BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem Yang Sedang Berjalan

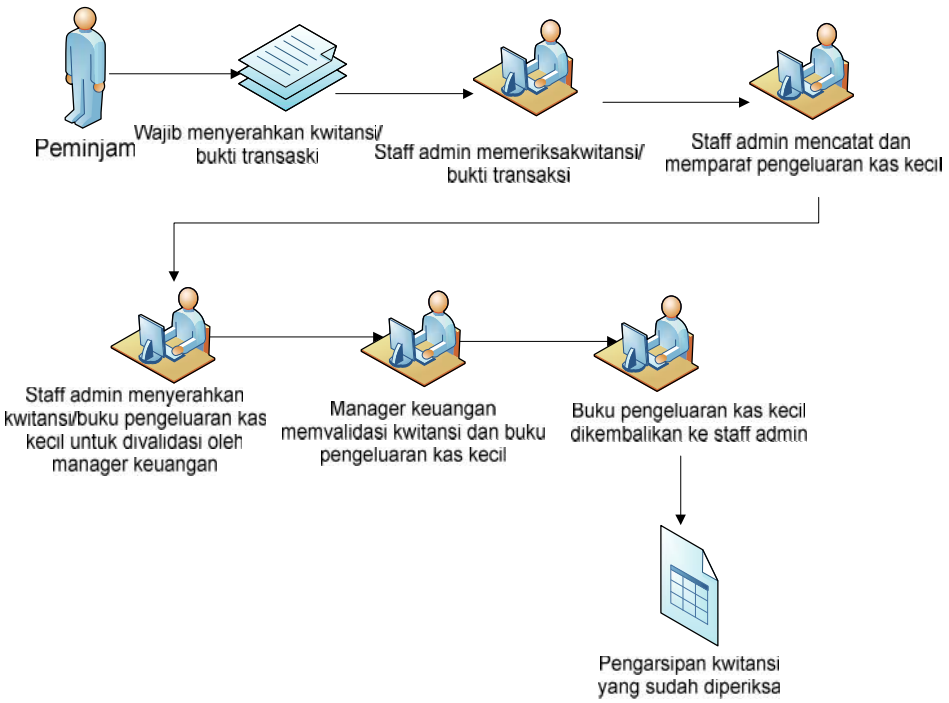
Kegiatan analisis sistem yang berjalan dilakukan dengan analisis yang berorientasi pada objek-objek yang diperlukan oleh sistem yang dirancang, dimaksudkan untuk menitik beratkan kepada fungsi sistem yang berjalan dengan tidak terlalu menitik beratkan kepada alur proses dari sistem. Selanjutnya hasil analisis ini digambarkan dan didokumentasikan dengan metodologi berorientasi objek melalui diagram *use case*, skenario *use case* dan aktifitas diagram. Pertimbangan penggunaan diagram ini karena dianggap diagram-diagram tersebut mewakili secara keseluruhan sistem yang berjalan dan dapat dimengerti oleh *user*.

4.1.1 Analisis Kebutuhan

Sebelum membuat suatu sistem, hendaknya melakukan analisis terlebih dahulu terhadap kebutuhan-kebutuhan apa saja yang diperlukan dengan menggunakan metode-metode yang telah ada.

4.1.1.1 Proses Bisnis Yang Sedang Berjalan

Proses bisnis dari sistem yang sedang berjalan saat ini adalah sebagai berikut :



Gambar 4. 1 Proses Bisnis sistem yang sedang berjalan

Keterangan:

1. Peminjam wajib menyerahkan kwitansi atau bukti transaksi.
2. Staff admin memeriksa dan mensortir kwitansi/bukti transaksi.
3. Staff admin mencatat dan memparaf pengeluaran kas kecil.
4. Staff admin menyerahkan bukti transaksi untuk divalidasi oleh manager keuangan.
5. Manager memvalidasi kwitansi dan buku pengeluaran kas.
6. Buku pengeluaran kas dikembalikan lagi oleh manager keuangan ke staff admin.
7. Pengarsipan kwitansi atau bukti pengeluaran kas kecil.

4.1.1.2 Service Time

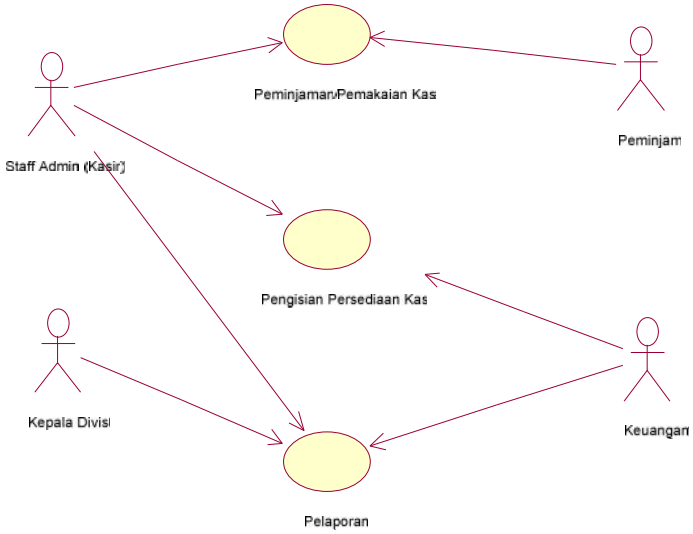
Sistem kas kecil yang ada pada saat ini di PT Suryaniaga Lokalestari masih merupakan sistem kas kecil yang semi komputerisasi. Maksudnya semi komputerisasi adalah proses penginputan dan pelaporan menggunakan excel biasa, tetapi sebagian besar proses pencatatan masih dilakukan secara manual.

Proses yang dijalankan masihlah belum efisien dari segi waktu. Staff admin masih harus melakukan pemeriksaan secara berulang dan melakukan pencatatan pelaporan beberapa kali, serta untuk penyerahan pelaporan masih harus dilakukan secara manual.

4.1.1.3 *Use Case* Diagram Yang Sedang Berjalan

Use case diagram (diagram *use case*) adalah diagram yang menyajikan interaksi antara *use case* dan *actor*. Dimana *actor* dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan – persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

Berilkut ini adalah gambar model *Use Case Diagram* sistem kas kecil yang sedang berjalan :



Gambar 4. 2 Use Case Sistem Kas Kecil yang sedang berjalan

4.1.1.4 Skenario *Use Case* Yang Sedang Berjalan

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Adapun tahapan dari skenario *use case* pada sistem kas kecil yang sedang berjalan adalah sebagai berikut:

Tabel 4. 1 Skenario use case proses peminjaman/pemakaian kas sistem yang sedang berjalan

Judul	Proses peminjaman/pemakaian kas
No. *Use case*	UC-CPETTYCASH-01
Deskripsi *Use Case*	Aktor membuat permohonan untuk pengambilan dana dari kas kecil
Aktor	Staff admin, peminjam
Trigger	Peminjam menyerahkan bukti transaksi.
Kondisi Awal	Staff admin ada dan siap membuat permohonan pengambilan dana.
Skenario Normal	

Aksi- Aktor	Reaksi Sistem
1. Peminjam datang untuk meminta/ meminjam dari kas kecil.	
2. Peminjam menyerahkan bukti transaksi kepada staff admin.	
	3. Staff admin memeriksa kebenaran data yang bukti transaksi.
	4. Staff admin memberikan dana sesuai dengan bukti transaksi.
	5. Staff admin melakukan pencatatan data peminjaman/penggunaan kas kecil.
Kondisi Akhir Skenario Normal: 1. Data peminjaman/penggunaan kas kecil dicatat dalam buku pelaporan, dan bukti transaksi disimpan.	

Tabel 4. 2 Skenario use case membuat pelaporan sistem yang sedang berjalan

Judul	Membuat pelaporan
No. *Use case*	UC-CPETTYCASH-02
Deskripsi *Use Case*	Aktor membuat data pelaporan penggunaan kas kecil
Aktor	Staff admin, Bagian Keuangan, Kepala Divisi
Trigger	Transaksi penggunaan kas kecil selesai
Kondisi Awal	Data bukti transaksi tersedia.

Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Staff admin memeriksa data bukti transaksi.	
2. Staff admin mencatat data transaksi dan data peminjaman/penggunaan kas kecil pada buku.	
3. Staff admin mencatat data penggunaan kas kecil pada program excell.	
	4. Data transaksi tersimpan pada database (excell).
5. Staff Admin menyerahkan data laporan pada manager workshop untuk ditandatangani.	
6. Kepala Divisi menandatangani laporan dan menyerahkan pada bagian keuangan.	
7. Bagian keuangan menerima laporan untuk diarsipkan.	
Kondisi Akhir Skenario Normal:	
Data peminjaman/penggunaan kas kecil dicatat dalam buku pelaporan tersimpan dalam pengarsipan.	

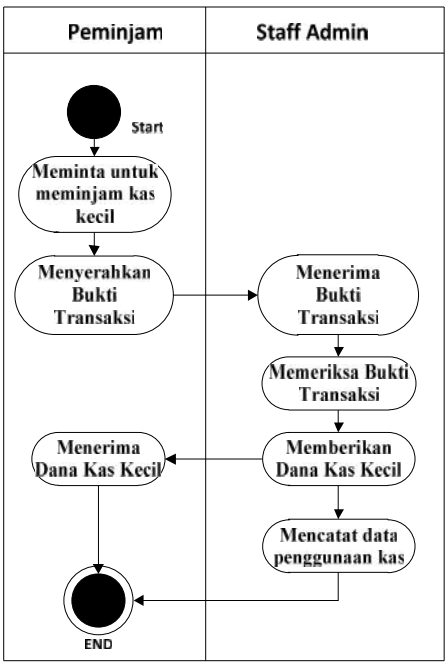
Tabel 4. 3 Skenario use case pengisian persediaan kas sistem yang sedang berjalan

Judul	Pengisian persediaan kas
No. *Use case*	UC-CPETTYCASH-03
Deskripsi *Use Case*	Staff admin membuat permohonan untuk pengisian dana kas kecil
Aktor	Staff admin, bagian keuangan
Trigger	Dana kas kecil mencapai batas minimum.

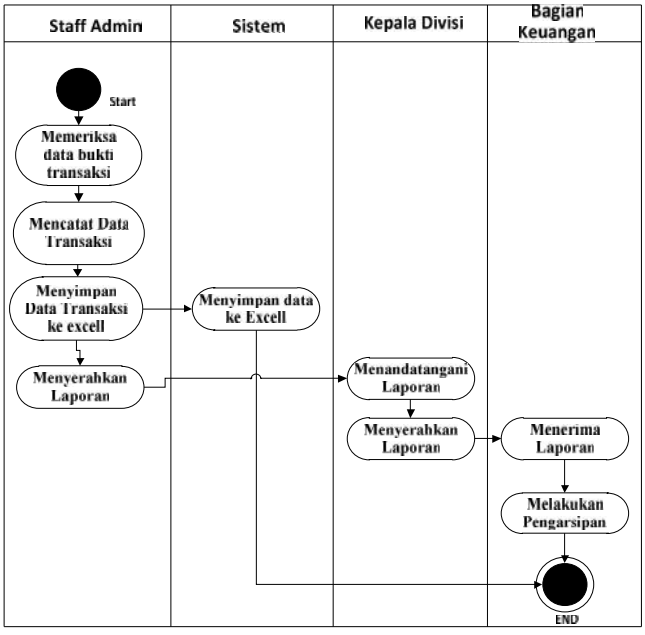
Kondisi Awal	Staff admin membuat permohonan pengisian persediaan dana kas kecil.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Staff admin membuat form permohonan pengisian dana kas kecil dan menyediakan bukti-bukti transaksi yang telah dilakukan.	
2. Staff admin menyerahkan bukti transaksi dan form permohonan kepada bagian keuangan.	
	3. Bagian keuangan melakukan pemeriksaan form dan bukti-bukti transaksi.
	4. Bagian keuangan memberikan dana untuk pengisian dana kas kecil sesuai dengan data transaksi yang diterima.
Kondisi Akhir Skenario Normal: Dana kas kecil kembali terisi.	

