-2-3-4-6-8
- Jalur 2 : 1-2-3-5-6-8
- Jalur 3 : 1-7-8

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.9 *Test case* untuk pengujian unit operasi menambah_Pelaksana

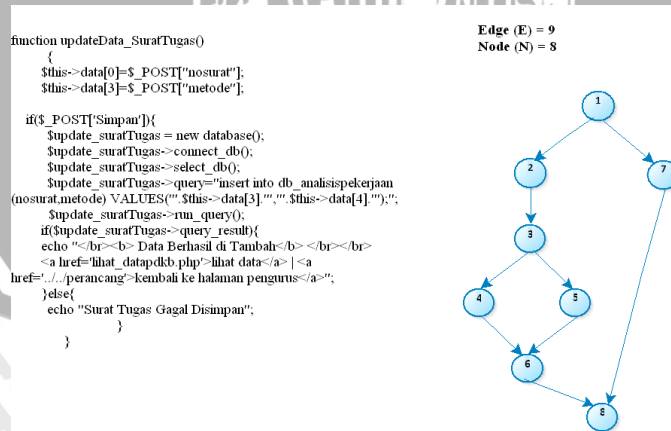
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil perancang, menambah_Pelaksana (id, jabatan_pekerjaan)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class</i> perancang	dikembalikan ke <i>class</i> perancang
3	Prosedure menyimpan tidak ditemukan	dikembalikan ke <i>form</i> tambah data	dikembalikan ke <i>form</i> tambah data

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit menambah_Pelaksana dari kelas analisis, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.1.10 Pengujian Unit untuk Operasi menambah_suratTugas

Operasi menambah_suratTugas merupakan salah satu metode yang terdapat dalam class Pengawas. Pada Gambar 6.13 memperlihatkan proses pemodelan dalam *flow graph* pada operasi menambah_suratTugas pada class pengawas.



Gambar 5.30 Flowgraph pada Operasi menambah_suratTugas

Sumber: Pengujian

Dari hasil pemodelan ke dalam *flow graph* pada Gambar 6.13 yang telah dilakukan terhadap operasi menambah_suratTugas, ditentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 9 - 8 + 2 \\
 &= 3
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu $V(G) = 2$ ditentukan dua basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-4-6-8

Jalur 2 : 1-2-3-5-6-8

Jalur 3 : 1-7-8

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah seperti pada Tabel 6.1.

Tabel 6.10 *Test case* untuk pengujian unit operasi menambah_suratTugas

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Memanggil pengawas. menambah_suratTugas (noSurat, metode)	Sistem Menampilkan Kembali Berupa String	Sistem Menampilkan Kembali Berupa String
2	Koneksi Tidak berhasil (database tidak ditemukan, table tidak ditemukan)	dikembalikan ke <i>class</i> pengawas	dikembalikan ke <i>class</i> pengawas
3	Prosedure menyimpan tidak ditemukan	dikembalikan ke <i>class</i> pengawas	dikembalikan ke <i>class</i> pengawas

Sumber : Pengujian

Dari hasil pengujian unit yang dilakukan pada unit menambah_suratTugas dari kelas pengawas, dapat diketahui bahwa unit ini dapat bekerja dengan baik karena dapat memberikan keluaran sesuai dengan harapan.

6.2.2 Pengujian Integrasi

Pengujian Integrasi diterapkan pada kebutuhan sistem yang mengintegrasikan fungsionalitas dari beberapa class untuk memenuhi suatu kebutuhan sistem. Pada pengujian Integrasi yang dijadikan objek uji adalah *Use-case* yang merupakan hasil dari analisis kebutuhan sistem.

Pengujian Integrasi didahului oleh pengujian pada unit-unit yang akan diintegrasikan. Pengujian integrasi ini menggunakan metode pengujian *Black box Testing*, karena pengujian ini lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan *Use-case*.

Penulisan Laporan Skripsi ini hanya dicantumkan hasil pengujian untuk beberapa *Use-case* saja.

6.2.2.1 Pengujian Integrasi untuk *Use-case* Menyusun Data Pegawai

Dalam memenuhi kebutuhan yang terdapat pada *Use-case* Menyusun Data Pegawai Layak diintegrasikan beberapa class yaitu class pengurus, dan class pengelola. Metode-metode yang dilibat untuk menjawab kebutuhan dari *Use-case* Menyusun Data Pegawai adalah memasukanDataKesehatan,memasukanData_Kelayakan, memasukanData_Pegawai.

Tujuan pengujian integrasi ini adalah menguji apakah data yang dimasukan dapat tersimpan dengan baik pada basis data.

Kasus uji untuk pengujian *Use-case* Menyusun Data Pegawai ditampilkan pada tabel dibawah ini.

