

**SISTEM INFORMASI INVENTARISASI BARANG
DI BALAI PELATIHAN PENDIDIK DAN TENAGA KEPENDIDIKAN
PENDIDIKAN KEJURUAN (BPPTKPK)**

LAPORAN KERJA PRAKTEK

Karya ilmiah sebagai syarat kelulusan matakuliah Kerja Praktek

Disusun Oleh:

Marcel Eka Putra | 10110076

Muhamad Nur Awaludin | 10110090



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
BANDUNG**

2014

LEMBAR PENGESAHAN

SISTEM INFORMASI INVENTARISASI BARANG

DI BALAI PELATIHAN PENDIDIK DAN TENAGA KEPENDIDIKAN

PENDIDIKAN KEJURUAN (BPPTKPK)

Marcel Eka Putra (10110076)

Muhamad Nur Awaludin (10110090)

Pembimbing Kerja Praktek 1

Pembimbing Kerja Praktek 2

H. Nurjaman

NIP. 19610312.198103.1.004

Dian Dharmayanti, S.T., M.Kom.

NIP. 4127.70.06.005

Mengetahui,

Ketua Program Studi Teknik Informatika

Irawan Afrianto, S.T., M.T.

NIP. 4127.70.06.009

KATA PENGANTAR

Dalam menyelesaikan tugas ini, tidak terlepas dari bantuan dan kerja sama dari berbagai pihak terutama atas bimbingan Dian Dharmayanti, S.T., M.Kom. selaku pembimbing jurusan, Bapak H. Nurjaman selaku pembimbing lapangan, kedua orang tua yang kami cintai, teman-teman seperjuangan dan kepada semua pihak, baik secara langsung maupun tidak langsung yang telah memberikan bantuan dan sumber referensi sehingga laporan Kerja Praktek ini dapat diselesaikan.

Kami menyadari bahwa apa yang kami buat ini masih jauh dari sempurna, oleh sebab itu saran dan kritik yang bersifat membangun akan sangat membantu kami. Harapan kami adalah semoga apa yang telah kami tuangkan dalam laporan ini dapat bermanfaat bagi kita semua.

Bandung, 18 Januari 2014

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	II
KATA PENGANTAR	III
DAFTAR ISI.....	I
DAFTAR TABEL.....	VIII
DAFTAR GAMBAR	IX
DAFTAR SIMBOL.....	XI
DAFTAR LAMPIRAN.....	VIII
BAB I.....	1
I.1. LATAR BELAKANG	1
I.2. IDENTIFIKASI MASALAH.....	2
I.3. MAKSUD DAN TUJUAN	2
I.3.1. MAKSUD.....	2
I.3.2. TUJUAN	2
I.4. BATASAN MASALAH	2
I.5. METODE PENELITIAN.....	3
I.6. SISTEMATIKA PENULISAN.....	5
BAB II.....	7
II.1. PROFIL TEMPAT KERJA PRAKTEK.....	7
II.1.1. Sejarah Singkat Perusahaan	7
II.2. VISI DAN MISI	7
II.2.1. Visi.....	7
II.2.2. Misi	8
II.3. STRUKTUR ORGANISASI	8
II.4. DESKRIPSI KERJA	9
II.5. LANDASAN TEORI	10
II.5.1. Pengertian Sistem.....	10
II.5.2. Pengertian Informasi	13
II.5.3. Pengertian Sistem Informasi	14
II.5.4. Pengertian Inventarisasi Barang.....	14
II.5.5. Tujuan Inventarisasi Barang	14
II.5.6. Fungsi Inventarisasi Barang.....	15
II.5.7. Konsep Dasar Analisis Sistem	15
II.5.7.1. OOP (Object Oriented Programming)	15
II.5.7.2. Kelas (<i>Class</i>).....	16
II.5.7.3. Objek (Object)	16
II.5.7.4. Abstraksi (Abstract)	16
II.5.7.5. Enkapsulasi (Encapsulation).....	17
II.5.7.6. Polimorfisme (Polymorphism).....	17
II.5.7.7. Inheritas (Inheritance).....	17
II.5.7.8. UML (Unified Modelling Language).....	17
BAB III	21
III.1. ANALISIS SISTEM	21

III.1.1.	Analisis Masalah	21
III.1.2.	Analisis Prosedur	22
III.1.3.	Analisis Kebutuhan Non Fungsional	24
III.1.3.1	Analisis Kebutuhan Perangkat Keras	25
III.1.3.2	Analisis Kebutuhan Perangkat Lunak	25
III.1.3.3	Analisis Perangkat Pikir (Brainware).....	26
III.1.3.4	Analisis Kebutuhan Pengkodean	28
III.1.4.	Analisis Kebutuhan Fungsional	29
III.2.	MODEL USE CASE	31
III.2.1.	Use Case Diagram.....	31
III.2.2.	Definisi Aktor	31
III.2.3.	Definisi Use Case.....	32
III.2.4.	Skenario Use Case	34
III.2.4.1	Use Case Login.....	34
III.2.4.2	Use Case <i>View</i> Barang	35
III.2.4.3	Use Case:Cetak Laporan	36
Cetak Laporan		36
III.2.4.4	Use Case: Filter Barang	37
III.2.4.5	Use Case Ubah Data Barang	38
III.2.4.6	Use Case Pengeluaran Barang	39
III.2.4.7	Use Case Pengajuan Barang	40
III.2.4.8	Use Case Penerimaan Barang	41
III.2.4.9	Use Case Pengolahan Data Master	42
III.3.	MODEL ANALISIS	43
III.3.1	Realisasi Use Case Tahap Analisis	43
III.3.1.1.	Use Case Login.....	43
III.3.1.2.	Use Case <i>View</i> Barang	44
III.3.1.3.	Use Case Cetak Laporan	45
III.3.1.4.	Use Case Filter Barang	46
III.3.1.5.	Use Case Ubah Data Barang	47
III.3.1.6.	Use Case Pengeluaran Barang	48
III.3.1.7.	Use Case Pengajuan Barang	49
III.3.1.8.	Use Case Penerimaan Barang	50
III.3.1.9.	Use Case Pengolahan Data Master	51
III.3.1.10.	<i>Class</i> Terlibat	52
III.3.1.11.	Diagram Kelas Keseluruhan	58
III.3.1.12.	Kelompok Kelas dalam Pattern MVC.....	59
III.4.	PERANCANGAN SISTEM	60
III.4.1.	Perancangan Data.....	60
III.4.1.1	Skema Relasi	60
III.4.1.2	Struktur File.....	62
III.5.	IMPLEMENTASI SISTEM.....	66
III.5.1.	Implementasi Antarmuka	66
III.5.1.1	Antarmuka <i>login</i> Awal.....	72
III.5.1.2	Antarmuka <i>login</i> Pegawai.....	73
III.5.1.3	Antarmuka Menu Utama.....	73
III.5.1.4	Antarmuka Menu Ruangan Dipilih.....	74
III.5.1.5	Antarmuka Form Barang Pegawai	74
III.5.1.6	Antarmuka Form Tambah Barang	75
III.5.1.7	Antarmuka Form Barang Belum Diperiksa Secara Fisik	75
III.5.1.8	Antarmuka Form Ruangan.....	76
III.5.1.9	Antarmuka Form Merk	76
III.5.1.10	Antarmuka Form Kategori	77
III.5.1.11	Antarmuka Form Divisi	77
III.5.1.12	Antarmuka Form Barang Umum.....	78
III.5.1.13	Antarmuka form Export.....	78
III.5.1.14	Antarmuka form pengeluaran barang.....	79
III.5.1.15	Antarmuka pesan konfirmasi kegagalan penyimpanan barang	79

III.5.1.16	Antarmuka Pesan Konfirmasi Kegagalan Penyimpanan form divisi.....	80
III.5.1.17	Antarmuka pesan konfirmasi kegagalan penyimpanan kategori form tidak lengkap.	80
III.5.1.18	Antarmuka pesan konfirmasi kegagalan penyimpanan kode divisi sudah eksis.....	81
III.5.1.19	Antarmuka pesan konfirmasi kegagalan penyimpanan kategori id sudah eksis	81
III.5.1.20	Antarmuka pesan konfirmasi kegagalan penyimpanan kode barang sudah eksis.....	82
III.5.1.21	Antarmuka pesan konfirmasi kegagalan penyimpanan merk form tidak lengkap.	82
III.5.1.22	Antarmuka pesan konfirmasi kegagalan penyimpanan merk Id sudah eksis.....	83
III.5.1.23	Antarmuka pesan konfirmasi kegagalan penyimpanan form ruangan yang tidak lengkap	83
III.5.1.24	Antarmuka pesan konfirmasi kegagalan penyimpanan ruangan Id sudah eksis	84
III.5.1.25	Antarmuka pesan konfirmasi kesalahan login.....	84
BAB IV		85
IV.1.	KESIMPULAN	85
IV.2.	SARAN.....	85
DAFTAR PUSTAKA		86

DAFTAR TABEL

Tabel III. 1. Definisi Aktor	32
Tabel III. 2. Definisi Use Case.....	32
Tabel III. 3 <i>Class</i> yang terlibat.....	52
Tabel III. 4 Struktur tabel login.....	65
Tabel III. 5 Struktur tabel barang.....	62
Tabel III. 6 Struktur tabel detail_barang	62
Tabel III. 7 Struktur tabel divisi.....	63
Tabel III. 8 Struktur tabel kategori.....	63
Tabel III. 9 Struktur tabel keadaan barang.....	63
Tabel III. 10 Struktur tabel merk	64
Tabel III. 11 Struktur tabel ruangan.....	64
Tabel III. 12 Struktur tabel transaksi_barang.....	65

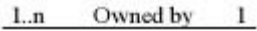
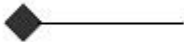


DAFTAR GAMBAR

Gambar I.1. Metode <i>Waterfall</i>	4
Gambar II.1. Struktur Organisasi Balai Pelatihan Pendidik dan Tenaga Kependidikan (BPPTKPK) Dinas Pendidikan Provinsi Jawa Barat.....	8
Gambar III. 1 <i>Activity Diagram</i> Pengadaan Barang	22
Gambar III. 2 <i>Activity Diagram</i> Penerimaan Barang	23
Gambar III. 3 <i>Activity Diagram</i> Pengeluaran Barang	24
Gambar III. 5 Use Case Diagram	31
Gambar III. 6 <i>Sequence Diagram</i> Login.....	43
Gambar III. 7 <i>Sequence Diagram</i> View Barang	44
Gambar III. 8 <i>Sequence Diagram</i> Cetak Laporan.....	45
Gambar III. 9 <i>Sequence Diagram</i> Filter Barang	46
Gambar III. 10 <i>Sequence Diagram</i> Tambah Data Barang	50
Gambar III. 11 <i>Sequence Diagram</i> Ubah Data Barang	47
Gambar III. 23 <i>Class Diagram</i> Keseluruhan	58
Gambar III. 24 Skema Relasi Inventarisasi Barang BPPTKPK	61
Gambar III. 25 Implementasi Antarmuka <i>login</i> Awal	72
Gambar III. 26 Implementasi Antarmuka <i>login</i> Pegawai	73
Gambar III. 27 Implementasi Antarmuka Menu Utama	73
Gambar III. 29 Implementasi Antarmuka <i>Form</i> Barang Pegawai	74
Gambar III. 30 Implementasi Antarmuka <i>Form</i> Tambah Barang.....	75
Gambar III. 31 Implementasi Antarmuka <i>Form</i> Barang Belum Diperiksa Secara Fisik.....	75
Gambar III. 32 Implementasi Antarmuka <i>Form</i> Ruangan	76
Gambar III. 33 Implementasi Antarmuka <i>Form</i> Merk.....	76
Gambar III. 34 Implementasi Antarmuka <i>Form</i> Kategori	77
Gambar III. 35 Implementasi Antarmuka <i>Form</i> Divisi	77
Gambar III. 36 Implementasi Antarmuka <i>Form</i> Barang Umum.....	78
Gambar III. 37 Implementasi Antarmuka <i>Form</i> Export	78
Gambar III. 38 Implementasi Antarmuka <i>Form</i> Pengeluaran Barang	79
Gambar III. 39 Implementasi Antarmuka Pesan Konfirmasi Kegagalan Penyimpanan Barang	79
Gambar III. 40 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Divisi <i>Form</i> Tidak Lengkap.....	80
Gambar III. 41 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Kategori <i>Form</i> Tidak Lengkap.....	80
Gambar III. 42 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Divisi Kode Sudah Eksis	81
Gambar III. 43 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Kategori Id Sudah Eksis.....	81
Gambar III. 44 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Kode Barang Sudah Eksis	82
Gambar III. 45 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Merk <i>Form</i> Tidak Lengkap.....	82




Gambar III. 46 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Merk Id Sudah Eksis	83
Gambar III. 47 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Ruangan <i>Form</i> Tidak Lengkap.....	83
Gambar III. 48 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Ruangan Id Sudah Eksis.....	84
Gambar III. 49 Implementasi Antarmuka Pesan Kesalahan Login.....	84

DAFTAR SIMBOL

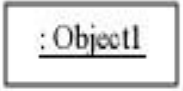



1. *Class Diagram* [7]

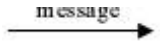
Nama Komponen	Keterangan	Simbol
<i>Class</i>	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>Class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>Class</i> . Bagian tengah mendefinisikan property/atribut <i>Class</i> . Bagian akhir mendefinisikan <i>method</i> dari sebuah <i>Class</i> .	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Nama <i>Class</i> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> + atribut + atribut + atribut </div> <div style="border: 1px solid black; padding: 5px;"> + <i>method</i> + <i>method</i> </div>
<i>Association</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>Class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>Class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> . (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>).	
<i>Composition</i>	Jika sebuah <i>Class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>Class</i> yang lain, maka <i>Class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>Class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
<i>Dependency</i>	Kadangkala sebuah <i>Class</i> menggunakan <i>Class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>Class</i> yang menggunakan <i>Class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.	
<i>Aggregation</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi	

2. Use Case Diagram [7]











Nama Komponen	Keterangan	Simbol
Use Case	<i>Use case</i> digambarkan sebagai lingkaran elips dengan nama <i>use case</i> dituliskan didalam elips tersebut.	
Aktor	<i>Aktor</i> adalah pengguna sistem. <i>Aktor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan <i>output</i> , maka aplikasi tersebut juga bisa dianggap sebagai <i>Aktor</i> .	
Association	Asosiasi digunakan untuk menghubungkan <i>Aktor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>Aktor</i> dengan <i>Use Case</i> .	

3. Sequence Diagram [7]

Nama Komponen	Keterangan	Simbol
Object	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>Class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>Class</i> (kotak) dengan nama <i>object</i> didalamnya yang diawali dengan sebuah titik koma.	
Aktor	<i>Aktor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>Aktor</i> juga dapat diurutkan sebagai kolom. Simbol <i>Aktor</i> sama dengan simbol pada <i>Aktor</i> Use Case Diagram.	
Lifeline	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah <i>object</i> .	
Activation	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . Mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.	

Nama Komponen	Keterangan	Simbol
Message	<i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation Message</i> mengindikasikan komunikasi antara <i>object –object</i> .	

5. Activity Diagram [7]

Simbol	Keterangan
	Titik awal
	Titik akhir
	<i>Activity</i>
	Pilihan untuk mengambil keputusan
	<i>Fork</i> ; Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Rake</i> ; Menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (<i>Flow Final</i>)

DAFTAR LAMPIRAN

Lampiran A Listing Program	A-1
Lampiran B Surat Balasan Penelitian Kerja Praktek	B-1
Lampiran C Daftar Nilai Kerja Praktek	C-1
Lampiran D Daftar Hadir Kerja Praktek	D-1
Lampiran E Lembar Pengesahan	E-1
Lampiran F Surat Keterangan Hak Eksklusif	F-1
Lampiran G Daftar Riwayat Hidup	G-1

BAB I

PENDAHULUAN

I.1. Latar Belakang

Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK) Jawa Barat secara fisik berdiri pada tahun 1975 dengan nama Pusat Latihan Pendidikan Teknik (PLPT). Lembaga ini mempunyai misi untuk optimalisasi dan pengembangan sumberdaya kelembagaan dalam upaya meningkatkan layanan pendidikan kejuruan secara produktif, efektif, efisien dan akuntabel juga disamping itu bertujuan untuk meningkatkan mutu, daya saing dan relevansi pendidikan melalui layanan pelatihan pendidik dan tenaga kependidikan pendidikan kejuruan yang menguasai teknologi dan berwawasan global.

Inventarisasi barang adalah hal yang cukup krusial di lembaga BPPTK dikarenakan terdapat data barang yang harus dipertanggungjawabkan validitasnya dari berbagai divisi seperti Tata Usaha, Multimedia, Listrik, Bangunan, Elektro, Mesin dan Otomotif. Selain daripada itu setiap kurun waktu 6 bulan sekali selalu ada pemeriksaan dari biro lembaga negara yaitu Biro Aset.

Proses inventarisasi barang yang meliputi pengadaan, pengiriman, pemeriksaan dan pengeluaran barang yang sedang berjalan saat ini dinilai kurang efektif dan efisien dikarenakan dalam prosesnya hanya dilakukan oleh satu orang petugas inventaris dengan menggunakan *Microsoft Excel*. Hal ini seringkali menyebabkan terjadinya kekeliruan pendataan barang seperti redudansi data dan ketidakvalidan data karena tidak adanya standarisasi inventaris barang pada sistem yang berjalan saat ini. Selain itu ketika petugas akan meng*update* data inventaris barang terhadap sebuah kode barang masih harus dilakukan *updating* secara berturut-turut untuk setiap data kode barang tersebut berada, dimana hal itu berimbas kepada lambatnya proses *updating* barang. Adapun disaat biro aset melakukan pemeriksaan data barang di lembaga BPPTKPK seringkali terjadi kesulitan dalam pencarian data dari tumpukan dokumen.

Berdasarkan masalah tersebut maka kami memutuskan untuk membuat “Sistem Informasi Inventarisasi Barang Balai Pelatihan Pendidik Dan Tenaga Kependidikan Pendidikan Kejuruan”.

I.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka munculah beberapa masalah seperti berikut:

1. Sering terjadi kekeliruan saat pendataan barang oleh petugas inventaris yang menyebabkan redudansi dan ketidakvalidan data barang.
2. Lambatnya proses *updating* data barang pada dokumen utama inventaris barang.
3. Sering terjadi kesulitan pencarian data barang dari tumpukan dokumen oleh petugas biro aset yang datang untuk memeriksa.

I.3. Maksud dan Tujuan

Adapun maksud dan tujuan dari kerja praktek yang dilakukan di lembaga BPPTKPK adalah:

I.3.1. Maksud

Maksud dari kerja praktek yang dilakukan di lembaga BPPTKPK adalah untuk membuat “Sistem Informasi Inventarisasi Barang Balai Pelatihan Pendidik Dan Tenaga Kependidikan Pendidikan Kejuruan”.

I.3.2. Tujuan

Tujuan dilakukan pembuatan “Sistem Informasi Inventarisasi Barang Balai Pelatihan Pendidik Dan Tenaga Kependidikan Pendidikan Kejuruan” adalah:

1. Mengurangi kekeliruan pendataan barang agar tidak terjadi redudansi dan ketidakvalidan data inventaris barang.
2. Mempercepat proses *updating* data barang pada dokumen utama inventaris barang.
3. Memudahkan petugas biro aset yang datang untuk memeriksa dalam melakukan pencarian data barang.

I.4. Batasan Masalah

Untuk pembuatan Sistem Informasi Inventarisasi Barang BPPTKPK, kami menentukan beberapa batasan masalah sebagai berikut :

1. Data

Data yang digunakan dalam pengolahan data antara lain data barang, data divisi, data ruangan dan data pegawai. Adapun pengkodean data barang dilakukan dengan format sistem yaitu : kode barang sesuai kategori-kode divisi-id ruangan.

2. Proses

Proses yang terdapat pada aplikasi ini adalah sebagai berikut :

- a. Pengolahan data Inventaris Barang yang meliputi pengadaan, penerimaan, pemeriksaan dan pengeluaran oleh pegawai inventaris.
- b. Pencarian data Inventaris Barang oleh setiap divisi di lingkungan BPPTKPK dan Biro Aset.
- c. Pembuatan laporan inventarisasi barang dari setiap ruangan untuk semua divisi di BPPTKPK.

