

BAB III METODOLOGI PENELITIAN

3.1. Alat dan Bahan

Dalam pembuatan aplikasi digunakan beberapa alat dan bahan untuk mendukung berjalannya perancangan dan pembuatan aplikasi.

3.1.1. Alat

Alat yang digunakan untuk membuat aplikasi berupa perangkat keras dan perangkat lunak.

a. Perangkat keras

1. Personal Computer (Laptop)

64bit architecture processor,

4 GB Random Access Memory (RAM),

Sistem Operasi Windows 8.

2. Printer Dokumen untuk mencetak laporan.

b. Perangkat Lunak

1. NetBeans IDE 8.2.

2. MySQL Server

3. IReport 5.6.0

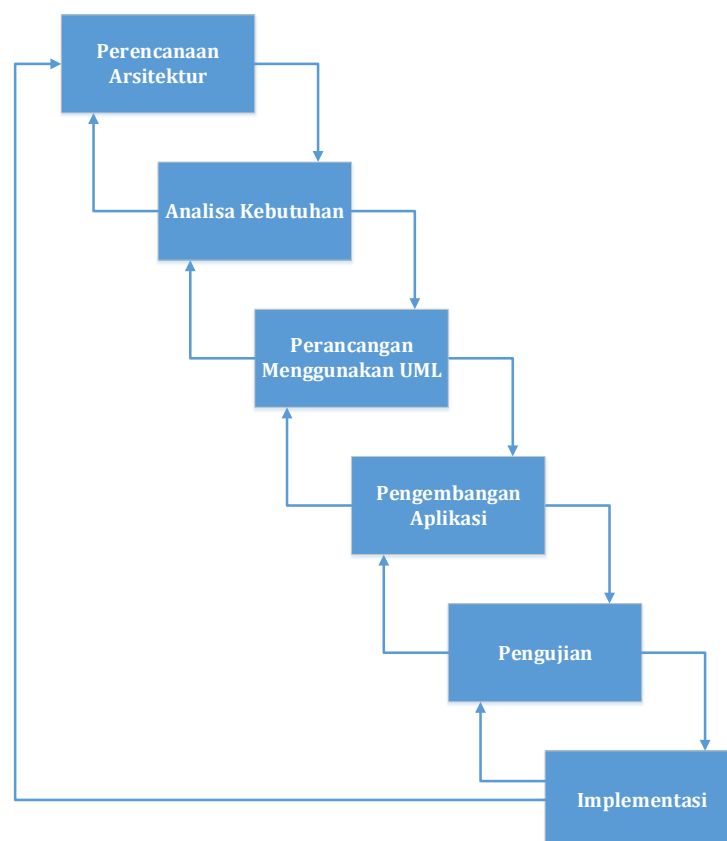
3.1.2. Bahan

Bahan yang akan digunakan dalam pembuatan aplikasi adalah data yang diperoleh dari dosen pembimbing dan berdasarkan LEMBAR PENGAJUAN

DANA BELANJA LAB TI FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH YOGYAKARTA.

3.2. Alur Penelitian

Untuk pembangunan sistem, penelitian menggunakan model SDLC (*Software Development Life Cycle*). Model SDLC yang dipakai dalam penelitian adalah model *Waterfall*. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Metode *Waterfall* ditunjukkan pada gambar 3.1.



Gambar 3.1 Metode *Waterfall*

Berikut penjelasan pembangunan sistem menggunakan SDLC yang terdiri dari 6 fase yaitu, Perencanaan Arsitektur, Analisa Kebutuhan, Perancangan Menggunakan UML, Pengembangan Aplikasi, Pengujian dan Implementasi.

1. Perencanaan Arsitektur

Pada fase perencanaan arsitektur ini merupakan fase awal dalam pembuatan sistem informasi yang mendefinisikan kebutuhan-kebutuhan sumber daya.

Dalam fase ini dilakukan langkah-langkah sebagai berikut:

- a. Mendefinisikan masalah dan menentukan tujuan sistem.
- b. Mengidentifikasi berbagai kendala sistem dan membuat studi kelayakan.

2. Analisa Kebutuhan

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan pengguna. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

3. Perancangan Menggunakan UML

Pada fase perancangan menggunakan UML merupakan proses perancangan sistem yang akan dibangun. Penelitian pada sistem yang berjalan dengan tujuan untuk merencanakan sistem yang baru menggunakan tools atau UML (*Unified Modelling Language*) dengan *software* Visio 2013.

4. Pengembangan Aplikasi

Fase pengembangan aplikasi merupakan fase mengimplementasikan design sistem pada fase-fase sebelumnya. Di dalam implementasi terdapat beberapa aktivitas yakni:

- a. Pembuatan database sesuai scema rancangan.
- b. Proses pembuatan aplikasi berdasarkan design sistem.
- c. Proses debugging atau pengujian dan perbaikan suatu aplikasi.

5. Pengujian

Tahap ini *software* hasil produksi harus diuji coba, termasuk semua fungsi-fungsinya, agar *software* bebas dari *error* dan hasilnya harus sesuai dengan kebutuhan.

6. Implementasi

Pada tahap implementasi ini, *software* yang sudah dibuat siap untuk di gunakan.

3.3. Analisis Kebutuhan

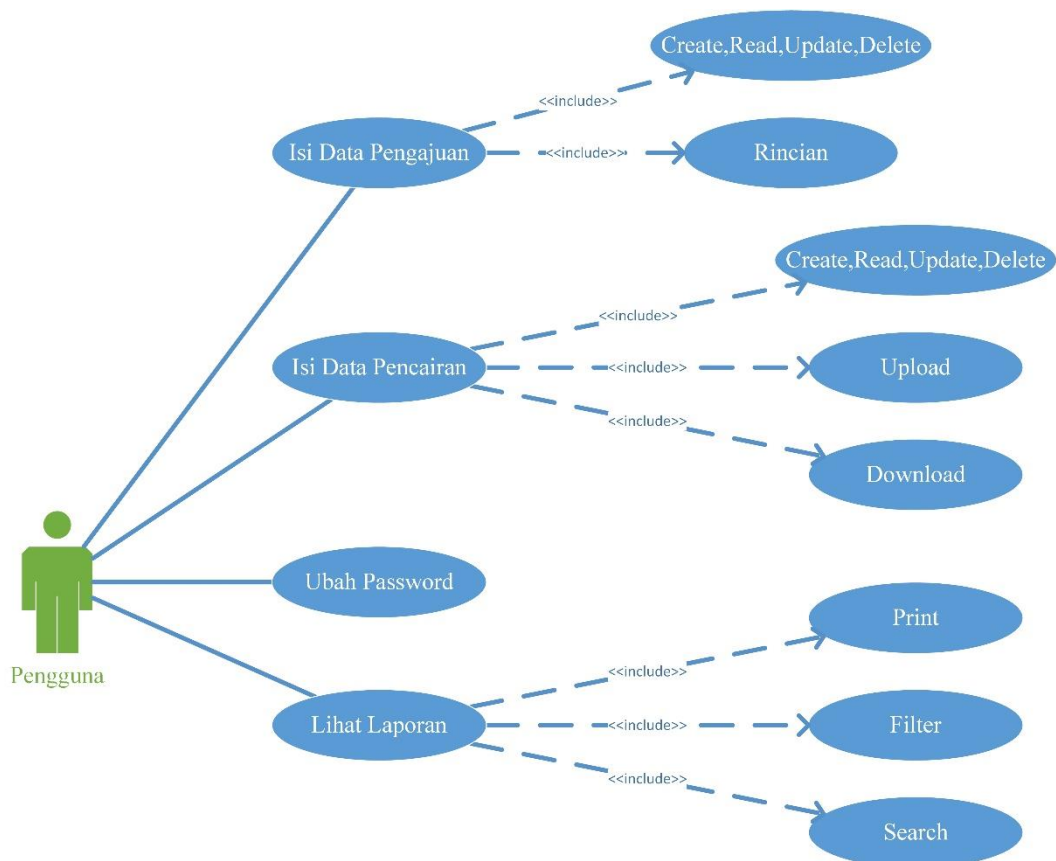
Analisis kebutuhan didapat dari diskusi Bersama dosen Prodi Teknik Informatika UMY. Berikut merupakan analisis kebutuhan pada aplikasi:

