BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Perancangan sistem informasi merupakan pengembangan sistem baru dari sistem lama yang ada, dimana masalah-masalah yang terjadi pada sistem lama diharapkan akan teratasi dengan sistem yang baru. Setelah tahap analisis sistem selesai digunakan, maka penulis telah mendapatkan gambaran apa yang harus dikerjakan selanjutnya. Pada perancangan sistem inilah saatnya penulis untuk memikirkan bagaimana membentuk sebuah aplikasi yang baru dengan merancang guna memberikan gambaran umum kepada user secara rinci.

4.1.1 Tujuan Perancangan Sistem

Tahap perancangan ini mempunyai tujuan utama yaitu untuk memenuhi kebutuhan pengguna atau biasa disebut dengan *internal entity*.Yaitu dalam penelitian ini yang disebut *entity* adalah admnistrasi dan manajemen. Sedangkan tujuan lain dari perancangan perangkat lunak ini yaitu untuk memberikan gambaran yang jelas dan hasil rancangan yang lengkap. Demi menghasilkan sebuah penyajian informasi tentang laporan piutang di PT. SAMUDRA APLIKASI INDONESIA.

4.1.2 Gambaran umum sistem yang diusulkan

Setelah mengadakan tahap analisis sistem kemudian penulis mengadakan sebuah perancangan perangkat lunak. Dimana tahap perancangan perangkat lunak laporan piutang ini akan diuraikan melalui sebuah gambaran umum sistem yang di

usulkan agar dapat dimengerti oleh pihak terkait, atau pihak perusahaan sebagai pengguna maupun sebagai informasi.

Tabel 4.1 Arsitektur Perangkat Lunak

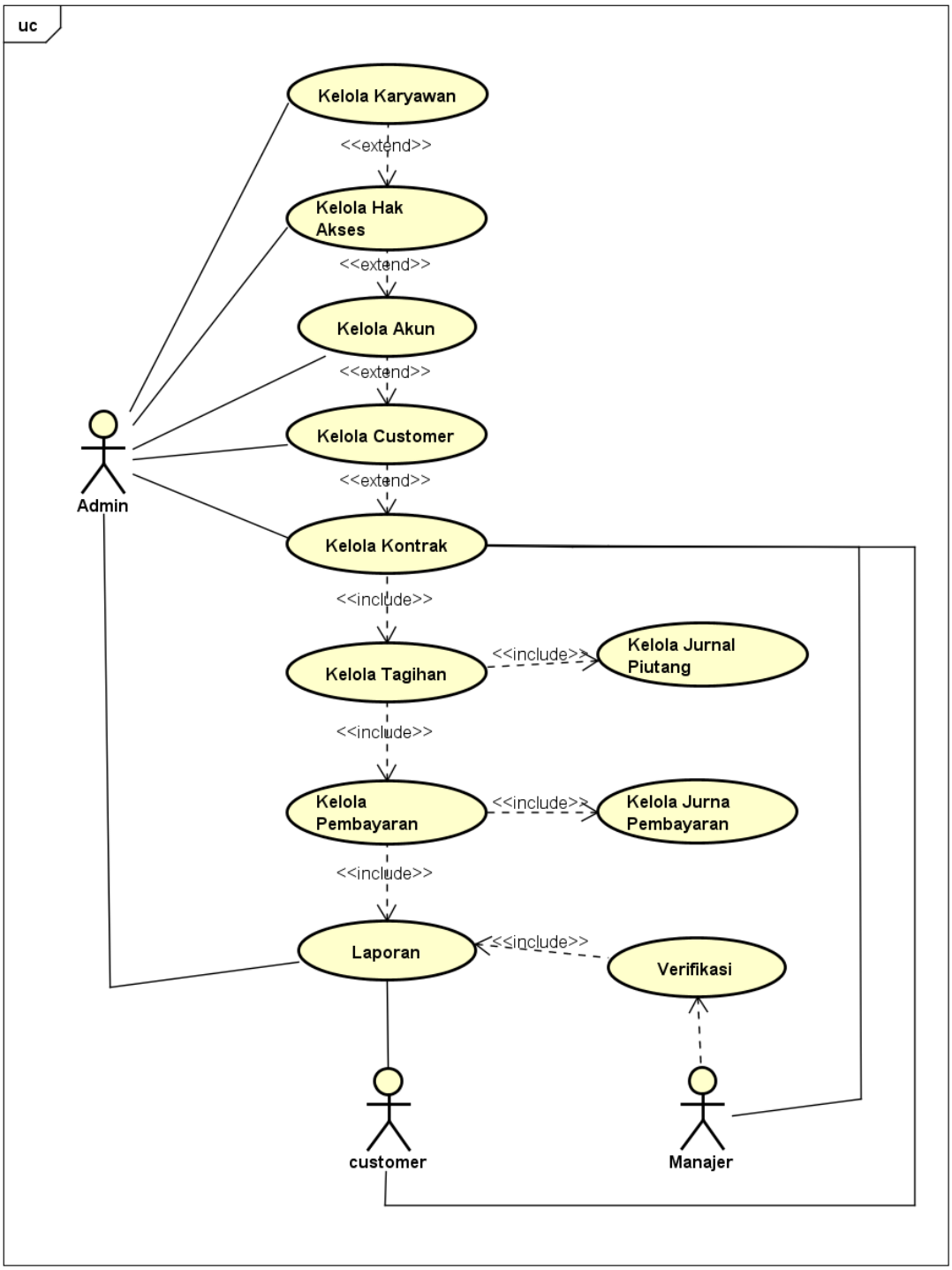
Kategori Pengguna	Tugas	Hak Akses
Administrasi	Melakukan kegiatan langsung kelapangan dan melakukan pencatatan maupun inputan data	1. Mengelola data karyawan 2. mengelola hak akses 3. mengelola data customer 4. mengelola akun 5. mengelola kontrak 6. mengelola tagihan 7. mengelola pembayaran 8. mengelola jurnal piutang 9. mengelola jurnal pembayaran 10. mengelola laporan
Manajemen	Memeriksa semua laporan	1. Melakukan pemeriksaan laporan sesuai dengan kebutuhan.

4.1.3 Perancangan prosedur yang diusulkan

Aplikasi laporan piutang ini merupakan sebuah sistem atau perangkat lunak yang diusulkan guna mempermudah pengolahan data.Dalam perancangan prosedur yang diusulkan ini meliputi *use case diagram*, *skenariouse case*, *activity diagram* dan *sequence diagram*. Yang bertujuan untuk memudahkan dalam pembuatan program dan memudahkan dalam menganalisa alur dokumen.

4.1.3.1 *Use case Diagram*

Use case dalam aplikasi yang akan dibangun mendeskripsikan menu atau informasi dari aplikasi yang nantinya dapat diakses oleh pengguna. Sedangkan pengguna/aktornya adalah manajemen, dan administrasi .



Gambar 4.1 *Use case* yang diusulkan

a. Definisi aktor dan deskripsinya

Tabel 4.2 Definisi Aktor

Aktor	Deskripsi
Administrasi	Pihak yang melakukan proses pengelolaan hak akses (kelola user) untuk menggunakan aplikasi, melakukan proses penginputan data dan perubahan data (data karyawan, hak akses, data akun, data customer, data kontrak, data tagihan, data pembayaran, jurnal piutang, jurnal pembayaran dan laporan), memiliki hak akses penuh dalam penghapusan dan perubahan data.
Manajemen	Pihak yang hanya memeriksa hasil laporan .

b. Definisi *Use case* dan deskripsinya

Tabel 4.3 Definisi *Use case*

Use case	Deskripsi
Kelola User	Use case ini memungkinkan untuk administrasi memberikan hak akses dalam menggunakan aplikasi
Kelola Karyawan	Use case ini memungkinkan untuk melakukan penginputan data karyawan seperti penambahan,

	penghapusan dan pengeditan data karyawan.
Kelola customer	Use case ini memungkinkan untuk melakukan proses pengelolaan data customer seperti penambahan data customer, penghapusan data customer, dan perubahan data costomer.
Kelola Hak Akses	Use case ini memungkinkan untuk administrasi memberikan hak akses dalam menggunakan aplikasi
Kelola Akun	Use case ini memungkinkan untuk melakukan proses pengelolaan data akun, penambahan data akun, penghapusan data akun dan perubahan data akun.
Kelola kontrak	Use case ini memungkinkan untuk melakukan proses pengelolaan data kontrak, penambahan data kontrak, penghapusan data kontrak dan perubahan data kontrak.
Kelola Tagihan	Use case ini memungkinkan untuk melakukan proses pengelolaan data tagihan, penambahan data tagihan, penghapusan data tagihan dan perubahan data tagihan.
Kelola Pembayaran	Use case ini memungkinkan untuk melakukan proses pengelolaan data pembayaran, penambahan data

	pembayaran, penghapusan data pembayaran dan perubahan data pembayaran.
Kelola Jurnal Piutang	Use case ini memungkinkan untuk melakukan pemeriksaan piutang.
Kelola Jurnal Pembayaran	Use case ini memungkinkan untuk melakukan pemeriksaan pembayaran.
Laporan	Use case ini memungkinkan untuk melakukan pengecekan data, maupun laporan yang dilakukan oleh manajemen sehingga dapat memberi sebuah informasi untuk diambil sebuah keputusannya.
Verifikasi	Use case ini memungkinkan menejer untuk memeriksa laporan

4.1.3.2 *Skenario Use case*

Skenario *use case* mendeskripsikan aktor-aktor yang melakukan prosedur dalam sistem, serta menjelaskan respon yang ditanggapi oleh sistem tersebut terhadap prosedur yang dilakukan oleh aktor. Adapun skenario *use case* pada Aplikasi Administrasi sarana prasarana pendidikan yang diusulkan adalah sebagai berikut

Tabel 4.4 *Skenario Use case* Kelola Hak Akses

Identifikasi	
Nama *Use case*	Hak Akses

Aktor	Administrasi
Tujuan	Untuk memberikan hak akses terhadap karyawan untuk menggunakan aplikasi
Deskripsi	Menampilkan data karyawan untuk, melakukan penambahan data hak akses, perubahaan data hak akses, dan penghapusan data hak akses
Skenario Utama	
Kondisi Awal	-
Aktor	**Sistem**
	1. Sistem menampilkan menu aplikasi.
2. Memilih menu Hak Akses	
	3. Menampilkan data hak akses
4. Memilih tombol tambah hak akses	5. Menampilkan form data hak akses
6. Menginputkan data hak akses dan memilih tombol simpan	7. Melakukan proses pengecekan data jika data yang di input valid akan dilakukan penyimpanan data dan kembali menampilkan data user, dan jika data tidak valid

	maka kembali ke point 6
8. Memilih data hak akses yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data user
10. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan perubahan data dan jika data tidak valid maka kembali ke poin 10, setelah data valid akan menampilkan data user
12. Memilih data hak akses yang akan dihapus dan memilih tombol hapus	13. Memberikan notifikasi hapus jika pilihannya iya maka akan dilakukan penghapusan data dan jika tidak maka penghapusan data akan dibatalkan.
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

Tabel 4.5 *Skenario Use case* Kelola Data Karyawan

Identifikasi	
Nama *Use case*	Kelola Karyawan

Aktor	Administrasi
Tujuan	Untuk mengelola data Karyawan
Deskripsi	Menampilkan data karyawan, melakukan penambahan data karyawan, perubahaan data karyawan, dan penghapusan data karyawan
Skenario Utama	
Kondisi Awal	Tambah data karyawan
Aktor	**Sistem**
	1. Menampilkan menu aplikasi
2. Admin memilih menu karyawan	3. Menampilkan data karyawan
4. Memilih tombol tambah	5. Menampilkan form tambah karyawan
6. Menginput data karyawan dan memilih tombol tambah.	7. Melakukan proses pengecekan data jika data valid maka dilakukan penyimpanan data dan jika tidak valid kembali ke point 6.
8. Memilih data karyawan yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data karyawan

0. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan perubahan data dan jika data tidak valid maka kembali ke poin 10, setelah data valid akan menampilkan data karyawan
12 Memilih data karyawan yang akan dihapus dan memilih tombol hapus	13. Memberikan notifikasi hapus jika pilihannya iya maka akan dilakukan penghapusan data dan jika tidak maka penghapusan data akan dibatalkan.
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

Tabel 4.6 *SkenarioUse case* Kelola Akun

Identifikasi	
Nama *Use case*	Kelola Akun
Aktor	Administrasi
Tujuan	Untuk mengolah data akun
Deskripsi	Menampilkan data pembayaran, melakukan penambahan data akun,

	perubahaan data akun, dan penghapusan data akun
Skenario Utama	
Kondisi awal	Tambah data akun
Aktor	**Sistem**
	1. Menampilkan menu aplikasi
2. Admin memilih menu akun	3. Menampilkan data akun
4. Memilih tombol tambah	5. Menampilkan form tambah akun
6. Menginput data akun dan memilih tombol tambah.	7. Melakukan proses pengecekan data jika data valid maka dilakukan penyimpanan data dan jika tidak valid kembali ke point 6.
8. Memilih data akun yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data akun
10. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan perubahan data dan jika data tidak valid maka kembali poin 10,

	setelah data valid akan menampilkan data akun
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

Tabel 4.7 *Skenario Use case* Kelola Customer

Identifikasi	
Nama *Use case*	Kelola Customer
Aktor	Administrasi
Tujuan	Untuk mengelola data customer
Deskripsi	Menampilkan data customer, melakukan penambahan data customer, perubahaan data customer, dan penghapusan data customer
Skenario Utama	
Kondisi Awal	- Tambah data customer
Aktor	**Sistem**
	1. Menampilkan menu aplikasi
2. Admin memilih menu customer	3. Menampilkan data karyawan
4. Memilih tombol tambah	5. Menampilkan form tambah customer

6. Menginput data customer dan memilih tombol tambah.	7. Melakukan proses pengecekan data jika data valid maka dilakukan penyimpanan data dan jika tidak valid kembali ke point 6.
8. Memilih data customer yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data customer
10. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan perubahan data dan jika data tidak valid maka kembali ke poin 10, setelah data valid akan menampilkan data karyawan
12 Memilih data customer yang akan dihapus dan memilih tombol hapus	13. Memberikan notifikasi hapus jika pilihannya iya maka akan dilakukan penghapusan data dan jika tidak maka

	penghapusan data akan dibatalkan.
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

Tabel 4.8 *Skenario Use case* Kelola Kontrak

Identifikasi	
Nama *Use case*	Kelola Kontrak
Aktor	Administrasi
Tujuan	Untuk mengelola data kontrak
Deskripsi	Menampilkan data kontrak, melakukan penambahan data customer, perubahaan data customer, dan penghapusan data customer
Skenario Utama	
Kondisi awal	Tambah data kontrak
Aktor	**Sistem**
	1. Menampilkan menu aplikasi
2. Admin memilih menu kontrak	3. Menampilkan data kontrak
4. Memilih tombol tambah	5. Menampilkan form tambah kontrak
6. Menginput data kontrak dan	7. Melakukan proses pengecekan

memilih tombol tambah.	data jika data valid maka dilakukan penyimpanan data dan jika tidak valid kembali ke point 6.
8. Memilih data kontrak yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data kontrak
10. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan perubahan data dan jika data tidak valid maka kembali ke poin 10, setelah data valid akan menampilkan data kontrak
12 Memilih data kontrak yang akan dihapus dan memilih tombol hapus	13. Memberikan notifikasi hapus jika pilihannya iya maka akan dilakukan penghapusan data dan jika tidak maka penghapusan data akan dibatalkan.
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

Tabel 4.9 *SkenarioUse case* Kelola Tagihan

Identifikasi	
Nama *Use case*	Kelola tagihan
Aktor	Administrasi
Tujuan	Untuk mengolah data tagihan
Deskripsi	Menampilkan data tagihanan, melakukan penambahan data tagihanan, perubahaan data tagihanan, dan penghapusan data tagihanan
Skenario Utama	
Kondisi awal	Tambah data tagihan
Aktor	**Sistem**
	1. Menampilkan menu aplikasi
2. Admin memilih menu tagihan	3. Menampilkan data tagihan
4. Memilih tombol tambah	5. Menampilkan form tambah tagihan
6. Menginput data tagihan dan memilih tombol tambah.	7. Melakukan proses pengecekan data jika data valid maka dilakukan penyimpanan data dan

	jika tidak valid kembali ke point 6.
8. Memilih data tagihan yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data tagihan
10. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan perubahan data dan jika data tidak valid maka kembali poin 10, setelah data valid akan menampilkan data tagihan
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

Tabel 4.10 *SkenarioUse case* Pembayaran

Identifikasi	
Nama *Use case*	Kelola Pembayaran
Aktor	Administrasi
Tujuan	Untuk mengolah data pembayaran
Deskripsi	Menampilkan data pembayaran, melakukan penambahan data

	pembayaran, perubahaan data pembayaran, dan penghapusan data pembayaran
Skenario Utama	
Kondisi awal	Tambah data pembayaran
Aktor	**Sistem**
	1. Menampilkan menu aplikasi
2. Admin memilih menu pembayaran	3. Menampilkan data pembayaran
4. Memilih tombol tambah	5. Menampilkan form tambah pembayaran
6. Menginput data pembayaran dan memilih tombol tambah.	7. Melakukan proses pengecekan data jika data valid maka dilakukan penyimpanan data dan jika tidak valid kembali ke point 6.
8. Memilih data pembayaran yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data pembayaran
10. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan

	perubahan data dan jika data tidak valid maka kembali poin 10, setelah data valid akan menampilkan data pembayaran
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

Tabel 4.11 *SkenarioUse case* Kelola Jurnal Piutang

Identifikasi	
Nama *Use case*	Kelola Jurnal Piutang
Aktor	Administrasi
Tujuan	Untuk mengolah jurnal piutang
Deskripsi	Menampilkan jurnal piutang, melakukan penambahan data jurnal piutang, perubahaan data jurnal piutang, dan penghapusan data jurnal piutang
Skenario Utama	
Kondisi awal	Tambah data jurnal piutang
Aktor	**Sistem**
	1. Menampilkan menu aplikasi
2. Admin memilih menu jurnal	3. Menampilkan data jurnal piutang

piutang	
4. Memilih tombol tambah	5. Menampilkan form tambah jurnal piutang
6. Menginput data jurnal piutang dan memilih tombol tambah.	7. Melakukan proses pengecekan data jika data valid maka dilakukan penyimpanan data dan jika tidak valid kembali ke point 6.
8. Memilih data jurnal piutang yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data jurnal piutang
10. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan perubahan data dan jika data tidak valid maka kembali poin 10, setelah data valid akan menampilkan data jurnal piutang
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

Tabel 4.12 *SkenarioUse case* Kelola Jurnal Pembayaran

Identifikasi	
Nama *Use case*	Kelola jurnal pembayaran
Aktor	Administrasi
Tujuan	Untuk mengolah data jurnal pembayaran
Deskripsi	Menampilkan data pembayaran, melakukan penambahan data jurnal pembayaran, perubahaan data jurnal pembayaran, dan penghapusan data jurnal pembayaran
Skenario Utama	
Kondisi awal	Tambah data jurnal pembayaran
Aktor	**Sistem**
	1. Menampilkan menu aplikasi
2. Admin memilih menu jurnal pembayaran	3. Menampilkan data jurnal pembayaran
4. Memilih tombol tambah	5. Menampilkan form tambah jurnal pembayaran
6. Menginput data jurnal pembayaran dan memilih tombol tambah.	7. Melakukan proses pengecekan data jika data valid maka dilakukan penyimpanan data dan

	jika tidak valid kembali ke point 6.
8. Memilih data jurnal pembayaran yang akan dilakukan perubahan dan memilih tombol ubah	9. Melakukan pengecekan data yang akan di rubah dan menampilkan form ubah data jurnal pembayaran
10. Menginputkan data yang dirubah didalam form ubah data dan memilih tombol ubah	11. Melakukan proses pengecekan data, jika data yang diinputkan valid maka akan dilakukan perubahan data dan jika data tidak valid maka kembali poin 10, setelah data valid akan menampilkan data jurnal pembayaran
Kondisi Akhir	Penyimpanan data , perubahan data, dan penghapusan data.

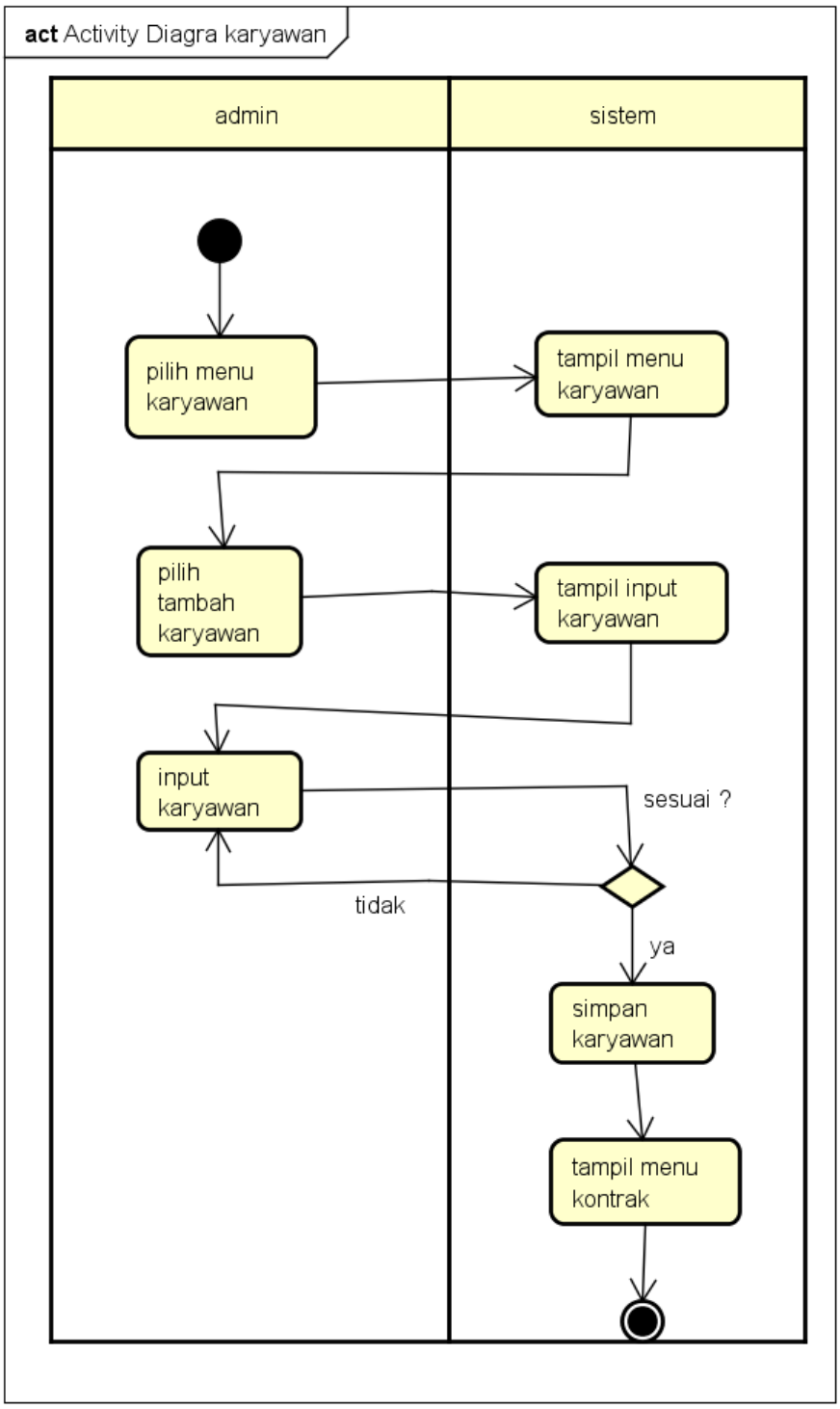
Tabel 4.13 *SkenarioUse case* Laporan

Identifikasi	
Nama *Use case*	Laporan
Aktor	Administrasi
Tujuan	Untuk mendapatkan laporan dari setiap proses olah data yang dilakukan

