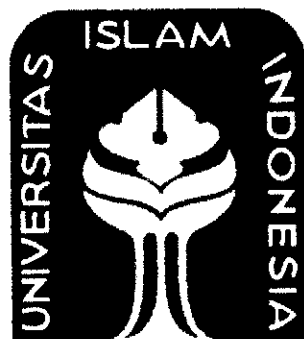


**SISTEM INFORMASI PEMBAYARAN VAKASI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh :

Nama : Firmadi Biastiko

No. Mahasiswa : 06 523 186

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2010

LEMBAR PENGESAHAN PEMBIMBING

SISTEM INFORMASI PEMBAYARAN VAKASI

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA



TUGAS AKHIR

Oleh :

Nama : Firmadi Biastiko

No. Mahasiswa : 06 523 186

الإسلامية
الجامعة الإسلامية
Pembimbing

Zainudin Zuhri, ST., M.I.T

LEMBAR PENGESAHAN PENGUJI

SISTEM INFORMASI PEMBAYARAN VAKASI FAKULTAS TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM INDONESIA TUGAS AKHIR

Oleh:
Nama : Firmadi Biastiko
No. Mahasiswa : 06 523 186

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 30 Juni 2010

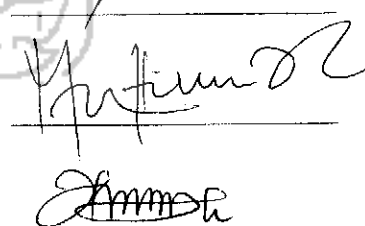
Tim Penguji

Tanda Tangan

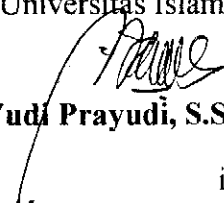
Zainudin Zuhri, ST., MIT.
Ketua Penguji

Izzati Muhimmah, ST., M.Sc., Ph. D.
Anggota Penguji

Hendrik, ST., M. Eng.
Anggota Penguji



Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia


(Yudi Prayudi, S.Si., M.Kom.)

HALAMAN PERSEMBAHAN

Mulailah segala sesuatu dengan Basmalah

Kupersembahkan Tugas Akhir Ini

Dengan Setulus Hatiku Untuk

Yang Tercinta :

Ayah (Turzani) dan Ibu (Ariyanti) Tercinta

Serta Adik-ku Cahyo Gunawan

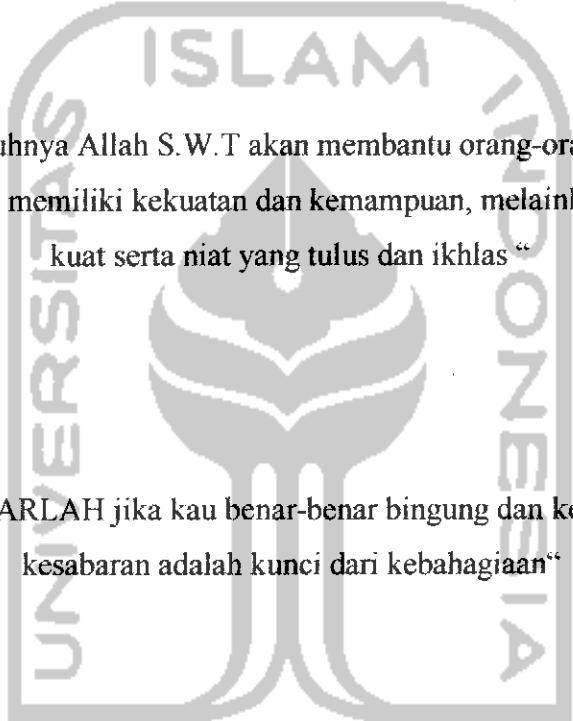
*Yang Selalu Memberikan Do'a, Semangat, Pengorbanan dan
Dukungan Yang Tiadatara Selama Dalam Menyelesaikan Studi*

Akhirilah segala sesuatu dengan Hamdalah

HALAMAN MOTTO

“Sesungguhnya setelah kesulitan tersimpan sebuah kemudahan”

(QS Alam Nasyrh:6)



“ Sesungguhnya Allah S.W.T akan membantu orang-orang yang berusaha, sekalipun ia tidak memiliki kekuatan dan kemampuan, melainkan kemauan yang kuat serta niat yang tulus dan ikhlas “

“ SABARLAH jika kau benar-benar bingung dan kerepotan, sebab kesabaran adalah kunci dari kebahagiaan“



“ Ridho Ibu merupakan rezeki yang tidak ada nilainya “

KATA PENGANTAR



Assalamu'alaika yaa Nabi Allah

Assalamu'alaika yaa Rasulullah

Assalamu'alaika yaa Habbibullah

Assalamu'alaika ayyuhan nabiyyu warohmatullahi wabarokatuh

Dengan nama Allah Yang Maha Pengasih Lagi Maha Penyayang. Segala puji hanya untuk Allah Tuhan alam semesta. Ya Allah, limpahkan sholawat dan salam kepada Sayyidina Muhammad, pencakup kebutuhanku, penghidup jiwaku, penghibur hatiku serta penyelamatku di dunia dan akhirat.

Alhamdulillah, segala puji dan syukur penulis hanya kepada Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini sebagaimana mestinya.

Tugas Akhir ini merupakan salah satu penerapan ilmu yang telah didapatkan selama kuliah. Dengan adanya penelitian ini, penulis InsyaAllah akan dapat memahami sistem informasi pembayaran vakasi di lingkungan Fakultas Teknologi Industri Universitas Islam Indonesia.

Penulis menyampaikan ucapan terimakasih dan penghargaan yang setinggi-tingginya atas bantuan, bimbingan serta dukungan dari berbagai pihak yang ikut membantu demi kelancaran pelaksanaan Tugas Akhir ini.

1. Bapak Gumbolo Hadi Susanto Ir, M.Sc selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Fathul Wahid ST, M.Sc selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia (FTI UII) pada waktu penulis memulai proses tugas akhir dan telah mengizinkan penulis melakukan penelitian di FTI UII.
3. Bapak Yudi Prayudi, M.Kom selaku Ketua Jurusan Teknik Informatika Universitas Islam Indonesia.
4. Bapak Zainudin Zukhri, ST., M.I.T. selaku dosen pembimbing. Terima kasih atas segala bantuan, dukungan, semangat dan pengetahuannya, serta kemudahan yang telah diberikan.
5. Bapak dan ibu Divisi Administrasi Umum (DAU), Divisi Administrasi Keuangan DAK, Divisi Administrasi Perkuliahan/ akademik(DAK), yang telah memberikan waktu untuk memberi informasi demi kelancaran pembuatan sistem ini.
6. Keluargaku tercinta, Ayah, Ibu serta adikku yang selalu mendukung dengan semangat dan doa.
7. Bapak Gatot dan Mas Ponidi dari BPPT dan Mas Made dari PSBA yang telah memberikan pengalaman yang baru.
8. Novianti Puspitasari, terima kasih selalu ada untukku serta semangat, dorongan dan selama menyelesaikan tugas akhir ini.
9. Teman-teman FIRE, terima kasih atas persahabatan, kekompakan, kerjasama, dan kekeluargaan serta dukungannya. Serta teman-teman informatika lain.
10. Sahabat-sahabatku yang selalu membuatku tertawa dan larut dalam persahabatan sampai saat ini, Kin-kin, Decha, Sony, Wisnu.
11. Para penghuni “the kontrakan” yang selalu menemani setiap hari dalam satu rumah.

12. Teman-teman unit 28, persahabatan yang terkenang dalam 1 bulan dan tak terlupakan.
13. Semua pihak yang tidak dapat penulis sebutkan satu persatu dalam membantu sejak pengumpulan data sampai penyusunan Tugas Akhir ini.

Semoga amal ibadah dan kebaikan yang telah diberikan mendapatkan imbalan yang setimpa dari Allah SWT.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan, oleh sebab itu penulis sangat mengharapkan kritik serta saran yang bersifat membangun untuk perbaikan di masa mendatang. Semoga Tugas Akhir ini bermanfaat untuk bersama. Amin.

Wassalamu'alaikum Warohmatullahi Wabarokatuh

Yogyakarta, 11 Juni 2009

Penyusun



SARI

Vakasi sangat berpengaruh terhadap aktivitas personal dan sebaliknya, terutama di lingkungan Fakultas Teknologi Industri Universitas Islam Indonesia (FTI UII). Kurangnya sistem informasi di lingkungan FTI UII yang dapat mengolah data vakasi dan data aktivitas untuk menghasilkan berbagai macam informasi. Tugas akhir ini bertujuan membangun sebuah aplikasi untuk memudahkan level operasional dalam mempermudah pendokumentasian data vakasi dan aktivitas, sedangkan untuk level manajerial dapat dijadikan referensi dalam mengambil kebijakan kedepannya. Secara umum aritektur dari aplikasi ini menggunakan konsep *client/erver* dan optimasi *database* untuk mempercepat kinerja aplikasi dalam mengintegrasikan data antara Divisi Administrasi Perkuliahan/Akademik (DAP), Divisi Administrasi Umum (DAU) dan Divisi Administrasi Keuangan (DAK). Hasil yang didapat dari tugas akhir ini adalah aplikasi yang dapat mempermudah serta meningkatkan kinerja DAK dalam pendokumentasian vakasi, selain itu aplikasi ini dapat memberikan informasi aktivitas yang berimbas pada vakasi yang didapat sebagai referensi dalam menentukan kebijakan oleh level operasional dalam hal ini pihak jurusan dan fakultas, sehingga pemerataan vakasi ataupun aktivitas personal di lingkungan FTI UII dapat tercapai.

Katakunci : Sistem informasi pembayaran vakasi, SIPeV.



TAKARIR

<i>Client</i>	computer yang terhubung dalam jaringan komputer
<i>Server</i>	komputer yang mengatur jalannya lalu lintas komunikasi
<i>Resource</i>	beban
<i>Multi-user</i>	banyak pengguna
<i>Relasional</i>	hubungan
<i>Recordset</i>	beberapa baris data
<i>Tier</i>	tingkatan
<i>Field</i>	kolom
<i>Query</i>	kumpulan perintah untuk menampilkan baris data
<i>User interface</i>	tampilan pengguna
<i>Bussines logic</i>	logika bisnis
<i>Disk server</i>	penggunaan kapasitas di computer utama
<i>Program logic</i>	logika pemrograman
<i>Layer</i>	lapisan
<i>Data interface</i>	tampilan data
<i>Transaction interface</i>	tampilan transaksi data
<i>External access interface</i>	tampilan akses luar
<i>Data logic</i>	logika data
<i>Bussines rule</i>	aturan bisnis

DAFTAR ISI

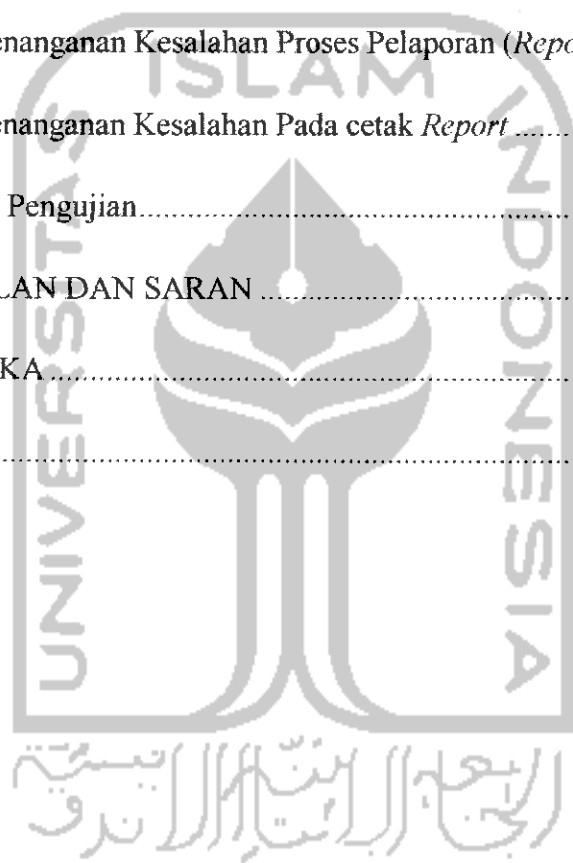
LEMBAR PENGESAHAN PEMBIMBING	i
LEMBAR PENGESAHAN PENGUJI	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
KATA PENGANTAR	v
SARI	viii
TAKARIR	ix
DAFTAR ISI	xi
DAFTAR TABEL	xvi
DAFTAR GAMBAR	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	4
1.6.1 Metode Pengumpulan Data	4
1.6.2 Metodologi Pengembangan Perangkat Lunak	5

1.7	Sistematika Penulisan.....	6
BAB 2	LANDASAN TEORI.....	8
2.1	Vakasi.....	8
2.2	Sistem Informasi Penggajian.....	10
2.3	Dasar dalam Membangun SIPeV.....	11
2.3.1	<i>Database</i>	13
2.3.2	Pemrograman <i>Client/Server</i>	19
2.3.3	<i>Stored Procedure</i>	22
BAB 3	ANALISIS KEBUTUHAN PERANGKAT LUNAK.....	27
3.1	Tinjauan Sistem.....	27
3.2	Analisis Kebutuhan Perangkat Lunak.....	28
3.2.1	Analisis Kebutuhan fungsional Perangkat Lunak.....	29
3.2.2	Analisis Kebutuhan non-fungsional Perangkat Lunak.....	41
BAB 4	PERANCANGAN PERANGKAT LUNAK SIPeV.....	43
4.1	Perancangan Sistem.....	43
4.1.1	<i>Use Case Diagram</i>	44
4.1.2	<i>Class Diagram</i>	48
4.1.3	<i>Sequence Diagram</i>	48
4.1.4	<i>Activity Diagram</i>	56
4.2	Perancangan <i>Database</i>	60
4.2.1	Perancangan Tabel.....	61

4.2.2	Relasi Tabel	68
4.2.3	Perancangan <i>Stored Procedure</i>	69
4.3	Perancangan <i>Interface</i>	71
BAB 5	IMPLEMENTASI PERANGKAT LUNAK SIPeV	85
5.1	Implementasi Secara Umum	85
5.2	Batasan Implementasi	85
5.3	Implementasi Antar Muka	86
5.3.1	Halaman <i>Splash Screen</i>	86
5.3.2	Halaman <i>Login</i>	87
5.3.3	Halaman Manajemen Personal	87
5.3.4	Halaman Manajemen <i>User</i>	89
5.3.5	Halaman Manajemen Surat	89
5.3.6	Halaman Tambah Data Surat	91
5.3.7	Halaman Ubah Data Surat	93
5.3.8	Halaman Manajemen Matakuliah	94
5.3.9	Halaman Tambah, Edit Data Mengajar	96
5.3.10	Halaman Manajemen Mengajar	96
5.3.11	Halaman Tambah, Edit Data Mengajar	97
5.3.12	Halaman Manajemen Jurusan	98
5.3.13	Halaman Manajemen Post-tarif	99
5.3.14	Halaman Manajemen Aktivitas	99

5.3.15	Halaman Tambah, Edit Data Aktivitas	100
5.3.16	Halaman Manajemen Bank.....	101
5.3.17	Halaman Manajemen Pembayaran Vakasi	102
5.3.18	Halaman Manajemen <i>Report</i> (Laporan Pembayaran).....	103
5.3.19	Halaman Cetak <i>Report</i> Bulanan Personal.....	104
5.3.20	Halaman <i>View</i> (Lihat) Frekwensi Vakasi Jurusan.....	104
5.3.21	Halaman <i>View</i> (Lihat) Frekwensi Vakasi Personal	105
5.3.22	Halaman <i>View</i> (Lihat) Vakasi Personal	106
5.3.23	Halaman Manajemen Profil.....	106
5.3.24	Halaman <i>Set Setting</i>	107
5.3.25	Halaman <i>About</i>	108
5.3.26	Halaman <i>Pop Up</i>	109
5.3.27	Halaman <i>Log</i> aktivitas	111
BAB 6	ANALISIS KINERJA PERANGKAT LUNAK	112
6.1	Batasan Pengujian.....	112
6.2	Pengujian Normal.....	112
6.2.1	Proses <i>Login</i>	112
6.2.2	Proses Penambahan data.....	113
6.2.3	Proses Pengubahan Data.....	113
6.2.4	Proses Penghapusan Data.....	114
6.3	Pengujian Tidak Normal.....	115

6.3.1	Penanganan Kesalahan Proses <i>Login</i>	115
6.3.2	Penanganan Kesalahan Pengetikan Pada <i>Form</i>	117
6.3.3	Penanganan Kesalahan Proses Penambahan Data.....	117
6.3.4	Penanganan Kesalahan Proses Pengubahan Data.....	118
6.3.5	Penanganan Kesalahan Pada Proses Pencarian.....	119
6.3.6	Penanganan Kesalahan Proses Pelaporan (<i>Reporting</i>).....	119
6.3.7	Penanganan Kesalahan Pada cetak <i>Report</i>	120
6.4	Hasil Pengujian.....	121
BAB 7	SIMPULAN DAN SARAN.....	123
	DAFTAR PUSTAKA.....	125
	LAMPIRAN.....	127



DAFTAR TABEL

Tabel 4-1 Statistik <i>database</i> db_sipev.....	61
Tabel 4-2 Tabel v_aktivas.....	61
Tabel 4-3 Tabel v_bank.....	62
Tabel 4-4 Tabel v_jurusan.....	62
Tabel 4-5 Tabel v_matakuliah.....	62
Tabel 4-6 Tabel v_detil_matkul.....	63
Tabel 4-7 Tabel v_mengajar.....	63
Tabel 4-8 Tabel v_personal.....	64
Tabel 4-9 Tabel v_posttarif.....	64
Tabel 4-10 Tabel v_surat.....	65
Tabel 4-11 Tabel v_detil_stk.....	65
Tabel 4-12 Tabel v_detil_spj.....	66
Tabel 4-13 Tabel v_akomodasi_spj.....	66
Tabel 4-14 Tabel v_pembayaran.....	66
Tabel 4-15 Tabel v_user.....	67
Tabel 4-16 Tabel v_setting.....	67
Tabel 4-17 Tabel v_profile.....	68
Tabel 4-18 Tabel v_log_aktivitas.....	68

DAFTAR GAMBAR

Gambar 2-1 Arsitektur <i>Database System</i>	18
Gambar 2-2 Arsitektur Komputer.....	21
Gambar 2-3 Ilustrasi dari Penggunaan Trigger.....	24
Gambar 3-1 Alur Proses Sistem SIPeV.....	28
Gambar 4-1 <i>Use Case Diagram</i>	45
Gambar 4-2 <i>Class Object</i> yang Berorientasi <i>Form</i> dan Prosesnya.....	49
Gambar 4-3 <i>Class Object</i> yang Berorientasi <i>Database</i>	50
Gambar 4-4 <i>Sequence Diagram</i> Tambah Surat Perjalanan Dinas.....	51
Gambar 4-5 <i>Sequence Diagram</i> Edit Surat Perjalanan Dinas.....	52
Gambar 4-6 <i>Sequence Diagram</i> Delete Surat Perjalanan Dinas.....	53
Gambar 4-7 <i>Sequence Diagram</i> Tambah Pembayaran.....	54
Gambar 4-8 <i>Sequence Diagram</i> Edit Pembayaran.....	55
Gambar 4-9 <i>Sequence</i> Delete Pembayaran.....	55
Gambar 4-10 <i>Activity Diagram</i> Halaman Admin.....	57
Gambar 4-11 <i>Activity Diagram</i> Seluruh Sistem.....	58
Gambar 4-12 <i>Activity Diagram</i> Halaman Manajemen Pembayaran Vakasi.....	59
Gambar 4-13 <i>Stored Procedure</i> Terkait Dengan Tabel <i>v_personal</i>	69
Gambar 4-14 <i>Stored Procedure</i> Terkait Dengan Tabel <i>v_matakuliah</i>	70
Gambar 4-15 <i>Stored Procedure</i> terkait dengan tabel <i>v_setting</i>	70

Gambar 4-16 <i>Stored Procedure</i> Terkait Dengan Tabel <i>v_detil_spj</i>	71
Gambar 4-17 Halaman <i>Splash Screen</i>	71
Gambar 4-18 Halaman <i>Login</i>	72
Gambar 4-19 Halaman <i>Main Menu</i>	73
Gambar 4-20 Halaman Manajemen Personal	74
Gambar 4-21 Halaman Manajemen Jurusan.....	74
Gambar 4-22 Halaman Manajemen Matakuliah.....	75
Gambar 4-23 Halaman Manajemen Mengajar.....	76
Gambar 4-24 Halaman Tambah dan Edit Mengajar	76
Gambar 4-25 Halaman Manajemen Post-tarif.....	77
Gambar 4-26 Halaman Manajemen Surat.....	78
Gambar 4-27 Halaman Tambah dan Edit Surat.....	78
Gambar 4-28 Halaman Manajemen Pembayaran	79
Gambar 4-29 Halaman Manajemen Aktivitas	80
Gambar 4-30 Halaman Tambah dan Edit Data Aktivitas	80
Gambar 4-31 Halaman Manajemen Bank.....	81
Gambar 4-32 Halaman Manajemen <i>User</i>	82
Gambar 4-33 Halaman Manajemen Report.....	82
Gambar 4-34 Halaman Manajemen <i>Report</i> Personal.....	83
Gambar 4-35 Halaman <i>View</i> Vakasi.....	83
Gambar 4-36 Halaman Log Aktivitas.....	84

Gambar 5-1 Halaman <i>Splash Screen</i>	86
Gambar 5-2 Halaman <i>Login</i>	87
Gambar 5-3 Halaman <i>Mainmenu</i>	88
Gambar 5-4 Halaman Manajemen Personal.....	88
Gambar 5-5 Halaman Manajemen <i>User</i>	89
Gambar 5-6 Halaman Manajemen Surat.....	90
Gambar 5-7 Menu Kategori Surat.....	91
Gambar 5-8 Halaman Tambah Data Surat Tugas.....	91
Gambar 5-9 Halaman Tambah Data Surat Keputusan.....	92
Gambar 5-10 Halaman Tambah Data Surat Perjalanan Dinas.....	93
Gambar 5-11 Halaman Ubah Data ST dan SK.....	94
Gambar 5-12 Halaman Ubah Data SPJ.....	94
Gambar 5-13 <i>Pseudocode</i> Save Matakuliah.....	95
Gambar 5-14 Halaman Manajemen Matakuliah.....	96
Gambar 5-15 Halaman Manajemen Mengajar.....	97
Gambar 5-16 Halaman Tambah, Edit Data Mengajar.....	98
Gambar 5-17 Halaman Manajemen Jurusan.....	98
Gambar 5-18 Halaman Manajemen Post-tarif.....	100
Gambar 5-19 Halaman Manajemen Aktivitas.....	100
Gambar 5-20 Halaman Tambah, Edit Data Aktivitas.....	101
Gambar 5-21 Halaman Manajemen Bank.....	102