- a. Halaman untuk pengimputan data pengajuan dana.
- b. Halaman untuk pengimputan data pencairan dana.
- c. Fungsi untuk mengupload file ke dalam sistem.
- d. Halaman untuk melihat laporan

3.4. Rancangan

Dalam pembuatan aplikasi dilakukan perancangan *database* menggunakan bantuan Diagram ER. Metode perancangan lain yang digunakan dalam aplikasi adalah *Unified Modelling Language (UML)*. Model UML yang dipakai dalam pengembangan aplikasi yaitu model *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

1.4.1. Use Case Diagram



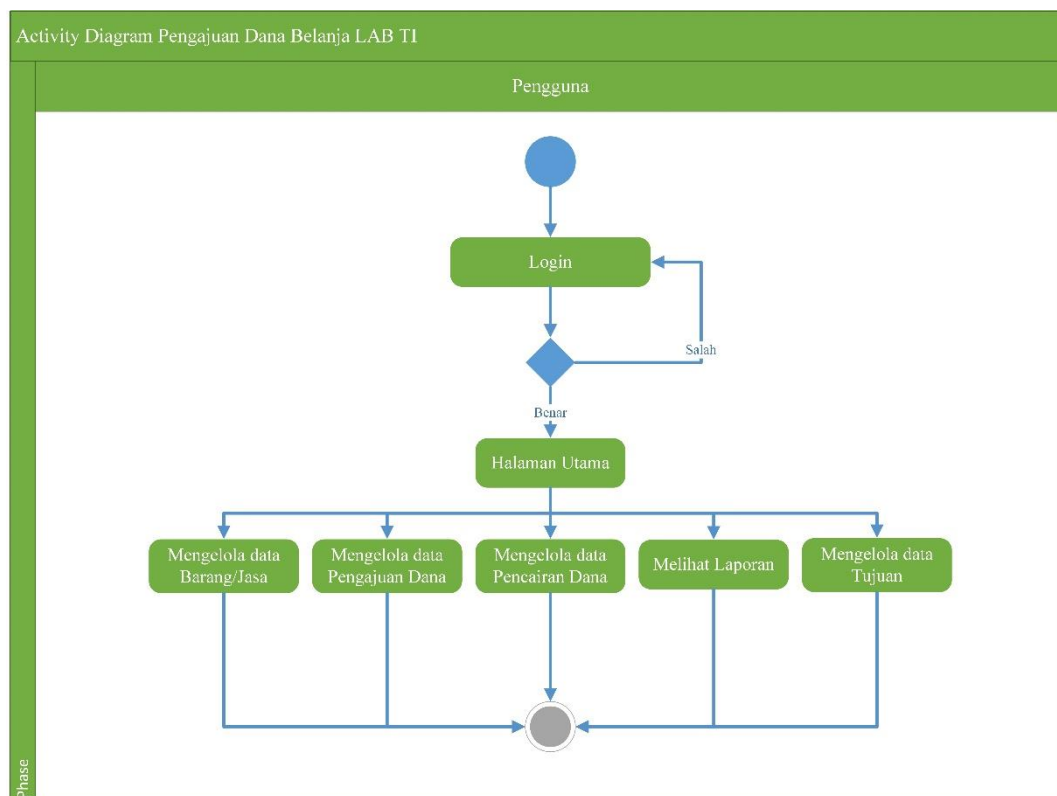
Gambar 3. 2 Use Case Diagram

Berdasarkan analisis kebutuhan dibuatlah *use case diagram* untuk membantu perancangan aplikasi. Terdapat 1 aktor yaitu pengguna atau admin.

Berikut ini adalah penjelasan *use case* yang dibuat:

- a. *Admin* dapat mengubah *password*.
- b. *Admin* dapat melakukan *create, read, update, delete* dan mengimput rincian dana pada tabel Pengajuan Dana.
- c. *Admin* dapat melakukan *create, read, update, delete, upload* dan *download* pada tabel Pencairan Dana.
- d. *Admin* dapat melakukan *print, filter, search* pada Laporan.

1.4.2. Activity Diagram



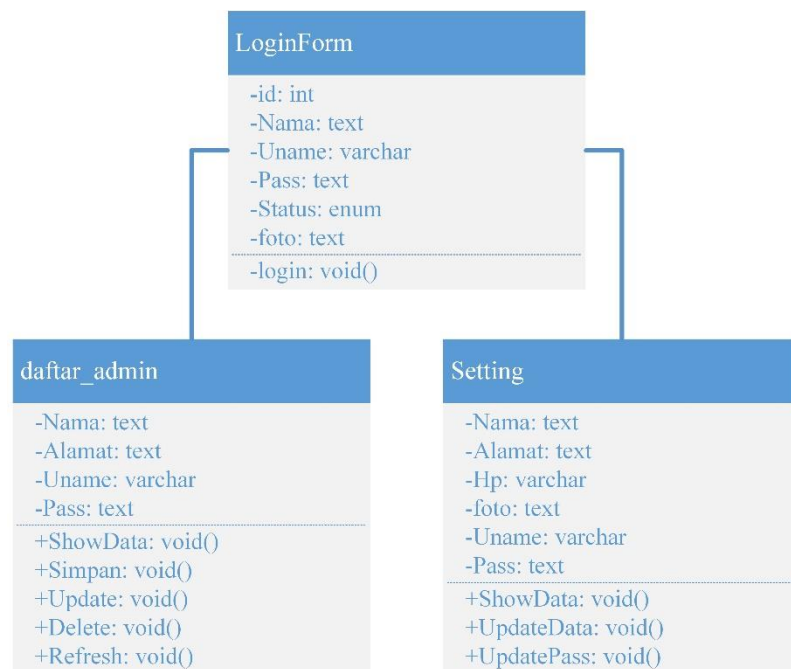
Gambar 3.3 Activity Diagram Pengajuan Dana Belanja LAB TI

Gambar 3.3 menunjukkan *Acitivity* Diagram pada kegiatan pengisian pengajuan dana belanja LAB TI. Jika saat login username atau password salah maka

kembali ke halaman login lagi namun jika username dan password benar maka halaman utama akan muncul. Selanjutnya di halaman utama pengguna dapat mengelola data pengajuan dana, data pencairan dana, data barang dan jasa, data tujuan dan melihat laporan.

1.4.3. Class Diagram

Gambaran *class* diagram yang digunakan dalam aplikasi dapat dilihat pada gambar 3.4 dan gambar 3.5.