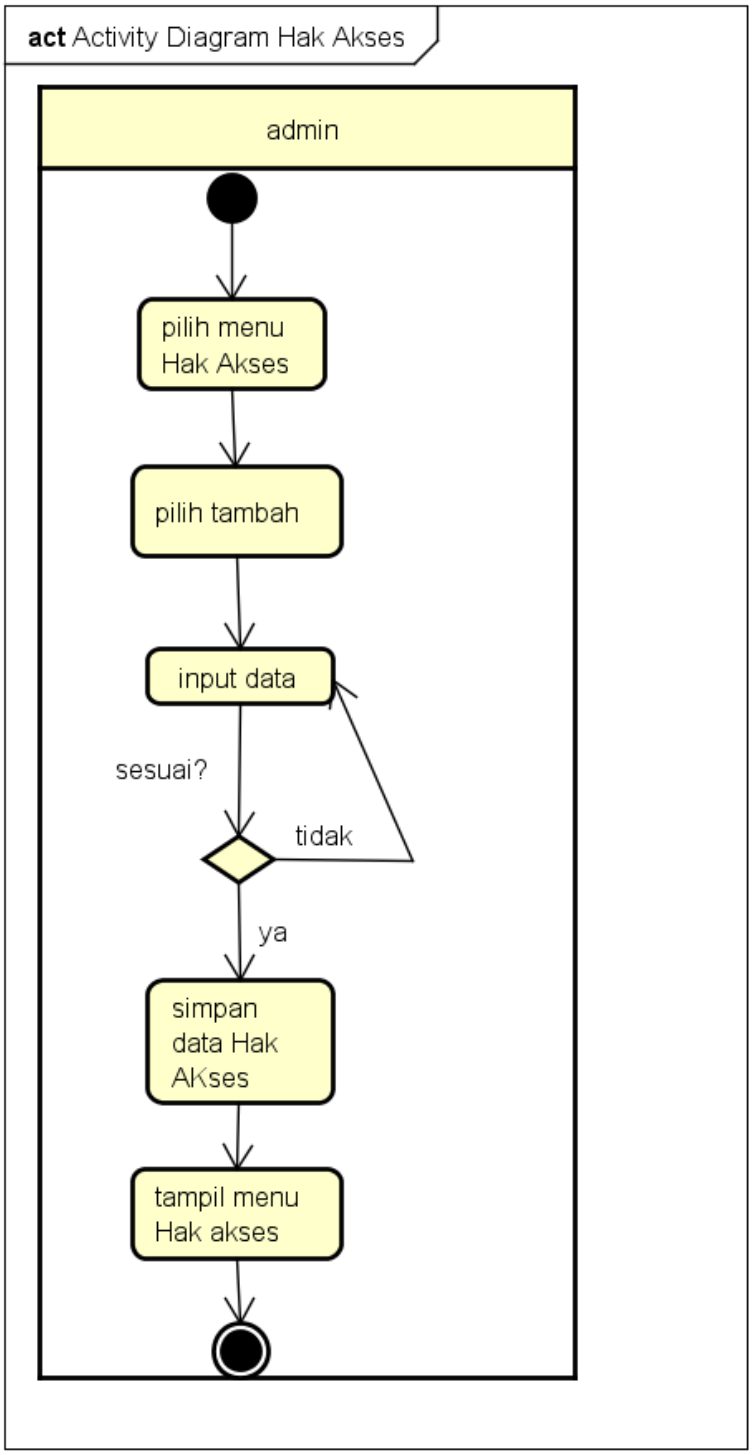
Deskripsi	Menampilkan menu laporan yang bertujuan untuk mencetak laporan dari setiap olah data yang dilakukan
Skenario Utama	
Kondisi Awal	Laporan belum di cetak
Aktor	**Sistem**
1. manajer memilih menu laporan.	
	2. Menampilkan menu laporan
3. manajer memilih data laporan yang akan dicetak	
	4. Menampilkan data laporan yang akan di cetak
	5. Sistem akan mencetak data laporan
Kondisi Akhir	Laporan telah di cetak

4.1.3.3 *Activity Diagram*

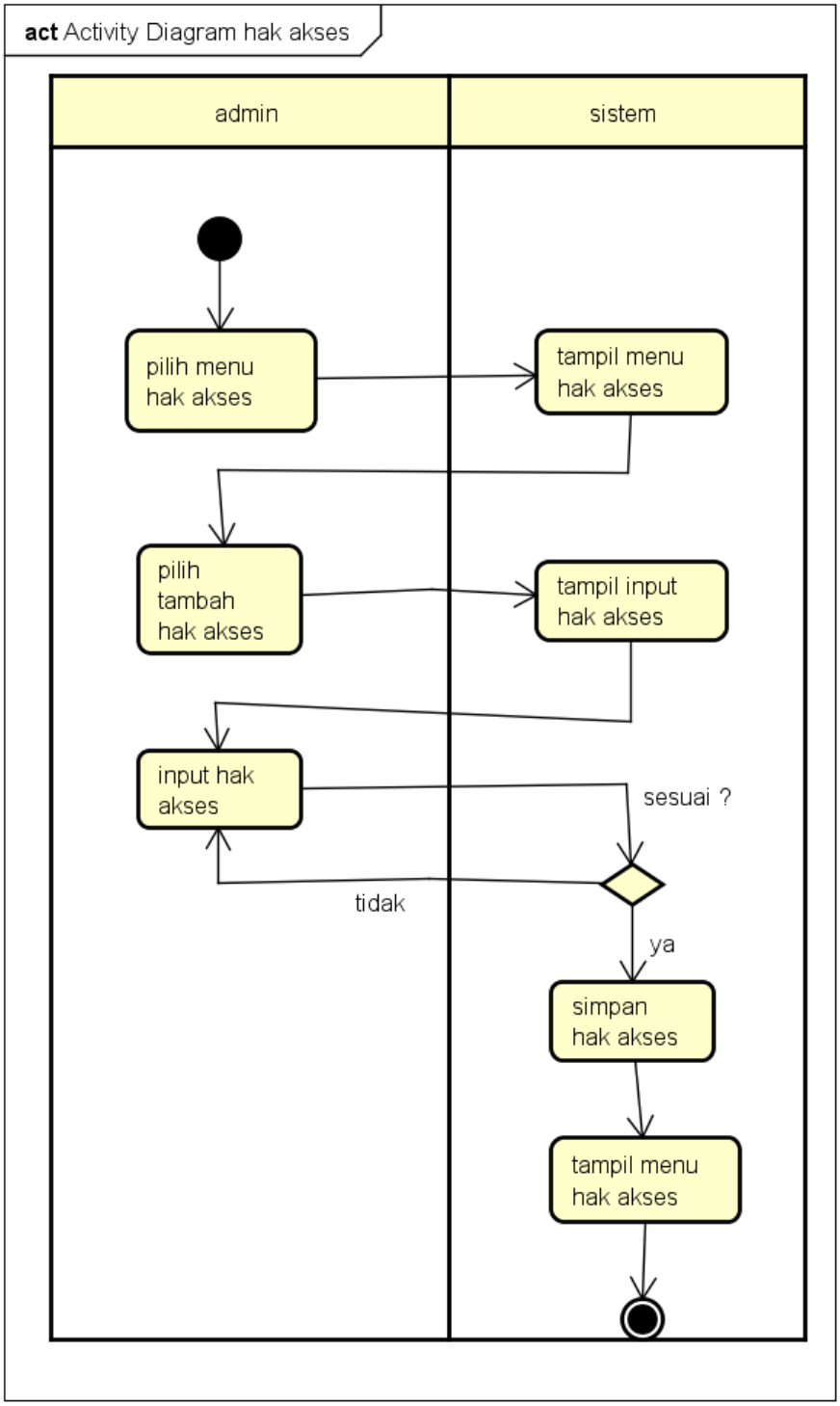
Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case*.



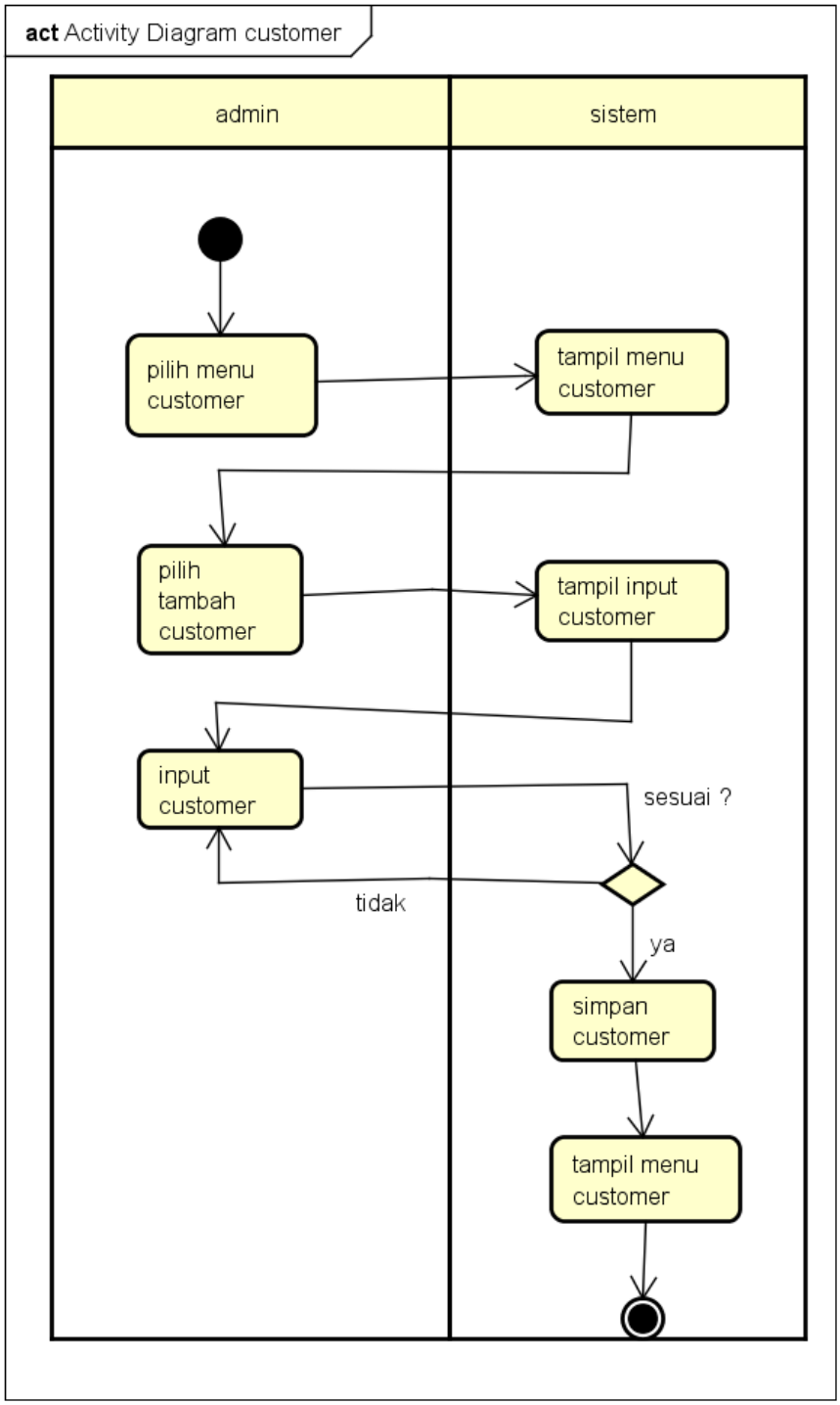
Gambar 4.2 *Activity Diagram* Kelola Karyawan


act Activity Diagram Hak Akses
admin
pilih menu Hak Akses
pilih tambah
input data
sesuai?
tidak
ya
simpan data Hak AKses
tampil menu Hak akses

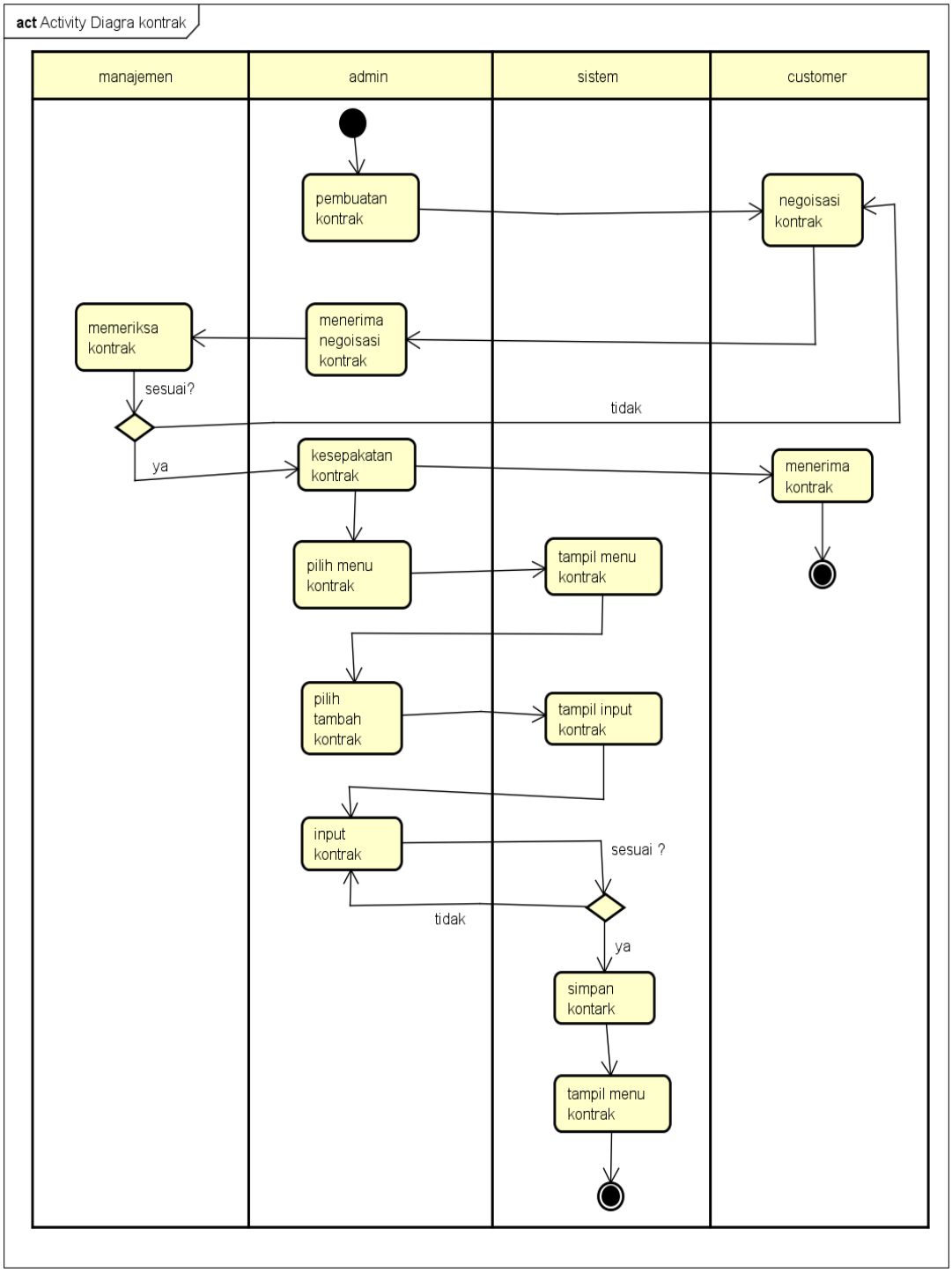
Gambar 4.3 *Activity Diagram* Kelola Hak Akses



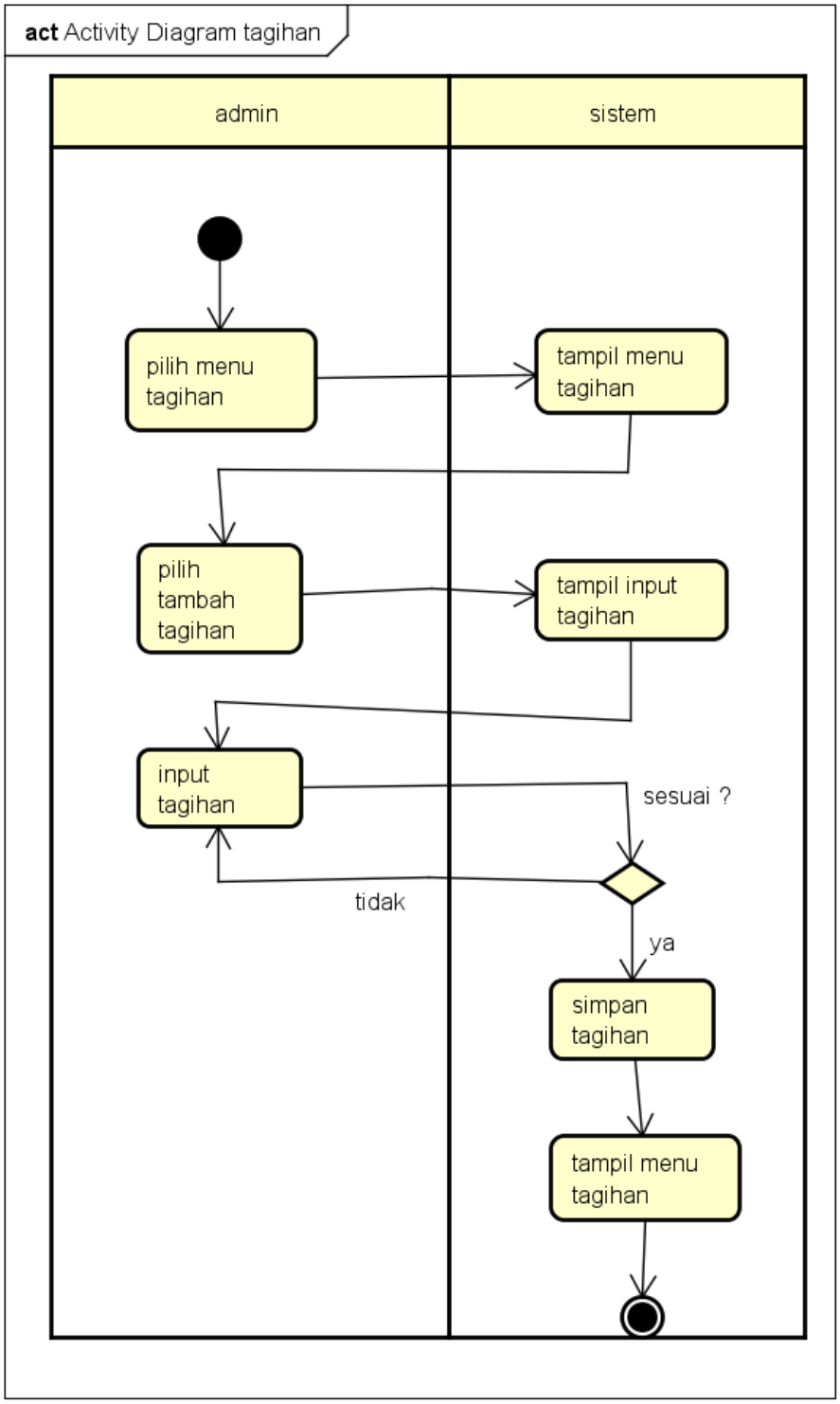
Gambar 4.4 *Activity Diagram* Kelola Akun



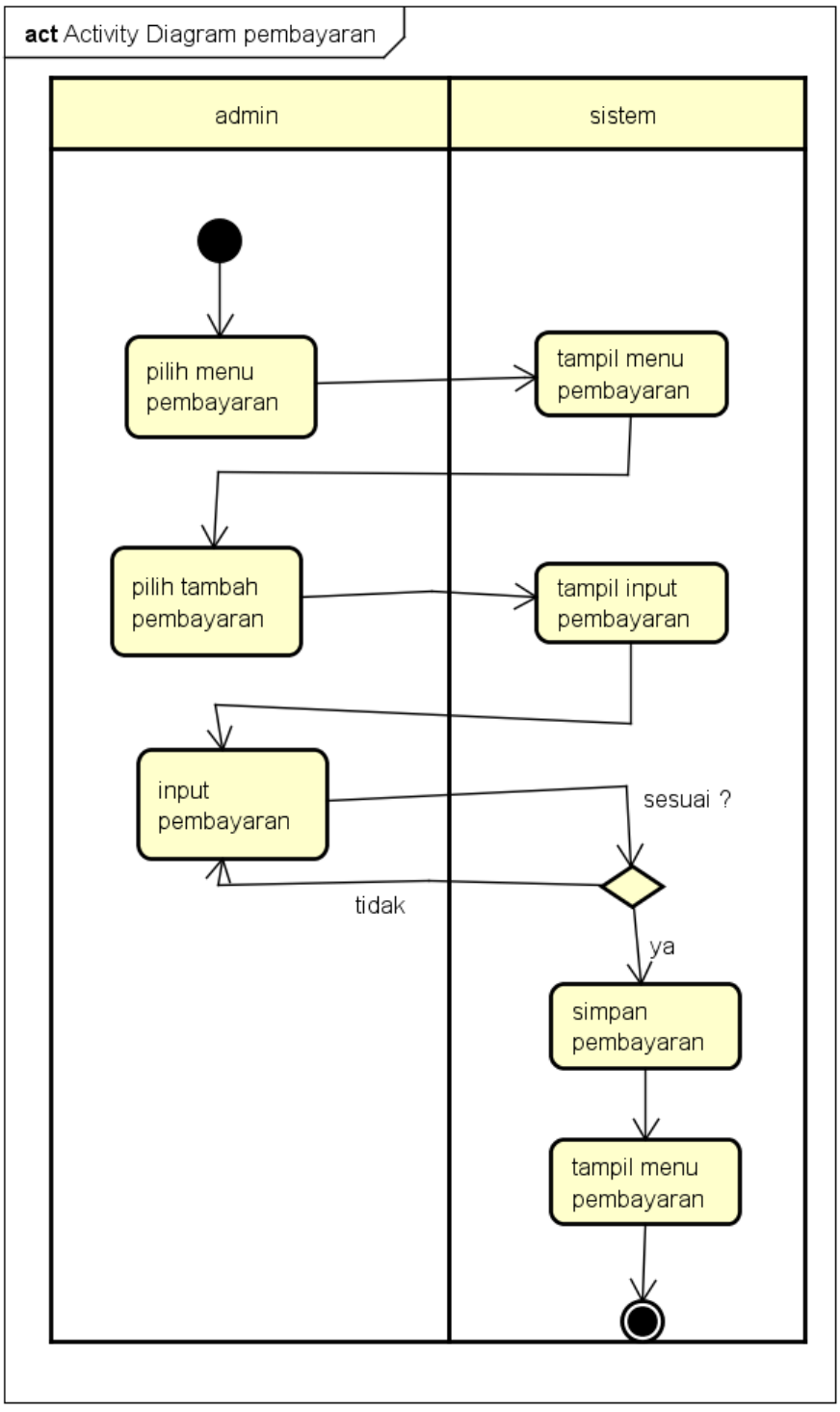
Gambar 4.5 ***Activity Diagram*** **Kelola Customer**