Tabel 6.11 Kasus Uji untuk pengujian integrasi pada *Use-case* Menyusun Data Pegawai [Pengujian]

Kasus Uji	:	Menjalankan prosedur untuk Menyusun Data pegawai melalui aplikasi Sistem Informasi Manajemen Kesehatan, Keselamatan, dan Keamanan Kerja.	
Prosedur	:	1	Aktor Login Sebagai pengurus
		2	Aktor memilih sub menu <i>Tambah Data</i> pada Menu Data Kesehatan
		3	Aktor memilih salah satu tombol <i>tambah</i> disamping daftar nama yang ditampilkan
		4	Aktor memasukan data kesehatan pada form kesehatan
		5	Aktor Logout sebagai pengurus, dan login sebagai Pengelola
		6	Aktor memilih sub menu <i>Tambah Data</i> pada Menu Atur Kelayakan Pelaksana
		7	Aktor memilih salah satu tombol <i>tambah</i> disamping daftar nama yang ditampilkan
		8	Aktor menentukan kelayakan pegawai
		9	Aktor Logout sebagai Pengelola, dan login sebagai pengurus
		10	Aktor memilih sub menu <i>Tambah Data</i> pada Menu Data Pegawai
		11	Aktor memilih salah satu tombol <i>tambah</i> disamping daftar nama yang ditampilkan
		12	Aktor menambahkan data pegawai
Hasil yang diharapkan	:	Sistem dapat menyimpan Data pegawai yang telah dibuat kedalam basis data dengan baik	

Sumber : Pengujian

Hasil yang diperoleh dari hasil pengujian melalui kasus uji pada table diatas ditampilkan pada table dibawah ini.

Tabel 6.12 Hasil pengujian ntegrasi pada *Use-case* Menyusun Data Pegawai [Pengujian]