4.1.1.5 *Activity* Diagram Yang Sedang Berjalan

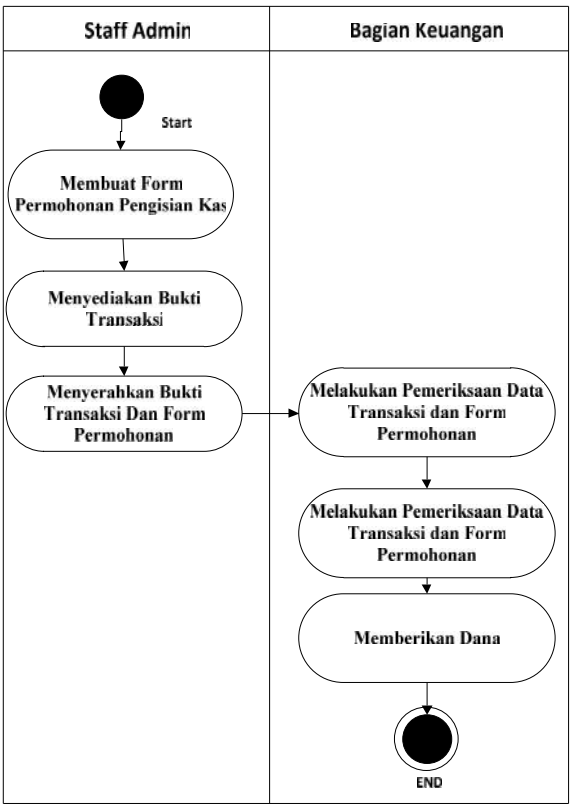
Pada bagian ini akan digambarkan dokumentasi alur kerja pada sistem yang sedang berjalan yang bertujuan untuk melihat alur proses sistem yang sedang berjalan.



Gambar 4. 3 Proses peminjaman/pemakaian kas sistem yang sedang berjalan



Gambar 4. 4 Proses Membuat laporan sistem yang sedang berjalan



Gambar 4. 5 Proses pengisian persediaan kas sistem yang sedang berjalan

4.1.2 Evaluasi Sistem Yang Sedang Berjalan

Dari hasil identifikasi yang telah dilakukan, sistem yang berjalan pada saat ini memiliki beberapa kelemahan yang berdampak kepada produktivitas perusahaan. Berikut ini adalah uraian permasalahannya dan sebuah rancangan pemecahan masalahnya. Dimana sistem informasi pengeluaran kas kecil akan dirancang sebagai solusi masalah yang akan diajukan :

Tabel 4. 4 Evaluasi Sistem Yang Sedang Berjalan

No	Permasalahan	Bagian	Pemecahan
1	Sistem informasi pengeluaran kas kecil di PT. Suryaniaga Lokalestari masih dilakukan secara manual.	Admin / Bag. Keuangan	Membuat sistem informasi pengeluaran kas kecil yang terkomputerisasi

2	Belum maksimalnya pengolahan dan pencatatan data pengeluaran kas kecil	Admin / Bag. Keuangan	Membuat sistem informasi yang dapat mengolah dan menginput data pengeluaran kas dengan baik
3	Masih terdapat kesalahan perhitungan dalam sistem keuangan saat ini karena *human error*.	Admin / Bag. Keuangan	Membuat sistem informasi pengeluaran kas kecil dengan perhitungan yang otomatis
4	Belum maksimalnya tahapan pembuatan laporan pengeluran kas kecil.	Admin / Bag. Keuangan	Dengan adanya sistem informasi pengeluaran kas diharapkan bisa memaksimalkan pembuatan laporan pengeluaran kas yang baik

4.2 Perancangan Sistem

Perancangan sistem adalah gambaran, perancangan dan pembuatan skema atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan mempunyai fungsi dan tujuan. Elemen-elemen sistem informasi dirancang dengan tujuan untuk dikomunikasikan kepada *user*.

Dalam perancangan sistem dapat berarti menyusun suatu sistem yang baru untuk menambah kinerja sistem yang ada, baik secara keseluruhan maupun meningkatkan kinerja sistem yang telah ada.

Pada tahap perancangan sistem ini akan dijelaskan mengenai perancangan sistem pada objek yang digunakan, perancangan arsitektur program yang akan dibuat, perancangan tampilan dan perancangan menu.

4.2.1 Tujuan Perancangan Sistem yang Diusulkan

Tujuan utama dari perancangan sistem adalah untuk memberikan gambaran secara umum kepada pemakai mengenai sistem informasi yang baru. Perancangan

sistem merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan sesuatu yang baru atau memperbaharui sistem yang ada untuk meningkatkan kinerja sistem itu sendiri, agar dapat memenuhi hasil yang diinginkan. Rancangan sistem yang baru, akan diterapkan suatu kegiatan untuk menemukan dan mengembangkan metoda, prosedur dan proses suatu data agar tujuan dari suatu organisasi dapat tercapai.

Adapun tujuan dari perancangan sistem ini adalah untuk menghasilkan suatu rancangan sistem yang memperbaharui sistem yang sudah ada untuk memaksimalkan proses pengeluaran kas kecil yang sedang berjalan pada PT Suryaniaga Lokalestari.

4.2.2 Gambaran Umum Sistem yang Diusulkan

Gambaran umum tentang sistem yang diusulkan pada proses perancangan ini adalah suatu sistem yang digunakan untuk mengolah data transaksi pengeluaran kas kecil pada PT Suryaniaga Lokalestari dan menangani proses pelaporan dengan harapan mampu menangani permasalahan yang ada pada sistem yang sedang berjalan, terkhusus pada optimalisasi waktu yang dibutuhkan dan membantu dalam proses pelaporan dan penyimpanan data transaksi pengeluaran kas kecil.

Sistem ini diharapkan akan membantu efektifitas dan optimalisasi waktu serta menghindari adanya ketidaksesuaian data atau kemungkinan data *miss* pada kas kecil PT Suryaniaga Lokalestari. Sistem informasi ini diharapkan dapat membantu mengatasi masalah yang ada, dan dapat menghasilkan informasi yang cepat, tepat, dan akurat.

4.2.3 Perancangan Prosedur yang Diusulkan

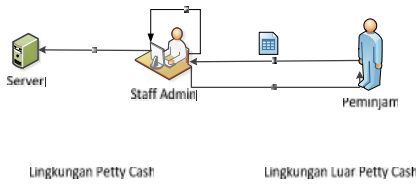
Perancangan Prosedur merupakan awal dari pembuatan sistem yang akan dibuat, dimana dapat dilihat proses-proses apa saja yang nantinya diperlukan dalam pembuatan suatu sistem. Sedangkan perancangan prosedur yang diusulkan merupakan tahap untuk memperbaiki atau meningkatkan efisiensi kerja.

Tahap perancangan sistem yang digambarkan merupakan tahap perancangan untuk membangun suatu sistem dan mengkonfigurasikan komponen-komponen perangkat lunak dan perangkat keras sehingga menghasilkan sistem yang baik. Sistem yang dirancang tersebut menjadi satu komponen.

Tahapan perancangan prosedur ini akan dijelaskan dengan menggunakan pemodelan sistem informasi berorientasi objek dengan menggunakan UML.

4.2.3.1 Proses Bisnis Yang diusulkan

Proses bisnis dari sistem yang diusulkan adalah sebagai berikut:

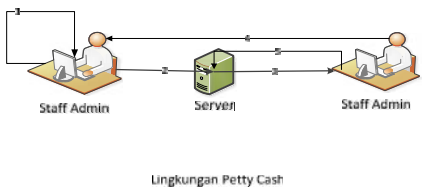


Gambar 4. 6 Proses Bisnis Sistem yang diusulkan Peminjaman/Penggantian

Keterangan:

1. Peminjam datang pada staff admin/kasir untuk meminta penggantian dari petty cash dan menyerahkan bukti transaksi.

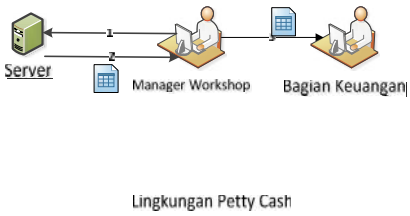
2. Staff Admin/kasir memeriksa bukti transaksi dan memasukkan data ke dalam sistem.
3. Sistem menyimpan data transaksi baru.
4. Staff Admin/kasir menyerahkan penggantian dari petty cash kepada peminjam.



Gambar 4. 7 Proses Bisnis Sistem yang diusulkan Pengisian Petty Cash

Keterangan:

1. Staff Admin/kasir mengisi permohonan data pengisian dana kembali untuk kas kecil.
2. Staff Admin/kasir menyimpan data permohonan dan menyimpannya ke server.
3. Sistem memberitahu bagian keuangan bahwa ada permohonan pengisian kembali kas kecil.
4. Bagian keuangan mengkonfirmasi dan memberikan pengisian kembali petty cash kepada Staff Admin.
5. Bagian keuangan menyimpan data transaksi penggantian dan sistem akan mengupdate data saldo dari petty cash.



Gambar 4. 8 Proses Bisnis Sistem yang diusulkan Untuk Pelaporan

Keterangan:

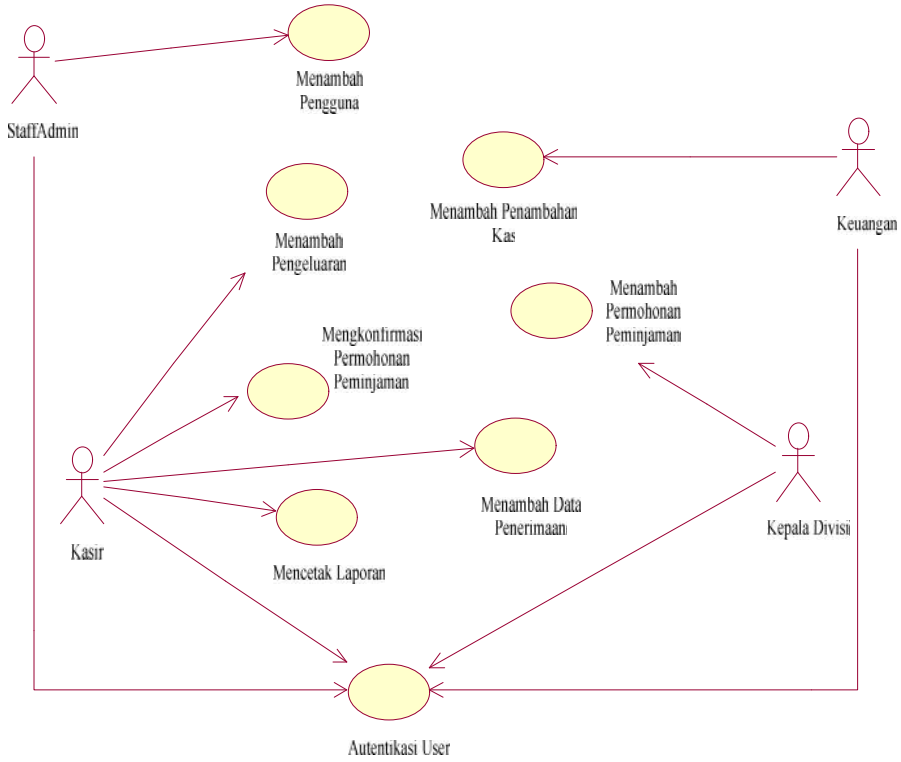
1. Manager workshop melakukan permintaan untuk laporan dengan memasukkan periode yang diinginkan.
2. Sistem memberikan laporan sesuai dengan yang diinginkan.
3. Manager Workshop menandatangani laporan dan menyerahkan laporan pada bagian keuangan.