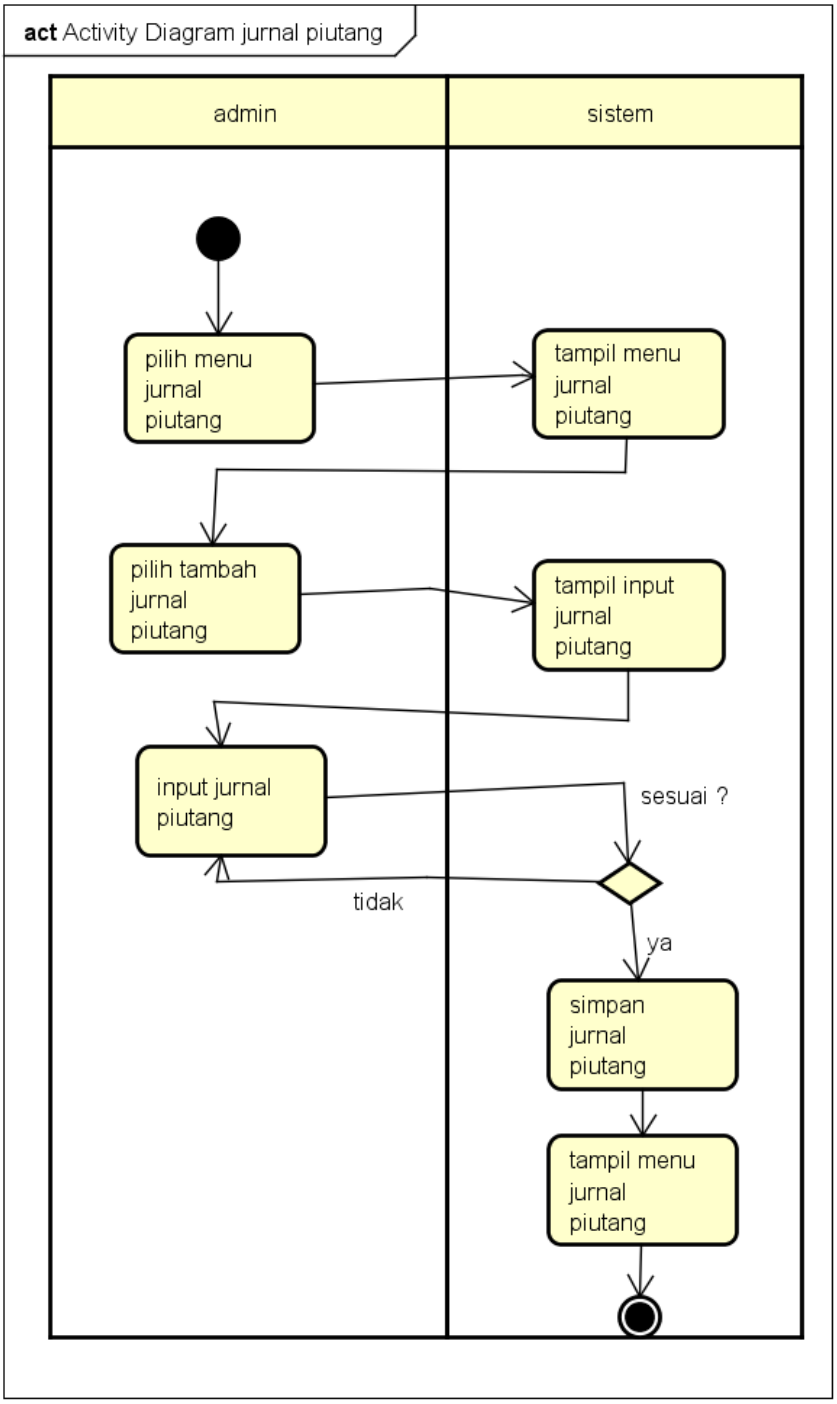
Gambar 4.6 *Activity Diagram* Kelola Kontrak



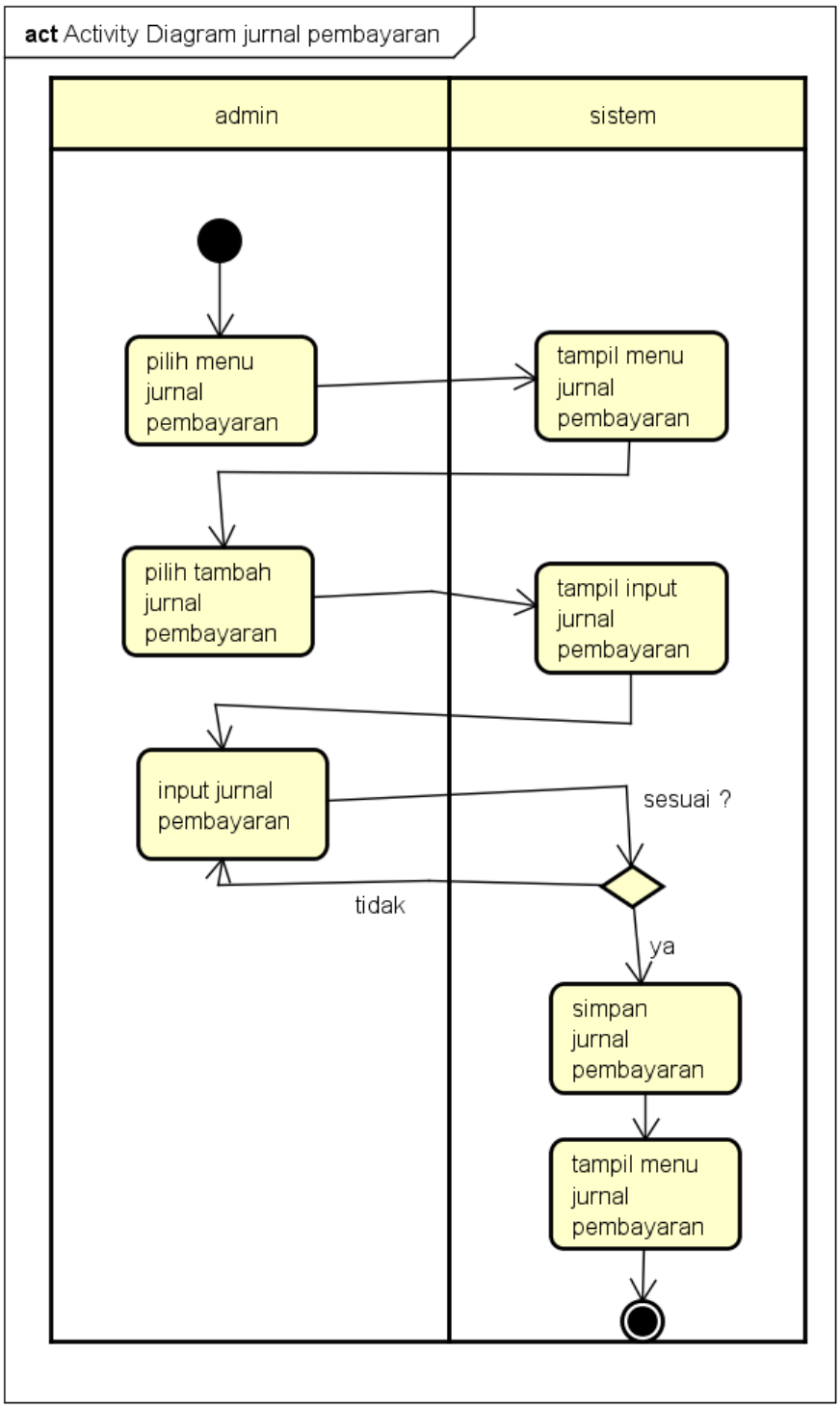
Gambar 4.7 ***Activity Diagram*** **Kelola Tagihan**



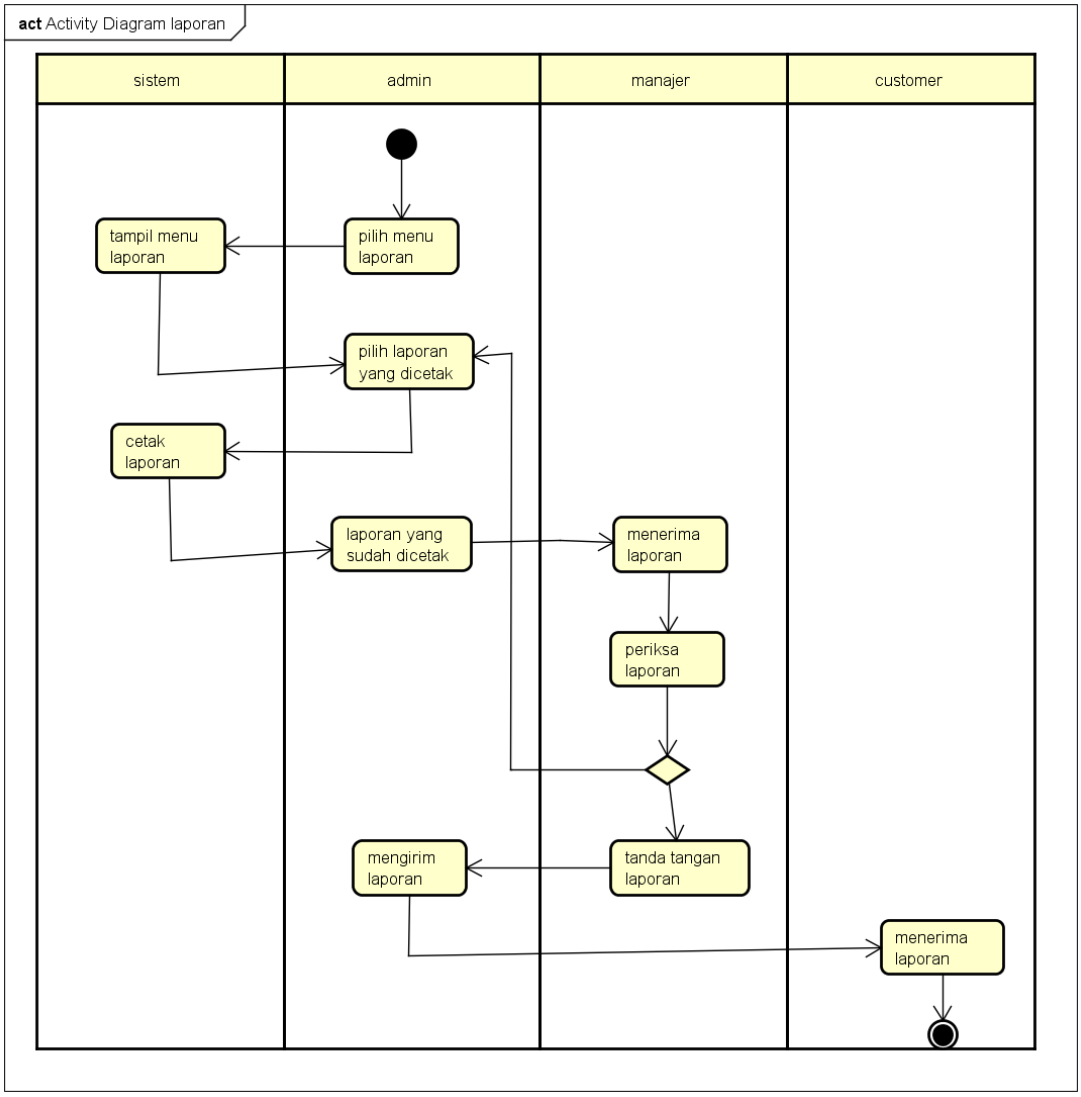
Gambar 4.8 *Activity Diagram* Kelola pembayaran



Gambar 4.9 *Activity Diagram* Kelola Jurnal Piutang



Gambar 4.10 *Activity Diagram* Kelola Jurnal Pembayaran

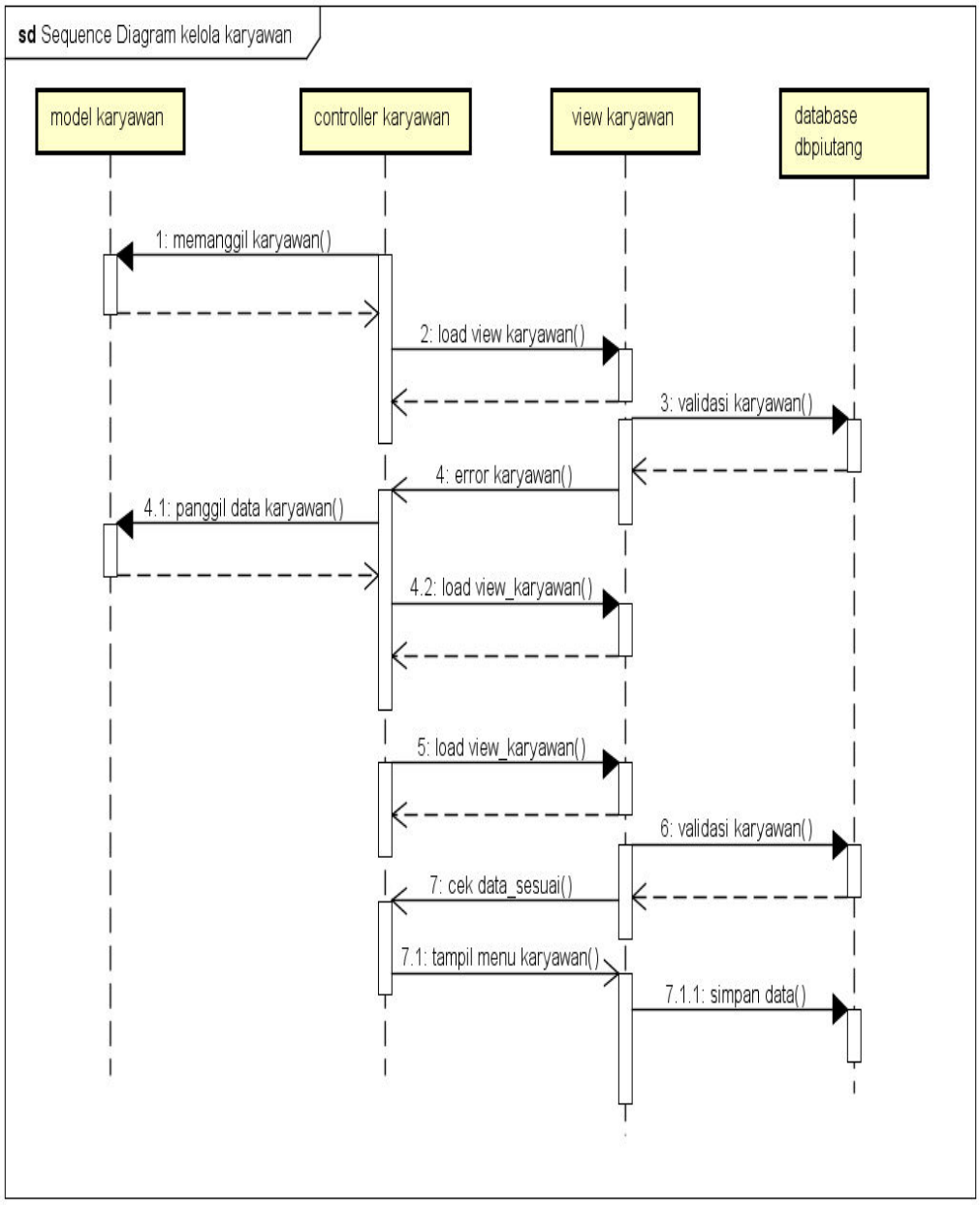


Gambar 4.11 ***Activity Diagram*** **Laporan**

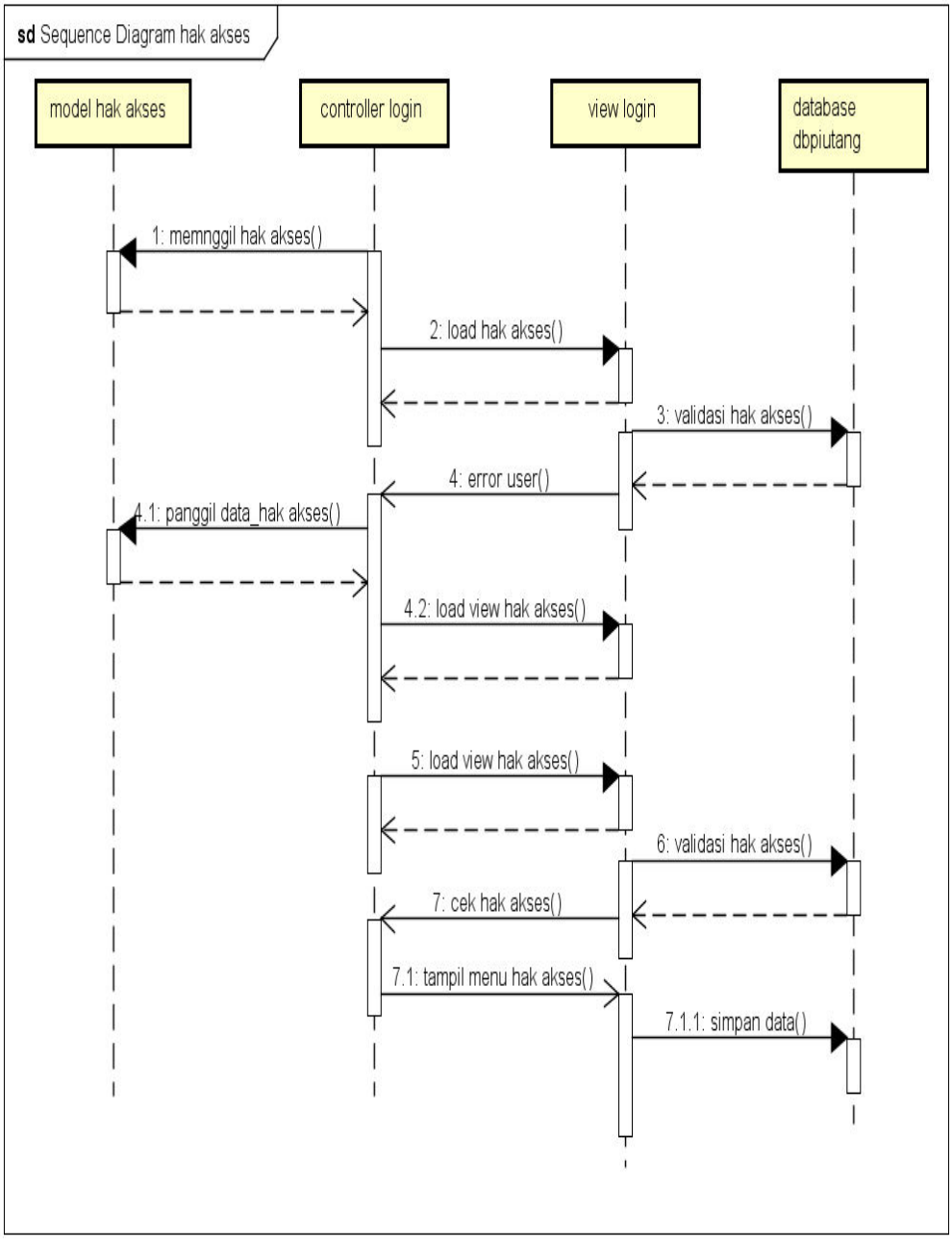
4.1.3.4 *Sequence Diagram*

Sequence diagram merupakan interaksi antara objek-objek dalam sistem ini dan terjadi komunikasi yang berupa pesan serta parameter waktu.*Sequencediagram*

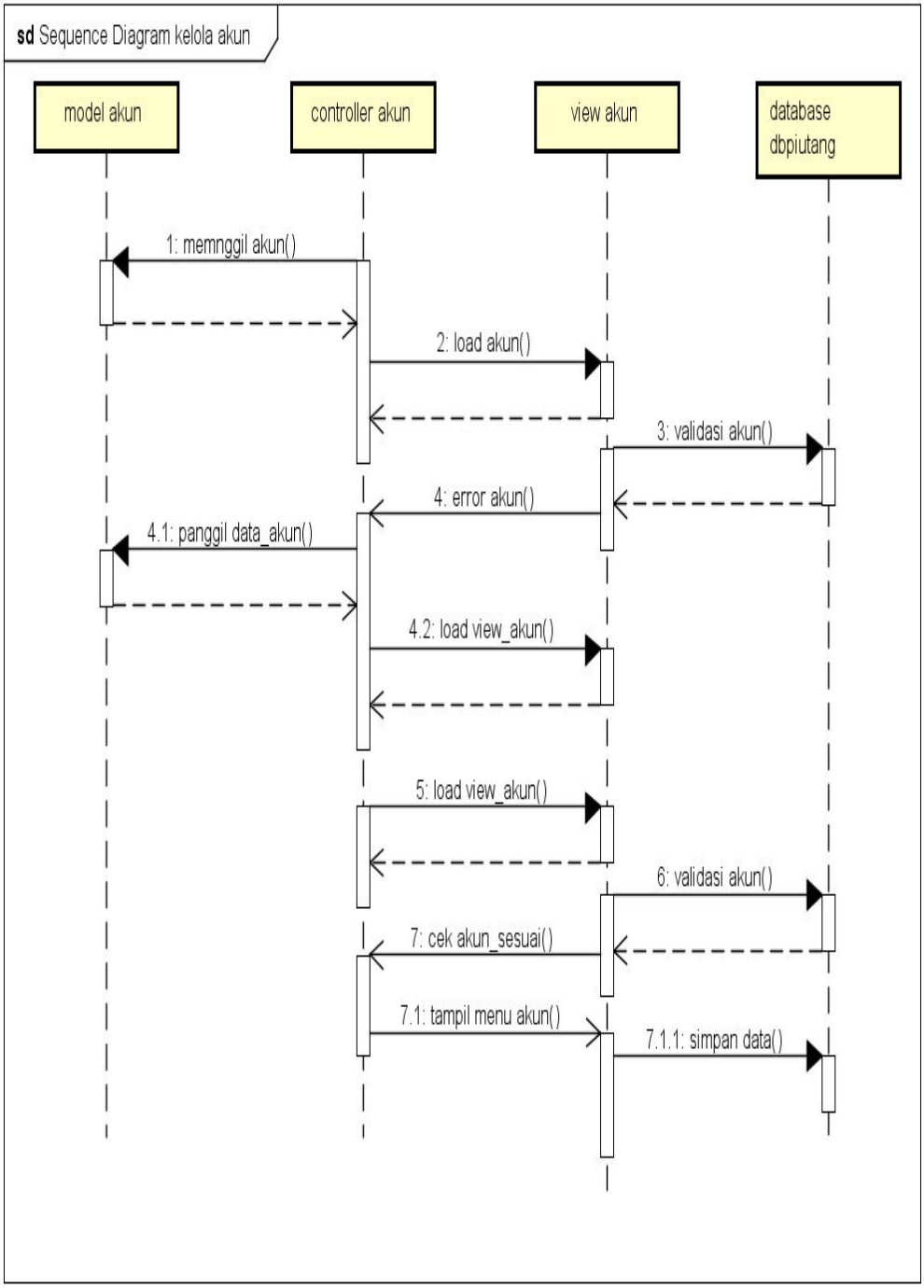
memiliki proses yang sama dengan *usecase*, perbedaannya dalam *sequence diagram* terdapat *time line* pada setiap prosesnya.