Hasil yang diharapkan	: Ssitem dapat menyimpan Data Pegawai yang telah dibuat kedalam basis data dengan baik																																																																													
Hasil yang didapatkan	<p>: Sistem dapat menyimpan DataKesehatan kedalam basis data</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="background-color: #d9e1f2;">Id</th> <th style="background-color: #d9e1f2;">Nama</th> <th style="background-color: #d9e1f2;">Tanggal Lahir</th> <th style="background-color: #d9e1f2;">Sex</th> <th style="background-color: #d9e1f2;">Golongan Sakit</th> <th style="background-color: #d9e1f2;">Keterangan</th> <th style="background-color: #d9e1f2;">Edit - Hapus</th> </tr> </thead> <tbody> <tr> <td>390022</td> <td>Hafid Ali</td> <td>09-08-2008</td> <td>Laki-lak</td> <td>Khusus</td> <td>alhamdulillah,SEHA1</td> <td>Edit - Hapus</td> </tr> <tr> <td>390010</td> <td>Priadi Antoro</td> <td>23-08-2007</td> <td>Laki-lak</td> <td>Paru-paru</td> <td>BAIK</td> <td>Edit - Hapus</td> </tr> <tr> <td>390007</td> <td>Wahyudiono</td> <td>11-12-2008</td> <td>Laki-lak</td> <td>Anak</td> <td>SEhat wal afiat</td> <td>Edit - Hapus</td> </tr> <tr> <td>390002</td> <td>Nanang Yudianto</td> <td>25-04-2007</td> <td>Laki-lak</td> <td>Penyakit Dalam</td> <td>Baek2 saja</td> <td>Edit - Hapus</td> </tr> <tr> <td>390022</td> <td>Sumarsono Tasmun</td> <td>17-06-2007</td> <td>Laki-lak</td> <td>Khusus</td> <td>Sehat walafiat</td> <td>Edit - Hapus</td> </tr> </tbody> </table> <p>Sistem dapat menyimpan DataKelayakan kedalam basis data</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="background-color: #d9e1f2;">Id Pegawai</th> <th style="background-color: #d9e1f2;">Nama</th> <th style="background-color: #d9e1f2;">Divisi</th> <th style="background-color: #d9e1f2;">Nik</th> <th style="background-color: #d9e1f2;">Golongan Sakit</th> <th style="background-color: #d9e1f2;">Keterangan Sakit</th> <th style="background-color: #d9e1f2;">Edit - Hapus</th> </tr> </thead> <tbody> <tr> <td>390023</td> <td>Hafid Ali</td> <td>own</td> <td>own</td> <td>Khusus</td> <td>alhamdulillah,SEHA1</td> <td>Edit - Hapus</td> </tr> <tr> <td>390007</td> <td>Wahyudiono</td> <td>K2/K3 Pengawas II</td> <td>UPJ Tumpang</td> <td>Anak</td> <td>SEhat wal afiat</td> <td>Edit - Hapus</td> </tr> <tr> <td>390022</td> <td>Sumarsono Tasmun</td> <td>K2/K3 Pelaksana II</td> <td>UPJ Lawang</td> <td>Khusus</td> <td>Sehat walafiat</td> <td>Edit - Hapus</td> </tr> <tr> <td>390010</td> <td>Priadi Antoro</td> <td>Kompetensi PDKB</td> <td>APJ Mlg</td> <td>Paru-paru</td> <td>BAIK</td> <td>Edit - Hapus</td> </tr> </tbody> </table> <p>Sistem dapat menyimpan Data Pegawai kedalam basis data</p> <p style="text-align: center;">Data Berhasil di Update</p> <p style="text-align: center;">lihat data kembali ke halaman pengurus</p>	Id	Nama	Tanggal Lahir	Sex	Golongan Sakit	Keterangan	Edit - Hapus	390022	Hafid Ali	09-08-2008	Laki-lak	Khusus	alhamdulillah,SEHA1	Edit - Hapus	390010	Priadi Antoro	23-08-2007	Laki-lak	Paru-paru	BAIK	Edit - Hapus	390007	Wahyudiono	11-12-2008	Laki-lak	Anak	SEhat wal afiat	Edit - Hapus	390002	Nanang Yudianto	25-04-2007	Laki-lak	Penyakit Dalam	Baek2 saja	Edit - Hapus	390022	Sumarsono Tasmun	17-06-2007	Laki-lak	Khusus	Sehat walafiat	Edit - Hapus	Id Pegawai	Nama	Divisi	Nik	Golongan Sakit	Keterangan Sakit	Edit - Hapus	390023	Hafid Ali	own	own	Khusus	alhamdulillah,SEHA1	Edit - Hapus	390007	Wahyudiono	K2/K3 Pengawas II	UPJ Tumpang	Anak	SEhat wal afiat	Edit - Hapus	390022	Sumarsono Tasmun	K2/K3 Pelaksana II	UPJ Lawang	Khusus	Sehat walafiat	Edit - Hapus	390010	Priadi Antoro	Kompetensi PDKB	APJ Mlg	Paru-paru	BAIK	Edit - Hapus
Id	Nama	Tanggal Lahir	Sex	Golongan Sakit	Keterangan	Edit - Hapus																																																																								
390022	Hafid Ali	09-08-2008	Laki-lak	Khusus	alhamdulillah,SEHA1	Edit - Hapus																																																																								
390010	Priadi Antoro	23-08-2007	Laki-lak	Paru-paru	BAIK	Edit - Hapus																																																																								
390007	Wahyudiono	11-12-2008	Laki-lak	Anak	SEhat wal afiat	Edit - Hapus																																																																								
390002	Nanang Yudianto	25-04-2007	Laki-lak	Penyakit Dalam	Baek2 saja	Edit - Hapus																																																																								
390022	Sumarsono Tasmun	17-06-2007	Laki-lak	Khusus	Sehat walafiat	Edit - Hapus																																																																								
Id Pegawai	Nama	Divisi	Nik	Golongan Sakit	Keterangan Sakit	Edit - Hapus																																																																								
390023	Hafid Ali	own	own	Khusus	alhamdulillah,SEHA1	Edit - Hapus																																																																								
390007	Wahyudiono	K2/K3 Pengawas II	UPJ Tumpang	Anak	SEhat wal afiat	Edit - Hapus																																																																								
390022	Sumarsono Tasmun	K2/K3 Pelaksana II	UPJ Lawang	Khusus	Sehat walafiat	Edit - Hapus																																																																								
390010	Priadi Antoro	Kompetensi PDKB	APJ Mlg	Paru-paru	BAIK	Edit - Hapus																																																																								

Sumber : Pengujian

Dari hasil pengujian integrasi pada *Use-case* Menyusun Data Pegawai diketahui sistem dapat menyimpan data pegawai yang telah dibuat kedalam basis data.

6.2.2.2 Pengujian Integrasi untuk *Use-case* Penanganan Kegiatan Kerja PDKB

Dalam memenuhi kebutuhan yang terdapat pada *Use-case* Penanganan Kegiatan Kerja PDKB diintegrasikan beberapa class yaitu class pengurus, dan class pengelola. Metode-metode yang dilibat untuk menjawab kebutuhan dari *Use-case* Penanganan Kegiatan Kerja PDKB adalah memasukanLap_kegiatanKerja, memasukanLap_Tindakan, memasukanLap_langkahKerja.

Tujuan pengujian integrasi ini adalah menguji apakah data yang dimasukan dapat tersimpan dengan baik pada basis data.

Kasus uji untuk pengujian *Use-case* Penanganan Kegiatan Kerja PDKB ditampilkan pada tabel dibawah ini.

