

BAB I

METODELOGI PENELITIAN

1.1. Alat dan Bahan Penelitian

Dalam penelitian dibutuhkan beberapa alat dan bahan untuk mendukung berjalannya perancangan dan implementasi *aplikasi web*.

A. Alat

Alat yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

1. Perangkat Keras (*Hardware*) *Personal Computer (PC)* atau laptop
 - a. Intel® Core™ i5 *processor*,
 - b. 8.00 GB RAM,
 - c. 64-bit *Operating System*,
 - d. Windows 10 *pro*.
 - e. *Printer* dokumen untuk mencetak laporan.
2. Perangkat Lunak (*Software*)
 - a. *Microsoft Visio 2016*
 - b. *SQL Server*
 - c. *Microsoft Visual Studio 2015*

B. Bahan

Data yang diperoleh dari penelitian ini melalui studi literatur dengan mencari referensi teori yang berhubungan dengan kasus atau permasalahan yang ditemukan. Referensi teori yang diperoleh dengan jalan penelitian studi literatur dijadikan sebagai fondasi dasar dalam mengembangkan aplikasi.

1.2. Teknik Pengumpulan Data

A. Observasi

Observasi adalah metode pengumpulan data kualitatif yang dianjurkan untuk mendapatkan data-data deskriptif. Metode pengumpulan data observasi berasal dari kata *observation* yang berarti pengamatan. Metode pengumpulan data observasi cocok digunakan untuk penelitian yang bertujuan untuk mempelajari aplikasi-aplikasi yang memiliki kemiripan fungsi dan kemudian dilakukan analisa terhadap keunggulan dan kelemahan aplikasi tersebut agar dapat dijadikan sebuah referensi.

B. Wawancara

Wawancara merupakan teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab antara peneliti dan narasumber secara lisan baik langsung maupun tidak langsung. Lexy J Moleong mendefinisikan wawancara sebagai percakapan dengan maksud tertentu.

Dalam hal ini peneliti akan menggunakan metode wawancara langsung dengan subjek informan dari *Staff* Prodi Teknologi Informasi. Peneliti menggunakan wawancara atau interview tak terstruktur yaitu wawancara yang bentuk pertanyaannya yang bersangkutan.

1.3. Gambaran Umum Sistem

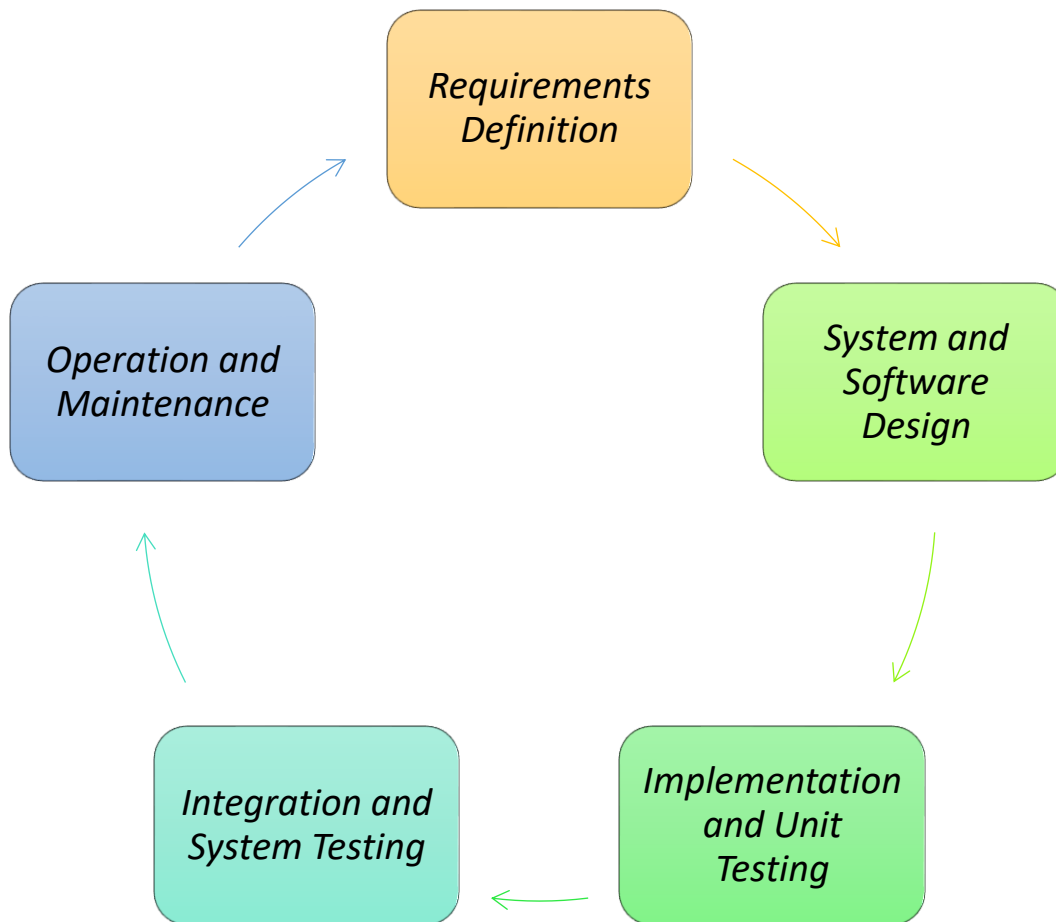
Sistem yang akan di bangun dalam tugas akhir ini adalah sebuah sistem pendataan dan pengelolaan kwitansi keuangan prodi Teknologi Informasi UMY yang berbasis *desktop*. Sistem ini bertujuan untuk membantu *staff* keuangan untuk mengefisiensi waktu pengelolaan, pendataan, dan penyimpanan data. Sistem ini digunakan hanya untuk staff keuangan yang bertanggung jawab untuk mengelola data – data keuangan prodi Teknologi Informasi.

1.4. Metodologi Penelitian

Pengembangan Aplikasi Pendataan Kwitansi Berbasis Desktop dalam studi kasus Prodi Teknologi Informasi Universitas Muhammadiyah Yogyakarta menggunakan metode *Software Development Life Cycle (SDLC)* model *Waterfall*. Metode SDLC dengan model *Waterfall* dipilih dengan alasan untuk memudahkan peneliti dalam menentukan pencapaian suatu sistem.

Model *Waterfall* dengan pendekatan SDLC merupakan awal dari peneliti dalam mengembangkan perangkat lunak yang dibuat. Hal ini juga disebut sebagai model SDLC *linear-sekuensial*. Hal ini juga mempermudah peneliti dalam memahami dan mengimplementasikannya dalam sebuah sistem.

Dalam model *Waterfall*, setiap tahapan harus berurutan dan tidak dapat meloncat ke tahap berikutnya yang artinya bahwa peneliti tidak dapat lanjut ke tahap kedua ataupun seterusnya jika belum menyelesaikan proses pada tahap pertama. Hal ini juga dijadikan sebuah pertimbangan oleh peneliti dalam membuat sistem Pendataan Kwitansi Berbasis Dekstop ini, karena sistem ini hanya dikerjakan oleh satu orang dan untuk mempermudah pembuatan sistem maka peneliti harus mengembangkan sistem dengan tahap yang berurutan juga.



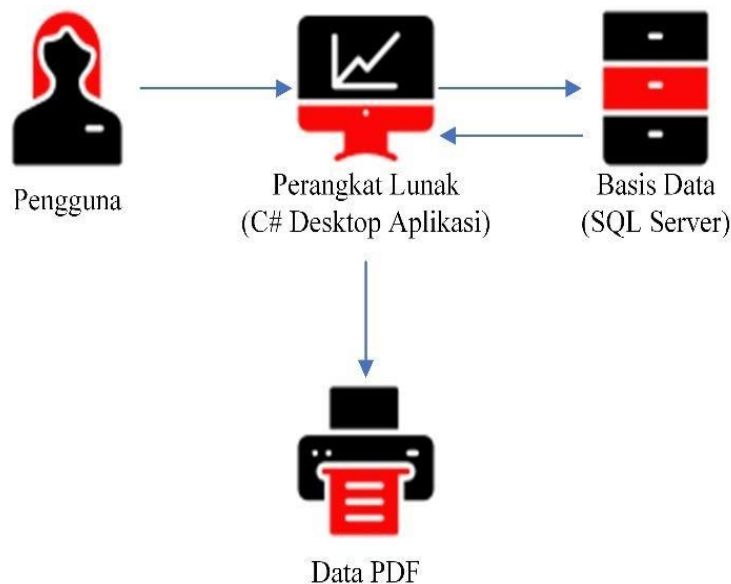
Tahapan pengembangan sistem dengan menggunakan model *Waterfall* dengan alur SDLC bisa dilihat pada diagram di atas.

1. *Requirements Definition* – merupakan tahap perencanaan sistem yang menyangkut kebutuhan pengguna, kelayakan baik secara teknik maupun secara teknologi. Setelah segala kebutuhan pengguna telah terkumpul secara lengkap lalu dilakukan proses analisis, dari proses analisis inilah dapat didefinisikan kebutuhan mana yang harus dipenuhi oleh sistem yang akan dibangun. Tahap ini harus dikerjakan secara menyeluruh dan lengkap untuk dapat menghasilkan keluaran (*output*) yang lengkap yang diinginkan pengguna.
2. *System and Software Design* – merupakan proses perancangan sistem mulai dari pembagian persyaratan dalam sistem perangkat keras maupun perangkat lunak. Pada tahap ini ditentukan arsitektur sistem secara keseluruhan. Proses perancangan melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar. Tahap ini dikerjakan setelah proses tahap sebelumnya yaitu kebutuhan pengguna telah selesai dikumpulkan secara lengkap.

3. *Implementation and Unit Testing* – merupakan proses perancangan perangkat lunak direalisasikan dengan program atau unit program. Proses pengujian ini melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.
4. *Integration and System Testing* – merupakan proses pengujian unit program atau program individual yang diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa kebutuhan sistem telah terpenuhi.
5. *Operation and Maintenance* – merupakan proses pengoperasian sistem pada lingkungannya dan melakukan pemeliharaan sistem. Tahap ini merupakan tahap siklus hidup yang lama. Pemeliharaan mencakup koreksi dari berbagai *error* yang tidak ditemukan pada tahap-tahap sebelumnya yang kemudian dilakukan perbaikan kembali atas implementasi unit sistem dan pengembangan layanan sistem serta persyaratan-persyaratan baru yang ditambahkan.