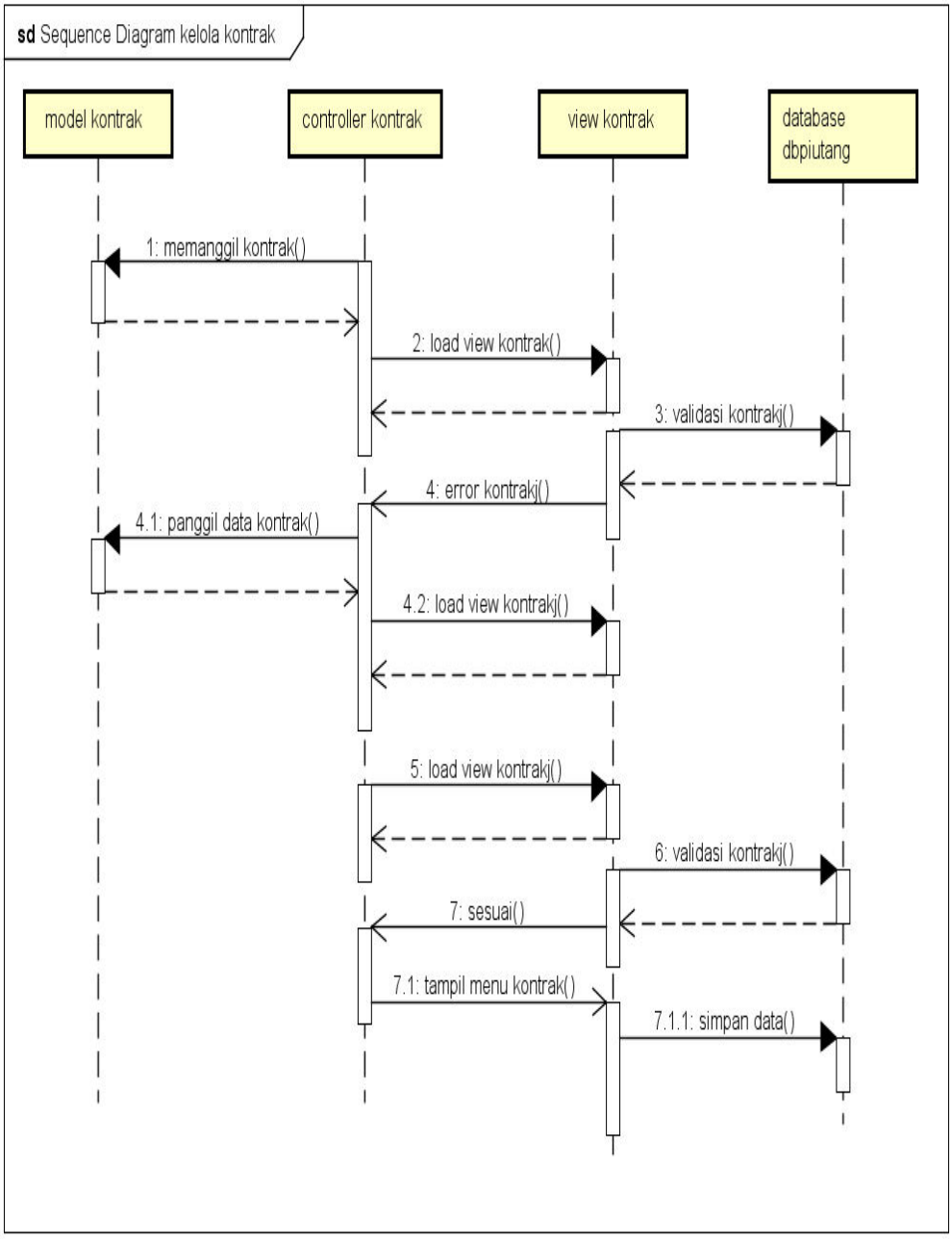
Gambar 4.12 *Sequence Diagram* Kelola Karyawan yang diusulkan



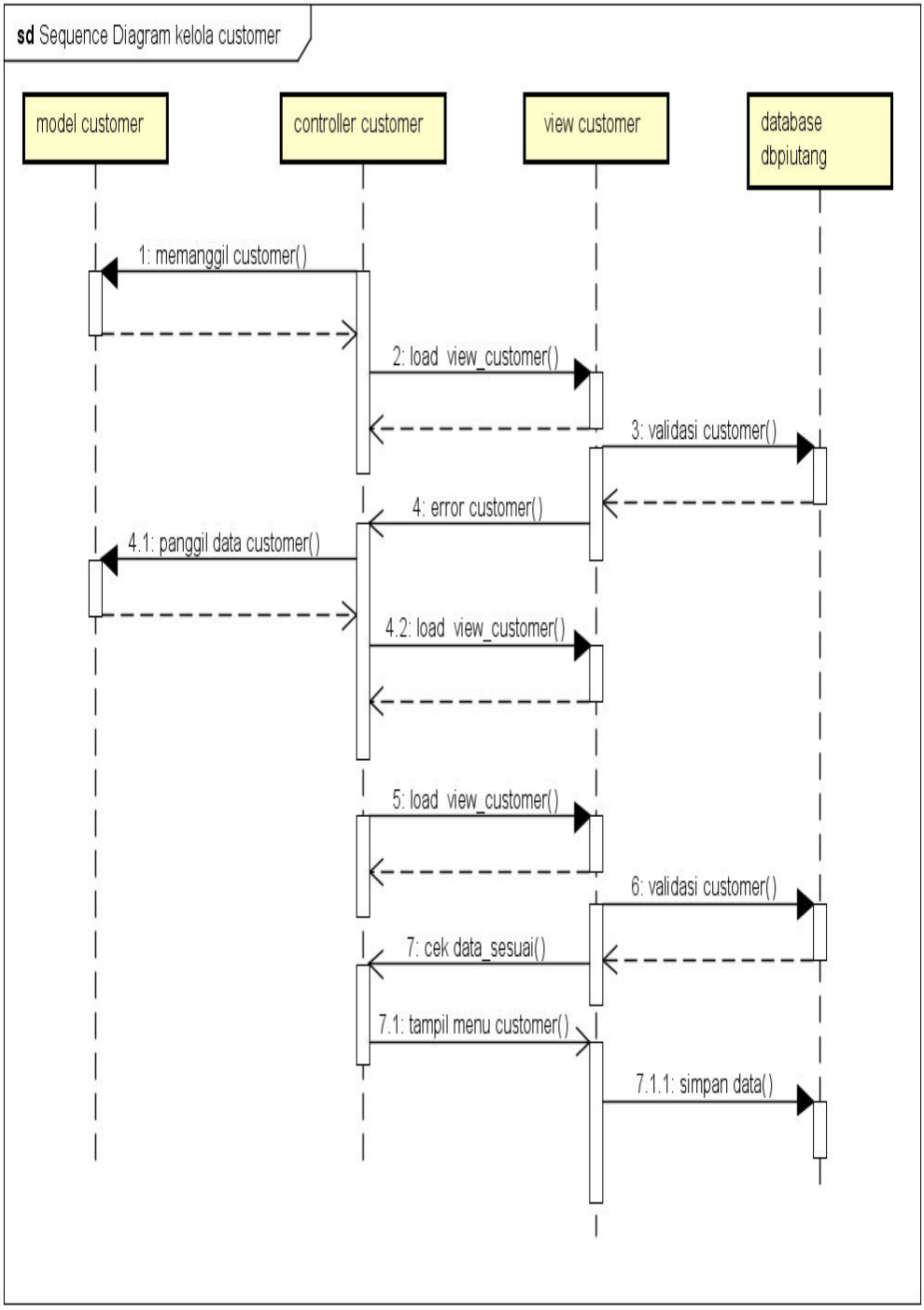
Gambar 4.13 *Sequence Diagram* Kelola Hak Akses yang diusulkan



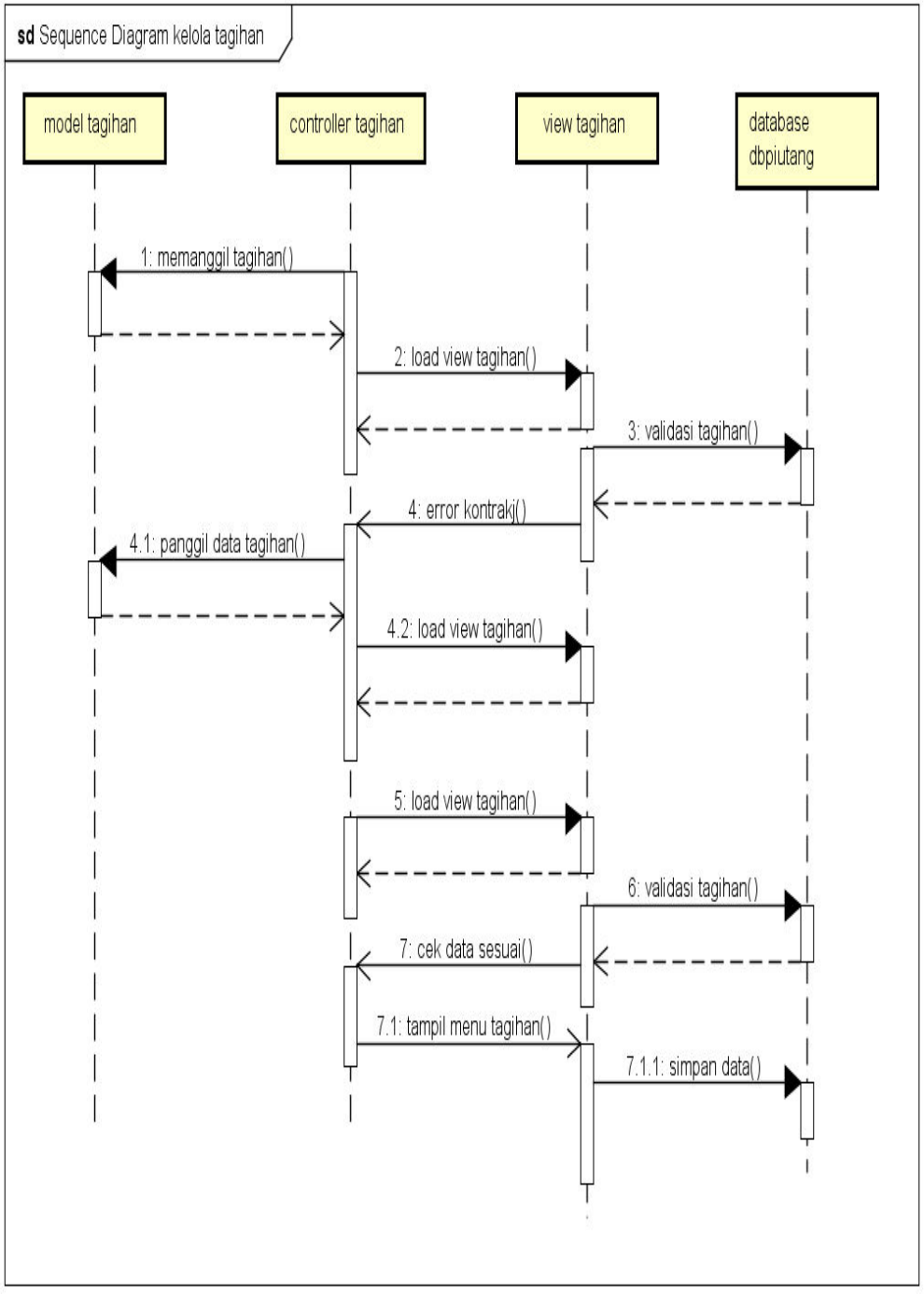
Gambar 4.14 *Sequence Diagram* Kelola Akun yang diusulkan



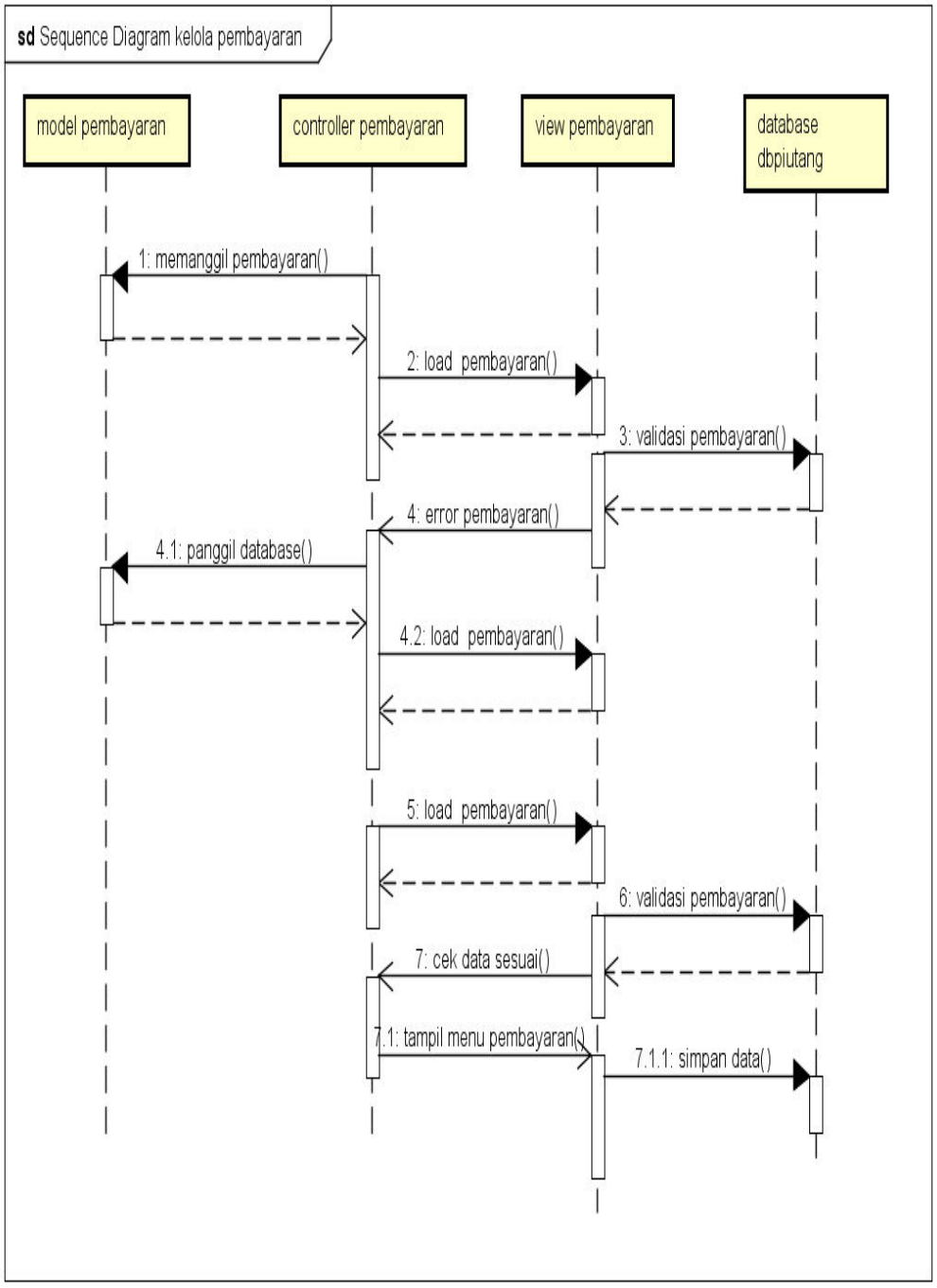
Gambar 4.15 ***Sequence Diagram*** **Kelola Kontrak yang diusulkan**



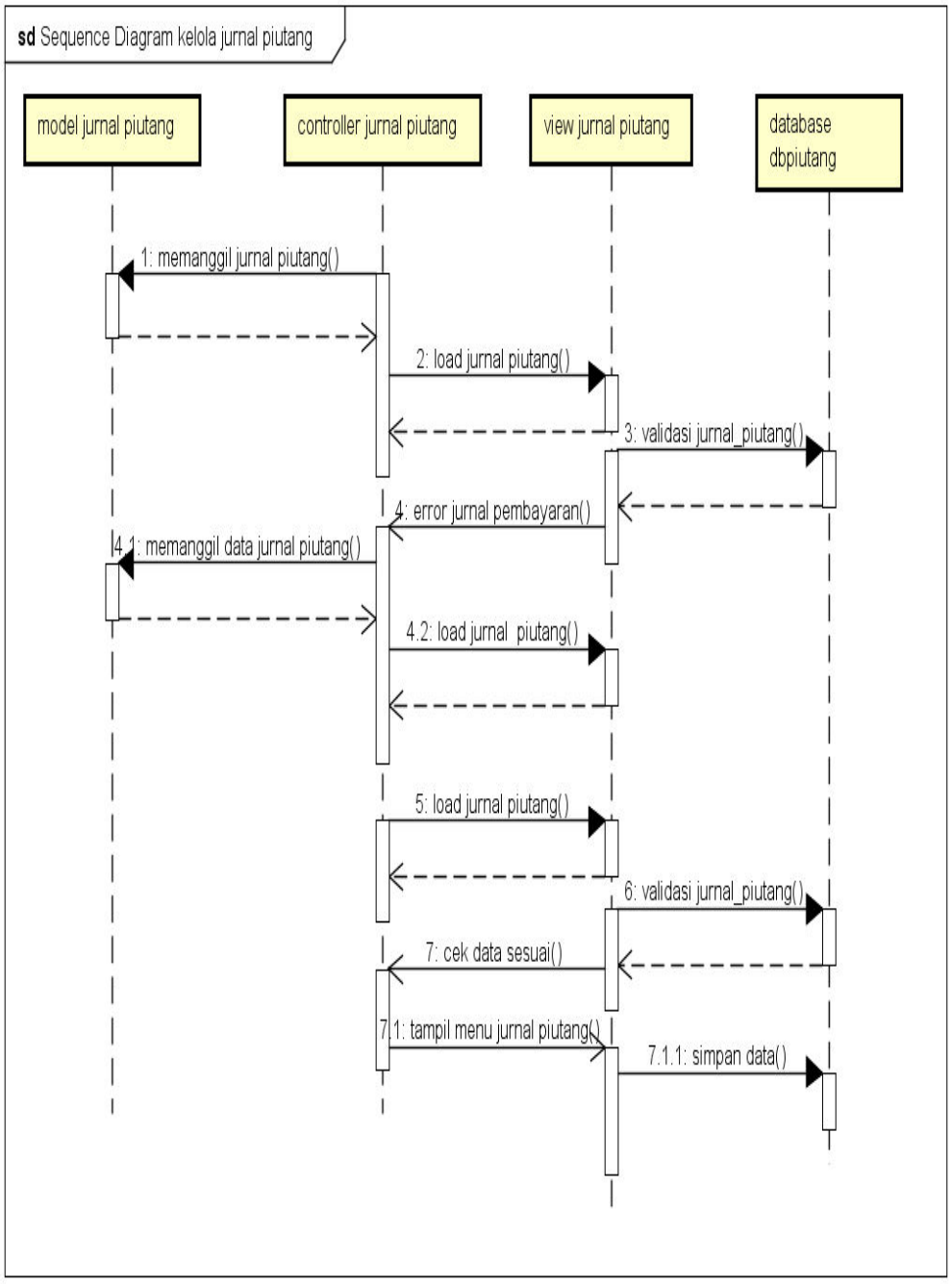
Gambar 4.16 *Sequence Diagram* Kelola Customer yang diusulkan



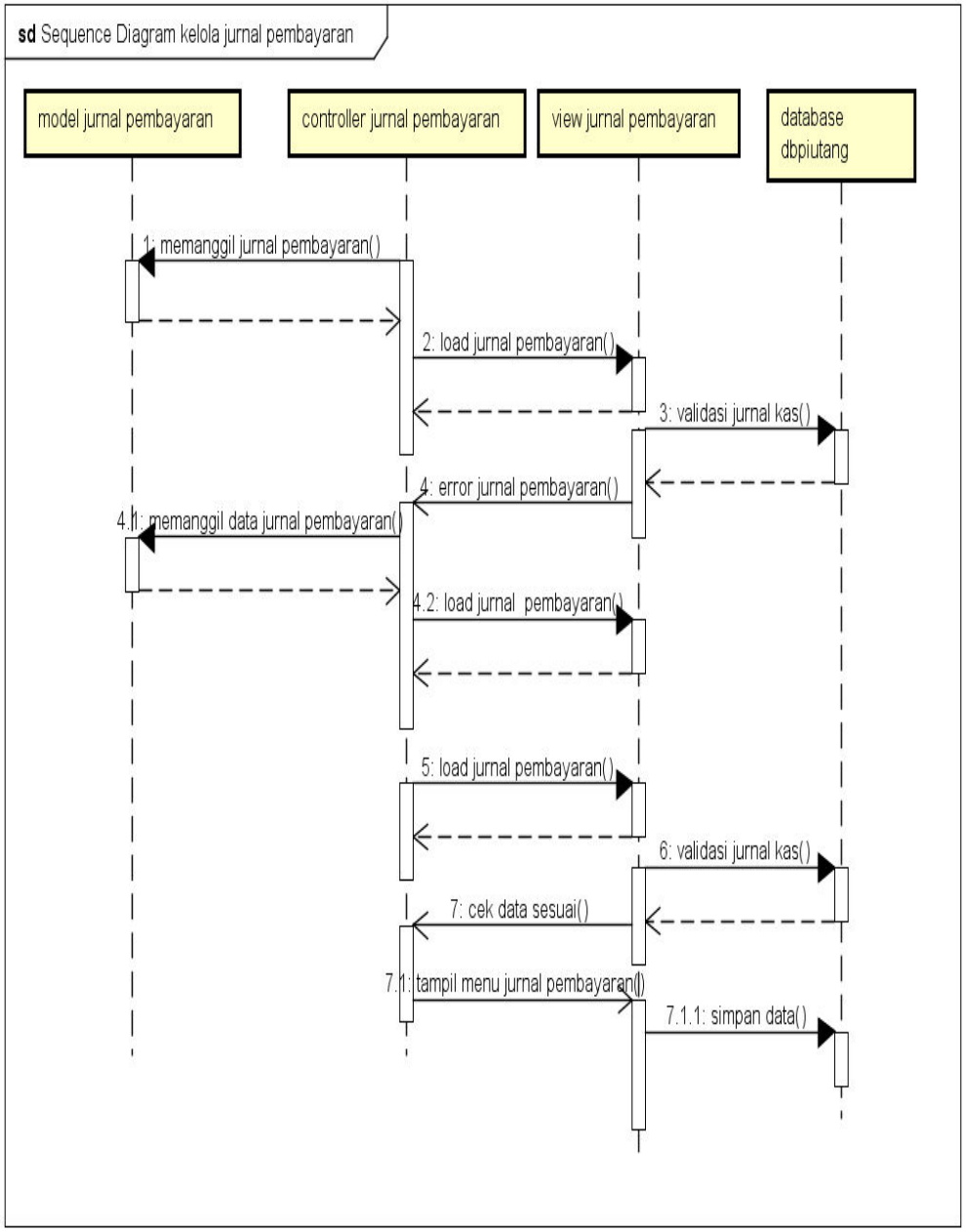
Gambar 4.17 *Sequence Diagram* Kelola Tagihan yang diusulkan



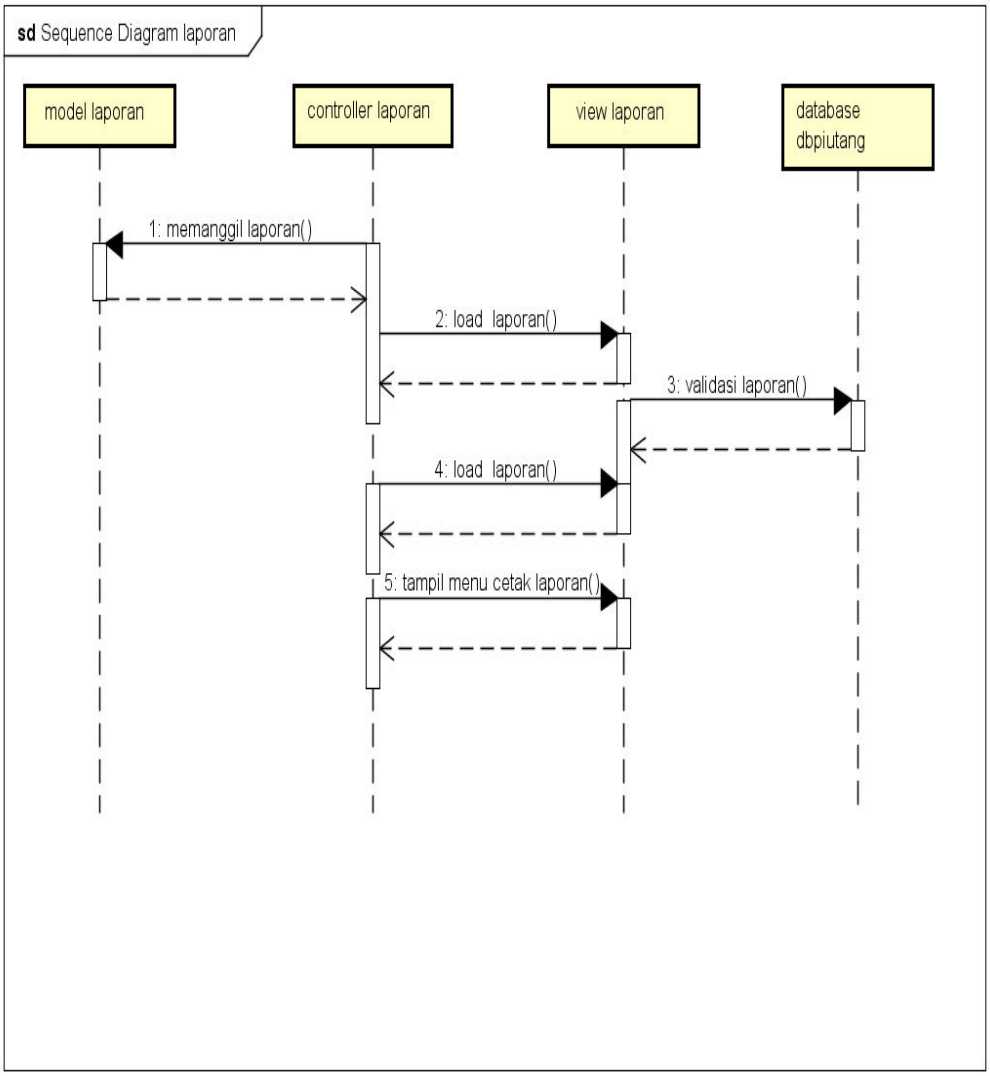
Gambar 4.18 *Sequence Diagram* Kelola Pembayaran yang diusulkan