Tabel 6.13 Kasus Uji untuk pengujian integrasi pada *Use-case* Penanganan Kegiatan Kerja PDKB [Pengujian]

Kasus Uji	:	Menjalankan prosedur untuk Penanganan Kegiatan Kerja PDKB melalui aplikasi Sistem Informasi Manajemen Kesehatan, Keselamatan, dan Keamanan Kerja.	
Prosedur	:	1	Aktor Login Sebagai pengurus
		2	Aktor memilih sub menu <i>Tambah Data</i> pada Menu Kerja PDKB
		3	Aktor memasukan laporan data kerusakan/perawatan pada form.
		4	Aktor Logout sebagai pengurus, dan login sebagai Pengelola
		5	Aktor memilih sub menu <i>Tambah Data</i> pada Menu Tindakan Kerja PDKB
		6	Aktor memasukan Data Tindakan PDKB
		7	Akan muncul daftar Uraian pekerjaan
		8	Aktor memilih salah satu tombol <i>tambah</i> disamping daftar uraian pekerjaan yang ditampilkan
		9	Aktor memasukan Data Langkah Kerja
Hasil yang diharapkan	:	Sistem dapat menyimpan Data Kegiatan Kerja yang telah dibuat kedalam basis data dengan baik	

Sumber : Pengujian

Hasil yang diperoleh dari hasil pengujian melalui kasus uji pada table diatas ditampilkan pada table dibawah ini.

Tabel 6.14 Hasil pengujian ntegrasi pada *Use-case* Penanganan Kegiatan Kerja PDKB [Pengujian]

<p>Hasil yang diharapkan</p>	<p>: Sistem dapat menyimpan Kegiatan kerja yang telah dibuat kedalam basis data dengan baik</p>																																																									
<p>Hasil yang didapatkan</p>	<p>: Sistem dapat menyimpan Data Laporan kerusakan/perawatan (Kegiatan kerja) kedalam basis data</p> <div data-bbox="711 441 1339 787" style="border: 1px solid black; padding: 5px;"> <p>• Tekan toboleedit - hapus untuk menghapus atau mengubah nama pegawai.</p> <table border="1"> <thead> <tr> <th>No Lap</th> <th>Lokasi</th> <th>Uraian Pekerjaan</th> <th>Keterangan</th> <th>Tanggal Laporan</th> <th>Edit - Hapus</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>UPJ Gondanglegi</td> <td>Perbaikan Gangguan SUTM Km Bencana Alam Angin</td> <td>Volume 1,306 Kms</td> <td>01.11.2008</td> <td>Edit - Hapus</td> </tr> <tr> <td>2</td> <td>UPJ Singosari</td> <td>Pemas Aresrer untuk penangkalan petir</td> <td>volume 6 set</td> <td>11.11.2008</td> <td>Edit - Hapus</td> </tr> <tr> <td>3</td> <td>APJ Malang</td> <td>Penggantian Trafo Untuk Program Premian Trafo</td> <td>volume 31 Unit</td> <td>22.05.2009</td> <td>Edit - Hapus</td> </tr> <tr> <td>4</td> <td>UPJ Nglantang</td> <td>Penambahan Grounding TR U/perbaikan teg ujung</td> <td>Volume 22 Set</td> <td>01.01.2009</td> <td>Edit - Hapus</td> </tr> <tr> <td>5</td> <td>UPJ Kepanjen</td> <td>Pengadaan Kwh meter 3phasa</td> <td>Volume 50 buah</td> <td>28.03.2009</td> <td>Edit - Hapus</td> </tr> <tr> <td>6</td> <td>APJ Malang</td> <td>Gangguan Isolator</td> <td>Volume 6 buah</td> <td>06.04.2008</td> <td>Edit - Hapus</td> </tr> </tbody> </table> <p>Page 1 of 1 (6 rows)</p> </div> <p>Sistem dapat menyimpan Data Tindakan Kerja kedalam basis data</p> <div data-bbox="722 919 1328 1255" style="border: 1px solid black; padding: 5px;"> <p>ISI DATA KEGIATAN KERJA</p> <p>Data Telah Ditambahkan</p> <p>Langkah berikutnya adalah menambahkan Langkah & Syarat Kerja : Tambahkan Langkah & Syarat Kerja kembali ke halaman pengurus</p> <div data-bbox="743 1050 1307 1228" style="border: 1px solid black; padding: 5px;"> <p>• Tekan toboleedit - hapus untuk menghapus atau mengubah nama pegawai.