BAB 4
PERANCANGAN PERANGKAT LUNAK SIPeV

Pada bab ini akan dibahas perancangan Sistem Manajamen Pembayaran Vakasi Fakultas Teknologi Industri Universitas Islam Indonesia (SIPeV FTI UII), khususnya pada bagian pemrosesan dan penyajian data. Model perancangan dari SIPeV menggunakan *Unfied Modelling Language* (UML). Perancangan meliputi *use case diagram, class diagram, sequence diagram, activity diagram*, perancangan *database*, dan perancangan antarmuka (*interface*).

4.1 Perancangan Sistem

Perancangan sistem pada SIPeV ini menggunakan UML *(Unified Modeling Language)* adalah sebuah bahasa pemodelan yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Visualisasi adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya (Sirkel, 2008).

UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus yang memiliki makna tertentu untuk menggambarkan berbagai aliran diagram perangkat lunak, sedangkan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

Dalam membangun suatu model perangkat lunak dengan UML, digunakan bentuk-bentuk diagram atau simbol untuk mempresentasikan elemen-elemen dalam sistem, seperti yang telah dijelaskan diatas bentuk diagram yang digunakan untuk merepresentasikan sistem SIPeV ini adalah *Use case diagram, Class diagram, Sequence diagram, Activity diagram.*

4.1.1 *Use Case Diagram*

Use case Diagram adalah suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem (Sirkel, 2008). *Use case Diagram* pada umumnya tersusun dari elemen-elemen sebagai berikut:

1. Use case ( )

   Use case merupakan proses-proses yang terjadi dalam suatu sistem.

2. *Actor* ( )

   Actor adalah seseorang atau sesuatu yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Sirkel, 2008).

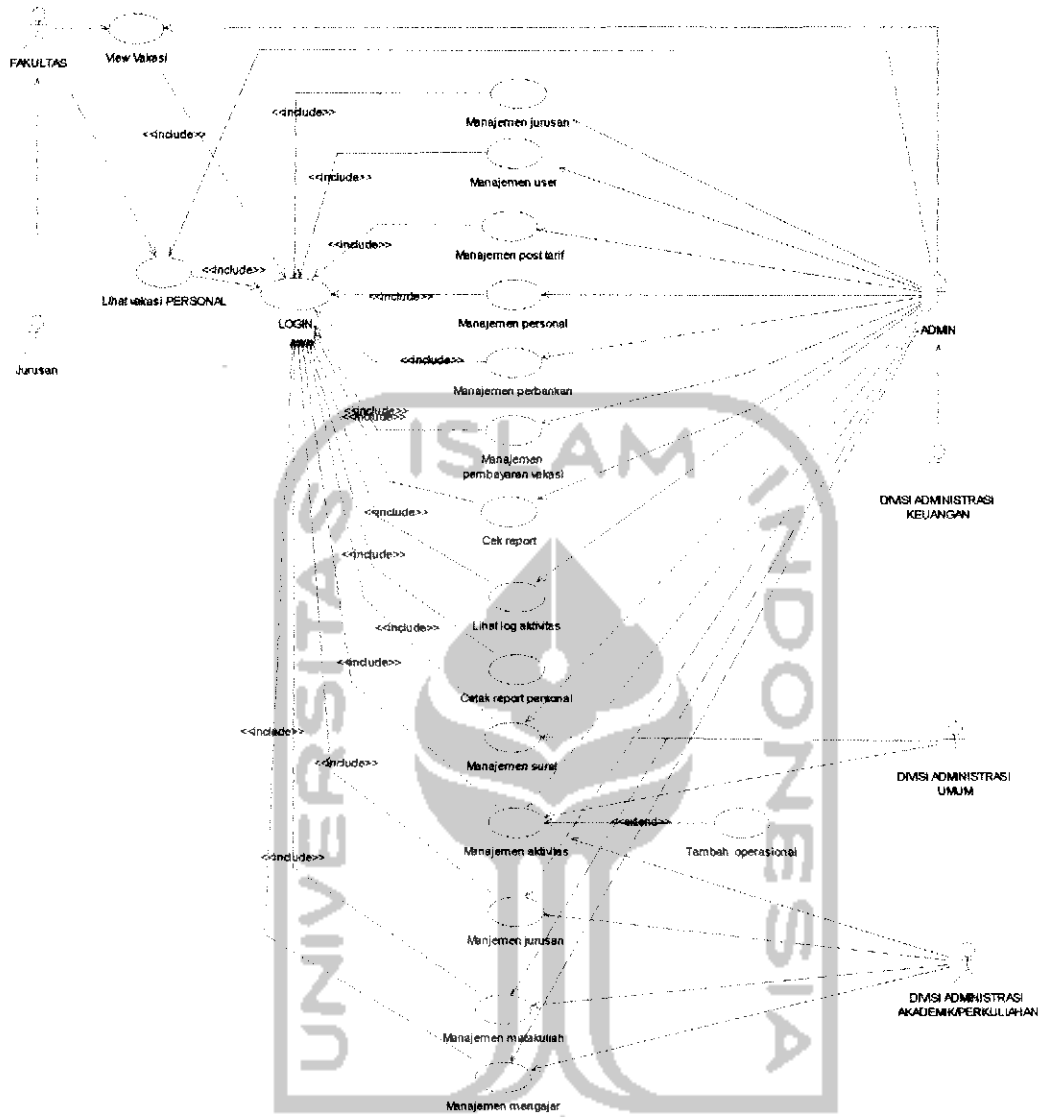
3. Relasi

   Relasi atau *relationship* merupakan hubungan antar elemen dalam *Use case Diagram.*

Gambar 4-1 akan menggambarkan fungsionalitas yang terdapat pada SIPeV, dan akan mempresentasikan interaksi antara pengguna dengan SIPeV. Berdasarkan *Use Case Diagram* pada Gambar 4-1 dapat dijelaskan bahwa di dalam SIPeV terdapat beberapa proses sistem yang didasarkan pada masing-masing divisi atau *user* yaitu sebagai berikut:

1. Divisi Administrasi Umum (DAU)

   Divisi administrasi umum harus terlebih dahulu melakukan proses *login* untuk mendapatkan hak akses sebagai *user* DAU dan mendapatkan proses manajemen surat untuk mengolah data surat. Dalam hal ini DAU dapat menambah data surat keluar yang telah disetujui oleh pihak Fakultas, mengubah atau menghapus data surat keluar yang tidak diperlukan.



Gambar 4-1 *Use Case Diagram*

2. Divisi Administrasi Perkuliahan/Akademik (DAP).

   Divisi administrasi perkuliahan/akademik harus terlebih dahulu melakukan proses *login* untuk mendapatkan hak akses sebagai *user* DAP dan mendapatkan proses:

   a) Manajemen data matakuliah

      DAP dapat menambah data matakuliah jika data belum ada, atau dapat menghapus dan mengubah data jika data tersebut salah atau data tersebut tidak diperlukan.

b) Manajemen data mengajar

DAP dapat menambah data mengajar dosen terhadap matakuliah yang dikehendaki jika data belum ada, atau dapat menghapus dan mengubah data jika data tersebut salah atau data tersebut tidak diperlukan.

c) Manajemen data aktivitas

DAP dapat menambah data manajemen aktivitas dosen. DAP menambah data aktivitas dosen yang mendapat aktivitas, mengajar atau mengoreksi hasil ujian untuk mendapatkan vakasi yang dikehendaki jika data belum ada, atau dapat menghapus dan mengubah data jika data tersebut salah atau data tersebut tidak diperlukan.

d) Manajemen data jurusan

DAP dapat menambah data jurusan yang ada di FTI jika data belum ada, atau dapat menghapus dan mengubah data jika data tersebut salah atau data tersebut tidak diperlukan.

3. Divisi Administrasi Keuangan (DAK).

Divisi administrasi keuangan harus terlebih dahulu melakukan proses *login* untuk mendapatkan hak akses sebagai *user* DAK dan mendapatkan proses:

a) Manajemen data personal

DAK dapat menambah data personal yang berada dalam struktural unit FTI. Selain itu DAK dapat mengubah data atau menghapus data personal jika terjadi kesalahan.

b) Manajemen data pembayaran vaksi

DAK dapat menambah data pembayaran yang belum ada, atau dapat menghapus dan mengubah data jika data tersebut salah atau data tersebut tidak diperlukan. Data pembayaran yang ditambahkan berupa pembayaran vakasi untuk staf dan pengajar di FTI sesuai dengan aktivitas yang dilakukan dan postarif yang ada.

UNIVERSITAS ISLAM INDONESIA

UNIVERSITAS ISLAM INDONESIA

c) Manajemen data perbankan

DAK dapat menambah data perbankan dari masing-masing personal yang belum ada, atau dapat menghapus dan mengubah data jika data tersebut salah atau data tersebut tidak diperlukan. Di dalam data perbankan ini terdapat nomor rekening dan nama bank dari masing-masing personal yang digunakan dalam pembayaran vakasi.

d) Manajemen data post tarif

DAK dapat menambah data post tarif berdasarkan pada aturan yang telah ditetapkan bukan hanya di FTI tetapi di UII. Selain menambah data DAK juga mendapat hak akses untuk mengubah data post tarif dan juga menghapus data posttarif.

e) Cek data *report*

Dalam hal ini sistem secara otomatis menambahkan data *report*, dalam hal ini DAK mendapat hak akses untuk melihat atau cek data vakasi yang telah dibayarkan.

f) Cetak *report* Personal

Setiap bulan DAK bertugas memberikan *report* kepada setiap personal berdasarkan aktivitas yang sudah dibayarkan, untuk memfasilitasinya sistem akan mencetak *report* vakasi bulanan personal yang telah dibayarkan.

4. Jurusan

Pihak manajerial yaitu masing-masing jurusan harus melakukan *login* terlebih dahulu sebelum mendapatkan hak akses dari sistem untuk melihat data vakasi personal setiap jurusan tersebut dan melihat frekuensi vakasi personal.

5. Fakultas

Dekan harus melakukan *login* terlebih dahulu sebelum mendaptkan hak akses dari sistem untuk melihat frekuensi vakasi personal, melihat frekuensi vakasi jurusan, dan melihat data vakasi personal.

6. Admin

   Admin harus melakukan *login* terlebih dahulu sebelum mendaptkan hak akses dari sistem, admin mendapat semua hak akses yang dimiliki oleh semua user yang lain ditambah dengan hak untuk mengakses *setting* atas aplikasi SIPeV dan melihat *log* (sejarah) pengunaan sistem SIPeV yang telah dilakukan oleh *user*.

4.1.2 *Class Diagram*

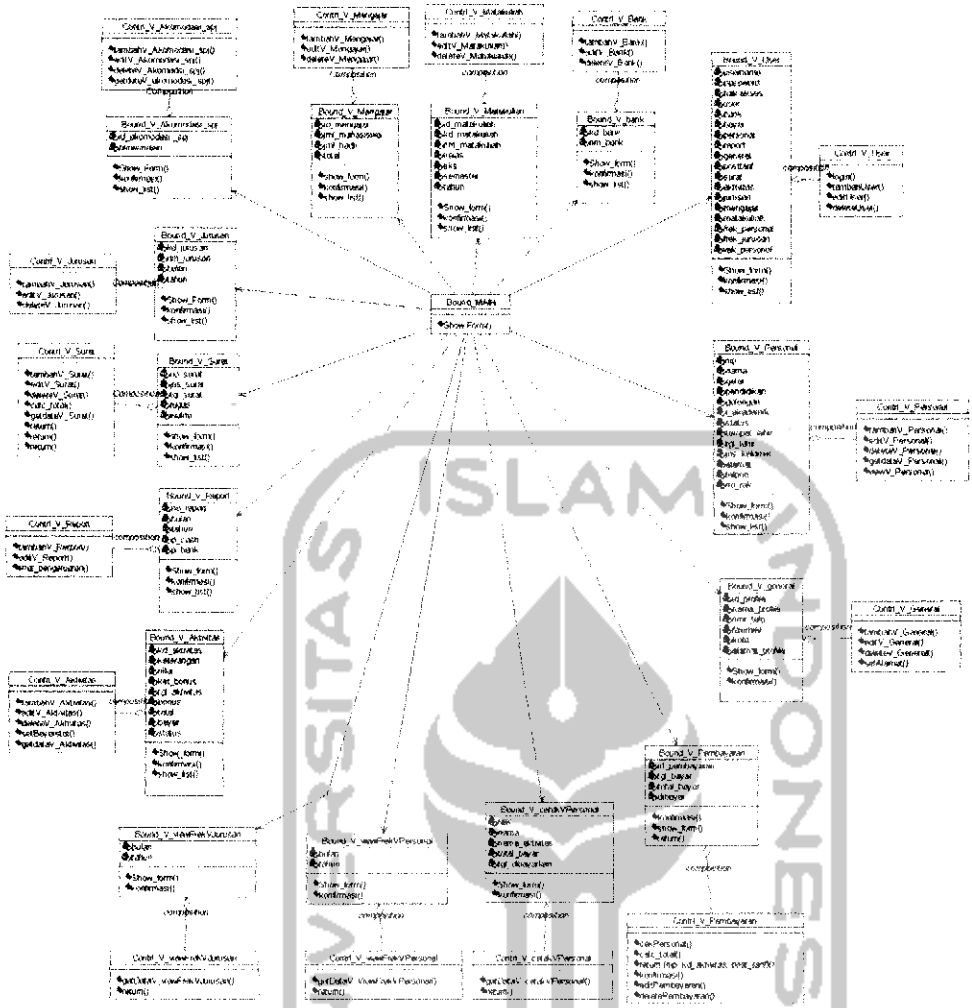
Class Diagram menggambarkan struktur dan deskripsi *class, package,* dan *objeck* beserta hubungan antar-*class* di dalam sistem seperti *containment,* pewarisan, asosiasi, dan lain-lain (Sirkel, 2008). Pada umumnya *Class diagram* tersusun dari elemen *class, interface, dependency, generalization,* dan *association* (Hermawan, 2004).