Gambar 4.19 *Sequence Diagram* Kelola Jurnal Piutang yang diusulkan



Gambar 4.20 *Sequence Diagram* Kelola Jurnal Pembayaran yang diusulkan



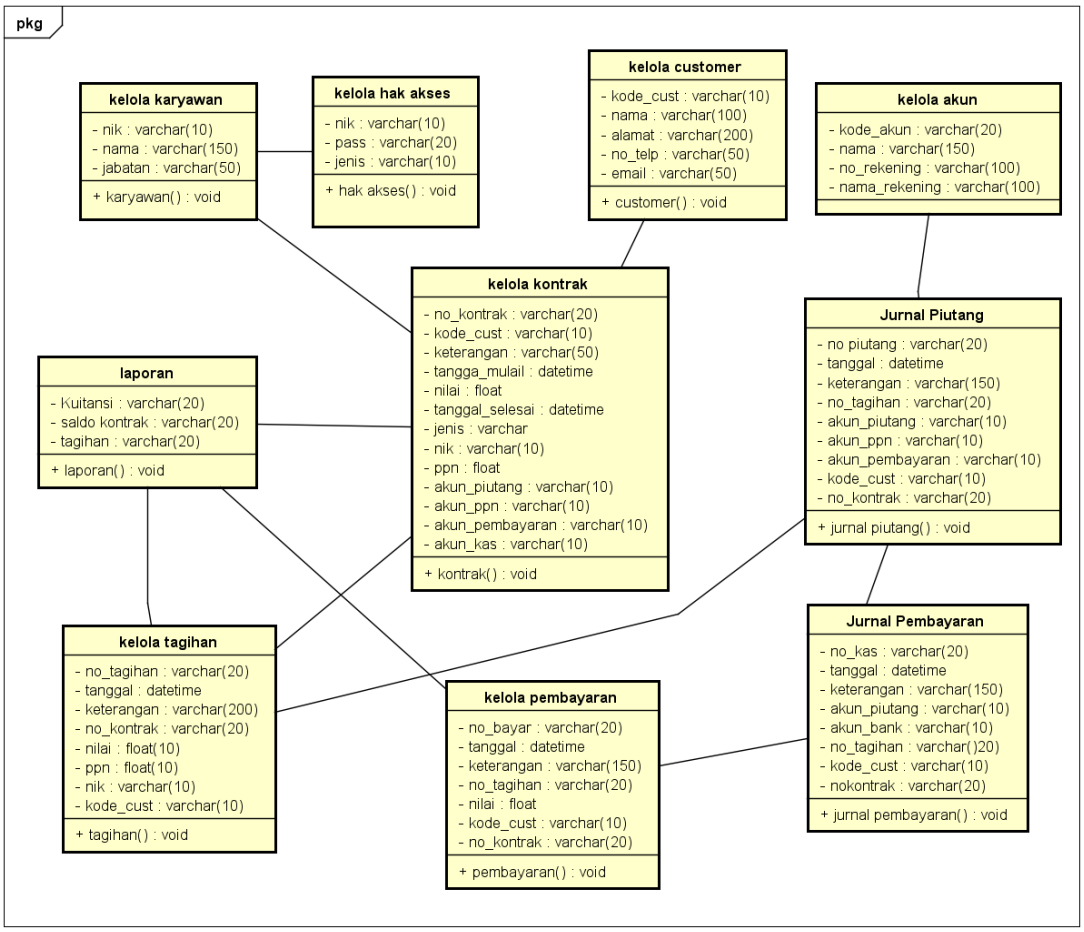
Gambar 4.21 *Sequence Diagram* Laporan yang diusulkan

4.1.4. Perancangan data

4.1.4.4 *Class Diagram*

Class diagram adalah suatu diagram yang memperlihatkan atau menampilkan struktur dari sebuah sistem. Sistem tersebut akan menampilkan sistem kelas, atribut dan hubungan antara kelas

Berikut adalah class diagram dari aplikasi Piutang usaha PT. SAMUDRA APLIKASI INDONESIA Bandung :

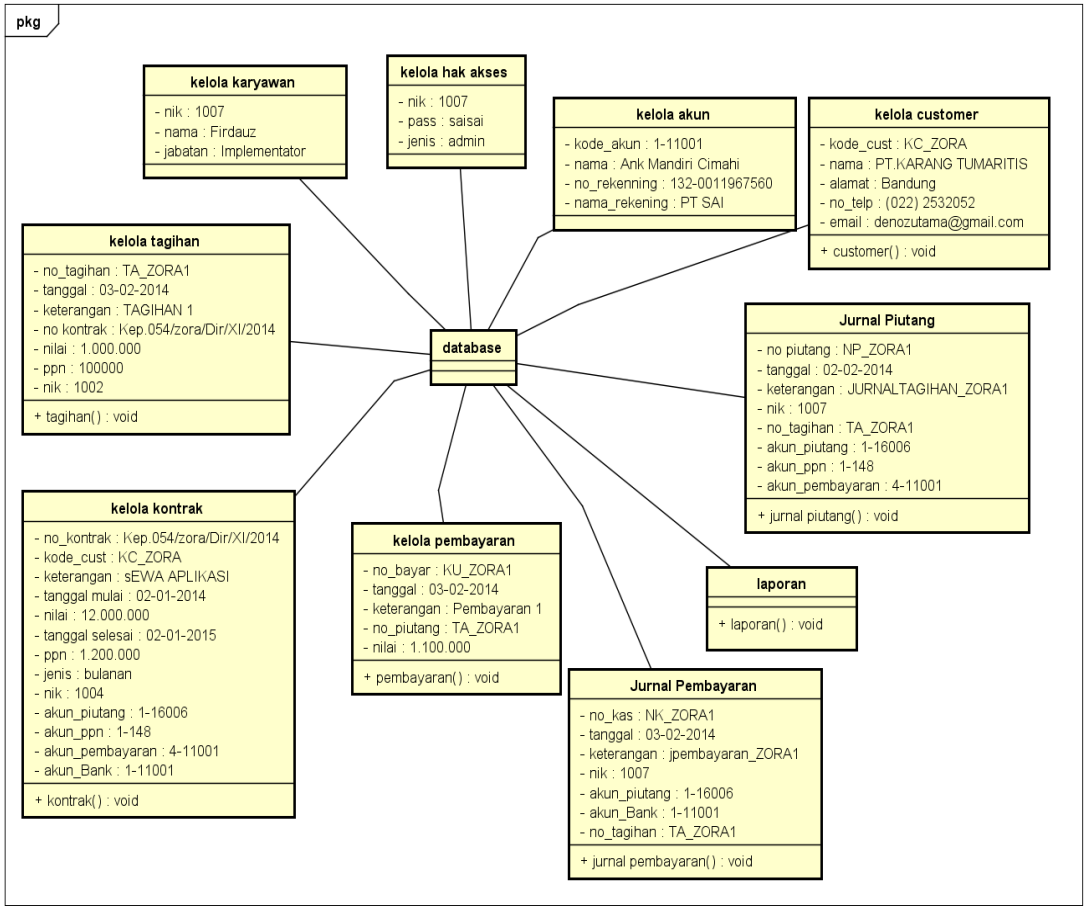


Gambar 4.22 *Class Diagram* yang diusulkan

4.1.4.2 *Objek Diagram*

Object Diagram menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Dalam diagram objek harus di pastikan bahwa semua kelas yang sudah didefinisikan pada diagram kelas harus pakai objeknya, karena jika tidak pendefinisian kelas itu tidak dapat dipertanggung jawabkan.

Sebuah diagram objek merupakan gambaran objek-objek pada sebuah sistem pada satu titik waktu. Karena lebih menonjolkan perintah-perintah dari pada kelas diagram objek sering disebut juga sebagai perintah.

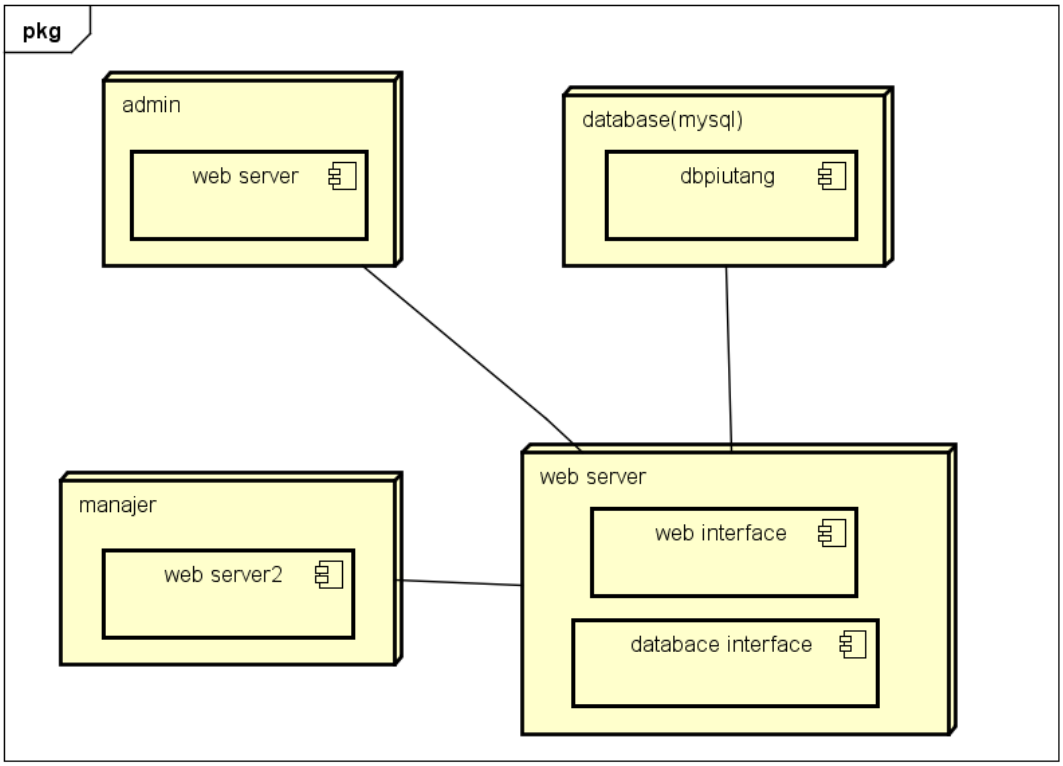


Gambar 4.23 *Objek Diagram* yang diusulkan

4.1.4.3 *Deployment Diagram*

Deployment diagram jika diartikan dalam bahasa Indonesia berarti diagram pendistribusian. Sesuai dengan kebutuhan sistem yang akan diterapkan dimana system membutuhkan jaringan lokal atau *local area network* (LAN) untuk menghubungkan komputer *server* dengan komputer *client*

Berikut adalah *deployment diagram* pada aplikasi laporan piutang

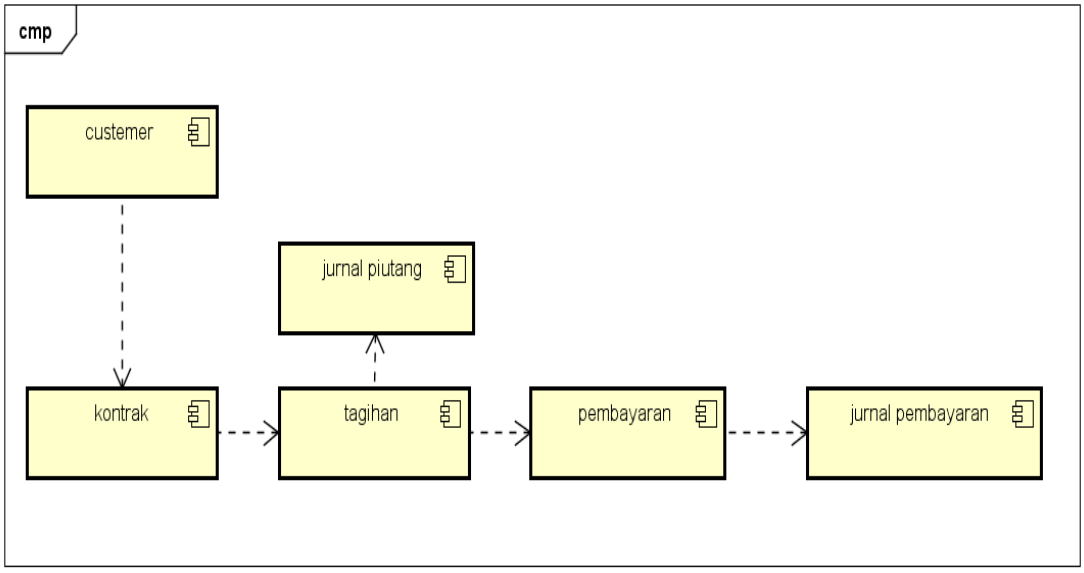


Gambar 4.24 *Deployment Diagram* yang diusulkan

4.1.4.4. *Componant Diagram*

Component diagram yaitu salah satu jenis diagram pada UML yang menampilkan komponen dalam system dan hubungan anatara mereka.

Berikut adalah *component diagram* pada aplikasi laporan piutang



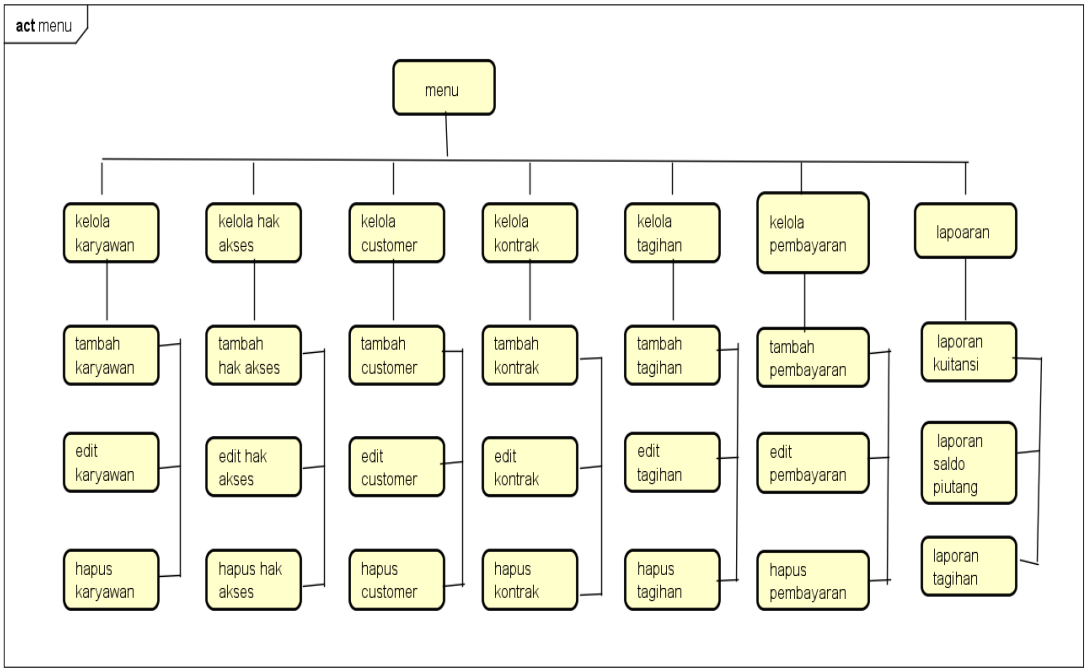
Gambar 4.25 *Componant Diagram* yang diusulkan

4.2. Perancangan Antar Muka

Pada perancangan antar muka atau *interface* ini akan menggambarkan rancangan tampilan yang ada dalam sistem yang diusulkan. Adapun yang akan dibahas yaitu struktur menu, perancangan *input* dan perancangan *output*.

4.2.1 Struktur menu

Pada perancangan struktur menu ini akan dijelaskan tentang link yang saling berhubungan dalam sistem yang diusulkan. Berikut adalah struktur menu pada aplikasi administrasi sarana & prasarana yang diusulkan.

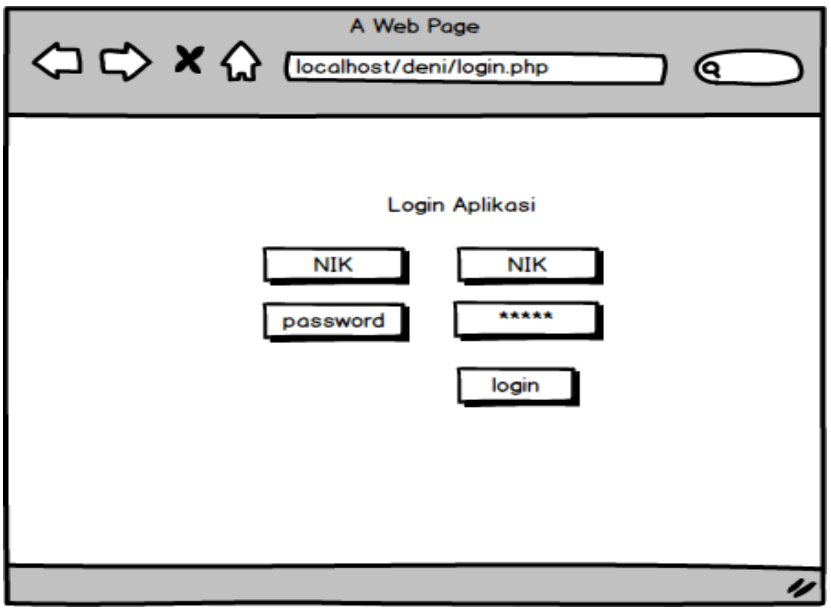


Gambar 4.26 Struktur Menu yang diusulkan

4.2.2 Perancangan input

Perancangan input adalah perancangan tampilan pada saat melakukan proses penginputan data pada form pengisian yang berkaitan dengan aplikasi piutang usaha. Adapun perancangan input yang diusulkan pada Aplikasi piutang usaha adalah sebagai berikut :

1. Perancangan tampilan login admin dan user harus melakukan login terlebih dahulu untuk dapat masuk ke aplikasi piutang usaha dengan memasukan username dan password yang sesuai dengan sistem jika username dan password tidak sesuai maka tidak dapat masuk ke aplikasi piutang usaha dan akan kembali ke form login untuk memasukan username dan password kembali dengan benar

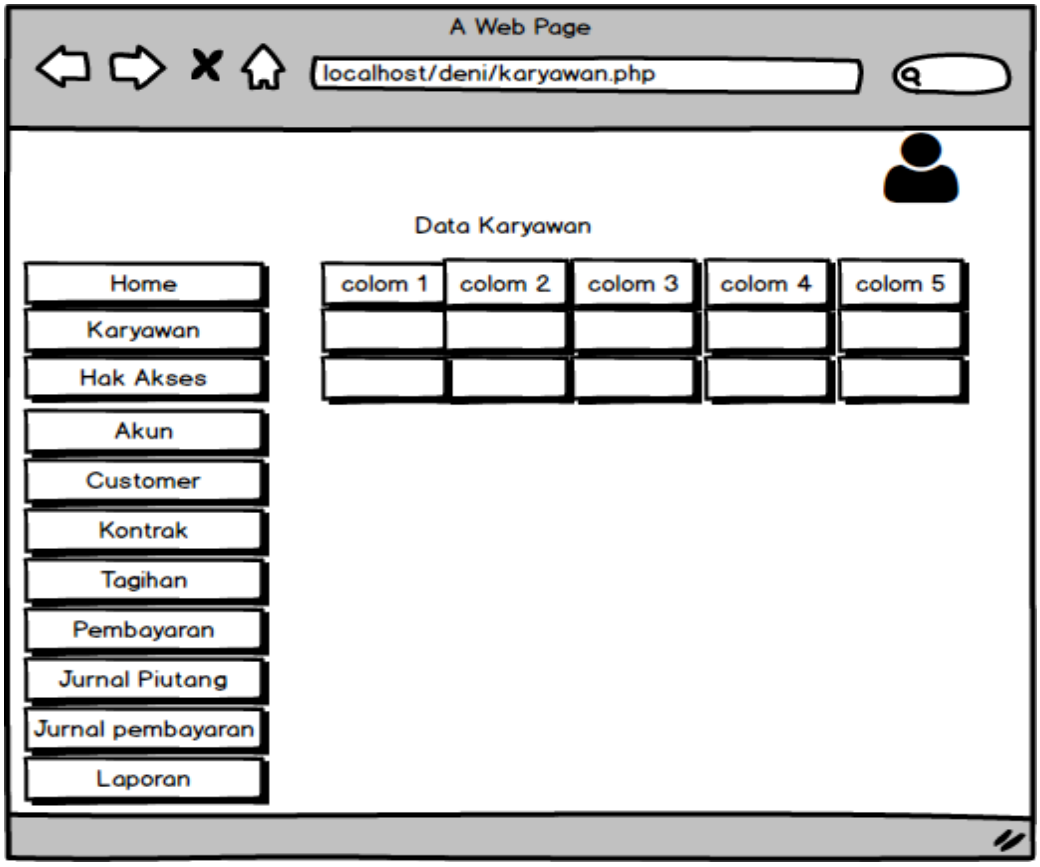


Gambar 4.27 Perancangan Tampilan Login

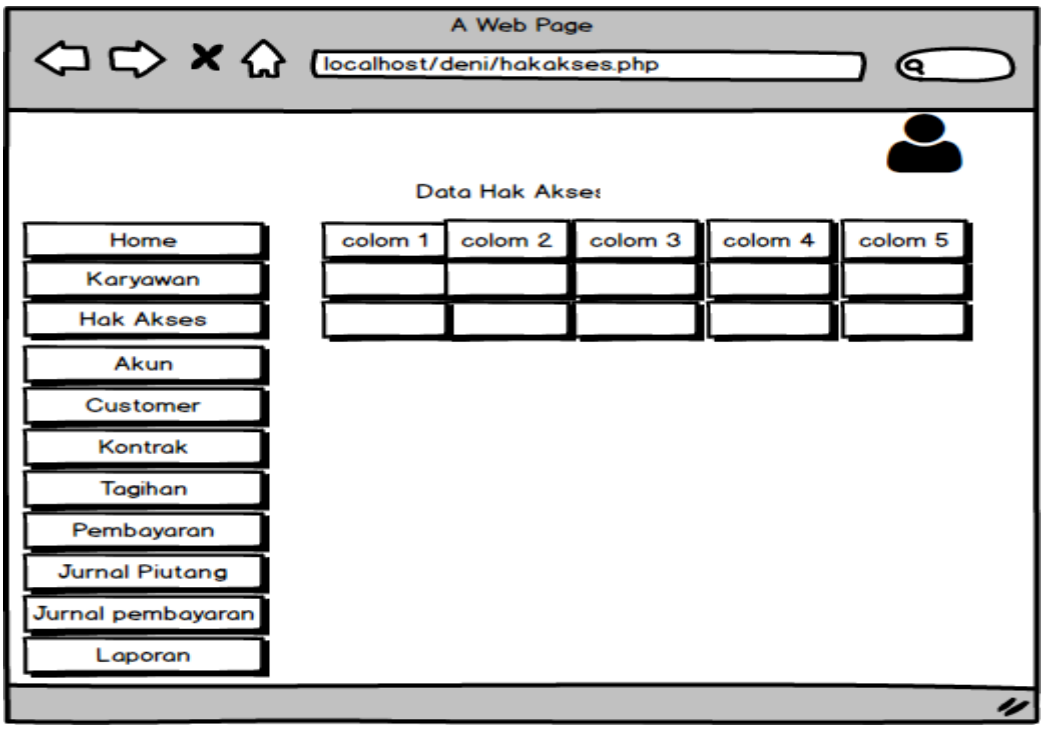
2. Perancangan menu utama

   Menu ini adalah menu utama untuk admin, setiap admin akan mengelola menu masing-masing user sesuai dengan hak akses user.