Gambar 3. 4 Class Diagram Pengguna



Gambar 3. 5 Class Diagram Aplikasi

Penjelasan Fungsi

1. *Class LoginForm*

Pada *class LoginForm* terdapat fungsi *login* yang berfungsi untuk masuk ke menu aplikasi.

2. *Class Setting*

Pada *class Setting* terdapat fungsi *ShowData*, *UpdateData* dan *UpdatePass*. *ShowData* berfungsi untuk menampilkan data pengguna dari database, *UpdateData* berfungsi untuk mengupdate data pengguna yang telah dirubah dan *UpdatePass* berfungsi untuk mengupdate *password* yang telah dirubah oleh pengguna.

3. *Class daftar_admin*

Pada *class daftar_admin* terdapat fungsi *ShowData*, *Simpan*, *Update*, *Delete* dan *Refresh*. *ShowData* berfungsi untuk menampilkan data dari database ke dalam tabel, *Simpan* berfungsi untuk menyimpan data pengguna yg baru di inputkan, *Update* berfungsi untuk menyimpan data yang telah di *edit*, *Delete* berfungsi untuk menghapus data dan *Refresh* berfungsi untuk mengembalikan form pada tampilan awal.

4. *Class PengajuanDana*

Pada *class PengajuanDana* terdapat fungsi *ShowDataRincian*, *Simpan*, *Update*, *Delete*, *RefreshData* dan *ShowData*. *ShowDataRincian* berfungsi untuk menampilkan data rincian, *Simpan* berfungsi untuk menyimpan data pengajuan dana yang baru di inputkan, *Update* berfungsi untuk menyimpan data yang telah di *edit*, *Delete* berfungsi untuk menghapus data, *Refresh*

berfungsi untuk mengembalikan *form* pada tampilan awal dan *ShowData* berfungsi untuk menampilkan data pengajuan dana.

5. *Class Rincian*

Pada *class Rincian* terdapat fungsi *getPengajuanDana*, *ShowData*, *Simpan*, *Update*, *Delete*, *RefreshData*, *comboboxNoMemo*, *update_total_biaya* dan *get_total_biaya*. *getPengajuanDana* berfungsi untuk menampilkan data pengajuan dana, *ShowData* berfungsi untuk menampilkan data rincian, *Simpan* berfungsi untuk menyimpan data rincian yang baru di inputkan, *Update* berfungsi untuk menyimpan data yang telah di *edit*, *Delete* berfungsi untuk menghapus data, *RefreshData* berfungsi untuk mengembalikan *form* pada tampilan awal, *comboboxNoMemo* berfungsi untuk menampilkan kode barang berdasarkan tujuan pengajuan dana yang dipilih, *update_total_biaya* berfungsi untuk menampilkan total pada saat mengisi data rincian dan *get_total_biaya* berfungsi untuk menampilkan total rincian saat akan mengedit data rincian.

6. *Class Barang_Jasa*

Pada *class Barang_Jasa* terdapat fungsi *ShowDataBarang*, *Simpan*, *Update*, *Delete* dan *Refresh*. *ShowDataBarang* berfungsi untuk menampilkan data barang dan jasa, *Simpan* berfungsi untuk menyimpan data barang dan jasa yang baru di inputkan, *Update* berfungsi untuk menyimpan data yang telah di *edit*, *Delete* berfungsi untuk menghapus data dan *Refresh* berfungsi untuk mengembalikan *form* pada tampilan awal.

7. *Class Tujuan*

Pada *class Tujuan* terdapat fungsi *ShowDataTujuan*, *Simpan*, *Update*, *Delete* dan *Refresh*. *ShowDataTujuan* berfungsi untuk menampilkan data tujuan, *Simpan* berfungsi untuk menyimpan data tujuan yang baru di inputkan, *Update* berfungsi untuk menyimpan data yang telah di *edit*, *Delete* berfungsi untuk menghapus data dan *Refresh* berfungsi untuk mengembalikan form pada tampilan awal.

8. *Class PencairanDana*

Pada *class PencairanDana* terdapat fungsi *ShowDataPengajuan*, *ShowDataPencairan*, *Simpan*, *Update*, *Delete*, *Refresh*, *UploadPengajuan*, *UploadTransfer*, *DownloadPengajuan* dan *DownloadTransfer*. *ShowDataPengajuan* berfungsi untuk menampilkan data pengajuan dana, *ShowDataPencairan* berfungsi untuk menampilkan data pencairan dana, *Simpan* berfungsi untuk menyimpan data pencairan dana yang baru di inputkan, *Update* berfungsi untuk menyimpan data yang telah di *edit*, *Delete* berfungsi untuk menghapus data, *Refresh* berfungsi untuk mengembalikan *form* pada tampilan awal. *UploadPengajuan* berfungsi untuk mengupload bukti gambar hasil scan lembar pengajuan dana, *UploadTransfer* berfungsi untuk mengupload bukti gambar hasil scan lembar *transfer*, *DownloadPengajuan* berfungsi untuk mendownload bukti pengajuan dana yang telah di *upload* sebelumnya dan *DownloadTransfer* berfungsi untuk mendownload bukti *transfer* yang telah di *upload* sebelumnya.

9. *Class Download*

Pada *class Download* terdapat fungsi *save* yang berfungsi untuk mengatur letak penyimpanan dan format gambar yang di *download*.

10. *Class ConvertGambar*

Pada *Class ConvertGambar* terdapat fungsi *imageToBase64String* dan *StringToImage*. *imageToBase64String* berfungsi untuk mengkonvert gambar ke *string* dan *string* ke gambar. *StringToImage* berfungsi untuk menyimpan gambar ke dalam bentuk *string* atau *varchar*.

11. *Class LaporanDana*

Pada *class LaporanDana* terdapat fungsi *ShowData*, *print_All*, *print_Pengajuan*, *print_Pencairan*, *Filter*, *Refresh*. *ShowData* berfungsi untuk menampilkan data dari gabungan pengajuan dana dan pencairan dana, *print_All* berfungsi untuk mengeprint data dari gabungan pengajuan dana dan pencairan dana, *print_Pengajuan* berfungsi untuk mengeprint data dari pengajuan dana, *print_Pencairan* berfungsi untuk mengeprint data dari pencairan dana, *Filter* berfungsi untuk memfilter data yang ada pada tabel dan *refresh* berfungsi untuk mengembalikan tabel pada tampilan awal.