1.5. Analisis Penelitian

3.5.1. Arsitektur



Gambar 2. 1 Arsitektur Dasar

Sesuai dengan gambar 2.2 pengguna akan berinteraksi dengan perangkat lunak yang ada di dalam komputer atau laptop nya untuk menggunakan aplikasi yang akan dikembangkan. Bahasa yang di gunakan untuk membangun aplikasi ini adalah bahasa *C#(sharp)*. Sedangkan basis data yang akan digunakan untuk mendukung pembangunan aplikasi adalah *SQL Server*. Data yang ada di dalam Basis data akan di

tampilkan dalam 2 jenis , jenis yang pertama akan ditampilkan di dalam aplikasi sebagai fitur *Preview*, sedangkan jenis yang ke 2 dapat dilihat dalam format file PDF.

3.5.2. Analisis Pengguna

Aplikasi yang akan dibangun dapat digunakan oleh user yang sudah terdaftar atau yang sudah mempunyai wewenang sebagai staff keuangan. User tersebut memiliki peran untuk mengelola data, seperti menambahkan data, menghapus data, mengubah data, mencari data yang di butuhkan, dan membuat laporan dari data. Data yang di kelola adalah data kwitansi keuangan. Berikut ini merupakan hasil analisis kebutuhan pembangunan pada aplikasi ini :

1. Dapat melakukan login kedalam sistem.
2. Dapat melihat data kwitansi yang tersimpan.
3. Dapat melakukan *insert* data kwitansi .
4. Dapat melakukan *editing* data kwitansi.
5. Dapat melakukan pencarian data sesuai dengan tanggal.
6. Dapat melakukan pencarian data sesuai dengan nama penerima.
7. Dapat melakukan pencarian data sesuai dengan kode kwitansi.
8. Dapat melakukan proses pencetakan laporan kwitansi sesuai dengan format yang di ajukan.
9. Dapat melakukan *logout*.

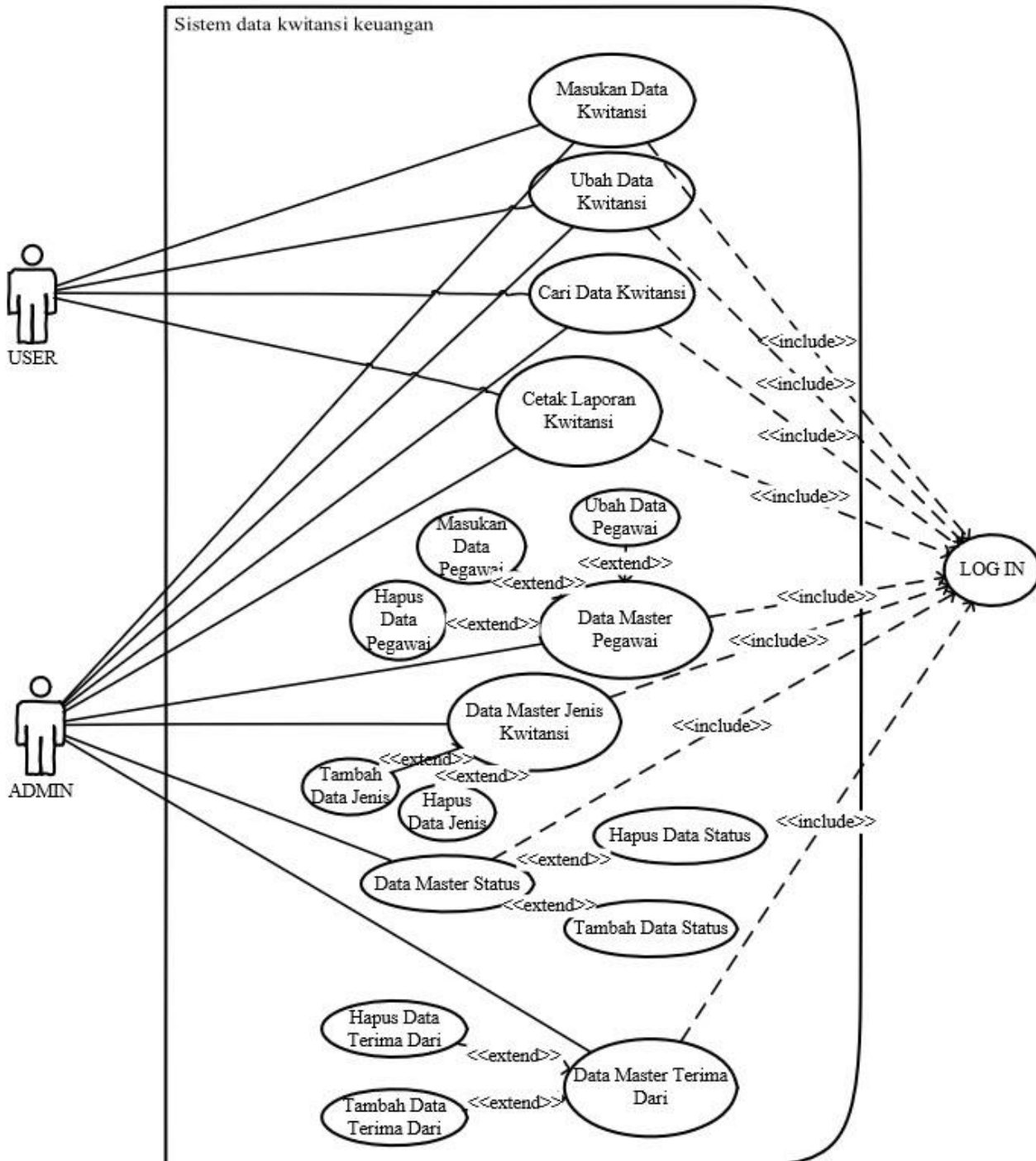
1.6. Desain Sistem

Setelah melakukan proses analisis kebutuhan yang sesuai dengan metode SDLC terhadap sistem yang akan di buat langkah selanjutnya yang harus di lakukan adalah membuat sebuah rancangan untuk sistem yang akan dibangun. Dalam melakukan pembangunan pada aplikasi ini akan ada beberapa rancangan yang di butuhkan. Rancangan yang di butuhkan adalah *use case diagram*, *ERD(entity relationship diagram)*, *activity diagram*, *class diagram*, dan *prototype*.

1.6.0. Use Case Diagram

Use case diagram yang di buat di dalam sistem ini memiliki 2 Aktor yaitu User dan Admin. Admin padaa sistem ini bertuga untuk mengelola data master yang akan digunakan untuk membuat data kwitansi. sedangkan User pada sistem ini bertugas untuk mengelola data kwitansi. Pengelolaan data yang di maksud adalah seperti

melakukan *insert* data kwitansi, melakukan *edit* data kwitansi, melihat data kwitansi, melakukan pencarian data yang di perlukan, melakukan proses pencetakan laporan kwitansi yang dibutuhkan. Untuk memperjelas penjelasan di atas dapat dilihat pada gambar 3.1.



Gambar 3. 1 Usecase Diagram

1.6.1. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram atau disingkat ERD adalah suatu model yang berfungsi untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

Rancangan ERD yang di buat untuk sistem ini memiliki 3 entitas dengan beberapa atributnya masing-masing. Entitas yang buat sebagai berikut :

1. Entitas *User*

Entitas *User* disini merupakan entitas kuat karna entitas ini tidak memiliki relasi atau ketergantungan dengan entitas lain.

2. Entitas Pegawai

Entitas Pegawai memiliki relasi dengan entitas kwitansi. Bentuk relasi antara dua entitas ini adalah *one-to-many*. Dengan detail minimal 1 maksimal banyak. Dengan dependensi entitas pegawai sebagai *parent* dari entitas kwitansi.

Entitas Pegawai juga memiliki relasi dengan entitas status. Dengan bentuk relasi antara dua entitas ini adalah *many-to-one*. Dengan detail minimal 1 maksimal 1. Dependensi entitas pegawai bergantung pada entitas Status.

3. Entitas Kwitansi

Entitas kwitansi memiliki relasi dengan entitas pegawai. Bentuk relasi antara dua entitas ini adalah *many-to-one*. Dengan detail minimal 1 maksimal 1. Dengan dependensi entitas kwitansi bergantung pada entitas pegawai.

Entitas kwitansi juga memiliki relasi antara entitas jenis dan entitas Dari. bentuk relasi antara entitas kwitansi dengan jenis adalah *many-to-one* dengan dependensi entitas kwitansi bergantung pada entitas jenis. Sama halnya dengan relasi antara entitas kwitansi dengan entitas Dari. Kedua entitas ini memiliki relasi *many-to-one* dengan dependensi entitas kwitansi bergantung pada entitas dari.

4. Entitas Jenis

Entitas Jenis memiliki Relasi dengan Entitas Kwitansi. Bentuk relasi antara dua entitas ini adalah *one-to-many*. Dengan detail minimal 1 maksimal banyak. Dengan dependensi entitas jenis sebagai *parent* dari entitas kwitansi.