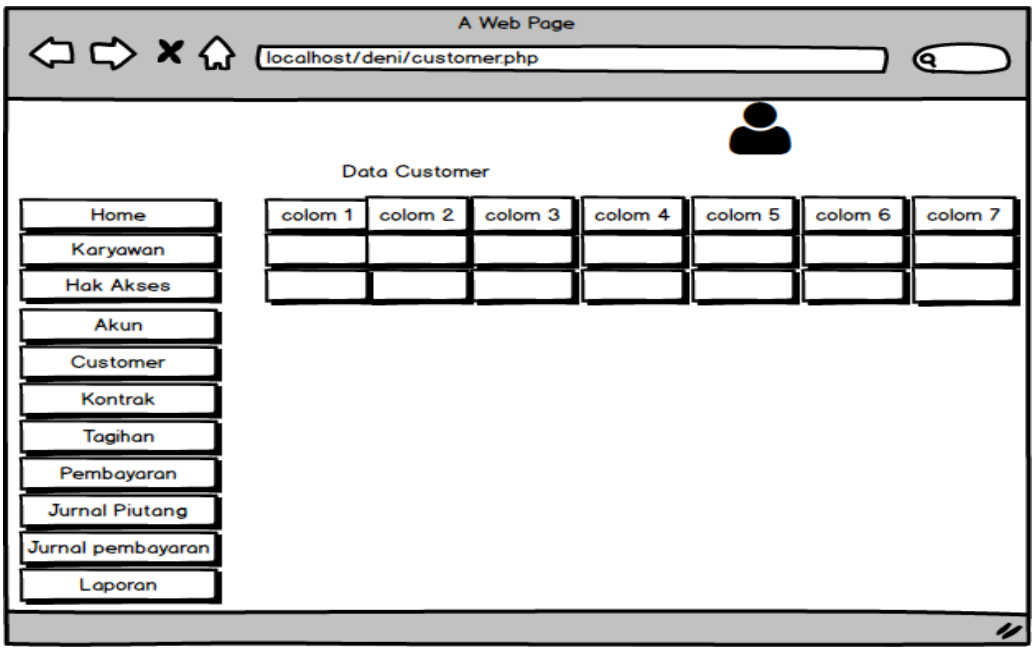
Gambar 4.28 Perancangan Menu Utama (home)



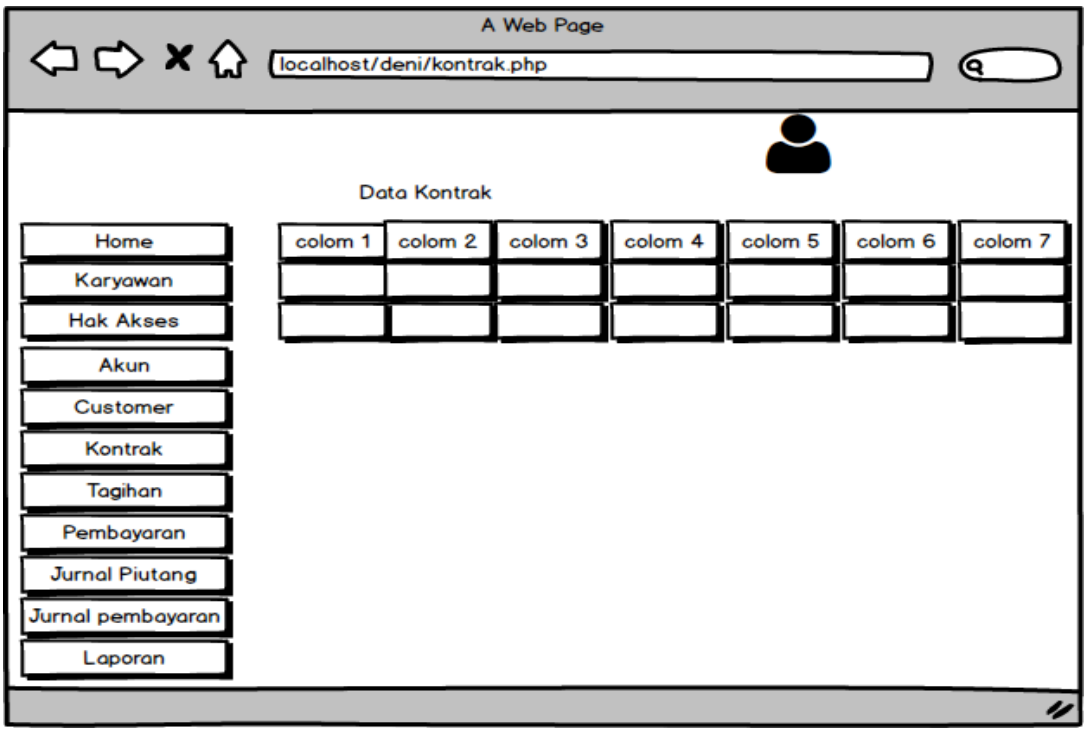
Gambar 4.28 Perancangan Menu Karyawan



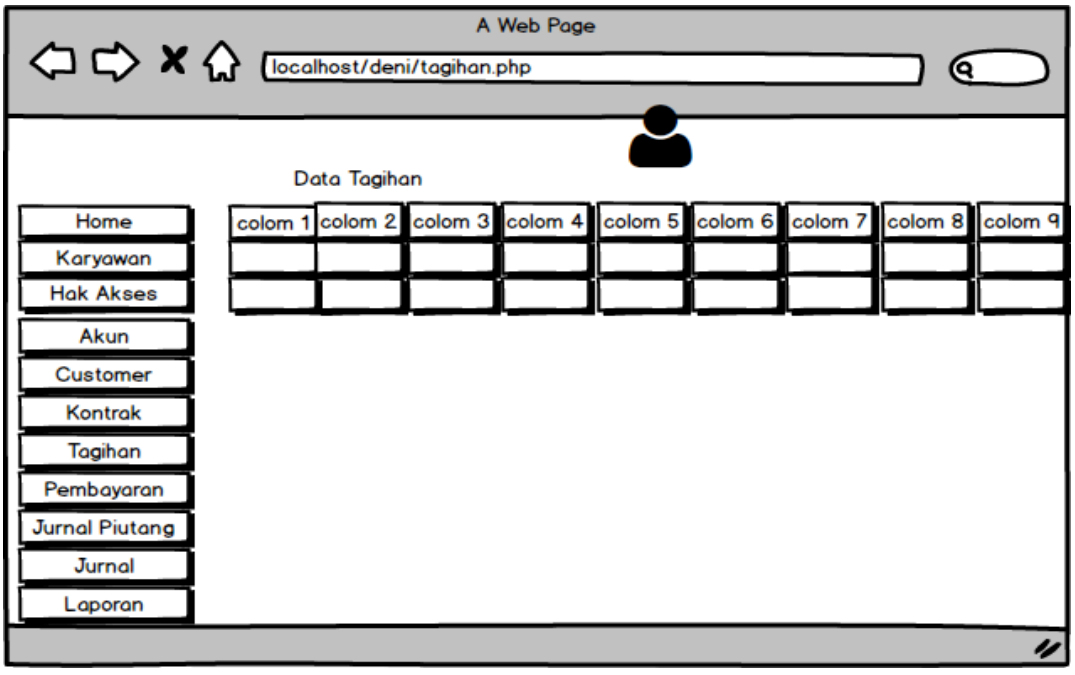
Gambar 4.29 Perancangan Menu Hak Akses



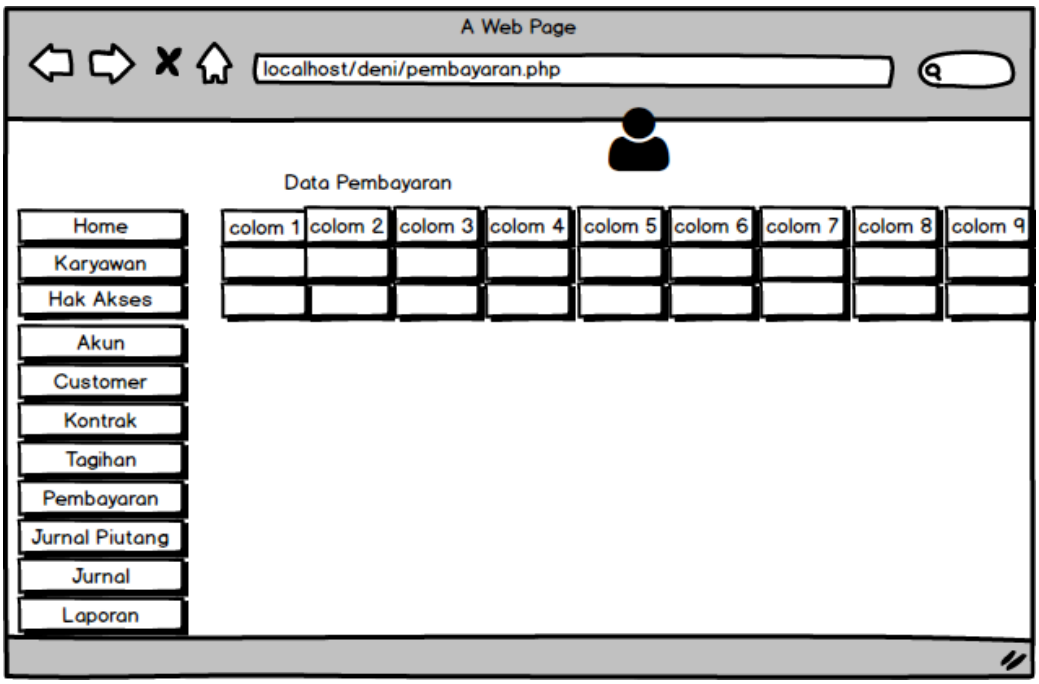
Gambar 4.30 PerancanganMenu Data Customer



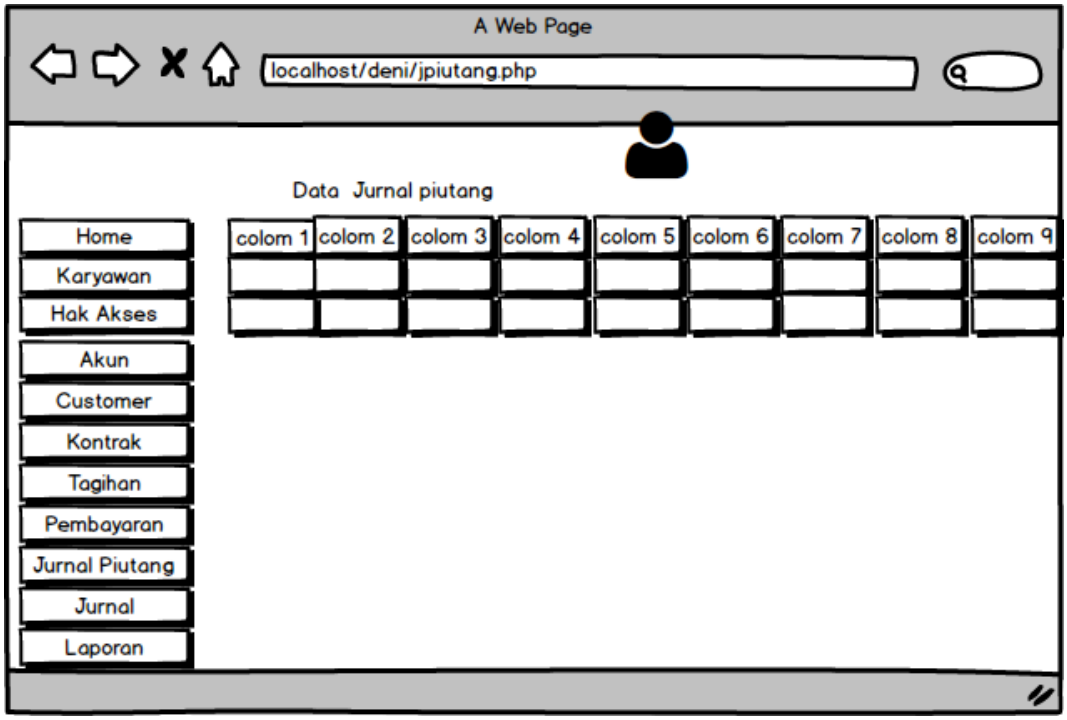
Gambar 4.31 Perancangan menu Kontrak



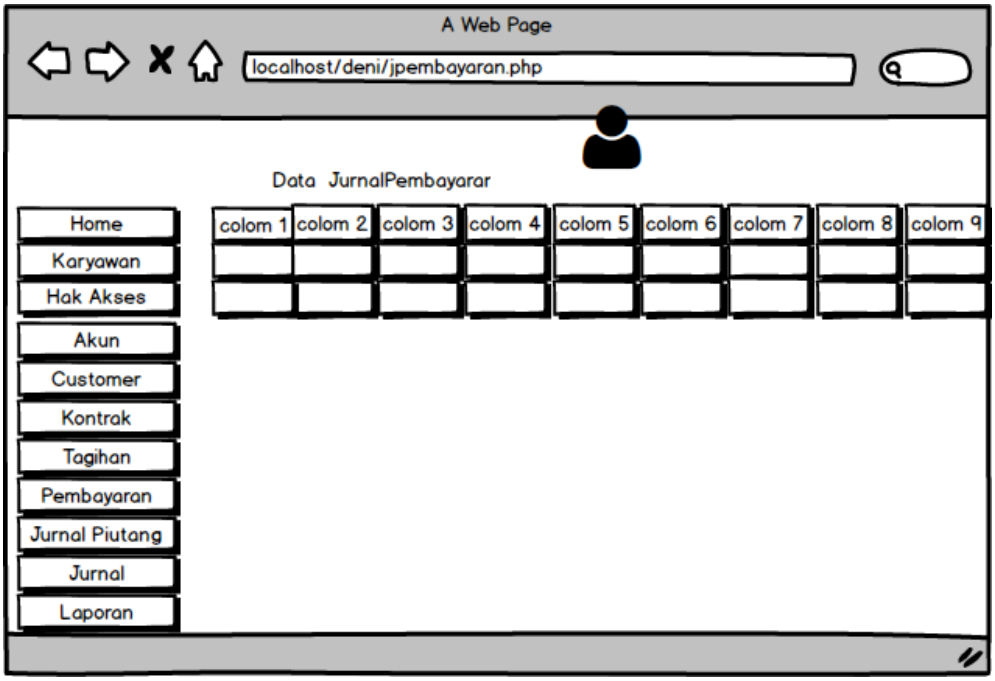
Gambar 4.32 Perancangan Menu Tagihan



Gambar 4.33 Perancangan Menu Pembayaran



Gambar 4.34 Perancangan Menu Jurnal Piutang



Gambar 4.35 Perancangan Menu Jurnal Pembayaran

4.2.3 Perancangan output

Perancangan output ini sangat berkaitan dengan data yang sudah diolah di dalam Aplikasi piutang usaha yang berbentuk sebuah laporan. Berikut ini perancangan nya

1. Perancangan laporan Kuitansi

   Perancangan ini berisikan keseluruhan laporan pembayaran.

Tabel 4.14 Laporan Kuitansi

No_bayar	Tanggal	Keterangan	No_tagihan	Nilai
KU_ZORA1	03-02-2014	PEMBAYARAN 1	TA_ZORA1	1100000

2. Perancangan laporan saldo piutang

Tabel 4.15 Laporan Saldo Kontrak

No_Kontrak	Tgl Mulai	Tgl Selesai	Customer	Keterangan	Nilai Kontrak	Nilai Tagihan	Nilai Pembayaran	Saldo Tagihan	Saldo Kontrak	Jumlah

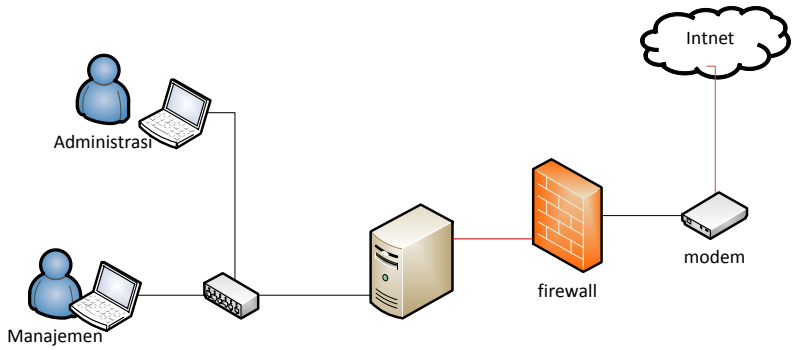
3.Perancangan laporan tagihan

Tabel 4.16 Laporan Tagihan

No_Tagihan	Tanggal	Keterangan	No_Kontrak	Nilai	Ppn	Nik
NT_ZORA1	02-02-2014	TAGIHAN 1	Kep.054.zora/dir/xi/2014	1000000	100000	1004

4.3 Perancangan Arsitektur Jaringan

Arsitektur jaringan Local Area Network suatu perangkat jaringan yang menghubungan komputer yang satu dengan komputer lainnya dengan menggunakan kabel dan router sebagai penyambungnya.. Arsitektur ini menempatkan sebuah *switch hub/router* sebagai penyambung jaringan dan server ini yang bertugas memberi jalur komunikasi yang diatur dengan firewall jaringan. Arsitektur jaringan ini juga menggunakan *topologi star.*



Gambar 4.36 Perancangan Aritektur Jaringan *Local Area Network*

4.4 Pengujian

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak.Tujuan dari pengujian adalah untuk menjamin perangkat lunak yang di bangun memilki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri. Dalam analisis dan perancangan ini

4.4.1 Rencana pengujian

Pengujian perangkat lunak menggunakan *black box testing*, dalam *black box testing*yang diuji adalah tentang masukan dan keluarannya.Dengan adanya masukan yang dilakukan diharapkan dapat memberikan keluaran seperti yang diharapkan atau sesuai dengan tujuan.

Tabel 4.17 Rencana Pengujian

Kelas Pengujian	Pengujian	Tingkat Pengujian	Jenis Pengujian
Login	Admin/User	Integrasi	Black Box
Pengujian penginputan data	Data Kelola Karyawan	Integrasi	Black Box
	Data Akun	Integrasi	Black Box
	Data Customer	Integrasi	Black Box
	Data Kontrak	Integrasi	Black Box
	Data Tagihan	Integrasi	Black Box
	Data Pembayaran	Integrasi	Black Box
	Data Jurnal Piutang	Integrasi	Black Box
	Data Jurnal Pembayaran	Integrasi	Black Box
Output	Laporan Kuitansi	Integrasi	Black Box
	Laporan Piutang	Integrasi	Black Box
	Laporan Piutang Usaha	Integrasi	Black Box

4.4.2 Kasus dan hasil pengujian

Setelah tahap rencana pengujian, dilanjutkan dengan kasus dan hasil pengujian berdasarkan kelas uji yaitu sebagai berikut

Tabel 4.18 Pengujian Kelola Hak Akses

Kasus dan hasil uji login petugas (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username : 1001 Password : admin	Dapat Login dan masuk ke halaman utama sebagai admin.	Dapat login , dan masuk ke menu.	[X] Diterima [ ] Ditolak
Kasus dan hasil uji Kelola User (Data Salah)			
Username : xxxx Password : asal	Tidak dapat Login dan Menampilkan pesan kesalahan username dan password salah	Pengguna tidak dapat login jika salah username dan password.	[ ] Diterima [X] Ditolak

Tabel 4.19 Pengujian Penginputan Data Kelola Karyawan

Kasus dan hasil uji data anggota (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input Data Karyawan	Data dapat ditampilkan sesuai dengan data yang telah diinputkan	Data telah sesuai dengan yang di inputkan	[X] Diterima [ ] Ditolak
Kasus dan hasil uji penginputan Data Kelola Karyawan (Data Salah)			

Input data karyawan tidak lengkap	Jika pada form ada yang tidak diisi maka klik simpan tidak bias	Sistem menampilkan peringatan Data kurang lengkap	[ ] Diterima [X] Ditolak

Tabel 4.20 Pengujian Penginputan Data Kelola Data Akun

Kasus dan hasil uji penginputan Akun (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input Data akun	Input berhasil	Data akun terisi lengkap dan sesuai	[X] Diterima [ ] Ditolak
Kasus dan hasil uji Data Kelola Data Akun(Data Salah)			
Input Data akun tidak lengkap	Sistem memberikan alert atau informasi data tidak boleh kosong	Data tidak tersimpan ke database karena ada field yang kosong	[ ] Diterima [X] Ditolak

Tabel 4.21 Pengujian Penginputan Data Kelola Data Kontrak

Kasus dan hasil uji penginputan Kontrak (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input Data kontrak	Input berhasil	Data kontrak terisi lengkap dan sesuai	[X] Diterima [ ] Ditolak

Kasus dan hasil uji Data Kelola Data Kontrak(Data Salah)			
Input Data kontrak tidak lengkap	Sistem memberikan alert atau informasi data tidak boleh kosong	Data tidak tersimpan ke database karena ada field yang kosong	[ ] Diterima [X] Ditolak

Tabel 4.22 Pengujian Penginputan Data Kelola Data Tagihan

Kasus dan hasil uji penginputan Tagihan (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input Data tagihan	Input berhasil	Data tagihan terisi lengkap dan sesuai	[X] Diterima [ ] Ditolak
Kasus dan hasil uji Data Kelola Data Tagihan(Data Salah)			
Input Data tagihan tidak lengkap	Sistem memberikan alert atau informasi data tidak boleh kosong	Data tidak tersimpan ke database karena ada field yang kosong	[ ] Diterima [X] Ditolak

Tabel 4.23 Pengujian Penginputan Data Kelola Data Pembayaran

Kasus dan hasil uji penginputan Pembayaran (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input Data pembayaran	Input berhasil	Data pembayaran terisi lengkap dan sesuai	[X] Diterima [ ] Ditolak
Kasus dan hasil uji Data Kelola Data Pembayaran(Data Salah)			
Input Data pembayaran tidak lengkap	Sistem memberikan alert atau informasi data tidak boleh kosong	Data tidak tersimpan ke database karena ada field yang kosong	[ ] Diterima [X] Ditolak

Tabel 4.24 Pengujian Penginputan Data Kelola Data Customer

Kasus dan hasil uji penginputan Customer (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input Data customer	Input berhasil	Data customer terisi lengkap dan sesuai	[X] Diterima [ ] Ditolak
Kasus dan hasil uji Data Kelola Data Customer(Data Salah)			
Input Data customer tidak	Sistem memberikan alert	Data tidak tersimpan ke	[ ] Diterima [X] Ditolak

lengkap	atau informasi data tidak boleh kosong	database karena ada field yang kosong	

Tabel 4.25 Pengujian Penginputan Data Kelola Data Jurnal Piutang

Kasus dan hasil uji penginputan Jurnal Piutang (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input Data jurnal piutang	Input berhasil	Data jurnal piutang terisi lengkap dan sesuai	[X] Diterima [ ] Ditolak
Kasus dan hasil uji Data Kelola Data Jurnal Piutang(Data Salah)			
Input Data jurnal piutang tidak lengkap	Sistem memberikan alert atau informasi data tidak boleh kosong	Data tidak tersimpan ke database karena ada field yang kosong	[ ] Diterima [X] Ditolak

Tabel 4.26 Pengujian Penginputan Data Kelola Data Jurnal Pembayaran

Kasus dan hasil uji penginputan Jurnal Pembayaran (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input Data jurnal pembayaran	Input berhasil	Data jurnal pembayaran terisi	[X] Diterima [ ] Ditolak

		lengkap dan sesuai	
Kasus dan hasil uji Data Kelola Data Jurnal Pembayaran (Data Salah)			
Input Data jurnal pembayaran tidak lengkap	Sistem memberikan alert atau informasi data tidak boleh kosong	Data tidak tersimpan ke database karena ada field yang kosong	[ ] Diterima [X] Ditolak

Tabel 4.27 Pengujian Proses Laporan Kuitansi

Kasus dan hasil uji penginputan Laporan Kuitansi(Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
			[X] Diterima [ ] Ditolak
Kasus dan hasil uji data pengembalian (Data Salah)			
			[] Diterima [X] Ditolak

Tabel 4.28 Pengujian Proses Laporan Piutang Kontrak

Kasus dan hasil uji penginputan Laporan Piutang Usaha(Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
			[X] Diterima [ ] Ditolak

Kasus dan hasil uji data pengembalian (Data Salah)			
			[] Diterima [X] Ditolak

Tabel 4.9 Pengujian Proses Laporan Tagihan

Kasus dan hasil uji penginputan Laporan Tagihan(Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
			[X] Diterima [ ] Ditolak
Kasus dan hasil uji data pengembalian (Data Salah)			
			[] Diterima [X] Ditolak

4.4.3 Kesimpulan hasil pengujian

Berdasarkan hasil pengujian diatas yang di lakukan menggunakan metode blackbox dapat disimpulkan bahwa sisitem informasi sudah layak pakai dan sesuai dengan fungsinya

4.5 Implementasi

Implementasi dari aplikasi piutang usaha, dilakukan menggunakan bahasa pemrograman PHP dengan *Notepad*++ sebagai perangkat lunaknya dan SQL sebagai basis data dengan MySQL sebagai perangkat lunaknya.