Pada Gambar 4-3 menggambarkan *class object* yang berorientasi ke *database*, sedangkan *class object* yang berorientasi dengan *form* dan prosesnya akan disinggung dibagian *Sequence Diagram* karena di dalam *Sequence Diagram* menjelaskan proses yang berjalan dalam aplikasi SIPeV. *Class Diagram* akan menggambarkan struktur dan deskripsi *class, package, dan object* berserta hubungan antar *class* dalam aplikasi SIPeV.

4.1.3 *Sequence Diagram*

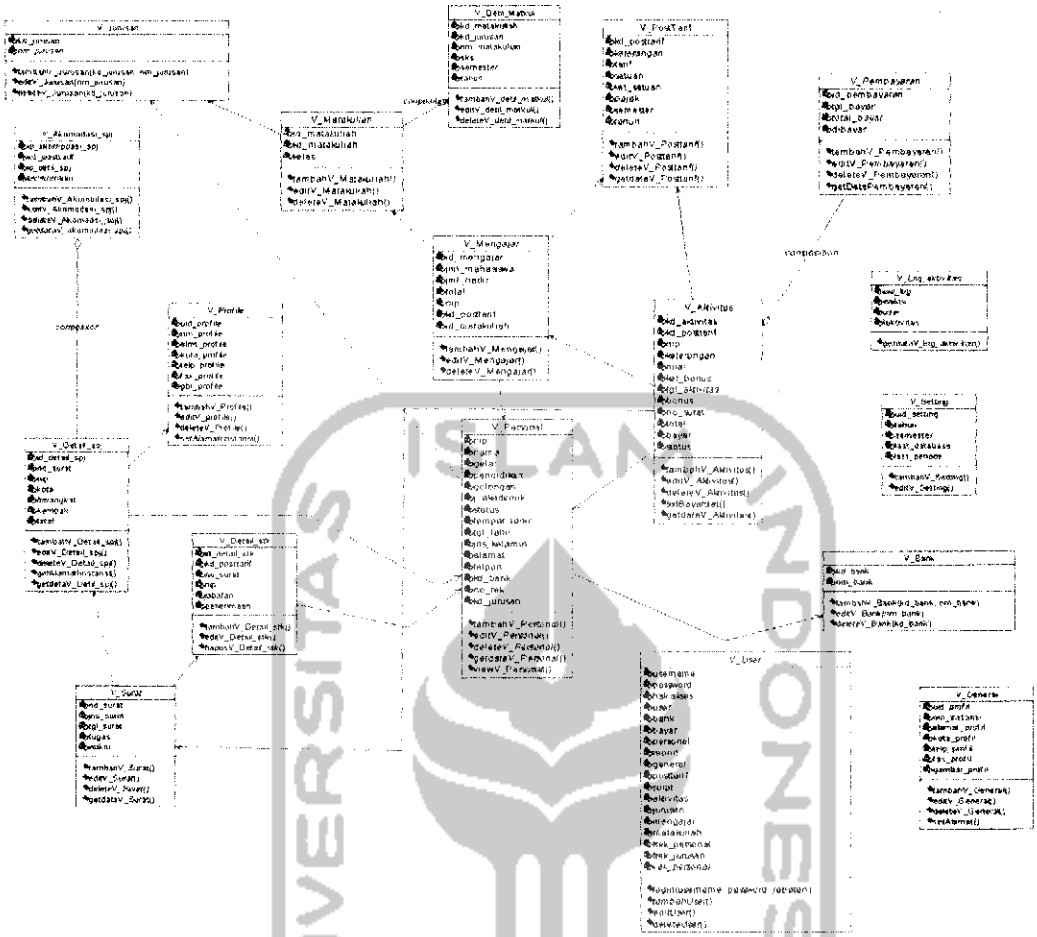
Sequence diagram merupakan gambaran interaksi antar objek dalam dan sekitar sistem (termasuk pengguna dan *display*) yang saling berkomunikasi menggunakan pesan (*message*) dan memiliki parameter waktu atau time (Sirkel, 2008).

Sequence diagram kali ini juga terdapat *class object* yang berorientasi dengan *form/ boudary* (berinteraksi langsung antara aktor dan *interface/* tampilan) dan proses/ *controler* (aktivitas yang ada dalam setiap boundary), seperti yang digambarkan pada Gambar 4-2.



Gambar 4-2 *Class Object* yang Berorientasi *Form* dan Prosesnya

Dari gambar diatas dapat dijelaskan bahwa setiap *boundary* atau *form* hampir selalu mempunyai *controler* atau proses atau biasa dikatakan antara boundary dan controler mempunyai hubungan composition (mempunyai hubungan yang lebih kuat), setelah mengetahui class yang berorientasi pada database, *form*, dan proses dapat menggambarkan secara lengkap sebuah *squence*, dalam hal ini *squence* hanya akan digambarkan 6 untuk mewakili proses yang sama.

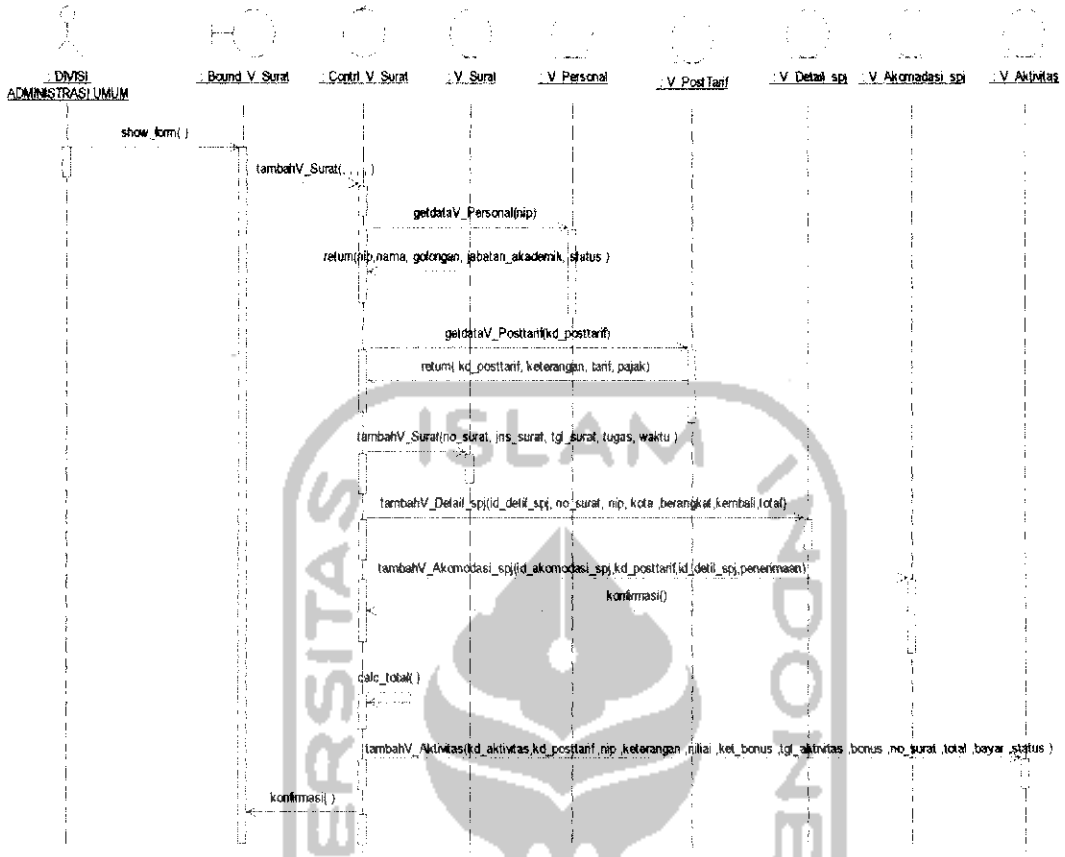


Gambar 4-3 *Class Object* yang Berorientasi *Database*

Sequence diagram dalam perancangan SIPeV akan diwakili oleh 6 *squence* diagram yaitu:

1. *Sequence diagram* tambah Surat Perjalanan Dinas (SPJ) yang akan digambarkan pada Gambar 4-4, dan mewakili proses penambahan data.

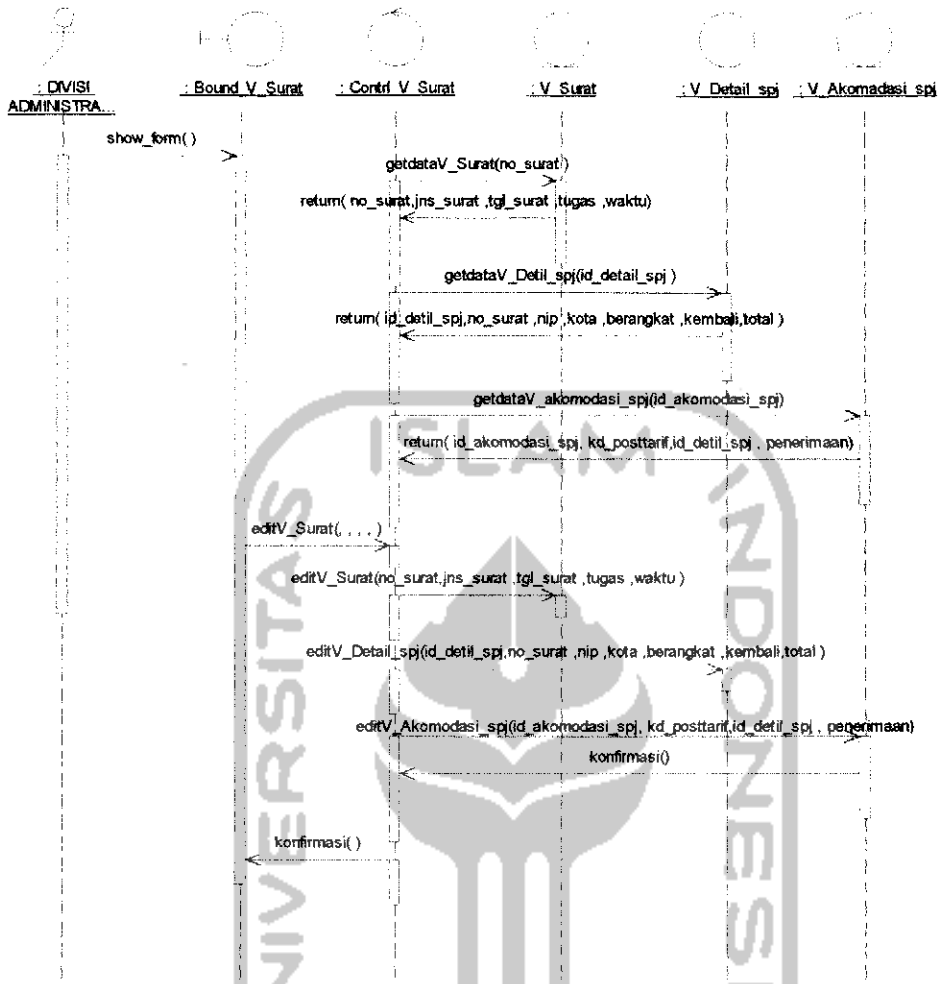
   Sequence Diagram tambah SPJ dilakakukan oleh DAU yang merupakan salah satu dari surat keluar fakultas. *Squence* diagram ini mewakili *squence* diagram untuk penambahan data. Proses penambahan data pada SPJ sangat kompleks sehingga dapat mewakili proses penambahan data yang lain dalam SIPeV. DAU menambahkan data surat keluar dengan kebutuhan input pada proses penambahan data SPJ yang telah dijelaskan pada bab 3. Setelah DAU menambahkan data SPJ secara otomatis sistem akan melakukan proses kalkulasi total, dan proses penambahan data aktivitas berdasarkan NIP.



Gambar 4-4 *Sequence Diagram* Tambah Surat Perjalanan Dinas

2. *Sequence diagram* edit Surat Perjalanan Dinas (SPJ) yang akan digambarkan pada Gambar 4-5, dan mewakili proses pengubahan (edit) data.