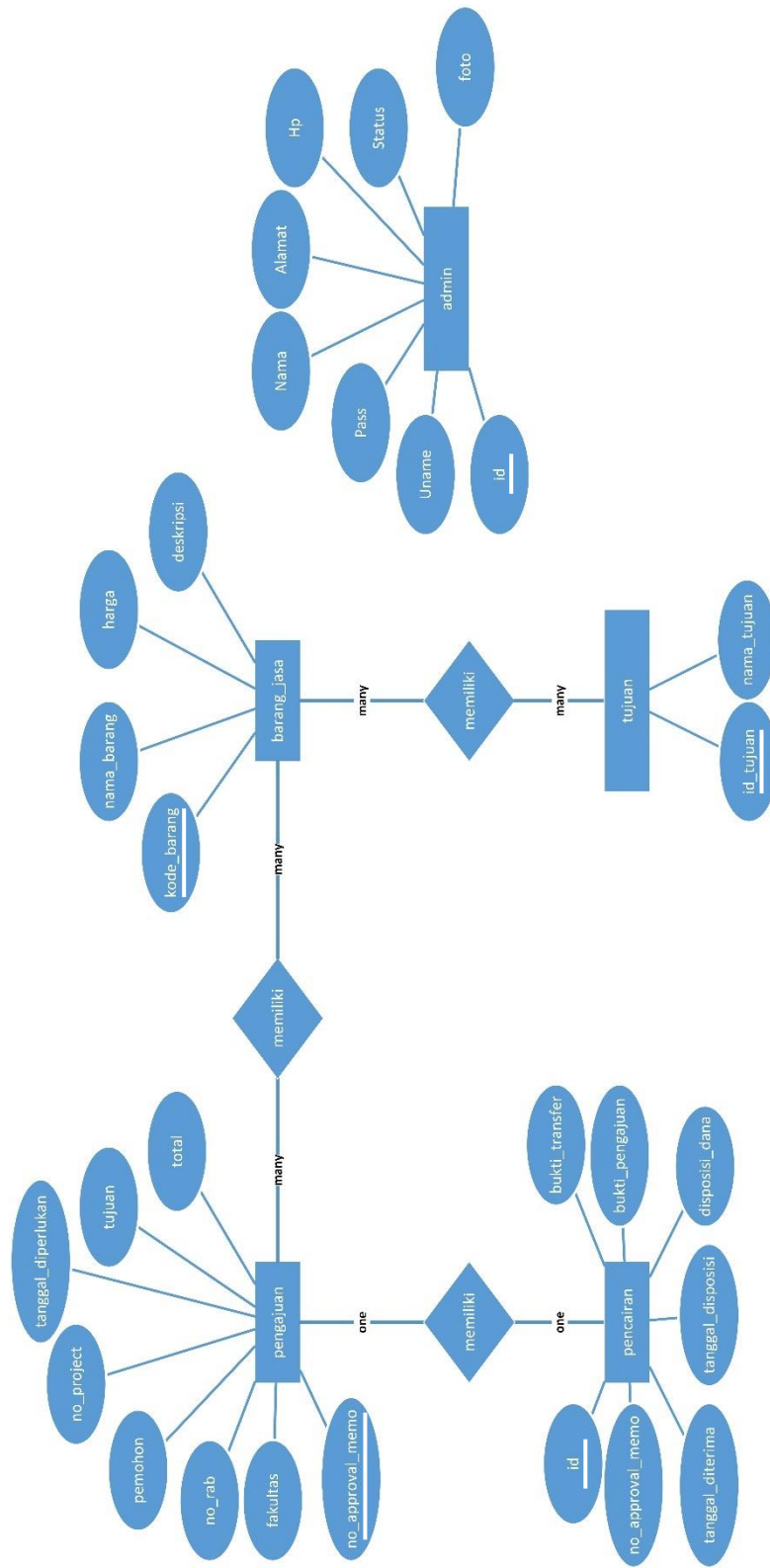
Berikut ini merupakan penjelasan *Class Diagram* pada gambar 3.4:

- a. Terdapat kelas pada diagram digunakan untuk inialisasi kedalam *table model* dan basis data aplikasi.
- b. Kelas *PengajuanDana* memiliki asosiasi dengan kelas *Rincian*. *PengajuanDana* ditambahkan ke dalam kelas *Rincian* melalui method

- getPengajuanDana(). Rincian ditambahkan kedalam kelas PengajuanDana melalui method ShowDataRincian().
- c. Kelas PencairanDana memiliki asosiasi dengan kelas PengajuanDana. PengajuanDana ditambahkan ke dalam kelas PencairanDana melalui method ShowDataPengajuan().
 - d. Kelas Barang_Jasa memiliki asosiasi dengan kelas Rincian. Barang_Jasa ditambahkan ke dalam kelas Rincian melalui method ShowData().
 - e. Kelas Tujuan memiliki asosiasi dengan kelas Rincian. Tujuan ditambahkan ke dalam kelas Rincian melalui method comboboxNoMemo().
 - f. Kelas LaporanDana memiliki asosiasi dengan kelas PengajuanDana dan PencairanDana. PengajuanDana dan PencairanDana ditambahkan ke dalam kelas LaporanDana melalui method ShowData().
 - g. Kelas *Download* memiliki asosiasi dengan kelas Rincian. *Download* ditambahkan ke dalam kelas Rincian melalui method DownloadPengajuan() dan DownloadTransfer().
 - h. Kelas ConvertGambar memiliki asosiasi dengan kelas Rincian. ConvertGambar ditambahkan ke dalam kelas Rincian melalui method UploadPengajuan() dan UploadTransfer().
 - i. Kelas *LoginForm* memiliki asosiasi dengan kelas Setting dan kelas daftar_admin melalui method login().

1.4.4. *Entity Relationship Diagram*

Entity Relationship Diagram (ER Diagram) merupakan diagram yang digunakan untuk merancang basis data dan memperlihatkan relasi antar entitas atau objek dengan atributnya. Tujuan ER Diagram dalam pengembangan aplikasi adalah dapat memberikan gambaran umum tentang sistem yang dikembangkan sehingga memudahkan dalam merancang basis data. Gambaran ER Diagram yang digunakan dalam aplikasi dapat dilihat pada gambar 3.6.



Gambar 3. 6 ER Diagram

Pada gambar 3.6 dapat dilihat bahwa *database* yang dirancang memiliki 5 buah entitas yaitu:

- a. *Admin*
- b. Pengajuan
- c. Pencairan
- d. Barang atau Jasa
- e. Tujuan

Berikut ini merupakan penjelasan dari ER diagram *database* aplikasi pada gambar 3.5:

1. Entitas *admin* tidak memiliki relasi dengan entitas lain. Artinya, entitas *admin* berdiri sendiri atau tidak bergantung dengan entitas lainnya.
2. Entitas pengajuan memiliki relasi dengan entitas pencairan. Relasi pada entitas pengajuan dan entitas pencairan adalah *one-to-one*. Relasi *one-to-one* pada entitas pengajuan dan pencairan mempunyai arti bahwa satu data pada entitas pencairan hanya dapat memiliki satu data pada entitas pengajuan, dan sebaliknya satu data pada entitas pengajuan hanya dapat dimiliki satu data pada entitas pencairan.
3. Entitas pengajuan memiliki relasi ke entitas barang_jasa. relasi antara kedua entitas tersebut menghasilkan rincian_pengajuan. Relasi antar kedua entitas tersebut adalah *many-to-many*. Relasi *many-to-many* pada entitas pengajuan dan entitas barang_jasa mempunyai arti bahwa banyak data pada entitas barang_jasa dapat memiliki banyak data pada entitas barang_jasa begitu juga sebaliknya.

4. Entitas *barang_jasa* memiliki relasi ke entitas tujuan. Relasi antara kedua entitas tersebut menghasilkan *list_kode_barang*. Relasi antar kedua entitas tersebut adalah *many-to-many*. Relasi *many-to-many* pada entitas tujuan dan entitas *barang_jasa* mempunyai arti bahwa banyak data pada entitas tujuan dapat memiliki banyak data pada entitas *barang_jasa* begitu juga sebaliknya.