3. Informasi yang dihasilkan

Informasi yang dihasilkan dari Sistem Informasi Inventarisasi Barang BPPTKPK adalah :

- a. Laporan data barang dari setiap ruangan (KIR).
- b. Laporan data barang dari setiap divisi.

4. Model proses dan data

Model proses dan data yang digunakan dalam pembangunan aplikasi adalah model OOP.

5. Development Tools

- a. Bahasa pemrograman aplikasi adalah C#.
- b. DBMS yang digunakan adalah Mysql.

I.5. Metode Penelitian

Dalam kerja praktek ini, dilakukan pengamatan pada sistem yang sedang berjalan. Adapun beberapa tahapan yang dilakukan dalam metode penelitian kerja praktek ini, yaitu :

I.5.1. Tahapan pengumpulan data

a. Studi Literatur

Pengumpulan data dengan cara mengumpulkan literatur dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

b. Observasi

Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil.

c. Interview

Teknik pengumpulan data dengan mengadakan tanya jawab secara langsung yang ada kaitannya dengan topik yang diambil.

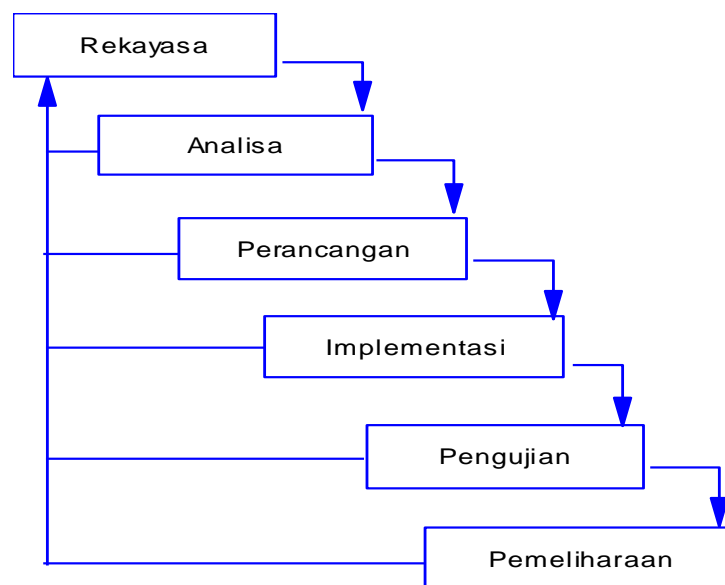
I.5.2. Tahapan pembangunan aplikasi

Tahapan pembangunan aplikasi yang kami gunakan adalah *waterfall modified*. Pada pengembangan suatu perangkat lunak dengan metode *Waterfall*, tahapan-tahapan pengembangan yang dilakukan adalah rekayasa sistem, analisis, perancangan, implementasi, pengujian, dan pemeliharaan.

Tahapan-tahapan tersebut saling terkait dan saling mempengaruhi (*life cycle*). Selain itu tahapan-tahapan tersebut dapat membentuk suatu siklus hal ini berarti pada suatu tahapan ternyata ada data-data yang seharusnya diproses pada tahapan sebelumnya tetapi belum dilaksanakan, maka dapat kembali ketahapan sebelumnya.

Dalam pembangunan aplikasi inventaris barang ini digunakan metode daur hidup (*System Development Life Cycle*). Metode ini memiliki beberapa tahapan proses, yaitu tahapan perencanaan, analisis, perancangan, penerapan, evaluasi, penggunaan dan pemeliharaan. Sementara itu, dalam setiap tahapan dilakukan proses pendokumentasian atas segala yang telah dilakukan atau disepakati dalam setiap tahapan tertentu.

Keterkaitan tahapan-tahapan pengembangan perangkat lunak tersebut dapat dilihat pada gambar berikut :



Gambar I.1. Metode *Waterfall*

Prosedur *waterfall* menurut Roger S. Pressman adalah sebagai berikut:

- a. Rekayasa sistem (*sistem engineering*), adalah pembuatan suatu perangkat lunak yang merupakan bagian terbesar dari pengerjaan suatu proyek. Untuk pekerjaan dimulai dengan menetapkan segala hal yang diperlukan dalam pelaksanaan proyek.
- b. Analisis (*analysis*), merupakan tahapan dimana sistem engineering (rekayasa sistem) menganalisis hal-hal yang diperlihatkan dalam pelaksanaan proyek pembuatan atau pengembangan perangkat lunak dan bertujuan untuk memahami sistem yang ada, mengidentifikasi masalah dan mencari solusinya.
- c. Desain (*design*), tahap ini merupakan tahap penterjemahan dari keperluan atau data yang telah dianalisis kedalam bentuk yang mudah dimengerti oleh pemakai (*user*).
- d. Kode (*coding*), yaitu menterjemahkan data atau pemecahan masalah yang dirancang kedalam bahasa pemrograman yang telah ditentukan.
- e. Pengetesan (*testing*), setelah program selesai dibuat maka tahap berikutnya adalah ujicoba terhadap program tersebut.
- f. Pemeliharaan (*maintenance*), yaitu penerapan secara keseluruhan disertai pemeliharaan jika terjadi perubahan struktur, baik dari segi *software* maupun *hardware*. [1]

I.6. Sistematika Penulisan

Adapun sistematika yang dipakai penulis dalam penyusunan laporan Kerja Praktek ini adalah:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang kerja praktek, masalah kerja praktek, tujuan kerja praktek, batasan masalah dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini membahas tentang profil perusahaan berupa sejarah, logo dan struktur organisasi, serta landasan teori yang digunakan.

BAB III PEMBAHASAN

Bab ini membahas tentang analisis permasalahan, analisis data, analisis pengguna, kebutuhan perangkat keras, perangkat lunak, perancangan sistem informasi.

BAB IV KESIMPULAN DAN SARAN

Bab ini merupakan penutup penyusunan laporan yang berisi kesimpulan dari pengembangan Sistem Informasi yang telah dilaksanakan. Selain itu berisi pula saran yang diharapkan dapat menjadi masukan yang bermanfaat dan bersifat membangun.

BAB II

TINJAUAN PUSTAKA

II.1. Profil Tempat Kerja Praktek

Tinjauan umum perusahaan praktek kerja lapangan pada bab ini penulis menjelaskan secara singkat profil Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK) Dinas Provinsi Jawa Barat.

II.1.1. Sejarah Singkat Perusahaan

Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK) secara fisik berdiri pada tahun 1975 dengan nama Pusat Latihan Pendidikan Teknik (PLPT). Tahun 1975 s.d. 1978 PLPT dipimpin oleh Drs. M. Bakrie, MA. Tahun 1978 PLPT berganti nama menjadi Balai Latihan Pendidikan Teknik (BLPT). BLPT mengalami 6 (enam) kepemimpinan yaitu : (1) Drs. M. Bakrie, MA. tahun 1978 s.d. 1981, (2) Drs. Djanakum tahun 1981 s.d. 1983, (3) Drs. Daslam Karmananjaya tahun 1983 s.d. 1985, (4) Drs. Didikh Suryana, M.Ed. tahun 1985 s.d. 1987, (5) Drs. Rakhmad Dumadi, BE. tahun 1988 s.d. 1994, (6) Drs. H. Supri Raya tahun 1995 s.d. 2001.

Tahun 2002 melalui Keputusan Gubernur Jawa Barat Nomor 39 Tahun 2001 BLPT berubah menjadi Balai Pengembangan Teknologi Pendidikan (BPTP), yang melalui 4 (empat) kepemimpinan, yaitu : (1) Drs. F.H. Abiyanto, MM. tahun 2002 s.d. 2003, (2) Drs. H. Nanang Mustaram, M.Ed

Tahun 2003 s.d. 2005, (3) Drs. H. Otji S. Wiharjadi, M.Pd. tahun 2005 s.d. 2008, (4) Drs. H. Nandang Djunaedi, MM tahun 2009 s.d. 2010.

Tahun 2010 melalui Peraturan Gubernur Jawa Barat No. 113 tahun 2009 BPTP berubah menjadi Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK), dipimpin oleh Drs. H. Nandang Djunaedi, MM. hingga sekarang tahun 2010 s.d. sekarang.

II.2. Visi dan Misi

Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK) memiliki visi dan misi, yaitu:

II.2.1. Visi

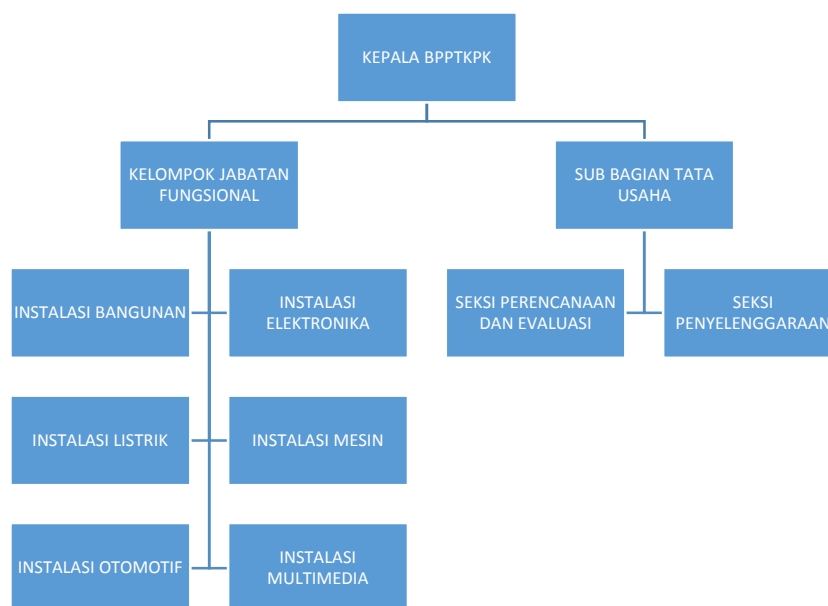
“Akselerasi Peningkatan Mutu Pendidikan Kejuruan menuju masyarakat Jawa Barat Bertqwa, Mandiri, Dinamis dan Sejahtera”.

II.2.2. Misi

- 1) Optimalisasi dan pengembangan sumberdaya kelembagaan dalam upaya meningkatkan layanan pendidikan kejuruan secara produktif, efektif, efisien dan akuntabel.
- 2) Meningkatkan mutu, daya saing dan relevansi pendidikan melalui layanan pelatihan pendidik dan tenaga kependidikan Pendidikan kejuruan yang menguasai teknologi dan berwawasan global.

II.3. Struktur Organisasi

Organisasi adalah tempat atau wadah orang berkumpul untuk saling bekerjasama untuk mencapai tujuan dan manfaat bersama. Dengan demikian dapat dikatakan bahwa struktur organisasi pada suatu organisasi merupakan kerangka dasar yang menggambarkan alur hubungan antara bagian yang satu dengan yang lainnya, sehingga suatu bagian dalam organisasi tersebut menjadi jelas kedudukan, jabatan, wewenang dan juga tanggung jawabnya.



Gambar II.1. Struktur Organisasi Balai Pelatihan Pendidik dan Tenaga Kependidikan (BPPTKPK) Dinas Pendidikan Provinsi Jawa Barat

II.4. Deskripsi Kerja

Penjelasan mengenai tugas dari masing-masing bidang dapat digambarkan sebagai berikut:

1) Kepala BPPTKPK

a) Tugas Pokok

Memimpin, mengkoordinasikan, dan memotivasi serta mengawasi ruang lingkup tugas pokok Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK) Dinas Provinsi Jawa Barat dan memberikan keleluasan kepada 26 kabupaten Jawa Barat untuk melakukan pelatihan

b) Fungsi

Kepanjangan tugas dan Dinas Pendidikan Provinsi Jawa Barat untuk pengembangan teknologi umum dan teknologi pendidikan.

2) Sub Bagian Tata Usaha

Sub Bagian Tata usaha mempunyai tugas membantu Kepala BPPTKPK di bidang ketatausahaan, kepegawaian, keuangan, perlengkapan umum serta perencanaan dan pelaporan dalam administrator manajemen Balai dan mengawasi struktural dalam pelaksanaan tugas pokok BPPTKPK.

3) Seksi Penyelenggaraan

Tugas pokoknya adalah mengumpulkan, menghimpun, menganalisa serta menyusun rencana dan program pelaksanaan pelatihan

4) Seksi Perencanaan dan Evaluasi

Tugas pokoknya adalah merencanakan pengembangan program model dan sistem pembelajaran, mengevaluasi, memantau dan membina pemanfaatan teknologi informasi untuk pendidikan serta program perencanaan dalam jangka pendek dan jangka panjang.

5) Kelompok Jabatan Fungsional

Kelompok Jabatan Fungsional mempunyai tugas pokok melaksanakan proses pembelajaran kepada seluruh SMK terdiri dari sejumlah jabatan fungsional yang terbagi dalam berbagai kelompok sesuai dengan bidang keahliannya. Kelompok Jabatan fungsional di BPPTKPK ada 6 Jabatan Fungsional dalam data ketenagaan sesuai dengan keahliannya sebagai berikut:

- a. Instalasi Bangunan
- b. Instalasi Elektronika

- c. Instalasi Listrik
- d. Instalasi Mesin
- e. Instalasi Otomotif
- f. Instalasi Multimedia

II.5. Landasan Teori

Subbab ini berisikan teori - teori pendukung yang digunakan dalam proses analisis dan implementasi pada permasalahan yang ada di BPPTKPK.

II.5.1. Pengertian Sistem

Sistem adalah suatu jaringan kerja dari dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. [2]

1. Elemen Sistem

Sistem informasi terdiri dari elemen-elemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen ini merupakan komponen fisik.

a. Orang

Orang atau personil yang dimaksudkan yaitu operator komputer, analisis sistem, programmer, personil data entry dan manajer sistem informasi

b. Prosedur

Prosedur merupakan elemen fisik. Hal ini di sebabkan karena prosedur disediakan dalam bentuk fisik seperti buku panduan dan instruksi. Ada 3 jenis prosedur yang dibutuhkan, yaitu instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.

c. Perangkat keras

Perangkat keras bagi suatu sistem informasi terdiri atas komputer (pusat pengolah, unit masukan/keluaran), peralatan penyiapan data dan terminal masukan/keluaran.

d. Perangkat Lunak

Perangkat lunak dapat dibagi dalam tiga jenis utama:

- 1) Sistem perangkat umum, seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.

- 2) Aplikasi perangkat lunak umum, seperti model analisis dan keputusan.
 - 3) Aplikasi perangkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.
- e. Basis data
- File yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik, seperti diskette, harddisk, *magnetic tape* dan sebagainya. File juga meliputi keluaran tercetak dan catatan lain diatas kertas, mikro film dan lain sebagainya.
- f. Jaringan Komputer
- Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.
- g. Komunikasi Data
- Komunikasi data merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer dapat berkomunikasi satu sama lain.

2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat tertentu, yaitu mempunyai: [2]

- a. Komponen. (*Components*)
Komponen terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama membentuk satu kesatuan.
- b. Batas Sistem (*Boundary*)
Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau dengan lingkungan luarnya.
- c. Lingkungan Luar Sistem (*Environments*)
Lingkungan luar sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.
- d. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara subsistem, yang memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

e. Masukan sistem (*input*)

Masukan sistem adalah energi yang dimasukkan kedalam sistem, yang dapat berupa masukan perawatan (Maintenance input) dan masukan signal (signal input).

f. Keluaran sistem (*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolahan sistem (*process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran atau Tujuan Sistem

Suatu system mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil apabila mengenal sasaran atau tujuannya karena sasaran sangat berpengaruh pada masukan dan keluaran sistem yang dihasilkan.

3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya sebagai berikut ini: [2]

- a. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*) Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi dan lain sebagainya.
- b. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*) Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine* sistem atau ada yang menyebut dengan *man-machine system*. Sistem informasi

merupakan contoh *man-machine system*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

- c. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tidak tentu (*probabilistic system*) Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
- d. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka hanya untuk pengaruh yang baik saja.

Suatu sistem yang dihubungkan dengan lingkungannya melalui arus sumber daya disebut sistem terbuka. Sebuah sistem pemanas atau pendingin ruangan, contohnya, mendapatkan input-nya dari perusahaan listrik, dan menyediakan panas/dinginnya bagi ruangan yang ditempatinya. Dengan menggunakan logika yang sama, suatu sistem yang tidak dihubungkan dengan lingkungannya adalah sistem tertutup. Sebagai contohnya, sistem tertutup hanya terdapat pada situasi laboratorium yang dikontrol ketat.

II.5.2. Pengertian Informasi

Secara singkat informasi adalah data yang telah diolah menjadi bentuk yang berarti bagi penerimanya dan berguna untuk pengambilan keputusan saat ini atau di masa mendatang. Akan tetapi secara lengkap informasi bisa disebut sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penirimnya yang

menggambarkan suatu kejadian-kejadian yang nyata yang dapat digunakan untuk pengambilan keputusan. [2]

Dari kedua pengertian diatas dapat disimpulkan bahwa Informasi adalah data yang sudah diproses atau diolah sehingga mempunyai nilai bagi penerimanya dan dapat digunakan untuk dasar pengambilan keputusan yang disampaikan melalui media kertas (*hardcopy*), tampilan (*display*) atau suara (*audio*).

II.5.3. Pengertian Sistem Informasi

Sistem informasi adalah kumpulan informasi di dalam sebuah basis data menggunakan model dan media teknologi informasi digunakan di dalam pengambilan keputusan bisnis sebuah organisasi. [3] Di dalam suatu organisasi, informasi merupakan sesuatu yang penting di dalam mendukung proses pengambilan keputusan oleh pihak manajemen. Sistem ini memanfaatkan perangkat keras dan perangkat lunak komputer, prosedur manual, model manajemen dan basis data. Sistem informasi memiliki komponen berupa subsistem yang merupakan elemen-elemen yang lebih kecil yang membentuk sistem informasi tersebut misalnya bagian input, proses, output. Dimana input itu sendiri adalah sekumpulan data yang akan kita olah menjadi sebuah informasi yang nantinya akan kita sajikan bagi masyarakat, sedangkan proses adalah suatu kegiatan dimana kita mengolah seluruh data yang ada untuk menghasilkan suatu informasi dan Output adalah sekumpulan informasi yang dapat dengan mudah diperoleh, dimengerti dan dimanfaatkan oleh masyarakat. Tanpa ketiga itu sistem informasi tidak dapat berjalan dengan baik.

II.5.4. Pengertian Inventarisasi Barang

Inventarisasi barang adalah kegiatan melaksanakan pengurusan berupa penyelenggaraan, pengaturan, pencatatan barang-brang, menyusun daftar barang yang bersangkutan ke dalam suatu daftar inventaris barang secara teratur dan menurut ketentuan yang berlaku. [4]

II.5.5. Tujuan Inventarisasi Barang

Tujuan umum dilakukan inventarisasi adalah dalam rangka usaha penyempurnaan pengurusan dan pengawasan yang efektif terhadap barang-barang milik negara atau swasta. Adapun tujuan khususnya adalah: [4]

1. Untuk menjaga ketertiban administrasi barang yang dimiliki
2. Untuk menghemat keuangan
3. Sebagai Bahan pedoman untuk menghitung kekayaan

4. Untuk Memudahkan pengawasan dan pengendalian barang.

II.5.6. Fungsi Inventarisasi Barang

Fungsi inventarisasi barang adalah: [4]

1. Menyediakan data dan informasi dalam rangka menentukan kebutuhan dan menyusun rencana kebutuhan barang.
2. Memberikan data dan informasi untuk dijadikan bahan / pedoman dalam pengarahan pengadaan barang.
3. Memberikan data dan informasi untuk dijadikan bahan/pedoman dalam penyaluran barang.
4. Memberikan data dan informasi dalam.
5. Menentukan keadaan barang (barang yang rusak/tua) sebagai dasar untuk menetapkan penghapusannya.
6. Memberikan data dan informasi dalam rangka memudahkan pengawasan dan pengendalian barang.