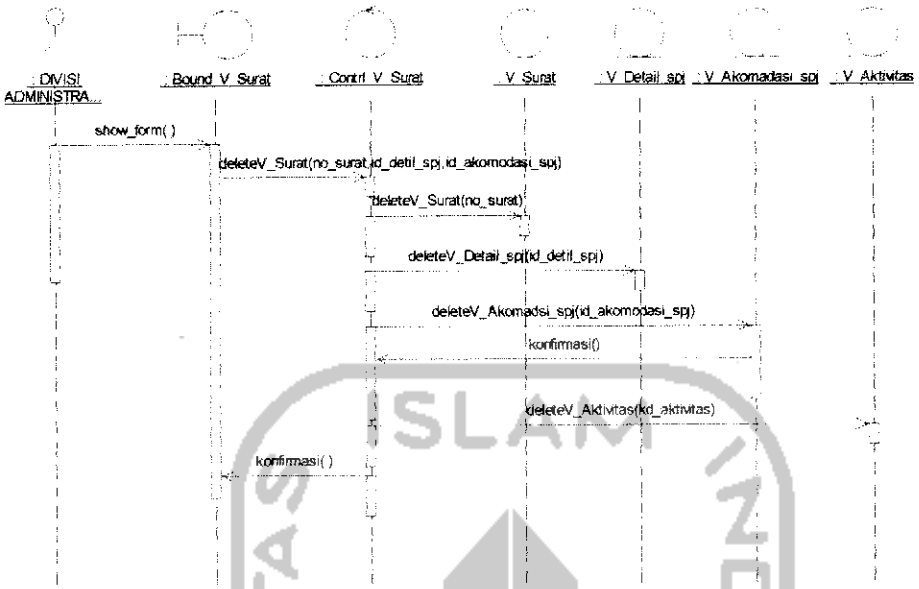
   Sequence Diagram edit SPJ dilakakukan oleh DAU jika data SPJ yang telah ditambahkan terjadi kesalahan. Maka DAU dapat memanggil data yang salah kemudian dilakukan pengubahan (edit). Pada proses pengubahan akan mempengaruhi data total biaya aktivitas.



Gambar 4-5 *Sequence Diagram* Edit Surat Perjalanan Dinas

3. *Sequence diagram delete* Surat Perjalanan Dinas (SPJ) yang akan digambarkan pada Gambar 4-6, dan mewakili hapus (delete) data.

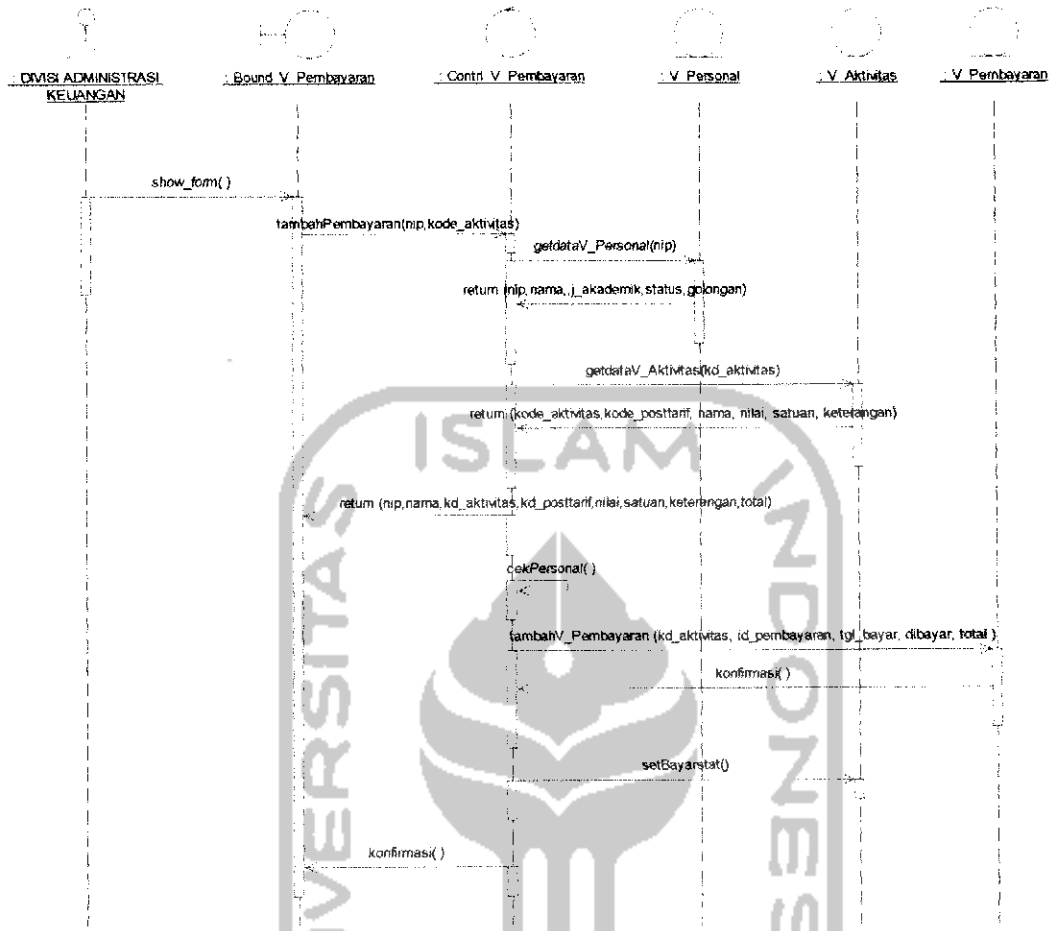
   Sequence Diagram delete SPJ dilakakukan oleh DAU jika data SPJ yang telah ditambahkan terjadi kesalahan atau tidak terpakai. Maka DAU dapat memanggil data yang salah kemudian dilakukan panghapusan (*delete*). Pada proses panghapusan akan otomatis menghapus data aktifitas SPJ berdasarkan NIP.



Gambar 4-6 *Sequence Diagram Delete* Surat Perjalanan Dinas

4. *Sequence diagram* tambah pembayaran yang akan digambarkan pada Gambar 4-7.

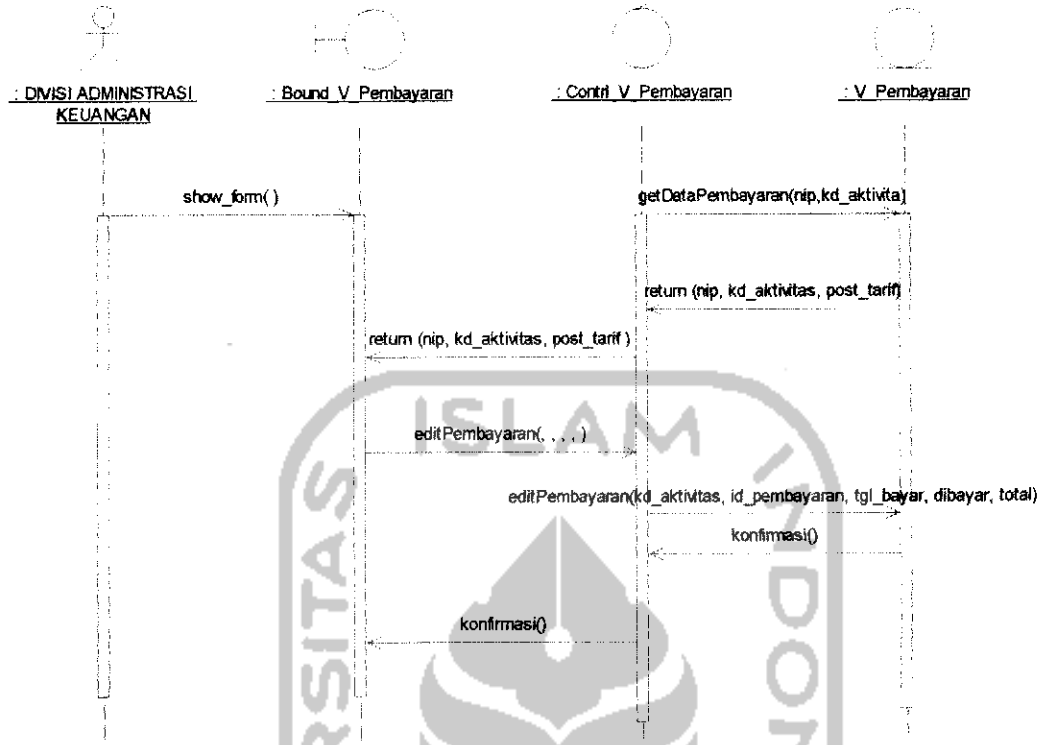
   Sequence Diagram Tambah Pembayaran pada SIPeV dimulai dengan aktor yang dalam hal ini Divisi Administrasi Keuangan (DAK) mengisi *form* manajemen pembayaran, selanjutnya DAK memasukkan NIP dan sistem akan mencari detil NIP dari database. Begitu pula jika telah memasukkan kode aktivitas maka sistem akan mencari detil kegiatan yang telah dilakukan oleh NIP tersebut. setelah didapat detil dari keduanya maka akan terjadi penghitungan total dari vakasi yang didapat selanjutnya DAK dapat menambahkan atau menyimpan data tersebut kedalam *database* dan data aktivitas yang sudah dibayarkan akan diset agar aktivitas tidak muncul yang berakibat aktivitas tersebut dapat dibayar ulang kemudian sistem akan mengeluarkan pesan konfirmasi bahwa data berhasil disimpan.



Gambar 4-7 *Sequence Diagram* Tambah Pembayaran

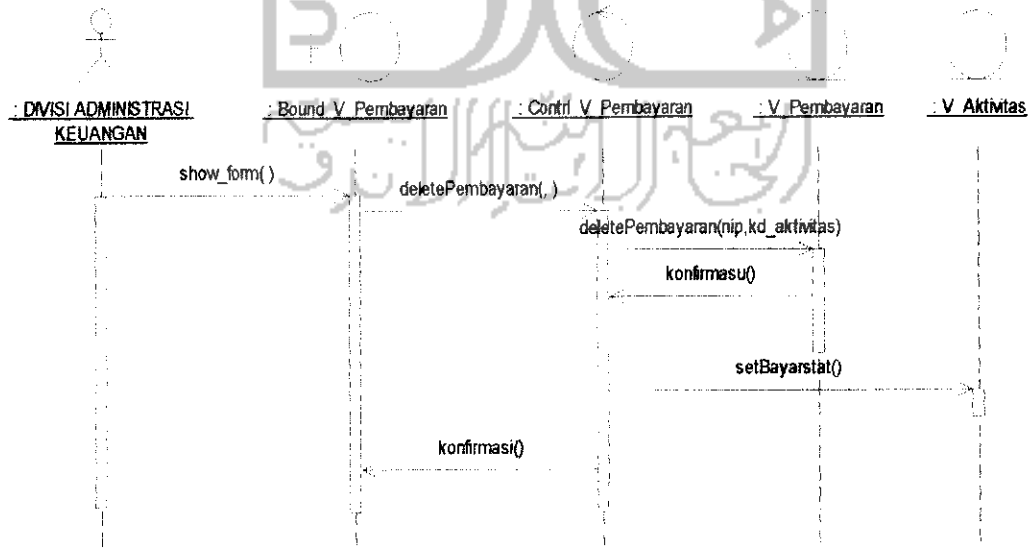
5. *Sequence diagram* edit pembayaran yang akan digambarkan pada Gambar 4-8.

   Sequence Diagram Edit Pembayaran pada SIPeV dimulai dengan aktor yang dalam hal ini Divisi Administrasi Keuangan (DAK) mengisi *form* manajemen pembayaran, data baru yang telah ditambahkan pada bagian tambah pembayaran, dapat diubah jika terdapat kesalahan. DAK akan memberikan data yang akan diubah berupa NIP dan kode aktivitas, selanjutnya sistem akan mencari data tersebut kedalam database, dan kemudian ditampilkan ke dalam *form* edit pembayaran. DAK dapat merubah data yang salah setelah data tersebut ditampilkan oleh sistem dan akan mendapatkan konfirmasi jika data tersebut berhasil dirubah.



Gambar 4-8 *Sequnce Diagram* Edit Pembayaran

6. *Sequence diagram delete* pembayaran yang akan digambarkan pada Gambar 4-9.



Gambar 4-9 *Sequnce* Delete Pembayaran

Sequence Diagram Delete Pembayaran pada SIPeV dimulai dengan aktor yang dalam hal ini Divisi Administrasi Keuangan (DAK) mengisi *form* manajemen pembayaran, selanjutnya DAK memasukkan nip dan kode aktivitas, jika kedua data tersebut sudah ada, maka data tersebut akan tampil dan DAK dapat menghapus data yang ada jika data tersebut sudah tidak terpakai lagi. Selanjutnya sistem akan memberikan konfirmasi apabila data tersebut berhasil di hapus.