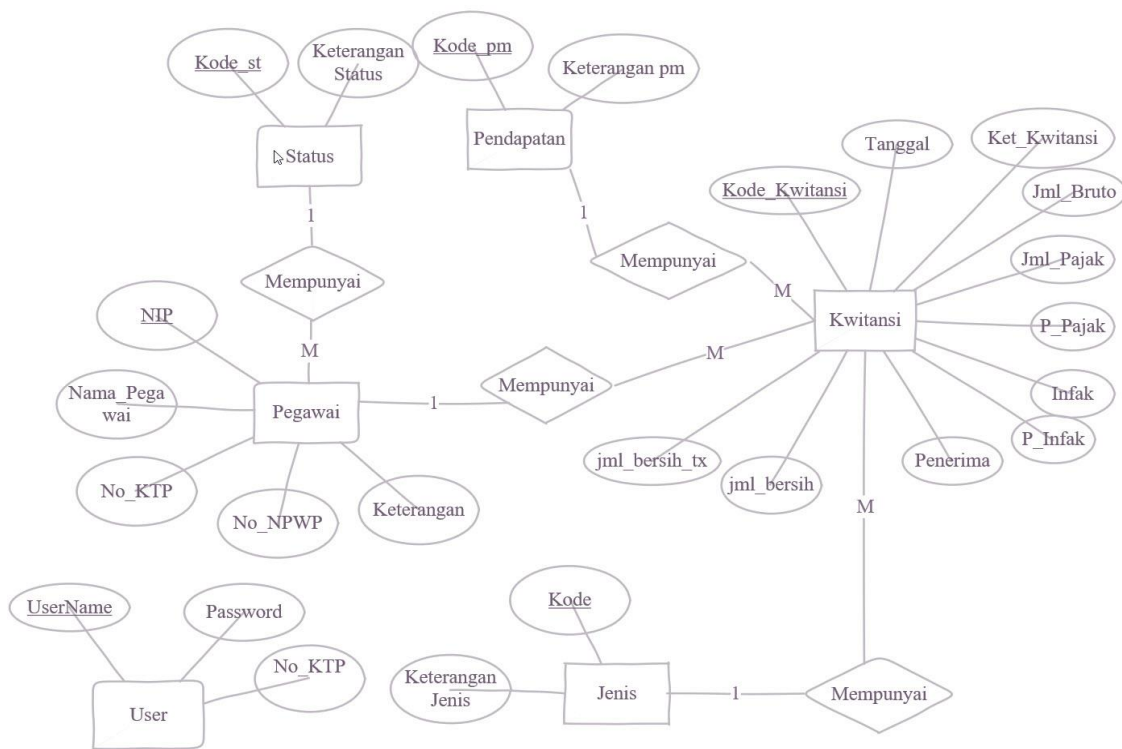
a. Entitas Status

Entitas Status memiliki jenis relasi dengan Entitas Pegawai. Bentuk relasi antara ke dua entitas tersebut adalah one-to-many. Dengan detail minimal 1 maksimal banyak. Dengan dependensi entitas status sebagai *parent* dari entitas pegawai.

b. Entitas Dari

Entitas Dari memiliki Relasi dengan Entitas Kwitansi. Bentuk relasi antara dua entitas ini adalah one-to-many. Dengan detail minimal 1 maksimal banyak. Dengan dependensi entitas dari sebagai *parent* dari entitas kwitansi.

Untuk memperjelas entitas yang telah di jelaskan sebelumnya dapat dilihat pada gambar 3.2



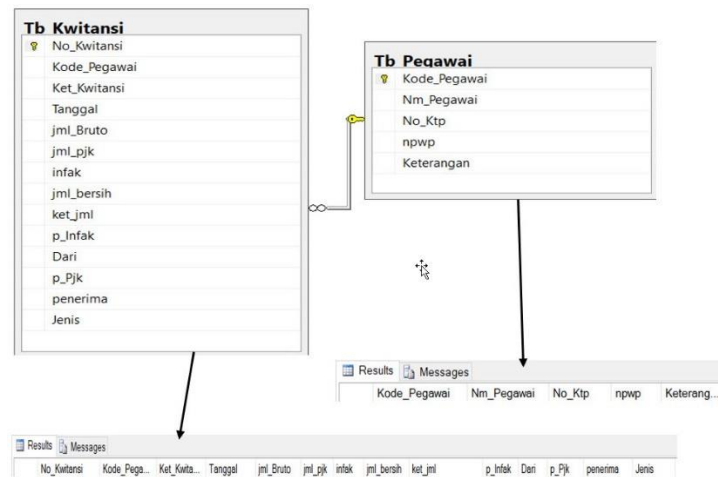
Gambar 3. 2 ER Diagram

Konversi ER Diagram ke bentuk tabel adalah proses mengkonversi entitas menjadi tabel dan *attributes* menjadi kolom beserta relasinya ke dalam setiap tabel yang saling berhubungan.

5. Konversi dengan Relasi *one to many*

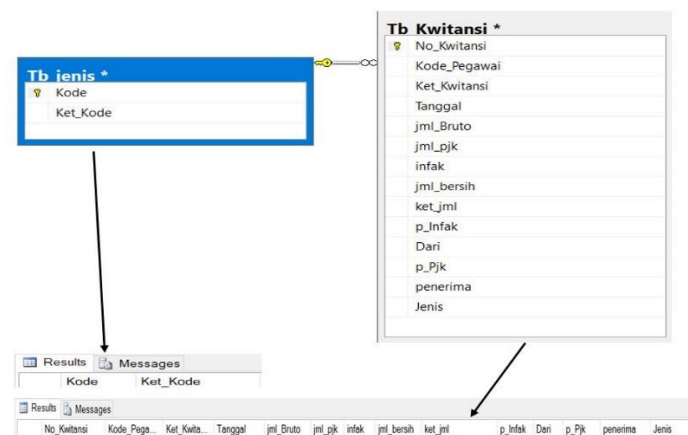
Konversi entitas yang memiliki relasi *one to many* (1-M) adalah atribut kunci pada entitas yang memiliki kardinal satu (*one*), akan menjadi *foreign key* pada entitas

yang memiliki kardinal banyak (*many*). Pada ERD yang ditunjukkan pada gambar. Pada gambar 4 . menunjukkan adanya relasi one to many antara entitas , entitas yang berhubungan mempunyai relasi ini adalah entitas Pegawai yang memiliki relasi dengan entitas Kwitansi, entitas pegawai dengan entitas status, entitas jenis dengan entitas kwitansi, dan relasi antar entitas pendapatan dengan entitas kwitansi. entitas tersebut akan direpresentasikan kedalam tabel, dapat dilihat pada gambar 3.3.



Gambar 3.3 Konversi Tabel Pegawai dan Kwitansi

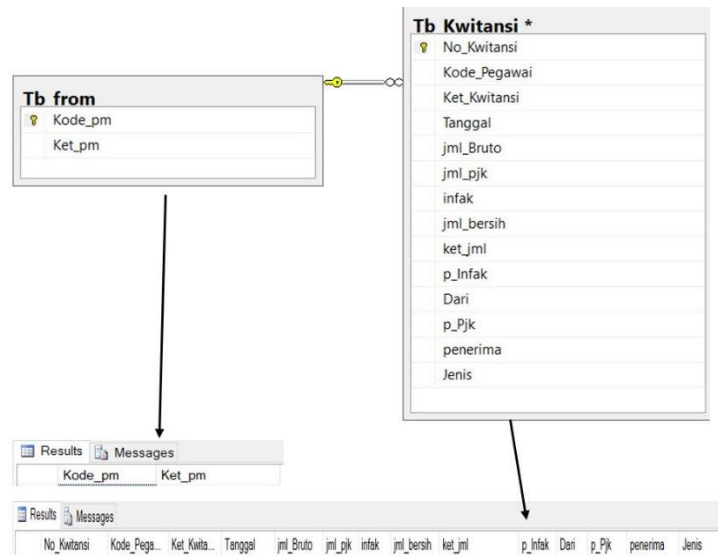
Pada gambar 3.3 entitas Pegawai memiliki relasi one to many dengan entitas Kwitansi yang berarti satu pegawai dapat memiliki lebih dari satu kwitansi. Maka dari itu kolom NIP yang terdapat pada tabel pegawai perlu ditambahkan ke dalam tabel kwitansi sebagai *foreign key* pada tabel kwitansi.



Gambar 3.4 Konversi Tabel kwitansi dan jenis

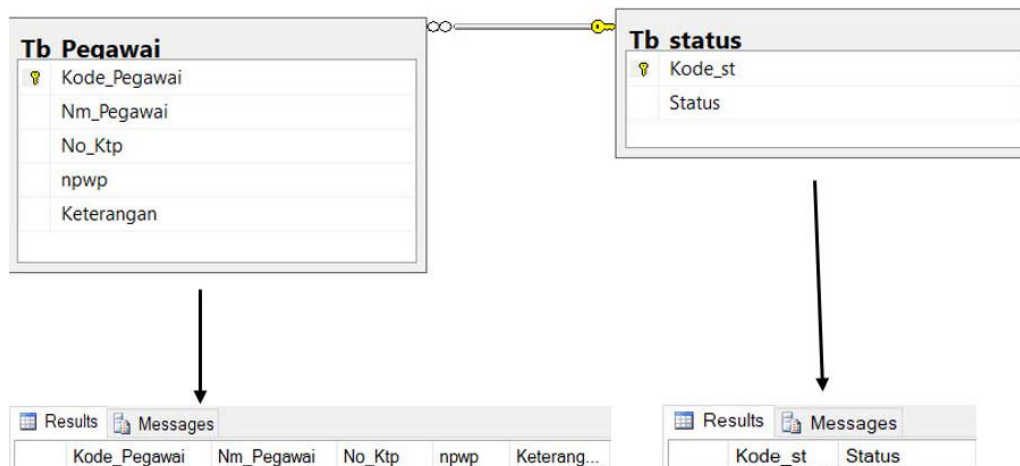
Pada gambar 3.4 entitas Jenis memiliki relasi *one to many* dengan entitas Kwitansi yang berarti satu jenis dapat memiliki lebih dari satu kwitansi. Maka dari

itu kolom Kode yang terdapat pada tabel jenis perlu ditambahkan ke dalam tabel kwitansi sebagai *foreign key* dengan nama jenis pada tabel kwitansi.



Gambar 3.5 Konversi table Kwitansi dan Dari

Pada gambar 3.5 entitas from(pendapatan) memiliki relasi one to many dengan entitas Kwitansi yang berarti satu from dapat memiliki lebih dari satu kwitansi. Maka dari itu kolom Kode_pm yang terdapat pada tabel from perlu ditambahkan ke dalam tabel kwitansi sebagai *foreign key* dengan nama kolom dari pada tabel kwitansi.



Gambar 3.6 Konversi Tabel Pegawai dan Status

Pada gambar 3.6 entitas Status memiliki relasi *one to many* dengan entitas Pegawai yang berarti satu status dapat dimiliki lebih dari satu Pegawai. Maka dari itu kolom kode_st yang terdapat pada tabel status perlu ditambahkan ke dalam tabel pegawai sebagai *foreign key* dengan nama kolom keterangan pada tabel kwitansi.

