

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

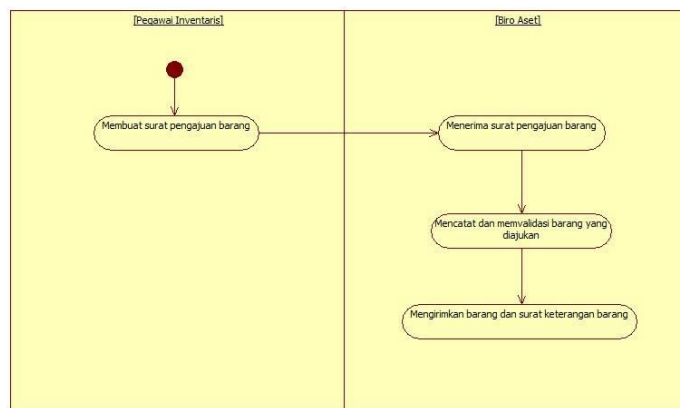
Berdasarkan pengamatan sistem yang sedang berjalan, diperlukan suatu perbaikan proses pendataan inventarisasi barang di BPPTKPK yang meliputi penginputan dan perubahan data barang yang masih berjalan secara manual, pembuatan laporan yang belum dilakukan

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

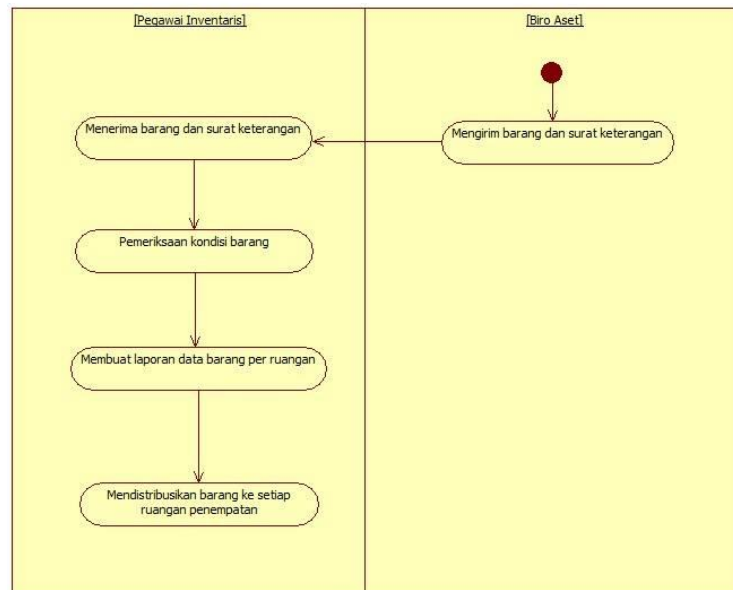
- a. Pegawai inventaris membuat surat pengajuan barang kepada Biro Aset.
- b. Biro Aset menerima surat pengajuan barang dari lembaga BPPTKPK yang diwakili oleh pegawai inventaris.
- c. Biro Aset mencatat dan memvalidasi seluruh barang yang telah diajukan.
- d. Biro Aset mengirimkan barang yang telah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



Gambar III. 1 Activity Diagram Pengadaan Barang

2. Prosedur Penerimaan Barang dari Biro Aset

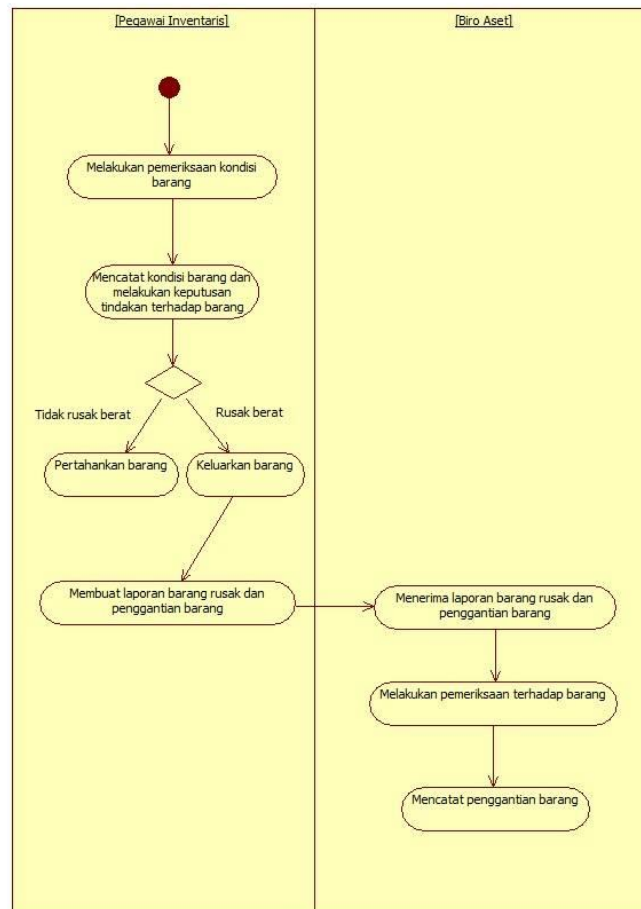
- a. Biro Aset mengadakan inventaris barang baru setiap 2 tahun sekali.
- b. Pegawai inventaris menerima barang dan surat keterangan barang dari biro aset.
- c. Pegawai inventaris mengecek barang yang telah diterima terlebih dahulu.
- d. Pegawai inventaris membuat laporan data barang yang akan didistribusikan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

- a. 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.
- b. Pegawai inventaris mencatat laporan keadaan barang (baik, rusak ringan, rusak berat) dari setiap ruangan yang diperiksa maupun dari laporan keadaan barang dari pegawai BPPTKPK.
- c. Apabila barang tersebut dalam keadaan rusak berat maka akan dikeluarkan dari ruangan, apabila tidak dalam keadaan rusak berat maka barang tersebut akan tetap dipertahankan.
- d. Pegawai inventaris membuat laporan pengeluaran dan penggantian barang kepada Biro Aset.
- e. Biro Aset menerima laporan pengeluaran barang dari pegawai inventaris barang.
- f. Biro Aset melakukan pengecekan data barang yang diajukan.
- g. 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.

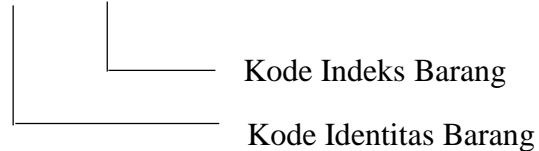
III.1.3.4 Analisis Kebutuhan Pengkodean

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



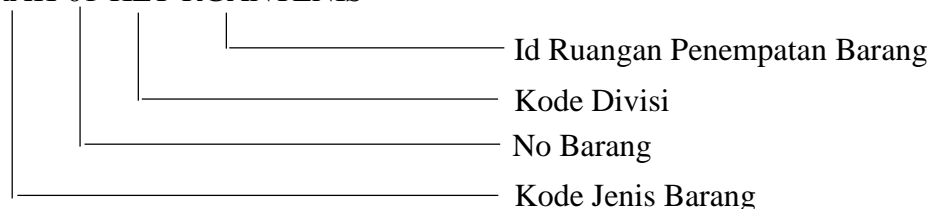
Pegawai inventaris hanya mencatat data barang berdasarkan nama barang dan merk barang saja sehingga dapat disimpulkan bahwa pada pendataan barang inventarisasi di kantor BPPTKPK belum memiliki pengkodean.