</p> <table border="1"> <thead> <tr> <th>SOP</th> <th>Uraian</th> <th>Tambah Syarat & Langkah kerja</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>Tambah - Hapus</td> </tr> <tr> <td>99</td> <td>dddddddddddddd</td> <td>Tambah - Hapus</td> </tr> <tr> <td>20</td> <td>PEMASANGAN JUMPER TRANSFORMATOR 3 PHASA PADA DUA TIANG/PORTAL DENGAN METODE BERJARAK</td> <td>Tambah - Hapus</td> </tr> <tr> <td>33</td> <td>PENGADAAN ISOLATOR PENEGANGAN DENGAN METODE BERJARAK</td> <td>Tambah - Hapus</td> </tr> </tbody> </table> <p>Page << < 1,2 of 2 (14 rows)</p> </div> </div> <p>Sistem dapat menyimpan DataLangkah Kerja kedalam basis data</p> <p>Data Berhasil di Update</p> <p>lihat data kembali ke halaman pengurus</p>	No Lap	Lokasi	Uraian Pekerjaan	Keterangan	Tanggal Laporan	Edit - Hapus	1	UPJ Gondanglegi	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	01.11.2008	Edit - Hapus	2	UPJ Singosari	Pemas Aresrer untuk penangkalan petir	volume 6 set	11.11.2008	Edit - Hapus	3	APJ Malang	Penggantian Trafo Untuk Program Premian Trafo	volume 31 Unit	22.05.2009	Edit - Hapus	4	UPJ Nglantang	Penambahan Grounding TR U/perbaikan teg ujung	Volume 22 Set	01.01.2009	Edit - Hapus	5	UPJ Kepanjen	Pengadaan Kwh meter 3phasa	Volume 50 buah	28.03.2009	Edit - Hapus	6	APJ Malang	Gangguan Isolator	Volume 6 buah	06.04.2008	Edit - Hapus	SOP	Uraian	Tambah Syarat & Langkah kerja	0		Tambah - Hapus	99	dddddddddddddd	Tambah - Hapus	20	PEMASANGAN JUMPER TRANSFORMATOR 3 PHASA PADA DUA TIANG/PORTAL DENGAN METODE BERJARAK	Tambah - Hapus	33	PENGADAAN ISOLATOR PENEGANGAN DENGAN METODE BERJARAK	Tambah - Hapus
No Lap	Lokasi	Uraian Pekerjaan	Keterangan	Tanggal Laporan	Edit - Hapus																																																					
1	UPJ Gondanglegi	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	01.11.2008	Edit - Hapus																																																					
2	UPJ Singosari	Pemas Aresrer untuk penangkalan petir	volume 6 set	11.11.2008	Edit - Hapus																																																					
3	APJ Malang	Penggantian Trafo Untuk Program Premian Trafo	volume 31 Unit	22.05.2009	Edit - Hapus																																																					
4	UPJ Nglantang	Penambahan Grounding TR U/perbaikan teg ujung	Volume 22 Set	01.01.2009	Edit - Hapus																																																					
5	UPJ Kepanjen	Pengadaan Kwh meter 3phasa	Volume 50 buah	28.03.2009	Edit - Hapus																																																					
6	APJ Malang	Gangguan Isolator	Volume 6 buah	06.04.2008	Edit - Hapus																																																					
SOP	Uraian	Tambah Syarat & Langkah kerja																																																								
0		Tambah - Hapus																																																								
99	dddddddddddddd	Tambah - Hapus																																																								
20	PEMASANGAN JUMPER TRANSFORMATOR 3 PHASA PADA DUA TIANG/PORTAL DENGAN METODE BERJARAK	Tambah - Hapus																																																								
33	PENGADAAN ISOLATOR PENEGANGAN DENGAN METODE BERJARAK	Tambah - Hapus																																																								