4.5.1 Implementasi perangkat lunak

Dalam melakukan implementasi Aplikasi administrasi sarana prasarana pendidikan ini menggunkan perangkat lunak yang di uraikan sebagai berikut :

1. Program Apache.
2. Database MYSQL
3. *Notepad++*
4. *Navicat Premium*

4.5.2 Implementasi perangkat keras

Dalam melakukan implementasi Aplikasi piutang usaha ini menggunkan perangkat keras yang mendukung pengunaan perangkat lunak nya yang di uraikan sebagai berikut :

1. Spesifikasi hardware untuk komputer server
   a. Processor : Xeon
   b. Memori (RAM) : 4Gb
   c. Hard disk : 160Gb
   d. *Printer* digunakan sebagai peralatan mencetak *output* laporan
   e. *Mouse*, *keyboard* dan *monitor* sebagai perlengkapan alat antar muka

4.5.3 Implementasi basis data

Implementasi basis data pada Aplikasi Administrasi sarana prasarana pendidikan ini menggunakan MYSQL, berikut ini uraian nya :

1. Tabel Karyawan

```
DROP TABLE IF EXISTS `karyawan`;
CREATE TABLE `karyawan` (
`nik` varchar(10) NOT NULL DEFAULT '',
`nama` varchar(150) DEFAULT NULL,
`jabatan` varchar(50) DEFAULT NULL,
PRIMARY KEY (`nik`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

2. Tabel Hak Akses

```
DROP TABLE IF EXISTS `hakakses`;
CREATE TABLE `hakakses` (
`nik` varchar(10) NOT NULL DEFAULT '',
`pass` varchar(20) DEFAULT NULL,
`jenis` varchar(10) DEFAULT NULL,
`menu` varchar(50) DEFAULT NULL,
PRIMARY KEY (`nik`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3. Tabel Akun

```
DROP TABLE IF EXISTS `akun`;
CREATE TABLE `akun` (
`kode_akun` varchar(20) NOT NULL DEFAULT '',
`nama` varchar(150) DEFAULT NULL,
```

```
`no_rekening` varchar(100) DEFAULT NULL,
`nama_rekening` varchar(100) DEFAULT NULL,
PRIMARY KEY (`kode_akun`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4. Tabel Customer

```
DROP TABLE IF EXISTS `customer`;
CREATE TABLE `customer` (
  `kode_cust` varchar(20) NOT NULL DEFAULT '',
  `nama` varchar(100) DEFAULT NULL,
  `alamat` varchar(200) DEFAULT NULL,
  `no_telp` varchar(50) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`kode_cust`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

5. Tabel Kontrak

```
DROP TABLE IF EXISTS `kontrak`;
CREATE TABLE `kontrak` (
  `no_kontrak` varchar(50) NOT NULL DEFAULT '',
  `kode_cust` varchar(20) DEFAULT NULL,
  `keterangan` varchar(150) DEFAULT NULL,
  `tgl_mulai` datetime DEFAULT NULL,
```

```
  `nilai` float DEFAULT NULL,
  `nik` varchar(10) DEFAULT NULL,
  `tgl_selesai` datetime DEFAULT NULL,
  `ppn` float DEFAULT NULL,
  `file_dok` varchar(150) DEFAULT NULL,
  `jenis` varchar(10) DEFAULT NULL,
  `akun_ppn` varchar(10) DEFAULT NULL,
  `akun_piutang` varchar(10) DEFAULT NULL,
  `akun_pendapatan` varchar(10) DEFAULT NULL,
  `akun_kas` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`no_kontrak`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6. Tabel Tagihan

```
DROP TABLE IF EXISTS `tagihan`;
CREATE TABLE `tagihan` (
  `no_tagihan` varchar(20) NOT NULL DEFAULT '',
  `tanggal` datetime DEFAULT NULL,
  `keterangan` varchar(200) DEFAULT NULL,
  `no_kontrak` varchar(50) DEFAULT NULL,
  `nilai` float(10,0) DEFAULT NULL,
  `ppn` float(10,0) DEFAULT NULL,
```

```
  `nik` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`no_tagihan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

7. Tabel Pembayaran

```
DROP TABLE IF EXISTS `pembayaran`;
CREATE TABLE `pembayaran` (
  `no_bayar` varchar(20) NOT NULL DEFAULT '',
  `tanggal` datetime DEFAULT NULL,
  `keterangan` varchar(150) DEFAULT NULL,
  `no_tagihan` varchar(20) DEFAULT NULL,
  `nilai` float DEFAULT NULL,
  PRIMARY KEY (`no_bayar`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

8. Tabel Jurnal Piutang

```
DROP TABLE IF EXISTS `jurnal_piutang`;
CREATE TABLE `jurnal_piutang` (
  `no_piutang` varchar(20) NOT NULL DEFAULT '',
  `tanggal` datetime DEFAULT NULL,
  `nik` varchar(10) DEFAULT NULL,
  `no_tagihan` varchar(20) DEFAULT NULL,
  `akun_piutang` varchar(10) DEFAULT NULL,
```

```
  `akun_ppn` varchar(10) DEFAULT NULL,
  `akun_pendapatan` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