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

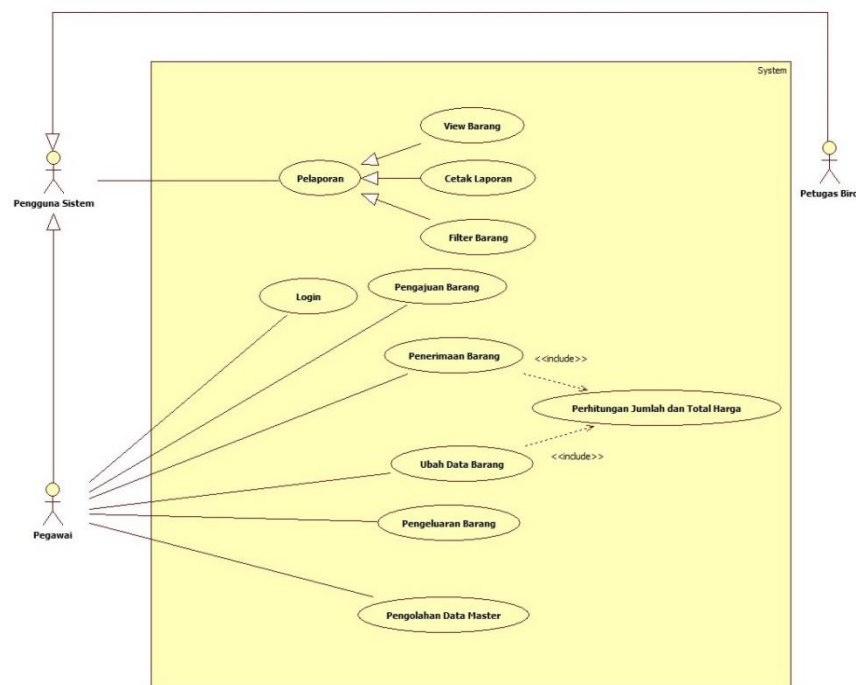
13.	SIB-BPPTKPK-F-1-13	Pegawai Inventaris dapat mengelola pengeluaran barang inventaris
14.	SIB-BPPTKPK-F-1-14	Pegawai Inventaris dapat mengelola pemeriksaan barang inventaris
15.	SIB-BPPTKPK-F-1-15	Pegawai Inventaris dapat mencetak surat pengajuan barang untuk diserahkan kepada biro aset
16.	SIB-BPPTKPK-F-2-01	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 :

III.2.3. Definisi Use Case

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.

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.	<i>View</i> 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 dalam database.
6.	Pengeluaran Barang	Sistem akan melakukan pelabelan ketidak eksistensian data barang. Jika data pengeluaran barang diinputkan dengan lengkap oleh pegawai

No	Aktor	Deskripsi
		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 2 3 4	Pegawai berada di <i>Form</i> Barang Pegawai menekan tombol tambah Pegawai menginputkan data barang 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.

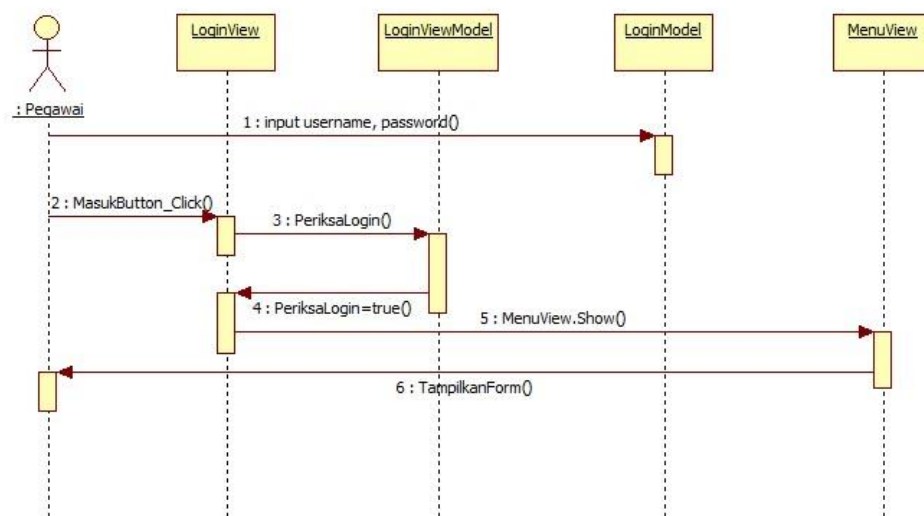
III.3.1 Realisasi Use Case Tahap Analisis

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



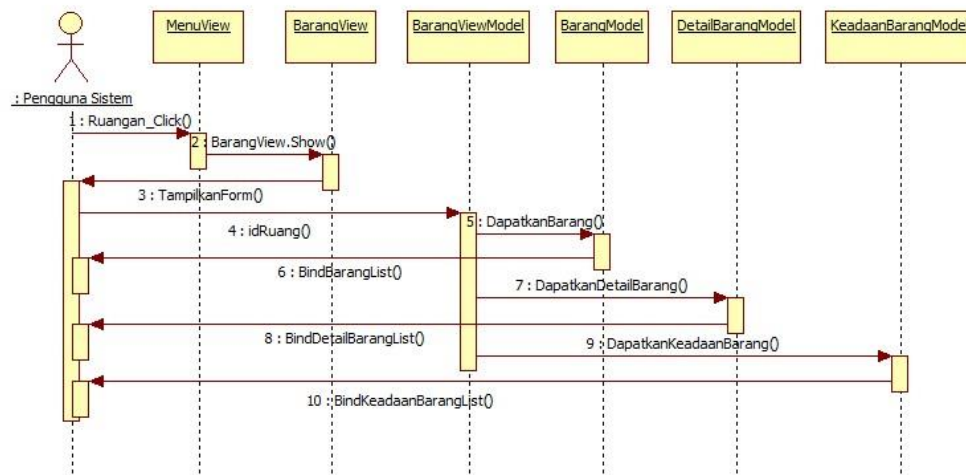
Gambar III. 5 *Sequence Diagram Login*

- Pegawai memasukkan username dan password pada *Form Login*
- Pegawai menekan tombol “Masuk”
- Kemudian *Form login* melakukan pemeriksaan data login
- 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