4.2.3.2 *Service Time*

Dengan sistem transaksi dan pelaporan yang ditargetkan, proses transaksi peminjaman, pengisian kembali petty cash, dan pelaporan akan mampu dioptimalisasikan waktu yang dibutuhkannya dan database data transaksi akan tersimpan lebih baik. Hal ini dikarenakan proses transaksi baik pengeluaran maupun pengisian ulang dari petty cash sudah dapat dikelola dengan baik.

4.2.3.3 *Use Case* Diagram Yang Diusulkan

Use case diagram (diagram *use case*) adalah diagram yang menyajikan interaksi antara use case dan actor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. Use case menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.



Gambar 4. 9 Use Case Diagram untuk Sistem yang diusulkan

4.2.3.4 Skenario *Use Case Yang Diusulkan*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut.

4.2.3.4.1 Skenario Menambah Data Pengguna

Tabel 4. 5 Tabel skenario use case menambah data Pengguna

Judul	Menambah Pengguna
No. *Use case*	UC-PETTYCASH-01
Deskripsi *Use Case*	Aktor menambahkan data Pengguna.
Aktor	Staff Admin
Trigger	Staff Admin memilih menu untuk menambahkan data pengguna.
Kondisi Awal	Aplikasi menampilkan *form* menu utama.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor memilih menu untuk menambahkan data pengguna.	
	2. Sistem menampilkan form untuk menambahkan data pengguna.
3. Aktor mengisi data pengguna yang baru.	
4. Aktur men-*submit* data pengguna yang baru.	5. Sistem memeriksa masukan data.
	6. Sistem menyimpan data pengguna yang baru pada *database*.
Kondisi Akhir Skenario Normal: Data pengguna yang baru dimasukkan tampil pada list data pengguna.	
Skenario Abnormal-1	

Aksi- Aktor	Reaksi Sistem
1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi *field-field* pada form.	2. *Form* pengisian data pengguna ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data pengguna tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data pengguna.	

4.2.3.4.2 Skenario Menambah Data Pengeluaran

Tabel 4. 6 Tabel skenario use case menambah data Pengeluaran

Judul	Menambah data Pengeluaran
No. *Use case*	UC-PETTYCASH-02
Deskripsi *Use Case*	Aktor menambahkan data Pengeluaran.
Aktor	Kasir
Trigger	Staff Admin memilih menu untuk menambahkan data pengeluaran.
Kondisi Awal	Aplikasi menampilkan *form* menu utama.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor memilih menu untuk menambahkan data pengeluaran.	
	2. Sistem menampilkan form untuk menambahkan data pengeluaran.

3. Aktor mengisi data pengeluaran yang baru.	
4. Aktur men-*submit* data pengeluaran yang baru.	5. Sistem memeriksa data saldo dari petty cash.
	6. Sistem menyimpan data pengeluaran yang baru pada *database* dan mengurangi saldo sesuai dengan data transaksi.
Kondisi Akhir Skenario Normal: Data transaksi yang baru dimasukkan tampil pada list data transaksi, dan data saldo akhir untuk petty cash berkurang sesuai dengan data transaksi.	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi *field-field* pada form.	2. *Form* pengisian data transaksi ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data transaksi tidak ditambahkan pada *database*.
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data transaksi.	

4.2.3.4.3 Skenario menambah permohonan peminjaman

Tabel 4. 7 Skenario use case menambah permohonan peminjaman

Judul	Menambah Permohonan Peminjaman
No. *Use case*	UC-PETTYCASH-03
Deskripsi *Use Case*	Aktor menambahkan data Permohonan peminjaman ulang petty cash.

Aktor	Kepala Divisi
Trigger	Kepala divisi memilih menu untuk menambahkan data permohonan.
Kondisi Awal	Aplikasi menampilkan *form* menu utama.

Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor memilih menu untuk menambahkan data permohonan.	
	2. Sistem menampilkan form untuk menambahkan data permohonan.
3. Aktor mengisi data permohonan yang baru.	
4. Aktor men-*submit* data permohonan yang baru.	
	5. Sistem menyimpan data permohonan yang baru pada *database*.
Kondisi Akhir Skenario Normal: Data permohonan yang baru dimasukkan tampil pada list data permohonan.	

Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi *field-field* pada form.	2. *Form* pengisian data permohonan ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data permohonan tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data permohonan.	

4.2.3.4.4 Mengkonfirmasi data permohonan

Tabel 4. 8 Skenario use case mengkonfirmasi data permohonan

Judul	Mengkonfirmasi data permohonan
No. *Use case*	UC-PETTYCASH-04
Deskripsi *Use Case*	Aktor mencari dan menampilkan data permohonan untuk dikonfirmasi.
Aktor	Bagian Keuangan
Trigger	Aktor memilih menu untuk mengkonfirmasi data permohonan.
Kondisi Awal	Aplikasi menampilkan *form* utama.

Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor memilih menu untuk mengkonfirmasi data permohonan.	2. Sistem menampilkan data permohonan yang sudah dibuat.
3. Aktor melakukan pemeriksaan data permohonan peminjaman uang kas.	
4. Aktor mengkonfirmasi data permohonan peminjaman	5. Sistem mengurangi data saldo pada sistem dan data permohonan berubah statusnya menjadi telah dikonfirmasi
Kondisi Akhir Skenario Normal: Sistem mengurangi data saldo pada sistem dan data permohonan berubah statusnya menjadi telah dikonfirmasi	

Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
Kondisi Akhir Skenario Abnormal-1:	

Catatan	
1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi *field-field* pada form.	2. *Form* pengisian data permohonan ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data permohonan tidak ditambahkan pada *database*.
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data permohonan.	

4.2.3.4.5 Menambah data penerimaan

Tabel 4. 9 Skenario use case menambah data penerimaan

Judul	Menambah data penerimaan
No. *Use case*	UC-PETTYCASH-05
Deskripsi *Use Case*	Aktor mengolah menambah data penerimaan.
Aktor	Kasir
Trigger	Aktor memilih menu untuk menambahkan data penerimaan
Kondisi Awal	Aplikasi menampilkan *form* utama.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengisi *data* yang ingin ditambahkan pada *form* dari data penerimaan yang diinginkan.	

2. Aktor men-*submit* data penerimaan yang ingin ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data penerimaan telah diisi dan sesuai.
	4. Data penerimaan ditambahkan pada *database*.
Kondisi Akhir Skenario Normal: Data penerimaan tersimpan pada database.	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi atau mengisi data yang salah *field-field* pada form.	2. *Form* pengisian data penerimaan ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi atau ada data penerimaan yang salah.
	3. Data penerimaan tidak ditambahkan pada *database*.
Kondisi Akhir Skenario Abnormal-1: Applikasi menampilkan halaman menambahkan data penerimaan	
Catatan	
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data permohonan.	

4.2.3.4.6 Autentikasi User

Tabel 4. 10 Skenario use case autentikasi user

Judul	Autentikasi User
No. *Use case*	UC-PETTYCASH-05
Deskripsi	Aktor melakukan autentikasi user untuk penentuan hak akses.

Use Case	
Aktor	Staff Admin, Bagian Keuangan, Kepala Divisi, Kasir
Trigger	Aktor memilih untuk melakukan autentikasi user.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk autentikasi user.

Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengisi data login pada *field* yang disediakan.	
2. Aktor men-*submit* data login yang baru ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data penjualan telah diisi dan sesuai.
	4. Data proses login diperiksa dan menampilkan pesan autentikasi user sesuai.
	5. Data login menentukan hak akses masing-masing user
Kondisi Akhir Skenario Normal: Applikasi menampilkan *form* dari menu utama	

Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* untuk menambahkan data transaksi penjualan.	
Catatan	
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data permohonan.	

4.2.3.4.7 Mencetak Pelaporan

Tabel 4. 11 Skenario use case Mencetak Pelaporan

Judul	Mencetak Laporan Pelaporan
No. *Use case*	UC-PETTYCASH-06
Deskripsi *Use case*	Aktor mencetak laporan transaksi yang terjadi pada range waktu yang diinginkan.
Aktor	Kasir, Keuangan, Kepala Divisi.
Trigger	Aktor memilih menu untuk mencetak laporan manager.
Kondisi Awal	Aplikasi menampilkan form untuk menentukan periode laporan yang diinginkan.

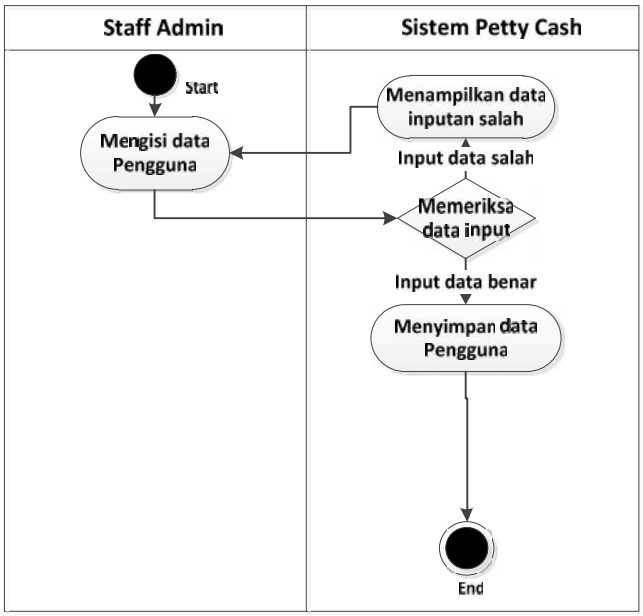
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengisi periode waktu yang diinginkan untuk laporan.	
2. Aktor menekan tombol untuk mencetak laporan	
	3. Sistem mencetak laporan berdasarkan transaksi dan penyimpanan data yang terjadi pada periode yang dipilih.
Kondisi Akhir Skenario Normal: Laporan Petty Cash dicetak.	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor memilih periode yang tidak memiliki data penjualan samasekali.	

2. Aktor menekan tombol untuk mencetak laporan	
	3. Sistem menampilkan pesan bahwa transaksi penjualan pada periode tersebut kosong.
Kondisi Akhir Skenario Abnormal-1: Sistem menampilkan form untuk melakukan pengisian periode laporan yang diinginkan.	