II.5.7. Konsep Dasar Analisis Sistem

Adapun konsep dasar analisis sistem adalah sebagai berikut: [5]

II.5.7.1. OOP (Object Oriented Programming)

OOP (*Object Oriented Programming*) atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus ke dalam kelas-kelas atau objek-objek. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bagian

administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri. Pemrograman orientasi-objek menekankan konsep berikut:

II.5.7.2. Kelas (*Class*)

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh '*Class of dog*' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah *Class* adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah *Class* secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah *Class* sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

II.5.7.3. Objek (*Object*)

Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

II.5.7.4. Abstraksi (*Abstract*)

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

II.5.7.5. Enkapsulasi (*Encapsulation*)

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

II.5.7.6. Polimorfisme (*Polymorphism*)

Melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan. Metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

II.5.7.7. Inheritas (*Inheritance*)

Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas).

II.5.7.8. UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem. Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang

telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP.

UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. 25 UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan system. UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu: [6]

1. Use Case Diagram

Use case diagram adalah gambaran graphical dari beberapa atau semua Aktor, use-case, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. Use-case diagram menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.

Use-case diagram dapat digunakan selama proses analisis untuk menangkap requirement sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case diagram* berperan untuk menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa use-case diagram. Kebutuhan atau *requirements system* adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan pada model use-case yang menggambarkan fungsi sistem yang diharapkan (*use case*), dan yang mengelilinginya (*Aktor*), serta hubungan antara Aktor dengan use case (*use case diagram*) itu sendiri.

2. Conceptual Diagram

Sebuah diagram konseptual merupakan representasi visual dari cara di mana konsep-konsep abstrak terkait. Hal ini digunakan sebagai bantuan dalam memvisualisasikan proses atau sistem tingkat tinggi melalui serangkaian garis yang unik dan bagan. Diagram konseptual secara luas digunakan dalam segala bidang seperti bisnis, ilmu pengetahuan, dan manufaktur.

3. *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence Diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang *men-trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

4. *Collaboration Diagram*

Collaboration diagram yaitu diagram yang mengelompokkan pesan pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram tersebut terdapat method yang dijalankan antara objek yang satu dan objek lainnya. Di diagram kolaborasi ini, objek harus melakukan sinkronisasi pesan dengan serangkaian pesan-pesan lainnya. *Collaboration Diagram* lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan.

5. *State Diagram*

State Diagram adalah diagram untuk menggambarkan behavior, yaitu perubahan state di suatu *Class* berdasarkan event dan pesan yang dikirimkan dan diterima oleh *Class* tersebut. Setiap diagram state hanya boleh memiliki satu start state (*initial state*) dan boleh memiliki satu atau lebih dari satu stop states (*final state*).

6. *Activity Diagram*

Activity Diagram memiliki pengertian yaitu lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Memiliki struktur diagram yang mirip *flowchart* atau data *flow diagram* pada perancangan terstruktur. Memiliki pula manfaat yaitu apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Dan activity dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.

7. *Class Diagram*

Class diagram adalah sebuah *Class* yang menggambarkan struktur dan penjelasan *Class*, paket, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* juga menjelaskan hubungan antar *Class* dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.

8. *Object Diagram*

Objek diagram adalah diagram yang memberikan gambaran struktur model sebuah sistem, dalam kurun waktu tertentu. Diagram objek yang berasal dari diagram kelas sehingga diagram objek tergantung pada diagram kelas. Objek Diagram, kadang-kadang disebut sebagai *diagram instance* sangat mirip dengan diagram kelas. Seperti diagram kelas *object diagram* juga menunjukkan hubungan antara obyek, tetapi *object diagram* menggunakan contoh-contoh dunia nyata. *Object diagram* digunakan untuk menunjukkan bagaimana sistem akan terlihat seperti pada waktu tertentu. Karena ada data yang tersedia di objek-objek diagram sering digunakan untuk menjelaskan hubungan yang kompleks antara objek.

9. *Component Diagram*

Component diagram adalah diagram UML yang menampilkan komponen dalam system dan hubungan antara mereka. Pada *component View*, akan difokuskan pada organisasi fisik system. Pertama, diputuskan bagaimana kelas-kelas akan diorganisasikan menjadi kode pustaka. Kemudian akan dilihat bagaimana perbedaan antara berkas eksekusi, berkas *dynamic link library* (DDL), dan berkas runtime lainnya dalam system.

10. *Deployment Diagram*

Deployment Diagram adalah diagram yang menggambarkan detail bagaimana komponen disebar kedalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, node, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik.

BAB III

PEMBAHASAN

III.1. Analisis Sistem

Analisis merupakan suatu tahapan pemahaman terhadap sistem atau aplikasi yang sedang berjalan maupun yang akan dibuat. Tahapan analisis bertujuan untuk mengetahui mekanisme atau prosedur kerja dari proses yang sedang berjalan maupun yang akan dibuat.

III.1.1. Analisis Masalah

Adapun analisis masalah pada sistem yang berjalan di Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK) adalah sebagai berikut:

- 1 Belum adanya sebuah sistem terkomputerisasi yang mendukung kemampuan pengolahan data barang secara cepat, aman dan dapat meminimalkan kesalahan-kesalahan yang dapat terjadi ketika pendataan barang inventaris.
- 2 Selama ini penginputan data barang inventaris dilakukan secara manual terhadap beberapa dokumen yang saling berelasi dan belum ada pencegahan oleh sistem terhadap data inputan yang beredudansi. Ketika terjadi perubahan pada suatu dokumen maka pegawai inventarisasi barang pun perlu melakukan perubahan pada dokumen berelasi lainnya sehingga selain menyebabkan kelelahan, banyak waktu yang terbuang hanya untuk melakukan proses penambahan dan perubahan data barang.
- 3 Penginputan barang yang manual menyebabkan sering adanya kesulitan untuk merekap data barang untuk menjadi sebuah laporan ketika Biro Aset datang untuk memeriksa. Akses pada dokumen pun tidak dilengkapi dengan autentifikasi pengguna terhadap dokumen sehingga menyebabkan dokumen inventarisasi barang mudah untuk dilihat dan dibuka oleh pihak yang tidak berkewenangan mengingat komputer pada ruangan invetarisasi barang bagian ketatausahaan dipakai secara umum.

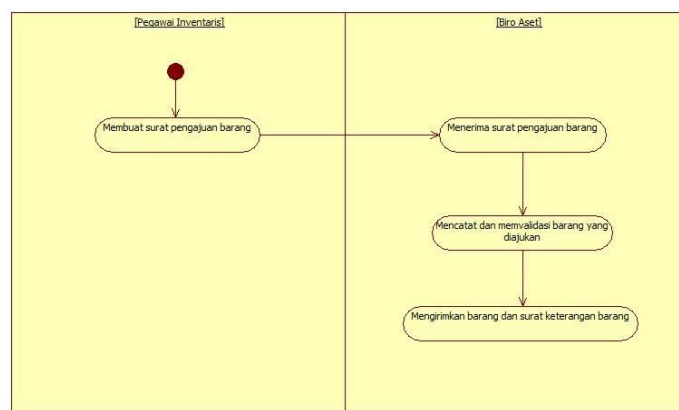
III.1.2. Analisis Prosedur

Berdasarkan pengamatan sistem yang sedang berjalan, diperlukan suatu perbaikan proses pendataan inventarisasi barang di BPPTKPK yang meliputi penginputan dan perubahan data barang yang masih berjalan secara manual, pembuatan laporan yang belum dilakukan secara otomatis sehingga dapat memakan waktu kerja bagi pegawai inventarisasi barang, serta keamanan dokumen yang masih rentan terhadap akses dari pihak yang tidak berkewenangan.

Adapun prosedur inventarisasi barang yang sedang berjalan saat ini adalah sebagai berikut:

1. Prosedur Pengajuan Barang kepada Biro Aset

- a. Pegawai inventaris membuat surat pengajuan barang kepada Biro Aset.
- b. Biro Aset menerima surat pengajuan barang dari lembaga BPPTKPK yang diwakili oleh pegawai inventaris.
- c. Biro Aset mencatat dan memvalidasi seluruh barang yang telah diajukan.
- d. Biro Aset mengirimkan barang yang telah diajukan kepada lembaga BPPTKPK.

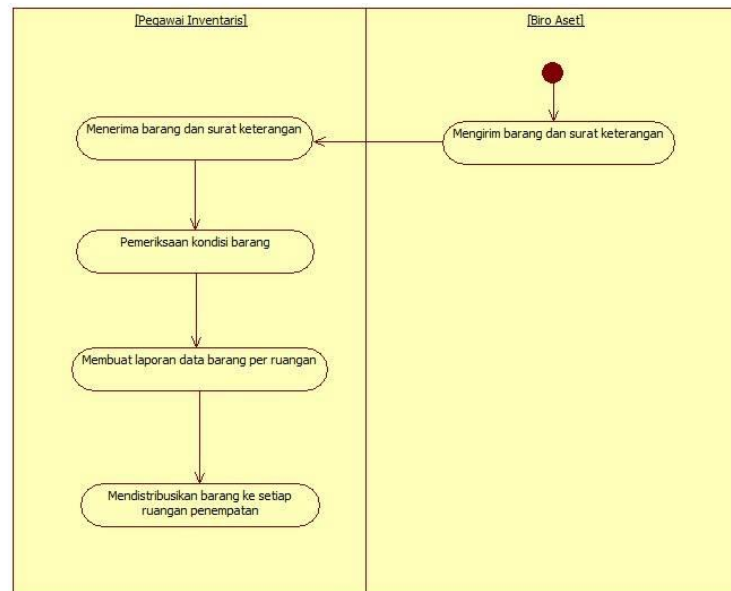


Gambar III. 1 Activity Diagram Pengadaan Barang

2. Prosedur Penerimaan Barang dari Biro Aset

- a. Biro Aset mengadakan inventaris barang baru setiap 2 tahun sekali.
- b. Pegawai inventaris menerima barang dan surat keterangan barang dari biro aset.
- c. Pegawai inventaris mengecek barang yang telah diterima terlebih dahulu.
- d. Pegawai inventaris membuat laporan data barang yang akan didistribusikan ke setiap ruangnya.

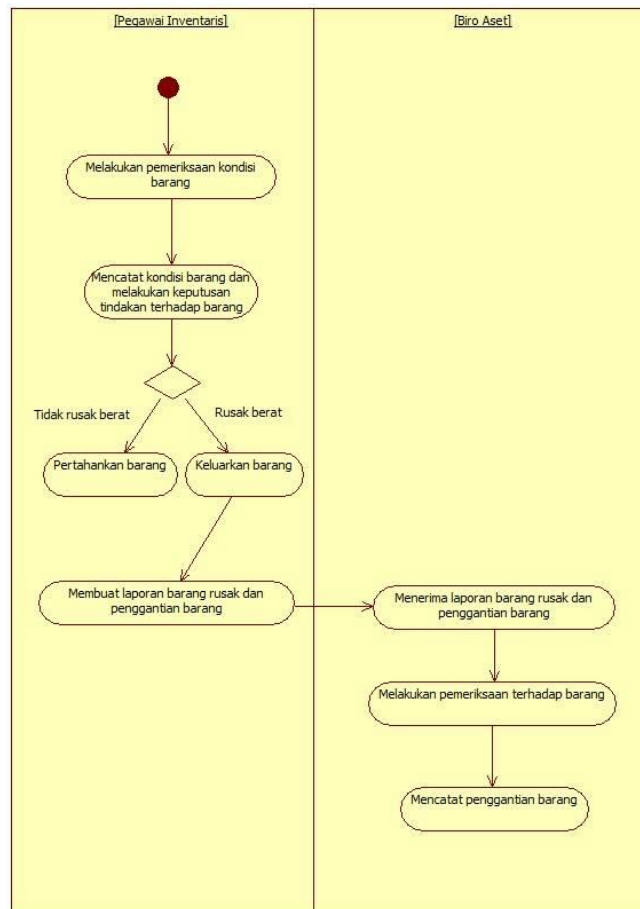
- e. Pegawai inventaris melakukan penempatan barang-barang yang sudah diterima ke setiap ruangan sesuai kebutuhan.



Gambar III. 2 Activity Diagram Penerimaan Barang

3. Prosedur Pengeluaran Barang dari lingkungan BPPTKPK

- Pegawai inventaris melakukan pemeriksaan barang yang ada di lingkungan BPPTKPK setiap kurun waktu tertentu atau menerima laporan barang yang sudah tidak memenuhi standar (rusak berat) dari pegawai BPPTKPK.
- Pegawai inventaris mencatat laporan keadaan barang (baik, rusak ringan, rusak berat) dari setiap ruangan yang diperiksa maupun dari laporan keadaan barang dari pegawai BPPTKPK.
- Apabila barang tersebut dalam keadaan rusak berat maka akan dikeluarkan dari ruangan, apabila tidak dalam keadaan rusak berat maka barang tersebut akan tetap dipertahankan.
- Pegawai inventaris membuat laporan pengeluaran dan penggantian barang kepada Biro Aset.
- Biro Aset menerima laporan pengeluaran barang dari pegawai inventaris barang.
- Biro Aset melakukan pengecekan data barang yang diajukan.
- Biro Aset mencatat barang penggantian untuk setiap barang yang dikeluarkan dari lembaga BPPTKPK.



Gambar III. 3 Activity Diagram Pengeluaran Barang

III.1.3. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen-elemen atau komponen-komponen yang dibutuhkan oleh sistem untuk dibangun sampai dengan sistem tersebut dapat diimplementasikan untuk pendataan inventarisasi barang di BPPTKPK. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem, dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

III.1.3.1 Analisis Kebutuhan Perangkat Keras

Adapun analisis spesifikasi kebutuhan perangkat keras yang dibutuhkan untuk sistem yang akan dibangun adalah sebagai berikut :

1. Fakta Perangkat Keras Yang Dimiliki Pegawai Inventaris
 - a. Komputer dengan OS Windows 7 atau XP
 - b. Processor minimum 2.93 GHz
 - c. RAM minimum 1 GB
 - d. Free Hard Disk 40GB
 - e. Monitor 14.1”
 - f. Keyboard dan mouse
 - g. Printer
2. Kebutuhan Perangkat Keras Untuk Pegawai Inventaris
 - a. Komputer dengan OS Windows 7 atau XP
 - b. Processor minimum 2.93 GHz
 - c. RAM minimum 1 GB
 - d. Free Hard Disk 40GB
 - e. Monitor 14.1”
 - f. Keyboard dan mouse
 - g. Printer
3. Kesimpulan Perangkat keras yang tersedia untuk pegawai inventaris di lembaga BPPTKPK saat ini sudah memenuhi standard untuk menjalankan sistem yang dibangun.

III.1.3.2 Analisis Kebutuhan Perangkat Lunak

Adapun analisis spesifikasi kebutuhan perangkat lunak yang digunakan adalah sebagai berikut :

1. Fakta Perangkat Keras yang dimiliki Pegawai Inventaris
 - a. OS Windows 7
 - b. Microsoft Office 2007
2. Kebutuhan Perangkat Keras Untuk Pegawai Inventaris
 - a. OS Windows 7
 - b. Framework .Net 4.5
 - c. Wamp Server 2.0

3. Kesimpulan Perangkat lunak yang tersedia untuk pegawai inventaris di lembaga BPPTKPK saat ini perlu diinstal Framework .Net 4.5 dan Wamp Server 2.0 untuk memenuhi standar sistem yang akan dibangun.

III.1.3.3 Analisis Perangkat Pikir (Brainware)

Adapun analisis kemampuan minimum pengguna untuk mengatasi permasalahan didalam sistem adalah sebagai berikut:

1. Fakta Perangkat Pikir

No.	Stakeholder	Tanggung Jawab	Tingkat Pendidikan	Tingkat keterampilan yang dimiliki	Pengalaman menggunakan computer
1	Biro Aset	Mengadakan Barang, melakukan pengecekan barang setiap kurun waktu 3 bulan, mengadakan penggantian barang untuk barang yang rusak	S1	Microsoft Office	8 Tahun
2	Pegawai Inventaris	Mengelola Data Barang Inventaris di lingkungan BPPTKPK	D3	Microsoft Office	8 Tahun

2. Kebutuhan Perangkat Pikir

No.	Stakeholder	Tanggung Jawab	Tingkat Pendidikan	Tingkat keterampilan yang dimiliki	Pengalaman menggunakan computer
1	Biro Aset	Mengadakan Barang, melakukan pengecekan barang setiap kurun waktu 3 bulan, mengadakan penggantian barang untuk barang yang rusak	S1	Microsoft Office	2 tahun
2	Pegawai Inventaris	Mengelola data Barang Inventaris di lingkungan BPPTKPK	D3	Microsoft Office, Pengetahuan Database, Penggunaan Software khusus Inventaris	2 Tahun

3. Kesimpulan

Perangkat fikir yang tersedia di BPPTKPK belum sesuai dengan kebutuhan perangkat fikir yang akan dibangun khususnya untuk pegawai inventaris, sehingga diperlukan diadakannya sedikit pelatihan untuk menyesuaikan kebutuhan.

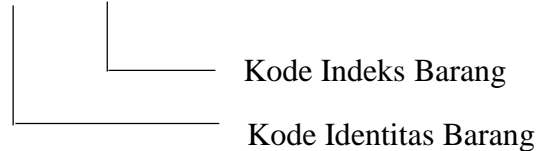
III.1.3.4 Analisis Kebutuhan Pengkodean

Adapun kebutuhan pengkodean kantor BPPTKPK adalah pengkodean barang inventaris sesuai dengan kode ruangan.

1. Fakta pengkodean

Sesuai analisis prosedur yang sedang berjalan pada proses pendataan barang inventarisasi di kantor BPPTKPK, didapatkan bahwa pengkodean barang selama ini mengikuti id produk dari setiap barang yang hanya diketahui oleh petugas biro dengan format pengkodean barang yaitu kode identitas barang/indes pengadaan. Contoh kode barang mesin bubut yaitu “2030101001”.

Contoh Pengkodean No 2030101-001



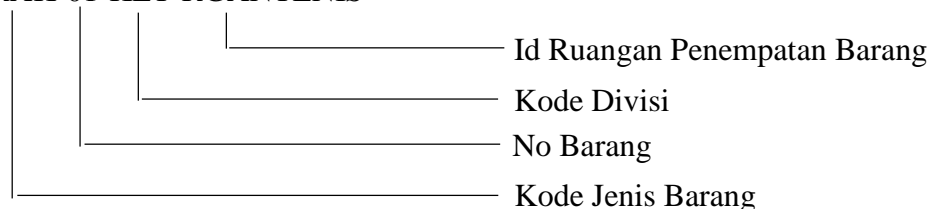
Pegawai inventaris hanya mencatat data barang berdasarkan nama barang dan merk barang saja sehingga dapat disimpulkan bahwa pada pendataan barang inventarisasi di kantor BPPTKPK belum memiliki pengkodean.

2. Kebutuhan Sistem

Dalam aplikasi inventarisasi barang yang dibangun perlu adanya pengkodean barang dengan maksud mempermudah pegawai inventaris ketika mengacu suatu barang di dalam sistem, dan mencegah kesalahan pengacuan barang. Sebagai contoh format pengkodean barang yang dibutuhkan oleh sistem adalah sebagai berikut

Contoh Kebutuhan Pengkodean

RAK-01-KET-RGANTENIS



3. Kesimpulan

Perlu diterapkannya analisis kebutuhan pengkodean data barang, divisi, kategori, merk, dan ruangan dengan maksud untuk mengurangi kemungkinan data yang redudansi.

III.1.4. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Analisis kebutuhan fungsional dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam ke satuan yang utuh dan berfungsi. Tahapan ini menyangkut pengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah dilakukan instalasi dari sistem akan benar-benar memuaskan dari rancangan pembangunan yang telah ditetapkan pada akhir tahap analisis sistem. Alat bantu yang digunakan untuk menggambar sistem secara umum yang akan dibangun yaitu diagram konteks dan UML. Untuk menjelaskan bagaimana suatu masukan diproses pada sistem maka digunakan spesifikasi proses dan kamus data untuk mengetahui aliran data yang mengalir pada sistem.