6. Konversi Entitas Kuat

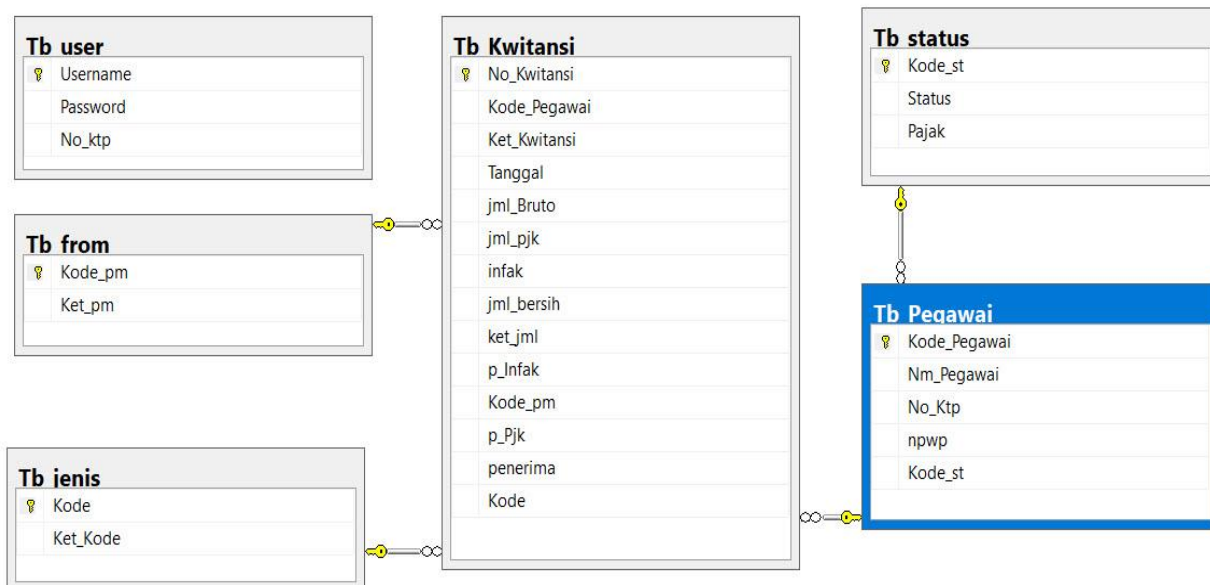
Konversi entitas kuat (*strong entity*) adalah entitas yang dicantumkan secara jelas dalam ERD dan tidak memiliki ketergantungan terhadap entitas lain.

Pada gambar 3.2 terdapat entitas kuat yaitu entitas *user*. Pada proses konversi entitas user akan diubah ke dalam bentuk table. entitas akan di konversi menjadi nama tabel dan attributes yang ada pada entitas menjadi nama kolom. Hasil dari konversi dapat dilihat pada gambar 3.7.



Gambar 3.7 Konversi Tabel User

Setelah melakukan proses konversi ERD ke dalam bentuk table maka hasil dari proses tersebut adalah sebuah RAT (relasi antar table) yang dapat dilihat pada gambar 3.8.



Gambar 3.8 RAT

- Tabel User adalah tabel yang di hasilkan dari entitas ERD sebagai entitas kuat atau entitas yang dapat berdiri sendiri.
- Tabel Pegawai adalah table yang memiliki relasi one to many dengan tabel kwitansi dan memiliki relasi many-to-one dengan table status. Penghubung antara table pegawai dengan tabel kwitansi adalah dengan menambahkan kolom yang berisi foreign key dari primary key dari tabel pegawai yang di jadikan foreign key pada tabel

kwitansi. Sedangkan penghubung antara table pegawai dengan table status adalah dengan menambahkan kolom yang berisi foreign key pada table pegawai yang di ambil dari primary key pada tabel status.

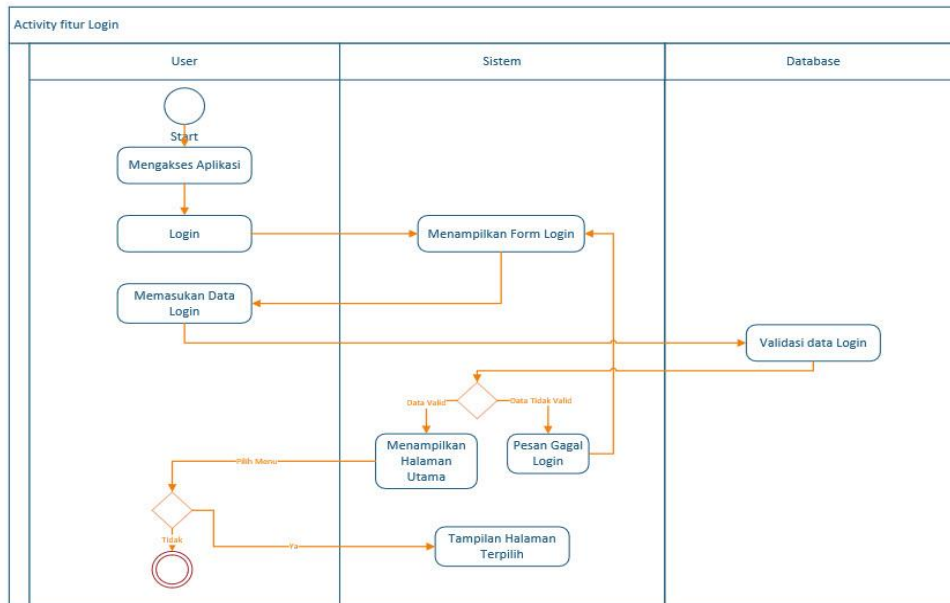
- c. Tabel Kwitansi adalah table yang memiliki relasi antra tabel jenis , tabel from(pendapatan) dan tabel pegawai. Relasi antara table kwitansi dengan ketiga table tersebut adalah many-to-one.

1.6.2. Activity Diagram

Activity diagram menjelaskan aktivitas-aktivitas yang dilakukan secara sistematis, yang sedang dirancang dan bagaimana masing-masing aliran berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir

1. Activity Diagram Login

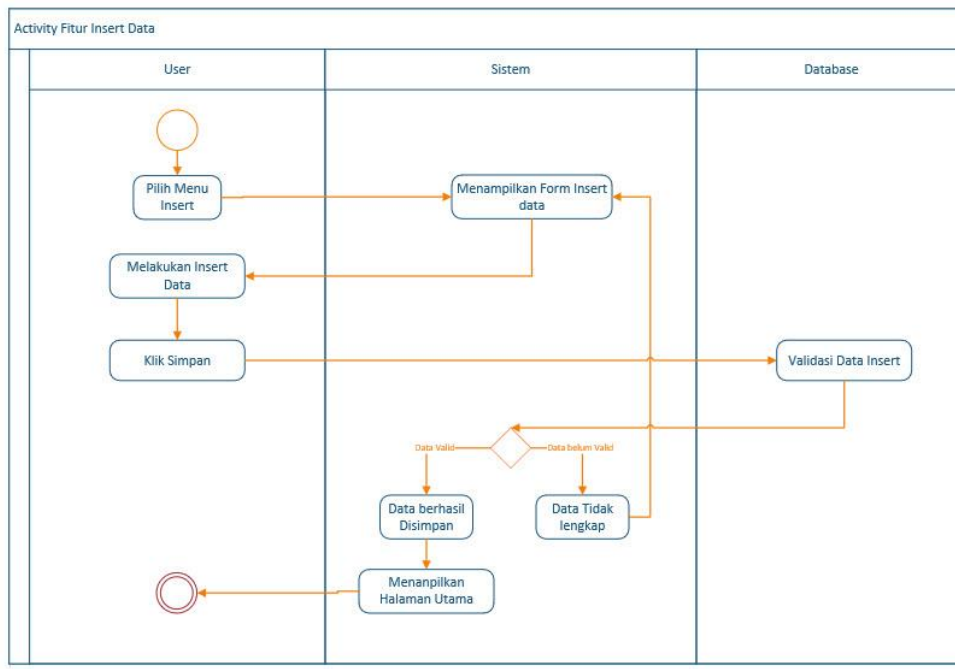
Aktor utama dalam diagram *activity* ini adalah user , user ini adalah *staff* keuangan yang bertanggung jawab untuk mengolah data kwitansi. Aktivitas yang pertama kali dilakukan setelah *user* membuka aplikasi adalah menampilkan halaman *login*. Setelah halaman *login* muncul user harus memasukkan data *username* dan *password*. Setelah memasukkan data, database akan melakukan validasi data. Jika data masukan tersebut tidak valid dengan data yang ada di database maka akan memunculkan pesan bahwa *login* gagal. Jika data yang di masukan oleh *user* *valid* dengan data yang ada di database maka sistem akan menampilkan halaman home dari aplikasi tersebut. Dalam tampilan *home* terdapat beberapa pilihan menu yang dapat dipilih oleh *user*. Untuk memperjelas penjelasan di atas dapat dilihat pada gambar 3.9.