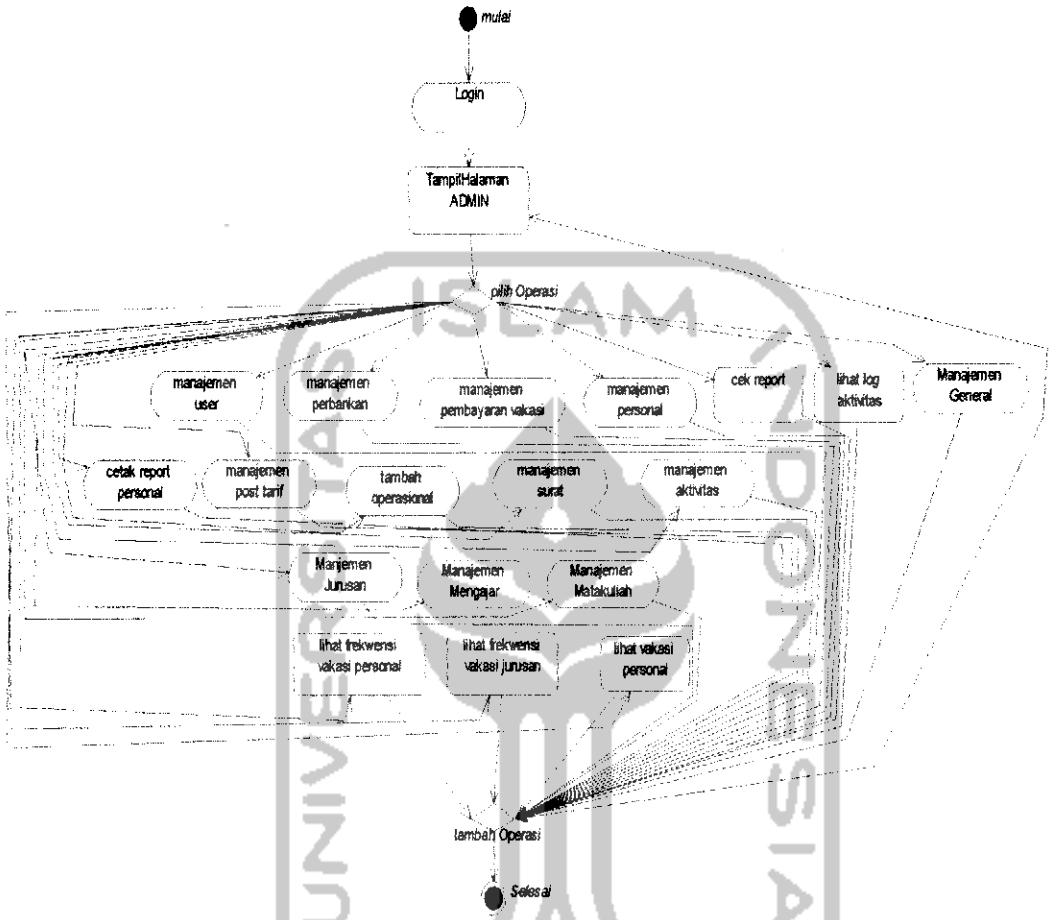
4.1.4 *Activity Diagram*

Activity Diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnin, dan aliran kerja dalam banyak kasus (Munawar, 2005).

Pada umumnya *Activity Diagram* tidak menampilkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan prosesnya (Sirkel, 2008). Diagram ini mirip dengan *flowchart* karena dapat menggambarkan sebuah alur kerja dari aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam *state*. Perbedaannya dengan *flowchart* adalah pada *Activity Diagram* dapat mendukung perilaku paralel, sedangkan pada *flowchart* tidak bisa.

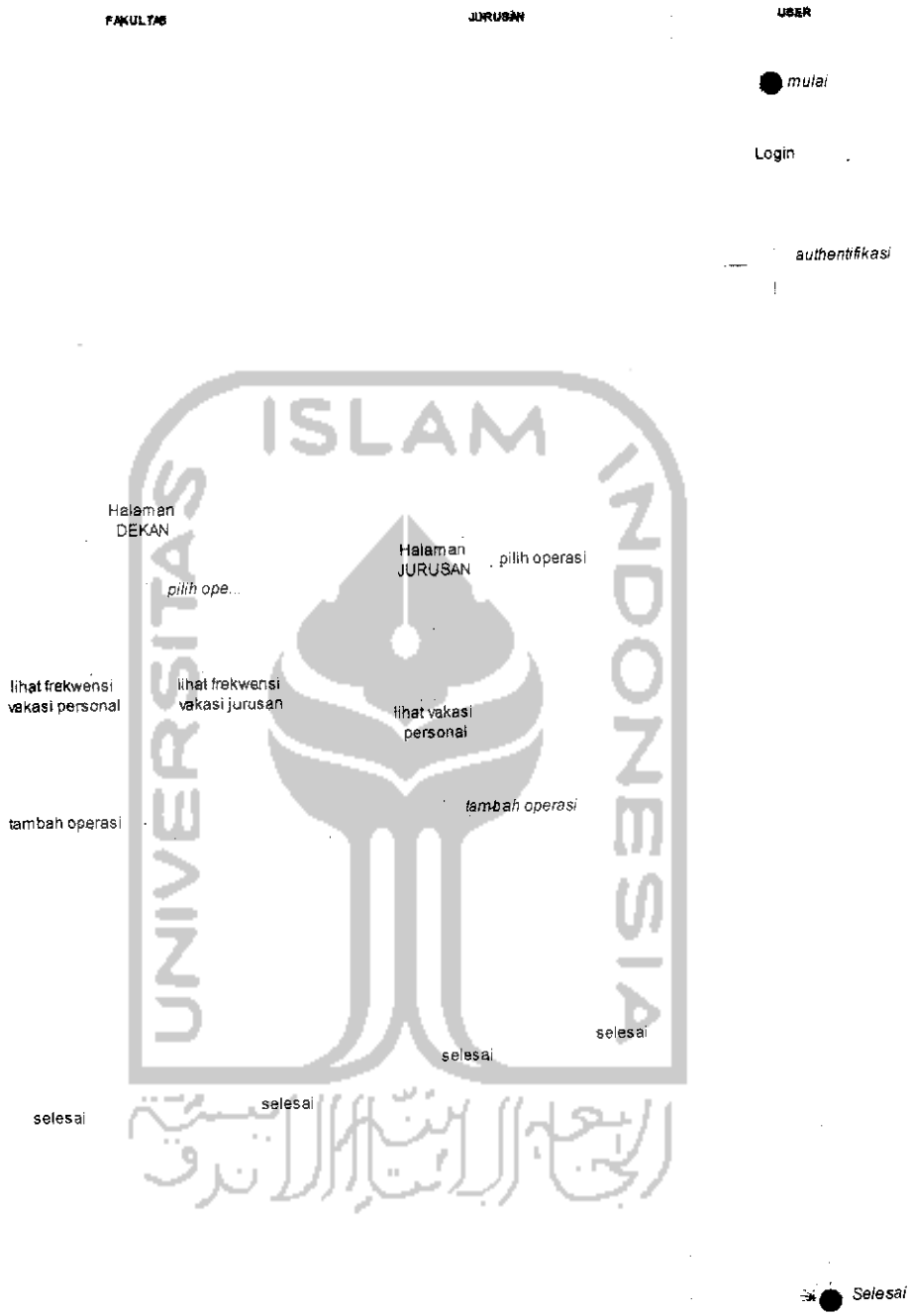
Gambar 4-11 menunjukan aktivitas yang terjadi di dalam aplikasi SIPeV, pengelompokan dari masing-masing aktivitas berdasarkan hak akses yang digunakan pengguna aplikasi SIPeV pada waktu melakukan *login*. Hak akses sebagai jurusan dapat melakukan aktivitas lihat vakasi personal. Hak akses sebagai fakultas dapat melakukan aktivitas lihat frek. vakasi personal,lihat. frek. vakasi jurusan,lihat vakasi personal. Hak akses sebagai DAP dapat melakukan aktivitas manajemen jurusan, manajemen aktivitas, manajemen matakuliah, manajemen mengajar. Hak akses sebagai DAU dapat melakukan aktivitas tambah operasioan, manajemen surat, manajemen personal. Hak akses sebagai DAK dapat melakukan aktivitas manajemen perbankan, manajemen posttarif, manajemen pembayaran vakasi, sedangkan hak akses sebagai admin dapat melakukan aktivitas keseluruhan dari setiap hak akses ditambah aktivitas manajemen *user* dan lihat *log* aktivitas *user* yang akan ditampilkan pada Gambar 4-10 dan untuk

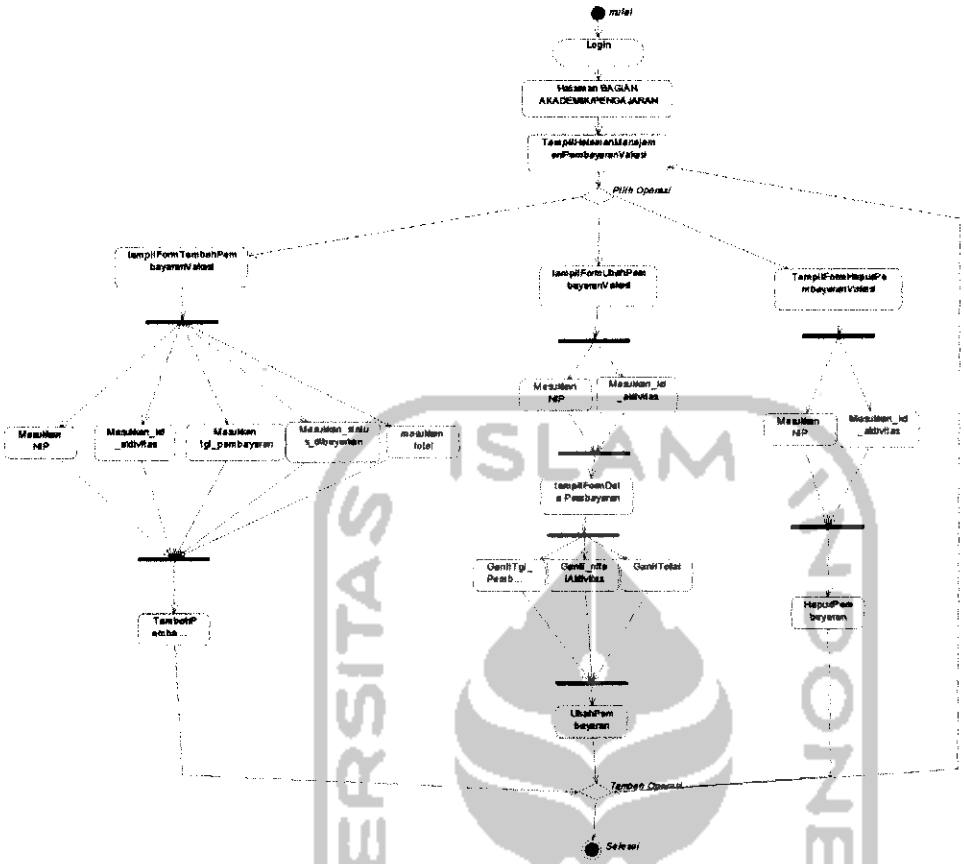
Gambar 4-12 menunjukan aktivitas dalam manajemen pembayaran, yang hanya dapat dilakukan jika *login* dan mempunyai hak akses sebagai DAK atau admin.



Gambar 4-10 *Activity Diagram* Halaman Admin

Dalam *activity diagram* untuk halaman Admin dapat dilihat bahwa Admin memulai aktivitas dengan *login* kemudian setelah *login* berhasil seorang admin mempunyai hak untuk mengakses aktivitas-aktivitas sebagai berikut, manajemen *user*, manajemen perbankan, manajemen pembayaran vakasi, menajemen personal, manajemen *report*, manajemen posttarif, tambah operasional, manajemen surat, manajemen aktivitas, manajemen jurusan, manajemen matakuliah, manajemen mengajar, lihat frekwensi vakasi personal, lihat frekwensi vakasi jurusan, lihat vakasi personal. setelah selesai dengan satu manajemen seorang admin dapat memilih manajemen yang lain atau memilih selesai.


FAKULTAS
JURUSAN
USER
mulai
Login
authentifikasi
Halaman DEKAN
Halaman JURUSAN
pilih operasi
pilih ope...
lihat frekwensi vakasi personal
lihat frekwensi vakasi jurusan
lihat vakasi personal
tambah operasi
tambah operasi
selesai
selesai
selesai
selesai
Selesai



Gambar 4-12 *Activity Diagram* Halaman Manajemen Pembayaran Vakasi.

Pada gambar diatas menjelaskan DAK dalam aktivitas manajemen pembayaran vakasi. DAK harus *login* dahulu setelah *login* berhasil, maka akan tampil halaman manajemen pembayaran vakasi, untuk menambah data DAK dapat memilih *form* tambah pembayaran vakasi dengan memasukan data NIP, kode aktivitas, tanggal pembayaran, total yang dibayarkan, dan status dibayarkan. Terdapat dua pilihan untuk status dibayarkan yaitu dibayarkan tunai (*cash*) atau transfer ke dalam rekening personal. untuk aktivitas ubah atau hapus data pembayaran. DAK harus memasukkan NIP dan kode aktivitas, selanjutnya data akan ditampilkan untuk diubah jika terjadi kesalahan atau dihapus jika data tidak terpakai.

4.2 Perancangan *Database*

Basisdata atau *database* adalah sebuah cara mendokumentasikan berbagai macam data yang kemudian dimanajemen dengan sebuah sistem dalam sebuah media penyimpanan (Nugroho, 2005). Dengan demikian data tersebut dapat diakses dengan mudah dan cepat.

Dalam *database*, data dikelola oleh sebuah sistem pengaturan *database* yang sering disebut *Database Manajement Sistem* (DBMS). Dengan demikian suatu data dengan jumlah besar dan kompleks dapat tersusun sangat baik sehingga memungkinkan pengaksesan data dengan mudah dan cepat oleh pengguna. Dalam *database* aplikasi SIPeV terdapat tabel, kolom, *view*, *stored procedure*, dan *trigger*. Tabel berfungsi untuk menyimpan sebuah kelompok data. Tabel sering disebut sebagai *entitas* atau *entity*. Sedangkan kolom merupakan bentuk terkecil dari *database*, berfungsi untuk menyimpan data. Kolom sering disebut sebagai *attribute* atau *field*. *View* adalah suatu *object database*, yang secara logika merepresentasikan sub himpunan dari data yang berasal dari satu atau lebih tabel (Badriyah,2007). Fungsi dari *view* adalah membuat *query/ syntax* SQL yang kompleks secara mudah. *Stored procedure* atau *trigger* adalah suatu *programing language* atau bahasa pemrograman yang digunakan untuk membuat suatu pemrograman di dalam *database* yang membedakan hanya proses eksekusinya, pada *trigger* akan tereksekusi secara otomatis setelah proses `insert`, `update` atau `delete`, sedangkan *stored procedure* terekseskusi berdasarkan kebutuhan akan proses tersebut. *Stored procedure*, *view* dan *trigger* digunakan untuk mengurangi beban *resource* dalam komputer atau dengan kata lain dapat memperingan kerja aplikasi SIPeV, selain itu fungsi *trigger* yang lain adalah dapat menghemat proses fungsional dalam SIPeV, yaitu pada penghitungan *report*. Statistic *database* `db_sipev` akan digambarkan pada Tabel 4-1.