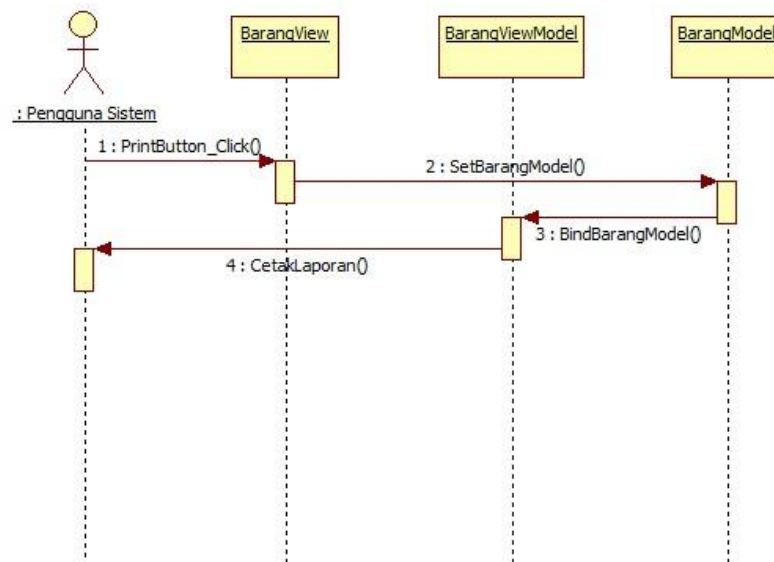
Gambar III. 6 Sequence Diagram View Barang

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



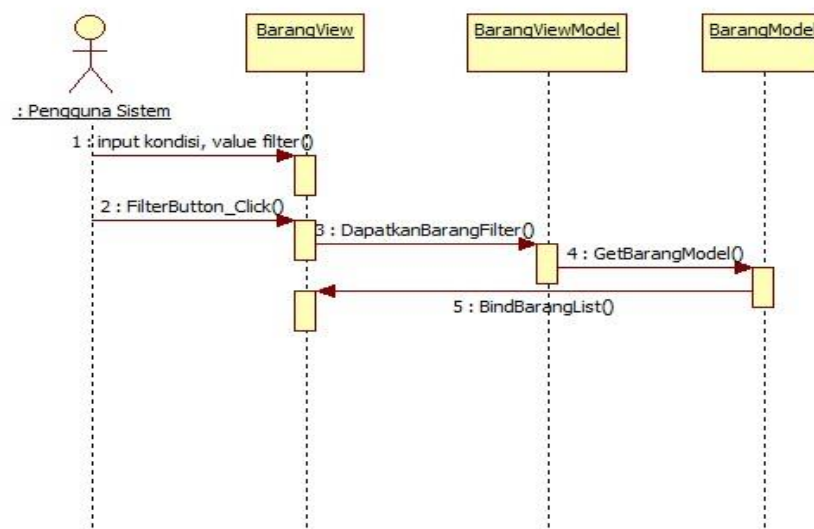
Gambar III. 7 *Sequence Diagram Cetak Laporan*

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



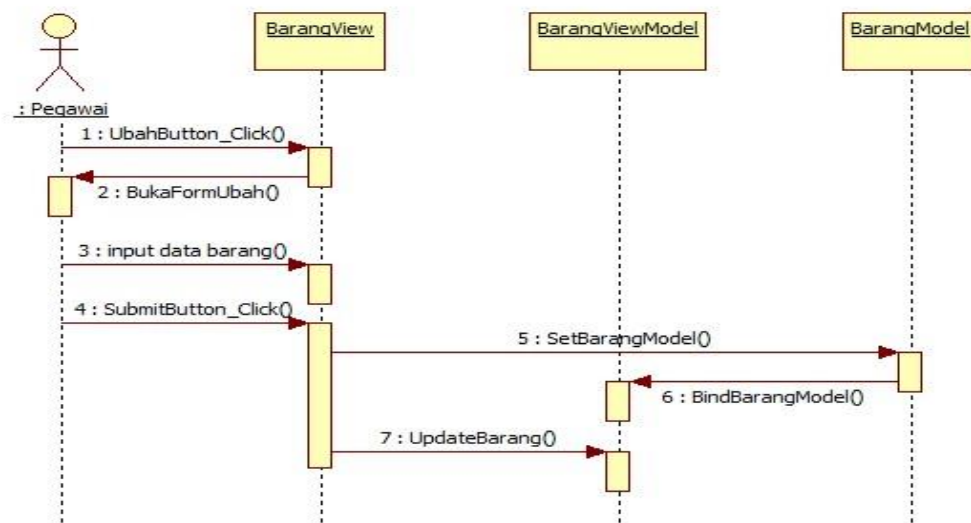
Gambar III. 8 Sequence Diagram Filter Barang

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



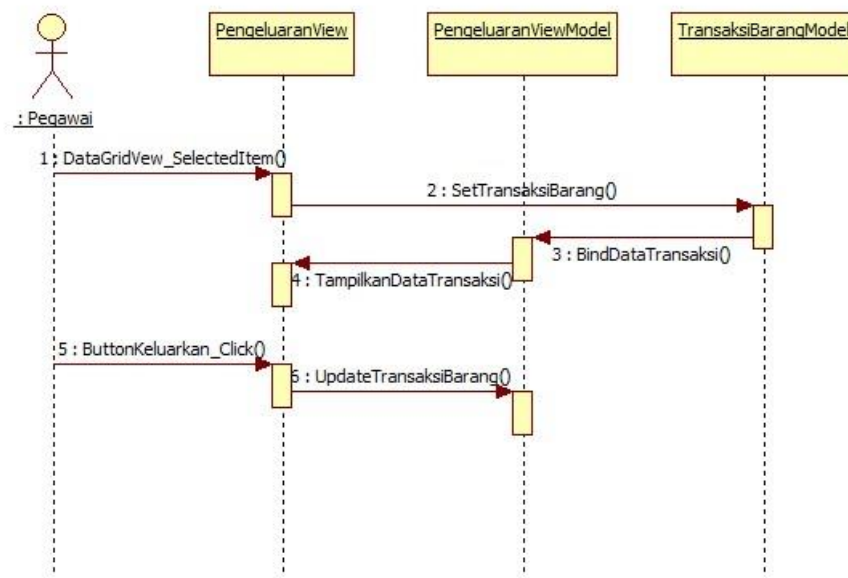
Gambar III. 9 Sequence Diagram Ubah Data Barang

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



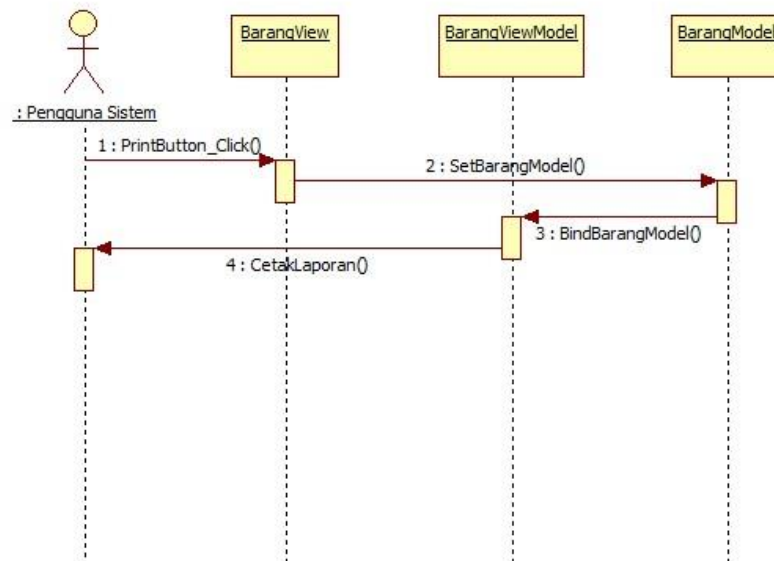
Gambar III.10 Sequence Diagram Pengeluaran Barang