Berikut adalah kebutuhan fungsional untuk sistem informasi inventaris barang di lembaga BPPTKPK:

1.	SIB-BPPTKPK-F-1-01	Pegawai Inventaris dapat memasukan data barang inventaris
2.	SIB-BPPTKPK-F-1-02	Pegawai Inventaris dapat mengubah data barang inventaris
3.	SIB-BPPTKPK-F-1-03	Pegawai Inventaris dapat memisahkan barang sesuai dengan keadaan barang tertentu
4.	SIB-BPPTKPK-F-1-04	Pegawai Inventaris dapat mencetak laporan data barang inventaris berdasarkan ruangan dan jangka waktu tertentu
5.	SIB-BPPTKPK-F-1-05	Pegawai Inventaris dapat memasukan data divisi baru
6.	SIB-BPPTKPK-F-1-06	Pegawai Inventaris dapat mengubah data divisi
7.	SIB-BPPTKPK-F-1-07	Pegawai Inventaris dapat memasukan data ruangan baru
8.	SIB-BPPTKPK-F-1-08	Pegawai Inventaris dapat mengubah data ruangan
9.	SIB-BPPTKPK-F-1-09	Pegawai Inventaris dapat memasukan data merk baru
10.	SIB-BPPTKPK-F-1-10	Pegawai Inventaris dapat mengubah data merk
11.	SIB-BPPTKPK-F-1-11	Pegawai Inventaris dapat memasukan data kategori baru
12.	SIB-BPPTKPK-F-1-12	Pegawai Inventaris dapat mengubah data kategori

13.	SIB-BPPTKPK-F-1-13	Pegawai Inventaris dapat mengelola pengeluaran barang inventaris
14.	SIB-BPPTKPK-F-1-14	Pegawai Inventaris dapat mengelola pemeriksaan barang inventaris
15.	SIB-BPPTKPK-F-1-15	Pegawai Inventaris dapat mencetak surat pengajuan barang untuk diserahkan kepada biro aset
16.	SIB-BPPTKPK-F-2-01	Biro Aset dapat melakukan <i>filtering</i> terhadap data inventaris barang di semua ruangan kantor BPPTKPK
17.	SIB-BPPTKPK-F-2-02	Biro Aset dapat mencetak laporan data barang inventaris berdasarkan ruangan dan jangka waktu tertentu

III.2. Model Use Case

Model use case merupakan pemodelan struktural yang mencerminkan fungsionalitas sistem. Model use case juga menunjukkan apa yang bisa dilakukan oleh sistem

III.2.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah system. Yang ditekankan adalah “apa” yang diperbuat oleh system, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan system.



Gambar III. 4 Use Case Diagram

III.2.2. Definisi Aktor

Definisi aktor dibuat untuk menggambarkan aktor yang terlibat pada berjalannya sistem inventarisasi barang BPPTKPK dan deskripsi *role* untuk aktor tersebut. Deskripsi *role* menjelaskan wewenang yang dapat dilakukan oleh aktor dalam aplikasi. Berikut adalah tabel definisi aktor :

Tabel III. 1. Definisi Aktor

No	Aktor	Deskripsi
1.	Pegawai	Aktor dengan role ini mempunyai wewenang untuk melakukan login, tambah data barang, ubah data barang, tambah data divisi, <i>view</i> barang, filter barang dan cetak laporan.
2.	Biro Aset	Aktor dengan role ini hanya dapat mengakses <i>view</i> barang, filter barang, dan cetak laporan.

III.2.3. Definisi Use Case

Bagian ini berisi daftar use case dan deskripsi singkat dari setiap use case. Berikut adalah tabel definisi use case :

Tabel III. 2. Definisi Use Case

No	Aktor	Deskripsi
1.	Login	Sistem akan memeriksa data ID dan password yang telah diinputkan oleh pegawai inventaris. Jika data yang diinputkan sesuai maka pegawai inventaris dapat masuk ke dalam sistem dan melakukan proses selanjutnya. Dan jika data yang diinputkan tidak sesuai maka aplikasi akan mengirimkan pesan error kepada pengguna dan pengguna tidak dapat melakukan proses selanjutnya.
2.	View Barang	Sistem akan menampilkan data barang sesuai dengan ruangan yang dimaksud oleh pegawai.
3.	Cetak Laporan	Sistem akan menampilkan optional untuk mencetak laporan dengan format Html ataupun Excel. Jika format sudah dipilih oleh pegawai sistem akan meng-export file dengan format yang telah dipilih.
4.	Filter Barang	Sistem akan menampilkan data barang sesuai dengan kondisi yang diinputkan pegawai terhadap sistem.
5.	Ubah Data Barang	Sistem akan melakukan perubahan data barang yang sebelumnya sudah tersimpan di database sesuai inputan pegawai. Jika data barang lengkap maka record barang akan diubah dan disimpan ke dalam database. Dan jika data barang tidak lengkap maka sistem ini akan mengirimkan pesan error kepada pegawai dan data tidak tersimpan ke

No	Aktor	Deskripsi
		dalam database.
6.	Pengeluaran Barang	Sistem akan melakukan pelabelan ketidak eksistensian data barang. Jika data pengeluaran barang diinputkan dengan lengkap oleh pegawai maka data barang akan diberi label tidak eksis. Dan jika data pengeluaran barang tidak lengkap maka sistem akan mengirimkan pesan error kepada pegawai dan data barang tidak diberi label tidak eksis.
7.	Pengajuan Barang	Sistem akan membuat file laporan pengajuan barang yang berekstensi .doc. Jika data barang yang diajukan diinputkan secara lengkap oleh pegawai maka file laporan akan diciptakan. Dan jika data barang yang diajukan tidak diinputkan dengan lengkap maka file laporan tidak akan diciptakan.
8.	Penerimaan Barang	Sistem akan menyimpan data barang yang diterima oleh pegawai inventaris ke dalam sistem sesuai dengan ruangan penempatan ruangan. Jika data barang diinputkan dengan lengkap oleh pegawai maka data barang akan disimpan. Dan jika data barang tidak diinputkan dengan lengkap maka sistem akan memunculkan pesan error kepada pegawai dan data barang tidak disimpan.
9.	Pengolahan Data Master	Sistem akan menyimpan data master yang dibutuhkan sebagai data pendukung yang dibutuhkan sistem sesuai dengan inputan pegawai inventaris terhadap <i>form</i> inputan data master yang meliputi data divisi, ruangan, merk dan kategori.

III.2.4. Skenario Use Case

Skenario Use Case merupakan *flow of event* untuk use case utama yang dapat menggambarkan urutan interaksi aktor dengan use case tersebut dimulai dari awal aktor berinteraksi hingga selesai. Berikut adalah scenario use case yang terbentuk :

III.2.4.1 Use Case Login

Use case *login* digunakan pegawai inventaris ketika melakukan autentikasi pengguna terhadap sistem.

Usecase name	Login	
Related Requirement	-	
Goal In Context	Login berhasil dan masuk kedalam system	
Precondition	Pegawai inventaris memasukkan username, password serta hak akses	
Success end condition	Jika <i>login</i> berhasil maka akan masuk kedalam system	
Failed end condition	Jika <i>login</i> gagal maka akan tidak akan masuk kedalam sistem	
Primary Aktor	Pegawai inventaris	
Main Flow	Step	Action
Extentions	1	Pegawai inventaris memilih menu login
	2	Sistem menampilkan <i>form</i> login
	3	Pegawai inventaris mengisi username dan password
	4	Klik tombol login
	5	Sistem akan memeriksa/memverifikasi username dan password yang diinputkan pegawai inventaris.
	Step	Branching Actions
	5.1	Jika username dan password valid maka akan masuk kedalam system
	5.2	Jika username dan password invalid maka muncul pesan Gagal Login

III.2.4.2 Use Case *View Barang*

Use case *view* barang digunakan oleh pegawai ketika akan menampilkan data barang pada suatu ruangan.

Usecase name	<i>View</i> Barang	
Related Requirement	-	
Goal In Context	Data barang ditampilkan sesuai id ruangan yang dipilih	
Precondition	Pegawai inventaris memilih ruangan yang ingin dilihat data barangnya	
Success end condition	Jika pegawai inventaris sudah masuk ke dalam <i>form</i> barang sesuai id ruangan maka data barang yang berada pada ruangan tersebut ditampilkan.	
Failed end condition	Jika pegawai inventaris gagal masuk ke dalam <i>form</i> barang maka data barang tidak dapat ditampilkan.	
Primary Aktor	Pegawai inventaris	
Main Flow	Step	Action
Extentions	1	Pegawai inventaris memilih ruangan yang akan dilihat daftar barang inventarisnya
	2	Sistem mengarahkan pegawai inventaris kepada <i>Form</i> Barang dengan membuka akses sesuai dengan hak akses pegawai inventaris
	3	Data barang akan ditampilkan sesuai id ruangan
	Step	Branching Actions
	2.1	Jika pegawai inventaris adalah pegawai inventaris maka sistem akan membuka akses tambah barang, update barang, pengolahan data master dan cetak laporan
	2.2	Jika pegawai inventaris adalah pegawai inventaris umum atau Biro Aset sistem akan mengarahkan pegawai inventaris kepada <i>Form</i> Barang dengan hanya membuka akses filter barang dan cetak laporan

III.2.4.3 Use Case:Cetak Laporan

Use case cetak laporan digunakan ketika pegawai inventaris akan mengambil data barang sebagai laporan data barang inventaris.

Usecase name	Cetak Laporan	
Related Requirement	-	
Goal In Context	File laporan data barang berhasil diciptakan oleh sistem	
Precondition	Pegawai inventaris sudah memilih format laporan dan menekan tombol “Export”	
Success end condition	Jika cetak laporan berhasil dieksekusi maka file laporan data barang diciptakan oleh sistem	
Failed end condition	Jika cetak laporan gagal dieksekusi maka file laporan data barang tidak diciptakan oleh sistem	
Primary Aktor	Pegawai inventaris	
Main Flow	Step	Action
Extentions	1	Pegawai inventaris berada di <i>Form</i> Barang
	2	Pegawai inventaris menekan tombol “Print”
	3	Sistem menampilkan optional format untuk export laporan
	4	Pegawai inventaris memilih format dan menekan tombol “Export”
	5	Sistem menciptakan file dengan format yang telah dipilih yang berisi data barang.

III.2.4.4 Use Case: Filter Barang

Use case filter barang digunakan oleh pegawai inventaris ketika akan melihat data barang sesuai dengan suatu kondisi tertentu.

Usecase name	Filter Barang	
Related Requirement	-	
Goal In Context	Data barang yang berada pada <i>form</i> barang ditampilkan sesuai dengan kondisi filter yang dipilih	
Precondition	Pegawai inventaris sudah menginputkan kondisi filter barang	
Success end condition	Jika kondisi filter barang telah diinputkan maka data barang akan ditampilkan sesuai dengan kondisi inputan	
Failed end condition	Jika kondisi filter barang tidak diinputkan maka filter barang tidak berhasil	
Primary Aktor	Pegawai inventaris	
Main Flow	Step	Action
Extentions	1	Pegawai inventaris berada di <i>Form</i> Barang
	2	Pegawai inventaris menekan tombol “Filter” dan kemudian memilih kondisi yang diinginkan beserta <i>value</i> -nya laporan
	3	Sistem akan menampilkan data barang sesuai dengan kondisi yang ditentukan pegawai inventaris

III.2.4.5 Use Case Ubah Data Barang

Use case ubah data barang digunakan oleh pegawai ketika mengubah data suatu barang.

Usecase name	Ubah Data Barang	
Related Requirement	-	
Goal In Context	Data barang yang dipilih oleh pegawai berhasil berubah	
Precondition	Pegawai menginputkan data barang yang akan diubah dengan lengkap dan menekan tombol “OK”	
Success end condition	Jika pengubahan data barang berhasil dieksekusi maka sistem akan mengubah data barang yang dipilih sesuai inputan pegawai	
Failed end condition	Jika pengubahan data barang gagal dieksekusi maka sistem tidak akan mengubah data barang	
Primary Aktor	Pegawai Inventaris	
Main Flow	Step	Action
Extentions	1	Pegawai berada di <i>Form</i> Barang
	2	Pegawai memilih <i>record</i> barang yang akan diubah kemudian menekan tombol “Update”
	3	Sistem akan membuka <i>form</i> pengubahan barang
	4	Pegawai mengisi <i>form</i> pengubahan barang dengan data barang yang terbaru dan kemudian menekan tombol “OK”
	5	Sistem mengupdate data barang
	Step	Branching Actions
	5.1	Sistem menampilkan pesan error pengupdatean
	5.2	Pegawai mengisi kembali <i>form</i> barang dengan data barang dan kemudian menekan tombol “OK”

III.2.4.6 Use Case Pengeluaran Barang

Use case pengeluaran barang digunakan oleh pegawai ketika mengeluarkan barang dari suatu ruangan karena kondisi tertentu.

Usecase name	Pengeluaran Barang	
Related Requirement	-	
Goal In Context	Data barang berhasil diberi label tidak eksis	
Precondition	Pegawai memilih data barang yang akan dikeluarkan dan kemudian menekan tombol “Keluarkan Barang”	
Success end condition	Jika pengeluaran barang berhasil dieksekusi maka data barang yang dipilih oleh pegawai akan diberi label tidak eksis	
Failed end condition	Jika pengeluaran barang gagal dieksekusi maka data barang tidak akan diberi label	
Primary actor	Pegawai	
Main Flow	Step	Action
Extentions	1	Pegawai berada di <i>Form</i> pengeluaran barang
	2	Pegawai memilih data barang yang akan dikeluarkan
	3	Pegawai menekan tombol “Keluarkan Barang”
	4	Pegawai menginputkan jumlah barang yang dikeluarkan
	5	Pegawai menekan tombol “OK”
	Step	Branching Actions
	5.1	Sistem menampilkan pesan error pengeluaran barang
	5.2	Pegawai menginputkan jumlah barang kembali dan menekan tombol “OK”

III.2.4.7 Use Case Pengajuan Barang

Use case pengajuan barang digunakan oleh pegawai inventaris ketika akan mengajukan barang kepada petugas biro.

Usecase name	Pengajuan Barang	
Related Requirement	-	
Goal In Context	File surat pengajuan barang diciptakan oleh sistem.	
Precondition	Pegawai inventaris menginputkan data barang yang diajukan dengan lengkap.	
Success end condition	Jika pengajuan barang berhasil dieksekusi maka file surat pengajuan barang akan diciptakan oleh sistem.	
Failed end condition	Jika pengajuan barang gagal dieksekusi maka file surat pengajuan barang tidak akan diciptakan oleh sistem.	
Primary actor	Pegawai	
Main Flow	Step	Action
Extentions	1	Pegawai berada di <i>Form</i> Barang
	2	Pegawai menekan tombol export
	3	Pegawai memilih format surat pengajuan barang
	4	Pegawai menginputkan data barang yang diajukan Pegawai menekan tombol “OK”
	Step	Branching Actions
	4.1	Sistem menampilkan pesan error pengajuan barang
	4.2	Pegawai mengisi kembali data barang yang diajukan dan menekan tombol “OK”

III.2.4.8 Use Case Penerimaan Barang

Use case penerimaan barang digunakan oleh pegawai biro ketika menerima barang dari petugas biro dan akan mendistribusikan ke tiap-tiap ruangan.

Usecase name	Penerimaan Barang	
Related Requirement	-	
Goal In Context	Data barang berhasil tersimpan ke dalam sistem.	
Precondition	Pegawai inventaris menginputkan data barang dengan lengkap.	
Success end condition	Jika penerimaan barang berhasil dieksekusi maka data barang tersimpan ke dalam sistem.	
Failed end condition	Jika penerimaan barang gagal dieksekusi maka data barang tidak tersimpan ke dalam sistem dan menampilkan pesan error.	
Primary actor	Pegawai	
Main Flow	Step	Action
Extentions	1	Pegawai berada di <i>Form</i> Barang
	2	Pegawai menekan tombol tambah
	3	Pegawai menginputkan data barang
	4	Pegawai menekan tombol “OK”
	Step	Branching Actions
	4.1	Sistem menampilkan pesan error penambahan barang
	4.2	Pegawai mengisi kembali data barang dan menekan tombol “OK”

III.2.4.9 Use Case Pengolahan Data Master

Use case pengolahan data master digunakan oleh pegawai biro ketika melakukan penambahan, pengubahan dan penghapusan data master yang dibutuhkan sistem seperti data divisi, ruangan, merk, dan kategori.

Usecase name	Pengolahan Data Master	
Related Requirement	-	
Goal In Context	Penambahan atau perubahan data master berhasil disimpan ke dalam sistem.	
Precondition	Pegawai inventaris memilih optional data yang akan diolah, dan mengisi <i>form</i> inputan data master dengan lengkap.	
Success end condition	Jika pengolahan data master berhasil dieksekusi maka penambahan atau perubahan data master akan disimpan ke dalam sistem.	
Failed end condition	Jika pengolahan data master gagal dieksekusi maka penambahan atau perubahan data master tidak akan disimpan ke dalam sistem dan sistem akan menampilkan pesan error.	
Primary actor	Pegawai	
Main Flow	Step	Action
Extentions	1	Pegawai berada di <i>Form</i> Menu
	2	Pegawai memilih optional pengolahan data master pada menu strip
	3	Pegawai menginputkan data pada <i>form</i> inputan data master yang telah dipilih.
	4	Pegawai menekan tombol “OK”
	Step	Branching Actions
	4.1	Sistem menampilkan pesan error pengolahan data master
	4.2	Pegawai mengisi kembali data pada <i>form</i> inputan data master dan menekan tombol “OK”

III.3. Model Analisis

Model analisis adalah tahap analisa dasar sistem. Berikut adalah model analisis dari sistem yang akan dibangun.

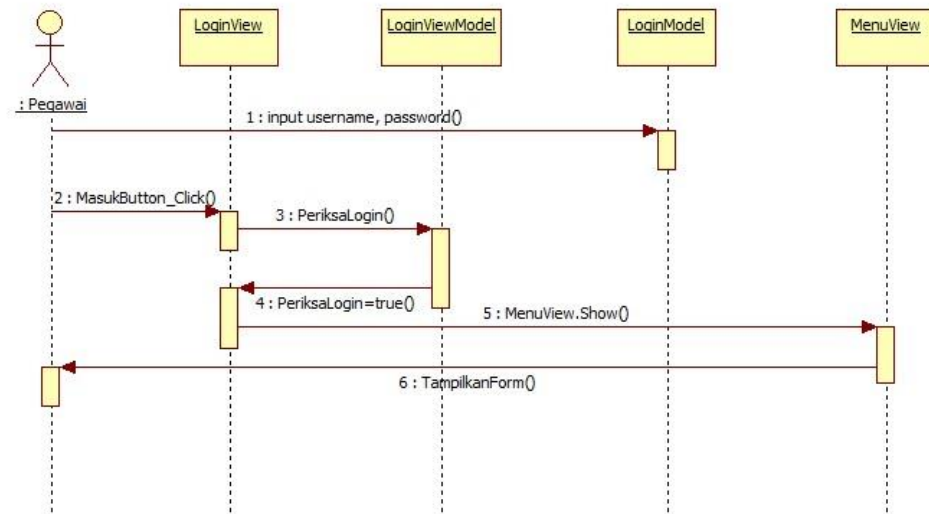
III.3.1 Realisasi Use Case Tahap Analisis

Bagian ini dibuat untuk memperjelas setiap use case yang dibuat. Setiap subbab use case diisi dengan diagram kelas analisis dan *Sequence Diagram* yang menggambarkan interaksi setiap objek dari kelas analisis yang terlibat di dalam use case tersebut.

III.3.1.1. Use Case Login

Pada use case login, sistem akan melakukan autentikasi pengguna berdasarkan inputan username dan password. User case *login* melibatkan *Class* LoginView, LoginViewModel, LoginModel, MenuView.