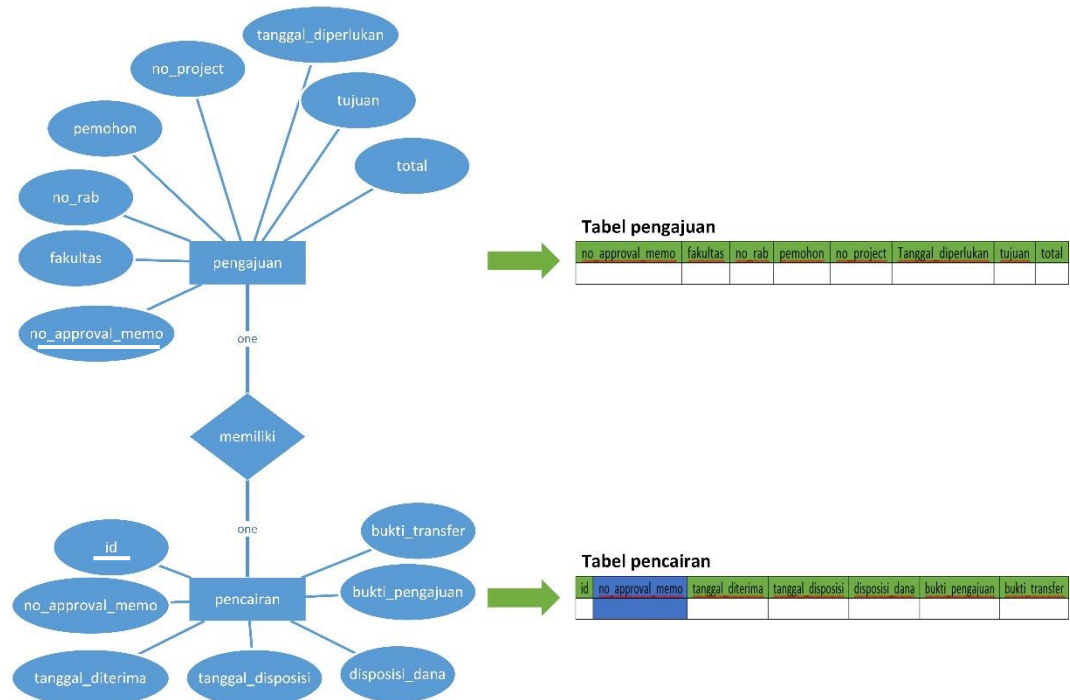
1.4.5. Konversi ERD ke dalam Tabel

Konversi ERD ke dalam bentuk tabel adalah proses memasukan entitas menjadi nama tabel dan attribute menjadi nama kolom beserta relasinya ke dalam sebuah tabel yang saling berhubungan.

1. Konversi dengan Relasi 1:1 (*one to one*)

Konversi entitas yang memiliki relasi 1:1 (*one to one*) yaitu menghubungkan dua entitas yang akan direpresentasikan dalam bentuk penambahan atau penyertaan atribut-atribut ke tabel yang mewakili salah satu dari kedua entitas tersebut. Pada ERD yang ditunjukkan pada gambar 3.6 ditunjukkan terdapat relasi *one to one* yaitu antara entitas pengajuan dengan entitas pencairan. Dari entitas-entitas tersebut akan direpresentasikan kedalam tabel, dapat dilihat pada gambar 3.7 sebagai berikut.

Entitas pengajuan dengan entitas pencairan



Gambar 3. 7 Konversi Relasi antara Entitas pengajuan dengan Entitas pencairan

Penjelasan konversi relasi entitas pengajuan dengan entitas pencairan pada gambar 3.7 adalah sebagai berikut:

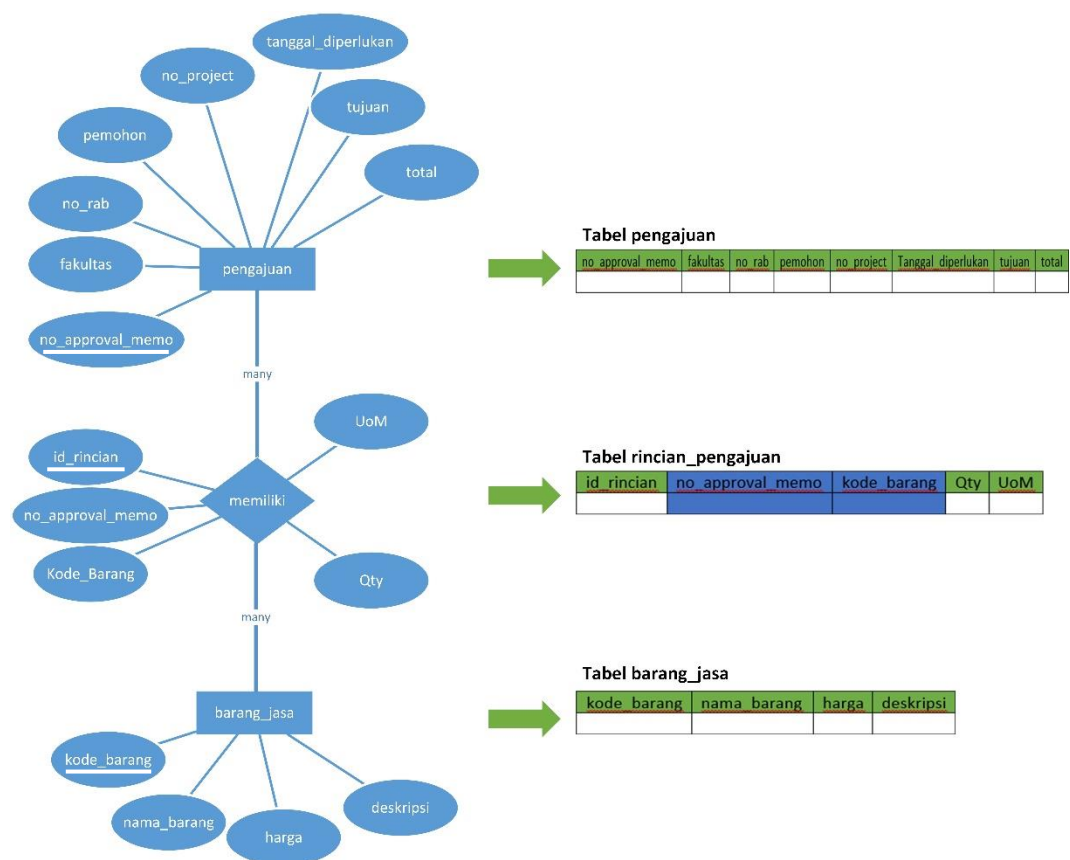
Entitas pencairan memiliki relasi *one to one* dengan entitas pengajuan yang artinya satu pencairan memiliki satu pengajuan maka kolom no_approval_memo yang ada pada tabel pengajuan perlu ditambahkan ke tabel pencairan yang berfungsi sebagai *foreign key* pada tabel pencairan.