Gambar 3. 9 Activity diagram login

2. Activity Diagram Insert

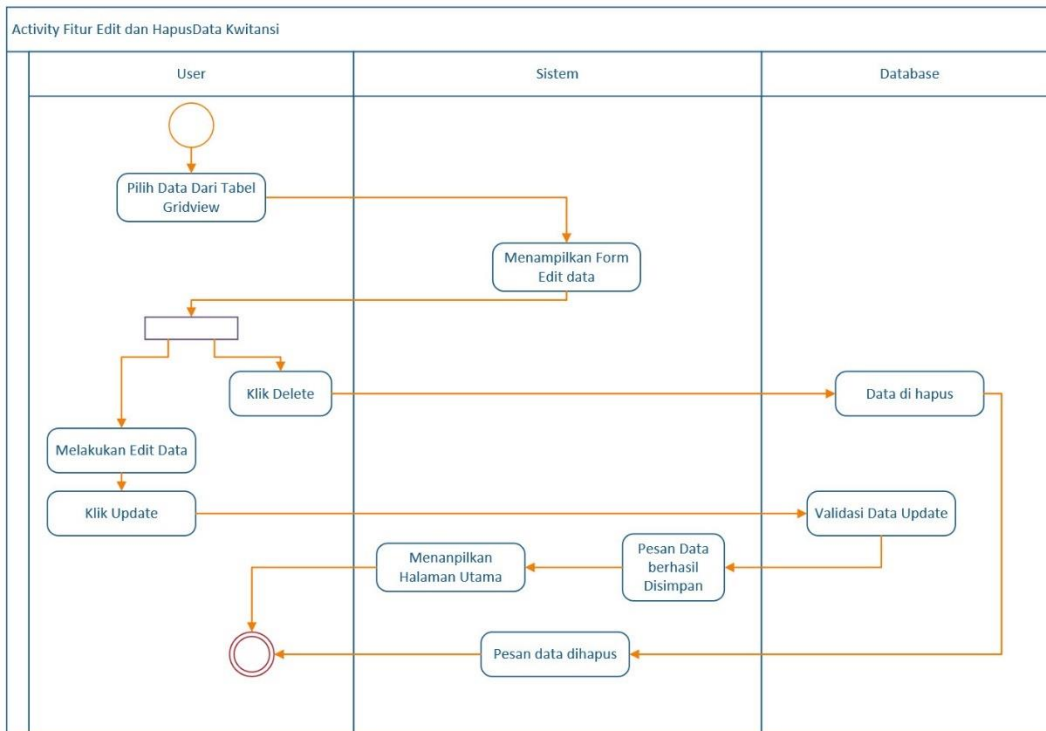
Pada halaman *home* terdapat menu *insert* data. *Insert* data dilakukan setelah user memilih menu insert data. Sistem akan menampilkan halaman yang berisi form input data. Setelah halaman tersebut tampil user di minta untuk mengisi form tersebut sesuai dengan ketentuan yang terdapat dalam form tersebut. Setelah user mengisi form tersebut, user harus mengklik tombol simpan untuk menyimpan data yang sudah di isi. Setelah itu data di masukan ke dalam databse. Jika data yang dimasukan tidak sesuai dengan ketentuan yang ada di dalam database maka sistem akan memunculkan pesan bahwa data yang anda masukan belum lengkap. Jika data yang di masukan sudah sesuai dengan ketentuan yang ada maka sistem akan memunculkan pesan bahwa data berhasil disimpan. Setelah berhasil menyimpan data sistem akan mengembalikan tampilan pada menu utama(*home*). Untuk memperjelas penjelasan diatas dapat dilihat pada gambar 3.10.



Gambar 3. 10 Activity diagram insert kwitansi

3. Activity Diagram Edit dan Delete

Pada tampilan *home* terdapat menu untuk mengubah data. Menu untuk mengubah data akan aktif jika user memilih data dari *data gridview* terlebih dahulu. Setelah memilih data maka sistem akan menampilkan halaman dengan *form* yang sudah terisi dengan data yang dipilih. Setelah melakukan perubahan pada data maka *user* dapat mengubah data. Apabila *user* ingin menghapus data maka *user* dapat mengklik tombol hapus dan data berhasil di hapus. Untuk memperjelas penjelasan di atas dapat dilihat pada gambar 3.11.



Gambar 3. 11 Activity diagram edit delete kwitansi

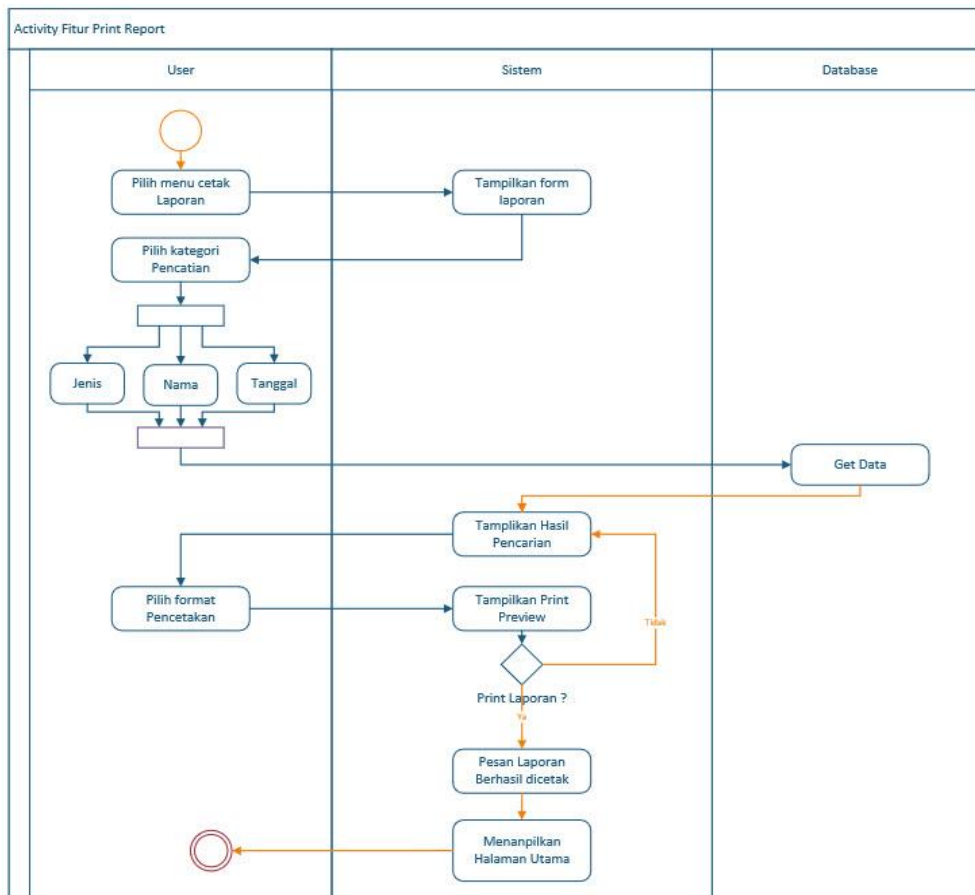
4. Activity Diagram Report

Halaman report akan muncul pada saat user memilih menu *report* di halaman utama(*home*). Setelah user memilih menu *report* maka sistem akan menampilkan halaman *report* yang berisi form untuk memilih data yang ingin di buat laporan. Pada halam *report* *user* dapat mencari data sesuai dengan ketegori pencarian seperti , berdasarkan nama , no.kwitansi, dan tanggal. Setelah *user* melakukan pencarian, *database* akan mengambil data sesuai dengan masukan *user*. Sistem akan menampilkan hasil pencarian di dalam sebuah *data gridview*. Pada halaman *report* terdapat 2 kategori format *report*. *User* dapat memilih salah satu format nya. Jika *user* sudah memilih

format pencetakan maka sistem akan memunculkan *print preview* sesuai dengan format yang dipilih. Untuk memperjelas penjelasan diatas dapat dilihat pada gambar 3.12.

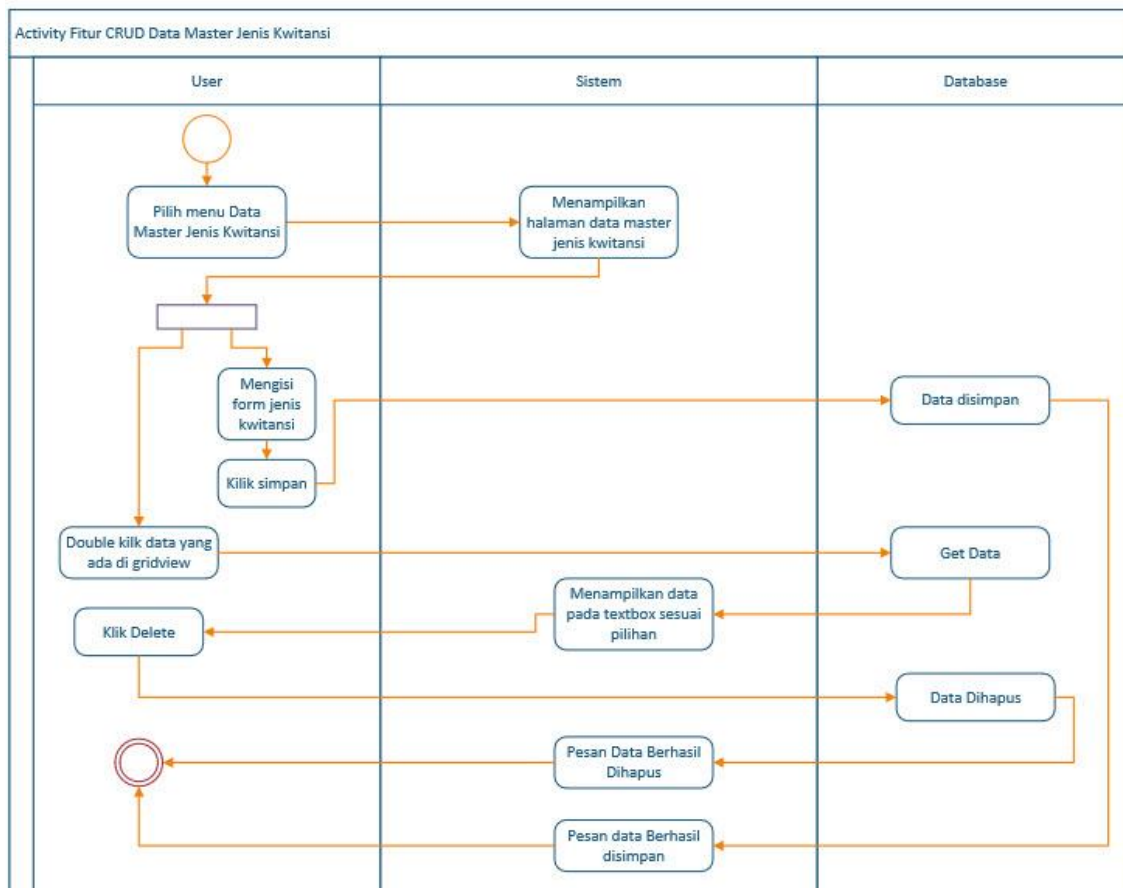
5. Activity Diagram Data Master Jenis

Pada tampilan *home* terdapat menu data master. Dalam menu data master terdapat pilihan untuk melakukan input data master salah satunya adalah data master jenis. Data master jenis ini adalah kumpulan data untuk jenis-jenis kwitansi. Pada