Sequence Diagram



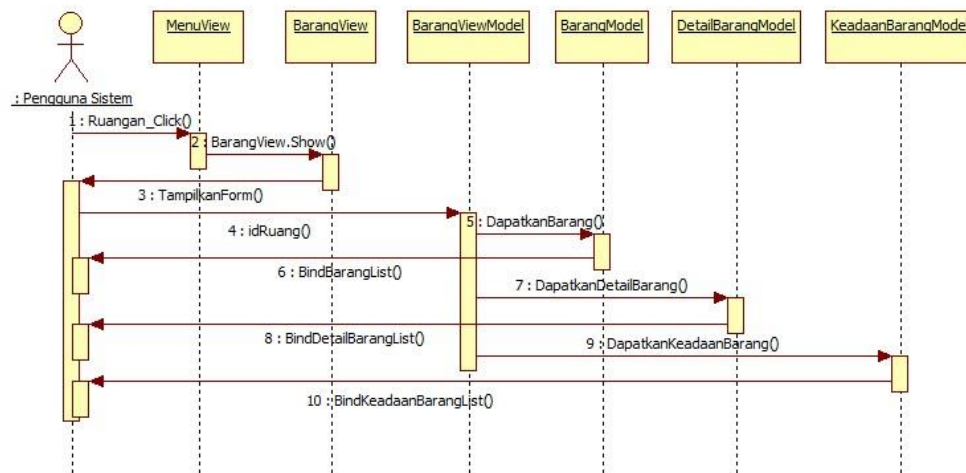
Gambar III. 5 *Sequence Diagram* Login

- a. Pegawai memasukkan username dan password pada *Form* Login
- b. Pegawai menekan tombol “Masuk”
- c. Kemudian *Form login* melakukan pemeriksaan data login
- d. Jika nilai balikkan pemeriksaan data *login* adalah “true” *Form* Menu akan ditampilkan kepada pegawai

III.3.1.2. Use Case View Barang

Pada use case *view* barang, sistem merespon inputan pilihan ruangan kepada pegawai dengan menampilkan data barang yang berada di ruangan tersebut. Use case ini melibatkan *Class* MenuView, BarangView, BarangViewModel, BarangModel, DetailBarangModel, KeadaanBarangModel.

Sequence Diagram



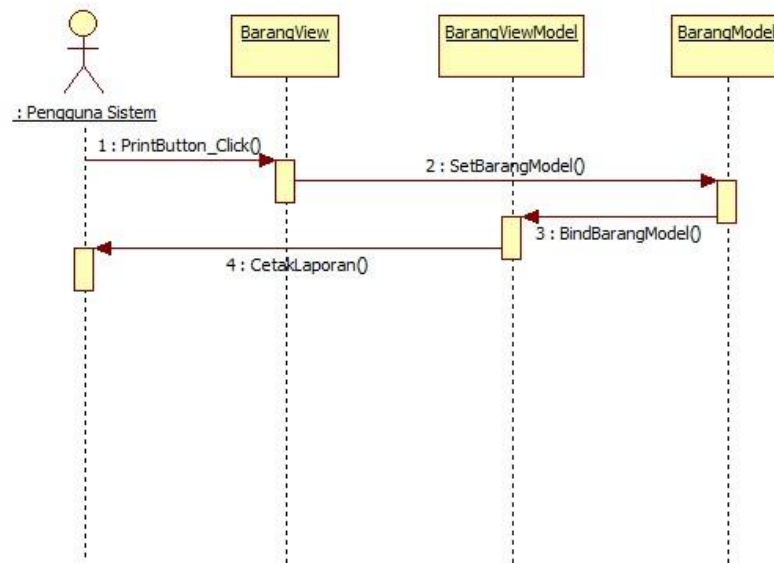
Gambar III. 6 Sequence Diagram View Barang

1. Pegawai inventaris melakukan klik pada suatu ruangan
2. *Form* Barang ditampilkan pada pegawai inventaris
3. *idRuang* dikirimkan kepada *Form* Barang dan kemudian diolah oleh *Class* *BarangViewModel*
4. *Class* *BarangViewModel* melakukan pemanggilan prosedur *DapatkanBarang()* yang memiliki nilai balikan *ListBarang* yang merupakan objek dari *Class* *BarangModel*
5. *Class* *DetailBarangViewModel* melakukan pemanggilan prosedur *DapatkanDetailBarang()* yang memiliki nilai balikan *ListDetailBarang* yang merupakan objek dari *Class* *DetailBarangModel*
6. *Class* *KeadaanBarangViewModel* melakukan pemanggilan prosedur *DapatkanKeadaanBarang()* yang memiliki nilai balikan *ListKeadaanBarang* yang merupakan objek dari *Class* *KeadaanBarangModel*
7. *Form* Barang melakukan *binding* terhadap *list-list* tersebut

III.3.1.3. Use Case Cetak Laporan

Pada use case cetak laporan, pegawai memilih format keluaran laporan dan sistem akan merespon dengan menciptakan file dengan format inputan pegawai yang berisi dengan data barang. Use case ini melibatkan *Class* *BarangView*, *BarangViewModel*, *BarangModel*.

Sequence Diagram



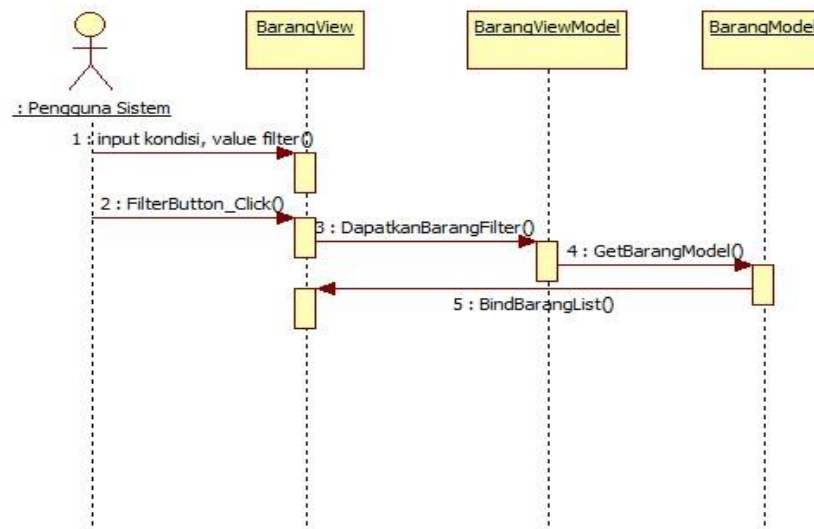
Gambar III. 7 Sequence Diagram Cetak Laporan

1. Pegawai inventaris menekan tombol “Print” dan dengan secara langsung menjalankan prosedur *SetBarangModel()* dengan data barang dari *datagridview* yang berada di *Form* *Barang*.
2. Setelah itu *BarangViewModel* melakukan *binding* terhadap objek *Class* *BarangModel*.
3. Setelah itu *BarangViewModel* melakukan prosedur *CetakLaporan()*.

III.3.1.4. Use Case Filter Barang

Pada use case *filter* barang, pegawai memilih kondisi *filter* berdasarkan kolom dan *value* kolom. Setelah pegawai menginputkan pilihan data barang akan dilakukan *filter* oleh sistem. Kelas yang terlibat dalam use case ini adalah *BarangView*, *BarangViewModel*, *BarangModel*.

Sequence Diagram



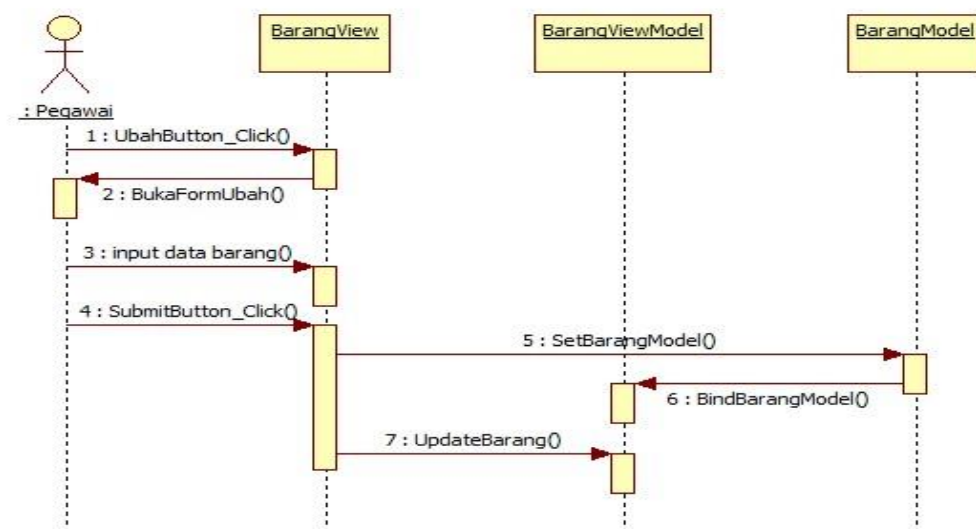
Gambar III. 8 Sequence Diagram Filter Barang

1. Pegawai inventaris melakukan inputan kondisi dan value dari kolom yang akan menjadi parameter filter.
2. Pegawai inventaris menekan tombol “Filter”
3. *Form* Barang melakukan prosedur *DapatkanBarangFilter()* yang berisi perintah *BarangViewModel* untuk melakukan *binding* data barang dengan kondisi filter kepada *Form* Barang.

III.3.1.5. Use Case Ubah Data Barang

Pada use case ubah data barang, pegawai menginputkan data barang yang baru ke dalam *form* pengubahan data barang kemudian sistem akan merespon dengan meng-update data barang ke dalam database. Kelas yang terlibat adalah *BarangView*, *BarangViewModel*, dan *BarangModel*.

Sequence Diagram



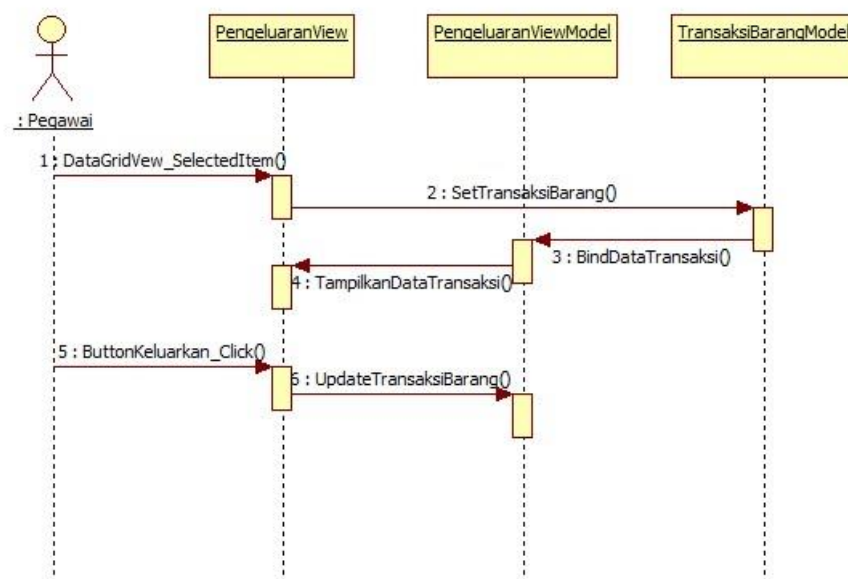
Gambar III. 9 Sequence Diagram Ubah Data Barang

1. Pegawai menekan tombol “Ubah” dan kemudian *form* pengubahan barang pun dibuka oleh *Form* Barang.
2. Pegawai menginputkan data barang yang baru.
3. Kemudian pegawai menekan tombol “Submit” yang berisi perintah untuk menerapkan value kepada objek *Class* *BarangModel* dan kemudian objek tersebut dipakai sebagai parameter masukkan dari prosedur *UpdateBarang()*

III.3.1.6. Use Case Pengeluaran Barang

Pada use case pengeluaran barang, pegawai memilih data barang yang akan dikeluarkan pada *form* pengeluaran barang dan kemudian mengeksekusi pengeluaran barang. Setelah itu sistem akan memberi label ketidak eksistensian menjadi tidak eksis. Kelas yang terlibat adalah PengeluaranView, PengeluaranViewModel dan TransaksiBarangModel.

Sequence Diagram



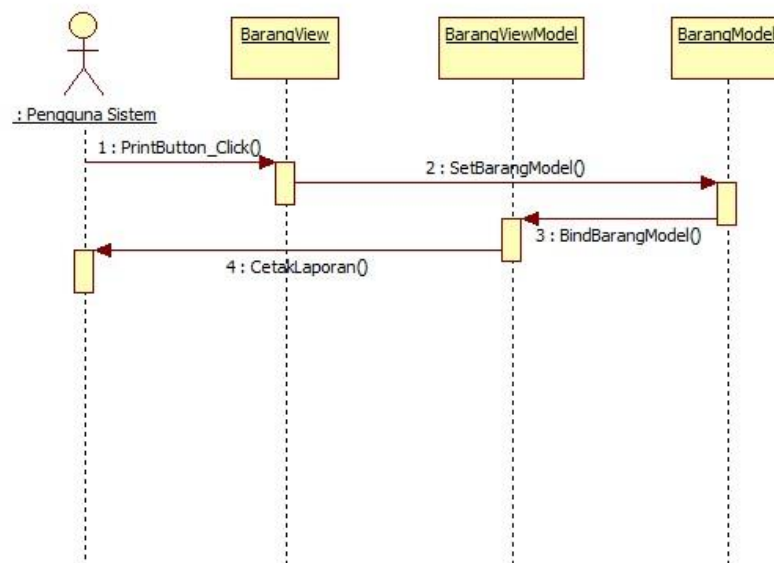
Gambar III.10 Sequence Diagram Pengeluaran Barang

1. Pegawai memilih data barang yang akan dikeluarkan
2. Pegawai menekan tombol Keluarkan
3. Kemudian sistem akan memberi label tidak eksis pada barang tersebut

III.3.1.7. Use Case Pengajuan Barang

Pada use case pengajuan barang, pegawai memilih format laporan pengajuan barang dan sistem akan merespon dengan menciptakan file dengan format inputan pegawai yang berisi dengan data barang yang diajukan oleh pegawai. Use case ini melibatkan *class* *BarangView*, *BarangViewModel*, *BarangModel*.

Sequence Diagram



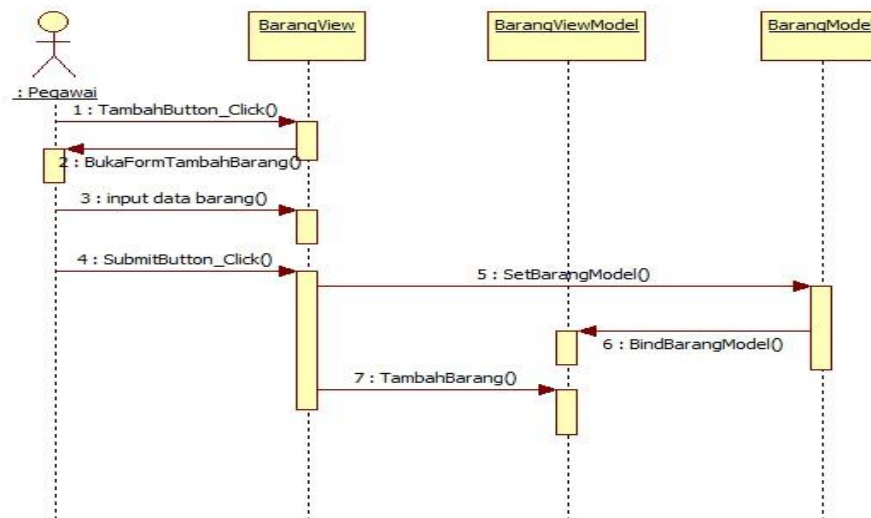
Gambar III.11 Sequence Diagram Cetak Pengajuan Barang

1. Operator menekan tombol “Print” dan dengan secara langsung menjalankan prosedur *SetBarangModel()* dengan data barang dari datagridview yang berada di *Form* *Barang*.
2. Setelah itu *BarangViewModel* melakukan *binding* terhadap objek *Class* *BarangModel*.
3. Setelah itu *BarangViewModel* melakukan prosedur *CetakLaporan()*

III.3.1.8. Use Case Penerimaan Barang

Pada use case penerimaan barang, pegawai menginputkan data barang ke dalam *form* penerimaan barang kemudian sistem akan merespon dengan menyimpan data barang ke dalam *database*. Kelas yang terlibat dalam use case ini adalah *BarangView*, *BarangViewModel*, dan *BarangModel*.

Sequence Diagram



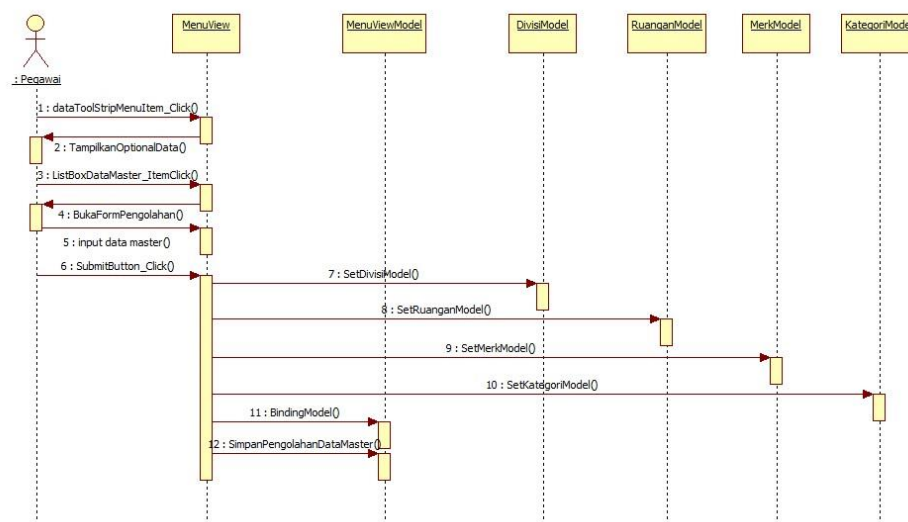
Gambar III. 12 Sequence Diagram Penerimaan Barang

1. Pegawai menekan tombol “Tambah” yang berisi perintah untuk membuka *form* inputan penerimaan barang.
2. Kemudian pegawai menginputkan data barang pada *form* inputan penerimaan barang.
3. Pegawai menekan tombol “Submit” yang berisi perintah untuk menerapkan value kepada objek *Class* *BarangModel* dan kemudian objek tersebut dipakai sebagai parameter masukkan pada prosedur *TambahBarang()*.

III.3.1.9. Use Case Pengolahan Data Master

Pada use case pengolahan data master, pegawai memilih menu pengolahan data master dan kemudian menginputkan data pengolahan data master ke dalam form inputan data master, kemudian sistem akan merespon dengan menyimpan data master ke dalam database. Kelas yang terlibat dalam use case ini adalah MenuView, MenuViewModel, DivisiModel, RuanganModel, KategoriModel, dan MerkModel.

Sequence Diagram



Gambar III. 13 Sequence Diagram Pengolahan Data Master

1. Pegawai menekan memilih optional pengolahan data master pada menu strip. Dan sistem akan menampilkan optional data yang dapat diolah kepada pegawai inventaris.
2. Setelah pegawai memilih data yang akan diolah, kemudian sistem akan menampilkan *form* inputan pengolahan data master sesuai pilihan pegawai.
3. Pegawai menekan tombol “Submit” setelah menginputkan data ke dalam *form* inputan dengan lengkap. Setelah itu data akan diterapkan kepada *class* tujuan dari inputan data master dan data akan di binding ke dalam MenuViewModel yang kemudian data akan disimpan ke dalam database dengan prosedur SimpanPengolahanDataMaster().