Gambar 5-22 Halaman Manajemen Pembayaran Vaksinasi	103
Gambar 5-23 Halaman Manajemen <i>report</i>	103
Gambar 5-24 Halaman Cetak Report Bulanan Personal.....	104
Gambar 5-25 Halaman <i>View</i> (Lihat) Frekwensi Vaksinasi Jurusan.....	105
Gambar 5-26 Halaman <i>View</i> (Lihat) Frekwensi Vaksinasi Personal	105
Gambar 5-27 Halaman <i>View</i> (Lihat) Vaksinasi Personal	106
Gambar 5-28 Halaman Manajemen Profil.....	107
Gambar 5-29 Halaman <i>Set Setting</i>	108
Gambar 5-30 Halaman <i>About</i>	108
Gambar 5-31 Form Untuk Memunculkan <i>Pop up</i> Personal	109
Gambar 5-32 Halaman <i>Pop up</i> Data Personal	109
Gambar 5-33 Form Untuk Memunculkan <i>Pop up</i> Matakuliah.....	110
Gambar 5-34 Halaman <i>Pop up</i> Data Matakuliah.....	110
Gambar 5-35 Form Untuk Memunculkan <i>Pop up</i> Data Post-tarif.....	110
Gambar 5-36 Halaman <i>Pop up</i> Data Post-tarif.....	111
Gambar 5-37 Halaman <i>log</i> aktivitas	111
Gambar 6-1 Pesan Berhasil Menambah Data Personal	113
Gambar 6-2 Pesan Berhasil Mengubah Data Personal	114
Gambar 6-3 Pesan Dialog Penghapusan Data	114
Gambar 6-4 Pesan Dialog Data Telah Berhasil Dihapus.....	114
Gambar 6-5 Pesan Kesalahan Jika <i>Username</i> dan <i>Password</i> Belum Diisi	115

Gambar 6-6 Pesan Kesalahan Jika <i>Password</i> Belum Diisi.....	116
Gambar 6-7 Kesalahan Jika <i>Username</i> Belum Diisi	116
Gambar 6-8 Kesalahan Jika <i>Username</i> Tidak terdapat dalam <i>Database</i>	116
Gambar 6-9 Kesalahan Jika <i>Username</i> Tidak Mempunyai Hak Akses.....	116
Gambar 6-10 Kesalahan Jika <i>Username</i> tidak sesuai dengan <i>password</i>	117
Gambar 6-11 Kesalahan pengetikan data pada <i>form</i>	117
Gambar 6-12 Pesan Data Baru.....	117
Gambar 6-13 Proses Data Gagal Dimasukkan Kedalam <i>Database</i>	118
Gambar 6-14 Proses Data Gagal Diubah.....	118
Gambar 6-15 Penanganan Kesalahan Ubah Data Surat.....	119
Gambar 6-16 Penanganan Kesalahan Ubah Data Mengajar.....	119
Gambar 6-17 Penangan Kesalahan Proses Pencarian.....	120
Gambar 6-18 Penanganan Kesalahan Data <i>Report</i> Kosong	120
Gambar 6-19 Penanganan Kesalahan Cetak Data Kosong.....	120
Gambar 6-20 Penanganan Kesalahan Cetak Data Dalam Bentuk <i>Form</i>	121

BAB 1

PENDAHULUAN

Pada bab ini akan berisi pendahuluan yang menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan tugas akhir dari pengembangan perangkat lunak Sistem Informasi Pembayaran Vaksin Fakultas Teknologi Industri Fakultas Teknologi Industri Universitas Islam Indonesia (FTI UII), khususnya pada bagian pemrosesan dan penyajian data.

1.1 Latar Belakang

Teknologi informasi memegang peranan penting dalam segala bidang kehidupan. Perkembangan pengolahan data secara digital menggunakan komputer yang semakin meluas secara cepat menandai semakin cepatnya perkembangan teknologi informasi. Kemajuan teknologi informasi tidak bisa dipungkiri sudah merambah ke semua bidang kehidupan, tidak terkecuali bidang pendidikan. Salah satu sistem yang biasa digunakan dalam pengelolaan data akutansi adalah SIAT (Sistem Informasi Akutansi Terpadu) yaitu suatu sistem sederhana yang berfungsi hanya menyimpan data dan *reporting* (pelaporan) data akutansi tersebut.

Sebagai lembaga pendidikan yang berkecimpung dalam bidang teknologi industri, tentu saja FTI UII tidak mau tertinggal dalam hal ini. Hanya saja, di tengah-tengah maraknya penggunaan teknologi informasi di bidang pendidikan ini, ternyata masih terdapat berbagai pekerjaan di fakultas ini yang dikerjakan belum optimal, yaitu manajemen pembayaran vaksin menggunakan SIAT (Sistem Informasi Akutansi Terpadu). Walaupun menggunakan Teknologi Informasi, penggunaannya tidak berbeda jauh hanya sebatas sebagai pelaporan tanpa adanya kemampuan memonitor di level manajerial.

Manajemen pembayaran vaksin menggunakan SIAT (Sistem Informasi Akutansi Terpadu) belum dirasa cukup untuk memenuhi kebutuhan pada level

operasional terlebih di level manajerial. Informasi yang didapat dari SIAT hanya berupa laporan hasil yang didapat perorangan saja, tetapi untuk memonitor sejauh mana pemerataan hasil vakasi belum maksimal. Sedangkan pemerataan hasil vakasi tersebut sangat penting terutama dalam menunjang pemerataan tugas – tugas *fakultatif*, sebagai contoh tugas penerimaan calon laboran jurusan teknik elektro FTI UII, delegasi tes wawancara beasiswa DAAD, di mana dalam realisasinya pembagian tugas belum merata untuk setiap personal dalam unit FTI UII.

Selain itu dalam SIAT informasi antar bagian belum terintegrasi sehingga pada level operasional pertukaran data masih bersifat manual, yaitu pertukaran data antar bagian dalam struktur fakultas yang terkait dengan bagian keuangan belum terkomputerisasi secara efektif. Alhasil, informasi yang dikeluarkan oleh bagian keuangan belum dapat diakses oleh komponen-komponen dalam level manajerial secara langsung. Witarto (2004) menyatakan bahwa informasi yang benar dan baru, dapat mengoreksi dan mengkonfirmasi informasi sebelumnya. Informasi dapat juga dikatakan sebagai data yang telah di proses, yang mempunyai nilai tindakan dan keputusan. Sementara proses pengambilan keputusan merupakan hal yang inheren dalam fungsi manajerial.

Dari kelemahan SIAT tersebut tidak menutup kemungkinan berdampak kepada individu setiap staf yang terkait di dalam FTI UII. Pemerataan pekerjaan dan kegiatan di luar tugas pokok yang berimbas pada vakasi yang didapat. Hal itu berpengaruh pada totalitas masing-masing personal untuk memajukan lembaga pendidikan ini, sehingga tidak ada anggapan dari setiap staf bahwa dirinya tidak dianggap dan tidak dilibatkan dalam pekerjaan, padahal mempunyai kompetensi diri di bidang tersebut.

Untuk mengatasi berbagai hal di atas, sistem informasi pembayaran vakasi dapat memberikan alternatif solusi yang efektif. Informasi tidak hanya sebagai laporan tetapi juga pada level manajerial dapat digunakan sebagai tolak ukur pemerataan pekerjaan sesuai dengan kompetensi diri setiap personal. Tetapi

aplikasi ini lebih dikhususkan untuk level operasional dalam hal ini bagian keuangan, yang bertujuan mempermudah peng-*entry*-an data sehingga meningkatkan kualitas pelayanan.

Sistem ini mengintegrasikan semua bagian dalam struktur fakultas yang terkait dengan bagian keuangan, sehingga bagian keuangan memperoleh data dengan cepat, akurat dan efektif. Informasi dari bagian keuangan yang berupa data vakasi dapat diakses secara langsung dan dijadikan referensi pada level manajerial untuk mengambil keputusan untuk pemerataan tugas bagi setiap personal.

1.2 Rumusan Masalah

Sistem Informasi Akuntansi Terpadu (SIAT) sesungguhnya sudah membantu dalam pengerjaan di level operasional tetapi tidak hanya level operasional yang berperan dalam memberikan informasi kepada setiap unit dalam fakultas teknologi industri, pada level manajerial juga sangat berperan karena pada level ini dapat memantau kinerja atau aktivitas dan pemerataan pekerjaan pada diri personal dalam keterlibatan memajukan fakultas teknologi industri.

Bagaimana membangun sebuah sistem informasi pembayaran vakasi yang terkomputerisasi dan terintegrasi antara bagian-bagian yang berada dalam fakultas teknologi industri dan juga dapat digunakan sebagai monitoring atas pemerataan pekerjaan yang berdampak pada vakasi yang didapat oleh masing-masing staf. Sebuah sistem yang familiar di level operasional sehingga mudah dan cepat dalam pengoperasiannya.

1.3 Batasan Masalah

Dalam sistem informasi pembayaran vakasi ini, terdapat beberapa batasan masalah, yaitu:

1. Hanya menangani masalah keuangan pembayaran vakasi.
2. Analisis sistem keuangan berdasarkan prosedur–prosedur kerja atau kebijakan yang ada dalam kepala bagian dan berkaitan dengan keuangan Fakultas Teknologi Industri.
3. Implementasi sistem terpisah dari sistem yang sudah ada.

1.4 Tujuan Penelitian

Tujuan dari tugas akhir ini adalah mengintegrasikan seluruh bagian yang terkait dengan bagian keuangan dan memberikan informasi supaya bagian manajerial dapat memberikan keputusan terkait dengan keadaan dari aktivitas atau pekerjaan personal.

1.5 Manfaat Penelitian

Beberapa manfaat dari tugas akhir ini yaitu:

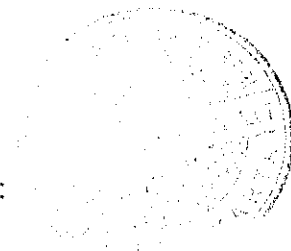
1. Memudahkan *monitoring* (pemantauan) dari aktivitas personal FTI-UII pada level manajerial
2. Memberikan kemudahan bagi level operasional dalam hal ini adalah bagian keuangan agar mempercepat pelaporan dan *peng-entry-an* data.
3. Memudahkan pertukaran data dan menambah efisiensi kerja karena setiap bagian terintegrasi.

1.6 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini meliputi metode pengumpulan data dan pengembangan sistem.

1.6.1 Metode Pengumpulan Data

Metode dalam mengumpulkan data terdiri dari 3 cara yaitu:



1. Wawancara

“Tujuan wawancara adalah untuk memperoleh informasi yang lebih akurat dan lengkap, untuk menyusun sistem baru agar sesuai dengan kebutuhan” (Witarto, 2004). Wawancara yang saya lakukan dengan pendekatan *bottom up*. Yaitu terlebih dahulu dengan mewawancarai bagian umum, bagian keuangan dan bagian-bagian yang terkait dengan bagian keuangan. Kemudian bertahap sampai tingkat pimpinan fakultas.

2. Observasi

Observasi dilakukan melalui pengamatan dan perekaman atau pencatatan data-data perbagian (bagian umum, bagian keuangan dan bagian-bagian yang terkait dengan bagian keuangan). Hal ini bertujuan untuk mendapatkan data yang akan digunakan untuk membuat sistem.

3. Studi pustaka

Cara ini dilakukan dengan mengumpulkan data dari buku, majalah, dan *literature* lain yang berhubungan dengan sistem informasi baik perancangan pembuatan dan penerapan. untuk menambah informasi serta sebagai referensi dalam penyelesaian tugas akhir ini

1.6.2 Metodologi Pengembangan Perangkat Lunak

Dalam metode ini meliputi:

1. Analisis sistem

Dalam tahap ini dilakukan pengamatan terhadap permasalahan proses pemberian tugas dan pelaporan untuk level manajerial yang kemudian digunakan sebagai dasar dalam pembuatan sistem ini. Analisis dilakukan pada *input*, *output*, dan operasi pada sistem.

2. Perancangan sistem

Dalam perancangan sistem informasi ini digunakan *Unified Modelling Language* (UML) dan dapat menggambarkan skenario yang lebih spesifik tentang kasus dan aktivitas yang terjadi di dalam suatu system untuk

visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Termasuk di dalamnya adalah perancangan *database* yang akan menyimpan data yang dibutuhkan sistem sesuai dengan perancangannya.

3. Implementasi sistem

Implementasi sistem dilakukan dengan cara pembuatan sistem sesuai dengan perancangan yang telah dibuat sebelumnya.

4. Analisis sistem

Pada tahap ini dilakukan analisis terhadap sistem yang telah dibuat guna mengetahui normal tidaknya sistem. Sistem akan dipasang di bagian keuangan dan menggali *bug* atau kesalahan dalam sistem yang bertujuan untuk memperbaikinya sehingga menjadi sistem yang handal dan setabil. Sistem yang handal yaitu sistem yang dapat di minimalisir kesalahan input yang menyebabkan program tidak berjalan. Dan stabil dalam sisi penyimpanan data agar data tidak ada yang *crash* (hilang), rusak, atau *redundant*.

1.7 Sistematika Penulisan

Sistematika penulisan bertujuan untuk menggambarkan secara singkat isi laporan pada tiap-tiap bagian sehingga laporan menjadi utuh dan jelas. Sistematika penulisan dan garis besar isi laporan ini adalah sebagai berikut:

1. BAB 1, PENDAHULUAN.

Pada bab ini akan berisi pendahuluan yang menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan tugas akhir dari pengembangan perangkat lunak Sistem Informasi Pembayaran Vakasi Fakultas Teknologi Industri Fakultas Teknologi Industri Universitas Islam Indonesia (SIPeV FTI UII).

2. BAB 2, LANDASAN TEORI.

Bab ini berisi mengenai dasar teori penggajian, vakasi dan teori yang berhubungan dengan pengembangan SIPeV FTI UII

3. BAB 3, ANALISIS KEBUTUHAN PERANGKAT LUNAK.

Pada bab ini akan menguraikan analisis kebutuhan perangkat lunak, yang memuat uraian tentang tinjauan sistem dan analisis kebutuhan fungsional dan non-fungsional yang diperlukan untuk membangun SIPeV FTI UII.

4. BAB 4, PERANCANGAN PERANGKAT LUNAK SIPeV

Pada bab ini akan dibahas perancangan SIPeV FTI UII, khususnya pada bagian pemrosesan dan penyajian data. Pada bagian perancangan perangkat lunak ini membahas tentang metode perancangan yang digunakan yakni metode pemodelan *object* menggunakan *Unfied Modelling Language* (UML), perancangan *database*, dan perancangan antarmuka (*interface*).

5. BAB 5, IMPLEMENTASI PERANGKAT LUNAK SIPeV

Pada bab ini memuat implementasi perangkat lunak dari hasil rancangan pada bab sebelumnya. Dalam bagian implementasi meliputi implementasi secara umum, batasan implementasi, dan implementasi antarmuka.

6. BAB 6, ANALISIS KINERJA PERANGKAT LUNAK

Dalam bab ini memuat dokumentasi hasil pengujian terhadap perangkat lunak yaitu pada SIPeV

7. BAB 7, SIMPULAN DAN SARAN

Dalam bab ini berisi kesimpulan berdasarkan pembahasan hasil penelitian, dan saran untuk perbaikan dan perkembangan penelitian berikutnya.

BAB 2

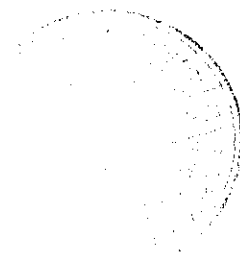
LANDASAN TEORI

Bab ini berisi mengenai dasar teori yang berhubungan dengan pengembangan sistem informasi pembayaran vakasi Fakultas Teknologi Industri Universitas Islam Indonesia (SIPeV FTI UII). Berdasarkan permasalahan yang ada maka dasar teori ini dapat dikelompokkan menjadi beberapa sub-bab, yaitu sistem informasi penggajian yang menjadi dasar dari SIPeV, *database* sebagai media penyimpanan dari SIPeV FTI UII, pemrograman *client/server* yang menjelaskan tentang dasar dari komunikasi data SIPeV FTI UII, dan *stored procedure* sebagai pengurang beban *resource* yang terpakai saat aplikasi dijalankan.

2.1 Vakasi

Menurut kamus besar bahasa Indonesia (KBBI) vakasi adalah uang imbalan bagi penguji atau pemeriksa kertas, dapat diambil di bagian keuangan. Berdasarkan pengertian tersebut vakasi adalah uang yang didapat pegawai setelah pegawai melakukan tugas atau kewajibannya. Dalam hal ini kewajiban yang dimaksud adalah sebagai penguji atau pemeriksa kertas ujian, dalam kasus SIPeV vakasi akan disamakan dengan gaji dan upah yang didapat oleh pegawai di Fakultas Teknologi Industri (FTI).

Peraturan Pemerintah No. 8 tahun 1981 tentang perlindungan upah memberikan definisi upah sebagai berikut: “.....suatu penerimaan sebagai imbalan dari pengusaha kepada tenaga kerja untuk suatu pekerjaan atau jasa yang telah atau akan dilakukan, dinyatakan atau dinilai dalam bentuk uang yang ditetapkan menurut persetujuan atau perturan perundang-undangan dan dibayarkan atas dasar suatu perjanjian kerja antara pengusaha (pemberi kerja) dan pekerja termasuk tunjangan baik untuk pekerja sendiri maupun keluarganya (Rucky, 2006).”



Peraturan Menteri No. 3 tahun 1996 tentang Pemutusan Hubungan Kerja memberikan definisi yang lebih detail tentang upah karena ditujukan untuk keperluan perhitungan pasongan. Dalam peraturan menteri 3/96 ini yang dimaksud dengan upah mencakup: “upah pokok, segala tunjangan berkala dan teratur, harga pembelian dari catu yang diberikan kepada pekerja, penggantian untuk perumahan yang diberikan cuma-cuma, dan penggantian untuk pengobatan dan perawatan kesehatan (Rucky, 2006).”

Dalam praktek di perusahaan dan dalam buku-buku manajemen sumber daya manusia istilah upah dan gaji digunakan dalam konteks yang berbeda. Pertama, istilah upah digunakan untuk menggambarkan pembayaran jasa kerja untuk satuan waktu pendek, misalnya per hari atau malahan per jam. Gaji/vakasi menggambarkan pembayaran jasa kerja untuk satuan waktu lebih panjang biasanya sebulan.

Kedua, untuk menggambarkan kaitan pekerja penerima upah dengan proses produksi pada industri manufaktur. Upah atau dalam bahasa Inggris disebut *wage* dibayarkan kepada pekerja yang terlibat langsung dalam proses produksi, baik terlibat langsung (*direct labor*) misalnya operator, maupun tidak langsung (*indirect labor*) misalnya bagian pemeliharaan mesin. Biaya untuk upah tersebut diikutkan langsung dalam perhitungan biaya produksi barang tersebut (*cost of goods sold*). Sebaliknya, pekerja yang menerima gaji/vakasi termasuk dalam kategori *fixed costs* (biaya tetap) atau *overhead costs*. Contohnya adalah para supervisor atau manajer, staf administrasi. Dalam industri proses misalnya industri pupuk, kimia, semen yang sukar untuk menghitung komponen biaya personel per unit produksi, tidak terdapat pengelompokan penerima gaji atau upah.

Dalam usaha untuk menghilangkan ketidakadilan dalam pengupahan/penggajian ini Indonesia baru saja dengan bangga meratifikasi Konvensi ILO nomor 100 yang menetapkan bahwa tidak ada diskriminasi dalam pengupahan untuk pekerja yang mempunyai nilai yang sama. Yang dimaksudkan

oleh Konvensi ini adalah bahwa pekerja-pekerja yang mempunyai nilai atau bobot (*value*) yang sama harus mendapat imbalan yang sama. Sebaliknya, dari Konvensi ini dapat disimpulkan bahwa yang harus menjadi dasar untuk menetapkan apakah dua atau beberapa jabatan dinyatakan mempunyai nilai/bobot yang sama bukan hanya tingkat pendidikan formal yang dimiliki oleh orang yang melakukannya atau masa kerjanya, tetapi juga harus dinilai faktor-faktor lain yang relevan. Sebenarnya bila dikatakan imbalan yang sama persis adalah tidak tepat, tetapi yang dimaksudkan adalah dimasukkan ke dalam *job class/ job grade* (penggolongan jabatan) yang sama sehingga dengan demikian akan masuk juga ke dalam patokan upah yang sama.

Masalah utama dalam menjamin tercapainya keadilan melalui nilai jabatan menentukan cara yang dianggap tepat untuk menetapkan nilai/bobot tersebut, sehingga pekerjaan/jabatan X yang kemudian atas dasar kriteria tertentu dianggap mempunyai nilai/bobot yang sama dengan jabatan Y akan mendapat imbalan yang sama. Pemberian nilai/bobot kepada sebuah jabatan/pekerja dilakukan melalui sebuah proses atau kegiatan yang biasa disebut evaluasi jabatan.

Dengan kata lain, evaluasi jabatan adalah sebuah proses yang dilakukan dalam ruang lingkup manajemen sumber daya manusia yang bertujuan menentukan nilai relatif dalam arti berat-ringannya suatu pekerjaan dibandingkan dengan pekerjaan-pekerjaan yang lain dalam sebuah organisasi.

2.2 Sistem Informasi Penggajian

Sumber daya keuangan merupakan bagian yang penting di dalam organisasi. Semua elemen organisasi mempunyai kepentingan terhadap sumber daya keuangan tersebut (Mulyanto, 2009). Dalam hal ini adalah pihak fakultas, jurusan, dan tiga divisi operasional meliputi divisi administrasi umum, divisi administrasi perkuliahan dan akademik, divisi administrasi keuangan sangat membutuhkan informasi mengenai keuangan yang ada di dalam Fakultas Teknologi Industri. Oleh karena itu, sumber daya keuangan harus dikelola dengan

baik agar dapat menghasilkan informasi yang berkualitas dan mendukung manajerial dalam mengambil keputusan.