4.2.3.5 Activity Diagram Yang Diusulkan

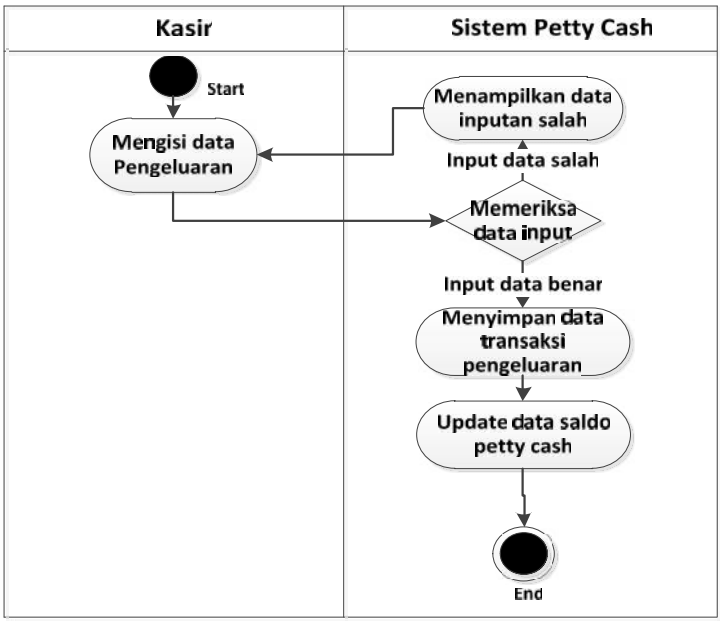
Activity diagram (diagram aktivitas) adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis *(business work flow).* Dapat juga digunakan untuk menggambarkan aliran kejadian *(flow of events).*

4.2.3.5.1 Activity Diagram Menambah Data Pengguna



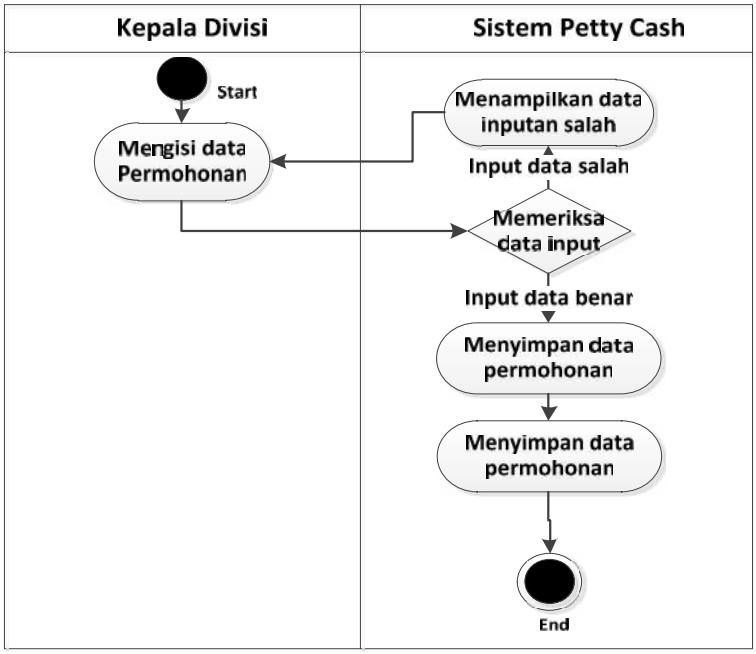
Gambar 4. 10 Activity Diagram Menambah Data Pengguna

4.2.3.5.2 Activity Diagram Menambah Data Pengeluaran



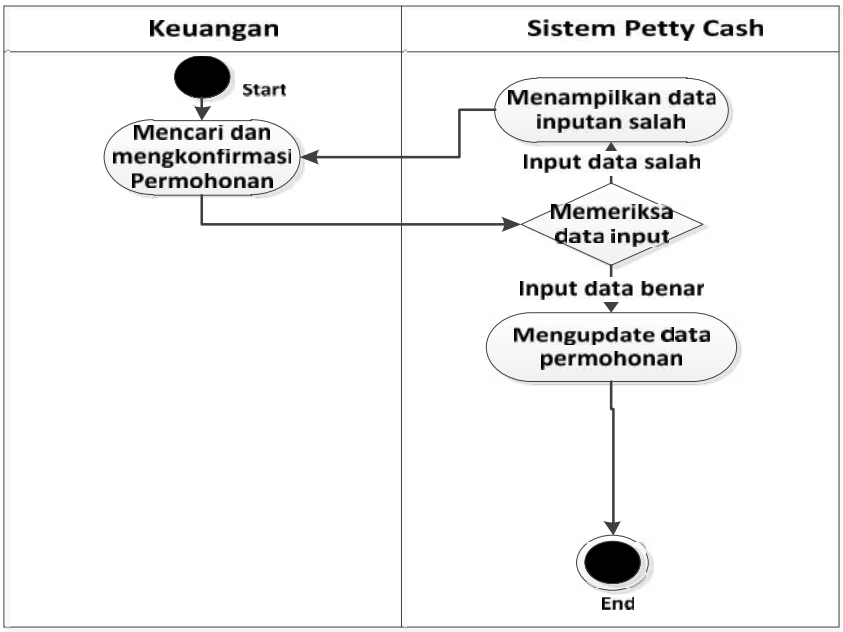
Gambar 4. 11 Activity Diagram Menambah Data Pengeluaran

4.2.3.5.3 Activity Diagram Menambah Data Permohonan Peminjaman



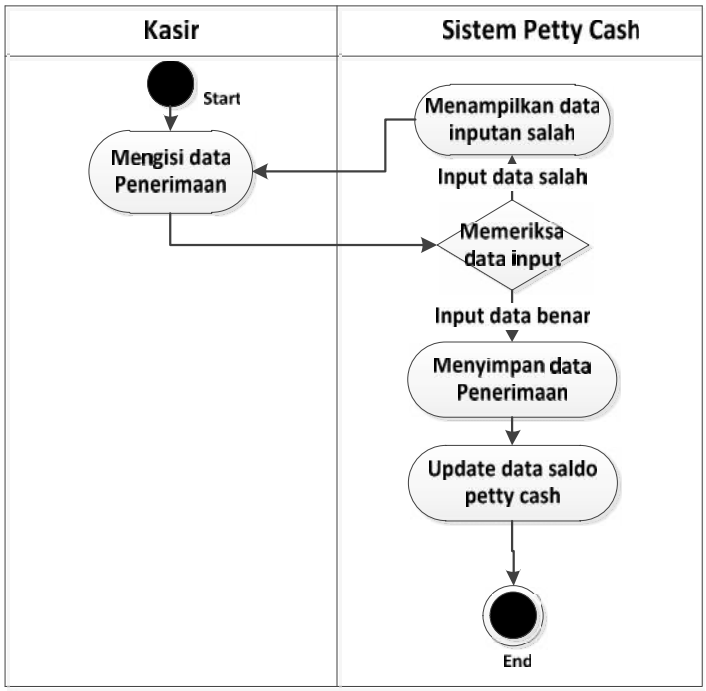
Gambar 4. 12 Activity Diagram Menambah Data Permohonan

4.2.3.5.4 Activity Diagram Mengkonfirmasi Data Permohonan



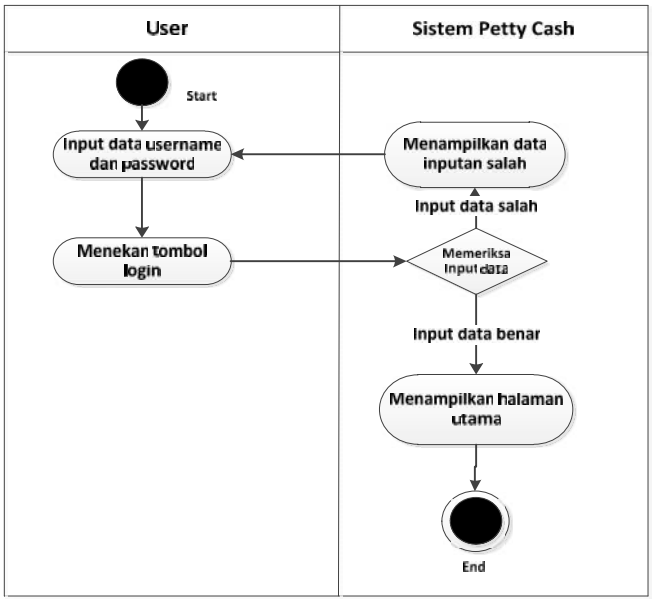
Gambar 4. 13 Mengkonfirmasi Data Permohonan

4.2.3.5.5 Activity Diagram Menambah Data Penerimaan



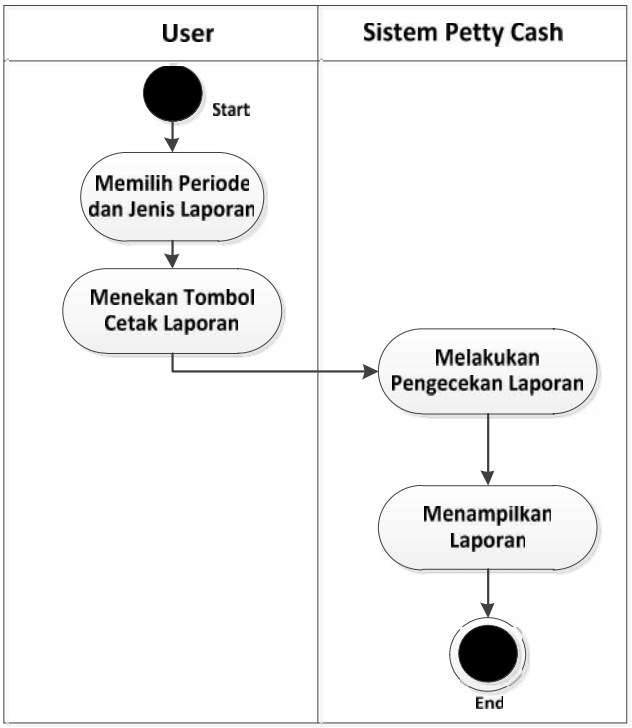
Gambar 4. 14 Menambah Data Penerimaan

4.2.3.5.6 Activity Diagram Autentikasi User



Gambar 4. 15 Activity Diagram Autentikasi User

4.2.3.5.7 Activity Diagram Mencetak data Pelaporan

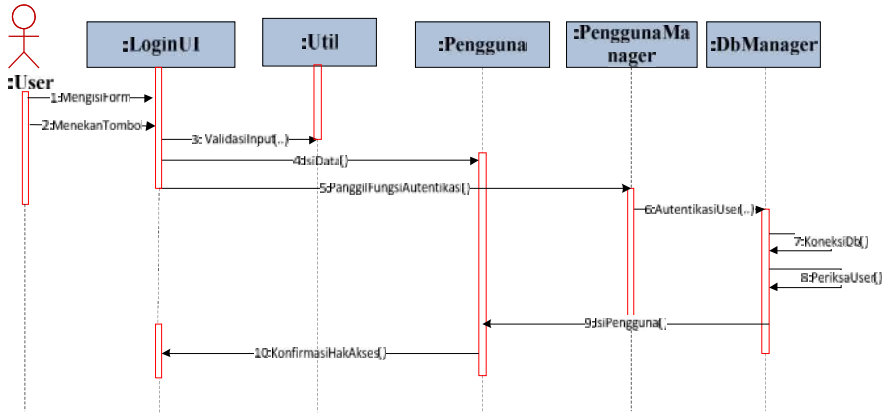


Gambar 4. 16 Activity Diagram Mencetak data Pelaporan

4.2.3.6 Sequence Diagram Yang Diusulkan

Pada setiap *sequence diagram* terdapat aksi aktor yang pertama sekali adalah terhadap *interface*. *Sequence Diagram* digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya *sequence diagram* digunakan dalam lapisan abstraksi model objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar object, juga interaksi antar objek, dan menunjukkan sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama, pesan diwakili oleh garis dengan tanda panah, dan waktu yang ditunjukkan dengan proses vertikal. Berikut adalah *sequence diagram* yang ada pada sistem apotik yang ditargetkan.