III.3.1.10. Class Terlibat

Berikut adalah tabel dari setiap *Class* yang terlibat pada masing-masing use case yang terbentuk:

Tabel III. 3 Class yang terlibat

Nama Class	Class	Terlibat pada Use Case
<i>LoginView Class</i>	<div> LoginView +hakAkses +Username_Click() +Password_Click() +Username_Leave() +Password_Leave() </div>	- Login
<i>LoginViewModel Class</i>	<div> LoginViewModel +LoginViewModel(Model.User) +PeriksaLogin() +RaisePropertyChanged() </div>	- Login
<i>UserModel Class</i>	<div> UserModel +username +password +Username() +Password() </div>	- Login
<i>MenuView Class</i>	<div> MenuView +hakAkses +TambahDivisi_Click() +UbahDivisi_Click() +TambahRuangan_Click() +UbahRuangan_Click() +TambahKategori_Click() +UbahKategori_Click() +TambahMerk_Click() +UbahMerk_Click() +Ruangan_Click() </div>	- View Barang - Pengolahan Data Master

Nama Class	Class	Terlibat pada Use Case
<i>MenuViewModel</i> Class	<div data-bbox="523 324 1082 1142"> <p>MenuViewModel</p> <p>+sqlConnection +divList +katList +ruangList +merkList</p> <p>+MenuViewModel() +TambahkanDataDivisi(Model.Divisi) +DapatkanKode(namaDivisi) +DapatkanNama(kodeDivisi) +TambahkanDataKategori(Model.Kategori) +DapatkanId(namaKategori) +DapatkanNama(idKategori) +TambahkanDataRuangan(Model.Ruangan) +DapatkanId(namaRuangan) +DapatkanNama(idRuangan) +TambahkanDataMerk(Model.Merk) +DapatkanId(namaMerk) +DapatkanNama(idMerk) +UpdateDataKategori(Model.Kategori) +UpdateDataDivisi(Model.Divisi) +UpdateDataRuangan(Model.Ruangan) +UpdateDataMerk(Model.Merk)</p> </div>	<ul style="list-style-type: none"> - View Barang - Pengolahan Data Master
<i>BarangView</i> Class	<div data-bbox="651 1182 954 1668"> <p>BarangView</p> <p>+kodeBarang +dataFilter +bagian +hakAkses</p> <p>+ButtonUpdate_Click() +ButtonTambah_Click() +ButtonOK_Click() +ButtonExport_Click() +ButtonClose_Click() +ButtonFilter_Click() +Grid_RowSelected() +ButtonValid_Click()</p> </div>	<ul style="list-style-type: none"> - View Barang - Cetak Laporan - Filter Barang - Tambah Data Barang - Ubah Data Barang - Pemeriksaan Barang

Nama Class	Class	Terlibat pada Use Case
BarangViewModel Class	<div data-bbox="544 324 1061 817"> <p>BarangViewModel</p> <p>+sqlConnection +brgList +detBrgList +keadBrgList</p> <p>+TambahBarang(Model.Barang) +TambahDetailBarang(Model.DetailBarang) +TambahKeadaanBarang(Model.KeadaanBarang) +UpdateBarang(Model.Barang) +UpdateDetailBarang(Model.DetailBarang) +UpdateKeadaanBarang(Model.KeadaanBarang) +DapatkanBarang(idRuang) +DapatkanDetailBarang(idRuang) +DapatkanKeadaanBarang(idRuang) +DapatkanBarangFilter(kolom, value, idRuang) +CetakLaporan(Model.Barang)</p> </div>	<ul style="list-style-type: none"> - View Barang - Cetak Laporan - Filter Barang - Tambah Data Barang - Ubah Data Barang - Pemeriksaan Barang
BarangModel Class	<div data-bbox="679 848 922 1341"> <p>BarangModel</p> <p>+kodeBarang +namaBarang +idMerk +idKategori +kodeDivisi +idRuang</p> <p>+KodeBarang() +NamaBarang() +IdMerk() +IdKategori() +KodeDivisi() +IdRuang()</p> </div>	<ul style="list-style-type: none"> - View Barang - Cetak Laporan - Filter Barang - Tambah Data Barang - Ubah Data Barang - Pemeriksaan Barang

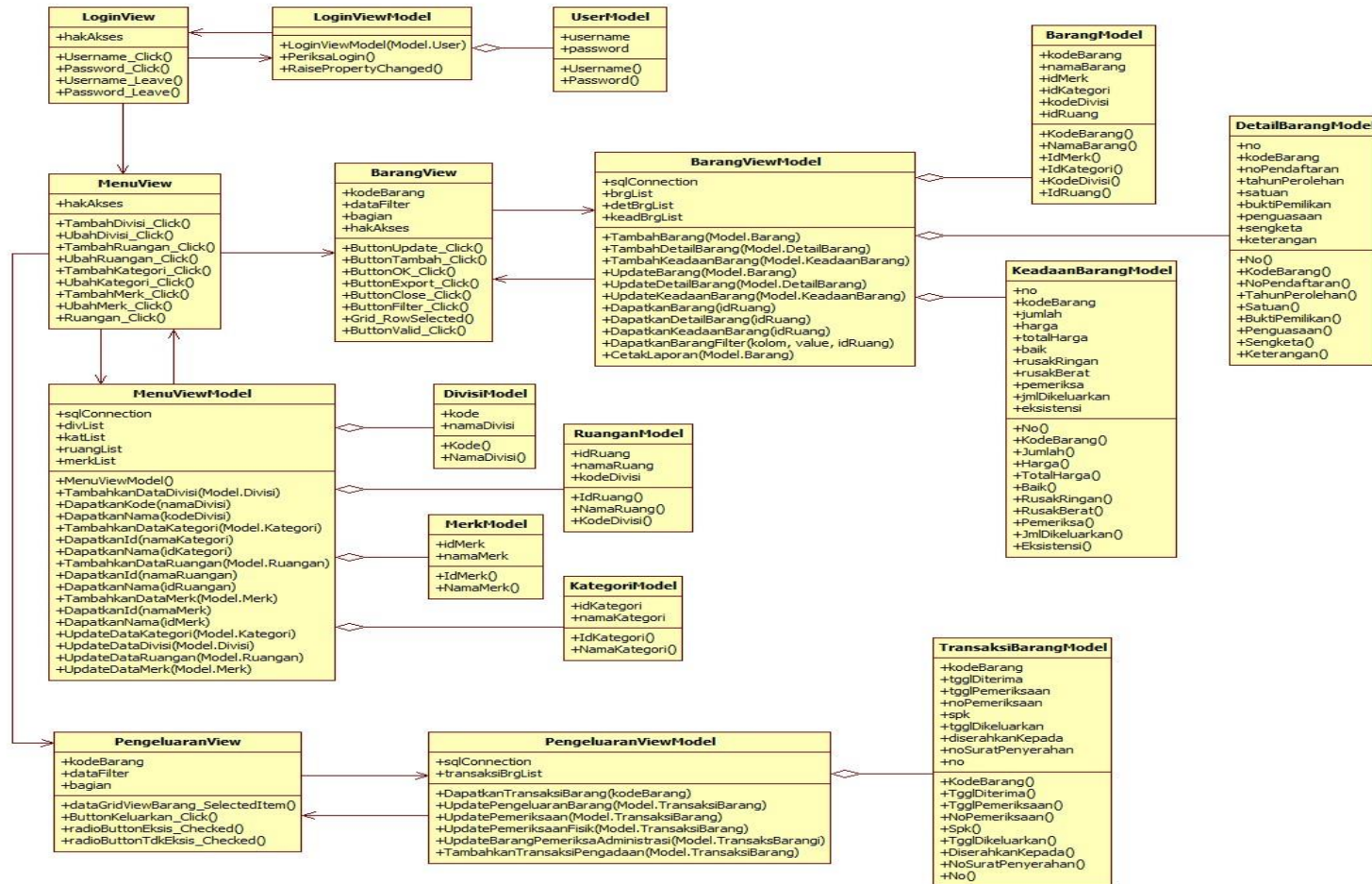
Nama Class	Class	Terlibat pada Use Case
DetailBarangModel Class	<div data-bbox="655 324 949 1019"> DetailBarangModel +no +kodeBarang +noPendaftaran +tahunPerolehan +satuan +buktiPemilikan +penguasaan +sengketa +keterangan +No() +KodeBarang() +NoPendaftaran() +TahunPerolehan() +Satuan() +BuktiPemilikan() +Penguasaan() +Sengketa() +Keterangan() </div>	- View Barang
KeadaanBarangModel Class	<div data-bbox="655 1057 949 1787"> KeadaanBarangModel +no +kodeBarang +jumlah +harga +totalHarga +baik +rusakRingan +rusakBerat +pemeriksa +jmlDikeluarkan +eksistensi +No() +KodeBarang() +Jumlah() +Harga() +TotalHarga() +Baik() +RusakRingan() +RusakBerat() +Pemeriksa() +JmlDikeluarkan() +Eksistensi() </div>	- View Barang

Nama Class	Class	Terlibat pada Use Case
DivisiModel Class	<div>DivisiModel</div> <div>+kode +namaDivisi</div> <div>+Kode() +NamaDivisi()</div>	- MenuViewModel
RuanganModel Class	<div>RuanganModel</div> <div>+idRuang +namaRuang +kodeDivisi</div> <div>+IdRuang() +NamaRuang() +KodeDivisi()</div>	- MenuViewModel
KategoriModel Class	<div>KategoriModel</div> <div>+idKategori +namaKategori</div> <div>+IdKategori() +NamaKategori()</div>	- MenuViewModel
MerkModel Class	<div>MerkModel</div> <div>+idMerk +namaMerk</div> <div>+IdMerk() +NamaMerk()</div>	- MenuViewModel
PengeluaranView Class	<div>PengeluaranView</div> <div>+kodeBarang +dataFilter +bagian</div> <div>+dataGridViewBarang_SelectedItem() +ButtonKeluarkan_Click() +radioButtonEksis_Checked() +radioButtonTdkEksis_Checked()</div>	- Pengeluaran Barang

Nama Class	Class	Terlibat pada Use Case
PengeluaranViewModel Class	<div data-bbox="507 322 1126 622"> PengeluaranViewModel +sqlConnection +transaksiBrgList +DapatkanTransaksiBarang(kodeBarang) +UpdatePengeluaranBarang(Model.TransaksiBarang) +UpdatePemeriksaan(Model.TransaksiBarang) +UpdatePemeriksaanFisik(Model.TransaksiBarang) +UpdateBarangPemeriksaAdministrasi(Model.TransaksiBarang) +TambahkanTransaksiPengadaan(Model.TransaksiBarang) </div>	- Pengeluaran Barang
TransaksiBarangModel Class	<div data-bbox="624 645 978 1346"> TransaksiBarangModel +kodeBarang +tgglDiterima +tgglPemeriksaan +noPemeriksaan +spk +tgglDikeluarkan +diserahkanKepada +noSuratPenyerahan +no +KodeBarang() +TgglDiterima() +TgglPemeriksaan() +NoPemeriksaan() +Spk() +TgglDikeluarkan() +DiserahkanKepada() +NoSuratPenyerahan() +No() </div>	- Pengeluaran Barang

III.3.1.11. Diagram Kelas Keseluruhan

Berikut adalah diagram kelas keseluruhan aplikasi inventarisasi barang BPPTKPK



Gambar III. 14 Class Diagram Keseluruhan

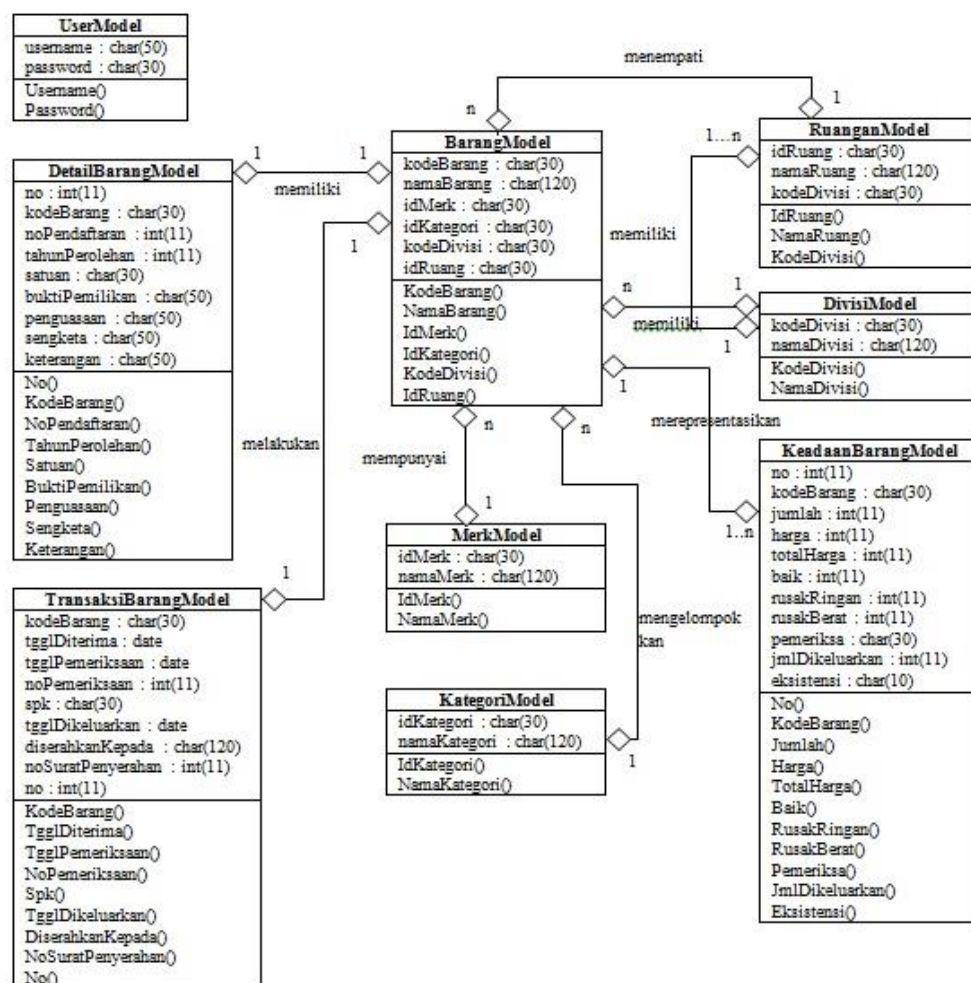
III.3.1.12. Kelompok Kelas dalam Pattern MVC

Sistem informasi inventarisasi barang di BPPTKPK dibangun dengan menerapkan pattern MVC (Model-View-Controller). Dengan MVC data dan method dibagi ke dalam beberapa kelompok *class* yaitu Model, View dan Controller yang masing-masing memiliki peran dalam penanganan data.

1. Model

Model merupakan kelompok *class* yang berhubungan langsung dengan basis data. Model diciptakan untuk mengatur respon terhadap permintaan data, serta memberi hak akses untuk memanipulasi data. *Class* yang termasuk ke dalam kelompok *class* Model yaitu UserModel, BarangModel, DetailBarangModel, KeadaanBarangModel, DivisiModel, RuanganModel, MerkModel, KategoriModel, dan TransaksiBarangModel.

Relasi Antar Class Model



Gambar III. 15 Relasi Antar Class Model

2. *View*

View merupakan *class* yang berada pada lapisan paling atas dan berisi kode yang berhubungan langsung dengan tampilan yang dilihat pengguna. *Class* yang termasuk ke dalam kelompok *class View* adalah *LoginView*, *MenuView*, *BarangView*, dan *PengeluaranView*.

3. *Controller*

Controller merupakan kelompok *class* yang berada di antara Model dan *View*. *Class* yang termasuk ke dalam kelompok *class controller* adalah *LoginViewModel*, *BarangViewModel*, *MenuViewModel*, dan *PengeluaranViewModel*.

III.4. Perancangan Sistem

Perancangan akan dimulai setelah tahap analisis terhadap system selesai dilakukan. Perancangan dapat didefinisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau system dalam detail yang memadai untuk memungkinkan realisasi fisiknya. Perancangan digambarkan sebagai proses multi-langkah dimana representasi perancangan data, skema relasi, struktur file, perancangan antar muka, dan perancangan prosedural dari persyaratan informasi.

III.4.1. Perancangan Data

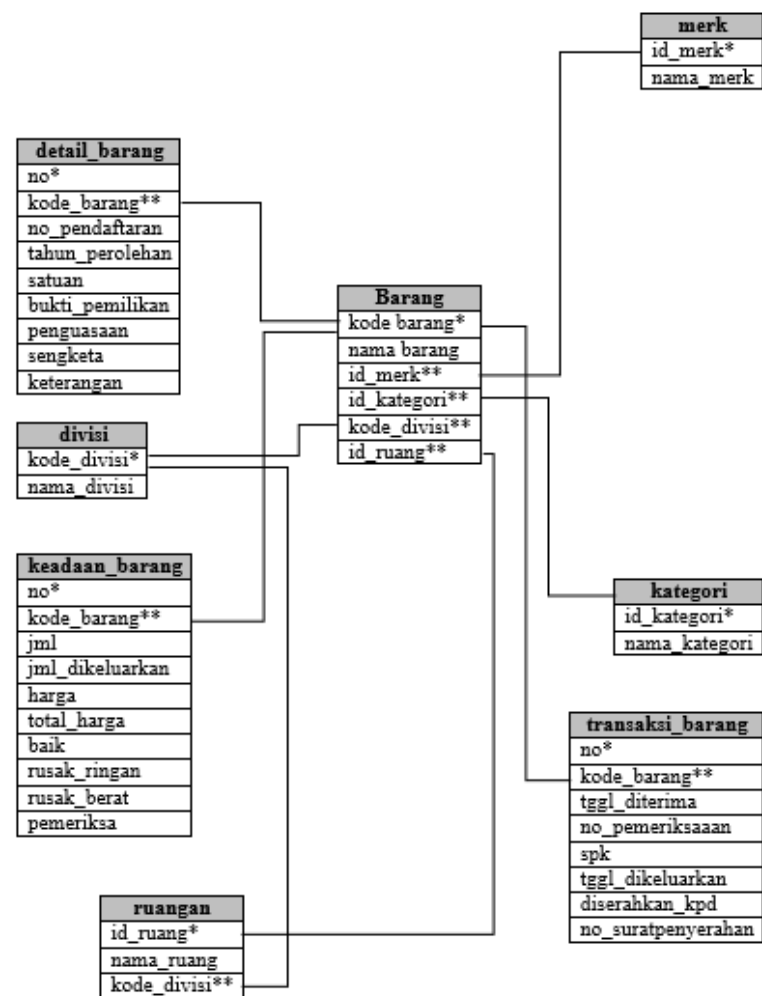
Perancangan data atau lebih dikenal dengan perancangan basis data yaitu menciptakan, atau merancang data yang terhubung dan disimpan secara bersama-sama. Untuk menggambarkannya digunakanlah skema relasi dan struktur *file*. Dan dua hasil tersebut merupakan basis data yang diimplementasikan pada aplikasi inventarisasi barang BPPTKPK.

III.4.1.1 Skema Relasi

Model data relasional merupakan model data di mana hubungan antar data, arti data dan batasannya dijelaskan dengan basis dan kolom. Secara formal, ke semuanya itu digambarkan ke dalam skema relasi dan diagram skema. Adapun skema relasi yang terbentuk adalah sebagai berikut :

1. login{#username, password}

2. barang{#kode_barang, nama_barang, #id_merk, #id_kategori, #kode_divisi, #id_ruang}
3. detail_barang{#no,#kode_barang, no_pendaftaran, tahun_perolehan, satuan, bukti_pemilikan, penguasaan, sengketa, keterangan}
4. keadaan_barang{#no, #kode_barang, jml, jml_dikeluarkan, harga, total_harga, baik, rusak_ringan, rusak_berat, pemeriksa, eksistensi}
5. divisi{#kode_divisi, nama_divisi}
6. ruangan{#id_ruang, nama_ruang, #kode_divisi}
7. kategori{#id_kategori, nama_kategori}
8. merk {#id_merk, nama_merk}
9. transaksi_barang {#no, #kode_barang, tggl_diterima, tggl_pemeriksaan, no_pemeriksaan, spk, tggl_dikeluarkan, diserahkan_kpd, no_suratpenyerahan}



Gambar III. 16 Skema Relasi Inventarisasi Barang BPPTKPK

III.4.1.2 Struktur File

Adapun struktur file yang terdapat di dalam aplikasi ini adalah:

1. barang

Tabel barang berisi data barang untuk setiap ruangan yang berada pada BPPTKPK dan kepemilikannya oleh suatu divisi.

Nama tabel: barang

Primary key: kode_barang

Tabel III. 4 Struktur tabel barang

Field	Jenis	Ukuran	Keterangan
kode_barang	varchar	30	primary key
nama_barang	varchar	120	
id_merk	varchar	30	foreign key ref merk(id_merk)
id_kategori	varchar	30	foreign key ref kategori(id_kategori)
kode_divisi	varchar	30	foreign key ref divisi(kode_divisi)
id_ruang	varchar	30	foreign key ref ruangan(id_ruang)

2. detail_barang

Tabel detail_barang berisi detail barang untuk setiap barang.

Nama tabel: detail_barang

Primary key: no

Tabel III. 5 Struktur tabel detail_barang

Field	Jenis	Ukuran	Keterangan
no	int	11	primary key
kode_barang	varchar	30	foreign key ref barang(kode_barang)
no_pendaftaran	int	11	
tahun_perolehan	Int	11	
satuan	varchar	30	
bukti_pemilikan	varchar	50	
penguasaan	varchar	50	
sengketa	varchar	50	
keterangan	varchar	200	

3. divisi

Tabel divisi berisi data semua divisi.

Nama tabel: divisi

Primary key: kode_divisi

Tabel III. 6 Struktur tabel divisi

Field	Jenis	Ukuran	Keterangan
kode_divisi	varchar	30	primary key
nama_divisi	varchar	120	

4. kategori

Tabel kategori berisi data semua kategori barang.