Sistem informasi keuangan merupakan suatu sistem yang kompleks dan terdapat beberapa sub sistem, salah satu dari sistem yang berhubungan dengan sistem informasi keuangan adalah sistem informasi pembayaran gaji atau dalam hal ini adalah SIPeV. SIPeV bertujuan menangani data pembayaran dan memberikan informasi kepada personal unit struktural yang berada di FTI mengenai vakasi atau gaji yang didapat setelah melakukan tugas dan kewajiban dari fakultas.

2.3 Dasar dalam Membangun SIPeV

Dalam membangun aplikasi SIPeV ini terdapat beberapa teori pengembangan software yang digunakan yaitu media penyimpanan data atau database, media dan model integrasi yang mengacu pada pemrograman *client/server* serta *stored procedure* yang digunakan untuk mengurangi beban *resource* aplikasi. Selain itu terdapat 3 persyaratan sistem informasi pembayaran yang dikemukakan oleh (Howland, 1996):

1. Persyaratan Keamanan.

Keamanan sistem harus jauh lebih ketat dikarenakan sistem pembayaran dapat dikatakan suatu hal yang rahasia, oleh karena itu sistem yang harus dibuat dapat meminimalisir dari celah yang dapat akses oleh pemakai (*user*) yang tidak mempunyai kepentingan terhadap sistem. Hal itu dilakukan agar data-data dari sistem dapat terjaga keamanannya.

2. Persyaratan Privasi.

Sistem pembayaran harus benar-benar berorientasi untuk memberikan pembayaran yang aman, dan tidak mengungkapkan informasi asing dalam situasi apapun. Seperti halnya Informasi tambahan harus diberikan (seperti pada permintaan vakasi, ketika nama dan alamat akan diperlukan), untuk hal tersebut sangat diperlukan suatu privasi yang dapat mengatur akses informasi

kepada pihak yang berhak mengetahui informasi dan pihak yang tidak boleh mengetahui informasi tersebut. Fitur yang terpenting lebih mengarah pada sistem pembayaran itu sendiri dimana pengguna (*user*) menggunakan informasi tambahan untuk memenuhi kebutuhan bukan untuk melemahkan sistem.

3. Persyaratan Pelaksanaan.

Dalam pelaksanaannya sebuah sistem pembayaran harus memenuhi aturan-aturan sebagai berikut:

- Sistem harus sederhana
Sistem secara keseluruhan harus sederhana, sistem yang sederhana dapat dengan mudah dipahami oleh pemakai, memberikan keyakinan yang lebih besar dalam keamanan. Selain itu, sebuah sistem sederhana akan lebih mudah dalam pengembangan sistem itu sendiri, dan mendorong lebih banyak implementasi.
- Sistem harus didasarkan pengujian.
Sistem harus melewati tahapan pengujian untuk mendapatkan tanggapan dari pengguna tentang keadaan sistem, sehingga sistem dapat dipakai oleh pengguna dan bukan memaksakan pengguna memakai sistem yang baru.
- Sistem harus ditujukan pada pelayanan pengguna (*user*) bukan untuk memaksa pelanggan mengikuti aturan dari sistem sehingga mengetahui kelebihan dari sistem tersebut berdasarkan pada penggunaannya.
- Biaya untuk mendapatkan aplikasi dan pembiayaan perawatan harus rendah.

2.3.1 Database

Database atau Basisdata merupakan komponen utama dalam membangun sebuah sistem yang menyangkut pendokumentasian data ke dalam sebuah *database* (Nugroho, 2005). *Database* adalah sebuah cara mendokumentasikan berbagai macam data yang kemudian diolah dengan sebuah sistem untuk selanjutnya disimpan dalam sebuah media penyimpanan. Dengan demikian data-data tersebut dapat diakses dengan mudah dan cepat. Dalam *database*, data yang ada tidak hanya diletakkan dan disimpan begitu saja dalam sebuah media penyimpanan, akan tetapi dikelola dengan sebuah sistem pengaturan *database* yang lebih dikenal dengan *Database Management System* (DBMS). Dengan begitu suatu data dengan jumlah besar dan kompleks dapat tersusun sangat baik sehingga memungkinkan pengaksesan data dengan mudah dan cepat oleh *user*.

2.3.1.1 Database Management System (DBMS)

Remakrishnan dan Geherke (2003) menyatakan *database management system* (DBMS) adalah perangkat lunak yang didesain untuk membantu memelihara dan memanfaatkan kumpulan data yang besar (Simarmata & Prayudi, 2006). Kebutuhan akan sistem termasuk pula penggunaannya yang berkembang pesat. Alternatif penggunaan DBMS adalah menyimpan data dalam file dan menulis kode aplikasi tertentu untuk mengaturnya. Penggunaan DBMS memiliki beberapa keuntungan dan kerugian.

1. Keuntungan menggunakan DBMS

DBMS memungkinkan perusahaan maupun pengguna individu untuk :

a) Mengurangi pengulangan data

Apabila dibandingkan dengan *file-file* komputer yang disimpan terpisah disetiap aplikasi komputer, DBMS mengurangi jumlah total *file* dengan menghapus data yang terduplikasi diberbagai *file*. Data terduplikasi selebihnya dapat ditempatkan dalam satu *file*.

b) Mencapai independensi data

Spesifikasi data disimpan dalam skema pada tiap program aplikasi. Perubahan dapat dibuat pada struktur data tanpa mempengaruhi program yang mengakses data.

c) Mengintegrasikan data beberapa file

Saat file dibentuk sehingga menyediakan kaitan logis, maka organisasi fisik bukan merupakan kendala. Organisasi logis, pandangan pengguna, dan program aplikasi tidak harus tercermin pada media penyimpanan fisik.

d) Mengambil data dan informasi dengan cepat

Hubungan-hubungan logis, bahasa manipulasi data, serta bahasa *query* memungkinkan pengguna mengambil data dalam hitungan detik atau menit.

e) Meningkatkan keamanan

DBMS mainframe maupun komputer mikro dapat menyertakan beberapa lapis keamanan seperti kata sandi (*password*), direktori pemakai, dan bahasa sandi (*encryption*) sehingga data yang dikelola akan lebih aman.

2. Kerugian DBMS

Keputusan menggunakan DBMS mengikat perusahaan atau pengguna untuk:

a) Memperoleh perangkat lunak yang mahal.

DBMS *mainframe* masih sangat mahal. Walaupun harga DBMS berbasis komputer mikro lebih murah, tetapi tetap merupakan pengeluaran besar bagi suatu organisasi kecil.

b) Memperoleh konfigurasi perangkat keras yang besar

DBMS sering memerlukan kapasitas penyimpanan dan memori lebih besar daripada program aplikasi lain.

- c) Mempekerjakan dan mempertahankan staf DBA.

DBMS memerlukan pengetahuan khusus agar dapat memanfaatkan kemampuannya secara penuh. Pengetahuan khusus ini disediakan paling baik oleh para pengelola basisdata (DBA).

Berdasarkan keuntungan dan kerugian DBMS tersebut dapat ditarik kesimpulan bahwa DBMS memberikan dasar-dasar menggunakan komputer sebagai suatu sistem informasi bagi spesialis informasi dan pengguna.

2.3.1.2 Struktur Query Language (SQL)

SQL adalah sebuah bahasa permintaan *database* yang terstruktur. Bahasa SQL dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam *database* maupun merelasikan antar *database*. Bahasa SQL ditulis langsung dalam sebuah program *database* sehingga seorang pengguna dapat melihat langsung permintaan yang diinginkan, sekaligus melihat hasilnya. MySQL sendiri adalah sebuah program *database server* yang memerlukan sebuah bahasa permintaan dalam melayani permintaan *user*. Jadi, SQL-lah yang menjadi bahasa permintaan yang digunakan dalam MySQL. SQL (*Structure Query Language*) dibagi menjadi dua bentuk *query*, yaitu:

1. DDL (*Data Definition Language*)

DDL (*Data Definition Language*) adalah sebuah metode *query* SQL yang berguna untuk mendefinisikan data pada sebuah *database*, adapun *query* yang dimiliki adalah sebagai berikut:

- a) CREATE: digunakan untuk melakukan pembuatan tabel dan *database*.
- b) DROP: digunakan untuk melakukan penghapusan tabel maupun *database*.
- c) ALTER: digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah *field* (*add*), mengganti nama *field* (*change*) ataupun menamakannya kembali (*rename*), serta menghapus (*drop*).

2. DML (*Data Manipulation Language*)

DML (*Data Manipulation Language*) adalah sebuah *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan manipulasi *database* yang telah ada atau telah dibuat sebelumnya. Adapun *query* yang termasuk di dalamnya adalah sebagai berikut:

a. Select

Adalah perintah untuk memilih/menyeleksi menampilkan data yang diambil dari suatu *table*.

b. Insert

Digunakan untuk melakukan peng-*input-an* /pemasukan data pada tabel *database*.

c. Update

Digunakan untuk melakukan perubahan atau pemajuaan terhadap data yang ada pada tabel.

d. Delete

Digunakan untuk melakukan penghapusan data pada tabel. Penghapusan ini dapat dilakukan secara sekaligus (seluruh isi tabel) maupun hanya beberapa *recordset*.

Kedua bentuk *query* SQL tersebut dapat digunakan pada semua model basis data yang mendukungnya, baik yang berbasis Unix maupun Windows. Di dalam Windows misalnya, *user* dapat mengimplementasikan pada program penghasil *database* seperti Microsoft Acces atau pada semua *database* bawaan program visual seperti *database desktop* yang bernama Paradox.

2.3.1.3 Arsitektur *Database System*

Secara garis besar, *database* sistem terdiri dari tiga buah komponen utama, masing-masing sebagai berikut.

1. Aplikasi

Komponen ini bertanggung jawab dalam memformat dan mempresenting data pada pengguna atau alat keluaran dan mengendalikan masukan pengguna dari *keyboard* atau alat input lainnya. Menyajikan data dalam bentuk tampilan atau *interface* yang terbagi menjadi empat *layer* yaitu *user interface*, *data interface*, *transaction interface*, dan *external access interface*.

2. *Database Engine* sebagai *bussiness logic*

Komponen ini berguna untuk menangani logika pemrosesan data (*data processing logic*), logika aturan bisnis (*business rule logic*), dan logika manajemen data (*data management logic*). Proses *data logic* merupakan aktifitas untuk memvalidasi data dan mengidentifikasi proses eror pada data. *Business rule* tidak mempunyai kode pada DSMS, tetapi mempunyai kode pada komponen pemrosesan. *Data management logic* mengidentifikasi data yang diperlukan/penting untuk memproses transaksi atau *query*.

3. *Database* fisik

Komponen ini bertanggung jawab pada penyimpanan data dan perbaikan data dari alat penyimpan yang bekerja dengan aplikasi.

Dari ketiga komponen diatas dapat digambarkan sebuah tahapan atau layer, seperti pada Gambar 2-1 (Darmayuda, 2007).

2.3.1.4 *Database Engine* dan Model Pengaksesan Data

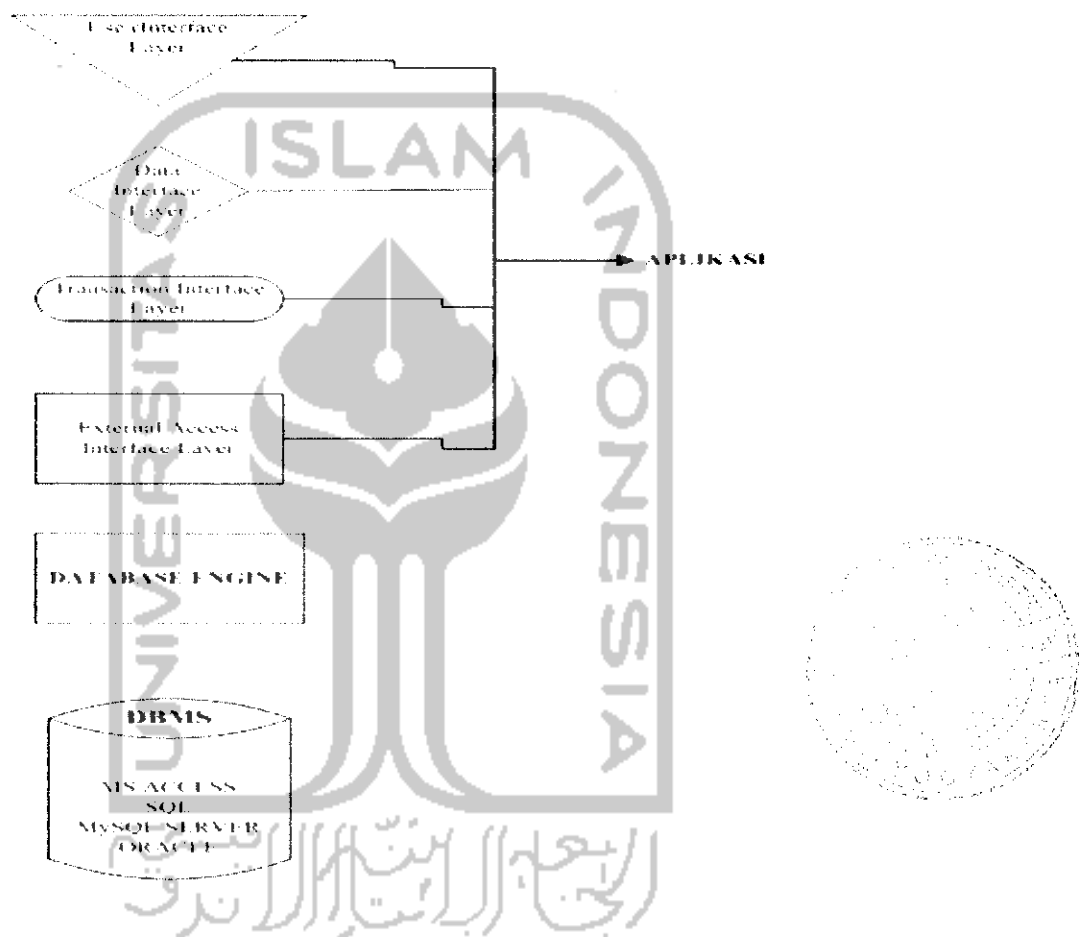
Bagian penghubung antara aplikasi dengan *database* adalah *database engine*. Bagian ini yang bekerja untuk mengatur lalu lintas data dari dan ke pemakai aplikasi. Pada skenario microsoft, terdapat dua jenis *database engine* yang dapat digunakan oleh berbagai jenis aplikasi. *Database engine* tersebut, adalah:

1. *Jet Database Engine*

Jet Database Engine ditujukan bagi aplikasi yang ingin mengakses sekelas *database* desktop (digunakan untuk *database* menengah ke bawah, misalnya *database Microsoft Access*).

2. *SQL Server Engine*

SQL Server Engine ditujukan bagi aplikasi yang ingin mengakses *database* bersekala besar, dengan jumlah pemakai mencapai ribuan. *SQL Server Engine* ini dirancang dengan arsitektur *client/server* dan ditujukan untuk *database* berukuran menengah ke atas (*Enterprise*).



Gambar 2-1 Arsitektur *Database System*

Untuk memanfaatkan layanan-layanan yang disediakan oleh *database engine*, Dapat menggunakan sejumlah model *object* pengaksesan data yang telah dipersiapkan oleh *Microsoft*. Model-model *object* tersebut meliputi:

1. *DAO (data access object)*

Data access object, adalah model pengaksesan *object* paling tua, di mana *engine* yang dapat diaksesnya hanya *jet engine*.

2. RDC (*remote data control*) dan RDO (*remote data object*).

Remote data object, merupakan perbaikan dari DAO dan ODBC (*Open Database Connectivity*), sebagai contoh untuk *database SQL Server* dan *oracle*.

3. *ActiveX data object*, merupakan model pengaksesan object yang menggunakan hirarki atau *overhead* lebih kecil, lebih cepat *database desktop*, sampai *database server*. Selain ODBC, ADO dapat mengakses sumber data lain, yaitu OLE DB (*object linking*).

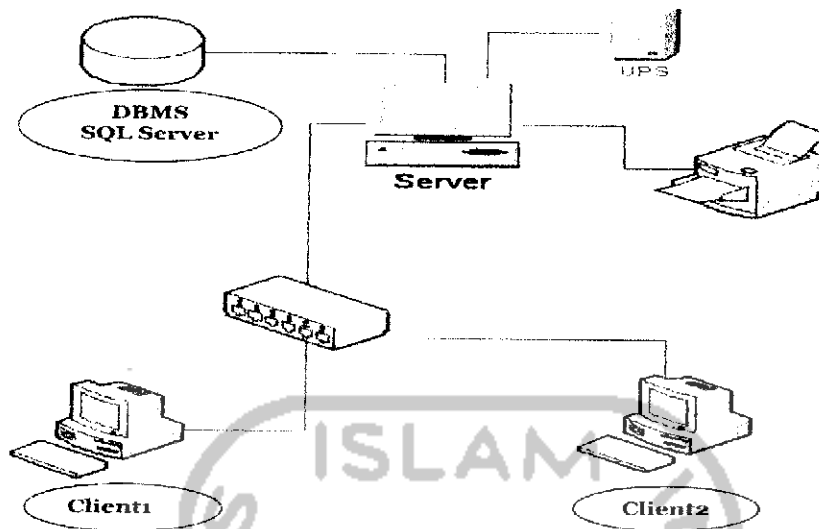
2.3.2 Pemrograman *Client/Server*

Pemrograman *client/server* adalah sebuah dasar dari pengembangan perangkat lunak dari integrasi data dalam lingkungan jaringan komputer. Program yang terdiri dari beberapa bagian yaitu *user interface*, *business logic*, dan data access yang semuanya dibuat dalam satu program, pola ini berkembang pada aplikasi *Personal Computer (PC)*, pada perkembangannya sistem membutuhkan akses data bersama (*shared resource*) melalui teknologi jaringan, sumber daya seperti *disk* dan file dapat diakses bersama, model komputasi ini dikenal dengan nama *disk server* atau *file server* (Darmayuda, 2007). Model komputasi ini sangat penting karena kebutuhan aplikasi akan kecepatan data dan pengaksesan data. Data lebih cepat dipertukarkan melalui teknologi jaringan daripada melalui tangan ke tangan. Untuk mendukung model komputasi ini dibutuhkan beberapa *database* yang mempunyai kelebihan *share data*, yaitu *database* yang mempunyai fasilitas dapat mengirim data yang diminta aplikasi melalui teknologi jaringan. Misalnya *database* yang mendukung model komputasi ini antara lain *Mysql* dan *SqlServer*. Mekanisme alur dari teknologi *client/server* adalah adanya keterkaitan fungsional antara *client/server*. Menurut Darmayuda (2007) *Database sever* menerima request (permintaan) berupa instruksi SQL (*Structure Query Language*), dan sebagai hasil *database server* akan menjawabnya, berhasil atau tidaknya perintah tersebut. Dalam perintah tersebut akan menuntut lokasi pemrosesan, sedangkan lokasi pemrosesan terbagi menjadi dua, yaitu:

1. Komputer Pertama mengolah *input output* ke pemakai (*user interface*). Disebut juga sebagai Presentasi. Di komputer tersebut *program logic* berjalan dan berhubungan dengan *user interface* dan *data interface* (dalam hal ini berbentuk *SQL-Request*)
2. Komputer Kedua menerima *SQL-Request*, memprosesnya dengan mengakses *database*, dan memberikan hasilnya kembali. Sebagai contoh, Komputer Pertama disebut sebagai *client (front end)*, yang terdiri atas aplikasi yang ditulis dalam bahasa pemrograman, misalnya dalam tugas akhir ini digunakan bahasa pemrograman Borland Delphi 2007. sedangkan Komputer Kedua disebut sebagai *server (Back End)*”, dalam hal ini adalah *Database Management System (DBMS)*.

2.3.2.1 Arsitektur Sistem Komputer

Dalam arsitektur pertukaran dan akses data yang terjadi dalam pemrograman *client server*. Dalam hal ini terdapat 3 buah komponen utama yang tidak bisa dipisahkan yaitu *client*, *server*, dan DBMS atau *database*. Data yang terimpan di database, akan ditempatkan dalam komputer *server* di jaringan tersebut. Dari *client* untuk mendapatkan data-data yang diperlukan akan menghubungi komputer *server* dimana data tersebut disimpan, selanjutnya komputer *server* akan memproses pengambilan data dari database dan selanjutnya mendistribusikan ke komputer *client* sehingga komputer *client* mendapatkan data-data tersebut, seperti pada Gambar 2-2.



Gambar 2-2 Arsitektur Komputer

Untuk membuat *database* sistem bekerja sesuai dengan volume atau skala transaksi dan pemakainnya, terlebih dahulu menentukan arsitektur yang akan diimplementasikan pada sistem tersebut. Arsitektur dalam *database* sistem terbagi dalam tiga *tier*. Istilah *tier* untuk menjelaskan pembagian sebuah aplikasi yang melalui client dan server. Pembagian proses kerja adalah bagian utama dari konsep *client/server* saat ini. Saat ini pembagian kerja pada *client* dan *server* telah diatur secara lebih spesifik. Masing-masing *tier* dalam dalam arsitektur *database* adalah sebagai berikut:

1. *Arsitektur Single-tier*

Arsitektur Single-tier merupakan arsitektur yang paling sederhana dimana seluruh komponen *database* sistem secara lohis dikelolmpokan ke dalam sebuah *tier*, contoh: sistem *stand-alone desktop database*.

2. *Arsitektur Two-tier*

Arsitektur Two-tier merupakan arsitektur dimana *database engine* dan *database* terletak pada tier yang berbeda dengan aplikasi. Arsitektur ini disebut sistem *client/server database* sistem seperti MySQL dan SQL Server.

3. *Arsitektur n-tier*

Arsitektur n-tier merupakan arsitektur dimana memisahkan seluruh komponen ke dalam tier-tier yang berbeda untuk menanggulangi masalah skabilitas sistem yang semakin membesar. Pada arsitektur teknologi yang biasanya digunakan adalah *Microsoft Transaction Service (MTS)* dan *Distributed Component Object Model (COM)*, serta *COM +*.

2.3.3 *Stored Procedure*

Stored procedure adalah sekumpulan perintah-perintah umum SQL berbentuk *script* yang berhubungan dengan *database* yang disimpan di *server*.