4.2.3.6.1 Sequence Diagram Autentikasi User



Gambar 4. 17 Sequence diagram Autentikasi User

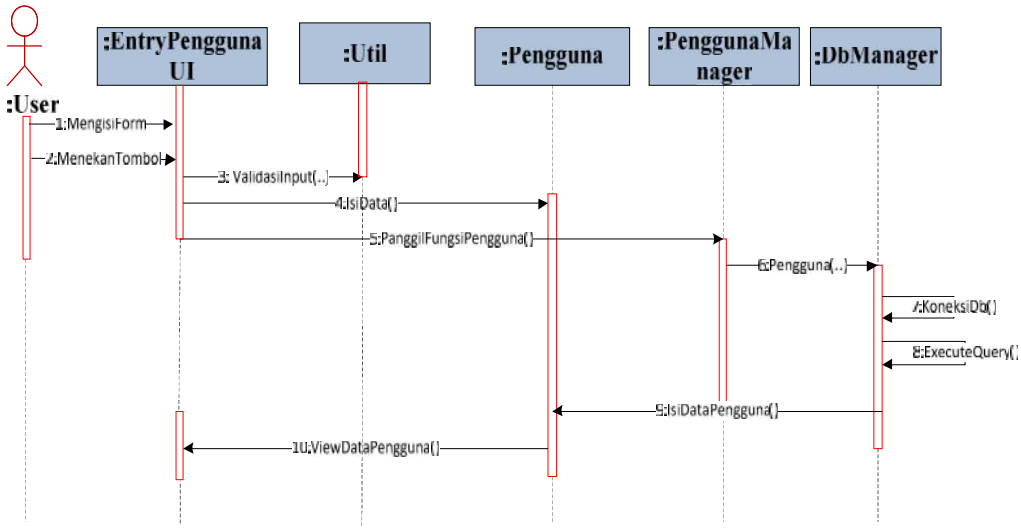
Keterangan:

Pada Gambar 4.17 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. LoginUI

2. Util
3. Pengguna
4. PenggunaManager
5. DbManager

4.2.3.6.2 Sequence Diagram menambah Pengguna



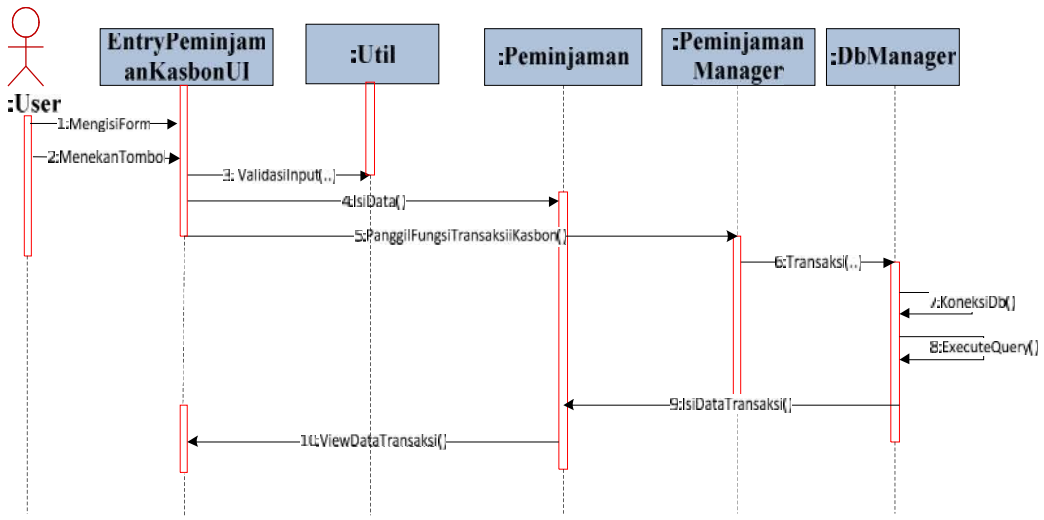
Gambar 4. 18 Sequence diagram Mengolah Pengguna

Keterangan:

Pada Gambar 4.18 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. EntryPenggunaUI
2. Util
3. Pengguna
4. PenggunaManager
5. DbManager

4.2.3.6.3 Sequence Diagram menambah data pengeluaran (Peminjaman Kasbon)



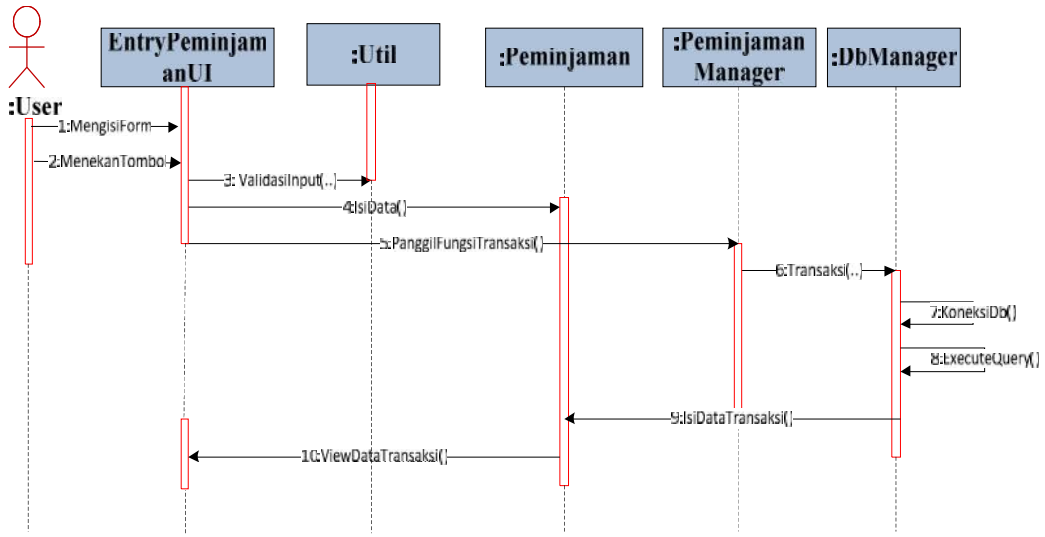
Gambar 4. 19 Sequence diagram Mengolah Pengeluaran (Peminjaman Kasbon)

Keterangan:

Pada Gambar 4.19 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. EntryPeminjamanKasbonUI
2. Util
3. Peminjaman
4. TransaksiManager
5. DbManager

4.2.3.6.4 Sequence Diagram manage pengeluaran (peminjaman biasa)



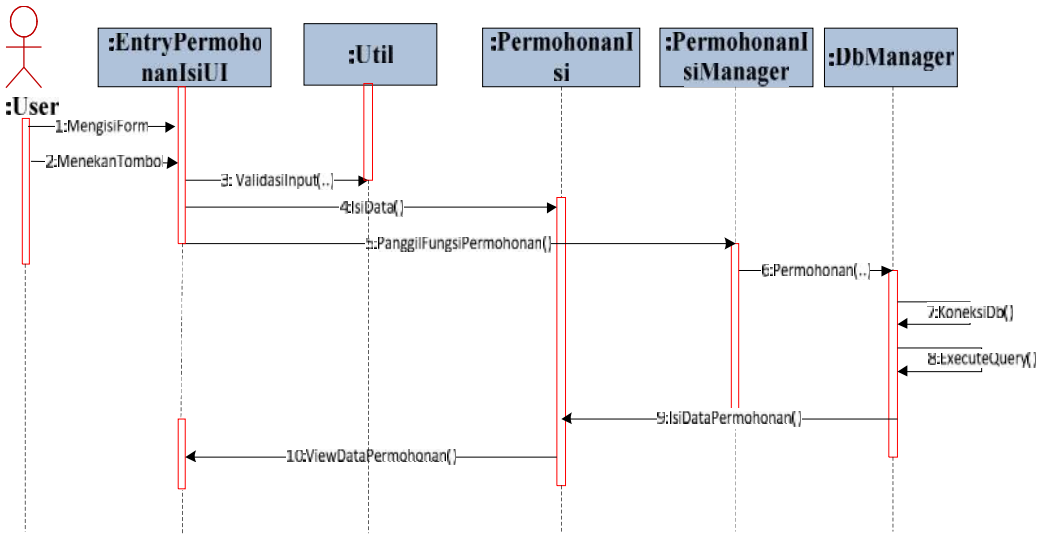
Gambar 4. 20 Sequence diagram Mengolah Pengeluaran (peminjaman biasa)

Keterangan:

Pada Gambar 4.20 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. EntryPeminjamanUI
2. Util
3. Peminjaman
4. TransaksiManager
5. DbManager

4.2.3.6.5. Sequence Diagram menambah pemohonan pengisian



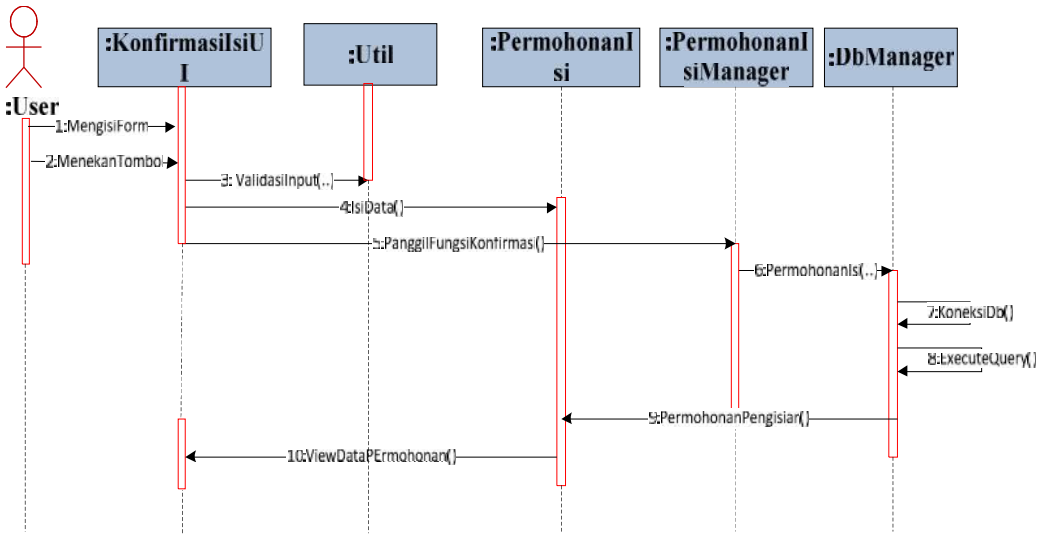
Gambar 4. 21 Sequence diagram Menambah Permohonan Pengisian kas

Keterangan:

Pada Gambar 4.21 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. EntryPermohonanIsiUI
2. Util
3. Permohonan
4. PermohonanIsiManager
5. DbManager