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



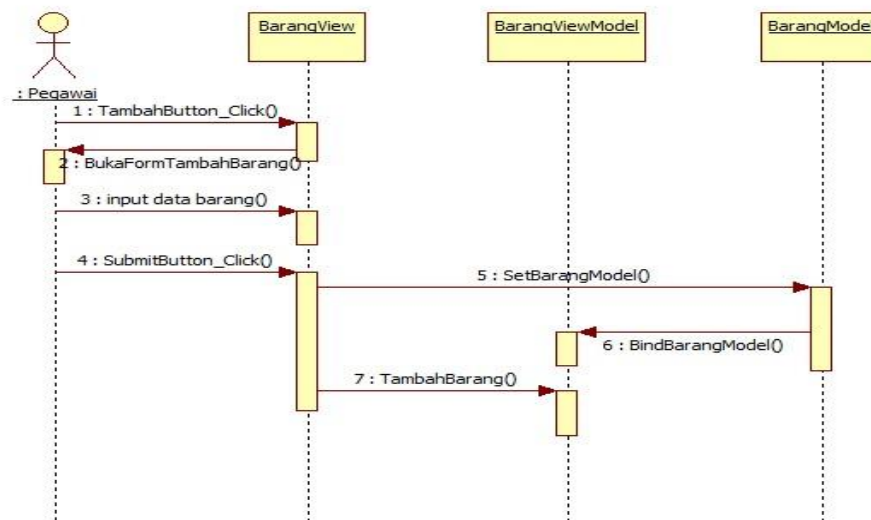
Gambar III.11 Sequence Diagram Cetak Pengajuan Barang

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



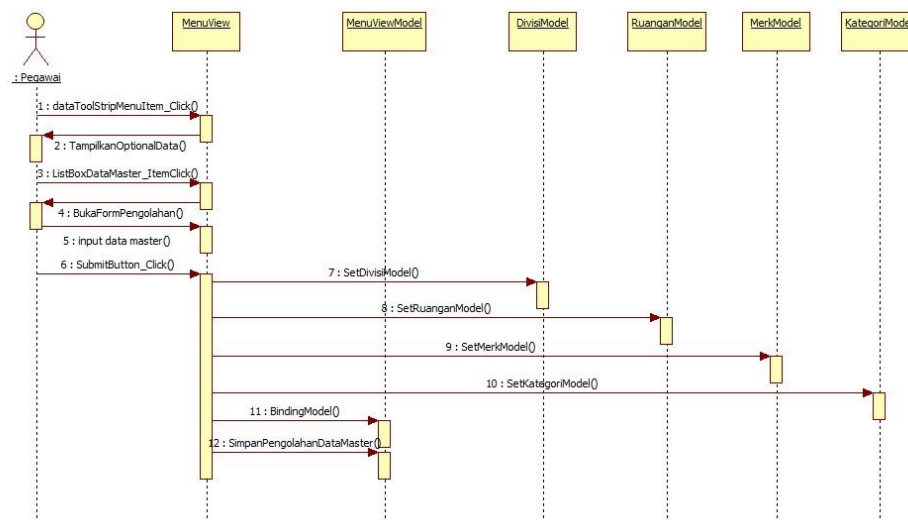
Gambar III. 12 Sequence Diagram Penerimaan Barang

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 masukan 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 Class</i>	<div data-bbox="592 324 1150 1142"> <pre> MenuViewModel +sqlConnection +divList +katList +ruangList +merkList +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) </pre> </div>	<ul style="list-style-type: none"> - View Barang - Pengolahan Data Master
<i>BarangView Class</i>	<div data-bbox="719 1182 1026 1668"> <pre> BarangView +kodeBarang +dataFilter +bagian +hakAkses +ButtonUpdate_Click() +ButtonTambah_Click() +ButtonOK_Click() +ButtonExport_Click() +ButtonClose_Click() +ButtonFilter_Click() +Grid_RowSelected() +ButtonValid_Click() </pre> </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 <i>Class</i>	<div data-bbox="616 327 1131 815"> <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 <i>Class</i>	<div data-bbox="751 848 992 1337"> <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

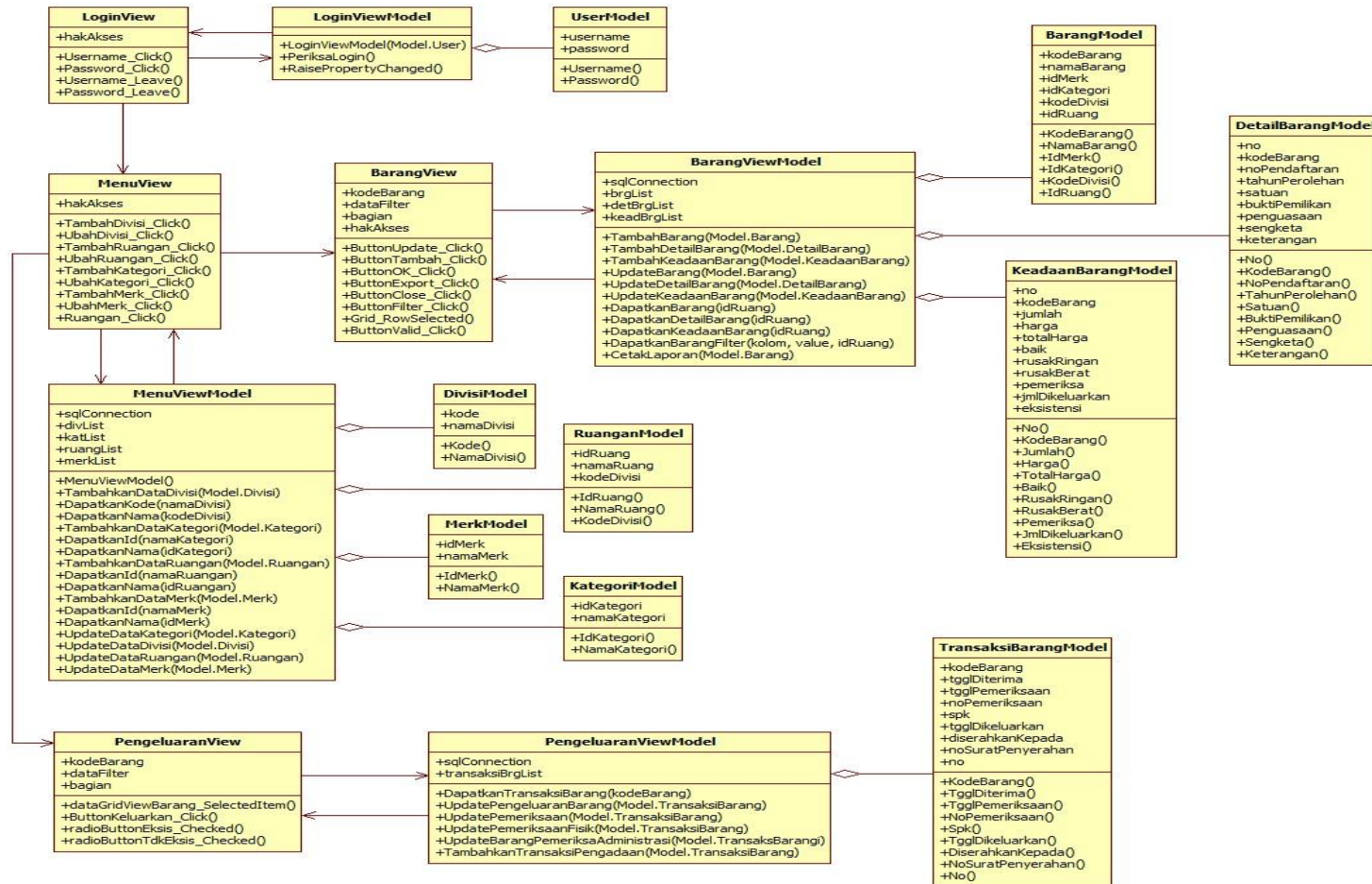
Nama Class	Class	Terlibat pada Use Case
DetailBarangModel Class	<div data-bbox="724 324 1019 1021"> DetailBarangModel +no +kodeBarang +noPendaftaran +tahunPerolehan +satuan +buktiPemilikan +penguasaan +sengketa +keterangan +No() +KodeBarang() +NoPendaftaran() +TahunPerolehan() +Satuan() +BuktiPemilikan() +Penguasaan() +Sengketa() +Keterangan() </div>	- View Barang
KeadaanBarangModel Class	<div data-bbox="724 1057 1019 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 <i>Class</i>	<div data-bbox="550 324 1195 622"> PengeluaranViewModel +sqlConnection +transaksiBrgList +DapatkanTransaksiBarang(kodeBarang) +UpdatePengeluaranBarang(Model.TransaksiBarang) +UpdatePemeriksaan(Model.TransaksiBarang) +UpdatePemeriksaanFisik(Model.TransaksiBarang) +UpdateBarangPemeriksaAdministrasi(Model.TransaksBarangi) +TambahkanTransaksiPengadaan(Model.TransaksiBarang) </div>	- Pengeluaran Barang
TransaksiBarangModel <i>Class</i>	<div data-bbox="694 654 1048 1344"> 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