Nama tabel: kategori

Primary key: id_kategori

Tabel III. 7 Struktur tabel kategori

Field	Jenis	Ukuran	Keterangan
id_kategori	varchar	30	primary key
nama_kategori	varchar	120	

5. keadaan_barang

Tabel keadaan_barang berisi keadaan barang pada suatu ruangan berdasarkan pemeriksaan yang dilakukan oleh bagian Pegawai Inventarisistrasi dan pemeriksa fisik.

Nama tabel: keadaan_barang

Primary key: no

Tabel III. 8 Struktur tabel keadaan barang

Field	Jenis	Ukuran	Keterangan
no	int	11	primary key
kode_barang	varchar	30	foreign key ref barang(kode_barang)
jml	int	11	
jml_dikeluarkan	Int	11	
harga	int	11	

Field	Jenis	Ukuran	Keterangan
total_harga	int	11	
baik	int	11	
rusak_ringan	int	11	
rusak_berat	int	11	
pemeriksa	varchar	30	default = 'administrasi'
eksistensi	varchar	10	default = 'ya'

6. merk

Tabel merk berisi data semua merk

Nama tabel: merk

Primary key: id_merk

Tabel III. 9 Struktur tabel merk

Field	Jenis	Ukuran	Keterangan
id_merk	varchar	30	primary key
nama_merk	varchar	120	

7. ruangan

Tabel ruangan berisi data semua ruangan dan informasi bagian dari suatu divisi.

Nama tabel: ruangan

Primary key: id_ruang

Tabel III. 10 Struktur tabel ruangan

Field	Jenis	Ukuran	Keterangan
id_ruang	varchar	30	primary key
nama_ruang	varchar	120	
kode_divisi	varchar	30	foreign key ref divisi(kode_divisi)

8. transaksi_barang

tabel transaksi_barang berisi data mengenai penerimaan, pemeriksaan, dan pengeluaran barang.

Nama tabel : transaksi_barang

Primary key : no

Tabel III. 11 Struktur tabel transaksi_barang

Field	Jenis	Ukuran	Keterangan
no	int	11	primary key
kode_barang	varchar	30	foreign key ref barang(kode_barang)
tggl_diterima	date		
tggl_pemeriksaan	date		
no_pemeriksaan	int	11	
spk	varchar	30	
tggl_dikeluarkan	date		
diserahkan_kpd	varchar	120	
no_suratpenyerahan	int	11	

Keterangan

Sistem memiliki fungsionalitas *login* sebagai verifikasi hak akses pengguna terhadap sistem, untuk itu dibutuhkan tabel *login* sebagai penyimpanan data *login*.

login

Tabel *login* ini berisi username dan password pegawai yang dipakai untuk autentikasi pengguna.

Nama tabel: *login*

Primary key: username

Tabel III. 12 Struktur tabel *login*

Field	Jenis	Ukuran	Keterangan
username	varchar	50	primary key
password	varchar	30	

III.5. Implementasi Sistem

Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman yang akan digunakan. Setelah implementasi maka dilakukan pengujian sistem yang baru dimana akan dilihat kekurangan-kekurangan pada aplikasi yang baru untuk selanjutnya diadakan pengembangan sistem. Tujuan implementasi adalah untuk mengkonfirmasi modul program perancangan pada para pelaku sistem sehingga pengguna dapat memberi masukan kepada pembangun sistem.

III.5.1. Implementasi Data

Bagian ini berisi mengenai data dan pengelompokkannya berdasarkan *class*-nya masing-masing. Dengan menerapkan MVC maka *class* dibagi menjadi beberapa kelompok yaitu model, view, dan controller sesuai dengan perannya terhadap penanganan data.

III.5.1.1. Model

Model merupakan kelompok *class* yang diciptakan untuk berkomunikasi dan memiliki hubungan langsung dengan basis data. Oleh karena itu secara struktur data *class* Model memiliki kesamaan dengan tabel pada basis data karena *class* Model sendiri merupakan model tahap analisa dari tabel basis data.

1. Pembuatan Basis Data

Query

```
CREATE DATABASE 'bpptkpk_db';
```

2. Class UserModel

UserModel merupakan *class* Model yang diciptakan untuk menangani data *login* dan berkomunikasi dengan tabel login.

Nama *class* : UserModel.cs

Nama tabel terhubung : login

Query

```
CREATE TABLE 'login' (
  'username' varchar(50) NOT NULL,
  'password' varchar(30) NOT NULL,
  PRIMARY KEY ('username')
```

)

3. **Class BarangModel**

BarangModel merupakan *class* Model yang diciptakan untuk menangani data penerimaan barang terhadap ruangan-ruangan penempatan barang dan berkomunikasi dengan tabel barang.

Nama *class* : BarangModel.cs

Nama tabel terhubung : barang

Query

```
CREATE TABLE 'barang' (
  'no' int(11) NOT NULL auto_increment,
  'kode_barang' varchar(30) NOT NULL,
  'id_merk' varchar(30) NOT NULL,
  'id_kategori' varchar(30) NOT NULL,
  'kode_divisi' varchar(30) NOT NULL,
  'id_ruang' varchar(30) NOT NULL,
  PRIMARY KEY ('no'),
  KEY 'kode_barang' ('kode_barang'),
  KEY 'id_merk' ('id_merk'),
  KEY 'id_kategori' ('id_kategori'),
  KEY 'kode_divisi' ('kode_divisi'),
  KEY 'id_ruang' ('id_ruang')
)
```

4. **Class DetailBarangModel**

DetailBarangModel merupakan *class* Model yang diciptakan untuk menangani data detail barang yang dimiliki oleh setiap barang inventaris dan berkomunikasi dengan tabel detail_barang.

Nama *class* : DetailBarangModel.cs

Nama tabel terhubung : detail_barang

Query

```
CREATE TABLE 'detail_barang' (
  'no' int(11) NOT NULL auto_increment,
  'kode_barang' varchar(30) NOT NULL,
  'no_pendaftaran' int(11) NOT NULL,
  'tahun_perolehan' int(11) NOT NULL,
  'satuan' varchar(30) NOT NULL,
```



```

' bukti_pemilikan' varchar(50) NOT NULL,
' penguasaan' varchar(50) NOT NULL,
' sengketa' varchar(50) NOT NULL,
' keterangan' varchar(200) NOT NULL,
PRIMARY KEY ('no'),
KEY 'kode_barang' ('kode_barang')
)

```

5. *Class KeadaanBarangModel*

KeadaanBarangModel merupakan *class Model* yang diciptakan untuk menangani data keadaan barang yang menggambarkan kondisi dari setiap barang inventaris dan berkomunikasi dengan tabel `keadaan_barang`.

Nama *class* : KeadaanBarangModel.cs

Nama tabel terhubung : `keadaan_barang`

Query

```

CREATE TABLE 'keadaan_barang' (
'no' int(11) NOT NULL auto_increment,
'kode_barang' varchar(30) NOT NULL,
'jml' int(11) NOT NULL,
'jml_dikeluarkan' int(11),
'harga' int(11) NOT NULL,
'total_harga' int(11) NOT NULL,
'baik' int(11) NOT NULL,
'rusak_ringan' int(11) NOT NULL,
'rusak_berat' int(11) NOT NULL,
'pemeriksa' varchar(30) NOT NULL default 'administrasi',
'eksistensi' varchar(10) NOT NULL default 'ya',
PRIMARY KEY ('no'),
KEY 'kode_barang' ('kode_barang')
)

```

6. *Class DivisiModel*

DivisiModel merupakan *class Model* yang diciptakan untuk menangani data divisi yang berada pada BPPTKPK dan berkomunikasi dengan tabel `divisi`.

Nama *class* : DivisiModel.cs

Nama tabel terhubung : `divisi`

Query

```
CREATE TABLE 'divisi' (
  'kode_divisi' varchar(30) NOT NULL,
  'nama_divisi' varchar(120) NOT NULL,
  PRIMARY KEY ('kode_divisi')
)
```

7. *Class RuanganModel*

RuanganModel merupakan *class* Model yang diciptakan untuk menangani data ruangan yang berada pada BPPTKPK dan merupakan bagian dari sebuah divisi dan berkomunikasi dengan tabel ruangan.

Nama *class* : RuanganModel.cs

Nama tabel terhubung : ruangan

Query

```
CREATE TABLE 'ruangan' (
  'id_ruang' varchar(30) NOT NULL,
  'nama_ruang' varchar(120) NOT NULL,
  'kode_divisi' varchar(30) NOT NULL,
  PRIMARY KEY ('id_ruang'),
  KEY 'kode_divisi' ('kode_divisi')
)
```

8. *Class MerkModel*

MerkModel merupakan *class* Model yang diciptakan untuk menangani data merk yang dimiliki oleh setiap barang inventaris dan berkomunikasi dengan tabel merk.

Nama *class* : MerkModel.cs

Nama tabel terhubung : merk

Query

```
CREATE TABLE 'merk' (
  'id_merk' varchar(30) NOT NULL,
  'nama_merk' varchar(120) NOT NULL,
  PRIMARY KEY ('id_merk')
)
```

9. *Class KategoriModel*

KategoriModel merupakan *class* Model yang diciptakan untuk menangani data kategori yang dimiliki oleh setiap barang inventaris dan berkomunikasi dengan tabel kategori.

Nama *class* : KategoriModel.cs

Nama tabel terhubung : kategori

Query

```
CREATE TABLE 'kategori' (
  'id_kategori' varchar(30) NOT NULL,
  'nama_kategori' varchar(120) NOT NULL,
  PRIMARY KEY ('id_kategori')
)
```

10. *Class TransaksiBarangModel*

TransaksiBarangModel merupakan *class* Model yang diciptakan untuk menangani data informasi mengenai transaksi yang dilakukan terhadap barang inventaris yang meliputi penerimaan barang, pemeriksaan barang, dan pengeluaran barang dan berkomunikasi dengan tabel transaksi_barang.

Nama *class* : TransaksiBarangModel.cs

Nama tabel terhubung : transaksi_barang

Query

```
CREATE TABLE 'transaksi_barang' (
  'no' int(11) NOT NULL auto_increment,
  'kode_barang' varchar(30) NOT NULL,
  'tggl_diterima' date NOT NULL,
  'tggl_pemeriksaan' date default NULL,
  'no_pemeriksaan' int(11) default NULL,
  'spk' varchar(30) default NULL,
  'tggl_dikeluarkan' date default NULL,
  'diserahkan_kpd' varchar(120) default NULL,
  'no_suratpenyerahan' int(11) default NULL,
  PRIMARY KEY ('no'),
  KEY 'kode_barang' ('kode_barang')
)
```

III.5.1.2. View

View merupakan kelompok *class* yang diciptakan untuk menangani data yang berhubungan dengan tampilan pengguna. *View* hanya merepresentasikan data dan menampilkannya terhadap pengguna, oleh karena itu *view* tidak memiliki unsur logika proses.

Tabel III. 14 Kelompok Class View

No	Nama Class	Ekstensi Class
1	LoginView	LoginView.cs
2	MenuView	MenuView.cs
3	BarangView	BarangView.cs
4	PengeluaranView	PengeluaranView.cs

III.5.1.3. Controller

Controller merupakan kelompok *class* yang diciptakan untuk menghubungkan antara *class* Model dan View. *Class* ini bertugas untuk meminta data dari *class* Model untuk kemudian direpresentasikan pada *class* View.

Tabel III. 14 Kelompok Class Controller

No	Nama Class	Ekstensi Class
1	LoginViewModel	LoginViewModel.cs
2	MenuViewModel	MenuViewModel.cs
3	BarangViewModel	BarangViewModel.cs
4	PengeluaranViewModel	PengeluaranViewModel.cs

III.5.2.Implementasi Antarmuka

Bagian ini berisi antarmuka yang ditampilkan kepada pegawai inventaris dan petugas biro aset ketika menggunakan sistem informasi inventarisasi barang di BPPTKPK.

III.5.2.1 Antarmuka *login* Awal

Pada antarmuka *login* awal pengguna sistem dapat memilih hak akses nya sebagai biro aset atau pegawai inventaris dengan menekan tombol yang sudah disediakan di dalam *form*.



Gambar III. 17 Implementasi Antarmuka *login* Awal

III.5.2.4 Antarmuka Menu Ruangan Dipilih

Pada antarmuka menu ruangan dipilih terdapat listbox yang berisi data-data ruangan pada divisi yang telah dipilih sebelumnya. Untuk masuk ke dalam *form* barang dari suatu ruangan dapat dilakukan dengan melakukan klik dua kali pada listbox item.



Gambar III. 20 Implementasi Antarmuka Menu Ruangan Dipilih

III.5.2.5 Antarmuka *Form* Barang Pegawai

Antarmuka *form* barang pegawai adalah *form* barang yang diakses oleh pegawai inventaris. Oleh karena itu di dalam *form* barang ini terdapat akses-akses pengolahan data barang seperti penerimaan data barang, pengubahan data barang, dan pengeluaran data barang.

 The screenshot shows a software window titled 'FormBarang'. It is divided into two main sections. The left section, titled 'Informasi Barang', contains several input fields for item details: 'Kode Barang' (set to AC01), 'Nama Barang' (set to Air Conditioner), 'Merk' (set to LG), 'Kategori' (set to Peralengkapan Umum), 'Merk Divisi' (set to Tataana Multimedia), and 'Berada di Ruang' (set to Ruang Kepala Balai). Below these is a 'Detail Barang Menurut' dropdown menu set to 'Administrasi'. Further down are fields for 'No Pendaftaran' (100), 'Tahun Perolehan' (1999), 'Satuan' (Unit), 'Balk' (1), 'Rusak Ringan' (1), 'Rusak Berat' (0), 'Jumlah' (2), 'Harga' (1200000), 'Total Harga' (2400000), 'Bukti Pemilikan' (Ada), 'Penggunaan' (Ada), and 'Sempurna' (Tidak Ada). A 'Keterangan' field at the bottom contains the text 'Barang Masih Baik'. The right section, titled 'Data Barang', shows a table with the header 'Data Barang Berada di Ruang Kepala Balai'. The table has columns: 'KodeBarang', 'NamaBarang', 'Merk', 'MerkKategori', and 'Kode'. It contains one row of data: 'AC01', 'Air Conditioner', 'LG', 'Peral', and 'MUL'. A 'Filter' button is located below the table.

Gambar III. 21 Implementasi Antarmuka *Form* Barang Pegawai

III.5.2.6 Antarmuka *Form* Tambah Barang

Antarmuka *form* tambah barang digunakan pegawai inventaris ketika menambahkan atau menempatkan barang pada suatu ruangan. Terdapat *form* inputan data barang yang harus diinputkan dengan lengkap sebelum menekan tombol “OK”.

Gambar III. 22 Implementasi Antarmuka *Form* Tambah Barang

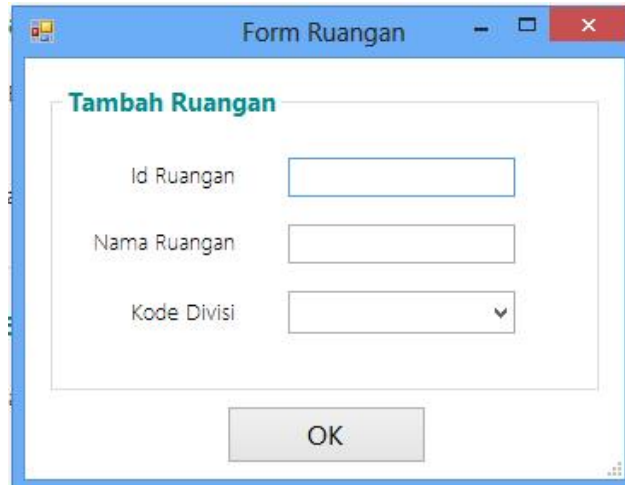
III.5.2.7 Antarmuka *Form* Barang Belum Diperiksa Secara Fisik

Antarmuka *form* barang belum diperiksa secara fisik akan ditampilkan kepada pegawai inventaris ketika mengubah combobox “Detail Barang Menurut” kepada nilai “Pemeriksa Fisik” dan barang belum diperiksa secara fisik. Tombol “Input” dan “Valid” digunakan untuk membuka *form* inputan data barang secara fisik.

Gambar III. 23 Implementasi Antarmuka *Form* Barang Belum Diperiksa Secara Fisik

III.5.2.8 Antarmuka *Form* Ruangan

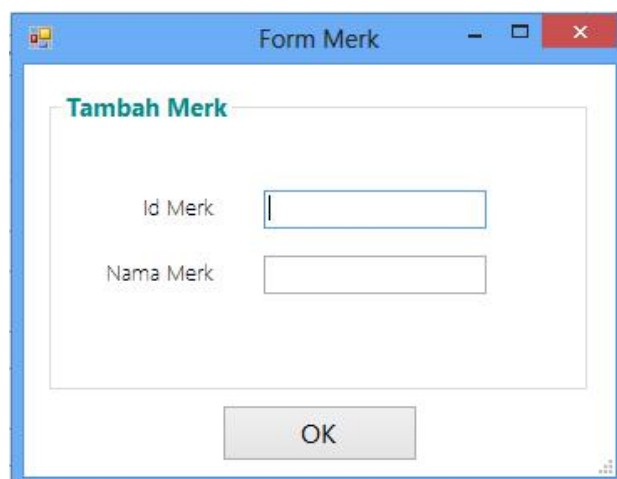
Antarmuka *form* ruangan ditampilkan kepada pegawai inventaris ketika akan menambahkan data ruangan. *Form* inputan data ruangan harus diinputkan dengan lengkap dengan id ruangan yang belum eksis.



Gambar III. 24 Implementasi Antarmuka *Form* Ruangan

III.5.2.9 Antarmuka *Form* Merk

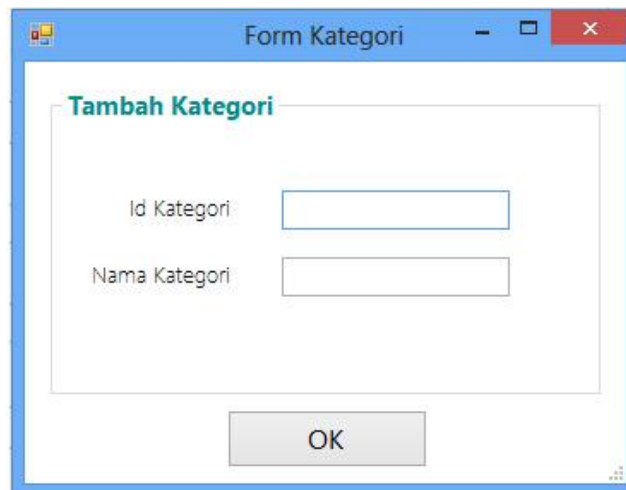
Antarmuka *form* merk ditampilkan kepada pegawai inventaris ketika akan menambahkan data merk. *Form* inputan data merk harus diinputkan dengan lengkap dengan id merk yang belum eksis.



Gambar III. 25 Implementasi Antarmuka *Form* Merk

III.5.2.10 Antarmuka *Form Kategori*

Antarmuka *form* kategori ditampilkan kepada pegawai inventaris ketika akan menambahkan data kategori. *Form* inputan data kategori harus diinputkan secara lengkap dengan id kategori yang belum eksis.



Gambar III. 26 Implementasi Antarmuka *Form Kategori*

III.5.2.11 Antarmuka *Form Divisi*

Antarmuka *form* divisi ditampilkan kepada pegawai inventaris ketika akan menambahkan data divisi. *Form* inputan divisi harus diinputkan secara lengkap dengan kode divisi yang belum eksis.



Gambar III. 27 Implementasi Antarmuka *Form Divisi*

III.5.2.12 Antarmuka *Form* Barang Umum

Antarmuka *form* barang umum adalah *form* barang yang ditampilkan kepada petugas biro yang datang untuk melakukan pemeriksaan barang. Pada *form* ini tidak disediakan akses untuk mengolah data barang.