4.2.3.6.6 Sequence Diagram Mengkonfirmasi Pengisian



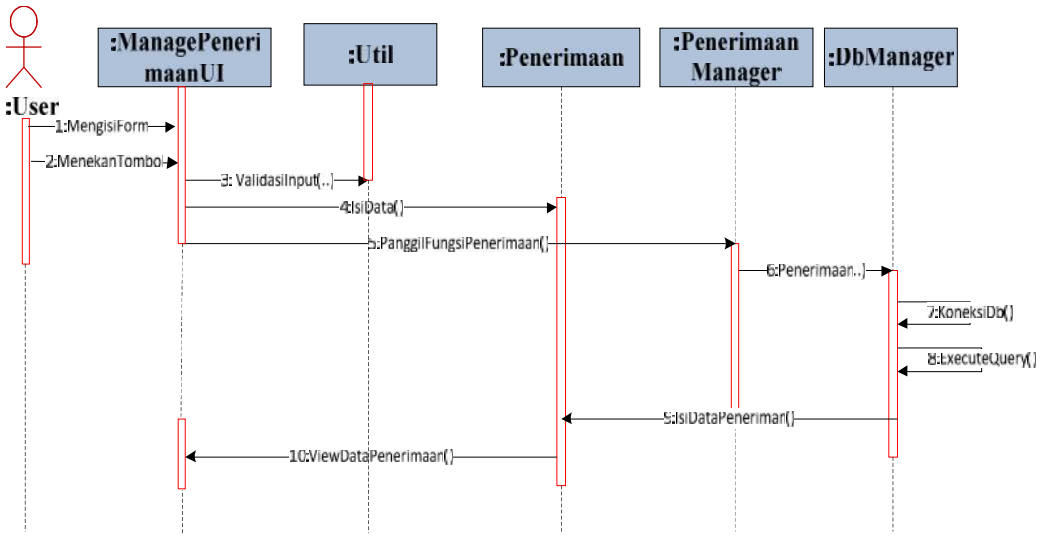
Gambar 4. 22 Sequence diagram Mengkonfirmasi Pengisian kas

Keterangan:

Pada Gambar 4.22 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. EntryPenambahanUI
2. Util
3. Penambahan
4. TransaksiManager
5. DbManager

4.2.3.6.7 Sequence Diagram Menambah Data Penerimaan



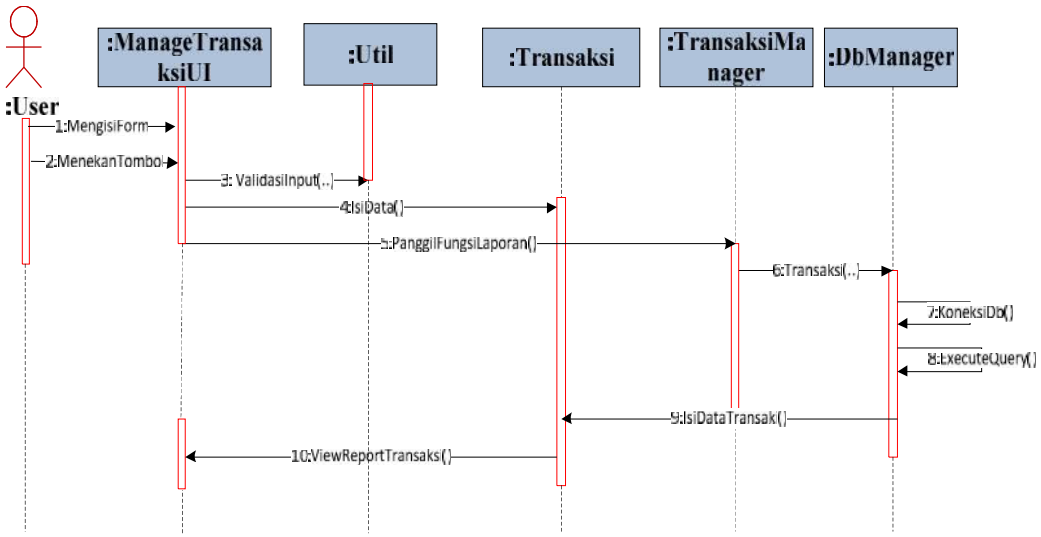
Gambar 4. 23 Sequence diagram Mengolah Penambahan/penggantian kas

Keterangan:

Pada Gambar 4.23 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. EntryPenerimaanUI
2. Util
3. Penerimaan
4. PenerimaanManager
5. DbManager

4.2.3.6.8 Sequence Diagram mencetak pelaporan



Gambar 4. 24 Sequence diagram Mengolah Pelaporan

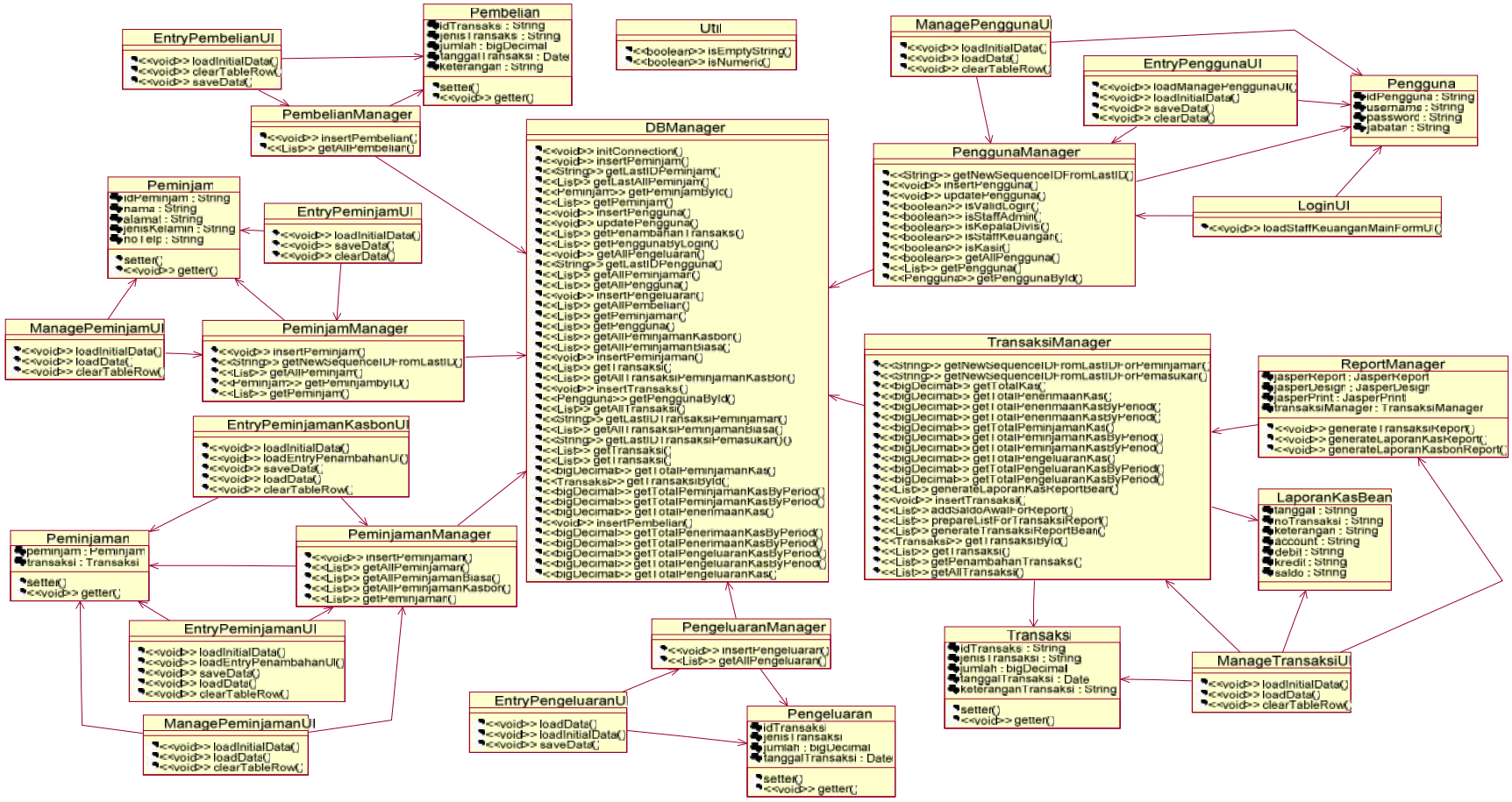
Keterangan:

Pada Gambar 4.24 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. ManagerTransaksiUI
2. Util
3. Transaksi
4. TransaksiManager
5. DbManager

4.2.3.7 Class Diagram Yang Diusulkan

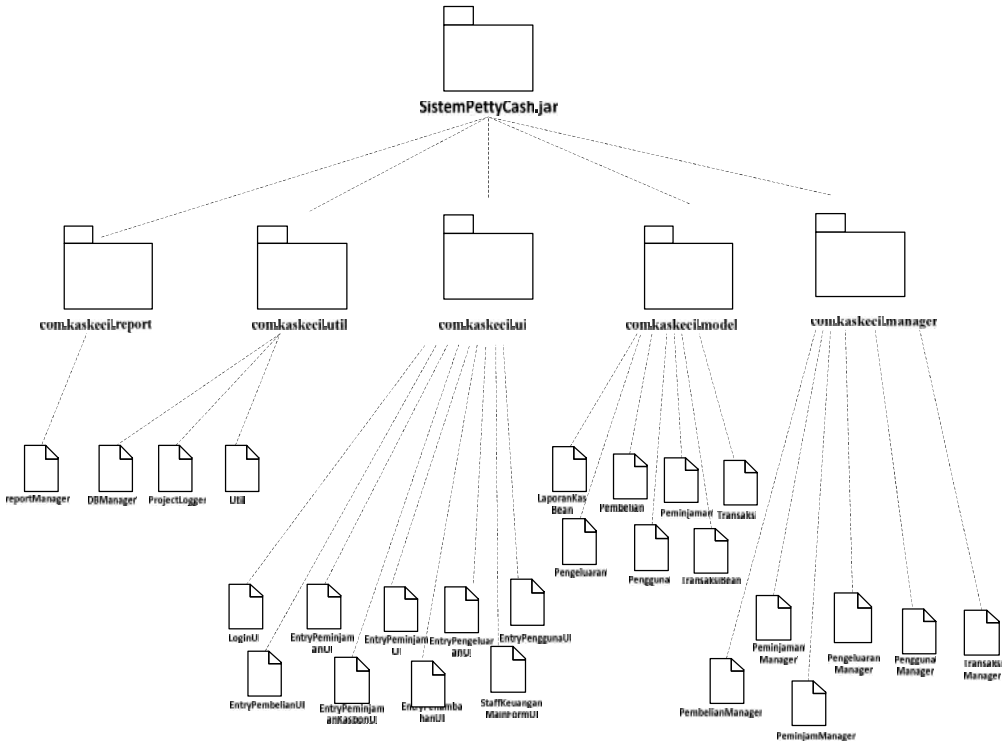
Class merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem pemasaran produk dari bagian penjualan beserta proses-proses yang terjadi, maka dapat dibuat *Class Diagram* Berikut *Class Diagram* Sistem Petty Cash yang diusulkan:



Gambar 4. 25 Class Diagram

4.2.3.8 Component Diagram Yang Diusulkan

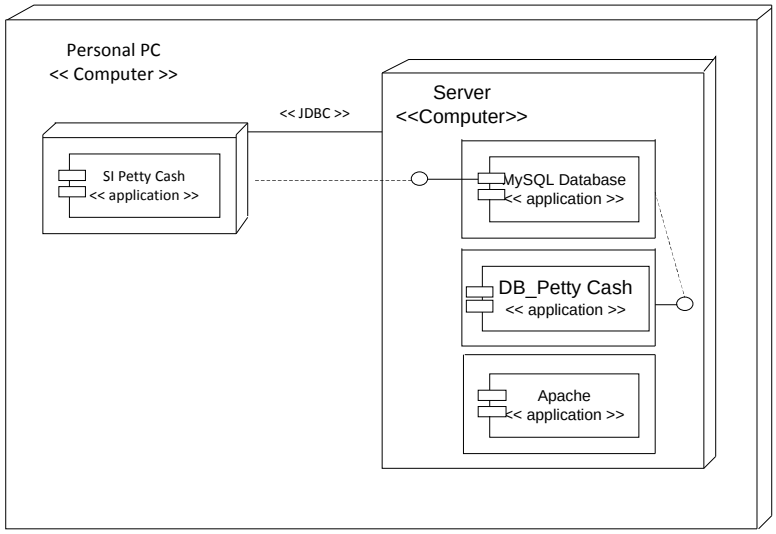
Component diagram menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*dependency*) di antaranya. *Component* piranti lunak adalah modul berisi *code*, baik *berisi source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Berikut ini adalah *component diagram* yang dibutuhkan :



Gambar 4. 26 Component Diagram

4.2.3.9 *Deployment* Diagram Yang Diusulkan

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. *Deployment diagram* berhubungan dengan diagram komponen dimana *deployment diagram* memuat satu atau lebih komponen-komponen.