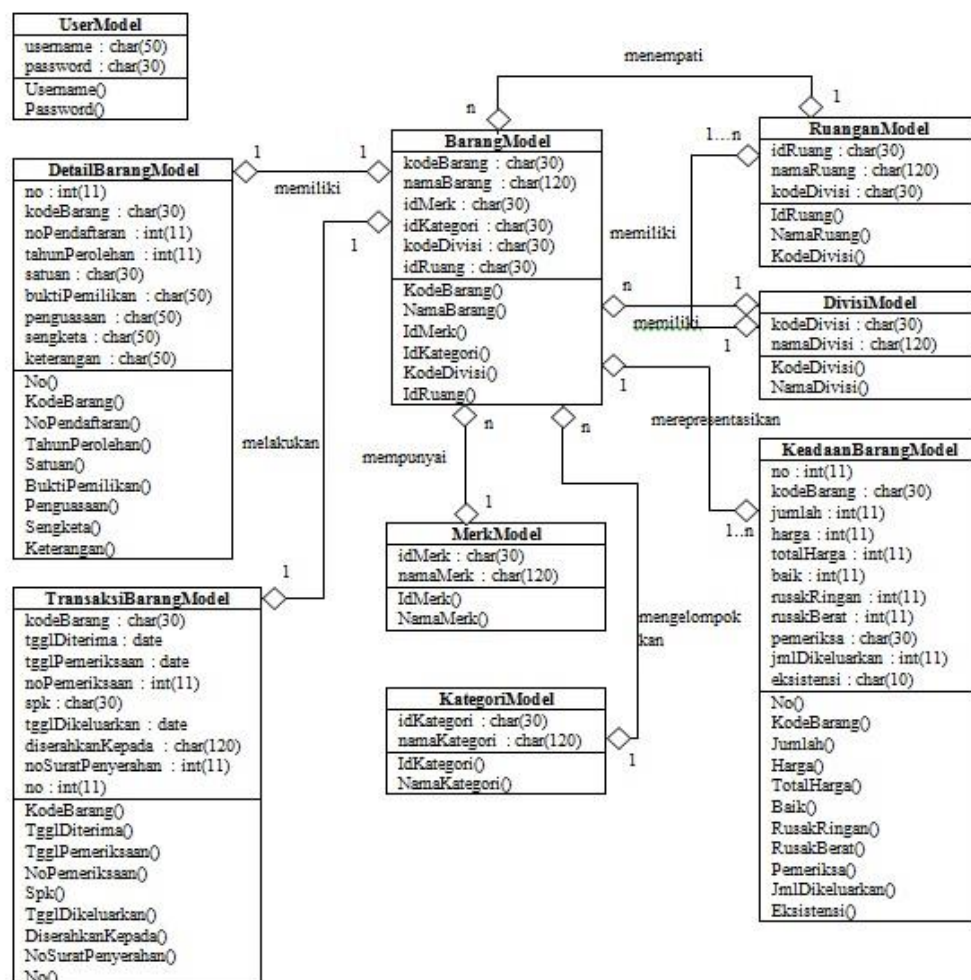
III.3.1.12. Kelompok Kelas dalam Pattern MVC

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

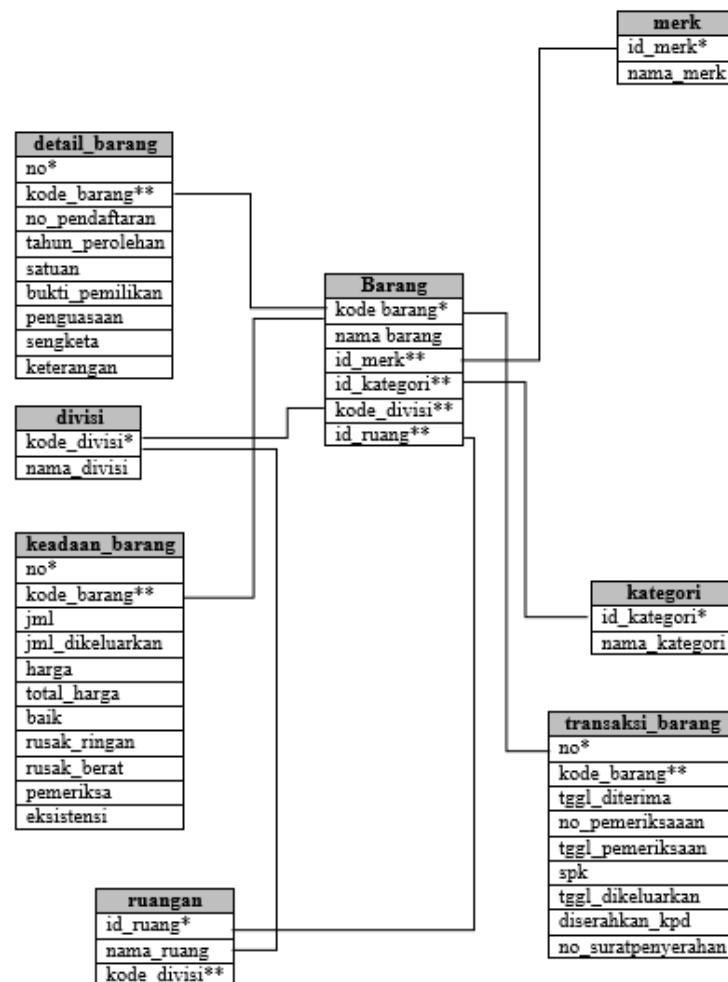
Perancangan data atau lebih dikenal dengan perancangan basis data yaitu menciptakan, atau merancang data yang terhubung dan disimpan secara bersama-sama. Untuk menggambarannya digunakanlah skema relasi dan struktur *file*. Dan dua hasil tersebut merupakan basis data yang diimplementasikan pada aplikasi 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	
total_harga	int	11	
baik	int	11	