Stored procedure akan meningkatkan *performance database*, keamanan *database*, dan memungkinkan membuat program untuk *database*. *Stored procedure* dapat dikategorikan menjadi beberapa jenis (Darmayuda, 2007), yaitu :

1. *Stored Procedure System*

Stored Procedure yang disimpan di *database Master*.

2. *Stored Procedure Lokal*

Stored procedure lokal biasanya disimpan dalam *database user* yang memang digunakan untuk kepentingan *database* yang bersangkutan.

3. *Stored Procedure Temporer*

Mirip dengan *stred procedure lokal* tetapi akan ada sampai koneksi yang membuatnya berhenti/ *SQL Server* dimatikan. Terdapat 3 jenis *stored procedure temporer*. Yaitu lokal (privat) yang diawali dengan #, global yang diawali dengan ##, dan yang langsung dibuat di dalam *tempdb*.

4. *Stored Procedure Extended*

Stored procedure ini memakai program eksternal dan dikompilasi sebagai *Dynamic Link Library (DLL) 32 bit* untuk menambahkan kemampuan *stored procedure*.

Function *OBJECT PROPERTY* dipakai untuk menentukan apakah sebuah stored procedure adalah *extended*. Jika hasilnya adalah 1 maka *extended*, jika hasilnya 0 bukan.

5. *Stored procedure remote*

Stored procedure ini berada pada instalasi sel lain. Jenis dipakai untuk kompilasi dengan versi sebelumnya.

Dalam MySQL versi 5.0 mampu mendukung *stored procedure*, *triggers*, *views*, *information schema*. *Stored procedure* mempunyai banyak keunggulan yaitu :

1. Membagi beban *resource* yang terpakai saat aplikasi di jalankan. Apabila semua query dijalankan pada aplikasi/*client* maka *resource* yang terpakai pada client tersebut akan besar, untuk mengatasi hal tersebut perintah *query* tersebut dibuat pada *procedure* ataupun *view* yang dieksekusi pada sisi server.
2. Mempermudah dalam menampilkan data, terkadang struktur *database* yang dibuat mempersulit pekerjaan dalam proses *query* dikarenakan kebutuhan akan proses spesial *reporting* maka pembuatan *view* dapat dijadikan alasan untuk mempermudah reporting.
3. Mempermudah *maintain* aplikasi, apabila ada proses *query* yang sama dan berulang maka digunakan *store procedure* akan membuat lebih sederhana dalam proses pembuatan aplikasi.

2.3.3.1 *Trigger*

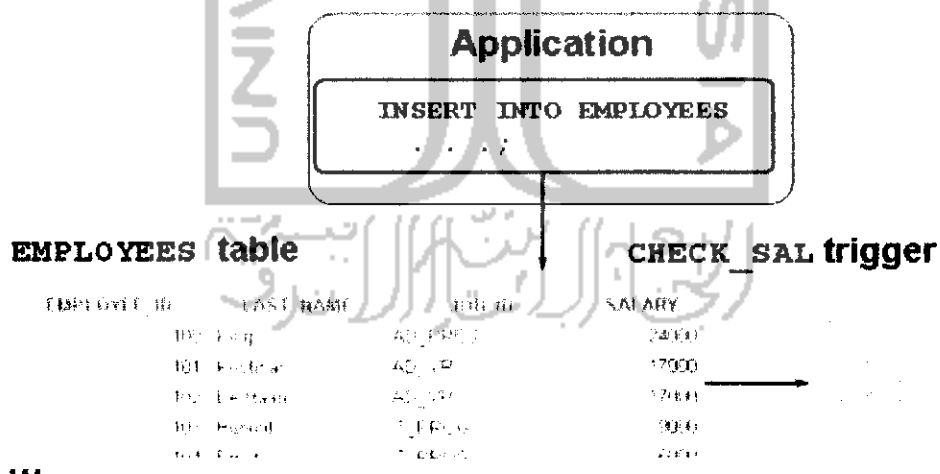
Trigger adalah blok PL/SQL atau prosedur yang berhubungan dengan *table*, *view*, skema atau *database* yang dijalankan secara implicit pada saat terjadi sebuah event (Badriyah, 2007). *trigger* dapat dibedakan menjadi 2 tipe, masing-masing adalah :

1. Application *trigger* : diaktifkan pada saat terjadi *event* yang berhubungan dengan sebuah aplikasi

2. Database *trigger* : diaktifkan pada saat terjadi event yang berhubungan dengan data (seperti operasi DML) atau event yang berhubungan dengan sistem (semisal *logon* atau *shutdown*) yang terjadi pada sebuah skema atau *database*.

Trigger dibuat sesuai dengan keperluan, Ada kalanya *trigger* perlu dibuat, dan kadangkala tidak perlu dibuat. Pembuatan *Trigger* diperlukan untuk membentuk sebuah aksi tertentu terhadap suatu event, selain itu *trigger* juga dibuat untuk memusatkan operasi global. *Trigger* tidak perlu dibuat, apabila fungsionalitas yang diperlukan suatu ada pada Oracle server dan duplikat atau sama dengan fungsi *trigger* yang lain.

Prosedur bisa dibuat dalam *database*, kemudian prosedur tersebut dipanggil pada *trigger*. Jika penggunaan *trigger* terlalu berlebihan, maka akan menyebabkan terjadi sifat ketidaktergantungan yang terlalu kompleks sehingga akan mempersulit pemeliharaan dari aplikasi yang besar. Gambar 2-3 menunjukkan ilustrasi dari penggunaan *trigger* (Badriyah, 2007) :



Gambar 2-3 Ilustrasi dari Penggunaan *Trigger*

Pada gambar tersebut, database *trigger* *check_sal* memeriksa nilai gaji pada saat suatu aplikasi mencoba untuk memasukkan baris baru ke dalam table *employees*. Nilai yang terletak pada jangkauan di luar kategori pekerjaan akan diabaikan. Sebuah *trigger* mempunyai enam komponen yaitu :

1. *Trigger timing*

Trigger timing adalah waktu kapan *trigger* diaktifkan. Ada tiga macam *trigger timing*, yaitu :

- a) BEFORE : *trigger* dijalankan sebelum DML event pada tabel
- b) AFTER : *trigger* dijalankan setelah DML event pada tabel
- c) INSTEAD OF : *trigger* dijalankan pada sebuah *view*.

2. *Trigger event*

Trigger event ada 3 kemungkinan : insert, update atau delete. Pada saat *trigger event* update, *user* dapat memasukkan daftar kolom untuk mengidentifikasi kolom mana yang berubah untuk mengaktifkan sebuah *trigger* (contoh : update of salary ...). Jika tidak ditentukan, maka perubahannya akan berlaku untuk semua kolom pada semua baris.

3. Nama tabel

4. Tipe *trigger*

5. Klausula when

6. *Trigger body*

Secara garis besar *trigger* ada 2 macam tipe dalam pengeksekusiannya, yaitu :

1. Statement : *trigger* dijalankan sekali saja pada saat terjadi sebuah event. Statement *trigger* juga dijalankan sekali, meskipun tidak ada satupun baris yang dipengaruhi oleh event yang terjadi.
2. Row : *trigger* dijalankan pada setiap baris yang dipengaruhi oleh terjadinya sebuah event. Row *trigger* tidak dijalankan jika event dari *trigger* tidak berpengaruh pada satu baris pun. *Trigger body* mendefinisikan tindakan yang

perlu dikerjakan pada saat terjadinya event yang mengakibatkan sebuah *trigger* menjadi aktif.

2.3.3.2 View

View merupakan tampilan data yang terdapat pada satu *table* atau beberapa *table* yang saling berhubungan. Fungsi utama membuat *view* adalah agar data pada *table* aslinya tidak dapat diakses secara langsung. Selain itu *view* dapat dijadikan pengurangan beban resource yang terpakai aplikasi karena aplikasi tidak secara langsung memanggil semua *record*, tetapi *record* telah di *buffer* (disimpan sementara) dalam *view*, selanjutnya aplikasi memanggil data yang terdapat dalam *view*.



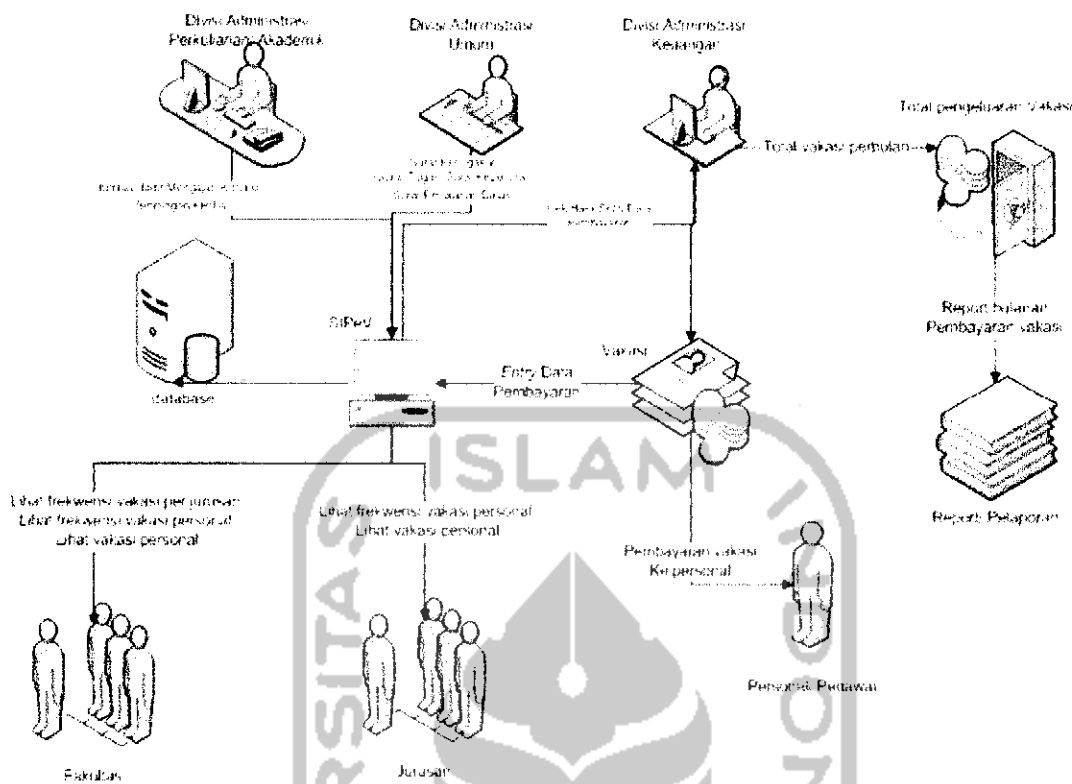
BAB 3

ANALISIS KEBUTUHAN PERANGKAT LUNAK

Pada bab ini akan disampaikan tahapan analisis kebutuhan perangkat lunak yang merupakan langkah awal dalam pengembangan perangkat lunak Sistem Informasi Pembayaran Vakasi Fakultas Teknologi Industri Universitas Islam Indonesia (SIPeV FTI UII). Analisis kebutuhan ini meliputi tinjauan, dan analisis kebutuhan perangkat lunak.

3.1 Tinjauan Sistem

SIPeV yang dikembangkan merupakan sistem berbasis visual atau *desktop*. Sistem ini dikembangkan memprioritaskan pada level operasional tetapi tidak melupakan level manajerial, pada level operasional akan mempermudah pemasukan data, sedangkan dalam level manajerial sistem ini membantu masalah *monitoring* atau pengawasan terhadap personal di unit struktural FTI UII. Alur proses yang akan berjalan di sistem ini diawali dari dua Divisi Administrasi Umum (DAU) dan Divisi Administrasi Perkuliahan (DAP) yang membawahi Bagian kerja praktek (KP) dan Tugas Akhir (TA). Data yang masuk dari kedua divisi tersebut kemudian diproses oleh sistem yang akhirnya digunakan oleh Divisi Administrasi Keuangan (DAK) untuk melakukan pembayaran kepada personal yang telah melakukan tugasnya di unit struktural FTI UII, setiap data pembayaran terlebih dahulu akan diproses oleh sistem sehingga terbentuk rekapan hasil pemasukan data pembayaran vakasi yang dilakukan oleh bagian keuangan. Level manajerial (Fakultas dan Jurusan) berhak mengetahui frekwensi hasil vakasi yang didapat oleh masing personal sebagai referensi untuk mengambil keputusan dalam penugasan selanjutnya kepada setiap personal, seperti dijelaskan pada Gambar 3-1.



Gambar 3-1 Alur Proses Sistem SIPeV

3.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan analisis terhadap semua kebutuhan yang diperlukan perangkat lunak yang akan dikembangkan dalam hal ini adalah pengembangan SIPeV itu sendiri. Analisis kebutuhan perangkat lunak mencakup analisis terhadap kebutuhan fungsional (kebutuhan akan fungsi-fungsi) dan analisis terhadap kebutuhan non-fungsional (kebutuhan-kebutuhan selain fungsi). Analisis kebutuhan yang dilakukan terhadap perangkat lunak akan menghasilkan spesifikasi kebutuhan perangkat lunak tersebut.

SIPeV dikembangkan berdasarkan memenuhi kebutuhan baik level operasional dan level manajerial, masing-masing level mempunyai keterkaitan dengan sistem ini, pada level operasional bertugas lebih mengenai penanganan data sedangkan pada level manajerial mendapatkan hasil dari pengolahan data yang telah dimasukkan.

3.2.1 Analisis Kebutuhan fungsional Perangkat Lunak

Analisis kebutuhan fungsional perangkat lunak merupakan analisis terhadap kebutuhan-kebutuhan fungsional yang diperlukan di dalam lingkup SIPeV yang akan dikembangkan. Seperti yang telah dijelaskan pada point 3.1 tentang tinjauan sistem, kebutuhan fungsional SIPeV meliputi data awal dari beberapa divisi seperti Divisi Administrasi Umum (DAU) mengumpulkan dan memasukan kedalam sistem berupa data Surat Penugasan (surat tugas, surat keputusan, surat perjalanan dinas), Divisi Administrasi Perkuliahan/Akademik (DAP) mengumpulkan dan memasukan kedalam sistem berupa data jumlah hadir, mengajar, koreksi, bimbingan KP/TA, Divisi Administrasi Keuangan (DAK) mengumpulkan dan memasukan kedalam sistem berupa data pembayaran vakasi dimana setiap personal mendapatkan vakasannya setelah melaksanakan tugasnya. Kebutuhan fungsional yang kedua adalah tentang penyimpanan data yang telah di-*entry* oleh masing-masing divisi, dan kebutuhan fungsional yang terakhir adalah penyajian data yang berupa pelaporan data yang telah diolah dan direkap oleh DAK.

3.2.1.1 Proses Login

Login digunakan untuk membedakan dan membatasi akses ke dalam sistem sesuai dengan masing – masing, seperti penjelasan di atas terdapat lima divisi yaitu DAU, DAK, DAP, jurusan, dan fakultas serta satu hak akses super *user* yaitu admin . Terdapat beberapa data masukan untuk login:

1. *Username*
2. *Password*

3.2.1.2 Proses Manajemen Surat

Dalam manajemen surat terdapat tiga proses yaitu:

1. Penambahan data surat, Kebutuhan masukan untuk proses penambahan data surat dikelompokkan menjadi 2 yaitu :

a) Surat Perjalanan Dinas (SPJ)

Kebutuhan masukannya adalah :

- Nomor surat
Nomor surat yang dikeluarkan oleh DAU akan di-*entry* manual, dikarenakan terdapat nomor surat yang dikeluarkan oleh pihak fakultas tanpa diproses oleh sistem.
- Maksud atau tujuan
- NIP
- Alamat, untuk alamat dapat diisi default yaitu alamat Universitas Islam Indonesia (UII).
- Tanggal pembuatan surat
- Tanggal keberangkatan
- Tanggal kembali
- Waktu
- Kota tujuan
- Kode Post-tarif
- Tarif
- Total biaya

b) Surat Tugas (SK) atau Surat Keputusan (SK)

Kebutuhan masukannya adalah :

- Nomor surat
- Nama tugas atau nama tim
- Tanggal pembuatan surat
- Waktu
- NIP
- Jabatan dalam tim
- Kode post-tarif
- Tarif

2. Perubahan data surat, seperti halnya pada penambahan data surat, kebutuhan masukan untuk proses perubahan data surat dikelompokkan menjadi 2 yaitu

a) Surat Perjalanan Dinas (SPJ)

Kebutuhan masukannya adalah :

- Nomor surat
Nomor surat yang dikeluarkan oleh DAU akan di-entry manual, dikarenakan terdapat nomor surat yang dikeluarkan oleh pihak fakultas tanpa diproses oleh sistem.
- Maksud atau tujuan
- NIP
- Alamat, untuk alamat dapat diisi default yaitu alamat Universitas Islam Indonesia (UII).
- Tanggal pembuatan surat
- Tanggal keberangkatan
- Tanggal kembali
- Waktu
- Kota tujuan
- Kode Post-tarif
- Tarif
- Total biaya

b) Surat Tugas (SK) atau Surat Keputusan (SK)

Kebutuhan masukannya adalah :

- Nomor surat
- Nama tugas atau nama tim
- Tanggal pembuatan surat
- Waktu
- NIP
- Jabatan dalam tim
- Kode post-tarif
- Tarif

3. Penghapusan data, kebutuhan masukan untuk proses penghapusan data surat dikelompokkan menjadi 3 yaitu:

a) Penghapusan data pelaksana yang berada dalam Surat Tugas (ST) atau Surat Keputusan (SK).

Kebutuhan masukannya adalah :

- NIP
- Jabatan

b) Penghapusan data akomodasi yang berada dalam Surat Perjalanan Dinas (SPJ).

Kebutuhan masukannya adalah :

- Kode post-tarif
- Tarif

c) Penghapusan data Surat.

Kebutuhan masukannya adalah :

- Nomor surat
- Jenis surat

3.2.1.3 Proses Manajemen Personal

Dalam manajemen personal terdapat tiga proses yaitu:

1. Penambahan data personal, dalam hal ini sistem membutuhkan:

- a) NIP
- b) Nama
- c) Pendidikan
- d) Golongan
- e) Jabatan akademik (jabatan struktural)
- f) Klarifikasi
- g) Status pegawai
- h) Jurusan
- i) Tempat lahir
- j) Tanggal lahir
- k) Jenis kelamin

- l) Alamat
 - m) Kode bank
 - n) Nomor rekening
 - o) Kode jurusan
2. Pengubahan data personal, sistem membutuhkan masukan berupa:
- a) NIP
 - b) Nama
 - c) Pendidikan
 - d) Golongan
 - e) Jabatan akademik (jabatan struktural)
 - f) Klarifikasi
 - g) Status pegawai
 - h) Jurusan
 - i) Tempat lahir
 - j) Tanggal lahir
 - k) Jenis kelamin
 - l) Alamat
 - m) Kode bank
 - n) Nomor rekening
 - o) Kode jurusan
3. Penghapusan data personal, sistem membutuhkan masukan berupa NIP

3.2.1.4 Proses Manajemen Post-tarif

Dalam manajemen post-tarif terdapat tiga proses yaitu:

1. Penambahan data post-tarif, kebutuhan masukannya adalah
 - a) Kode post-tarif
 - b) Nama kegiatan
 - c) Tarif
 - d) Satuan
 - e) Nilai satuan

2. Perubahan data post-tarif, kebutuhan masukannya adalah
 - a) Kode post-tarif
 - b) Nama kegiatan
 - c) Tarif
 - d) Satuan
 - e) Nilai satuan
3. Penghapusan data post-tarif, kebutuhan masukannya adalah
 - a) Kode post-tarif
 - b) Nama kegiatan

3.2.1.5 Proses Manajemen Aktivitas

Dalam manajemen post-tarif terdapat tiga proses yaitu:

1. Penambahan data aktivitas, kebutuhan masukannya adalah
 - a) NIP
 - b) Tanggal aktivitas
 - c) Keterangan Aktivitas
 - d) Nilai
 - e) Kode posttarif
 - f) Tarif
 - g) Bonus, terdapat dua pilihan yaitu Ya dan Tidak
 - h) Nilai bonus
 - i) Keterangan bonus
2. Perubahan data aktivitas, kebutuhan masukannya adalah
 - a) NIP
 - b) Tanggal aktivitas
 - c) Keterangan Aktivitas
 - d) Nilai
 - e) Kode posttarif
 - f) Tarif
 - g) Bonus, terdapat dua pilihan yaitu Ya dan Tidak
 - h) Nilai bonus

i) Keterangan bonus

3. Penghapusan data aktivitas, kebutuhan masukannya adalah kode aktivitas

3.2.1.6 Proses Manajemen Perbankan

Dalam manajemen perbankan terdapat tiga proses yaitu:

1. Penambahan data Perbankan, kebutuhan masukannya adalah
 - a) Kode bank
 - b) Nama bank
2. Pengubahan data Perbankan, kebutuhan masukannya adalah :
 - a) Kode bank
 - b) Nama bank
3. Penghapusan data Perbankan, kebutuhan masukannya adalah kode bank

3.2.1.7 Proses Manajemen Jurusan

Dalam manajemen Jurusan terdapat tiga proses yaitu:

1. Penambahan data Jurusan, kebutuhan masukannya adalah:
 - a) Kode jurusan
 - b) Nama jurusan
2. Pengubahan data Jurusan, kebutuhan masukannya adalah:
 - a) Kode jurusan
 - b) Nama jurusan
3. Penghapusan data Jurusan, kebutuhan masukannya adalah kode jurusan

3.2.1.8 Proses Manajemen Pembayaran Vakasi

Dalam manajemen pembayaran vakasi terdapat tiga proses yaitu:

1. Penambahan data pembayaran vakasi, kebutuhan masukannya adalah:
 - a) NIP
 - b) Kode aktivitas
 - c) Tanggal dibayar
 - d) Total bayar
 - e) Media pembayaran, terdapat dua pilihan yaitu *cash* dan *transfer*

2. Perubahan data pembayaran vakasi, kebutuhan masukannya adalah:
 - a) NIP
 - b) Kode aktivitas
 - c) Tanggal dibayar
 - d) Total bayar
 - e) Media pembayaran, terdapat dua pilihan yaitu *cash* dan *transfer*
3. Penghapusan data pembayaran vakasi, kebutuhan masukannya adalah:
 - a) NIP
 - b) Kode aktivitas