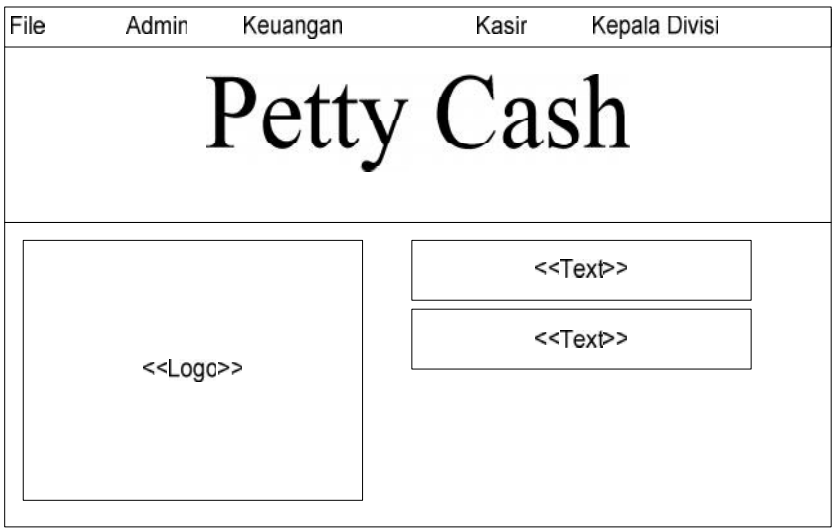
Gambar 4. 27 Deployment diagram sistem Petty Cash Diusulkan

4.2.4 Perancangan Antar Muka

Perancangan antar muka merupakan perancangan yang dibuat sebelum program aplikasi dibuat, perancangan antar muka pada sistem Petty Cash Untuk PT Suryaniaga Lokalestari adalah sebagai berikut:

4.2.4.1 Perancangan Struktur Menu

Perancangan menu dibuat sebagai alat antar muka dengan pengguna untuk memudahkan pengoperasian perangkat lunak. Berikut rancangan menu perangkat lunak ini:



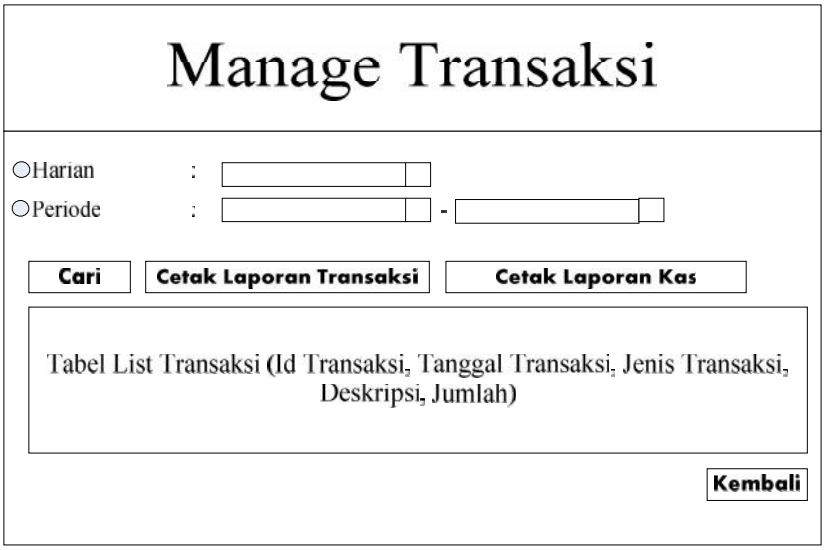
Gambar 4. 28 Perancangan menu sistem yang Diusulkan

4.2.4.2 Perancangan Input

Perancangan input merupakan perancangan tampilan yang akan digunakan guna memasukkan data pada sistem untuk kemudian diproses. Dalam perancangan input ini, data yang dimasukkan akan mempengaruhi hasil yang ditampilkan. Adapun perancangan-perancangan input yang ada dalam perancangan ini adalah:

1. Rancangan Manage Transaksi

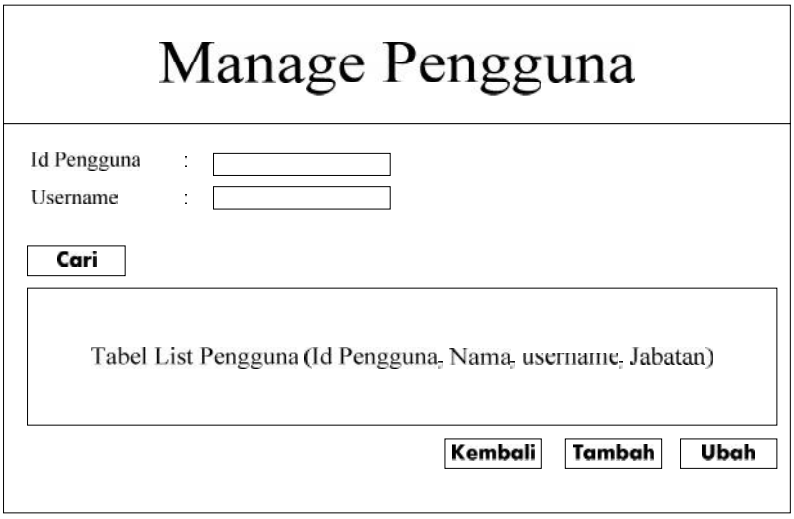
   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian dan pencetakan laporan data transaksi.



Gambar 4. 29 Rancangan Manage Transaksi

2. Rancangan Manage Pengguna

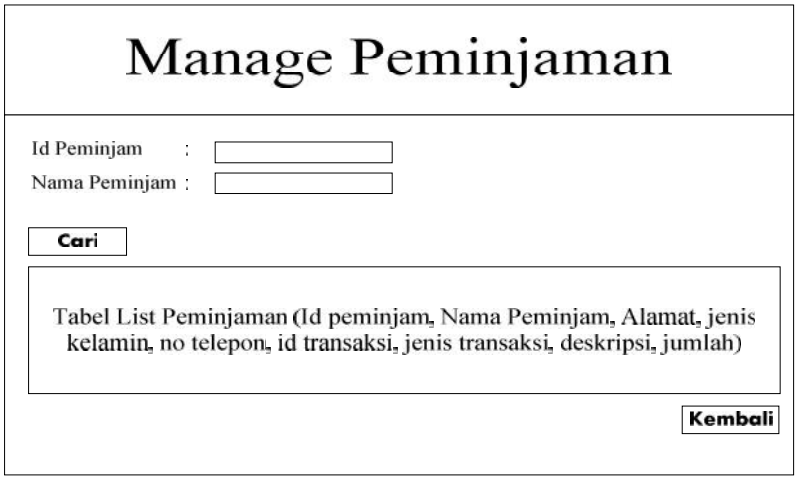
   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian data pengguna.



Gambar 4. 30 Rancangan Maintain data kategori

3. Rancangan Manage Peminjaman

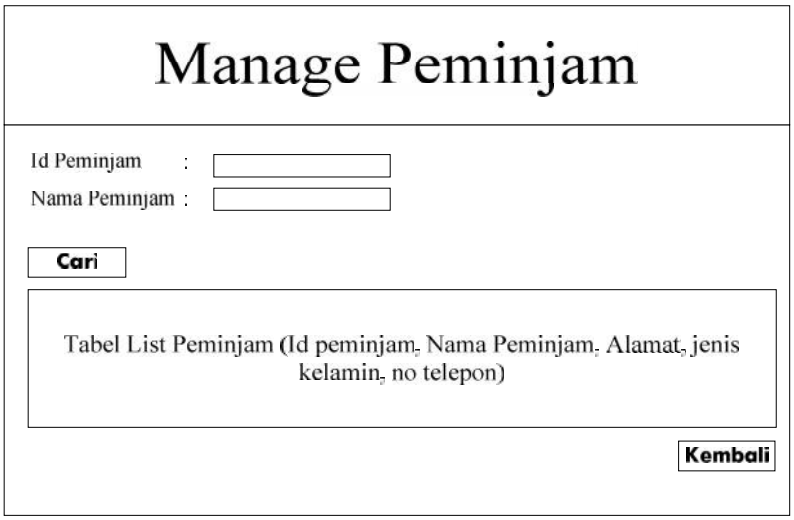
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian data peminjaman.



Gambar 4. 31 Rancangan Manage Peminjaman

4. Rancangan Manage Peminjam

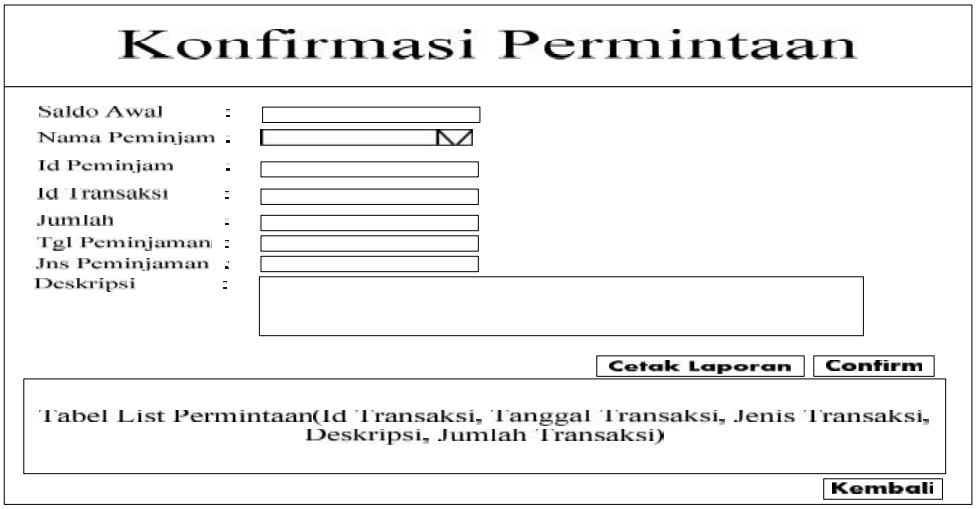
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian data peminjam.