Field	Jenis	Ukuran	Keterangan
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`),
  CONSTRAINT `barang_ibfk_6` FOREIGN KEY (`id_merk`) REFERENCES
`merk` (`id_merk`) ON UPDATE CASCADE,
  CONSTRAINT `barang_ibfk_7` FOREIGN KEY (`id_kategori`) REFERENCES
`kategori` (`id_kategori`) ON UPDATE CASCADE,
  CONSTRAINT `barang_ibfk_8` FOREIGN KEY (`kode_divisi`) REFERENCES
`divisi` (`kode_divisi`) ON UPDATE CASCADE,
  CONSTRAINT `barang_ibfk_9` FOREIGN KEY (`id_ruang`) REFERENCES
`ruangan` (`id_ruang`) ON UPDATE CASCADE
)
```

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`),
CONSTRAINT `detail_barang_ibfk_1` FOREIGN KEY (`kode_barang`)
REFERENCES `barang` (`kode_barang`) ON UPDATE CASCADE
)

```

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) DEFAULT '0',
  `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,
  `eksistensi` varchar(10) NOT NULL DEFAULT 'ya',
  PRIMARY KEY (`no`),
  KEY `kode_barang` (`kode_barang`),
  CONSTRAINT `keadaan_barang_ibfk_1` FOREIGN KEY (`kode_barang`)
  REFERENCES `barang` (`kode_barang`) ON UPDATE CASCADE
)

```


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`),
  CONSTRAINT `ruangan_ibfk_1` FOREIGN KEY (`kode_divisi`) REFERENCES
  `divisi` (`kode_divisi`) ON UPDATE CASCADE
)
```

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,
  `tgl_diterima` date NOT NULL,
  `tgl_pemeriksaan` date DEFAULT NULL,
  `no_pemeriksaan` int(11) DEFAULT NULL,
  `spk` varchar(30) DEFAULT NULL,
  `tgl_dikeluarkan` date DEFAULT NULL,
  `diserahkan_kpd` varchar(120) DEFAULT NULL,
```