3.2.1.9 Proses cek *report*

Dalam cek data *report* terdapat beberapa dua proses yaitu:

1. Bulan
2. Tahun

3.2.1.10 Proses Cetak *report* Personal

Dalam cetak *report* Personal terdapat beberapa satu proses yaitu Pencetakan *report* bulanan Personal, kebutuhan masukannya adalah:

1. NIP
2. Nama
3. Nama aktivitas
4. Total bayar
5. Tanggal dibayarkan

3.2.1.11 Proses Manajemen Matakuliah

Dalam manajemen matakuliah terdapat tiga proses yaitu:

1. Penambahan data matakuliah, kebutuhan masukannya adalah
 - a) Id matakuliah
 - b) Kode jurusan
 - c) Semester
 - d) Tahun ajaran
 - e) Kode matakuliah

- f) Nama matakuliah
 - g) Jumlah kelas yang dibuka
 - h) Sks
2. Pengubahan data matakuliah, kebutuhan masukannya adalah:
- a) Id matakuliah
 - b) Kode jurusan
 - c) Semester
 - d) Tahun ajaran
 - e) Kode matakuliah
 - f) Nama matakuliah
 - g) Jumlah kelas yang dibuka
 - h) sks
3. Penghapusan data matakuliah, kebutuhan masukannya adalah id matakuliah

3.2.1.12 Proses Manajemen Mengajar

Dalam manajemen mengajar terdapat tiga proses yaitu:

1. Penambahan data mengajar, kebutuhan masukannya adalah
- a) Id mengajar
 - b) NIP
 - c) Kode matakuliah.
 - d) Jumlah mahasiswa
 - e) Jumlah hadir dosen
 - f) Kode posttarif
 - g) tarif
2. Pengubahan data mengajar, kebutuhan masukannya adalah:
- a) Id mengajar
 - b) NIP
 - c) Kode matakuliah.
 - d) Jumlah mahasiswa
 - e) Jumlah hadir dosen
 - f) Kode posttarif

g) tarif

3. Penghapusan data mengajar, kebutuhan masukannya adalah id mengajar

3.2.1.13 Proses Manajemen General (Profil)

Dalam manajemen general terdapat beberapa dua proses yaitu:

1. Penambahan data manajemen general, kebutuhan masukannya adalah:
 - a) Nama Profil
 - b) Nomor telepon
 - c) *Faximile*
 - d) Kota
 - e) Alamat profile
2. Pengubahan data manajemen *general*, kebutuhan masukannya adalah:
 - a) Nama Profil
 - b) Nomor telepon
 - c) *Faximile*
 - d) Kota
 - e) Alamat profile

3.2.1.14 Proses Manajemen *Setting*

Dalam manajemen general, bertujuan untuk mengatur dasar *database* yang digunakan dan *setting* yang ada dalam aplikasi. Terdapat beberapa dua proses yaitu:

1. Penambahan data manajemen *setting*, kebutuhan masukannya adalah:
 - a) Nama database
 - b) *User*
 - c) *Password*
 - d) *Protocol*
 - e) *Host*
 - f) *Port*
2. Pengubahan data manajemen *setting*, kebutuhan masukannya adalah:

- a) Nama database
- b) *User*
- c) *Password*
- d) *Protocol*
- e) *Host*
- f) *Port*

3.2.1.15 Proses Manajemen *User*

Dalam manajemen *user*, kebutuhan masukannya adalah:

1. Penambahan data manajemen *user*, kebutuhan masukannya adalah:

- a) Id user
- b) *Username*
- c) *Password*
- d) Set manajemen *user*
- e) Set manajemen perbankan
- f) Set manajemen pembayaran vakasi
- g) Set manajemen personal
- h) Set manajemen *report*
- i) Set manajemen *report* personal
- j) Set manajemen general/profil
- k) Set manajemen post-tarif
- l) Set manajemen surat
- m) Set manajemen aktivitas
- n) Set manajemen mengajar
- o) Set manajemen matakuliah
- p) Set lihat frekwensi vakasi personal
- q) Set lihat frekwensi jurusan
- r) Set lihat vakasi personal
- s) Set manajemen jurusan

2. Perubahan data manajemen *user*, kebutuhan masukannya adalah:
 - a) Id user
 - b) *Username*
 - c) *Password*
 - d) Set manajemen *user*
 - e) Set manajemen perbankan
 - f) Set manajemen pembayaran vakasi
 - g) Set manajemen personal
 - h) Set manajemen *report*
 - i) Set manajemen *report* personal
 - j) Set manajemen general/profil
 - k) Set manajemen post-tarif
 - l) Set manajemen surat
 - m) Set manajemen aktivitas
 - n) Set manajemen mengajar
 - o) Set manajemen matakuliah
 - p) Set lihat frekwensi vakasi personal
 - q) Set lihat frekwensi jurusan
 - r) Set lihat vakasi personal
 - s) Set manajemen jurusan
3. Penghapusan data mengajar, kebutuhan masukannya adalah id user

3.2.1.16 Proses Lihat Vakasi

Dalam lihat vakasi terdapat tiga proses yaitu:

1. Lihat frekwensi vakasi jurusan, kebutuhan masukannya adalah bulan dan tahun.
2. Lihat frekwensi vakasi personal, kebutuhan masukannya adalah kode jurusan, bulan, dan tahun.
3. Lihat vakasi personal, kebutuhan masukannya adalah NIP, bulan, dan tahun.

SIPeV bertugas memudahkan memajemen data-data mengenai vakasi di fakultas selanjutnya akan menghasilkan suatu informasi maka harus ada data yang

akan terlebih dahulu kedalam sistem dan selanjutnya akan diolah dan kemudian disajikan menjadi suatu informasi. Seperti yang telah dijelaskan diatas, masing-masing divisi mempunyai peran dan fungsinya sendiri-sendiri.

3.2.2 Analisis Kebutuhan non-fungsional Perangkat Lunak

Kebutuhan non-fungsional merupakan kebutuhan perangkat lunak yang menyangkut aspek unjuk kerja (*performance*) yang harus dipenuhi oleh perangkat lunak yang akan dikembangkan. Analisis kebutuhan nonfungsional perangkat lunak SIPeV lebih mengarah kepada kebutuhan membangun sistem meliputi :

1. Kebutuhan perangkat keras (*hardware*)

Untuk membangun aplikasi SIPeV dibutuhkan perangkat keras yang memadai yaitu:

- Minimum Personal Computer (PC) P4 3,06 Mhz,
- RAM Memory 2 Gb.
- Hard disk space 80 Gb.
- Printer minimum 11 Ppm.
- Monitor minimum EGA/VGA.
- Mouse
- Keyboard

2. Kebutuhan perangkat lunak (*software*)

Tidak hanya dukungan perangkat keras, akan tetapi dukungan perangkat lunak sangat menentukan hasil dari pembangunan aplikasi SIPeV, perangkat lunak yang digunakan untuk membuat aplikasi SIPeV, yaitu:

- Rational Rose Rational Rose Enterprise Edition 2002.

Dengan aplikasi ini dapat dibuat cetak biru (*blue print*) berfungsi untuk perancangan sebagai UML *modelling*.

- Code Gear Delphi 2009.

Delphi 2009 merupakan penyempurnaan dari delphi sebelumnya, terdapat beberapa alasan yang dikemukakan oleh Lanusse (2009) "alasan untuk beralih ke delphi 2009 antara lain, adalah fitur-fitur yang

disediakan delphi 2009 sudah lengkap, delphi 2009 telah mendukung banyak dokumentasi dan mendukung UML *modeling*". Sedangkan alasan utamanya adalah untuk menciptakan aplikasi yang mempunyai kualitas yang bagus dengan menambah *performance* di dalam aplikasi tersebut.

- Adobe Photoshop CS 4.

Adobe photoshop merupakan perangkat lunak yang digunakan untuk melakukan pengubahan gambar. Perangkat lunak ini membantu dalam mendisain gambar dalam pembuatan suatu situs web, sehingga situs web akan lebih menarik. Adobe Photoshop memberikan kemudahan dalam pemberian efek-efek dan disain dalam gambar, dan dapat dikonversikan ke dalam *format* gambar yang sangat kecil seperti GIF dan JPEG.

- SQL *manager* 2010 for MySQL.

SQL *manager* 2010 for MySQL merupakan salah satu manajemen MySQL untuk memudahkan pengguna atau *user* dalam mengolah *database* di MySQL. Dalam aplikasi ini menyediakan fasilitas untuk pembuatan *stored procedure*, *trigger*, *view*.

3. Kebutuhan media penyimpanan (*database*):

Dalam menyimpan data aplikasi SIPeV membutuhkan MySQL *minimum* version 5.0. MySQL merupakan sebuah program *database server* yang mampu menerima dan mengirimkan data dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*). MySQL Versi 5.0 atau yang lebih tinggi telah mendukung adanya *stored procedure*, *trigger*, *view*.

4. Kebutuhan sistem operasi:

Kebutuhan akan sistem operasi yang digunakan untuk membangun SIPeV adalah *Windows eXperience Premium (XP) Service Pack (SP) 2*. *Windows XP* berfungsi sebagai sistem operasi, yaitu suatu perangkat lunak dasar yang berfungsi sepenuhnya untuk mengendalikan sistem komputer. *Windows XP* merupakan salah satu jenis sistem operasi *Windows*.

BAB 4

PERANCANGAN PERANGKAT LUNAK SIPeV

Pada bab ini akan dibahas perancangan Sistem Manajemen Pembayaran Vakasi Fakultas Teknologi Industri Universitas Islam Indonesia (SIPeV FTI UII), khususnya pada bagian pemrosesan dan penyajian data. Model perancangan dari SIPeV menggunakan *Unified 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).

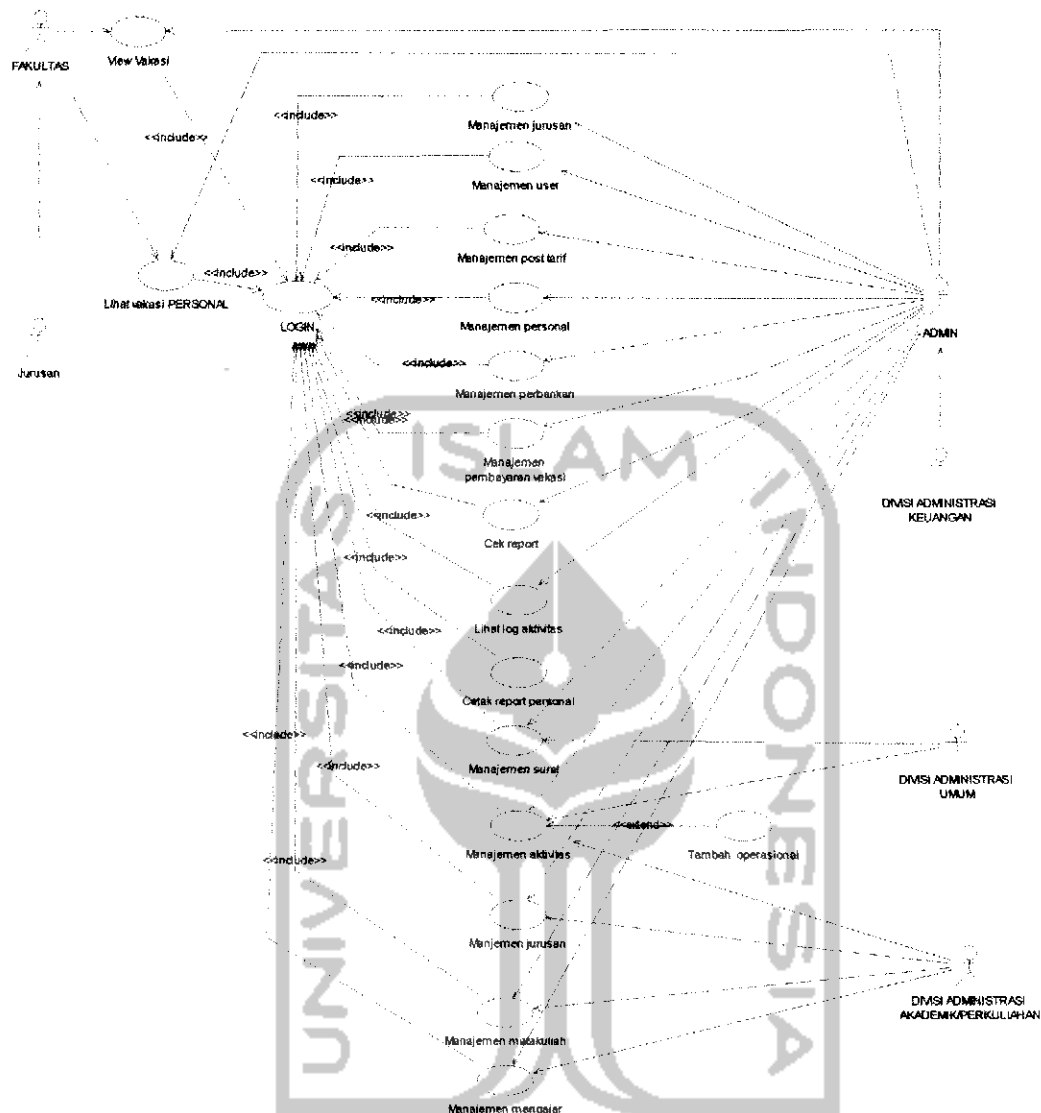
3. Relasi

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.





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 mendapat 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 mendapatkan 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) penggunaan sistem SIPeV yang telah dilakukan oleh *user*.

4.1.2 Class Diagram

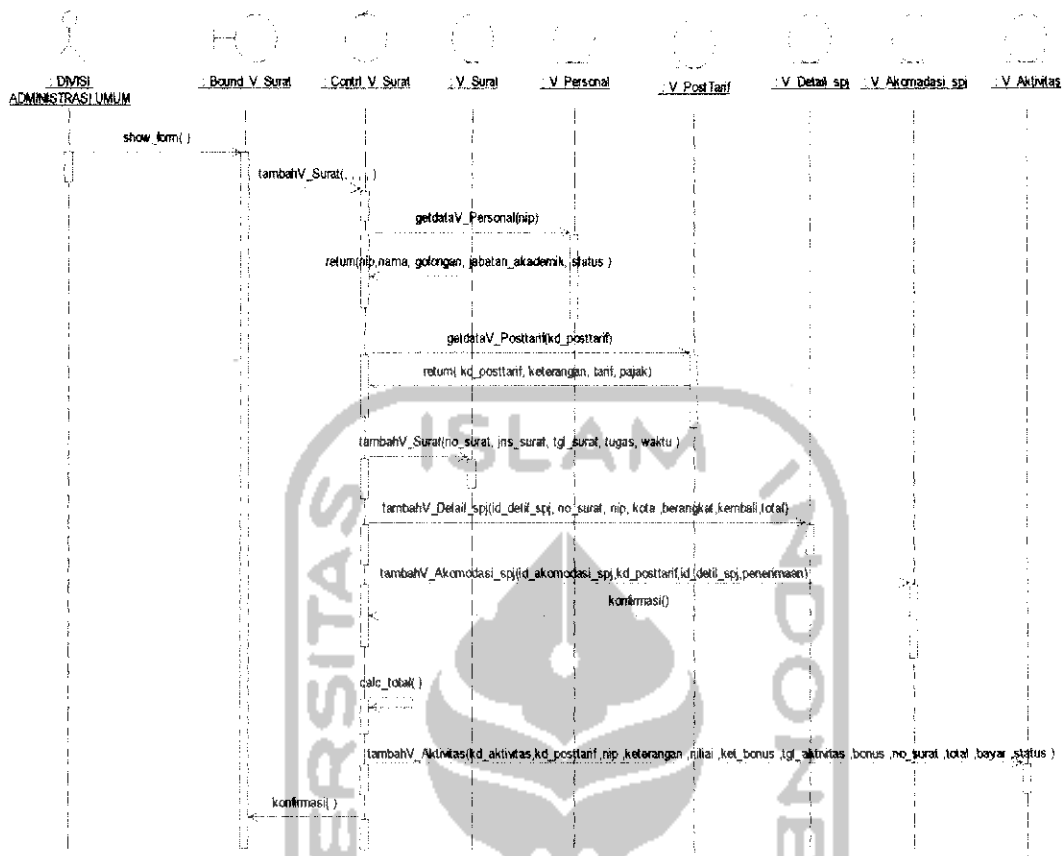
Class Diagram menggambarkan struktur dan deskripsi *class*, *package*, dan *object* beserta hubungan antar-*class* di dalam sistem seperti *containment*, pewarisan, asosiasi, dan lain-lain (Sirkel, 2008). Pada umumnya *Class diagram* tersusun dari elemen *class*, *interface*, *dependency*, *generalization*,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

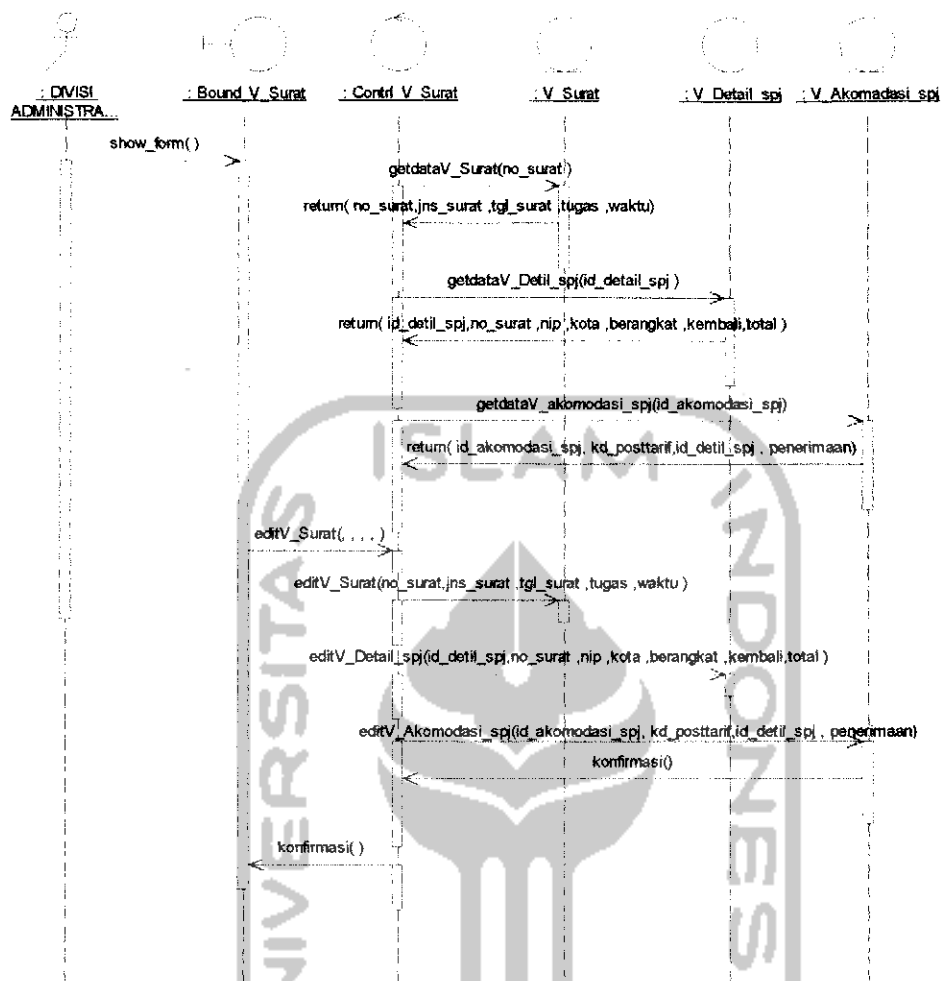
Sequence diagram kali ini juga terdapat *class object* yang berorientasi dengan *form/ boudary* (berinteraksi langsung antara aktor dan *interface/ tampilan*) dan proses/ *controler* (aktivitas yang ada dalam setiap *boundary*), seperti yang digambarkan pada Gambar 4-2.



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

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

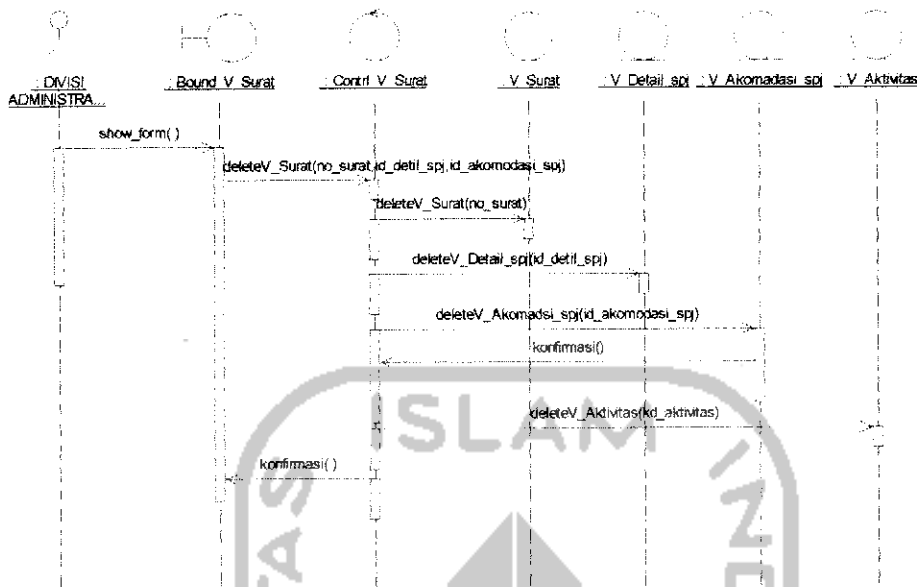
Sequence Diagram edit SPJ dilakukan oleh DAU jika data SPJ yang telah ditambahkan terjadi kesalahan. Maka DAU dapat memanggil data yang salah kemudian dilakukan perubahan (edit). Pada proses perubahan akan mempengaruhi data total biaya aktivitas.



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.