2. Konversi dengan Relasi M:M (*many to many*)

Konversi entitas yang memiliki relasi M:M (*many to many*) yaitu menghubungkan dua entitas, diwujudkan dalam bentuk tabel (*file data*) khusus yang memiliki *field (foreign key)* yang berasal dari kunci-kunci (*key*) dari

entitas yang dihubungkan sehingga menciptakan tabel baru. Pada ERD yang ditunjukkan pada gambar 3.6 ditunjukkan terdapat relasi *many to many* yaitu entitas pengajuan dengan entitas barang_jasa dan entitas barang_jasa dengan entitas tujuan. Dari entitas-entitas tersebut akan direpresentasikan kedalam tabel, dapat dilihat pada gambar 3.8 dan gambar 3.9 sebagai berikut:

a. Entitas pengajuan dengan Entitas barang_jasa

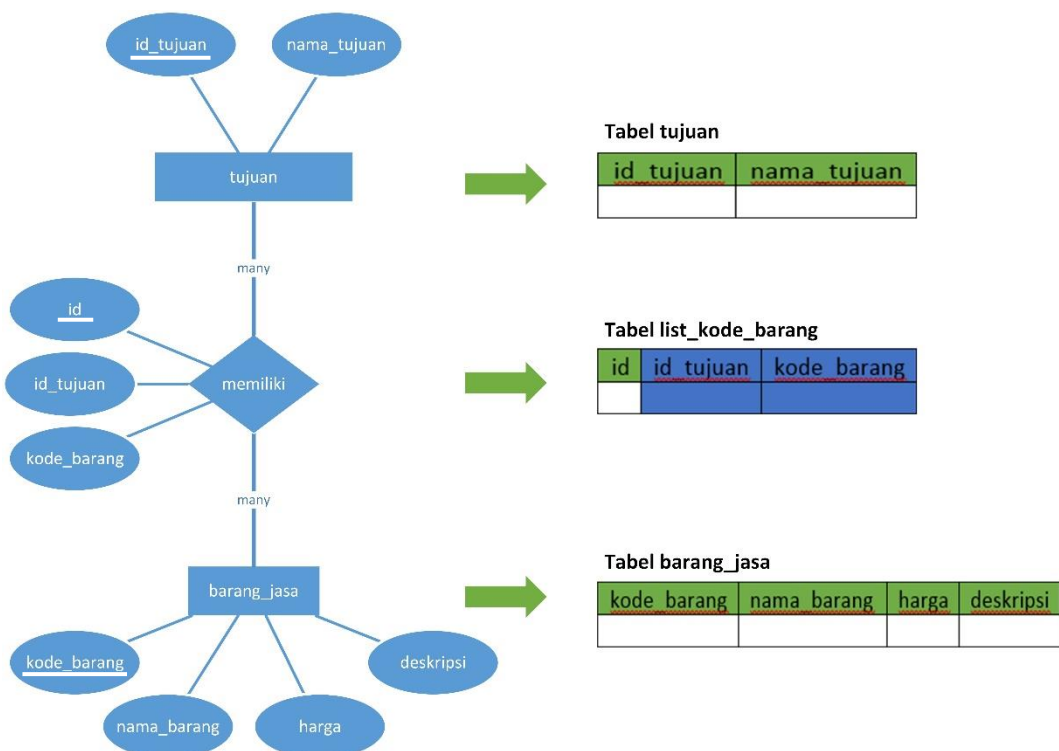


Gambar 3.8 Konversi Relasi antara Entitas pengajuan dengan Entitas barang_jasa

Penjelasan konversi relasi antara entitas pengajuan dengan entitas barang_jasa pada gambar 3.8 adalah sebagai berikut:

Entitas pengajuan memiliki relasi *many to many* dengan entitas barang_jasa sehingga tercipta tabel baru yang menghubungkan entitas pengajuan dengan entitas barang_jasa yang diberi nama rincian_pengajuan, dimana kolom `no_approval_memo` pada tabel pengajuan dan kolom `kode_barang` pada tabel barang_jasa ditambahkan ke dalam tabel rincian_pengajuan sebagai *foreign key*, maka perlu ditambahkan `id_rincian` sebagai *primary key* dan kolom lainnya sebagai penambah informasi seperti Qty dan UoM.

b. Entitas barang_jasa dengan Entitas tujuan



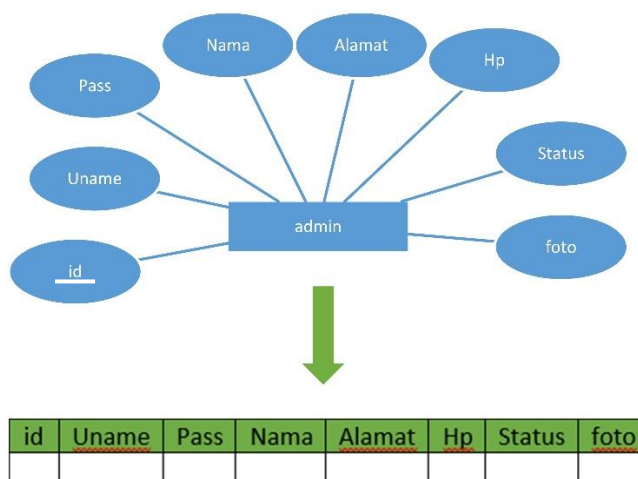
Gambar 3. 9 Konversi Relasi antara Entitas barang_jasa dengan Entitas Tujuan

Penjelasan konversi relasi antara entitas barang_jasa dengan entitas tujuan pada gambar 3.9 adalah sebagai berikut:

Entitas *barang_jasa* memiliki relasi *many to many* dengan entitas tujuan sehingga tercipta tabel baru yang menghubungkan entitas *barang_jasa* dengan entitas tujuan yang diberi nama *list_kode_barang*, dimana kolom *kode_barang* pada tabel *barang_jasa* dan kolom *id_tujuan* pada tabel tujuan ditambahkan ke dalam tabel *list_kode_barang* sebagai *foreign key*, maka perlu ditambahkan *id* sebagai *primary key* sebagai penambah informasi.

3. Konversi Entitas Kuat

Konversi entitas kuat adalah entitas yang dilibatkan atau dicantumkan secara jelas dalam ERD dan tidak memiliki ketergantungan terhadap entitas lain. Pada ERD yang ditunjukkan pada gambar 3.6 ditunjukkan terdapat entitas kuat yaitu entitas admin yang dapat dilihat pada gambar 3.10