Gambar 3. 12 Activity diagram report

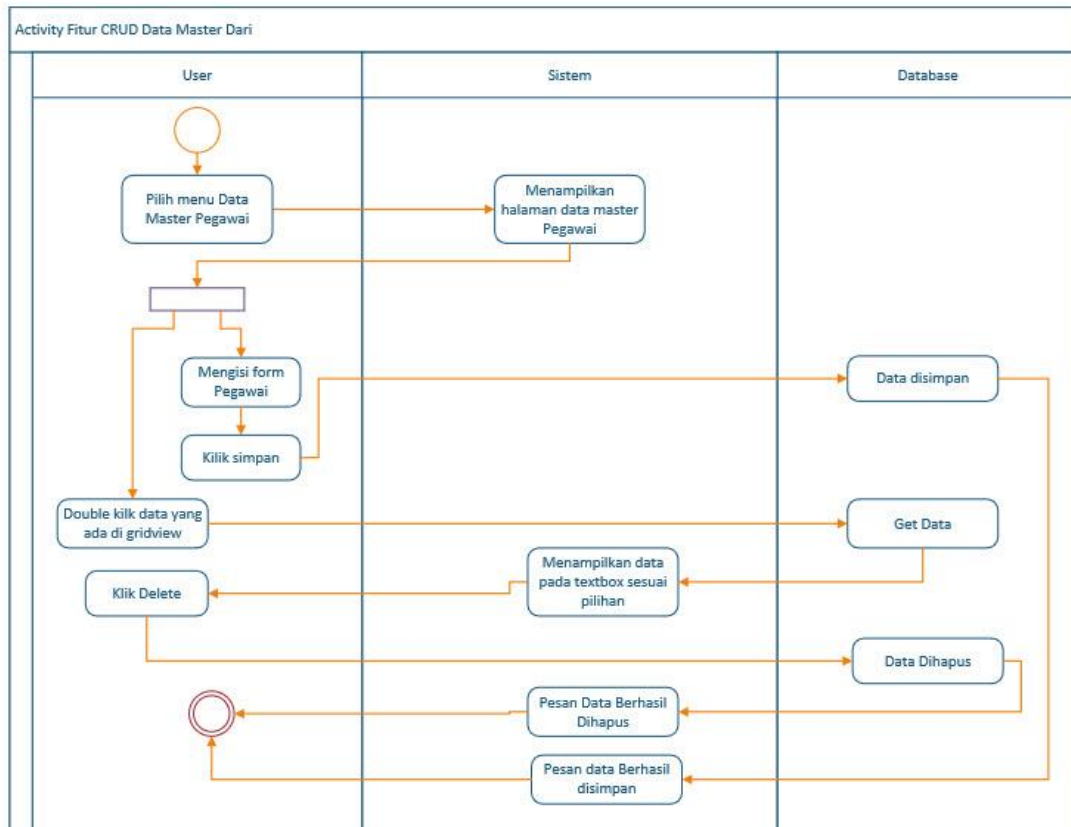
halaman jenis kwitansi user akan melakukan input kode jenis dan keterangan jenis. Pada halaman ini user hanya dapat menambah jenis dan menghapus data jenis kwitansi. Untuk memperjelas penjelasan di atas dapat di lihat pada gambar 3.13.



Gambar 3. 13 Activity diagram data master jenis

6. Activity Diagram Data Master Dari

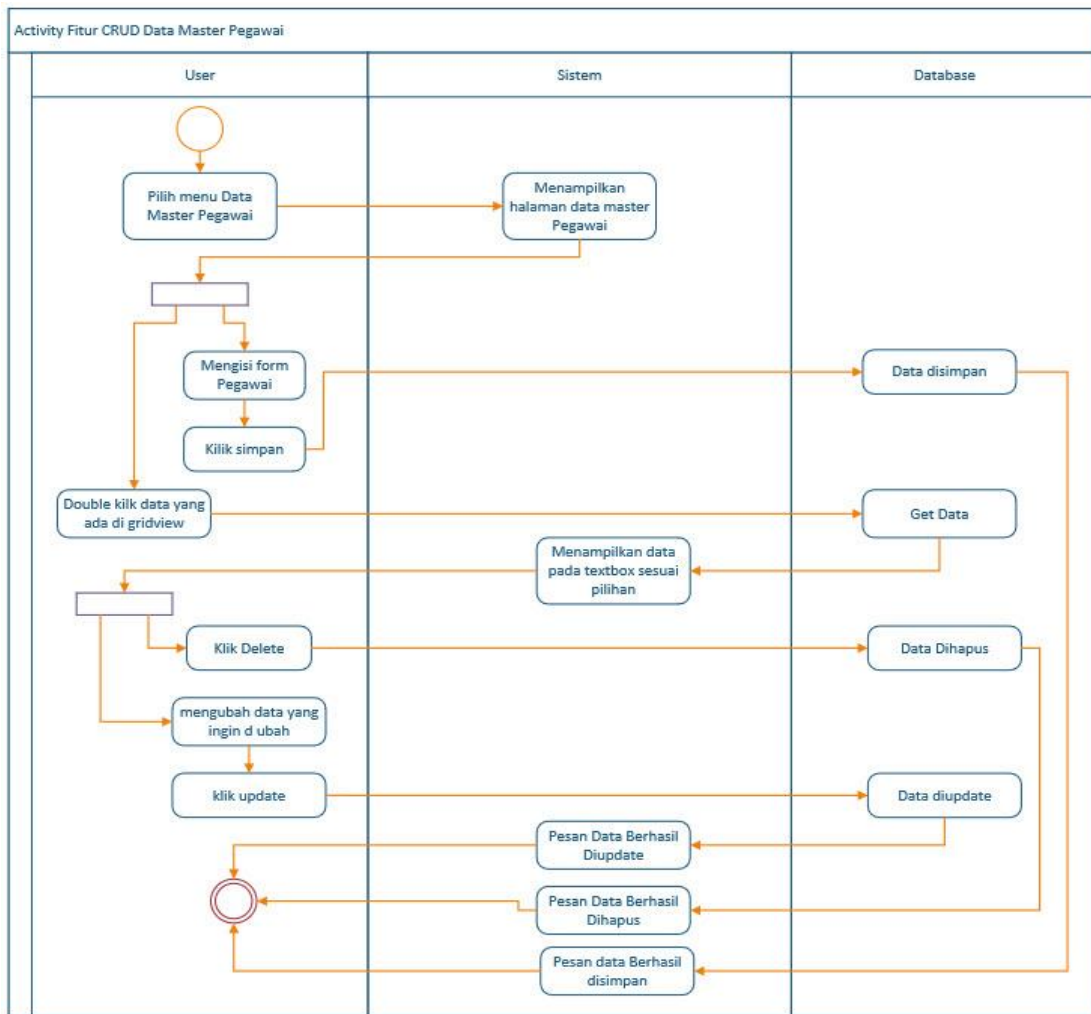
Pada tampilan home terdapat menu data master. Dalam menu data master terdapat pilihan untuk melakukan input data master salah satunya adalah data master Dari. Data master dari ini adalah kumpulan data untuk mengetahui dari mana kwitansi ini di berikan. Pada halaman data master dari, user akan melakukan input kode pemberi dan keterangan pemberi. Pada halaman ini user hanya dapat menambah pemberi dan menghapus data pemberi kwitansi. Untuk memperjelas penjelasan di atas dapat di lihat pada gambar 3.14.



Gambar 3. 14 Activity diagram data master dari

7. Activity Diagram Data Master Pegawai

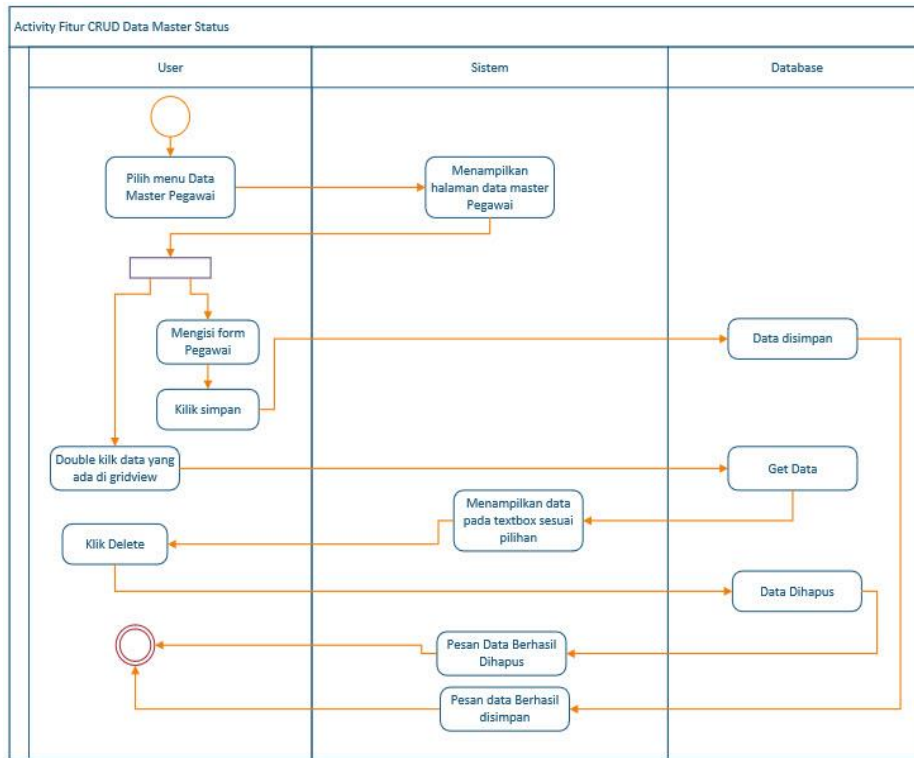
Pada tampilan home terdapat menu data master. Dalam menu data master terdapat pilihan untuk melakukan input data master salah satunya adalah data master Pegawai. Data master pegawai ini adalah kumpulan data untuk para pegawai yang akan menerima kwitansi. Pada halaman Pegawai, user akan melakukan input beberapa data yang berhubungan dengan pegawai tersebut. Pada halaman ini user hanya dapat menambah data pegawai, mengubah data pegawai dan menghapus data pegawai. Untuk memperjelas penjelasan di atas dapat di lihat pada gambar 3.15.