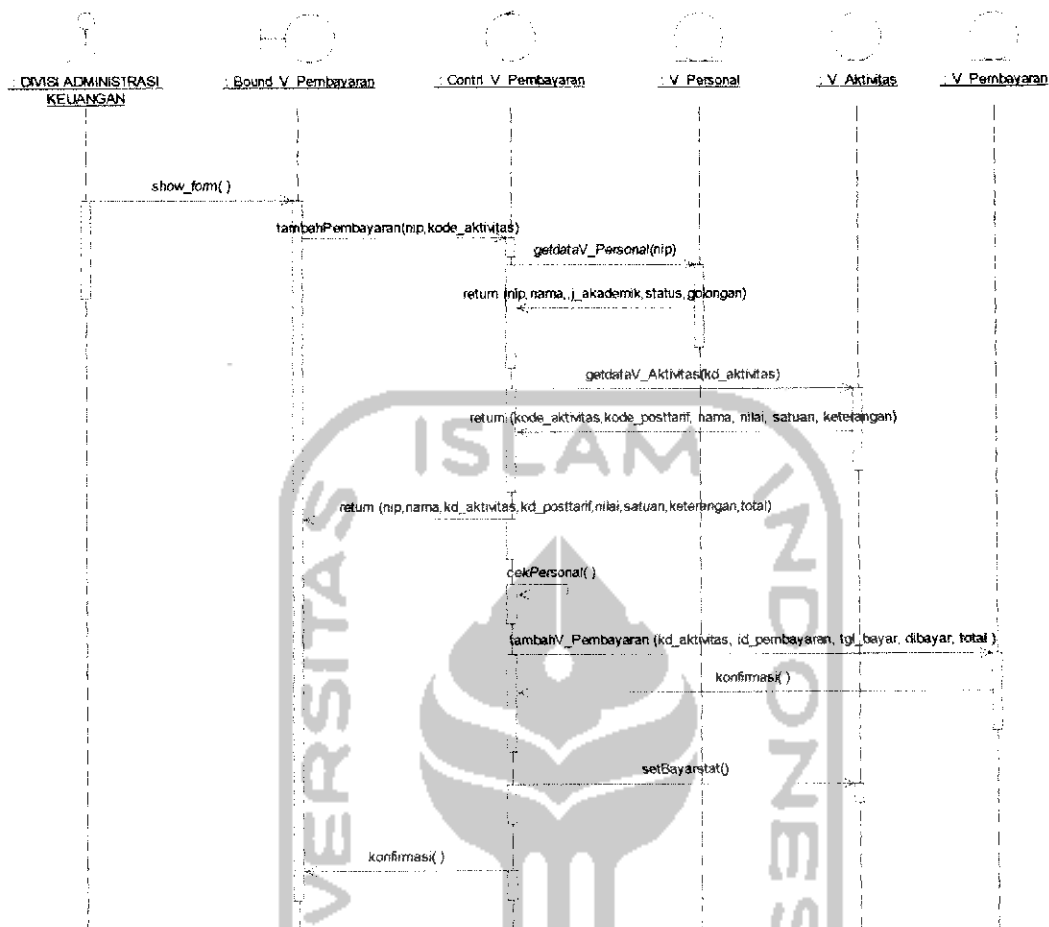
Sequence Diagram delete SPJ dilakukan oleh DAU jika data SPJ yang telah ditambahkan terjadi kesalahan atau tidak terpakai. Maka DAU dapat memanggil data yang salah kemudian dilakukan penghapusan (*delete*). Pada proses penghapusan akan otomatis menghapus data aktifitas SPJ berdasarkan NIP.



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

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

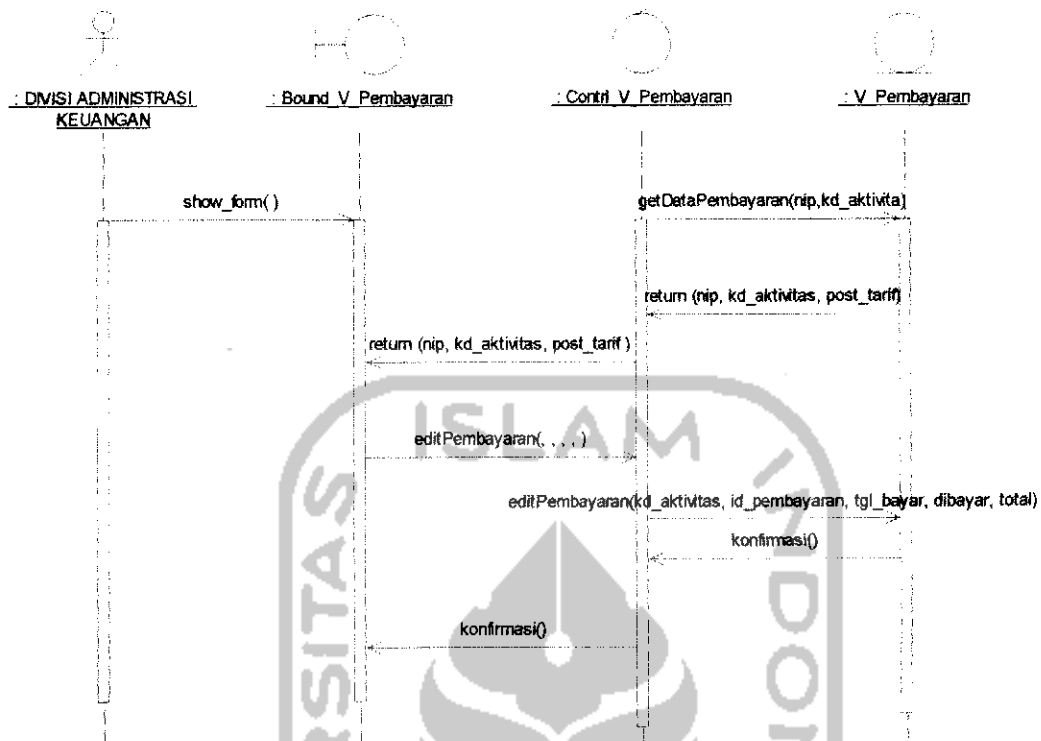
Sequence Diagram Tambah Pembayaran pada SIPeV dimulai dengan aktor yang dalam hal ini Divisi Administrasi Keuangan (DAK) mengisi *form* manajemen pembayaran, selanjutnya DAK memasukkan NIP dan sistem akan mencari detil NIP dari database. Begitu pula jika telah memasukkan kode aktivitas maka sistem akan 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

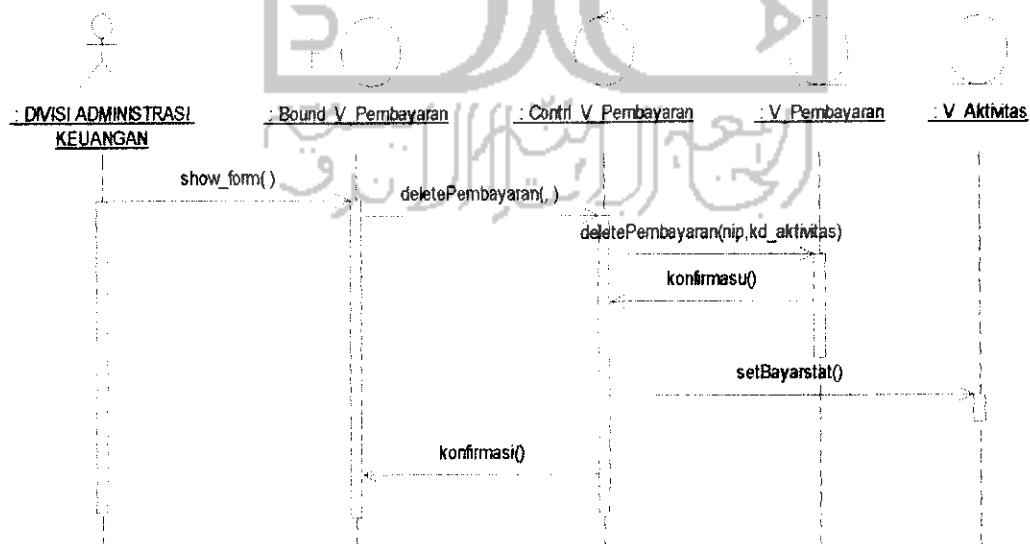
5. Sequence diagram edit pembayaran yang akan digambarkan pada Gambar 4-8.

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 *Sequence Diagram* Edit Pembayaran

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



Gambar 4-9 *Sequence 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.

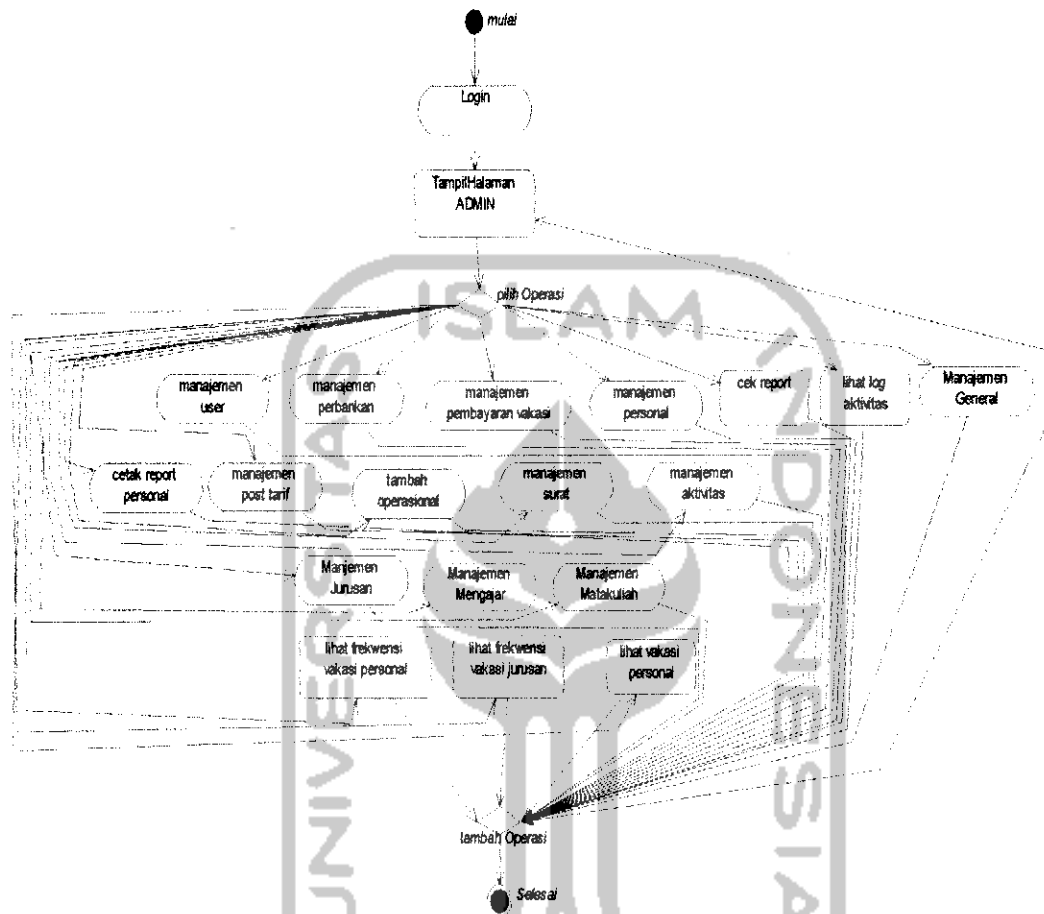
4.1.4 *Activity Diagram*

Activity Diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis, 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 menunjukkan 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 menunjukkan 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, manajemen 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

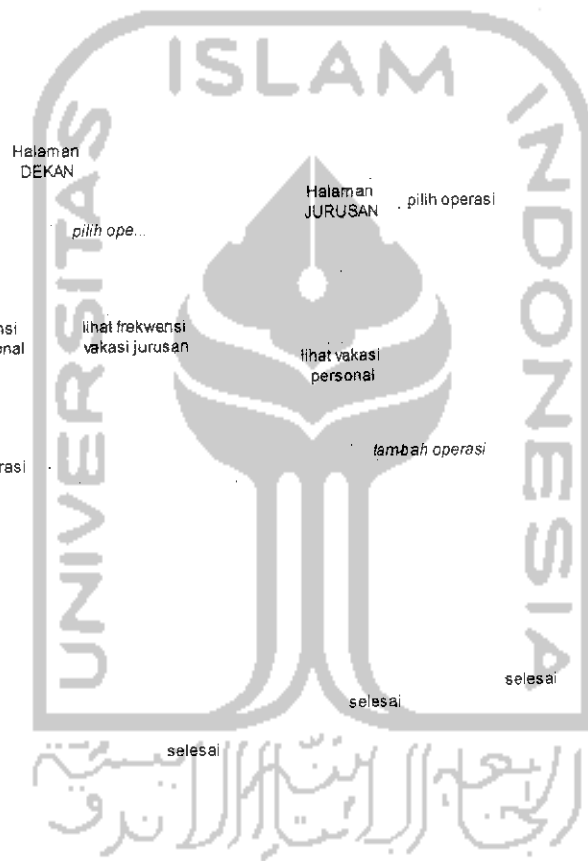
JURUSAN

USER

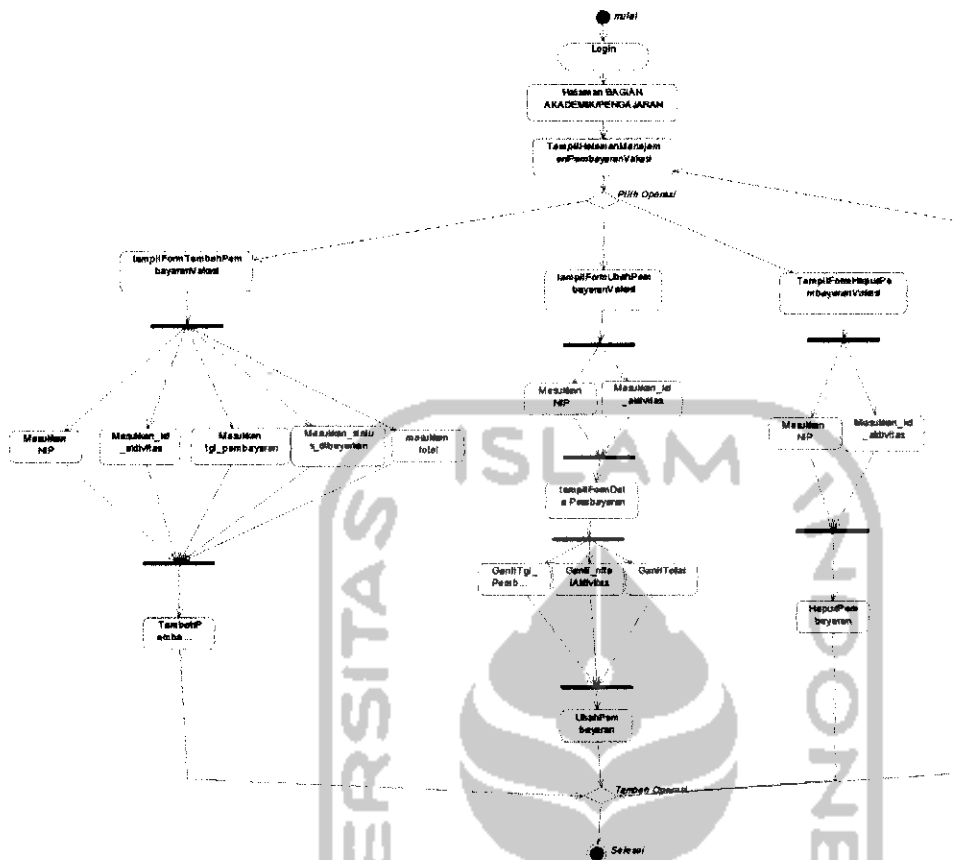
● mulai

Login

authentifikasi



● Selesai



Gambar 4-12 Activity Diagram Halaman Manajemen Pembayaran Vaksinasi.

Pada gambar diatas menjelaskan DAK dalam aktivitas manajemen pembayaran vaksinasi. DAK harus *login* dahulu setelah *login* berhasil, maka akan tampil halaman manajemen pembayaran vaksinasi, untuk menambah data DAK dapat memilih *form* tambah pembayaran vaksinasi dengan memasukkan 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 *programming 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* tereksekusi 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_aktivitas

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null
✓	kd_aktivitas	INTEGER	11	0			✓		✓
	kd_postarif	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 detail 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						
	telepon	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	U
✓	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	6	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 mengenai surat perjalanan dinas. Surat perjalanan 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 perjalanan 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_postarif	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 *privilege 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			✓		✓
	usemama	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, facsimile, 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					✓
	telep_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
✓	uid_log_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_log_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 menunjukkan pula relasi atau hubungan data yang terdapat dalam masing-masing tabel. Relasi tabel hanya menunjukkan

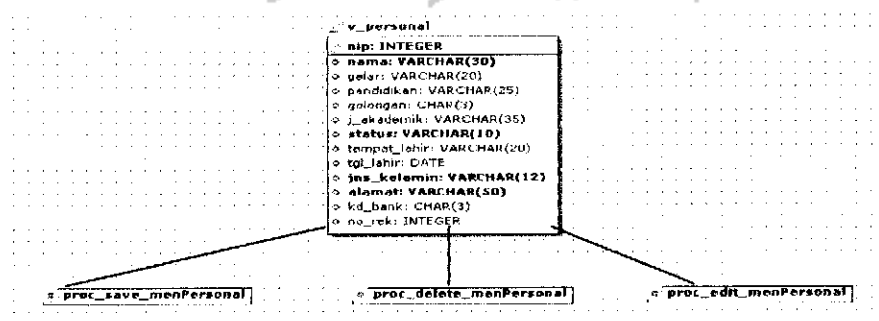
relasi antar tabel, bukan termasuk hubungan antara tabel dengan *stored procedure*, sebab *stored procedure* berdiri sendiri dan berada di luar tabel, tetapi berhubungan dengan *trigger*. *Trigger* berada di dalam tabel masing-masing sesuai dengan aksi (*insert*, *update*, *delete*) yang dilakukan. Dalam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

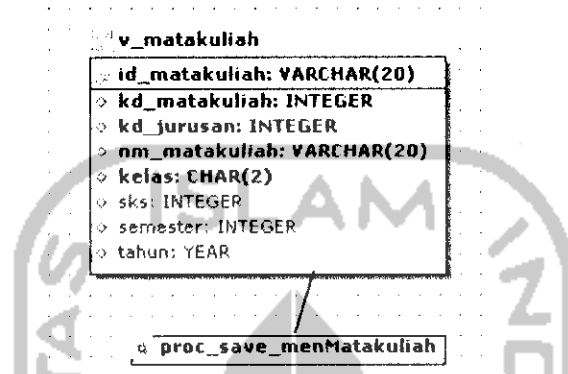
Stored procedure yang digunakan terdapat tiga macam, *proc_save_menPersonal* adalah *stored procedure* untuk menyimpan data ke dalam table *v_personal*, *proc_delete_menPersonal* adalah *stored procedure* untuk menghapus data ke dalam table *v_personal*, dan *proc_edit_menPersonal* adalah *stored procedure* 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*.

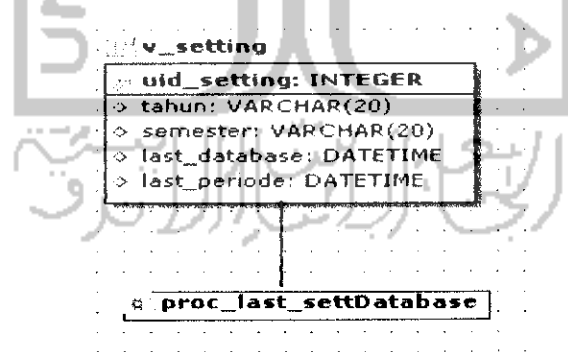
Stored Procedure yang digunakan untuk menyimpan data matakuliah kedalam tabel *v_matakuliah*. *Stored Procedure* terkait dengan tabel *v_matakuliah* akan digambarkan pada Gambar 4-14.



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.

v_detil_spj	
Id_detil_spj:	INTEGER
no_surat:	VARCHAR(30)
nip:	INTEGER
kota:	VARCHAR(20)
berangkat:	DATE
kembali:	DATE
total:	DOUBLE

proc_save_menSurat_SPJ	

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 menunjukkan proses pengecekan terhadap komponen, data dan database yang diperlukan oleh sistem untuk menjalankan program selanjutnya.

2. Halaman *Login*

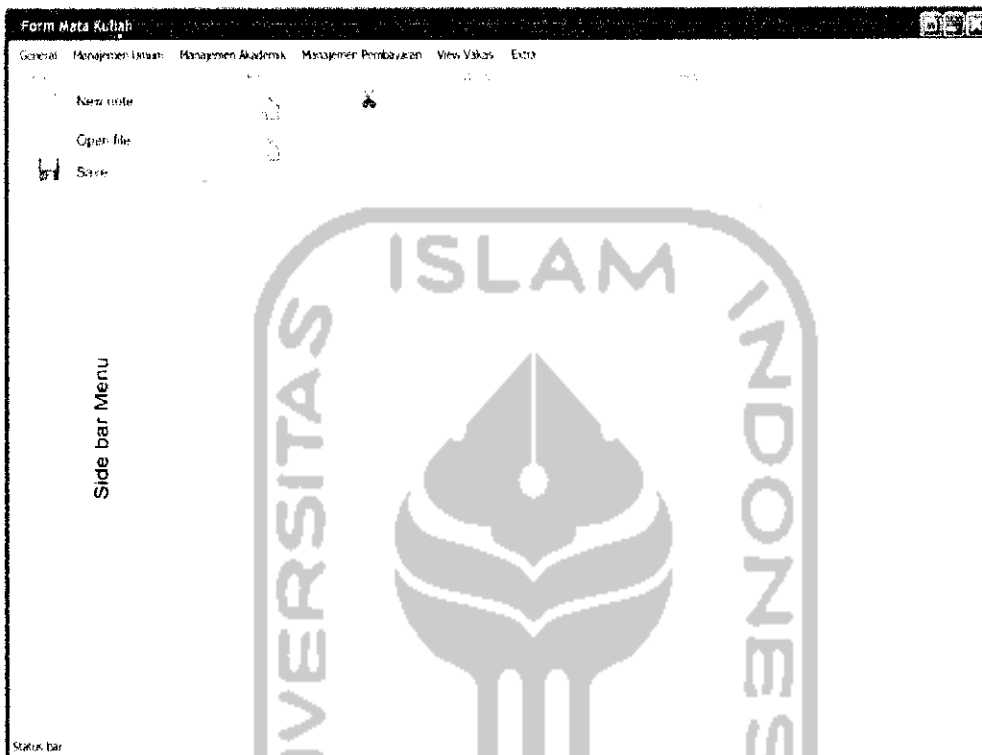
Halaman *login* merupakan sebuah pintu gerbang untuk masuk kedalam sistem sebagai *user* yang memiliki hak akses masing-masing seperti: DAU, DAP, DAK, jurusan, fakultas dan admin. Rancangan antarmuka untuk halaman *login* dapat dilihat pada gambar. *Form* jabatan akan secara otomatis terisi berdasarkan *username* yang diketikkan 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* terdapat 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

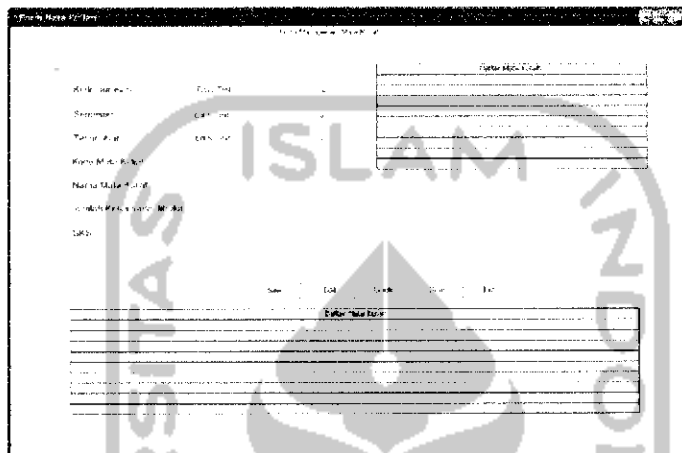
Pada halaman ini *user* yang mendapat hak akses Divisi Administrasi Perkuliahan & akademik (DAP) dapat menambahkan, mengubah atau menghapus data jurusan yang ada di FTI dengan menekan tombol *save*, *edit* atau *delete* yang berada dalam halaman ini. Rancangan antarmuka halaman manajemen jurusan dapat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

Gambar 4-21 Halaman Manajemen 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 detail 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 detail 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.