```

`no_suratpenyerahan` int(11) DEFAULT NULL,
PRIMARY KEY (`no`),
KEY `kode_barang` (`kode_barang`),
CONSTRAINT `transaksi_barang_ibfk_1` FOREIGN KEY (`kode_barang`)
REFERENCES `barang` (`kode_barang`) ON UPDATE CASCADE
)

```

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. 13 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.5.2.2 Antarmuka *login* Pegawai

Pada antarmuka *login* pegawai terdapat *form* inputan *login* yang harus diisi dengan lengkap oleh pegawai inventaris. Kemudian pegawai inventaris dapat menekan tombol “Masuk” untuk melakukan submit terhadap inputan *login* dan masuk ke dalam sistem.



Gambar III. 18 Implementasi Antarmuka *login* Pegawai

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 the 'FormBarang' application window. The 'Informasi Barang' section contains input fields for 'Kode Barang' (AC31), 'Nama Barang' (Air Conditioner), 'Merk' (LG), 'Kategori' (Peralengkapan Umum), 'Merk Divisi' (Toshiba Multimedia), and 'Berada di Ruang' (Ruang Kepala Balai). Below this is a 'Detail Barang Menurut' dropdown menu set to 'Administrasi'. Further down are fields for 'No Pendaftaran' (100), 'Tahun Perolehan' (1999), 'Satuan' (Unit), 'Baki' (1), 'Rusak Ringan' (1), 'Rusak Berat' (0), 'Jumlah' (2), 'Harga' (1200000), 'Total Harga' (2400000), 'Bukti Pemilikan' (Ada), 'Penggunaan' (Ada), and 'Sengista' (Tidak Ada). A 'Keterangan' field contains 'Barang Masih Baik'. The 'Data Barang' section on the right shows a table with columns 'KodeBarang', 'NamaBarang', 'Merk', 'MkKategori', and 'Kode'. It contains one row with values: AC31, Air Conditioner, LG, 100, and 100. 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”.

The screenshot shows the 'FormBarang' application window with empty input fields. The 'Informasi Barang' section has fields for 'Kode Barang', 'Nama Barang', 'Merk' (with a 'Reset' button and a 'Tambah Merk' link), 'Kategori' (with a 'Tambah Kategori' link), 'Merk Divisi' (with a 'Tambah Divisi' link), and 'Berada di Ruang' (with a 'Tambah Ruang' link). Below this is a 'Detail Barang Menurut' dropdown menu set to 'Administrasi'. Further down are fields for 'No Pendaftaran', 'Tahun Perolehan', 'Satuan', 'Baki', 'Rusak Ringan', 'Rusak Berat', 'Jumlah', 'Harga', 'Total Harga', 'Bukti Pemilikan', 'Penggunaan', and 'Sengista'. A 'Keterangan' field is empty. An 'OK' button is at the bottom center. The 'Data Barang' section on the right shows a table with columns 'KodeBarang', 'NamaBarang', 'Merk', 'MkKategori', and 'Kode'. It contains one row with values: AC31, Air Conditioner, LG, 100, and 100. A 'Filter' button is located below the table.

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

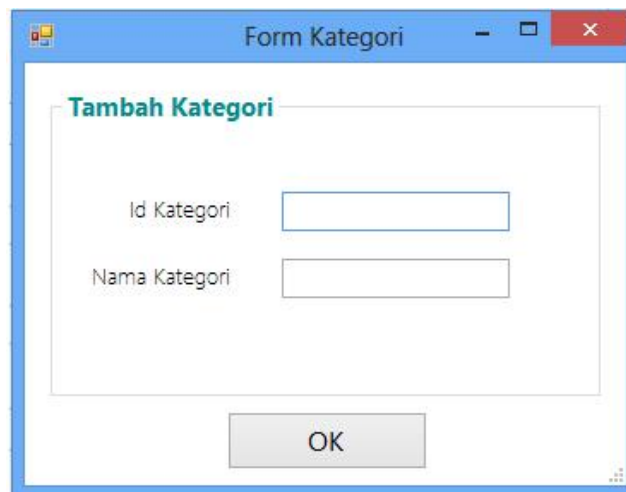
Antarmuka *form* merk ditampilkan kepada pegawai inventaris ketika akan menambahkan data merk. *Form* inputan data merk harus diinputkan dengan lengkap dengan id merk yang belum eksis.

The image shows a Windows-style window titled "Form Merk". Inside the window, there is a section titled "Tambah Merk" in green text. Below this title, there are two input fields: "Id Merk" and "Nama Merk". Each label is positioned to the left of its corresponding text box. At the bottom center of the window, there is a grey button labeled "OK". The window has a standard Windows title bar with minimize, maximize, and close buttons.

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.

The image shows a Windows-style window titled "Form Kategori". Inside the window, there is a section titled "Tambah Kategori" in green text. Below this title, there are two input fields: "Id Kategori" and "Nama Kategori". Each label is positioned to the left of its corresponding text box. At the bottom center of the window, there is a grey button labeled "OK". The window has a standard Windows title bar with minimize, maximize, and close buttons.

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.

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.

Gambar III. 29 Implementasi Antarmuka *Form Export*