Gambar 4. 32 Rancangan data peminjam

5. Rancangan Konfirmasi Permintaan

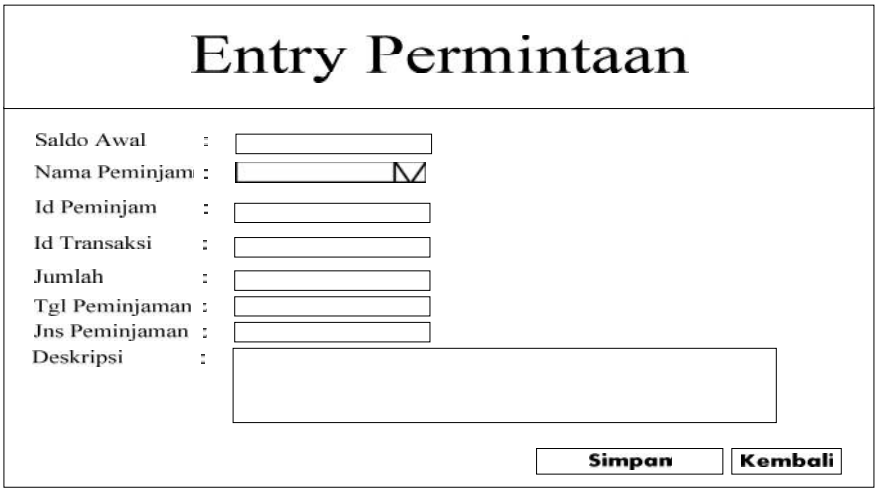
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan konfirmasi permintaan.



Gambar 4. 33 Rancangan Konfirmasi permintaan

6. Rancangan Entry Permintaan

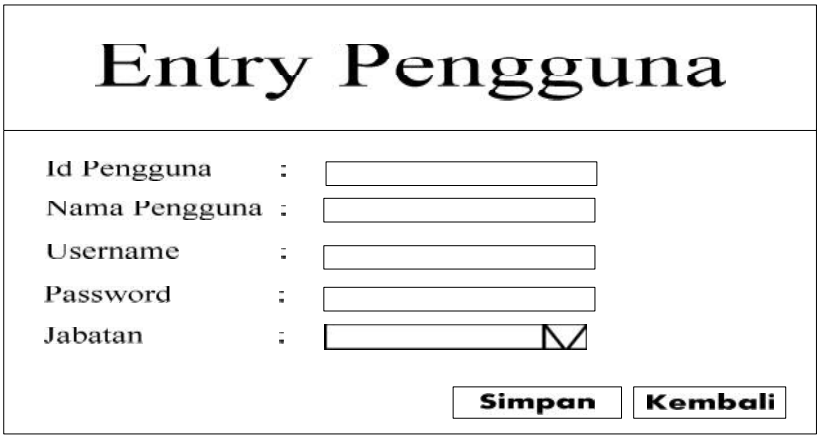
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan penambahan data Permintaan.



Gambar 4. 34 Rancangan Entry Permintaan

7. Rancangan Entry Pengguna

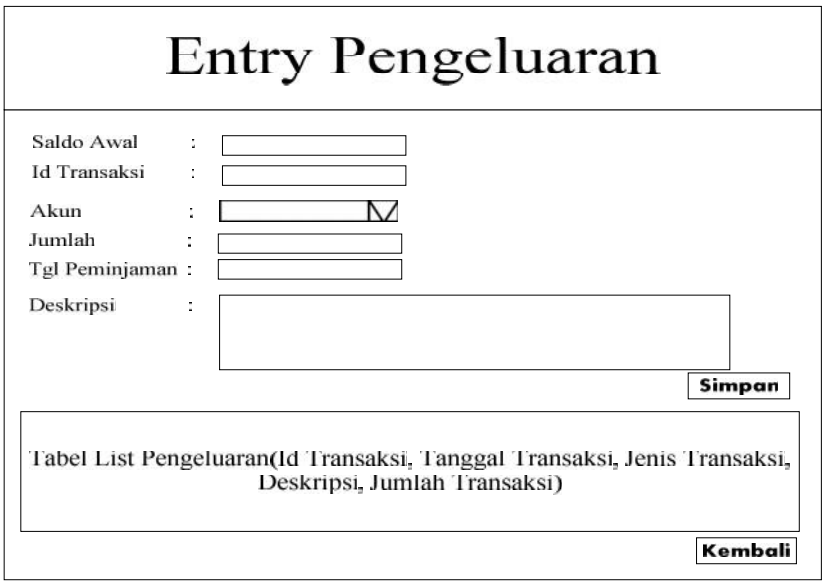
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan penambahan data pengguna.



Gambar 4. 35 Rancangan Entry Pengguna

8. Rancangan Entry Pengeluaran

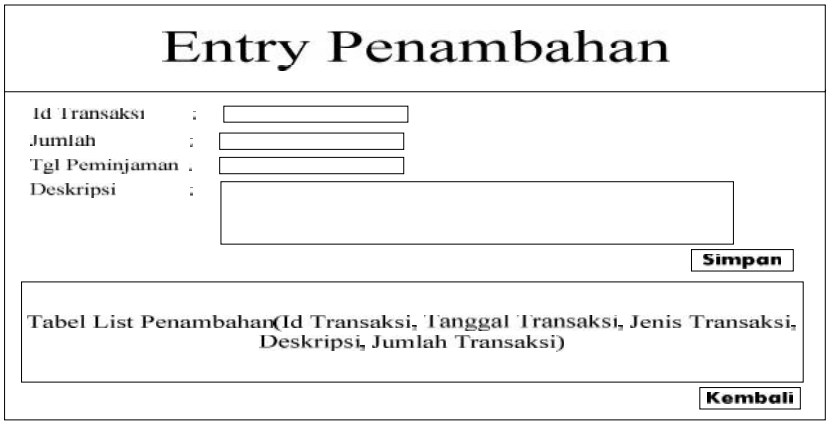
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan penambahan data pengeluaran.



Gambar 4. 36 Rancangan Entry Data Pengeluaran

9. Rancangan Entry Penambahan

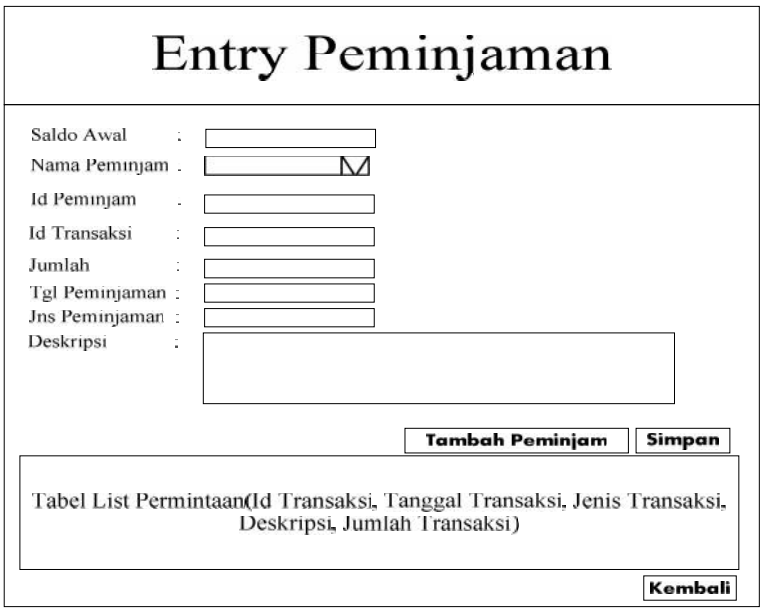
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan Entry data penambahan.



Gambar 4. 37 Rancangan Entry data penambahan

10. Rancangan Entry Peminjaman

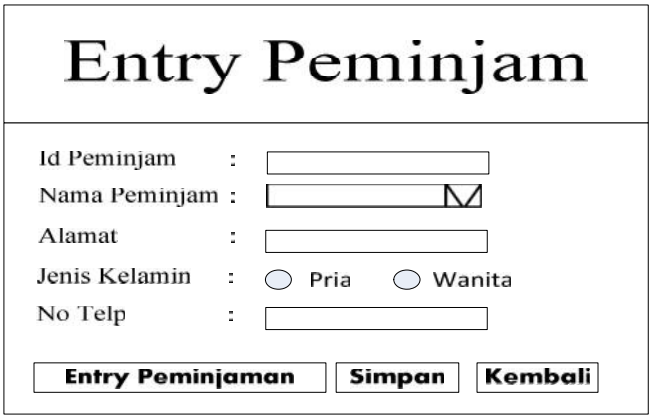
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan Entry data peminjaman.



Gambar 4. 38 Rancangan Entry data peminjaman

11. Rancangan Entry peminjam

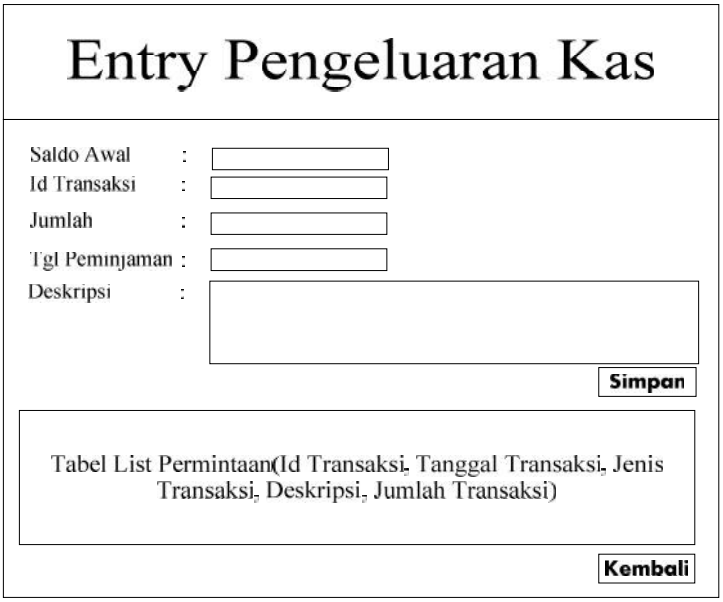
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan Entry data peminjam.



Gambar 4. 39 Rancangan Entry data peminjaman

12. Rancangan Entry Pengeluaran Kas

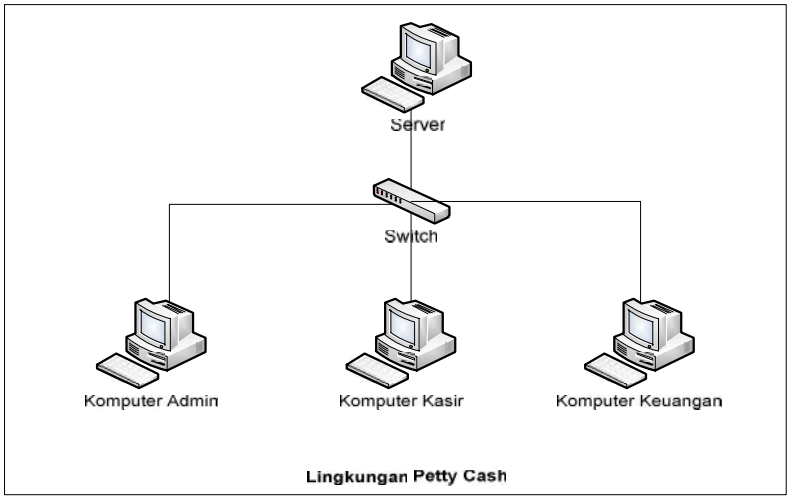
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan Entry data pengeluaran kas.



Gambar 4. 40 Rancangan Entry data peminjaman

4.2.5 Perancangan Arsitektur Jaringan

Arsitektur jaringan yang digunakan adalah sistem *client/server* mempunyai dua komponen utama yaitu komputer *server* dan komputer *client*. *Server* merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer client dan bertindak sebagai *server database* yang menyimpan data. *Client* yaitu komputer yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya.



Gambar 4. 41 Arsitektur jaringan yang diusulkan