Sumber : Pengujian

Dari hasil pengujian integrasi pada *Use-case* Penanganan Kegiatan Kerja PDKB diketahui sistem dapat menyimpan data pegawai yang telah dibuat kedalam basis data.

6.2.2.3 Pengujian Integrasi untuk *Use-case Menyusun Analisis Pekerjaan dan Pembagian Tugas*

Dalam memenuhi kebutuhan yang terdapat pada *Use-case* Menyusun Analisis Pekerjaan dan Pembagian Tugas diintegrasikan beberapa class yaitu class perancang. Metode-metode yang dilibat untuk menjawab kebutuhan dari *Use-case* Menyusun Analisis Pekerjaan dan Pembagian Tugas adalah memasukan_pelaksana, memasukan_analisisKerja.

Tujuan pengujian integrasi ini adalah menguji apakah data yang dimasukan dapat tersimpan dengan baik pada basis data.

Kasus uji untuk pengujian *Use-case* Menyusun Analisis Pekerjaan dan Pembagian Tugas ditampilkan pada tabel dibawah ini.

Tabel 6.15 Kasus Uji untuk pengujian integrasi pada *Use-case* Menyusun Analisis Pekerjaan dan Pembagian Tugas [Pengujian]

Kasus Uji	:	Menjalankan prosedur untuk Menyusun Analisis Pekerjaan dan Pembagian Tugas melalui aplikasi Sistem Informasi Manajemen Kesehatan, Keselamatan, dan Keamanan Kerja.	
Prosedur	:	1	Aktor Login Sebagai perancang
		2	Aktor memilih sub menu <i>Tentukan Pelaksana</i> pada Menu Atur Pelaksana Kerja
		3	Aktor memilih salah satu tombol <i>tentukan jabatan</i> disamping daftar nama yang ditampilkan
		4	Aktor menentukan jabatan pekerjaan
		6	Aktor memilih sub menu <i>Tambah Analisis</i> pada Menu Analisis pekrjaan
		7	Aktor memilih salah satu tombol <i>tambah</i> disamping Uraian Pekerjaan
		8	Aktor menambahkan data laporan analisis
Hasil yang diharapkan	:	Sistem dapat menyimpan Data Analisis Pekerjaan yang telah dibuat kedalam basis data dengan baik	

Sumber : Pengujian

Hasil yang diperoleh dari hasil pengujian melalui kasus uji pada table diatas ditampilkan pada table dibawah ini.

Tabel 6.16 Hasil pengujian ntegrasi pada *Use-case* Menyusun Analisis Pekerjaan dan Pembagian Tugas [Pengujian]

<p>Hasil yang diharapkan</p>	<p>: Ssitem dapat menyimpan Data Analisis Pekerjaan dan Pembagian Tugas yang telah dibuat kedalam basis data dengan baik</p>												
<p>Hasil yang didapatkan</p>	<p>: Sistem dapat menyimpan Data Pelaksana kedalam basis data</p> <p>update_db_usulanpekerjaan set idpegawai='390010',jabatan_pekerjaan='Kepala Operasi' WHERE idpegawai = 390010; Data Berhasil di Update</p> <p>lihat data kembali ke halaman pengurus</p> <p>Sistem dapat menyimpan Data Analisis Pekerjaan dan Pembagian Tugas basis data</p> <div data-bbox="724 772 1333 974" style="border: 1px solid black; padding: 5px;"> <p>• Tekan tombol edit - hapus untuk menghapus atau mengubah nama pegawai.</p> <table border="1"> <thead> <tr> <th>No Lap</th> <th>Uraian Pekerjaan</th> <th>Keterangan</th> <th>Tindakan Penanganan</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Perbaikan Gangguan SUTM Km Bencana Alam Angin</td> <td>Volume 1,306 Kms</td> <td>Tambah - Ubah - Hapus</td> </tr> <tr> <td>2</td> <td>Pemas Arester untuk penangkalan petir</td> <td>volume 6 set</td> <td>Tambah - Ubah - Hapus</td> </tr> </tbody> </table> <p>Page 1 of 1 (2 rows)</p> </div>	No Lap	Uraian Pekerjaan	Keterangan	Tindakan Penanganan	1	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	Tambah - Ubah - Hapus	2	Pemas Arester untuk penangkalan petir	volume 6 set	Tambah - Ubah - Hapus
No Lap	Uraian Pekerjaan	Keterangan	Tindakan Penanganan										
1	Perbaikan Gangguan SUTM Km Bencana Alam Angin	Volume 1,306 Kms	Tambah - Ubah - Hapus										
2	Pemas Arester untuk penangkalan petir	volume 6 set	Tambah - Ubah - Hapus										

Sumber : Pengujian

Dari hasil pengujian integrasi pada *Use-case* Menyusun Analisis Pekerjaan dan Pembagian Tugas diketahui sistem dapat menyimpan data pegawai yang telah dibuat kedalam basis data

6.2.2.4 Pengujian Integrasi untuk *Use-case* Membuat Surat Tugas

Dalam memenuhi kebutuhan yang terdapat pada *Use-case* Membuat Surat Tugas diintegrasikan beberapa class yaitu class perancang, dan class pengawas. Metode-metode yang dilibat untuk menjawab kebutuhan dari *Use-case* Membuat Surat Tugas adalah memasukan_analisisPekerjaan, memasukan_suratTugas.

Tujuan pengujian integrasi ini adalah menguji apakah data yang dimasukan dapat tersimpan dengan baik pada basis data.

Kasus uji untuk pengujian *Use-case* Membuat Surat Tugas ditampilkan pada tabel dibawah ini.