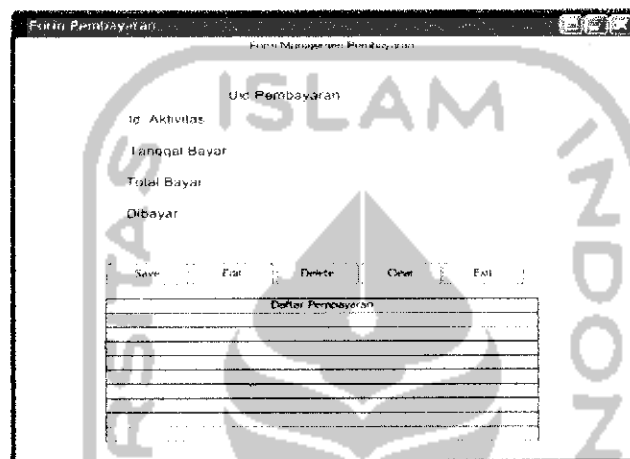
11. Halaman Tambah dan Edit Surat

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.

The screenshot shows a web application window titled "Uppm Mata Kuliah" with a subtitle "Halaman Manajemen Aktivitas". The main content area includes a "Nama:" label, a table with columns "No", "Tgl", "Deskripsi", "Rencana", and "Tgl", and a section titled "Aktivitas yang Selesai" with a table below it.

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.

The screenshot shows a web application window titled "Uppm Mata Kuliah" with a subtitle "Halaman Tambah dan Edit Data Aktivitas". The main content area is a form with various input fields and labels, including "Nama", "Kategori", "Kategori 1", "Kategori 2", "Kategori 3", "Kategori 4", "Kategori 5", "Kategori 6", "Kategori 7", "Kategori 8", "Kategori 9", "Kategori 10", "Kategori 11", "Kategori 12", "Kategori 13", "Kategori 14", "Kategori 15", "Kategori 16", "Kategori 17", "Kategori 18", "Kategori 19", "Kategori 20", "Kategori 21", "Kategori 22", "Kategori 23", "Kategori 24", "Kategori 25", "Kategori 26", "Kategori 27", "Kategori 28", "Kategori 29", "Kategori 30", "Kategori 31", "Kategori 32", "Kategori 33", "Kategori 34", "Kategori 35", "Kategori 36", "Kategori 37", "Kategori 38", "Kategori 39", "Kategori 40", "Kategori 41", "Kategori 42", "Kategori 43", "Kategori 44", "Kategori 45", "Kategori 46", "Kategori 47", "Kategori 48", "Kategori 49", "Kategori 50", "Kategori 51", "Kategori 52", "Kategori 53", "Kategori 54", "Kategori 55", "Kategori 56", "Kategori 57", "Kategori 58", "Kategori 59", "Kategori 60", "Kategori 61", "Kategori 62", "Kategori 63", "Kategori 64", "Kategori 65", "Kategori 66", "Kategori 67", "Kategori 68", "Kategori 69", "Kategori 70", "Kategori 71", "Kategori 72", "Kategori 73", "Kategori 74", "Kategori 75", "Kategori 76", "Kategori 77", "Kategori 78", "Kategori 79", "Kategori 80", "Kategori 81", "Kategori 82", "Kategori 83", "Kategori 84", "Kategori 85", "Kategori 86", "Kategori 87", "Kategori 88", "Kategori 89", "Kategori 90", "Kategori 91", "Kategori 92", "Kategori 93", "Kategori 94", "Kategori 95", "Kategori 96", "Kategori 97", "Kategori 98", "Kategori 99", "Kategori 100". At the bottom, there are buttons for "New", "Edit", and "Print".

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*

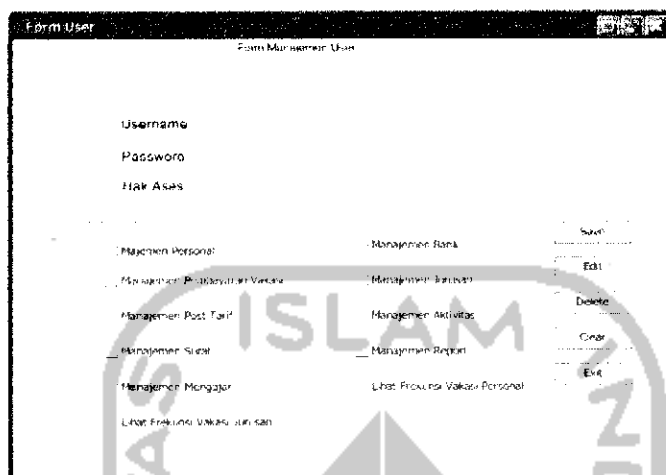
Pada halaman ini *user* yang mendapat hak akses sebagai administrator dapat menambahkan, mengubah atau menghapus data *user* dengan hak akses masing-masing bagian sesuai dengan kebutuhannya dengan menekan tombol *save*, edit atau *delete* yang berada dalam halaman ini. Rancangan antarmuka halaman manajemen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

17. Halaman Manajemen *Report*

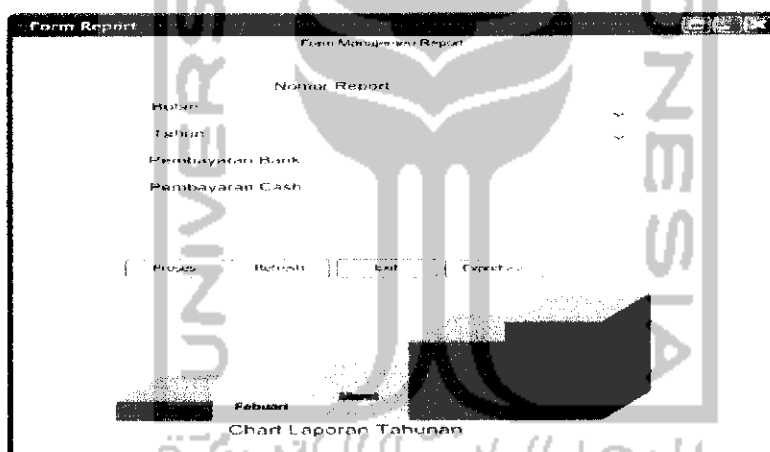
Pada halaman manajemen *report* dapat dibagi menjadi 2 kategori, yaitu:

- 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*

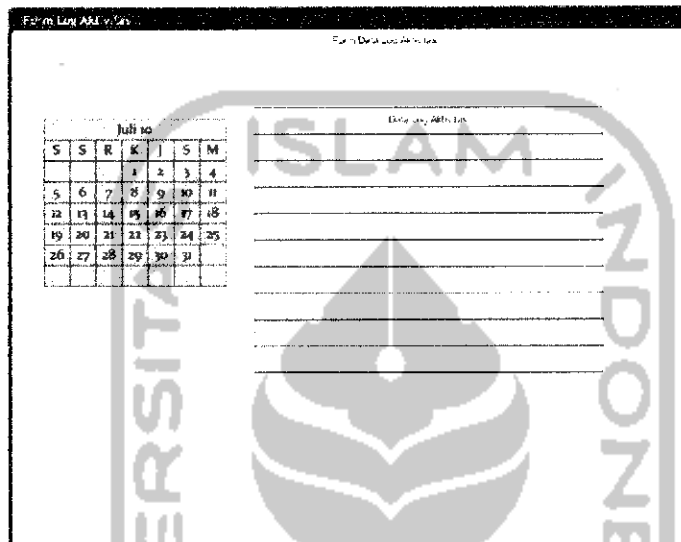


Gambar 4-33 Halaman Manajemen Report.

- 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.

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

BAB 5

IMPLEMENTASI PERANGKAT LUNAK SIPeV

Pada bab ini akan dijelaskan tentang implementasi dari perancangan perangkat lunak Sistem Informasi Pembayaran Vakasi (SIPeV). Bab ini meliputi implementasi secara umum, batasan implementasi dalam aplikasi SIPeV baik *hardware* yang digunakan untuk menjalankan aplikasi SIPeV ini maupun *software* yang mendukung berjalannya aplikasi SIPeV ini.

5.1 Implementasi Secara Umum

Implementasi adalah mempresentasikan hasil desain/ rancangan kedalam pemrograman. Implementasi merupakan kelanjutan dari tahap perencanaan yang telah dilakukan sebelumnya, sehingga implementasi dapat dikatakan sebagai pembuktian atas proses-proses perencanaan dan tidak menyimpang dari proses perencanaan tersebut.

5.2 Batasan Implementasi

Dalam batasan implementasi ini dapat ditinjau dari dua aspek yaitu:

1. Aspek perangkat keras (*hardware*)

Untuk mengimplementasikan aplikasi SIPeV ini dibutuhkan perangkat keras (*hardware*):

- Minimum *Personal Computer* (PC) P4 2,4 Mhz,
- RAM Memory 512 Gb.
- Hard disk space 40 Gb.
- Printer minimum 11 Ppm.
- Monitor minimum EGA/VGA.
- Mouse, *Key board*

2. Aspek perangkat lunak (*software*)

Aplikasi SIPeV tidak dapat diimplementasikan tanpa perangkat lunak pendukung:

- MySQL *minimum* version 5.0.

Media penyimpanan data yang digunakan dalam aplikasi SIPeV.

- Windows *eXperience Premium (XP) Home Edition*.

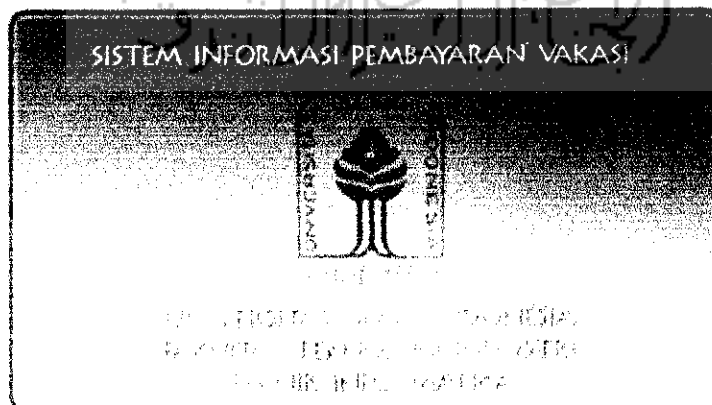
Sebuah sistem operasi yang mendasari berjalanya semua aplikasi termasuk aplikasi SIPeV.

5.3 Implementasi Antar Muka

Dalam implementasi antar muka akan dijelaskan mengenai perbagian dari sistem aplikasi SIPeV, baik dari halaman pertama yaitu halaman *splash* (halaman pertama kali dijalankan) sampai dengan halaman terakhir yaitu halaman *report*.

5.3.1 Halaman *Splash Screen*

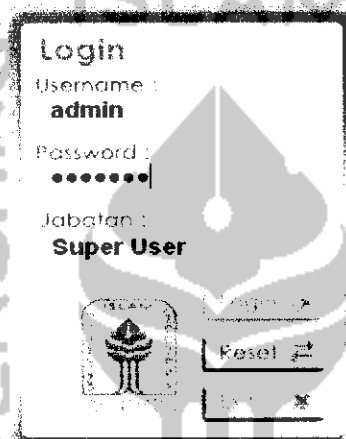
Halaman ini adalah halaman pertama yang ditampilkan, dalam halaman ini akan dieksekusi sebelum halaman *login* dimunculkan, halaman ini merupakan halaman yang bertugas mengecek aspek-aspek pendukung aplikasi SIPeV terpenuhi atau tidak. Tampilan halaman *splash screen* dapat dilihat pada Gambar 5-1.



Gambar 5-1 Halaman *Splash Screen*

5.3.2 Halaman *Login*

Halaman ini merupakan pintu masuk untuk membuka menu utama, dalam halaman *login* pengguna atau pegawai diminta memasukkan *username* (namayang sudah didaftarkan untuk menggunakan aplikasi SIPeV), dan *password* (adalah kata kunci untuk *username* yang dimasukkan). Jabatan akan secara otomatis muncul sesuai dengan data yang ada di dalam database berdasarkan *username* yang dimasukkan. Tampilan halaman *login* dapat dilihat pada Gambar 5-2.



Gambar 5-2 Halaman *Login*.

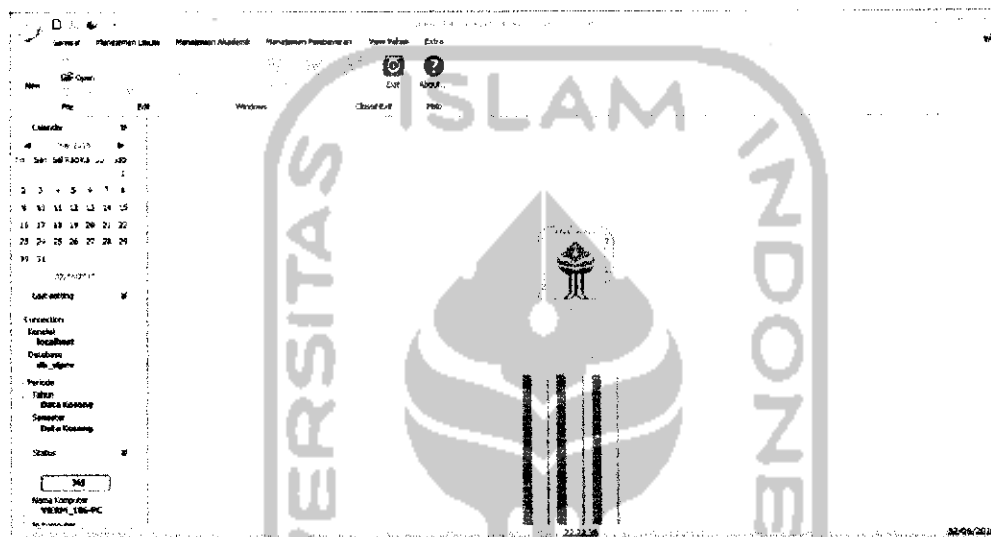
5.3.2.1 Halaman *Mainmenu* (Menu Utama).

Halaman ini adalah halaman utama dari sistem aplikasi SIPeV, dan mempunyai *group* menu general, manajemen umum, manajemen akademik, manajemen pembayaran, *view* vakasi, extra. Halaman ini dapat diakses pengguna apabila sudah melakukan proses *login*. *User* tersebut dapat mengakses menu-menu yang berada di halaman *mainmenu* sesuai dengan hak akses pada waktu *login*. Tampilan halaman *Mainmenu* dapat dilihat pada Gambar 5-3.

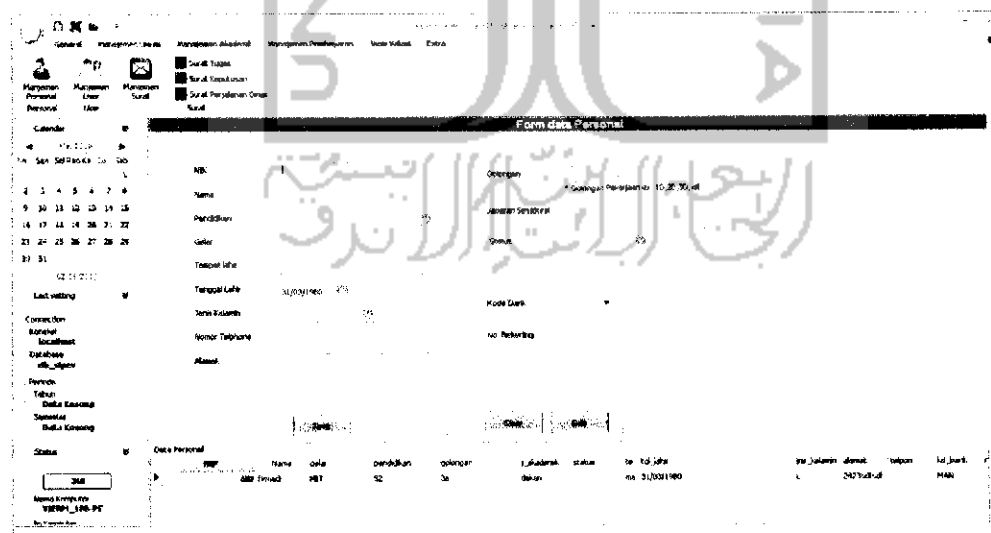
5.3.3 Halaman Manajemen Personal

Halaman manajemen personal merupakan sebuah halaman yang digunakan untuk menambah data personal, dan merupakan hak akses dari Divisi Administrasi Umum (DAU). Tidak hanya proses penambahan yang berada dalam

halaman ini tetapi terdapat proses lainnya, yaitu proses pengubahan data personal dan penghapusan data personal. Halaman ini menjadi dasar dari semua proses penting karena halaman ini digunakan untuk memasukkan personal yang akan menjalankan aktivitas berserta vakasi yang didapat. Halaman ini disertai *data grid* untuk memudahkan perubahan pada data apabila terjadi kesalahan. Tampilan halaman manajemen personal dapat dilihat pada Gambar 5-4.

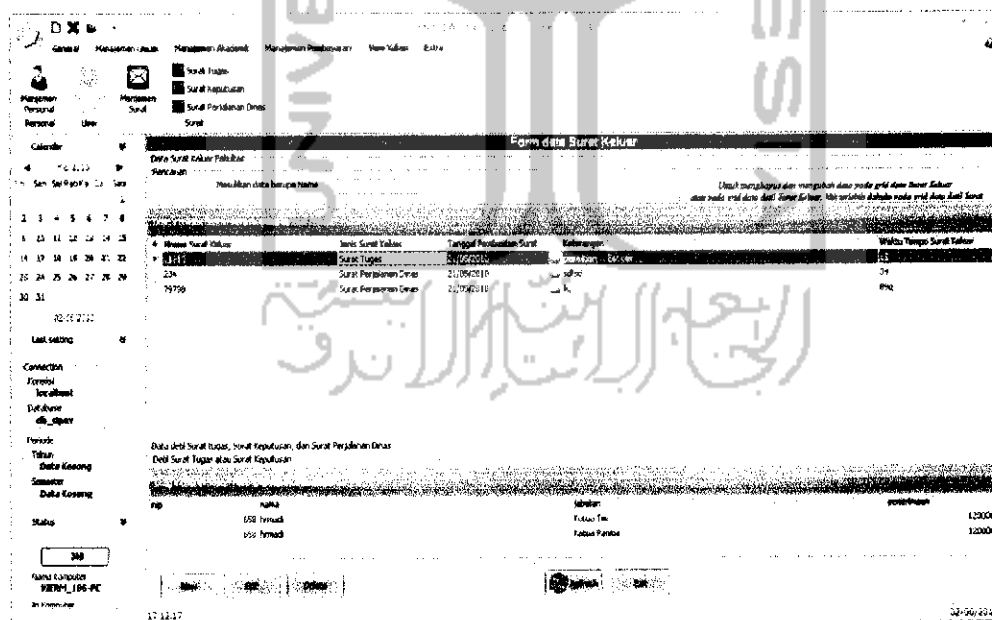


Gambar 5-3 Halaman *Mainmenu*

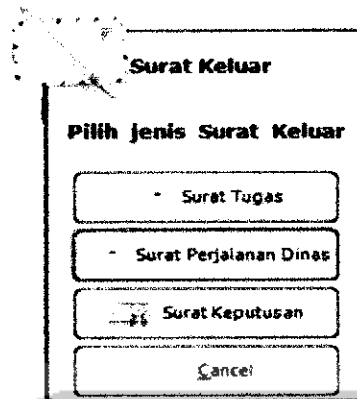


Gambar 5-4 Halaman Manajemen Personal.

master surat dan *detil* data surat beserta keterangan-keterangannya yang berupa *data grid* *detil* surat digunakan sebagai *detil* dari *master* surat, sehingga setiap data dalam *data grid* surat dipilih maka akan tampil *detil* mengenai surat tersebut. *User* dapat melakukan proses pencarian terhadap data surat, pencarian tersebut berdasarkan kepada kata yang dicari, sehingga setiap data yang mengandung data tersebut dalam *data grid* personal akan ditampilkan semua. Tampilan halaman manajemen surat dapat dilihat pada Gambar 5-6. Untuk menambah data surat dapat menekan tombol *new*, sehingga muncul menu atau kategori surat yang ingin dibuat, terdapat 3 menu kategori surat yaitu surat keputusan, surat tugas dan surat perjalanan dinas yang akan ditunjukkan pada Gambar 5-7. Untuk menghapus atau mengubah data surat terlebih dahulu memilih data surat pada *data grid* surat kemudian menekan tombol *delete* untuk menghapus data surat atau tombol edit untuk mengubah data surat, apabila menekan tombol edit yang berarti akan mengubah data surat maka sistem akan secara otomatis menampilkan halaman edit data surat berdasarkan kategori surat.