Gambar 3. 15 Activity diagram data master pegawai

8. Activity Diagram Data Master Status

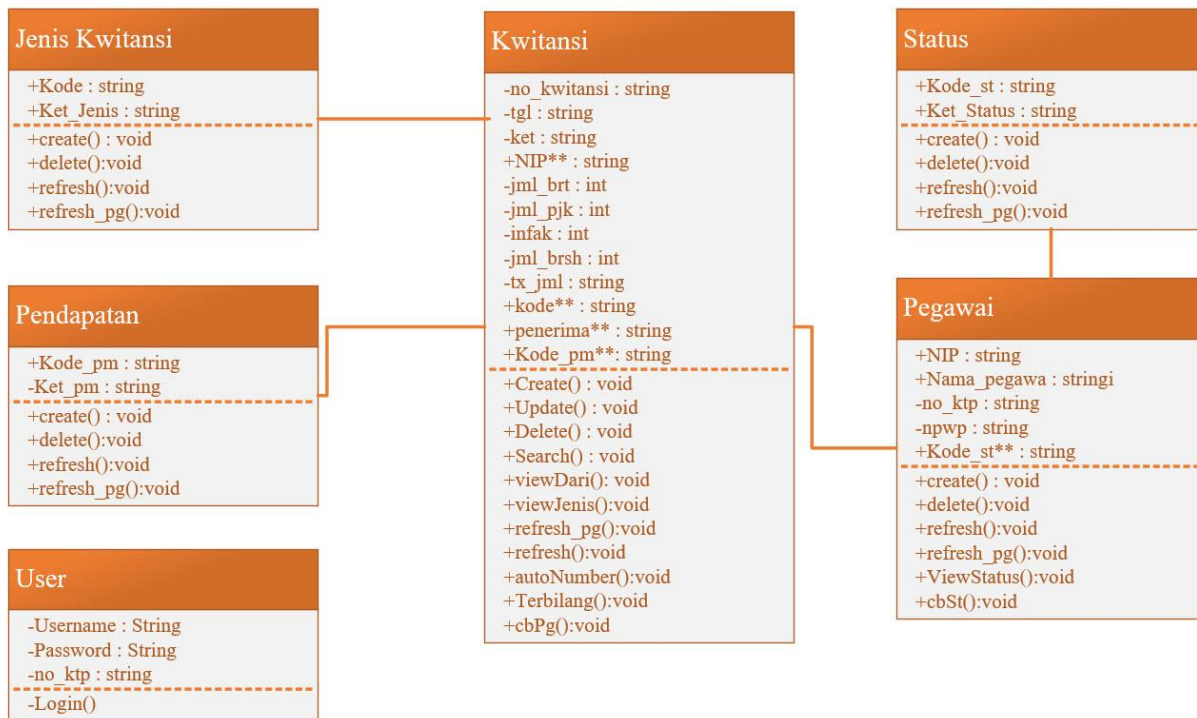
Pada tampilan home terdapat menu data master. Dalam menu data master terdapat pilihan untuk melakukan input data master salah satunya adalah data master Status. Data master status ini adalah kumpulan data untuk mendata status yang di miliki oleh penerima kwitansi. Pada halaman status, user akan melakukan input kode status dan keterangan status. Pada halaman ini user hanya dapat menambah data status dan menghapus data status. Untuk memperjelas penjelasan di atas dapat di lihat pada gambar 3.16.



Gambar 3. 16 Activity diagram data master status

1.6.3. Class Diagram

Class diagram atau diagram kelas merupakan inti pemrograman berbasis objek karena diagram ini memberikan pemetaan terhadap kelas-kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang digunakan oleh suatu aplikasi. Berikut ini adalah rancangan class diagram yang di buat untuk membangun aplikasi pendataan kwitansi.



Gambar 3. 17 Class diagram

Pada gambar 3.17 terdapat 6 class yaitu *class user*, *class status*, *class jenis*, *class* telah terima dari(from), *class pegawai* dan *class kwitansi*. Class user mempunyai fungsi untuk menyimpan data user. User yang di maksud adalah user yang dapat masuk kedalam aplikasi yang sudah di buat. Class kwitansi merupakan class yang berfungsi untuk mengelola data kwitansi dan menyimpan seluruh data kwitansi. Pengelolaan yang dimaksud yaitu menambah data, mengubah data, menghapus data, dan mencari data.

Class kwitansi memiliki hubungan dengan class jenis. hubungan antara kelas tersebut adalah data kode yang terdapat di dalam class jenis akan di gunakan untuk inisiaisasi untuk class kwitansi sebagai no_Kwitansi.

Class Kwitansi memiliki hubungan Class Pendapatan. Hubungan antara kelas tersebut adalah data kode dari kelas pendapatan akan di gunakan untuk mengisi kolom penerima yang terdapat pada kelas kwitansi.

Class Kwitansi memiliki hubungan dengan class Pegawai. Hubungan antara kelas tersebut adalah data NIP yang ada pada kelas pegawai akan di gunakan untuk mengisi kolom NIP yang terdapat pada kelas Kwitansi.

Sedangkan class pegawai memiliki relasi dengan class status. Hubungan antara kedua kelas tersebut adalah kode status yang ada pada kelas status akan digunakan untuk mengisi kolom keterangan yang terdapat pada kelas Pegawai.

Penjelasan *class diagram* berdasarkan relasi antar kelas :

1. Class kwitansi dengan class jenis menggunakan relasi *one-to-many* dengan simbol relasi asosiasi berarah (*directed association*).
2. Class Kwitansi dengan Class Pendapatan menggunakan relasi *one-to-many* dengan simbol relasi asosiasi berarah (*directed association*).
3. Class Kwitansi dengan class Pegawai menggunakan relasi *one-to-many* dengan simbol relasi asosiasi berarah (*directed association*).
4. class pegawai dengan class status menggunakan relasi *one-to-many* dengan simbol relasi asosiasi berarah (*directed association*).

1.6.4. Kamus Database

Berikut ini merupakan tabel – tabel dari kamus database pada “Aplikasi pendataan kuitansi berbasis dekstop”.

1. Kamus Tabel User

Kamus tabel user yang digunakan dapat dilihat pada tabel 3.1.

Tabel 3. 1 Kamus Tabel User

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Username	Varchar	50	Primary Key	ID User
Password	Varchar	6	-	Password User
No_ktp	Varchar	16	-	Data User

2. Kamus Tabel Pegawai

Kamus tabel pegawai yang digunakan dapat dilihat pada tabel 3.2.

Tabel 3. 2 Kamus Tabel Pegawai

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Kode_Pegawai	Varchar	11	Primary Key	NIP
Nm_Pegawai	Varchar	100	-	Nama Pegawai
No_ktp	Varchar	16	-	Nomor identitas
NPWP	Varchar	16	-	Nomor NPWP
Pajak	Int		-	Pajak dalam %
Keterangan	Varchar	100	-	Status Pegawai

3. Kamus Tabel Kwitansi

Kamus table kwitansi yang digunakan dapat dilihat pada tabel 3.3.

Tabel 3. 3 Kamus Tabel Kwitansi

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
No_Kwitansi	Varchar	11	Primary key	ID kwitansi
Kode_Pegawai	Varchar	11	Foreign key	ID pegawai
Ket_kwitansi	Varchar	50	-	Keterangan kwitansi
Tanggal	Date	-	-	Tanggal kwitansi
Jml_bruto	Money	-	-	Jumlah Bruto
Jml_pajak	Money	-	-	Jumlah pajak Pegawai
Infak	Money	-	-	Jumlah infak pegawai
Ket_jml	Varchar	225	-	Jumlah bersih dalam bentuk teks
Jml_bersih	Int	-	-	Jumlah bersih dari perhitungan
P_infak	Varchar	3	-	Infak dalam %

4. Kamus Tabel Status

Kamus tabel status yang digunakan dapat dilihat pada tabel 3.4.

Tabel 3. 4 Kamus Tabek Status

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Kode_st	Varchar	10	Primary Key	ID
Status	Varchar	50	-	Keterangan

5. Kamus Tabel Jenis

Kamus tabel status yang digunakan dapat dilihat pada tabel 3.5.

Tabel 3. 5 Kamus Tabel Jenis

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Kode	Varchar	10	Primary Key	ID
Ket_Kode	Varchar	100	-	Keterangan

6. Kamus Tabel From

Kamus tabel status yang digunakan dapat dilihat pada tabel 3.6.