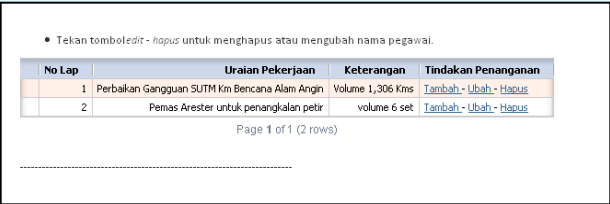
Tabel 6.17 Kasus Uji untuk pengujian integrasi pada *Use-case* Membuat Surat Tugas [Pengujian]

Kasus Uji	:	Menjalankan prosedur untuk Membuat Surat Tugas melalui aplikasi Sistem Informasi Manajemen Kesehatan, Keselamatan, dan Keamanan Kerja.	
Prosedur	:	1	Aktor Login Sebagai perancang
		2	Aktor memilih sub menu <i>Tambah Analisis</i> pada Menu Analisis pekerjaan
		3	Aktor memilih salah satu tombol <i>tambah</i> disamping Uraian Pekerjaan
		4	Aktor menambahkan data laporan analisis
		5	Aktor Logout sebagai perancang, dan login sebagai Pegawai
		6	Aktor memilih sub menu <i>Menyetujui Surat Tugas</i> pada Menu Surat Tugas
		7	Aktor memilih tombol <i>print</i> dibagian bawah, dan otomatis akan menyimpan data
Hasil yang diharapkan	:	Sistem dapat menyimpan Data pegawai yang telah dibuat kedalam basis data dengan baik	

Sumber : Pengujian

Hasil yang diperoleh dari hasil pengujian melalui kasus uji pada table diatas ditampilkan pada table dibawah ini.

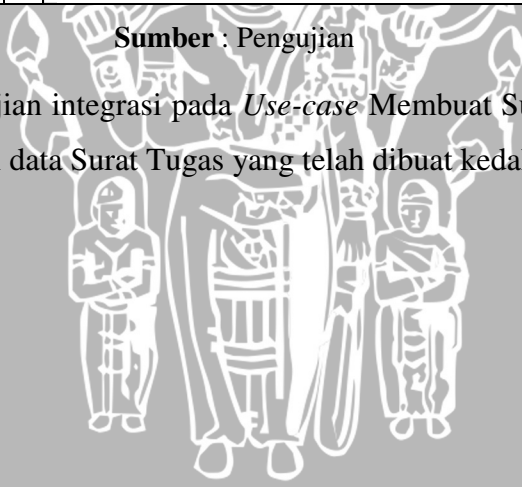
Tabel 6.18 Hasil pengujian ntegrasi pada *Use-case* Membuat Surat Tugas [Pengujian]

Hasil yang diharapkan	:	Ssitem dapat menyimpan Data Surat Tugas yang telah dibuat kedalam basis data dengan baik
Hasil yang didapatkan	:	<p>Sistem dapat menyimpan Data Analisis kedalam basis data</p>  <p>Sistem dapat menyimpan Data Surat Tugas kedalam basis</p>

	<p>data</p> <div style="border: 1px solid black; padding: 10px; background-color: #e0f0ff;"> <p>DATA ANALISIS PEKERJAAN</p> <p style="text-align: center;">SOP PDKB - TM NOMOR 05 SURAT PENUNJUKAN PENGAWASAN PEKERJAAN BERTEGANGAN (SP3B)</p> <p style="text-align: center;">No. 0003/PDKB-TM/2009 Tanggal : 28.03.2009</p> <p>Kepala Operasi atau Wakilnya, Memberikan Keewenangan Kepada Pengawas Pekerjaan : Hafid Ali Pemegang Sertifikat Kewenangan :</p> <p>UNTUK MELAKSANAKAN PENGAWASAN PEKERJAAN DALAM KEADAAN BERTEGANGAN PADA INSTALASI BERIKUT INI :</p> <p>JENIS PEKERJAAN YANG AKAN DILAKSANAKAN : Perbaikan Gangguan SUTM Km Bencana Alam Angin</p> <p>CARA OPERASI YANG DIPILIH OLEH PENGAWAS PEKERJAAN : Instalasi Kabel Grounding di 2000 bangunan dengan jarak < 3000M</p> <p>SYARAT OPERASI KHUSUS (Untuk Kategori kedua dan Ketiga)</p> <p>HUBUNGAN KOMUNIKASI DENGAN LOKASI : Pesawat HT</p> <p>KETERANGAN TAMBAHAN : Jarak EP Harap Diperhatikan</p> <p style="text-align: right;">PT.PLN (PERSERO) CABANG / APJ. UPT. Gondanglegi</p> <p style="text-align: center;">KEPALA OPERASI</p> <p style="text-align: center;">()</p> <p style="text-align: center;">PENGAWAS PEKERJAAN</p> <p style="text-align: center;">(Hafid Ali)</p> </div>
--	--

Sumber : Pengujian

Dari hasil pengujian integrasi pada *Use-case* Membuat Surat Tugas diketahui sistem dapat menyimpan data Surat Tugas yang telah dibuat kedalam basis data.



6.2.3 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. *Item-item* yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Dalam melakukan pengujian validasi digunakan metode pengujian *Black Box*, karena tidak memerlukan untuk berkonsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan. Obyek dari pengujian validasi adalah item-item kebutuhan yang telah dirumuskan pada daftar kebutuhan pada Tabel 4.6. Nama kasus uji dan obyek pengujian ditunjukkan pada Sub Bab 6.2.3.1 dan hasil pengujiannya akan ditunjukkan pada Sub Bab 6.2.3.2.