The screenshot shows a web application window titled "FormBarang". It contains two main sections:

- Informasi Barang (Left):** A form with input fields for "Kode Barang" (AC01), "Nama Barang" (Air Conditioner), and "Merk" (LG). Below these are category dropdowns: "Kategori" (Perengkapan Umum), "Merk Dival" (Tunasi Multimedia), and "Berada di Ruang" (Ruang Kepala Balai). A "Detail Barang Menurut" dropdown is set to "Administrasi". Below this are several input fields for "No Pendaftaran" (90), "Tahun Revisi" (200), "Setan" (Lini), "Bak" (1), "Rusak Ringan" (1), "Rusak Berat" (0), "Jumlah" (2), "Harga" (500000), "Total Harga" (240000), "Budi Remihan" (Ada), "Penggunaan" (Ada), and "Sengaja" (Tidak Ada). A "Keterangan" field contains the text "Barang Masih Baik".
- Data Barang (Right):** A table titled "Data Barang Berada di Ruang Kepala Balai". The table has columns: "KodeBarang", "NamaBarang", "Merk", "Mekanisme", and "Kode". It contains one row of data: "AC01", "Air Conditioner", "LG", "Tunasi Multimedia", and "90". Below the table is a "Filter" button.

Gambar III. 28 Implementasi Antarmuka *Form* Barang Umum

III.5.2.13 Antarmuka *form* Export

Antarmuka *form* export digunakan ketika pegawai inventaris atau petugas biro akan mencetak laporan. Pada *form* ini terdapat pilihan format laporan yang dapat dipilih sesuai dengan kebutuhan laporan dan juga optional ekstensi file hasil export laporan.

The screenshot shows a web application window titled "Form Export". It contains two main sections:

- Optional Laporan (Left):** A section titled "Format Laporan :" with two radio buttons: "Buku Barang Inventaris" (checked) and "Laporan Inventaris" (unchecked).
- Optional Export (Right):** A section titled "Export ke file :" with two radio buttons: "Excel" (unchecked) and "Html" (checked).

At the bottom center of the window is a large "Export" button.

Gambar III. 29 Implementasi Antarmuka *Form* Export

III.5.2.14 Antarmuka *form* pengeluaran barang

Antarmuka *form* pengeluaran barang ditampilkan kepada pegawai inventaris untuk melakukan inputan informasi data barang berkaitan dengan pengeluaran barang.

Gambar III. 30 Implementasi Antarmuka *Form* Pengeluaran Barang

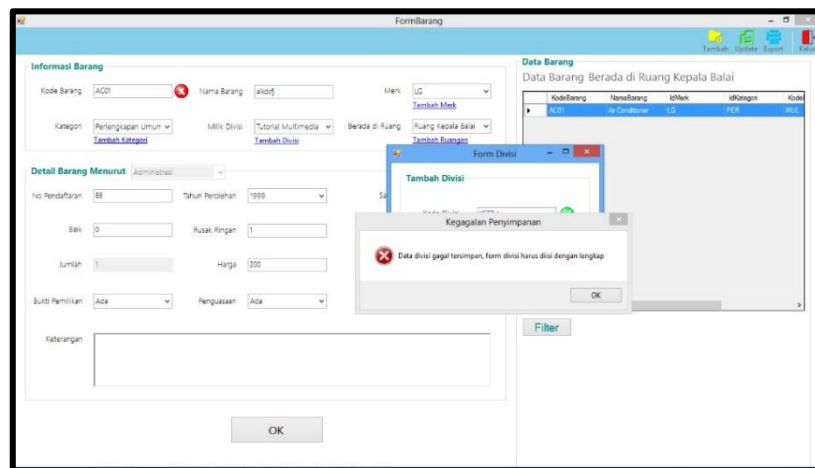
III.5.2.15 Antarmuka pesan konfirmasi kegagalan penyimpanan barang

Antarmuka pesan konfirmasi kegagalan penyimpanan barang ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data barang yang diakibatkan oleh kesalahan inputan data barang.

Gambar III. 31 Implementasi Antarmuka Pesan Konfirmasi Kegagalan Penyimpanan Barang

III.5.2.16 Antarmuka Pesan Konfirmasi Kegagalan Penyimpanan *form* divisi

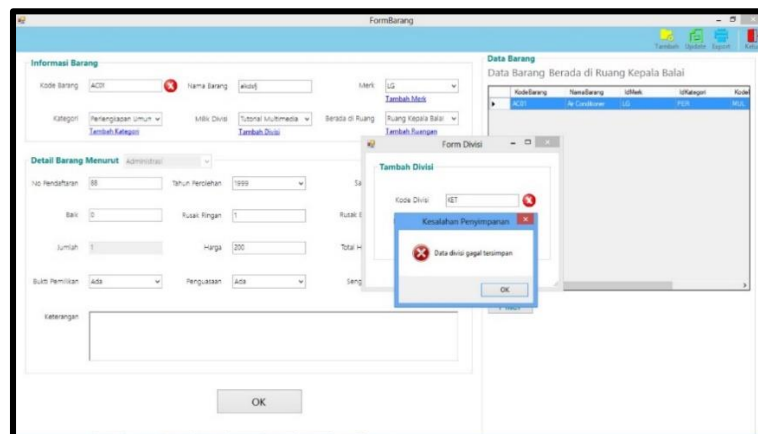
Antarmuka pesan konfirmasi kegagalan penyimpanan *form* divisi ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data divisi.



Gambar III. 32 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Divisi *Form* Tidak Lengkap

III.5.2.17 Antarmuka pesan konfirmasi kegagalan penyimpanan kategori *form* tidak lengkap.

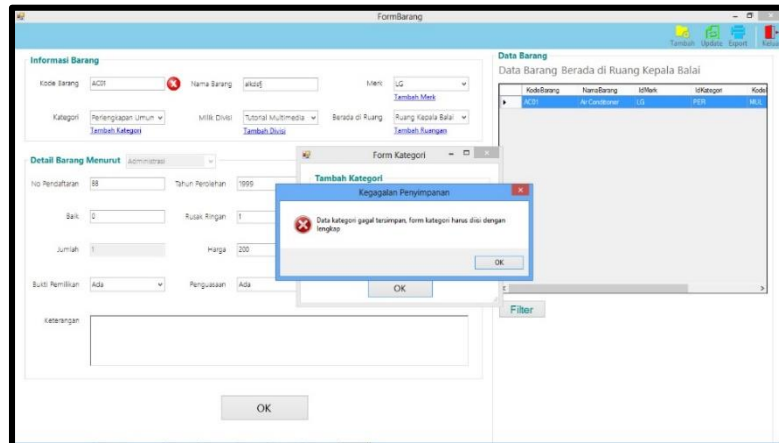
Antarmuka pesan konfirmasi kegagalan penyimpanan kategori *form* tidak lengkap ditampilkan oleh sistem kepada pegawai inventaris ketika adanya kegagalan penyimpanan data kategori diakibatkan oleh inputan data kategori yang tidak lengkap.



Gambar III. 33 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Kategori *Form* Tidak Lengkap

III.5.2.18 Antarmuka pesan konfirmasi kegagalan penyimpanan kode divisi sudah eksis

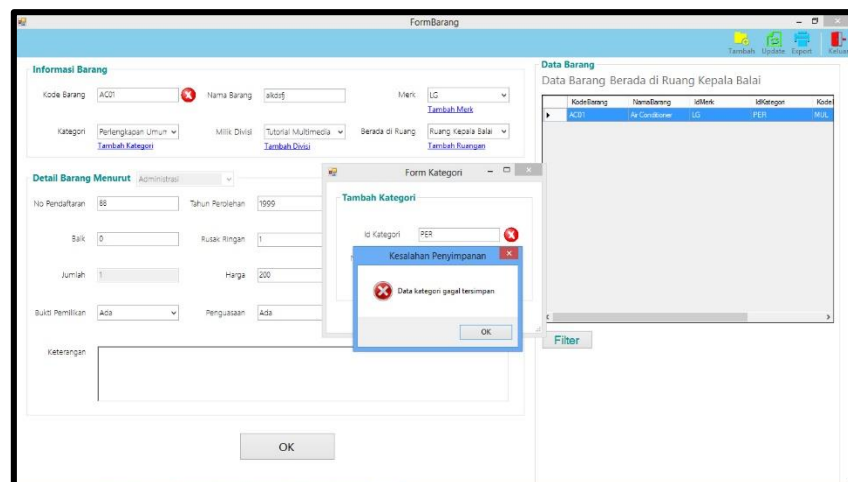
Antarmuka pesan konfirmasi kegagalan penyimpanan kode divisi sudah eksis ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data divisi yang diakibatkan redudansi kode divisi.



Gambar III. 34 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Divisi Kode Sudah Eksis

III.5.2.19 Antarmuka pesan konfirmasi kegagalan penyimpanan kategori id sudah eksis

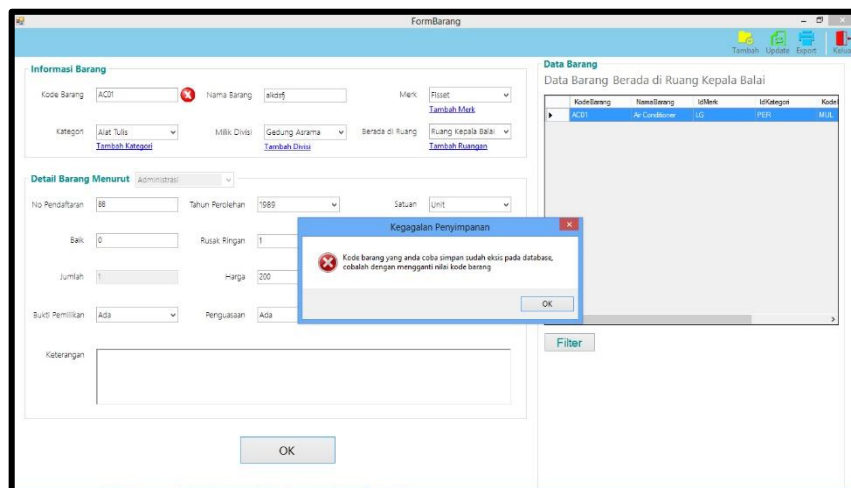
Antarmuka pesan konfirmasi kegagalan penyimpanan kategori id sudah eksis ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data kategori diakibatkan id kategori sudah eksis.



Gambar III. 35 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Kategori Id Sudah Eksis

III.5.2.20 Antarmuka pesan konfirmasi kegagalan penyimpanan kode barang sudah eksis.

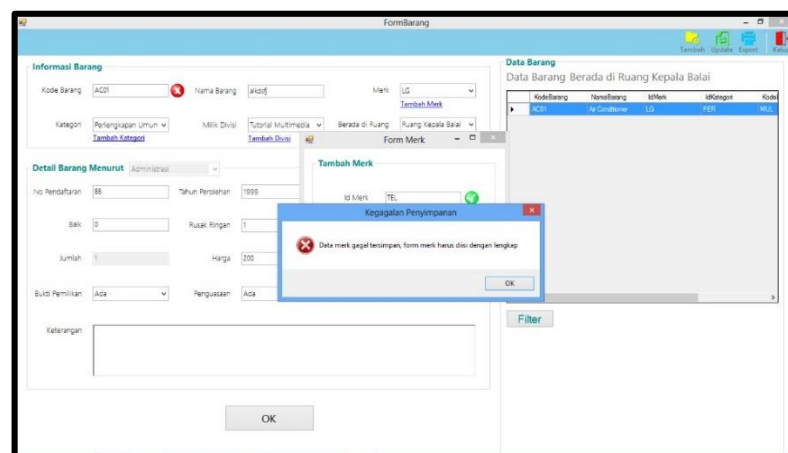
Antarmuka pesan konfirmasi kegagalan penyimpanan kode barang sudah eksis ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan kode barang diakibatkan kode barang sudah eksis.



Gambar III. 36 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Kode Barang Sudah Eksis

III.5.2.21 Antarmuka pesan konfirmasi kegagalan penyimpanan merk *form* tidak lengkap.

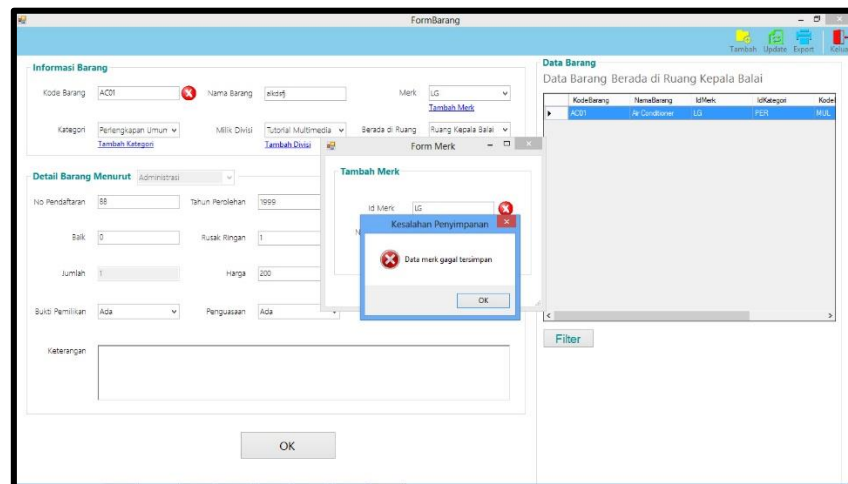
Antarmuka pesan konfirmasi kegagalan penyimpanan merk *form* tidak lengkap ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan merk diakibatkan *form* inputan tidak diisi dengan lengkap.



Gambar III. 37 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Merk *Form* Tidak Lengkap

III.5.2.22 Antarmuka pesan konfirmasi kegagalan penyimpanan merk Id sudah eksis

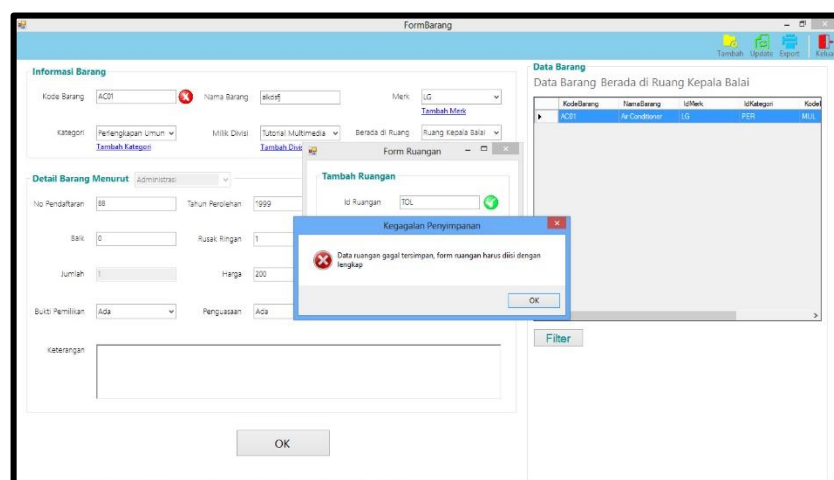
Antarmuka pesan konfirmasi kegagalan penyimpanan merk id sudah eksis ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data merk yang diakibatkan id merk sudah eksis.



Gambar III. 38 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Merk Id Sudah Eksis

III.5.2.23 Antarmuka pesan konfirmasi kegagalan penyimpanan *form* ruangan yang tidak lengkap

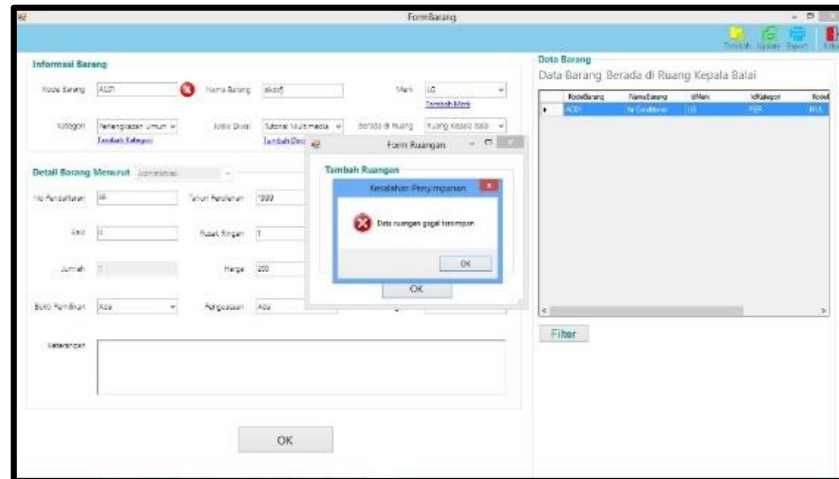
Antarmuka pesan konfirmasi kegagalan penyimpanan *form* ruangan yang tidak lengkap ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data ruangan yang diakibatkan inputan pada *form* ruangan tidak lengkap.



Gambar III. 39 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Ruangan *Form* Tidak Lengkap

III.5.2.24 Antarmuka pesan konfirmasi kegagalan penyimpanan ruangan Id sudah eksis

Antarmuka pesan konfirmasi kegagalan penyimpanan ruangan id sudah eksis ditampilkan kepada pegawai inventaris ketika adanya kegagalan pada penyimpanan data ruangan yang diakibatkan id ruangan sudah eksis.



Gambar III. 40 Implementasi Antarmuka Pesan Kegagalan Penyimpanan Ruang Id Sudah Eksis

III.5.2.25 Antarmuka pesan konfirmasi kesalahan login

Antarmuka pesan konfirmasi kesalahan *login* ditampilkan kepada pegawai inventaris ketika melakukan kesalahan inputan pada *form login*.



Gambar III. 41 Implementasi Antarmuka Pesan Kesalahan Login

BAB IV

KESIMPULAN DAN SARAN

IV.1. KESIMPULAN

Berdasarkan penelitian dan pengujian yang kami lakukan, dengan adanya “Sistem Informasi Inventarisasi Barang Di Balai Pelatihan Pendidik Dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK)”, dapat disimpulkan bahwa :

- 1 Sistem Informasi Inventarisasi Barang dapat Mengurangi kekeliruan pendataan barang agar tidak terjadi redudansi dan ketidakvalidan data inventaris barang karena sudah ada sudah ada sistem pengkodean dan pengecekan barang.
- 2 Sistem Informasi Inventarisasi Barang di Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan dan Kejuruan dapat Mempercepat proses *updating* data barang pada dokumen utama inventaris barang.
- 3 Sistem Informasi Inventarisasi Barang di Balai Pelatihan Pendidik dan Tenaga Kependidikan Pendidikan dan Kejuruan dapat Memudahkan petugas biro aset yang datang untuk memeriksa dalam melakukan pencarian data barang.

IV.2. SARAN

Berdasarkan kesimpulan yang didapat maka saran-saran yang dapat dikemukakan agar menjadi bahan masukan dan pertimbangan terhadap “Sistem Informasi Inventarisasi Barang Di Balai Pelatihan Pendidik Dan Tenaga Kependidikan Pendidikan Kejuruan (BPPTKPK)” adalah sebagai berikut :

1. Perlu adanya sinkronisasi antara aplikasi yang dibangun dengan aplikasi yang dimiliki oleh pegawai biro sehingga dapat memudahkan pegawai inventaris dalam mengajukan surat pengaduan barang.
2. Dapat dikembangkan Website agar para Biro Aset, Pimpinan Kantor BPPTKPK dapat memeriksa laporan lebih fleksible.

DAFTAR PUSTAKA

- [1] Roger S. Pressman, 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*, ANDI Yogyakarta.
- [2] Jogianto, H.M. 2000. *Pengenalan Komputer*, ANDI OFFSET. Yogyakarta.
- [3] Davis Bgordon. 1992, *Kerangka Dasar Sistem Informasi Manajemen Bagian 1*. PT Pustaka Binawan Presindo. Jakarta.
- [4] LABORATORIUM FISIKA KIP UNLAM, "Inventaris Alat," 30-12-2011. [Online]. Available: <http://fisikahappy.wordpress.com/2011/12/30/inventarisasi-alat/>. [Accessed 12 01 2014].
- [5] Ardira, "SISTEM ORIENTASI OBJEK," 04-01-2012. [Online]. Available: <http://febryrisdianto.blogspot.com/2011/04/makalah-pbo-pmograman-berorientasi.html>. [Accessed 12-01-2014].
- [6] Bowo, "Langkah-Langkah Penggunaan UML," 01 05 2010. [Online]. Available: <http://bowol.blogspot.com/2010/03/langkah-langkah-penggunaan-uml-unified.html>. [Accessed 12 01 2014].
- [7] Grady Booch. 1999. *Visual Modeling With Rational Rose 2000 And UML* .