Keterangan	Jumlah
Table	17
Trigger	17
View	5
Stored Procedure	7

Tabel 4-1 Statistik *database* db_sipev

4.2.1 Perancangan Tabel

Tabel-tabel berikut adalah merupakan tabel-tabel yang ada dalam SIPeV ini:

1. Tabel v_aktivitas,

   Tabel aktivitas digunakan untuk menyimpan data aktivitas dari masing-masing personal yang ada di FTI. Dengan Struktur sebagai berikut:

   Tabel 4-2 Tabel v_aktivas

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	kd_aktivitas	INTEGER	11	0			✓		✓
	kd_posttarif	VARCHAR	14	0		''			✓
	nip	INTEGER	12	0		0			✓
	Keterangan	VARCHAR	70	0					
	nilai	DOUBLE	15	3					
	ket_bonus	VARCHAR	70	0					
	tgl_aktivitas	DATE	0	0					
	bonus	DOUBLE	15	3					
	no_surat	VARCHAR	20	0					
	total	DOUBLE	15	3		0.000			✓
	bayar	ENUM	0	0	('yes','no')	no			✓
	status	INTEGER	1	0					

Indices

Name	Fields	Unique	Collation
PRIMARY	kd_aktivitas	✓	Ascending
nip	nip		Ascending

2. Tabel v_bank

   Tabel v_bank digunakan untuk menyimpan data bank setiap personal dalam lingkungan Fakultas Teknologi Industri. Struktur yang berada dalam tabel v_bank adalah sebagai berikut:

Tabel 4-3 Tabel v_bank

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	kd_bank	CHAR	3	0					✓
	nm_bank	VARCHAR	50	0					✓

Indices

Name	Fields	Unique	Collation
PRIMARY	kd_bank	✓	Ascending
kd_bank	kd_bank	✓	Ascending

3. Tabel v_jurusan

   Tabel v_jurusan digunakan untuk menyimpan data jurusan, dan Struktur dari tabel v_jurusan adalah sebagai berikut:

Tabel 4-4 Tabel v_jurusan

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	kd_jurusan	INTEGER	4	0		0			✓
	nm_jurusan	VARCHAR	30	0		''			✓

Indices

Name	Fields	Unique	Collation
PRIMARY	kd_jurusan	✓	Ascending

4. Tabel v_matakuliah

   Tabel v_matakuliah digunakan untuk menyimpan data matakuliah (id matakuliah, kode matakuliah dan kelas) dalam Fakultas Teknologi Industri.

Tabel 4-5 Tabel v_matakuliah

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	id_matakuliah	VARCHAR	20	0		NULL			✓
	kd_matakuliah	INTEGER	11	0		0			✓
	kelas	CHAR	2	0					✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id_matakuliah	✓	Ascending
id_matakuliah	id_matakuliah	✓	Ascending

5. Tabel v_detil_matkul

Tabel v_detil_matkul digunakan untuk menyimpan data detil matakuliah.

Tabel 4-6 Tabel v_detil_matkul

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	kd_matakuliah	INTEGER	11	0					✓
	nm_matakuliah	VARCHAR	200	0					✓
	kd_jurusan	INTEGER	4	0					
	sks	INTEGER	2	0					
	semester	INTEGER	2	0					
	tahun	YEAR	4	0					

Indices

Name	Fields	Unique	Collation
PRIMARY	kd_matakuliah	✓	Ascending

6. Tabel v_mengajar

Tabel v_mengajar digunakan untuk menyimpan data mengajar dosen terhadap matakuliah yang diampunya.

Tabel 4-7 Tabel v_mengajar

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	id_mengajar	VARCHAR	20	0		NULL			✓
	nip	INTEGER	14	0		0			✓
	id_matakuliah	VARCHAR	20	0		0			✓
	jml_mahasiswa	INTEGER	11	0					
	jml_hadir	INTEGER	11	0					
	kd_posttarif	INTEGER	11	0					✓
	total	DOUBLE	15	3					

Indices

Name	Fields	Unique	Collation
PRIMARY	id_mengajar	✓	Ascending
id_mengajar	id_mengajar	✓	Ascending

7. Tabel `v_personal`

Tabel `v_personal` digunakan untuk menyimpan data personal, Struktur yang berada dalam tabel personal:

Tabel 4-8 Tabel `v_personal`

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null	U
✓	nip	INTEGER	14	0			✓		✓	
	nama	VARCHAR	30	0		''			✓	
	gelar	VARCHAR	20	0						
	pendidikan	VARCHAR	25	0						
	golongan	VARCHAR	30	0						
	j_akademik	VARCHAR	35	0						
	status	VARCHAR	30	0						
	tempat_lahir	VARCHAR	20	0						
	tgl_lahir	DATE	0	0		1988-12-02				
	jns_kelamin	ENUM	0	0	('L','P')	L			✓	
	alamat	VARCHAR	50	0						
	telpon	VARCHAR	20	0						
	kd_bank	CHAR	3	0						
	no_rek	INTEGER	14	0						
	kd_jurusan	INTEGER	4	0						

Indices

Name	Fields	Unique	Collation
PRIMARY	nip	✓	Ascending

8. Tabel `v_posttarif`

Tabel `v_posttarif` digunakan untuk menyimpan data post tarif, berdasar posttarif yang telah ditetapkan.

Tabel 4-9 Tabel `v_posttarif`

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	kd_posttarif	INTEGER	11	0			✓		✓
	keterangan	VARCHAR	50	0					✓
	tarif	INTEGER	11	0					
	satuan	INTEGER	11	0					
	ket_satuan	VARCHAR	20	0					
	pajak	DOUBLE	15	3					
	semester	INTEGER	1	0					✓
	tahun	YEAR	4	0					✓

Indices

Name	Fields	Unique	Collation
PRIMARY	kd_posttarif	✓	Ascending

9. Tabel `v_surat`

Tabel surat `v_surat` digunakan untuk menyimpan data keseluruhan surat, baik surat tugas, surat keputusan ataupun surat perjalanan dinas.

Tabel 4-10 Tabel `v_surat`

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	no_surat	VARCHAR	30	0		''			✓
	jns_surat	VARCHAR	40	0		0			✓
	tgl_surat	DATE	0	0					
	tugas	TEXT	0	0					
	waktu	VARCHAR	20	0					

Indices

Name	Fields	Unique	Collation
PRIMARY	no_surat	✓	Ascending

10. Tabel `v_detil_stk`

Tabel `v_detil_stk` digunakan untuk menyimpan data mengenai rincian dari surat tugas dan surat keputusan, sebab kedua surat tersebut mempunyai struktur pengisian data yang sama, oleh karena itu dapat digabungkan menjadi satu tabel.

Tabel 4-11 Tabel `v_detil_stk`

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	id_detil_stk	INTEGER	11	0			✓		✓
	kd_posttarif	INTEGER	11	0					
	no_surat	VARCHAR	30	0					
	nip	INTEGER	14	0					
	jabatan	VARCHAR	30	0					✓
	penerimaan	DOUBLE	15	3					

Indices

Name	Fields	Unique	Collation
PRIMARY	id_detil_stk	✓	Ascending

11. Tabel `v_detil_spj`

Tabel `v_detil_spj` digunakan untuk menyimpan data mengenail surat perjalanan dinas. Surat perjalan dinas memiliki struktur pengisian data yang berbeda dengan surat keputusan ataupun surat perjalanan dinas, oleh karena itu tabel surat perjalanan dinas harus berdiri sendiri tidak bisa digabungkan dengan surat tugas atau surat keputusan.

Tabel 4-12 Tabel v_detil_spj

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	id_detil_spj	INTEGER	11	0			✓		✓
	no_surat	VARCHAR	30	0					✓
	nip	INTEGER	14	0					✓
	kota	VARCHAR	20	0					✓
	berangkat	DATE	0	0					
	kembali	DATE	0	0					
	total	DOUBLE	15	3					

Indices

Name	Fields	Unique	Collation
PRIMARY	id_detil_spj	✓	Ascending

12. Tabel v_akomodasi_spj

Tabel v_akomodasi_spj digunakan untuk menyimpan data rincian biaya akomodasi yang dikeluarkan untuk surat perjalan dinas.

Tabel 4-13 Tabel v_akomodasi_spj

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	id_akomodasi_spj	INTEGER	11	0			✓		✓
	kd_posttarif	INTEGER	11	0					
	id_detil_spj	INTEGER	11	0					✓
	penerimaan	DOUBLE	15	3					✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id_akomodasi_spj	✓	Ascending

13. Tabel v_pembayaran

Tabel v_pembayaran digunakan untuk menyimpan data pembayaran yang dilakukan oleh bagian keuangan.

Tabel 4-14 Tabel v_pembayaran

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	id_pembayaran	INTEGER	11	0			✓		✓
	kd_aktivitas	VARCHAR	20	0					
	tgl_bayar	DATE	0	0					
	total_bayar	DOUBLE	0	0					
	dbayar	ENUM	0	0	('cash','transfer')				✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id_pembayaran	✓	Ascending

14. Tabel `v_user`

Tabel `v_user` digunakan untuk menyimpan data *user* dan hak akses atau *priviledge user* yang digunakan untuk proses *authentivikasi* dalam *login*.

Tabel 4-15 Tabel `v_user`

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✔	id_user	INTEGER	11	0			✔		✔
	username	VARCHAR	30	0					
	password	VARCHAR	30	0					
	hak_akses	VARCHAR	30	0					
	user	ENUM	0	0	('enable','disable')	disable			✔
	bank	ENUM	0	0	('enable','disable')	disable			✔
	bayar	ENUM	0	0	('enable','disable')	disable			✔
	personal	ENUM	0	0	('enable','disable')	disable			✔
	report	ENUM	0	0	('enable','disable')	disable			✔
	general	ENUM	0	0	('enable','disable')	disable			✔
	posttarif	ENUM	0	0	('enable','disable')	disable			✔
	surat	ENUM	0	0	('enable','disable')	disable			✔
	aktivitas	ENUM	0	0	('enable','disable')	disable			✔
	jurusan	ENUM	0	0	('enable','disable')	disable			✔
	mengajar	ENUM	0	0	('enable','disable')	disable			✔
	matakuliah	ENUM	0	0	('enable','disable')	disable			✔
	frek_personal	ENUM	0	0	('enable','disable')	disable			✔
	frek_jurusan	ENUM	0	0	('enable','disable')	disable			✔
	vak_personal	ENUM	0	0	('enable','disable')	disable			✔

Indices

Name	Fields	Unique	Collation
PRIMARY	id_user	✔	Ascending

15. Tabel v_*setting*

Tabel v_*setting* digunakan untuk menyimpan aturan yang berjalan dalam sistem SIPeV, baik pengaturan semester dan tahun yang berjalan dalam sistem SIPeV.

Tabel 4-16 Tabel `v_setting`

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✔	uid_setting	INTEGER	11	0			✔		✔
	tahun	VARCHAR	20	0					
	semester	VARCHAR	20	0					
	last_database	DATETIME	0	0					
	last_periode	DATETIME	0	0					

Indices

Name	Fields	Unique	Collation
PRIMARY	uid_setting	✔	Ascending

16. Tabel `v_profile`

    Tabel `v_profile` digunakan untuk menyimpan profile FTI UII, seperti data alamat, nomor telepon, faxcimile, dan lain lain.

    Tabel 4-17 Tabel `v_profile`

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	uid_profile	INTEGER	11	0			✓		✓
	nm_profile	VARCHAR	30	0					✓
	almt_profile	VARCHAR	100	0					✓
	kota_profile	VARCHAR	20	0					✓
	telp_profile	VARCHAR	25	0					
	fax_profile	VARCHAR	25	0					
	gbr_profile	VARCHAR	100	0					

Indices

Name	Fields	Unique	Collation
PRIMARY	uid_profile	✓	Ascending

17. Tabel `v_log_aktivitas`

    Tabel `v_log_aktivitas` digunakan untuk menyimpan *log* (sejarah) aktivitas yang dilakukan oleh user terhadap sistem SIPeV.