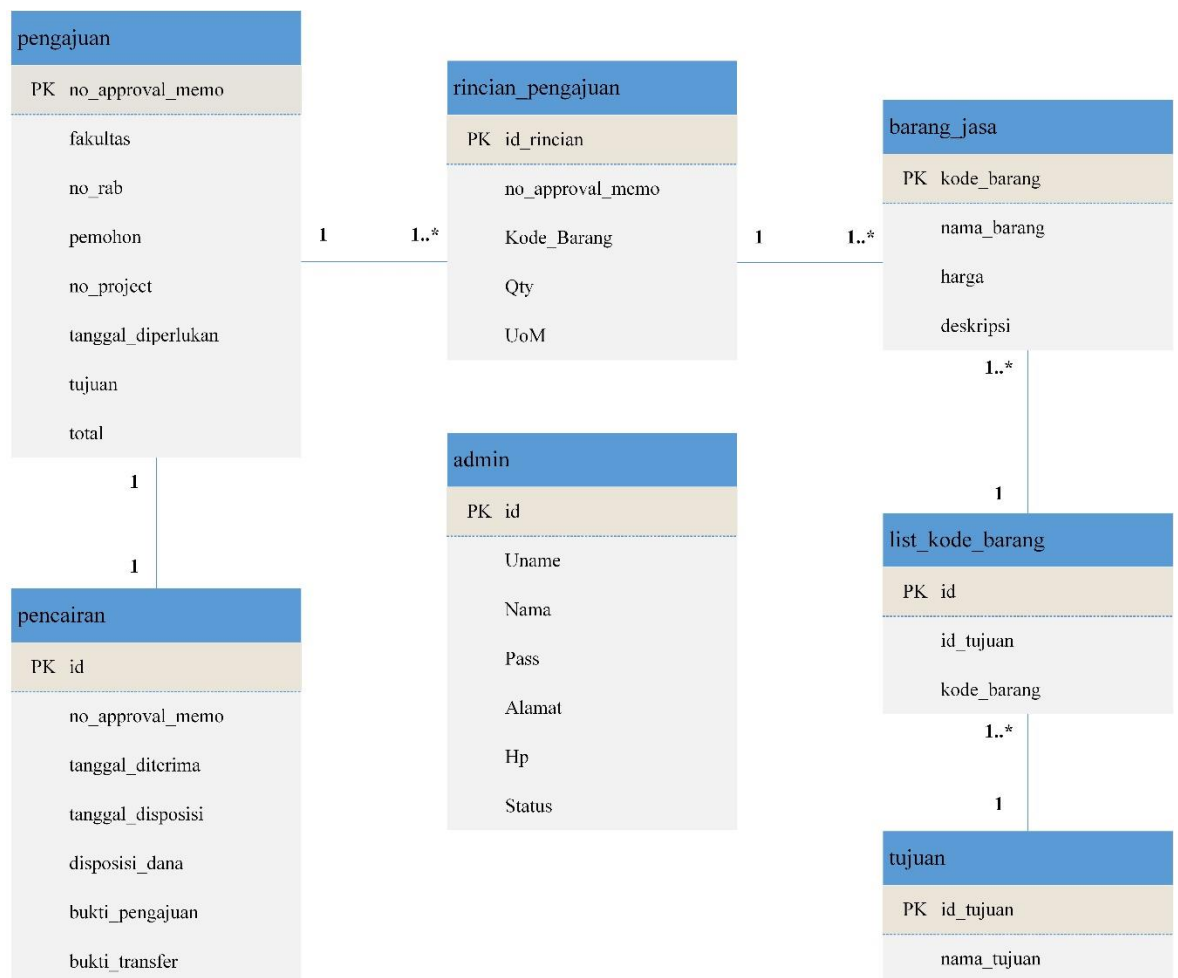


Gambar 3. 10 Konversi Entitas Kuat

Penjelasan konversi entitas kuat pada gambar 3.10 adalah sebagai berikut: Pada proses konversi entitas admin ke dalam bentuk tabel pada gambar 3.10 dijelaskan bahwa entitas menjadi nama tabel dan *attribute* menjadi nama kolom.

4. Relasi Antar Tabel

Setelah dilakukan konversi ERD ke dalam tabel maka akan didapatkan relasi antar tabel yang dapat dilihat pada gambar 3.11 sebagai berikut:



Gambar 3. 11 Relasi antar Tabel

1.4.6. Kamus Data Database

Berikut ini merupakan kamus data dari *database* yang dikelompokkan per tabel:

Tabel 3. 1 Kamus Data Tabel *Admin*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
ID	Int	Auto Increment	Primary Key	ID Pengguna berfungsi untuk mendata berapa banyak pengguna yang ada pada database
Uname	Varchar	10		Untuk menginput username pengguna
Pass	Varchar	10		Untuk menginput password pengguna
Nama	Varchar	20		Untuk menginput nama pengguna
Alamat	Varchar	40		Untuk menginput alamat pengguna
Hp	Varchar	12		Untuk menginput nomor handphone pengguna
Status	Enum	'Admin', 'Superadmin'		Untuk menentukan status pengguna admin atau super admin
Foto	Text			Untuk menginput foto pengguna

Tabel *Admin* berisi data *Admin* atau Pengguna. Pada tabel *Admin* terdapat delapan kolom yaitu kolom ID, Uname, Pass, Nama, Alamat, Hp, Status dan Foto dengan *primary key* yang terdapat pada kolom ID yang bertipe data *int*, untuk kolom lainnya bertipe data *varchar*, *text* dan *enum*, seperti yang terlihat pada tabel 3.1.

Tabel 3. 2 Kamus Data Tabel Pengajuan

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
No_Approval_Memo	Varchar	10	Primary Key	Nomor Approval Memo pengajuan dana
Fakultas	Varchar	6		Berisi nama fakultas yaitu “Teknik”
No_RAB	Varchar	5		Nomor RAB pengajuan dana
Pemohon	Varchar	20		Nama Pemohon pengajuan dana
No_Project	Varchar	5		Nomor Project pengajuan dana
Tanggal_Diperlukan	Date			Tanggal diperlukannya pengajuan dana tersebut
Tujuan	Text			Tujuan dari pengajuan dana
Total	Int	11		Total dari pengajuan dana yang di butuhkan

Tabel Pengajuan berisi data pengajuan dana. Pada tabel pengajuan terdapat delapan kolom yaitu kolom No_Approval_Memo, Fakultas, No_RAB, Pemohon, No_Project, Tanggal_Diperlukan, Tujuan dan Total dengan *primary key* yang terdapat pada kolom No_Approval_Memo yang bertipe data *varchar*, untuk kolom lainnya bertipe data *date*, *text* dan *int*, seperti yang terlihat pada tabel 3.2.

Tabel 3. 3 Kamus Data Tabel Rincian_Pengajuan

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
ID_Rincian	Int	Auto Increment	Primary Key	ID Rincian Pengajuan Dana
No_Approval_Memo	Varchar	10	Foreign Key	Nomor Approval Memo pengajuan dana yang diambil dari tabel pengajuan dana
Kode_Barang	Varchar	5	Foreign Key	Kode Barang yang diambil dari tabel barang_jasa
Qty	Int	11		Kuantiti barang atau jasa yang di butuhkan
UoM	Varchar	20		Ukuran Satuan untuk kuantiti yang di inputkan

Tabel Rincian_Pengajuan berisi data rincian dari pengajuan dana. Pada tabel rincian_pengajuan terdapat lima kolom yaitu kolom ID_Rincian, No_Approval_Memo, Kode_Barang, Qty dan UoM dengan *primary key* yang terdapat pada kolom ID_Rincian yang bertipe data *varchar*, pada tabel ini juga terdapat dua *foreign key* yaitu No_Approval_Memo dari tabel Pengajuan dan Kode_Barang dari tabel Barang_Jasa, seperti yang terlihat pada tabel 3.3.

Tabel 3. 4 Kamus Data Tabel Barang_ Jasa

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Kode_Barang	Varchar	5	Primary Key	Kode Barang atau jasa
Nama_Barang	Varchar	25		Nama Barang atau jasa
Harga	Int	11		Harga Barang atau jasa

Deskripsi Text Deskripsi Barang atau jasa

Tabel Barang atau Jasa berisi data Barang dan jasa. Pada tabel Barang atau Jasa terdapat empat kolom yaitu kolom Kode_Barang, Nama_Barang, Harga dan Deskripsi dengan *primary key* yang terdapat pada kolom Kode_Barang yang bertipe data *varchar*, untuk kolom lainnya bertipe data *text* dan *int*, seperti yang terlihat pada tabel 3.4.