III.5.2.14 Antarmuka *form* pengeluaran barang

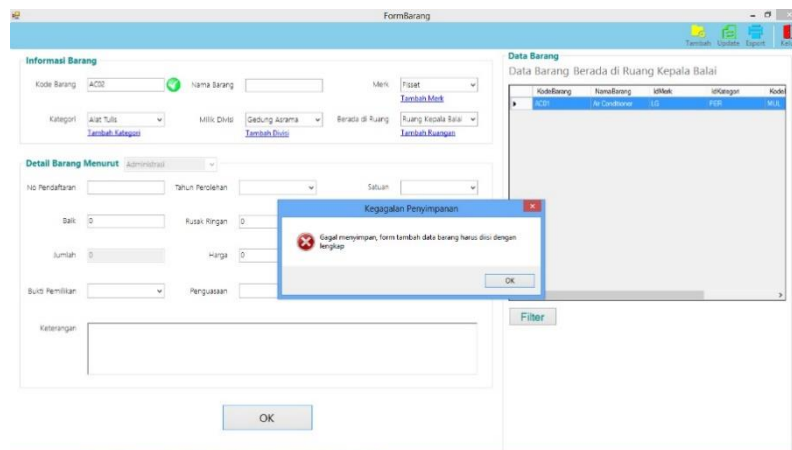
Antarmuka *form* pengeluaran barang ditampilkan kepada pegawai inventaris untuk melakukan inputan informasi data barang berkaitan dengan pengeluaran barang.

KodeBarang	NamaBarang	IdMerk	IdKategori	Kode
SUP01	Kursi Sofa	OLYM	FUR	Q70
LEM01	Lemari Buku	OLYM	FUR	KET
MEJA01	Mega	OLYM	FUR	KET
TR01	Troli	APE	PER	MES1

Gambar III. 30 Implementasi Antarmuka *Form Pengeluaran Barang*

III.5.2.15 Antarmuka pesan konfirmasi kegagalan penyimpanan barang

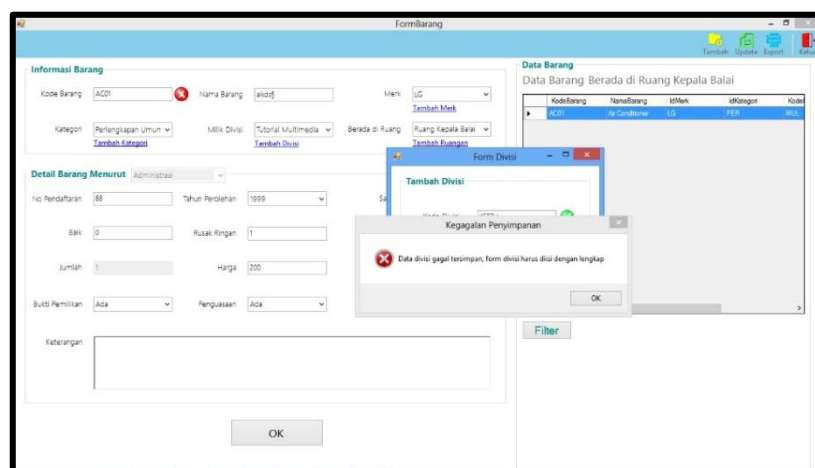
Antarmuka pesan konfirmasi kegagalan penyimpanan barang ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data barang yang diakibatkan oleh kesalahan inputan data barang.



Gambar III. 31 Implementasi Antarmuka Pesan Konfirmasi Kegagalan Penyimpanan Barang

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

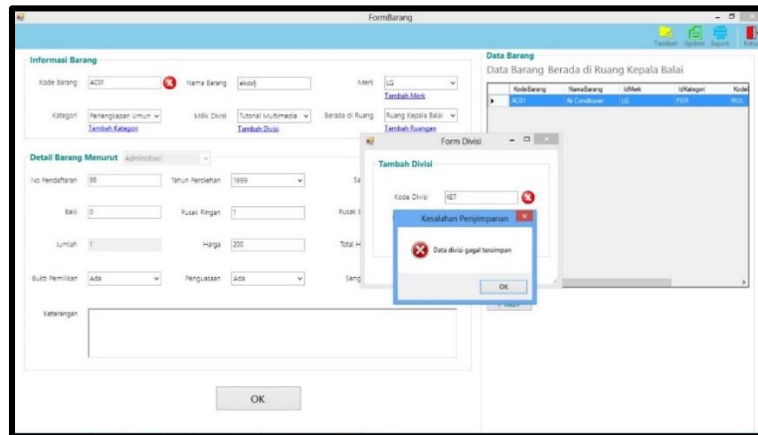
Antarmuka pesan konfirmasi kegagalan penyimpanan *form* divisi ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data divisi.



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.

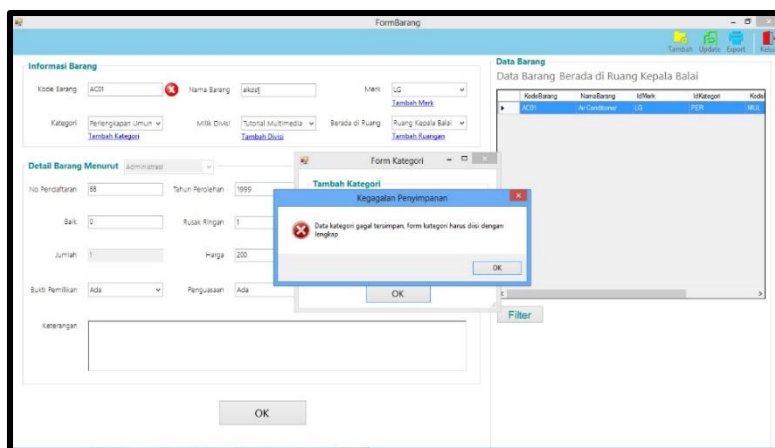
Antarmuka pesan konfirmasi kegagalan penyimpanan kategori *form* tidak lengkap ditampilkan oleh sistem kepada pegawai inventaris ketika adanya kegagalan penyimpanan data kategori diakibatkan oleh inputan data kategori yang tidak lengkap.



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.

The screenshot shows the 'FormBarang' application interface. A modal dialog box titled 'Kesalahan Penyimpanan' (Storage Error) is displayed in the center, with a red 'X' icon and the message 'Data kategori gagal tersimpan' (Category data failed to save). The background form is partially visible, showing the 'Informasi Barang' section with fields for 'Kode Barang' (Barcode), 'Nama Barang' (Item Name), 'Merk' (Brand), 'Kategori' (Category), 'Merk Divisi', 'Berada di Ruang' (Location), 'Detail Barang Menurut' (Item Details), and 'Data Barang' table.

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.

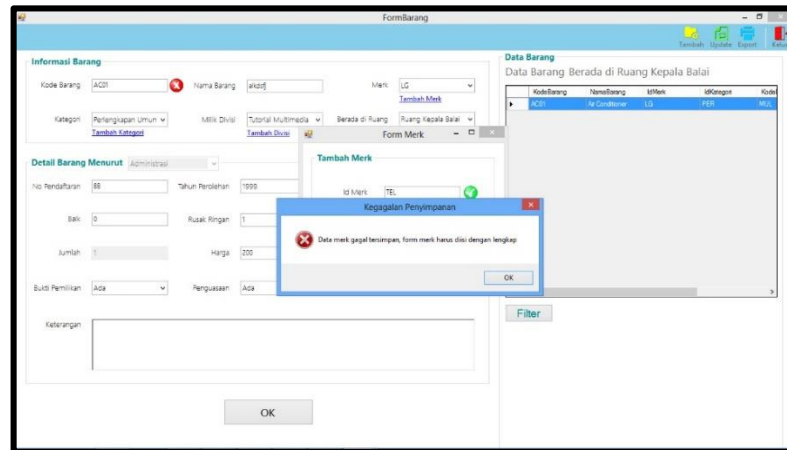
Antarmuka pesan konfirmasi kegagalan penyimpanan kode barang sudah eksis ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan kode barang diakibatkan kode barang sudah eksis.

The screenshot shows the 'FormBarang' application interface. A modal dialog box titled 'Kegagalan Penyimpanan' (Storage Error) is displayed in the center, with a red 'X' icon and the message 'Kode barang yang anda coba simpan sudah eksis pada database, cobalah dengan mengganti nilai kode barang' (The code item you are trying to save already exists in the database, try changing the code item value). The background form is partially visible, showing the 'Informasi Barang' section with fields for 'Kode Barang' (Barcode), 'Nama Barang' (Item Name), 'Merk' (Brand), 'Kategori' (Category), 'Merk Divisi', 'Berada di Ruang' (Location), 'Detail Barang Menurut' (Item Details), and 'Data Barang' table.

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.

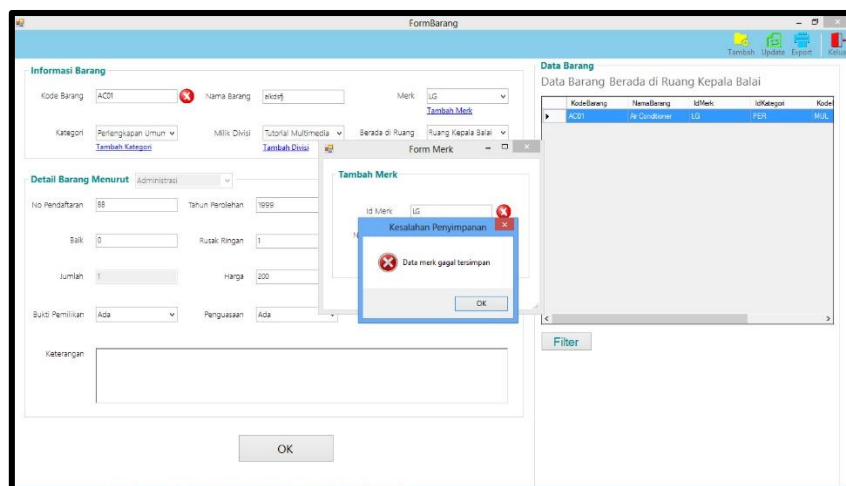
Antarmuka pesan konfirmasi kegagalan penyimpanan merk *form* tidak lengkap ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan merk diakibatkan *form* inputan tidak diisi dengan lengkap.



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

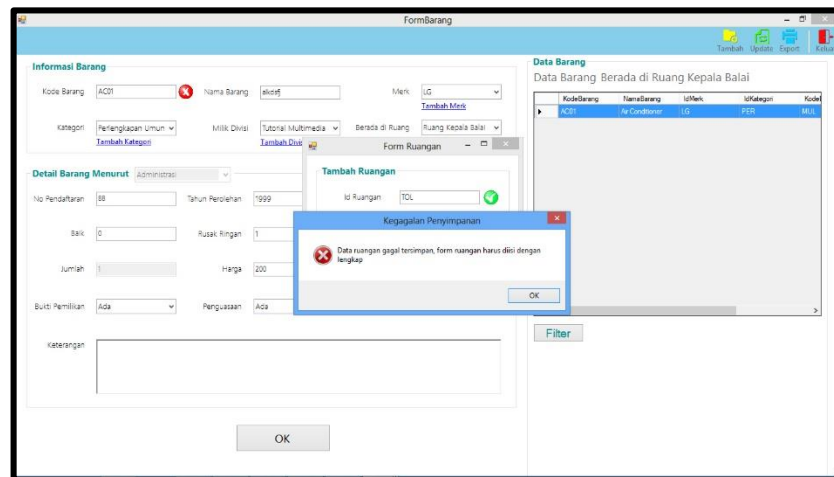
Antarmuka pesan konfirmasi kegagalan penyimpanan merk id sudah eksis ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data merk yang diakibatkan id merk sudah eksis.



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

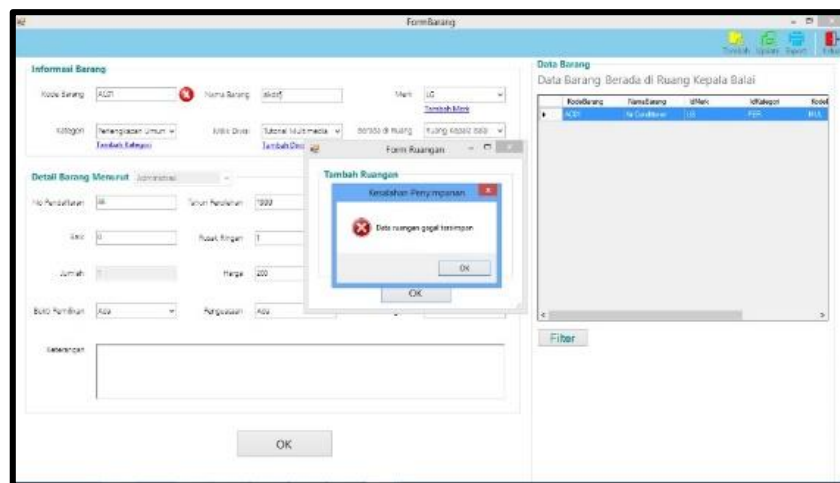
Antarmuka pesan konfirmasi kegagalan penyimpanan *form* ruangan yang tidak lengkap ditampilkan kepada pegawai inventaris ketika adanya kegagalan penyimpanan data ruangan yang diakibatkan inputan pada *form* ruangan tidak lengkap.



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 Ruangan 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