6.2.3.1 Kasus Uji Validasi

Untuk mengetahui kesesuaian antara kebutuhan dengan kinerja sistem, pada setiap kebutuhan (*requirement*) dilakukan proses pengujian dengan masing-masing kasus uji.

a. Kasus Uji Login

- | | |
|------------------|--|
| Nama Kasus Uji | : Kasus Uji Login Member |
| Objek Uji | : Kebutuhan no. 1 |
| Tujuan Pengujian | : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk <i>login</i> dengan menggunakan teknologi md5. |
| Prosedur Uji | : 1. Membuka antarmuka “Member Login” pada aplikasi <i>simk3</i> .
2. Mengisi <i>field</i> “Username” dan “Passsword” kemudian menekan tombol “Submit”. |
| Hasil yang | : Aplikasi dapat melakukan validasi nilai <i>username</i> dan <i>password</i> yang diisi pengguna dan menampilkan |

diharapkan antarmuka utama Member, serta terdapat pengacakan password dengan teknologi md5.

b. Kasus Uji Logout

Nama Kasus Uji : Kasus Uji Logout

Objek Uji : Kebutuhan no. 2

Tujuan Pengujian : Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk keluar dari antarmuka user.

Prosedur Uji : 1. Menekan menu “Logout” pada antarmuka utama user

Hasil yang diharapkan : Aplikasi dapat menghapus *session* member dan menampilkan informasi bahwa pengguna telah keluar dari antarmuka user.

6.1.5.1 Hasil Pengujian Validasi

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian pada sub pokok bahasan 6.2.3.1, didapatkan hasil seperti ditunjukkan pada Tabel 6.18.

Tabel 6.1 Hasil pengujian validasi

No	Kasus Uji	Hasil yang didapatkan	Status
1	Kasus Uji Login	Aplikasi dapat melakukan validasi nilai <i>username</i> dan <i>password</i> yang diisi pengguna dan menampilkan antarmuka utama dan password telah di enkripsi menggunakan teknologi md5.	Valid
2	Kasus Uji Logout	Aplikasi dapat menghapus <i>session</i> member dan menampilkan informasi bahwa pengguna telah keluar dari antarmuka	Valid

Sumber : Pengujian

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan hasil Perancangan, Implementasi, dan Pengujian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut :

1. Hasil pengujian *table* menunjukkan bahwa tabel basis data hasil implementasi pada aplikasi ini sesuai dengan tabel pada basis data hasil perancangan.
2. Hasil pengujian unit dan pengujian integrasi menunjukkan bahwa sistem dapat melakukan proses pengolahan dan penyimpanan data Kesehatan, keselamatan, dan keamanan kerja dengan baik, sesuai dengan hasil perancangan.
3. Sistem Informasi Manajemen Kesehatan Keselamatan dan keamanan kerja ini dapat menerima Informasi Data Pegawai, meliputi informasi kesehatan, dan informasi personal, Memfasilitasi Laporan gangguan, penanganan, dan tindakan yang akan dilakukan, Memfasilitasi pengusulan nama yang layak untuk bertugas, Memfasilitasi pembuatan surat tugas, dan memfasilitas laporan hasil pengerjaan kegiatan.
4. Pengujian aplikasi dilakukan dengan pengujian *E-R Diagram*, pengujian koneksi database, pengujian unit, pengujian integrasi, dan pengujian validasi.
5. Hasil dari seluruh rangkaian pengujian, menunjukkan bahwa sistem yang dikembangkan adalah benar sesuai dengan perancangan dan prosedur operasi yang telah disusun dari PT. PLN (persero) APJ Malang.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan lebih lanjut dari Sistem Informasi Manajemen Kesehatan Keselamatan dan Keamanan Kerja ini adalah :

1. Sistem Informasi Manajemen Kesehatan Keselamatan dan Keamanan Kerja dapat dikembangkan lebih luas, tidak hanya dalam lingkup PT.PLN (persero) APJ Malang saja.
2. Sistem Informasi Manajemen Kesehatan Keselamatan dan Keamanan Kerja dapat dikembangkan untuk diintegrasikan dengan sistem Informasi Sumber Daya Manusia (SDM) Lainnya.
3. Sistem ini masih belum memiliki tingkat keamanan yang baik, oleh karena itu keamanan dapat lebih ditingkatkan. Untuk meningkatkan keamanan pada sistem informasi ini, antara lain dapat digunakan *protocol* HTTPS untuk menghindari akses dari pihak-pihak yang tidak berwenang, terutama pada lalu lintas data pada *network*.