Tabel 3. 6 Kamus Tabel From

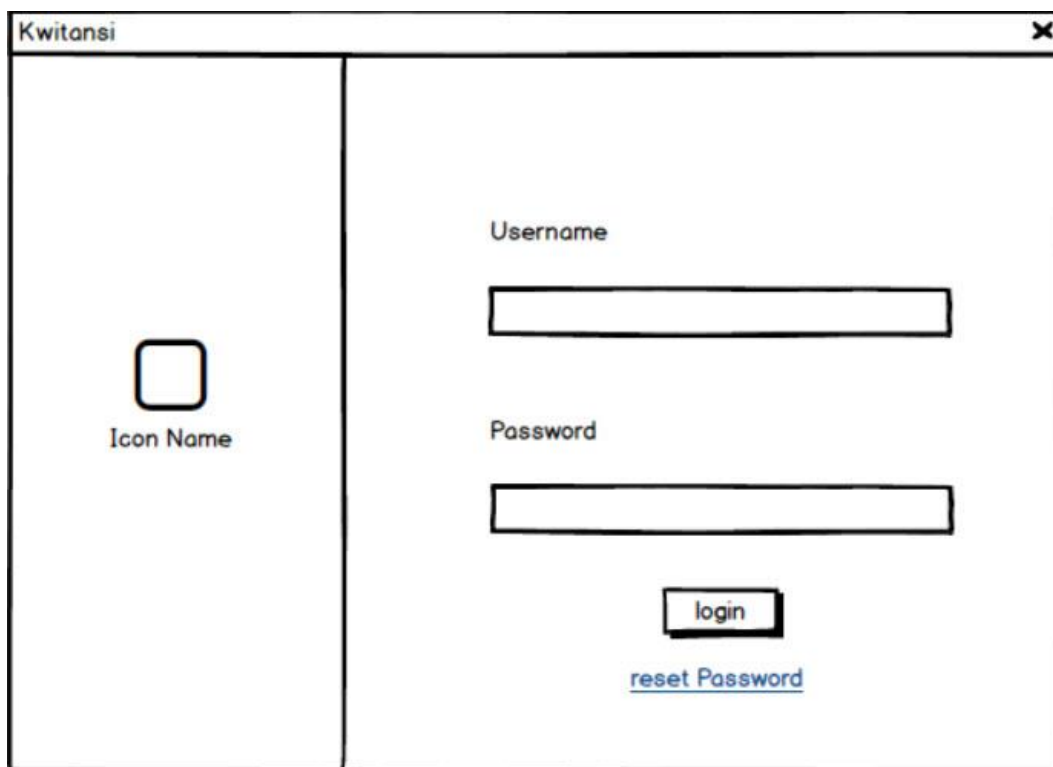
Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Kode_pm	Varchar	20	Primary Key	ID
Ket_pm	Varchar	100	-	Keterangan

1.6.5. Rancangan Antarmuka

Rancangan antarmuka bagian terpenting pada suatu aplikasi, karena bagian ini yang menjembatani antara pengguna dengan sistem. Desain user interface yang baik akan memudahkan pengguna dalam menggunakan aplikasi yang dibuat.

Dalam aplikasi ini rancangan antar muka di buat dengan beberapa halaman. Rancangan halaman yang di buat adalah sebagai berikut :

1. Rancangan Halaman Login

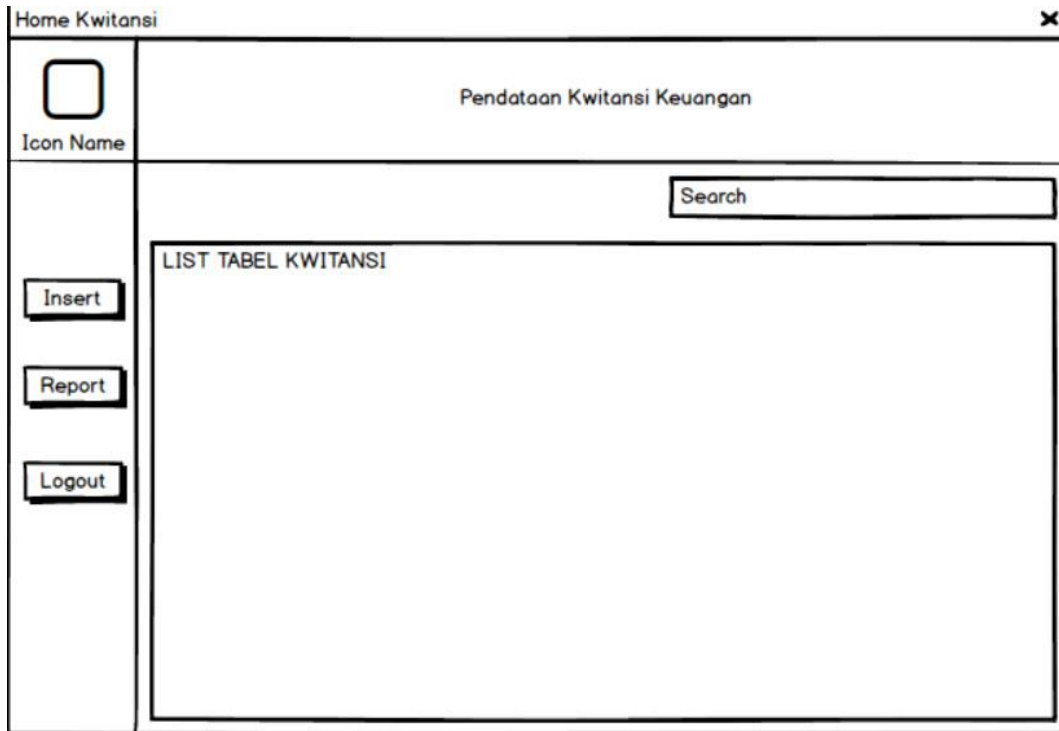


The image shows a login page titled "Kwitansi" in the top-left corner. The page is divided into two main sections by a vertical line. On the left side, there is a square icon placeholder labeled "Icon Name". On the right side, there are two text input fields: the top one is labeled "Username" and the bottom one is labeled "Password". Below the password field, there is a button labeled "login" and a link labeled "reset Password" in blue text. A close button (X) is located in the top-right corner of the window.

Gambar 3. 18 UI login

Rancangan yang dibuat untuk halaman login adalah seperti gambar 3.18. Terdapat 2 textbox yang di gunakan untuk mengisi input username dan password user. Dan terdapat tombol login untuk dapat melakukan login dan pindah ke halaman utama

2. Rancangan Halaman Home



Gambar 3. 19 UI Home

Rancangan untuk Halaman home dapat dilihat pada gambar 3.19. Di halaman tersebut terdapat beberapa tombol untuk mengakses fitur-fitur aplikasi seperti memasukkan data , mengubah data, menghapus data, membuat report, dan keluar dari aplikasi. Pada halaman ini juga terdapat satu textbox yang di gunakan untuk melakukan pencarian data. Selain itu pada halaman ini juga terdapat sebuah data gridview untuk menampilkan data kwitansi yang sudah tersimpan

3. Rancangan Halaman Insert Kwitansi

The image shows a window titled "Form Insert" with a close button (X) in the top right corner. The form is organized into two columns of input fields. The left column includes: "no. kwitansi" (text box), "tanggal" (text box with slashes and a calendar icon), "keterangan" (text box), "NIP" (text box with a dropdown arrow), "NPWP" (text box), "NoKTP" (text box), "keterangan" (text box), and "pajak" (text box). The right column includes: "jumlah bruto" (text box), "jumlah pajak" (text box), "infak" (text box with a dropdown arrow), and "jumlah bersih" (text box). At the bottom left is a "back" button, and at the bottom right is a "save" button.

Gambar 3. 20 UI Insert Kwitansi

Rancangan untuk halaman insert data dapat dilihat pada gambar 3.20. Pada halaman insert data terdapat beberapa textbox, combobox, button, dan satu datetime picker. Pada halaman ini user akan mengisi data sesuai dengan ketentuan yang ada pada halaman tersebut.

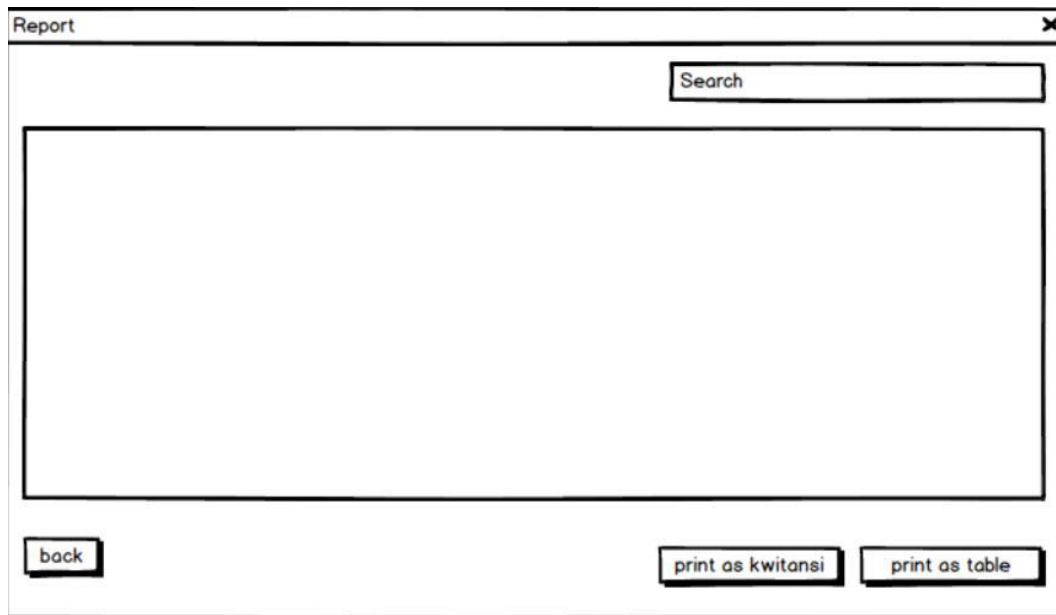
4. Rancangan Halaman Edit dan Hapus Kwitansi

The image shows a window titled "Form Edit" with a close button in the top right corner. The form is divided into two columns of input fields. The left column contains: "no. kwitansi" (text box), "tanggal" (text box with slashes and a calendar icon), "keterangan" (text box), "NIP" (text box with a dropdown arrow), an empty text box, "NPWP" (text box), "NoKTP" (text box), "keterangan" (text box), and "pajak" (text box). The right column contains: "jumlah bruto" (text box), "jumlah pajak" (text box), "infak" (text box with a dropdown arrow), "jumlah bersih" (text box with a mouse cursor), and another empty text box. At the bottom left is a "back" button and at the bottom right is an "Update" button.

Gambar 3. 21 UI Edi dan Hapus Kwitansi

Rancangan untuk halaman Edit data dapat dilihat pada gambar 3.21. Pada halaman edit data terdapat beberapa textbox, combobox, button, dan satu datetimepicker. Pada halaman ini user akan mengisi data sesuai dengan ketentuan yang ada pada halaman tersebut.

5. Rancangan Halaman Report



Gambar 3. 22 UI Report

Rancangan untuk halaman report dapat dilihat pada gambar 3.22 di atas. Pada halaman report akan ada satu textbox yang akan digunakan untuk melakukan pencarian data. Terdapat juga satu data gridview yang akan berfungsi untuk menampilkan data hasil pencarian. Pada halaman ini juga terdapat dua button yang di buat untuk menentukan format pencetakan laporan.