    Tabel 4-18 Tabel `v_log_aktivitas`

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null	I
✓	uid_lok_aktivitas	INTEGER	11	0			✓		✓	
	waktu	TIMESTAMP	0	0		CURRENT_TIMESTAMP				
	user	VARCHAR	70	0						
	aktivitas	VARCHAR	200	0						

Indices

Name	Fields	Unique	Collation
PRIMARY	uid_lok_aktivitas	✓	Ascending

4.2.2 Relasi Tabel

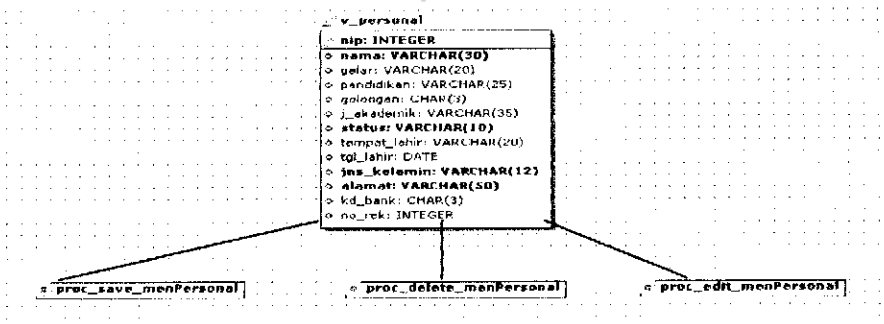
Relasional atau relasi merupakan bentuk hubungan antara dua tabel atau lebih. Dengan relasi maka sebuah data dapat disimpan dengan struktur yang terorganisir sehingga memudahkan dalam memasukkan informasi dan mengelola *database*-nya. Dalam relasi tabel dapat menunjukan pula relasi atau hubungan data yang terdapat dalam masing-masing tabel. Relasi tabel hanya menunjukan

relasi antar tabel, bukan termasuk hubungan antara tabel dengan *stored procedure*, sebab *stored procedure* berdiri sendiri dan berada di luar tabel, tetapi berhubungan dengan *trigger*. *Trigger* berada di dalam tabel masing-masing sesuai dengan aksi (*insert*, *update*, *delete*) yang dilakukan. Dalam perancangan tidak terdapat gambar mengenai relasi tabel dikarenakan tabel-tabel yang berelasi telah dirancang sesuai dengan *Class Object* Berorientasi D*atabase* yang terdapat pada bagian *class diagram* diatas.

4.2.3 Perancangan *Stored Procedure*

Stored procedure digunakan sebagai jembatan antara aplikasi dengan database untuk membawa data, sehingga pemrosesan akan lebih ringan, dan kinerja aplikasi tidak berat.

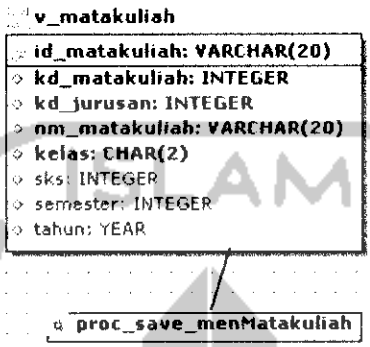
1. *Stored Procedure* terkait dengan tabel `v_personal`.
   Stored procedure yang digunakan terdapat tiga macam, `proc_save_menpersonal` adalah *stored procedure* untuk menyimpan data ke dalam table `v_personal`, `proc_delete_menPersonal` adalah *stored procedure* untuk menghapus data ke dalam table `v_personal`, dan `proc_edit_menPersonal` adalah *stored procedure* untuk mengubah data ke dalam table `v_personal`, ketiga *stored procedure* yang terkait dengan tabel `v_personal` akan digambarkan pada Gambar 4-13.



Gambar 4-13 *Stored Procedure* Terkait Dengan Tabel `v_personal`

2. *Stored Procedure* terkait dengan tabel `v_matakuliah`.

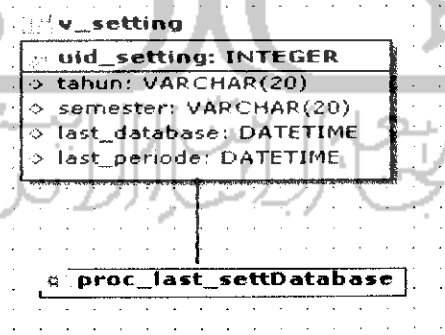
   Stored Procedure yang digunakan untuk menyimpan data matakuliah kedalam tabel `v_matakuliah`. *Stored Procedure* terkait dengan tabel `v_matakuliah` akan digambarkan pada Gambar 4-14.



Gambar 4-14 *Stored Procedure* Terkait Dengan Tabel `v_matakuliah`

3. *Stored Procedure* terkait dengan tabel `v_setting`

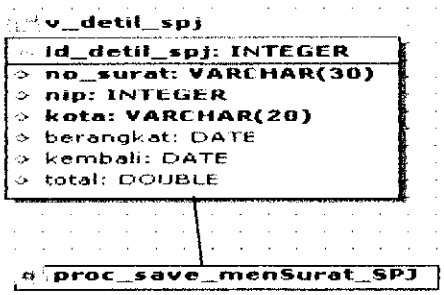
   Stored Procedure yang digunakan untuk menyimpan data *setting* sistem kedalam tabel `v_setting`. *Stored Procedure* terkait dengan tabel `v_setting` akan digambarkan pada Gambar 4-15.



Gambar 4-15 *Stored Procedure* terkait dengan tabel `v_setting`

4. *Stored Procedure* Terkait Dengan Tabel `v_detil_spj`

   Stored Procedure yang digunakan untuk menyimpan data *setting* sistem kedalam tabel `v_detil_spj`. *Stored Procedure* terkait dengan tabel `v_detil_spj` akan digambarkan pada Gambar 4-16.



Gambar 4-16 *Stored Procedure* Terkait Dengan Tabel `v_detil_spj`

4.3 Perancangan *Interface*

sebuah sistem yang bagus harus didukung dengan interface atau antarmuka yang bagus, sehingga pengguna atau *user* mudah menggunakan sistem tersebut.

1. Halaman *Splash Screen*

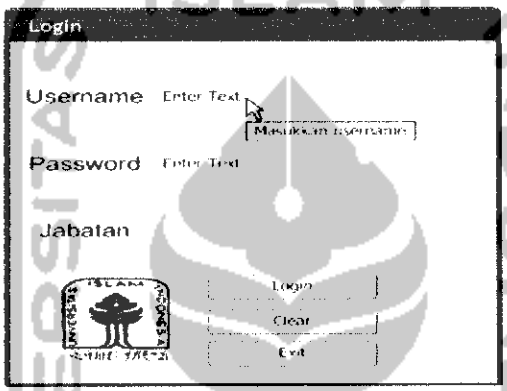
   Halaman ini adalah halaman pertama ketika sistem dijalankan, pada bagian sistem akan mengecek semua *runtimes* (komponen yang mendukung) dan juga akan mengecek data dan database. Rancangan antarmuka halaman *spalsh screen* dapat dilihat pada Gambar 4-17.

Gambar 4-17 Halaman *Splash Screen*

Halaman ini hanya berisikan visualisasi *progress bar* yang menunjukan proses pengecekan terhadap komponen, data dan database yang diperlukan oleh sistem untuk menjalankan program selanjutnya.

2. Halaman *Login*

Halaman *login* merupakan sebuah pintu gerbang untuk masuk kedalam sistem sebagai *user* yang memiliki hak akses masing-masing seperti: DAU, DAP, DAK, jurusan, fakultas dan admin. Rancangan antarmuka untuk halaman *login* dapat dilihat pada gambar. *Form* jabatan akan secara otomatis terisi berdasarkan *username* yang diketikan telah terdaftar sebelumnya sebagai *user* aplikasi SIPeV ini. Rancangan antarmuka halaman *login* dapat dilihat pada Gambar 4-18.

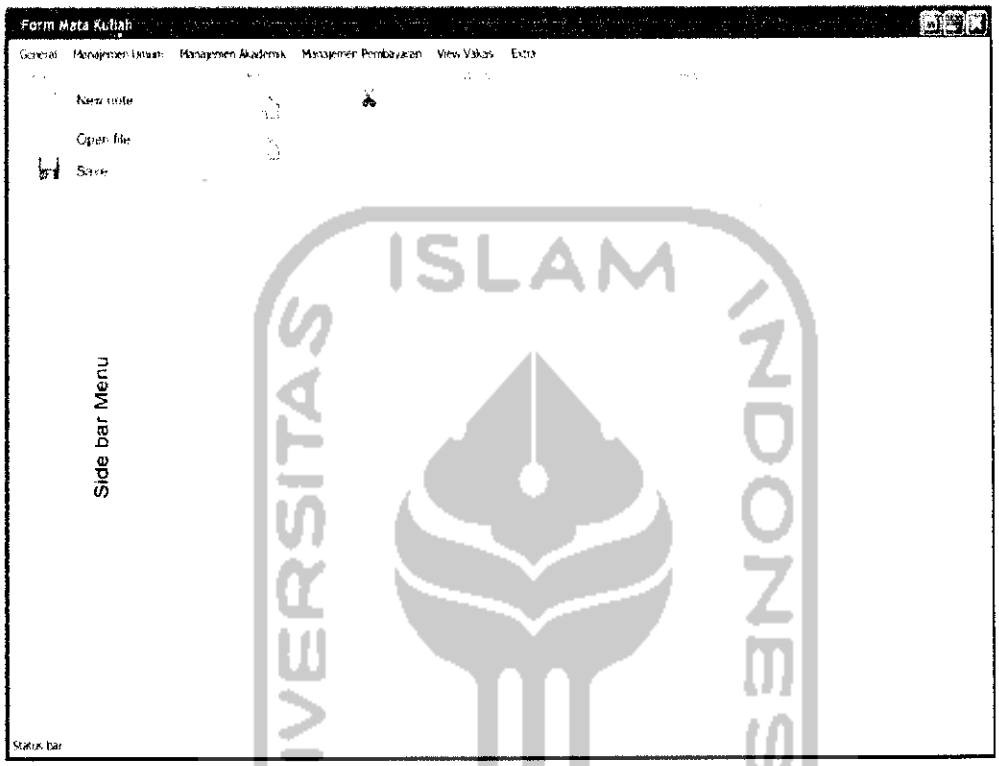


Gambar 4-18 Halaman *Login*

3. Halaman *main menu* (Menu Utama).

Halaman ini dapat diakses setelah *user* berhasil melakukan *login*. Halaman ini mempunyai *group menu* yang berisikan menu-menu untuk mengakses halaman-halaman sesuai dengan hak akses yang telah diberikan oleh administrator. *Group menu general* terdapata menu *file*, edit, *windows*, *close/ exit*, *help*. *Group menu* manajemen umum terdapat menu manajemen personal, manajemen *user*, manajemen surat. *Group menu* manajemen akademik terdapat menu manajemen matakuliah, manajemen mengajar, manajemen jurusan. *Group menu* pembayaran terdapat menu manajemen post-tarif, manajemen aktivitas, manajemen bank, manajemen pembayaran vakasi, manajemen *report*. *Group menu view* vakasi terdapat menu *view frekwensi* vakasi jurusan, *view frekwensi* vakasi personal, *view* vakasi

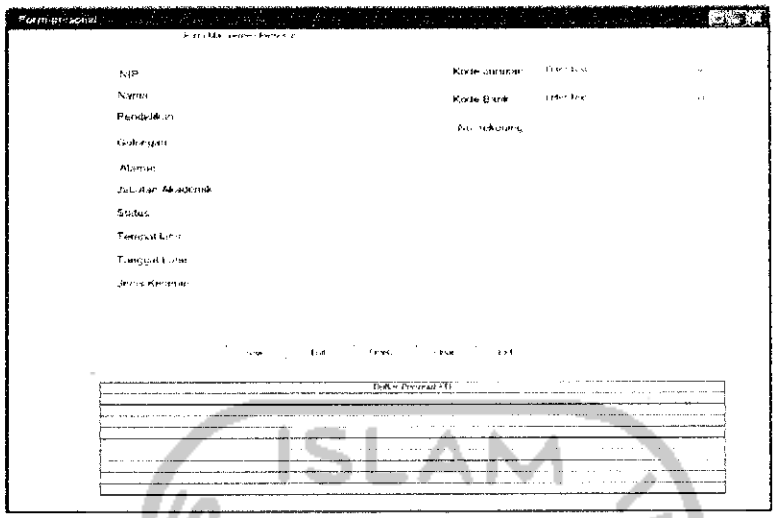
personal. *Group menu extra* terdapat menu manajemen profil FTI, manajemen *setting* sistem, *about box*. Rancangan antarmuka halaman manajemen personal dapat dilihat pada Gambar 4-19.



Gambar 4-19 Halaman *Main Menu*

4. Halaman Manajemen Personal

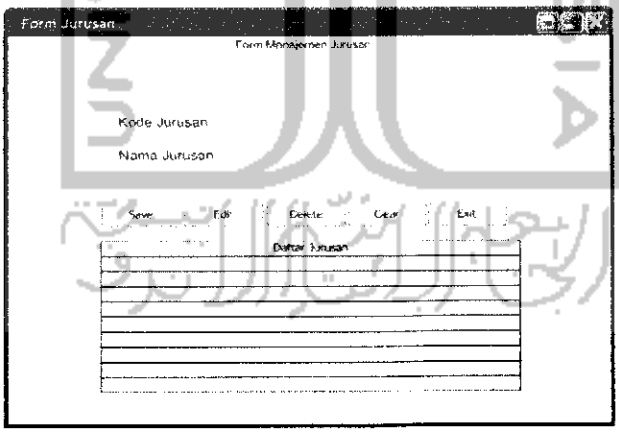
Rancangan antarmuka halaman manajemen personal hanya dapat diakses oleh DAU (Divisi Administrasi Umum), dalam halaman ini terdapat tombol *save* untuk menyimpan data, tombol edit untuk mengubah data, atau tombol *delete* untuk menghapus data NIP, nama, pendidikan, golongan, alamat, jabatan struktural, status dosen, tempat lahir, tanggal lahir, jenis kelamin, kode jurusan, kode bank, nomoe rekening. Rancangan antarmuka halaman manajemen personal dapat dilihat pada Gambar 4-20.