9. Tabel Jurnal Pembayaran

```
DROP TABLE IF EXISTS `jurnal_kas`;
CREATE TABLE `jurnal_kas` (
  `no_kas` varchar(20) NOT NULL DEFAULT '',
  `tanggal` datetime DEFAULT NULL,
  `keterangan` varchar(150) DEFAULT NULL,,
  `akun_piutang` varchar(10) DEFAULT NULL,
  `akun_kas` varchar(10) DEFAULT NULL,
  `no_tagihan` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`no_kas`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4.5.4 Implementasi antar muka

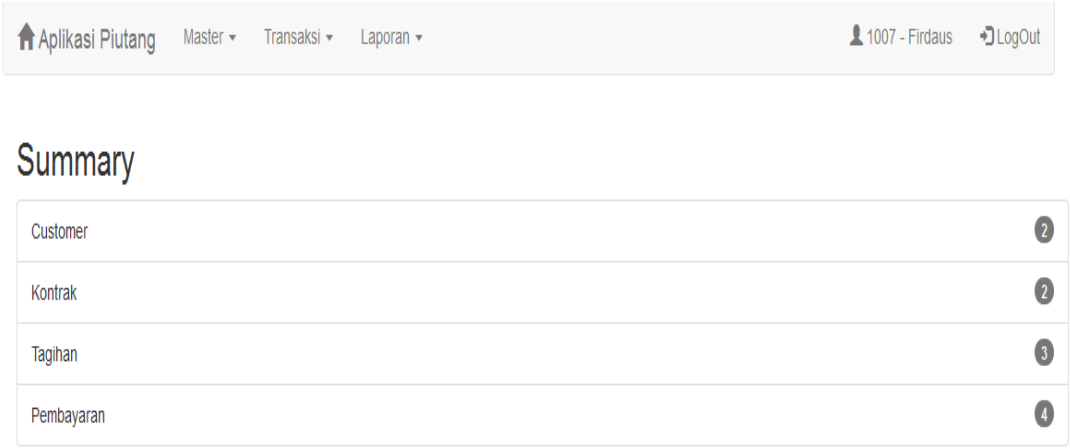
Implementasi antar muka merupakan gambaran mengenai antar muka dari perangkat lunak yang dibuat. Implementasi antar muka pada piutang usaha dirancang dengan menampilkan setiap tampilan program yang telah dibuat dalam bentuk file

1. Tampilan login untuk pengguna aplikasi



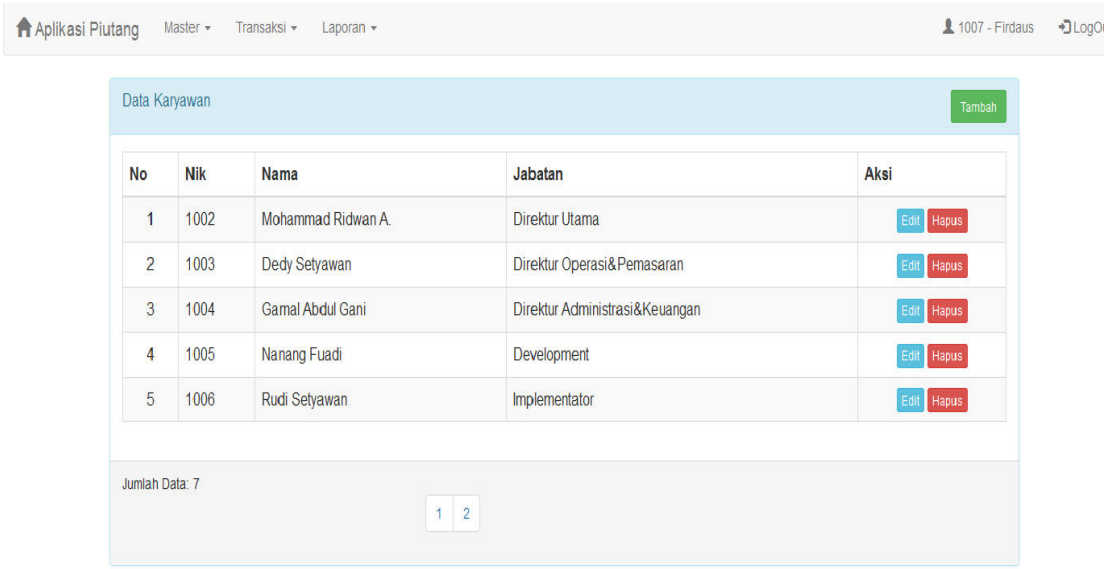
Gambar 4.37 Tampilan Login

2. Tampilan menu utama



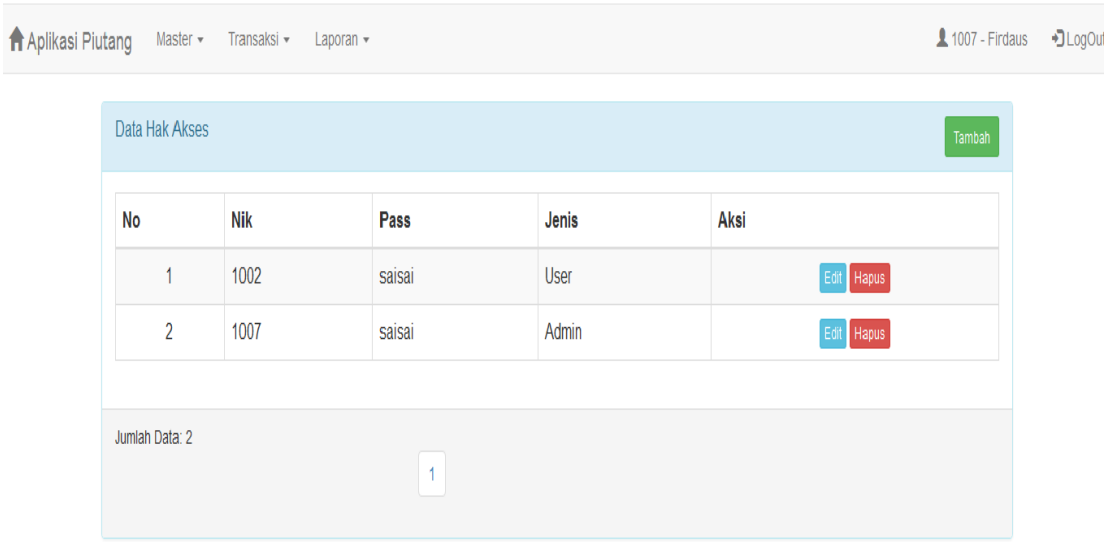
Gambar 4.38 Tampilan Menu Dashboard

3. Tampilan Kelola Karyawan



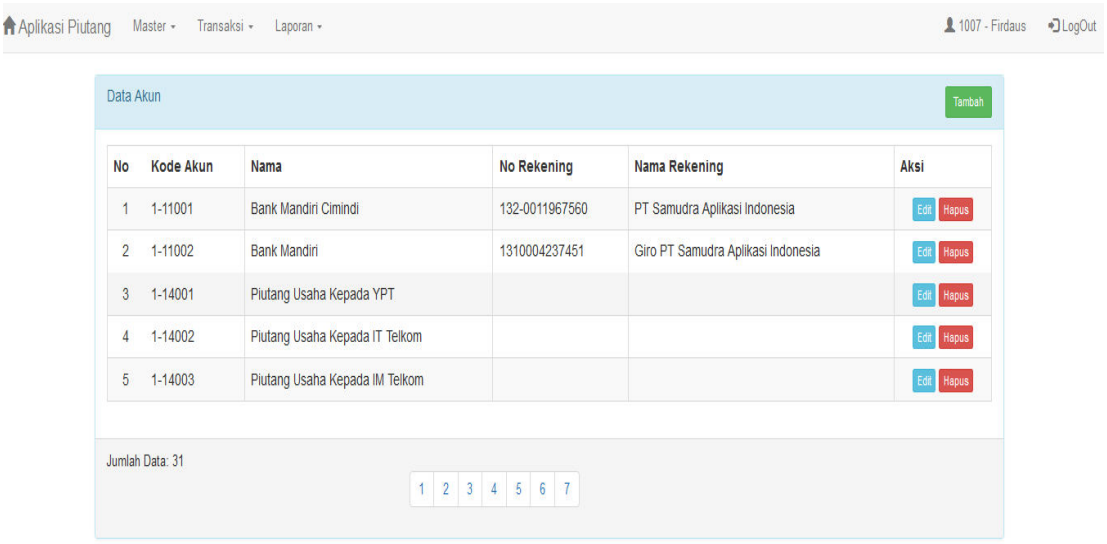
Gambar 4.39 Tampilan Menu Kelola Karyawan

4. Tampilan Kelola Hak Akses



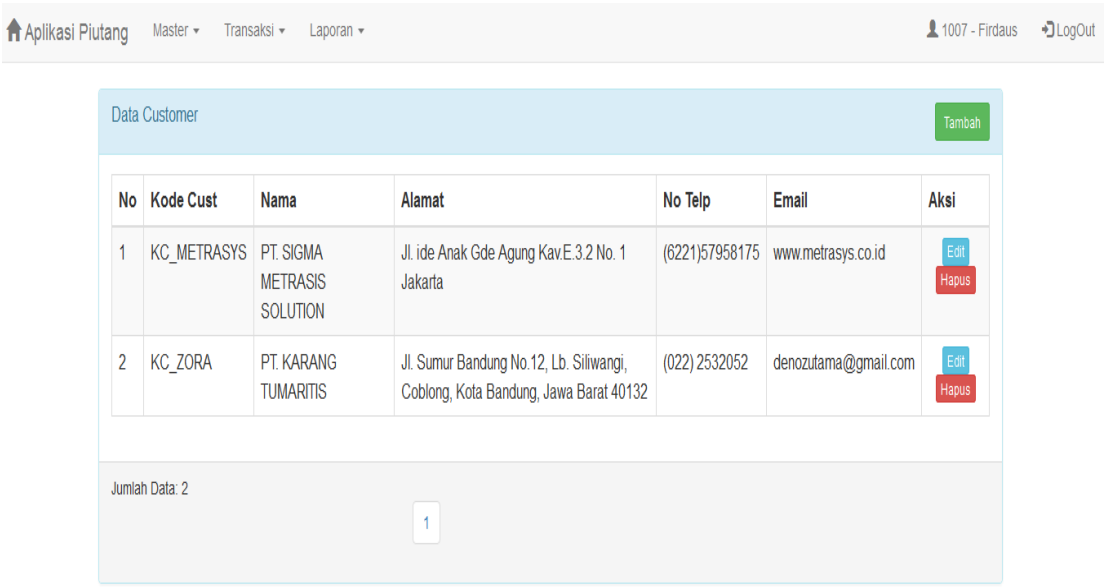
Gambar 4.40 Tampilan Menu Hak Akses

5. Tampilan Kelola Akun



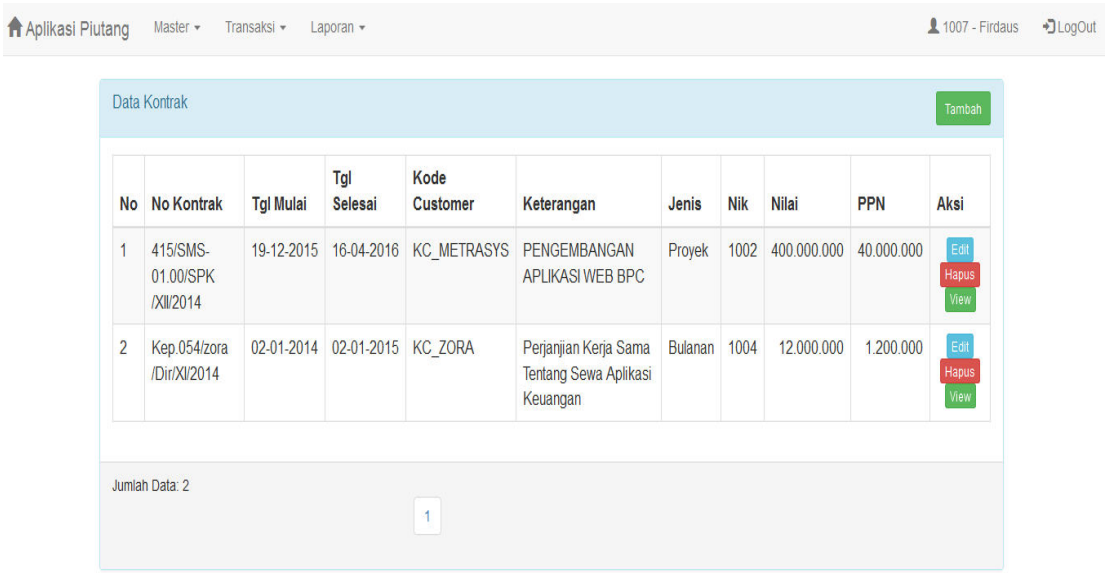
Gambar 4.41 Tampilan Kelola Akun

6. Tampilan Kelola Custome



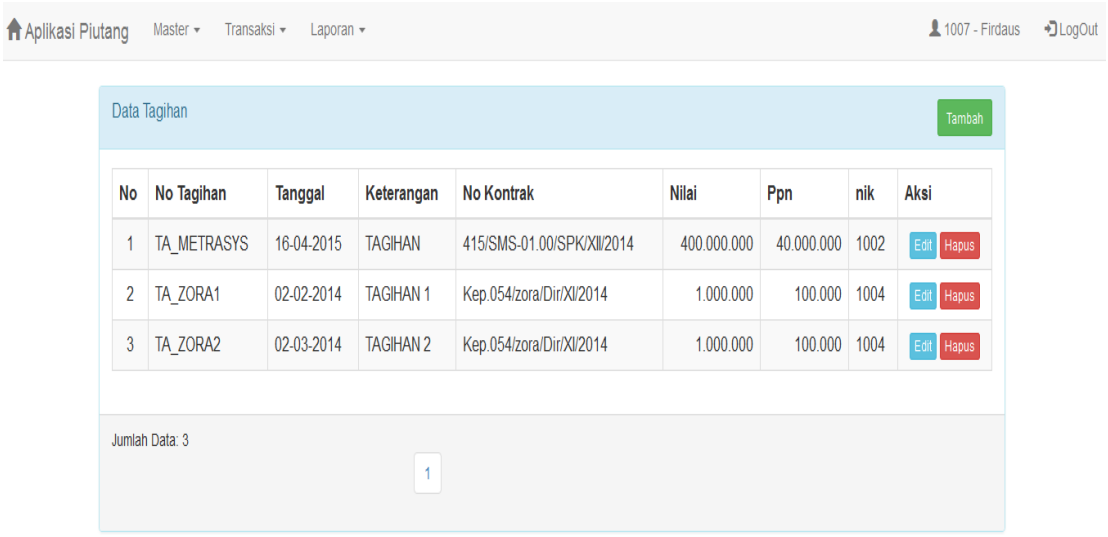
Gambar 4.42 Tampilan Kelola Customer

7. Tampilan Kelola Kontrak



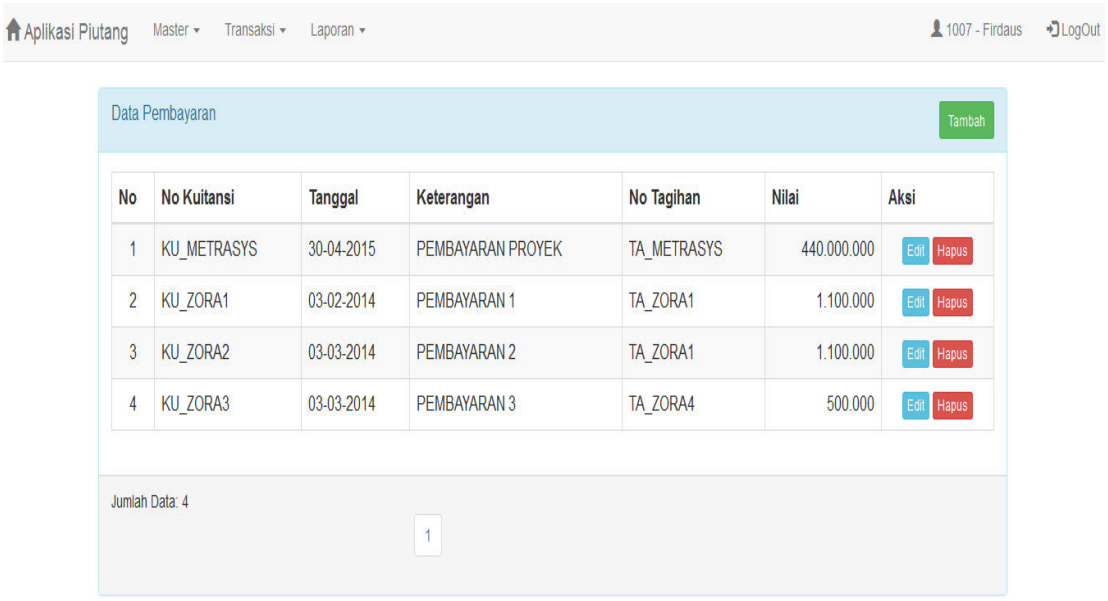
Gambar 4.43 Tampilan Kelola Kontrak

8. Tampilan Kelola Tagihan



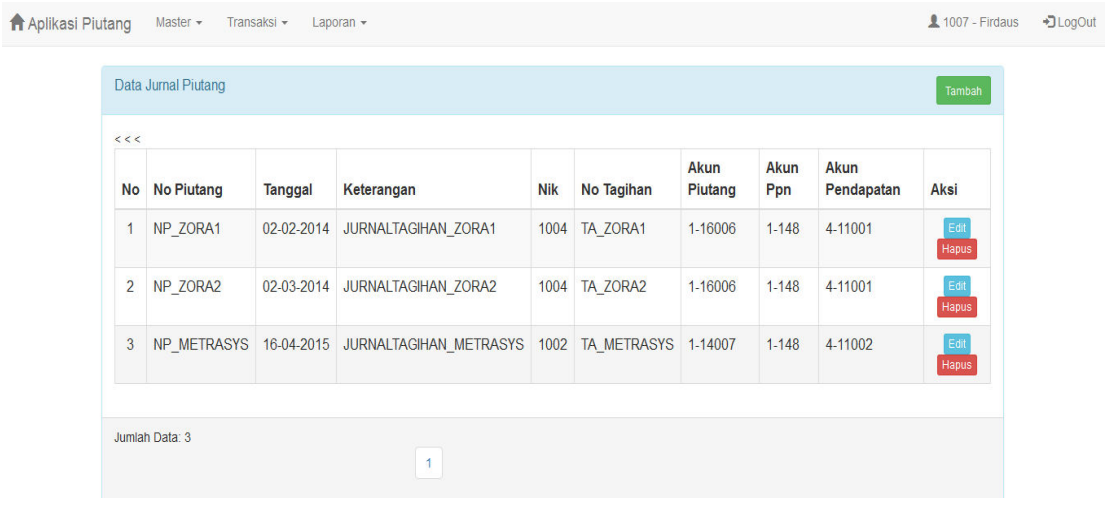
Gambar 4.44 Tampilan Kelola Tagihan

9. Tampilan Kelola Pembayaran



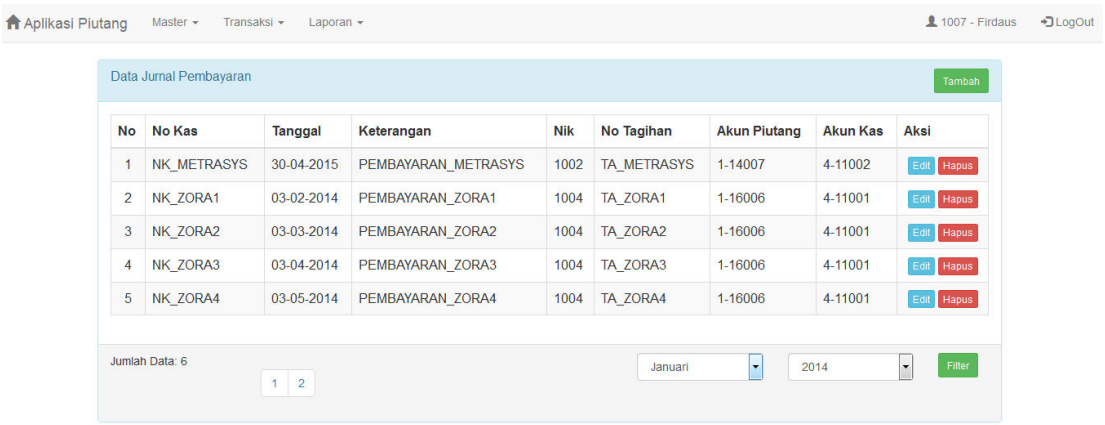
Gambar 4.45 Tampilan Kelola Pembayaran

10. Tampilan Kelola Jurnal Piutang



Gambar 4.46 Tampilan Kelola Jurnal Piutang

11. Tampilan Kelola Jurnal Pembayaran



Gambar 4.47 Tampilan Kelola Jurnal Pembayaran

4.5.5 Implementasi instalasi program

Berikut ini adalah langkah-langkah nya :

1. Download file installer Apache2.4
2. Setelah selesai di dowmload double klik file tersebut untuk memulai penginstalasian.



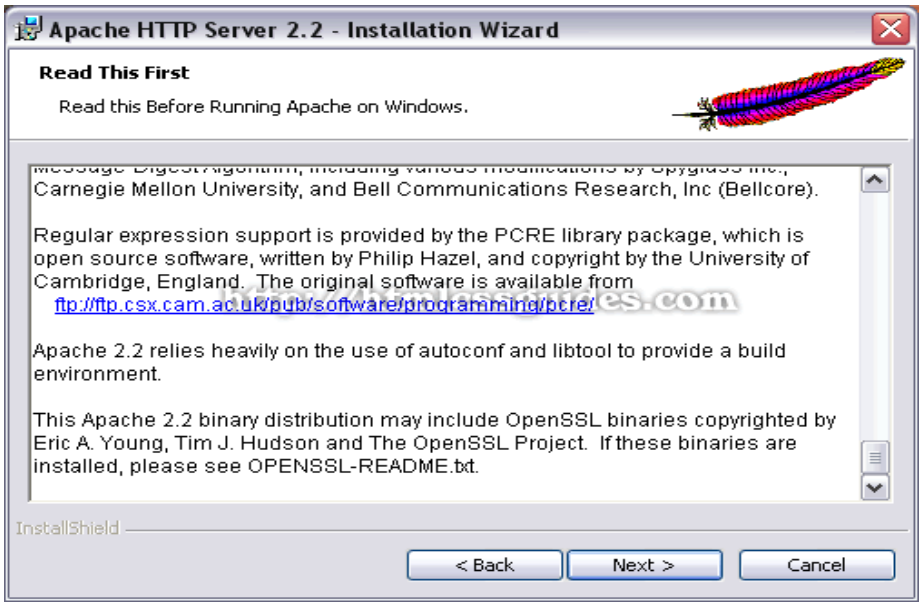
Gambar 4.48 Tampilan Instalasi Apache

3. klik next Tandai "I accept the terms in the lisense agreement," kemudian klik Next.



Gambar 4.49 Tampilan Apache

4. Anda bisa membaca penjelasan dan petunjuk tentang Apache Server, kemudian klik Next



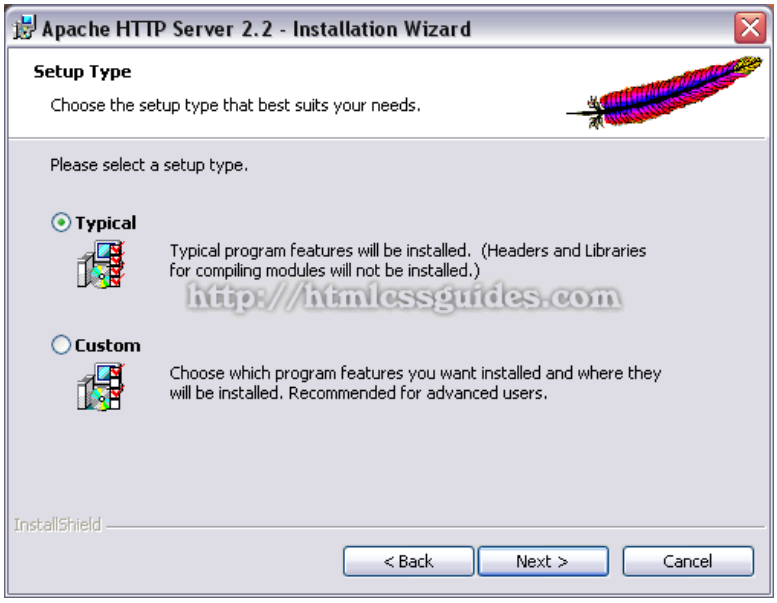
Gambar 4.50 Tampilan Penjelasan Apache

5. Selanjutnya isikan dengan data



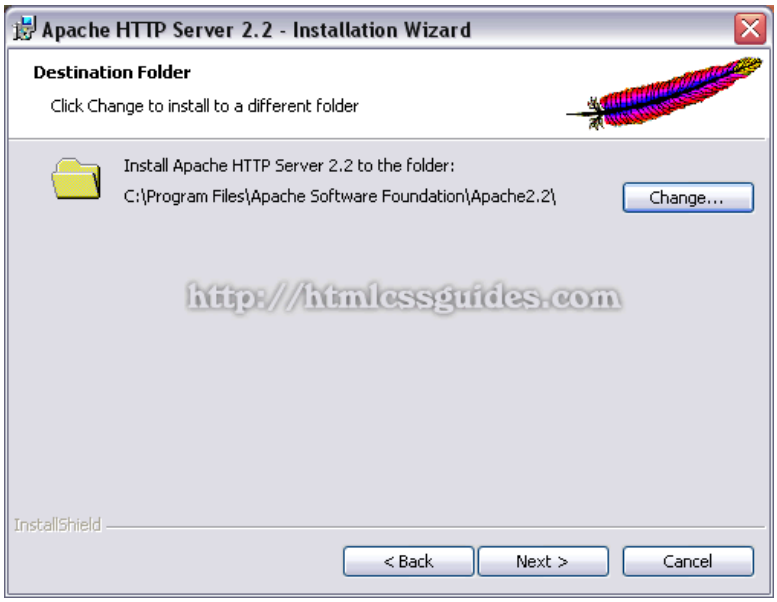
Gambar 4.51 Tampilan Pengisian Data

6. Pilih "Typikal" untuk tipe penginstalasian.



Gambar 4.52 Tampilan Tipe Penginstalan

7. Kemudian klik "Change" untuk mengubah folder tujuan penginstalasian



Gambar 4.53 Tampilan Penyimpanan

8. Buat sebuah folder baru di drive C dengan nama "Apache"



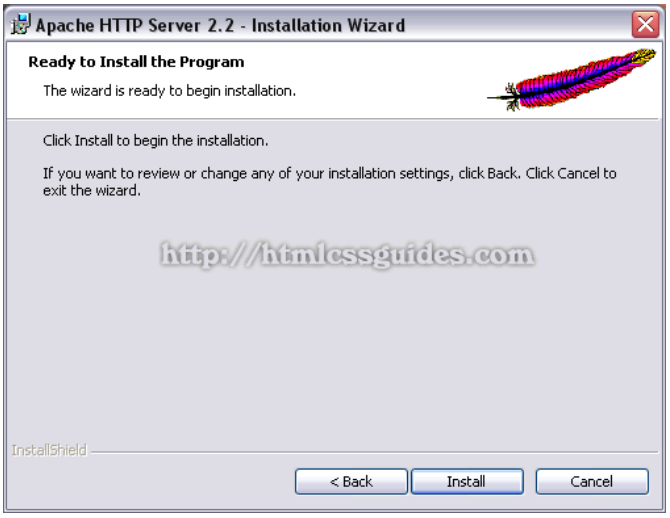
Gambar 4.54 Tampilan Proses Penggantian Folder

9. Setelah kita mengubah folder tujuan penginstalasian tersebut maka akan terlihat seperti gambar diatas.



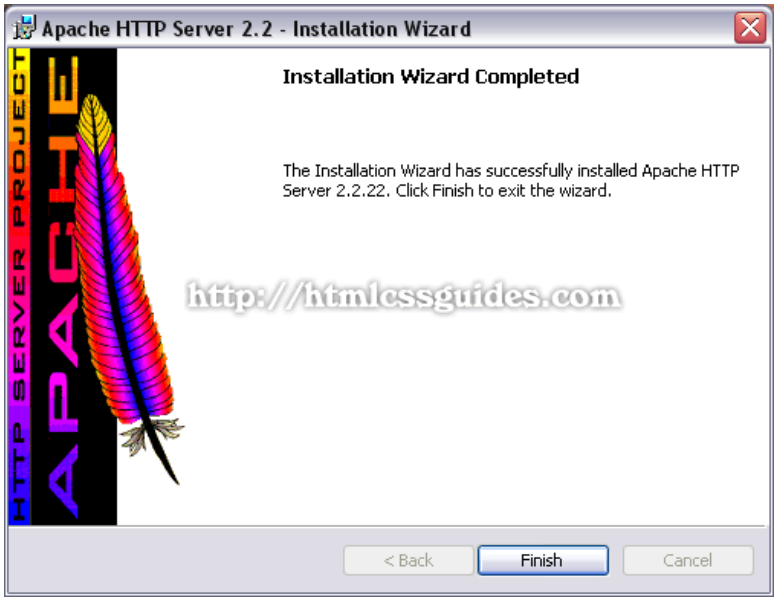
Gambar 4.55 Tampilan Penginstalan

10. Kemudian klik Install untuk menginstal Apache Server.



Gambar 4.56 Tampilan Penginstalan

11. setelah selesai klik "Finish" Apache Server siap digunakan.

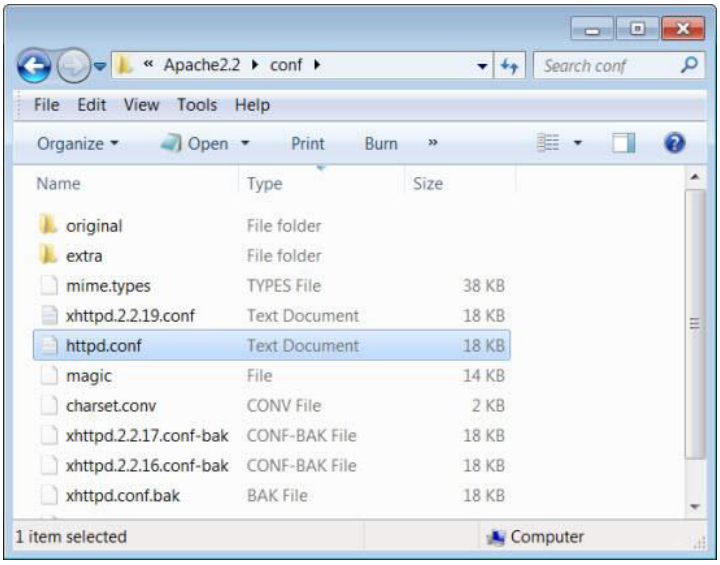


Gambar 4.57 Tampilan Selesai Penginstalan

4.5.6 Penggunaan program

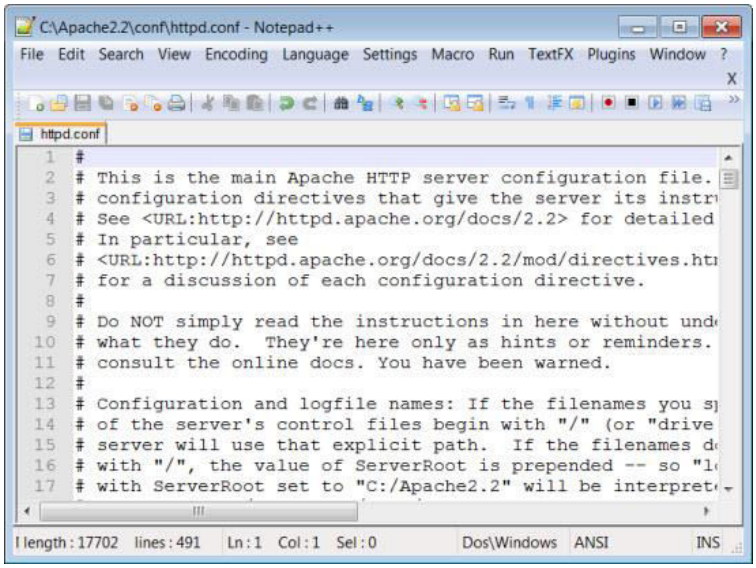
Penggunaan program adalah langkah-langkah atau cara-cara dalam menjalankan dan menggunakan sistem informasi yang digunakan oleh pengguna.

1. carilah file HTTPD.CONF yang berada di bawah direktori C:\Apache2.2\conf. Ini bila Anda melakukan proses instalasi mengikuti tutorial sebelumnya di situs ini. *Segala pengubahan konfigurasi Apache dilakukan pada file HTTPD.CONF*



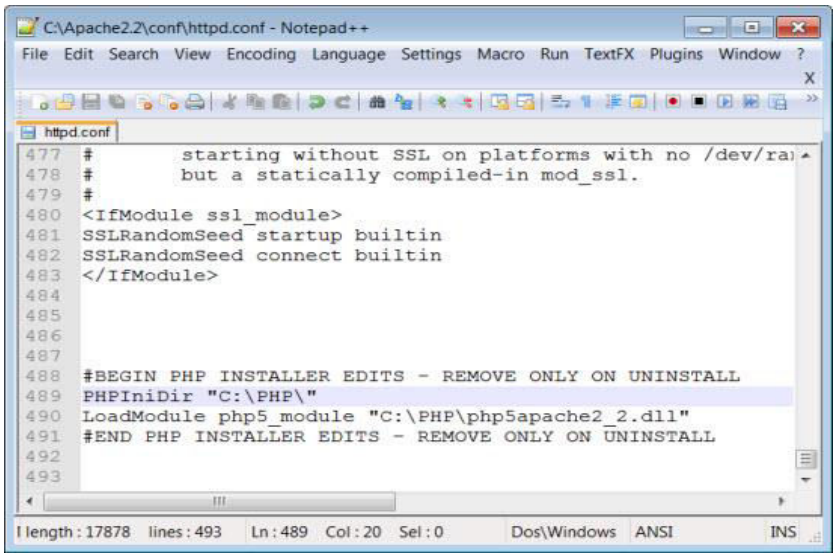
Gambar 4.58 Tampilan Konfigurasi Apache

2. editlah file HTTPD.CONF dengan menggunakan program teks editor biasa (seperti Notepad++, Notepad, Crimson Editor, dsb). Hindari menggunakan program text editor *WYSIWYG* seperti MS Word, dsb



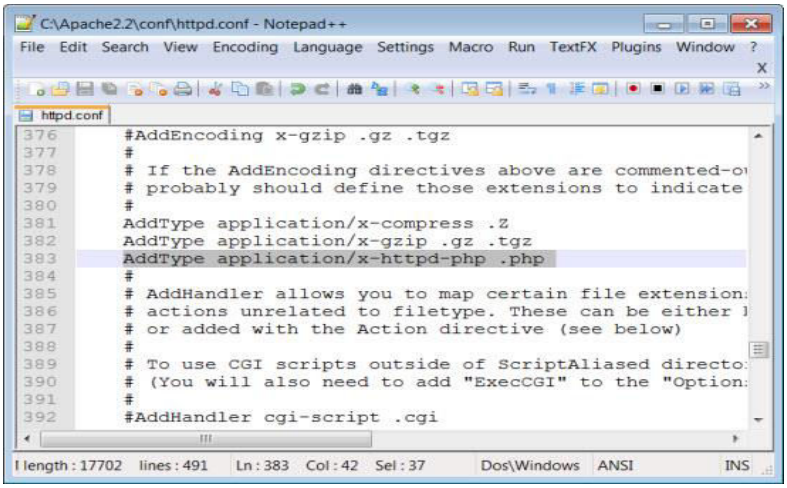
Gambar 4.59 Tampilan Edit File HTTPD.CONF

3. bila belum ada, tambahkan dua baris perintah berikut ini: LoadModule php5_module"c:/php/php5apache2_2.dll" dan PHPIniDir"C:/php" pada bagian akhir dari file HTTPD.CONF.



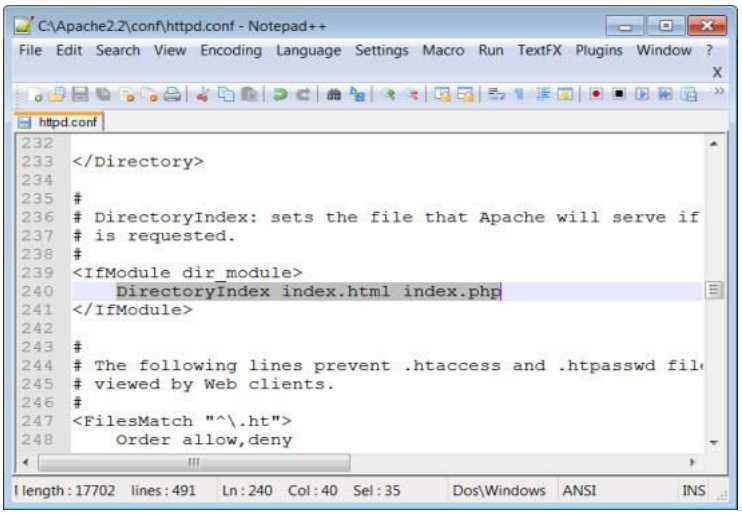
Gambar 4.60 Tampilan Edit File HTTPD.CONF

4. tambahkan baris perintah AddType application/x-httpd-php.php sehingga menjadi sebagai berikut (sekitar baris ke-384)



Gambar 4.61 Tampilan Edit File HTTPD.CONF

5. carilah baris perintah DirectoryIndex (sekitar baris nomor ke-212) berikut ini dan kemudian lengkapi menjadi DirectoryIndex index.html index.php sehingga tampilannya sebagai berikut:



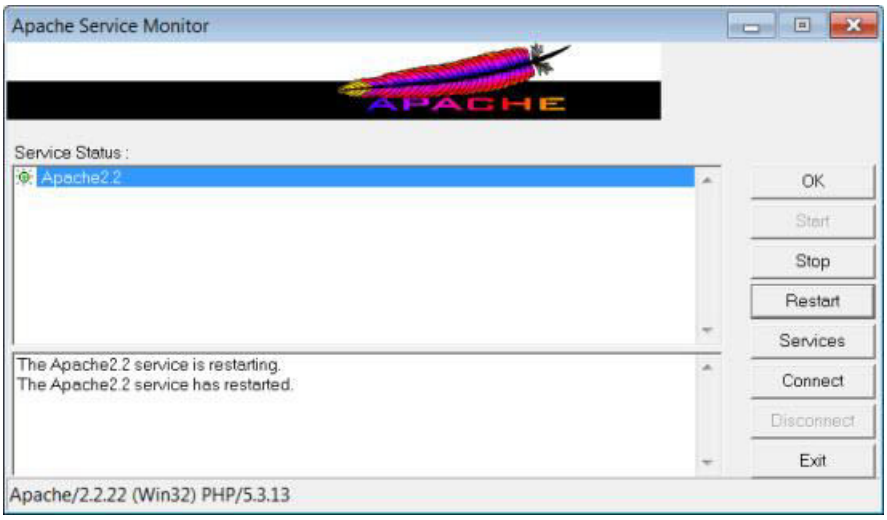
Gambar 4.62 Tampilan Edit File HTTPD.CONF

6. matikan program Apache server Anda dan kemudian jalankan kembali. Untuk mematikan Apache sever cukup dengan klik ganda pada ikon Apache Service Monitor, yang akan menampilkan Apache Service Monitor:



Gambar 4.63 Tampilan Pada Ikon Apache Service Monitor

7. klik satu kali pada tombol Restart, maka Apache akan dimatikan dan kemudian secara otomatis dijalankan kembali.



Gambar 4.64 Tampilan Apache Service Monitor