Tabel 3. 5 Kamus Data Tabel Pencairan

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
ID	Int	Auto Increment	Primary Key	ID Pencairan Dana Pengajuan
No_Approval_Memo	Varchar	10	Foreign Key	Nomor Approval Memo pengajuan dana yang diambil dari tabel pengajuan dana
Tanggal_Diterima	Date			Tanggal dana diterima
Tanggal_Disposisi	Date			Tanggal disposisi dana
Disposisi_Dana	Varchar	20		Nama yang menerima Disposisi dana
Bukti_Pengajuan	Text			Hasil Upload bukti pengajuan dana berupa hasil scan gambar
Bukti_Transfer	Text			Hasil Upload bukti transfer dana berupa hasil scan gambar

Tabel Pencairan berisi data pencairan dana. Pada tabel pencairan terdapat tujuh kolom dengan *primary key* yang terdapat pada kolom ID yang bertipe data *int*, pada tabel ini juga terdapat *foreign key* yaitu No_Approval_Memo dari tabel Pengajuan, seperti yang terlihat pada tabel 3.5.

Tabel 3. 6 Kamus Data Tabel Tujuan

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
ID_Tujuan	Int	Auto Increment	Primary Key	ID Tujuan pengajuan dana
Nama_Tujuan	Varchar	30		Daftar nama-nama untuk tujuan pengajuan dana

Tabel Tujuan berisi data tujuan untuk pengajuan dana. Pada tabel tujuan terdapat dua kolom yaitu kolom ID_Tujuan dan Nama_Tujuan dengan *primary key* yang terdapat pada kolom ID_Tujuan yang bertipe data *varchar*, seperti yang terlihat pada tabel 3.6.

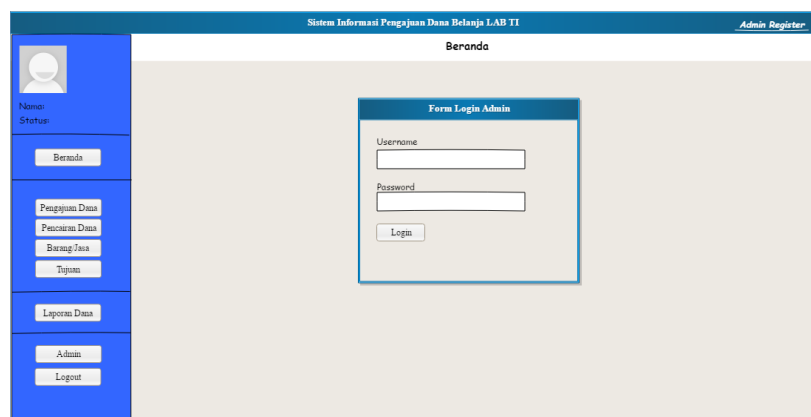
Tabel 3. 7 Kamus Data Tabel List_Kode_Barang

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
ID	Int	Auto Increment	Primary Key	ID List Kode Barang
ID_Tujuan	Int	2	Foreign Key	ID Tujuan pengajuan dana yang diambil dari tabel tujuan
Kode_Barang	Varchar	5	Foreign Key	Kode Barang atau jasa yang di ambil dari tabel barang_jasa

Tabel List_Kode_Barang berisi data untuk merelasikan tabel Barang_Jasa dengan tabel Tujuan. Pada tabel List_Kode_Barang terdapat tiga kolom yaitu kolom ID, ID_Tujuan dan Kode_Barang dengan *primary key* yang terdapat pada kolom ID dan dua *foreign key* yang terdapat pada kolom ID_Tujuan dan Kode_Barang, seperti yang terlihat pada tabel 3.7.

1.4.7. Rancangan Antar Muka

Antarmuka sangatlah penting dalam suatu aplikasi karena merupakan bagian dari perangkat lunak yang menjadi sarana komunikasi antar pengguna dengan sistem serta dapat memberikan kemudahan bagi pengguna dalam melakukan aktivitasnya.



Gambar 3. 12 Rancangan Antarmuka Halaman *Login*

Gambar 3.12 adalah rancangan antarmuka untuk halaman *login*. Pada halaman ini terdapat dua kontrol *textbox* untuk *input username* dan *password*, selain itu juga terdapat kontrol *button* untuk aksi *login*.



Gambar 3. 13 Rancangan Antarmuka Halaman Utama

Gambar 3.13 adalah halaman utama rancangan antarmuka halaman utama, pada halaman utama terdapat sembilan *button* menu yaitu: Beranda, Pengajuan Dana, Pencairan Dana, Barang atau Jasa, Tujuan, Laporan Dana, *logout* dan *Admin Register*.

Gambar 3. 14 Rancangan Antarmuka Halaman Pengajuan Dana

Gambar 3.14 adalah rancangan antarmuka halaman pengajuan dana, pada halaman ini terdapat lima *button* fungsi yaitu: simpan, *update*, *delete*, *refresh* dan rincian, dan juga dua *tabel* untuk data pengajuan dana dan rincian dana, serta lima *textbox*, satu *date* dan dua *combobox*.

Gambar 3. 15 Rancangan Antarmuka Halaman Rincian Pengajuan Dana

Gambar 3.15 adalah rancangan antar muka halaman rincian pengajuan dana, pada halaman rincian pengajuan dana terdapat satu *tabel*, satu *combobox*, empat *textbox* dan tiga *button* fungsi yaitu: simpan, *update*, *delete*.

Gambar 3. 16 Rancangan Antarmuka Halaman Pencairan Dana

Gambar 3.16 adalah rancangan antarmuka halaman pencairan dana, pada halaman pencairan dana terdapat dua *tabel* yaitu *tabel* pengajuan dana dan *tabel* pencairan dana, terdapat juga *button* fungsi *upload* dan *download*, serta lima *textbox*, dua *date*, dua *image* dan tiga *button* fungsi yaitu: simpan, *update*, *delete* dan *refresh*.

Gambar 3. 17 Rancangan Antarmuka Halaman Laporan Dana

Gambar 3.17 adalah rancangan antarmuka halaman laporan dana, pada halaman laporan dana terdapat satu *tabel*, dua *date*, satu *textbox* dan lima *button* fungsi yaitu: *filter*, *refresh*, *print all*, *print* laporan pengajuan dana dan *print* laporan pencairan dana

Gambar 3. 18 Rancangan Antarmuka Halaman Data Barang atau Jasa

Gambar 3.18 adalah rancangan antarmuka halaman data barang atau jasa, pada halaman data barang atau jasa terdapat satu *tabel*, empat *textbox* dan empat *button* fungsi yaitu: *simpan*, *update*, *delete* dan *refresh*.

Gambar 3. 19 Rancangan Antarmuka Halaman Tujuan

Gambar 3.19 adalah rancangan antarmuka halaman tujuan, pada halaman tujuan terdapat satu *tabel*, dua *textbox* dan empat *button* fungsi yaitu: *simpan*, *update*, *delete* dan *refresh*.

Gambar 3. 20 Rancangan Antarmuka Halaman *Setting Admin*

Gambar 3.20 adalah rancangan antarmuka halaman *setting admin*, pada halaman *setting admin* terdapat satu *image*, enam *textbox* dan dua *button* fungsi yaitu *ganti biodata* dan *ganti password*.

Title 1	Title 2	Title 3	Title 4	Title 5
1				
2				
3				

Gambar 3. 21 Rancangan Antarmuka Halaman *Admin Register*

Gambar 3.21 adalah rancangan antarmuka halaman *Admin Register*, pada halaman *Admin Register* terdapat satu *tabel*, empat *textbox* dan empat *button* fungsi yaitu: *simpan*, *update*, *delete* dan *refresh*.