Gambar 4-20 Halaman Manajemen Personal

5. Halaman Manajemen Jurusan

Pada halaman ini *user* yang mendapat hak akses Divisi Administrasi Perkuliahan & akademik (DAP) dapat menambahkan, mengubah atau menghapus data jurusan yang ada di FTI dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Rancangan antarmuka halaman manajemen jurusan dapat dilihat pada Gambar 4-21.

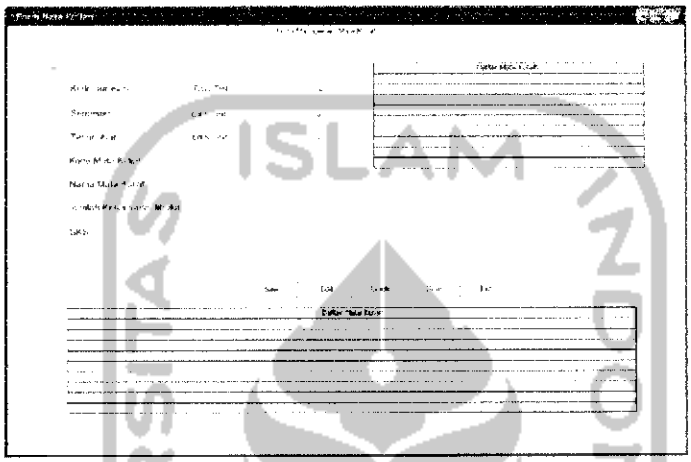


Gambar 4-21 Halaman Menajemen Jurusan

6. Halaman Manajemen Matakuliah

Pada halaman ini *user* yang mendapat hak akses Divisi Administrasi Perkuliahan & akademik (DAP) dapat menambahkan, mengubah atau menghapus data matakuliah yang ada di FTI dengan menekan tombol *save*,

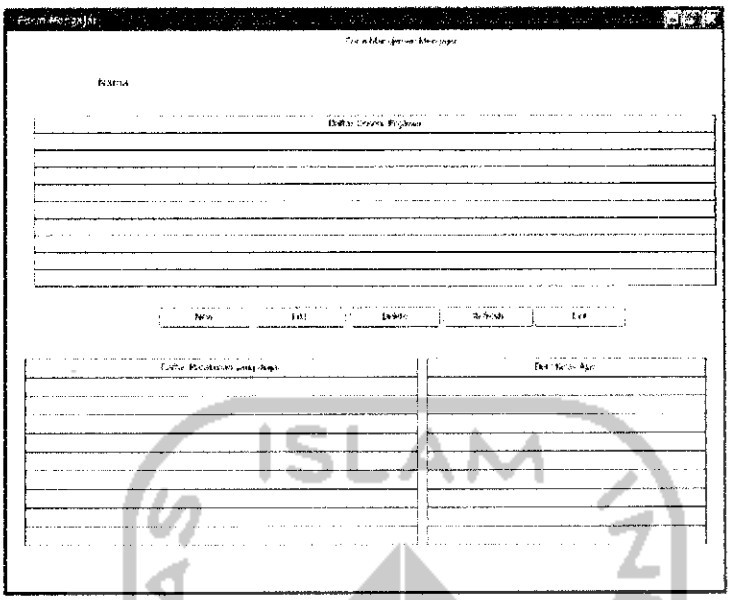
edit atau *delete* yang berada dalam halaman ini. Halaman ini memiliki 2 *data grid*, *data grid* yang pertama menampilkan data matakuliah berdasarkan kode jurusan yang dipilih, sedangkan *data grid* yang kedua menampilkan seluruh data matakuliah yang sudah berada dalam *database*. Rancangan antarmuka halaman manajemen matakuliah dapat dilihat pada Gambar 4-22.



Gambar 4-22 Halaman Manajemen Matakuliah

7. Halaman Manajemen Mengajar

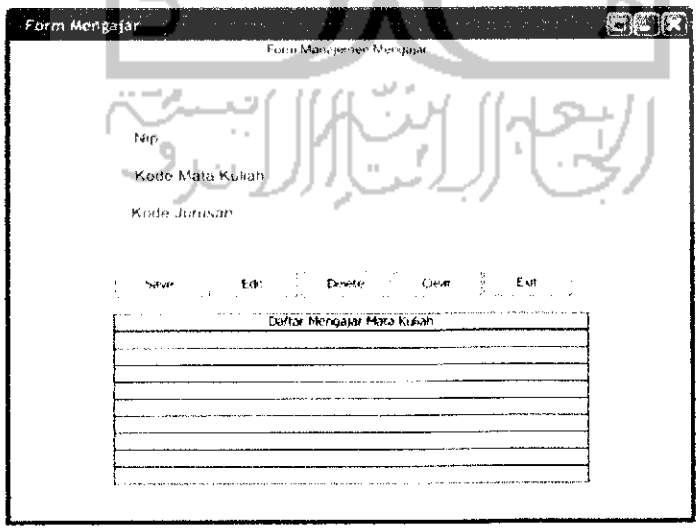
Pada halaman ini *user* yang mendapat hak akses Divisi Administrasi Perkuliahan & akademik (DAP) dapat menambahkan, mengubah atau menghapus data mengajar yang ada di FTI dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Halaman ini memiliki 3 *data grid*, *data grid* yang pertama menampilkan data dosen, *data grid* yang kedua menampilkan seluruh data matakuliah yang diampu oleh dosen yang telah berada dalam *database*, sedangkan *data grid* yang ketiga adalah data detil kelas yang telah dimasukkan sebelumnya kedalam *database* berdasarkan matakuliah yang diampu oleh dosen. Rancangan antarmuka halaman manajemen mengajar dapat dilihat pada Gambar 4-23.



Gambar 4-23 Halaman Manajemen Mengajar

8. Halaman Tambah dan Edit Mengajar

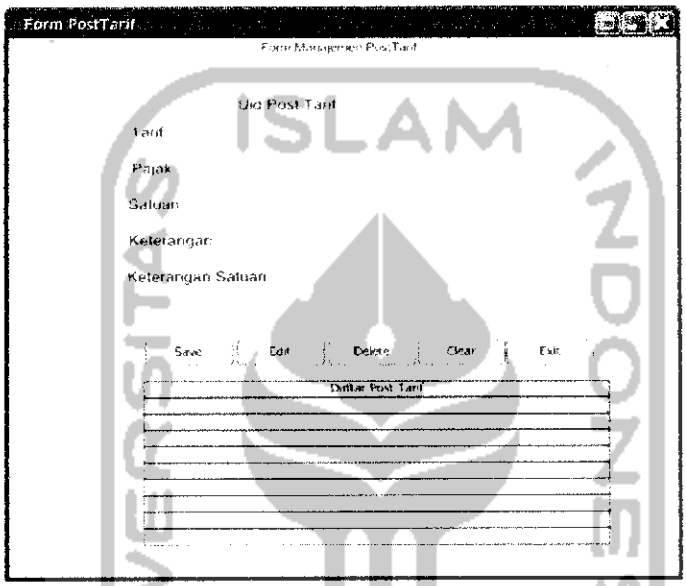
Halaman ini digunakan untuk memasukkan atau mengubah data mengajar dan menyimpan data tersebut kedalam *data base*. Halaman ini merupakan kelanjutan dari halaman manajemen mengajar. Setelah tombol *new* atau edit ditekan. Rancangan antarmuka halaman manajemen mengajar dapat dilihat pada Gambar 4-24.



Gambar 4-24 Halaman Tambah dan Edit Mengajar

9. Halaman Manajemen Post-tarif

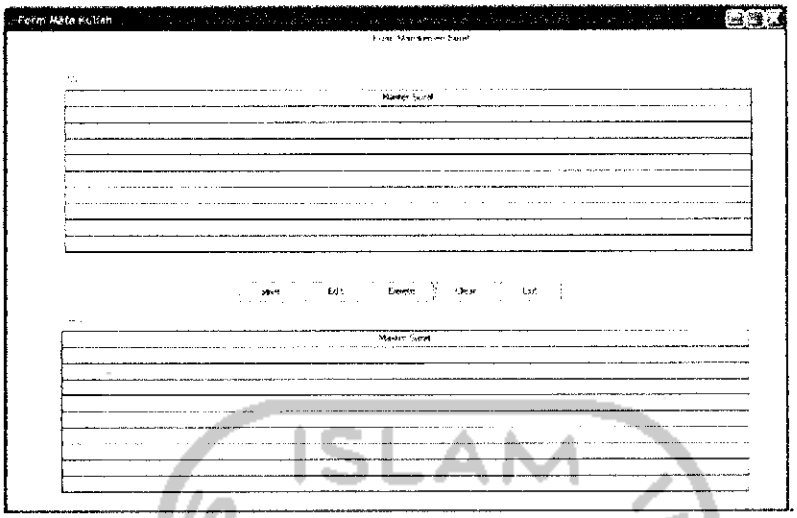
Pada halaman ini *user* yang mendapat hak akses Divisi Administrasi Keuangan (DAK) dapat menambahkan, mengubah atau menghapus data post-tarif yang ada di FTI dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Rancangan antarmuka halaman manajemen jurusan dapat dilihat pada Gambar 4-25.



Gambar 4-25 Halaman Manajemen Post-tarif

10. Halaman Manajemen Surat

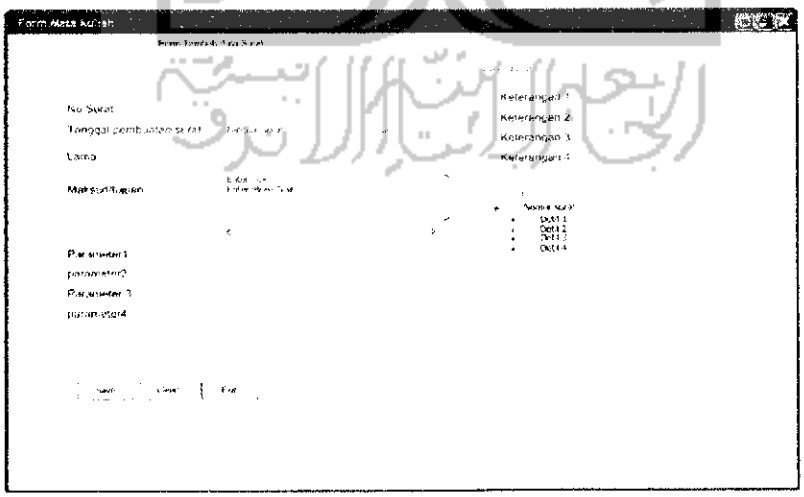
Pada halaman ini *user* yang mendapat hak akses Divisi Administrasi Umum (DAU) dapat menambahkan, mengubah atau menghapus data surat yang ada di FTI dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Halaman ini memiliki 2 *data grid*, *data grid* yang pertama menampilkan data surat, sedangkan *data grid* yang kedua menampilkan data detil surat berdasarkan jenis suratnya yang sudah berada dalam *database*. Rancangan antarmuka halaman manajemen surat dapat dilihat pada Gambar 4-26



Gambar 4-26 Halaman Manajemen Surat.

11. Halaman Tambah dan Edit Surat

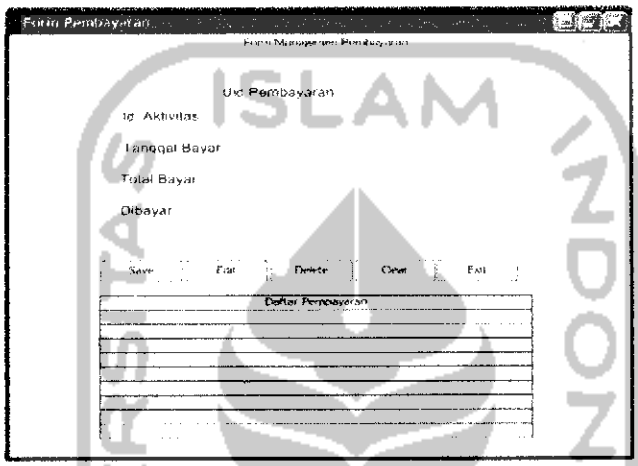
Halaman ini digunakan untuk memasukkan atau mengubah data surat dan menyimpan data tersebut kedalam *data base*. Halaman ini merupakan kelanjutan dari halaman manajemen surat. Setelah tombol *new* atau edit ditekan. Pada halaman ini DAU dapat menambah atau mengubah data Surat dan secara otomatis akan menambahkan kedalam data pembayaran yang mengacu pada NIP masing-masing personal. Rancangan antarmuka halaman manajemen mengajar dapat dilihat pada Gambar 4-27.



Gambar 4-27 Halaman Tambah dan Edit Surat

12. Halaman Manajemen Pembayaran

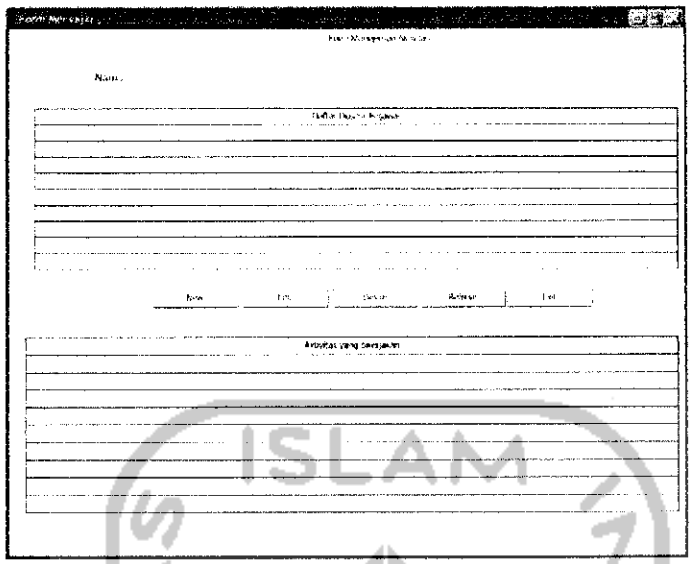
Pada halaman ini *user* yang mendapat hak akses Divisi Administrasi Keuangan (DAK) dapat menambahkan, mengubah atau menghapus data pembayaran yang ada di FTI dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Rancangan antarmuka halaman manajemen pembayaran dapat dilihat pada Gambar 4-28.



Gambar 4-28 Halaman Manajemen Pembayaran

13. Halaman Manajemen Aktivitas.

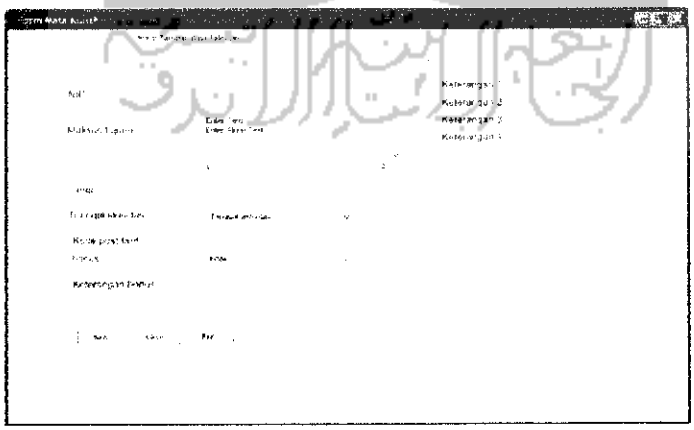
Pada halaman ini *user* yang mendapat hak akses untuk mengakses halaman ini dapat menambahkan, mengubah atau menghapus data aktivitas yang ada di FTI dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Halaman ini memiliki 2 *data grid*, *data grid* yang pertama menampilkan data personal, sedangkan *data grid* yang kedua menampilkan data aktivitas yang dilakukan berdasarkan NIP yang sudah berada dalam *database*. Rancangan antarmuka halaman manajemen aktivitas dapat dilihat pada Gambar 4-29.



Gambar 4-29 Halaman Manajemen Aktivitas

14. Halaman tambah dan edit data aktivitas

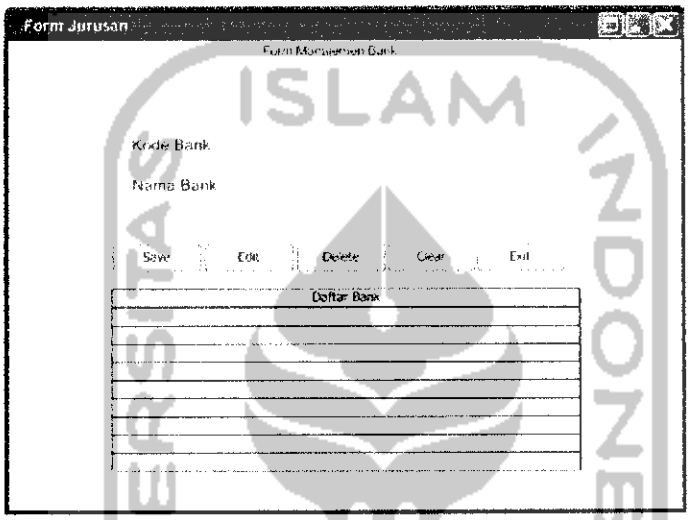
    Halaman ini digunakan untuk memasukkan atau mengubah data aktivitas dan menyimpan data tersebut kedalam *data base*. Halaman ini merupakan kelanjutan dari halaman manajemen aktivitas. Setelah tombol *new* atau edit ditekan. Pada halaman ini *user* dapat menambah atau mengubah data aktivitas dan secara otomatis akan menambahkan kedalam data pembayaran yang mengacu pada NIP masing-masing personal. Rancangan antarmuka halaman manajemen tambah dan edit data aktivitas dapat dilihat pada Gambar 4-30.



Gambar 4-30 Halaman Tambah dan Edit Data Aktivitas

15. Halaman Manajemen Bank

Pada halaman ini *user* yang mendapat hak akses Divisi Administrasi Keuangan (DAK) dapat menambahkan, mengubah atau menghapus data bank yang ada di FTI dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Rancangan antarmuka halaman manajemen bank dapat dilihat pada Gambar 4-31.



Gambar 4-31 Halaman Manajemen Bank

16. Halaman Manajemen *User*

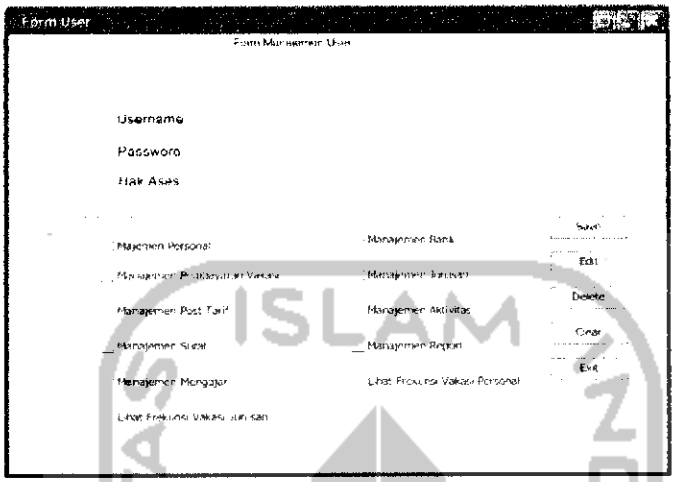
Pada halaman ini *user* yang mendapat hak akses sebagai administrator dapat menambahkan, mengubah atau menghapus data *user* dengan hak akses masing-masing bagian sesuai dengan kebutuhannya dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Rancangan antarmuka halaman manajemen user dapat dilihat pada Gambar 4-32.

17. Halaman Manajemen *Report*

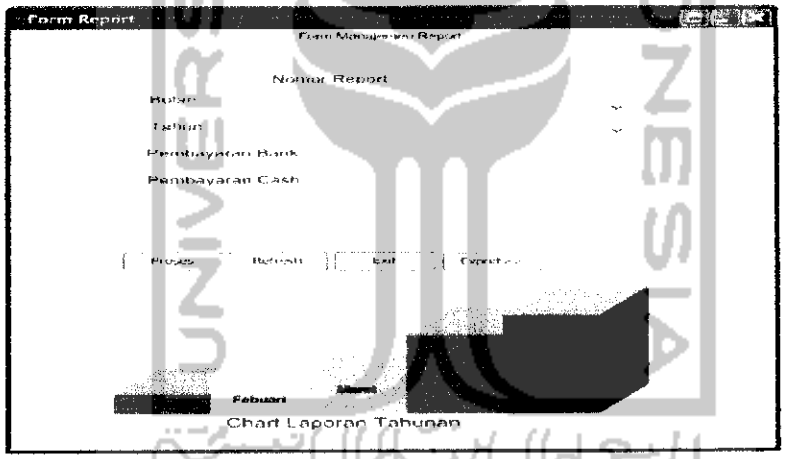
Pada halaman manajemen *report* dapat dibagi menjadi 2 ketegori, yaitu:

a) Halaman Manajemen *Report* untuk laporan vakasi tahunan digunakan *user* yang mendapat hak akses Divisi Administrasi Keuangan (DAK) untuk melihat *report* tahunan dan bulanan sesuai vakasi yang telah dibayarkan pada bulan dan tahun yang telah dipilih. Rancangan

antarmuka halaman manajemen *report* laporan tahunan dapat dilihat pada Gambar 4-33.

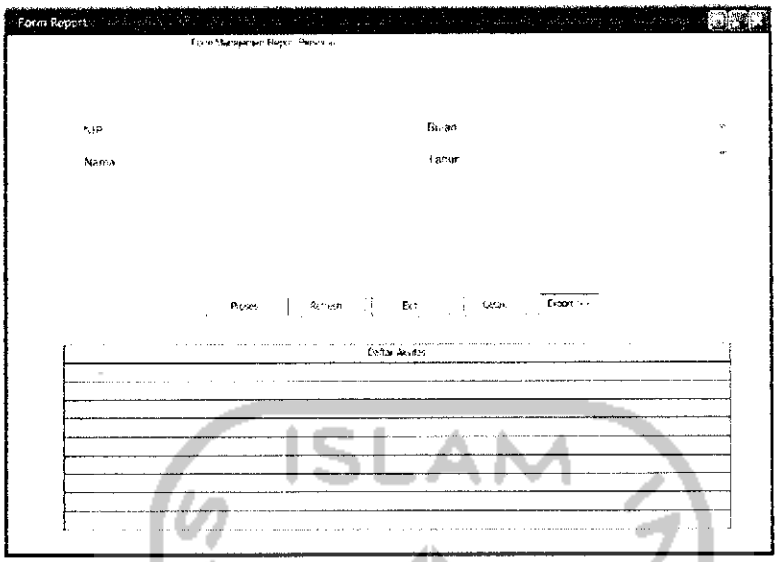


Gambar 4-32 Halaman Manajemen *User*



Gambar 4-33 Halaman Manajemen Report.

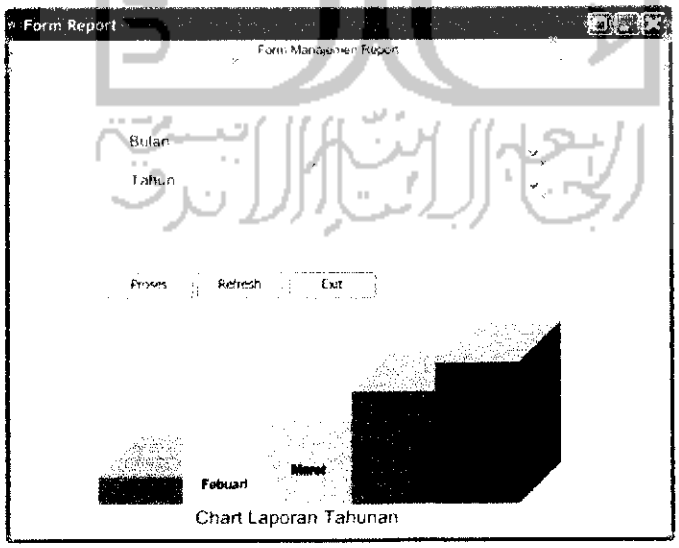
b) Halaman Manajemen *report* vakasi personal digunakan untuk laporan digunakan *user* yang mendapat hak akses Divisi Administrasi Keuangan (DAK) untuk melihat dan mencetak dan *report* bulanan sesuai vakasi yang telah dibayarkan kepada personal pada bulan yang telah dipilih. Rancangan antarmuka halaman manajemen bank dapat dilihat pada Gambar 4-34.



Gambar 4-34 Halaman Manajemen *Report* Personal.

13 Halaman *View* Vakasi

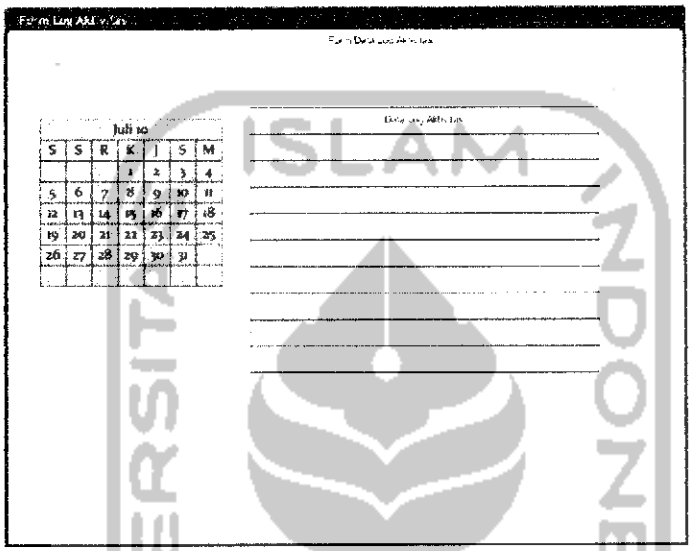
Halaman Manajemen *view* vakasi digunakan *user* yang mendapat hak akses jurusan atau fakultas untuk melihat total vakasi *frekwensi* semua jurusan atau personal dan melihat detil vakasi personal sesuai vakasi yang telah dibayarkan kepada personal pada bulan yang telah dipilih. Rancangan antarmuka *view* vakasi dapat dilihat pada Gambar 4-35.



Gambar 4-35 Halaman *View* Vakasi.

14. Halaman *Log* Aktivitas

Halaman *log* aktivitas digunakan bagi *user* yang mendapat hak akses sebagai administrator untuk melihat *log* (sejarah) aktivitas user dalam memakai SIPeV ini. Rancangan antarmuka *view* vakasi dapat dilihat pada Gambar 4-36.



Gambar 4-36 Halaman Log Aktivitas

UNIVERSITAS ISLAM INDONESIA