Gambar 5-6 Halaman Manajemen Surat.



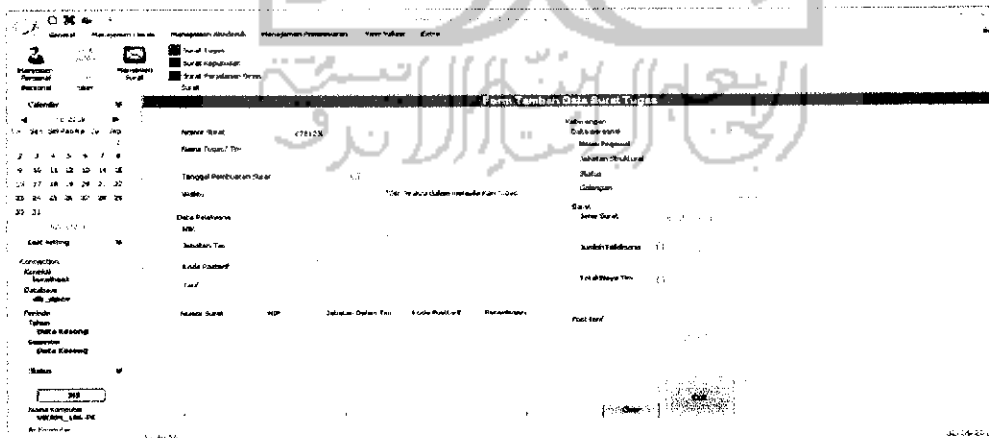
Gambar 5-7 Menu Kategori Surat.

5.3.6 Halaman Tambah Data Surat

Sesuai dengan menu kategori surat, terdapat 3 halaman tambah data surat yaitu :

1. Halaman tambah data Surat Tugas (ST).

Halaman tambah data surat tugas digunakan untuk menambah data surat tugas. Penambahan data pelaksana secara otomatis menambahkan aktivitas atas surat tugas berdasarkan pelaksana dalam hal ini personal disertai biaya penerimaan. Tampilan halaman tambah data surat tugas dapat dilihat pada Gambar 5-8.



Gambar 5-8 Halaman Tambah Data Surat Tugas.

2. Halaman tambah data Surat Keputusan (SK).

Halaman tambah data surat keputusan secara garis besar sama dengan *form* tambah data tugas dan digunakan untuk menambah data surat keputusan, yang membedakan hanya kategori atau jenis suratnya dan pertimbangan (*considerat*) yang melandasinya. Penambahan data pelaksana secara otomatis menambahkan aktivitas atas surat tugas berdasarkan pelaksana dalam hal ini personal disertai biaya penerimaan. Tampilan halaman tambah data surat tugas dapat dilihat pada Gambar 5-9.

Gambar 5-9 Halaman Tambah Data Surat Keputusan.

3. Halaman tambah data Surat Perjalanan Dinas (SPJ)

Halaman tambah data surat perjalanan dinas (SPJ) digunakan untuk menambah data SPJ, dan memiliki *form* yang lebih lengkap daripada ST dan SK, karena memiliki relasi dengan data personal, data akomodasi dan data detail surat SPJ. Sistem tidak membatasi jumlah penambahan data akomodasi dalam SPJ, dalam halaman tambah data SPJ terdapat 2 proses yaitu proses di aplikasi untuk masukkan datanya dan proses di dalam database untuk melakukan proses penambahan datanya. Hal itu dilakukan untuk meringankan kinerja aplikasi karena pada halaman ini memiliki banyak relasi data dalam proses penambahan data. Tampilan halaman tambah data SPJ dapat dilihat pada Gambar 5-10.

Gambar 5-10 Halaman Tambah Data Surat Perjalanan Dinas.

5.3.7 Halaman Ubah Data Surat

Halaman ubah data surat terbagi menjadi 2 kategori :

1. Halaman ubah data SK dan ST

Halaman ini digunakan untuk mengubah data surat khususnya data SK dan ST. Halaman ubah data SK dan ST menjadi satu halaman, sebab data antara kedua surat tersebut mempunyai kesamaan, yang membedakan hanya jenis surat yang telah ditentukan pada waktu proses penambahan data dan tersimpan di dalam database, sehingga untuk pengubahan data antara SK dan ST dapat dibuat menjadi satu halaman. Tampilan halaman untuk mengubah data SK dan ST dapat dilihat pada Gambar 5-11.

2. Halaman ubah data SPJ

Halaman ubah data SPJ memiliki *form* yang lebih lengkap dari pada halaman SK ataupun SPJ. Hal tersebut mengacu pada jumlah data saat penambahan data SPJ, dikarenakan untuk data SPJ memiliki relasi data beberapa relasi data, yaitu data personal, data surat dan data detil surat SPJ, seperti yang telah dijelaskan pada bagian Halaman tambah data Surat Perjalanan Dinas (SPJ). Tampilan Halaman SPJ dapat dilihat pada Gambar 5-12.

Gambar 5-11 Halaman Ubah Data ST dan SK

Gambar 5-12 Halaman Ubah Data SPJ.

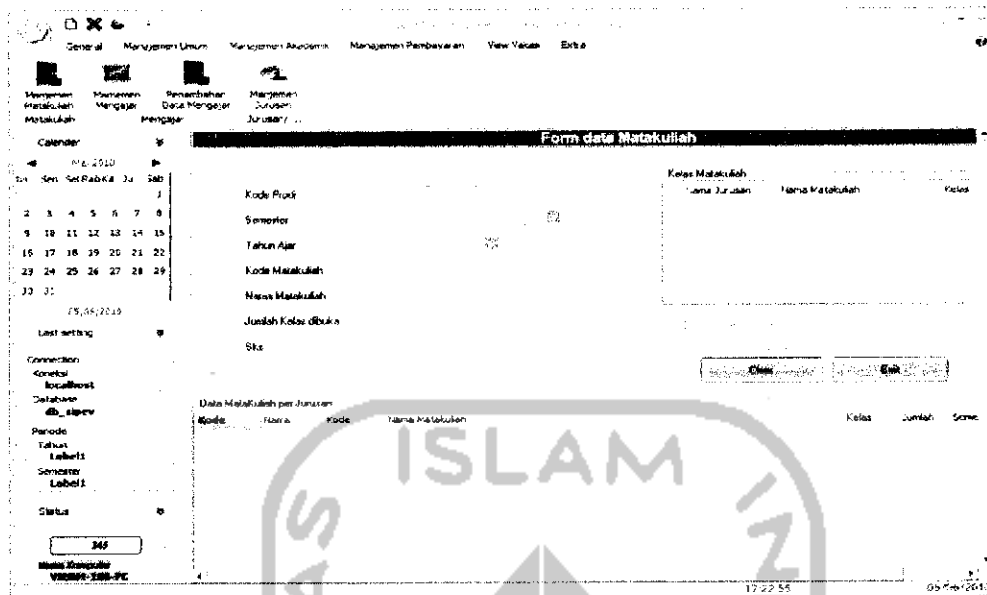
5.3.8 Halaman Manajemen Matakuliah

Halaman mata kuliah merupakan sebuah *form* yang digunakan untuk menambah data matakuliah, dan merupakan hak akses dari Divisi Administrasi Perkuliahan (DAP), tidak hanya proses penambahan yang berada dalam *form* ini tetapi terdapat proses lainnya, yaitu proses pengubahan data mata kuliah dan penghapusan data mata kuliah, dalam proses penambahan data matakuliah terdapat 2 tahapan proses yaitu proses di aplikasi untuk memasukkan datanya dan

proses di dalam *database* untuk melakukan proses penambahan datanya. Hal tersebut dilakukan karena dalam proses penambahan data matakuliah dapat melakukan penambahan data yang berulang dengan kelas yang berbeda, untuk *pseudocode*-nya dapat dilihat pada Gambar 5-13, sedangkan untuk megubah dan menghapus data dapat dilakukan dengan cara memilih terlebih dahulu data di dalam *datagrid* kemudian tekan tombol edit untuk mengubah data atau *delete* untuk menghapus data. Tampilan halaman manajemen matakuliah dapat dilihat pada Gambar 5-14.

<p><u>Procedure:</u></p> <pre>proc_save_menMatakuliah(input kd_matkul,kd_jur,jml_kelas,sks,semester,tahun:integer; nm_matakuliah:string)</pre>
<p><u>Kamus:</u></p> <pre>jmlh_kelas1:integer</pre>
<p><u>Algoritma:</u></p> <pre>jml_kelas1 ← select count(kelas) from v_matakuliah where kd_matakuliah ← kd_matkul and kd_jurusan ← kd_jur If (jml_kelas1 = 0) then jmlh_kelas1 ← 0 jmlh_kelas1 ← jmlh_kelas1 + jmlh_kelas While (jmlh_kelas1 < jmlh_kelas) do Insert into v_matakuliah values (kd_matakul, char(jml_kelas1+65)) jmlh_kelas1 ← jmlh_kelas1 + 1 Insert into v_detil_matkul values (kd_matakul,kd_jur, nm_matakuliah, sks,semester,tahun)</pre>

Gambar 5-13 Pseudocode Save Matakuliah.



Gambar 5-14 Halaman Manajemen Matakuliah.

5.3.9 Halaman Tambah, Edit Data Mengajar

Halaman ini merupakan kelanjutan dari halaman manajemen mengajar dan digunakan untuk memasukkan data tambah, ubah mengajar. Perubahan data yang sudah dipilih sebelumnya pada Halaman Manajemen Mengajar, kemudian dengan menekan tombol edit maka akan menampilkan halaman ini untuk melakukan perubahan data, atau apabila *user* menekan tombol *new* maka akan muncul halaman ini untuk melakukan penambahan data seperti yang ditunjukkan pada Gambar 5-16.

5.3.10 Halaman Manajemen Mengajar

Halaman manajemen mengajar merupakan sebuah halaman yang digunakan untuk mengolah data mengajar, dan merupakan hak akses dari Divisi Administrasi Perkuliahan (DAK). Halaman ini mempunyai persamaan konsep *master detail* dengan halaman manajemen surat. Halaman ini menggunakan 1 *master* yaitu *data grid* personal dan 2 *detail* yaitu *data grid* matakuliah dan *data grid* kelas. *Data grid* personal menampilkan detail personal, *data grid* matakuliah menampilkan *detail* data matakuliah yang diajar oleh personal, sedangkan untuk *data*

grid kelas menampilkan kelas yang diajar personal berdasarkan pada matakuliahnya. Halaman ini terdapat 3 proses yaitu penambahan (*new*), pengubahan (*edit*), dan penghapusan data (*delete*), untuk menambah data tekan tombol *new* maka akan muncul halaman penambahan data mengajar, sedangkan untuk mengubah atau menghapus data terlebih dahulu memilih data matakuliah di *data grid* matakuliah dan data kelas di *data grid* kelas, kemudian tekan tombol edit untuk mengubah dan tombol *delete* untuk menghapus data. Aktivitas mengajar akan secara otomatis menambah data di tabel *v_aktivitas* dan menambah vakasi ke NIP tersebut. Tampilah halaman manajemen mengajar dapat dilihat pada Gambar 5-15.

Nama Matakuliah	Kode Matakuliah	Nama Kelas	GK	Semester	Tahun	Kode	Nama Matakuliah	No. Matakuliah
TEORI INFORMATIKA	22110	Perencanaan Layanan Ilmu	2	2	2010	1	TEORI INFORMATIKA	22110
TEORI INFORMATIKA	80110	Perencanaan Layanan Ilmu	2	2	2010	2	TEORI INFORMATIKA	80110

Gambar 5-15 Halaman Manajemen Mengajar.

5.3.11 Halaman Tambah, Edit Data Mengajar

Halaman ini merupakan kelanjutan dari halaman manajemen mengajar dan digunakan untuk memasukkan data tambah, ubah mengajar. Pengubahan data yang sudah dipilih sebelumnya pada Halaman Manajemen Mengajar, kemudian dengan menekan tombol edit maka akan menampilkan halaman ini untuk melakukan pengubahan data, atau apabila *user* menekan tombol *new* maka akan muncul halaman ini untuk melakukan penambahan data seperti yang ditunjukkan pada Gambar 5-16.

Kode Jurusan	Nama Jurusan	Kode Mata Kuliah	Nama Mata Kuliah	Jumlah SKS
S22	TEKNIK INFORMATIKA	SI20450	Perencanaan Layanan Web	A
S22	TEKNIK INFORMATIKA	SI20451	Perencanaan Layanan Web	B
S22	TEKNIK INFORMATIKA	SI20410	Perencanaan Layanan Web	C
S22	TEKNIK INFORMATIKA	SI20440	Perencanaan non prosedural	A
S22	TEKNIK INFORMATIKA	SI20440	Perencanaan non prosedural	B
S22	TEKNIK INFORMATIKA	SI20440	Perencanaan non prosedural	A

Gambar 5-16 Halaman Tambah, Edit Data Mengajar.

5.3.12 Halaman Manajemen Jurusan

Halaman ini digunakan untuk mengolah data jurusan, baik menambah data yang belum ada, mengubah data atau menghapus data yang tidak terpakai. Dalam halaman ini apabila data kode jurusan yang dimasukkan sudah ada maka sistem akan secara otomatis menampilkan data yang sudah ada, dan *user* hanya bisa melakukan proses mengubah data (*edit*) atau menghapus data (*delete*). Tampilan halaman manajemen jurusan dapat dilihat Gambar 5-17.

Kode Jurusan	Nama Jurusan	Nama Jurusan
S22	TEKNIK INFORMATIKA	S22 TEKNIK INFORMATIKA

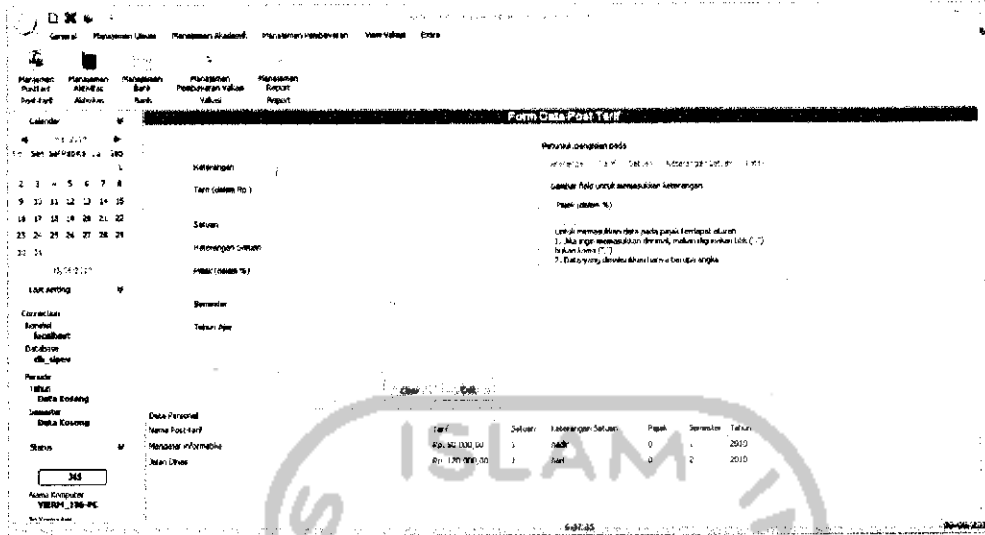
Gambar 5-17 Halaman Manajemen Jurusan

5.3.13 Halaman Manajemen Post-tarif

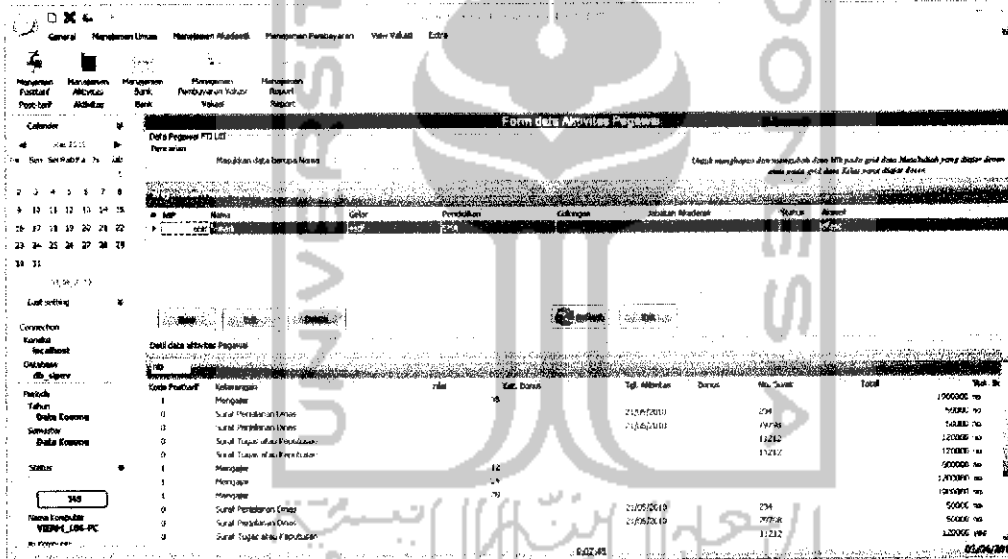
Halaman ini digunakan untuk menambah data post-tarif, dan merupakan hak akses dari Divisi Administrasi Keuangan (DAK). Tidak hanya proses penambahan yang berada dalam halaman ini tetapi terdapat proses lainnya, yaitu proses pengubahan data dan penghapusan data post-tarif. Halaman ini bertujuan memberi harga kepada masing-masing aktivitas. Proses halaman manajemen post-tarif memiliki proses yang sama dengan halaman yang lain, yaitu menambah data (*new*), mengubah data (*edit*), dan menghapus data (*delete*). Tampilan halaman manajemen post-tarif dapat dilihat pada Gambar 5-18.

5.3.14 Halaman Manajemen Aktivitas

Halaman ini digunakan untuk menambah data aktivitas, dan merupakan hak akses dari masing-masing divisi yang diatur oleh administrator aplikasi SIPeV untuk dapat melakukan proses mengolah data aktivitas, dikarenakan setiap divisi dapat mengolah data aktivitas sesuai dengan agenda yang berada dalam lingkungan Fakultas Teknologi Industri (FTI) dan dan dalam waktu yang belum ditentukan. Halaman ini mempunyai persamaan konsep *master detil* dengan halaman manajemen yang lain. Halaman ini menggunakan 1 *master* yaitu *data grid* personal dan 1 *detil* yaitu *data grid* aktivitas. *Data grid* personal menampilkan detil personal dan *data grid* aktivitas menampilkan *detil* aktivitas yang dilakukan oleh setiap personal, sedangkan untuk *data grid* kelas menampilkan kelas yang diajar personal berdasarkan pada matakuliahnya. Tidak hanya proses penambahan yang berada dalam halaman ini tetapi terdapat proses lainnya, yaitu proses pengubahan data dan penghapusan data post-tarif. Halaman ini bertujuan untuk mengolah data aktivitas apabila personal/ pegawai melakukan aktivitas di luar dari mengajar atau aktivitas yang pada surat keluar. Halaman ini memiliki 3 proses seperti halaman yang lain. Tampilan halaman manajemen aktivitas dapat dilihat pada Gambar 5-19.



Gambar 5-18 Halaman Manajemen Post-tarif



Gambar 5-19 Halaman Manajemen Aktivitas

5.3.15 Halaman Tambah, Edit Data Aktivitas

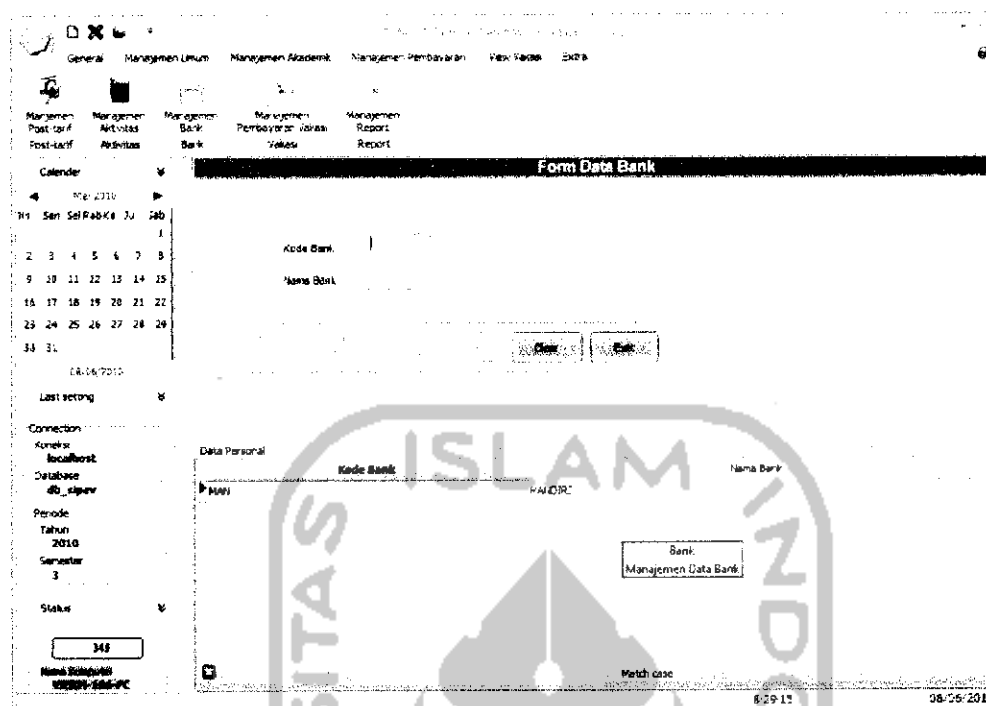
Halaman ini merupakan kelanjutan dari halaman manajemen aktivitas dan digunakan untuk memasukkan data atau mengubah data aktivitas yang telah dipilih sebelumnya pada Gambar 5-20. Dalam halaman ini terdapat fasilitas hitung berupa kalkulator, sehingga memudahkan *user* dalam melakukan perhitungan.

Data yang sudah dipilih di dalam Halaman Manajemen Aktivitas, jika *user* kemudian menekan tombol edit atau apabila *user* menekan tombol *new* maka akan muncul halaman seperti yang ditunjukkan pada Gambar 5-20.

Gambar 5-20 Halaman Tambah, Edit Data Aktivitas

5.3.16 Halaman Manajemen Bank

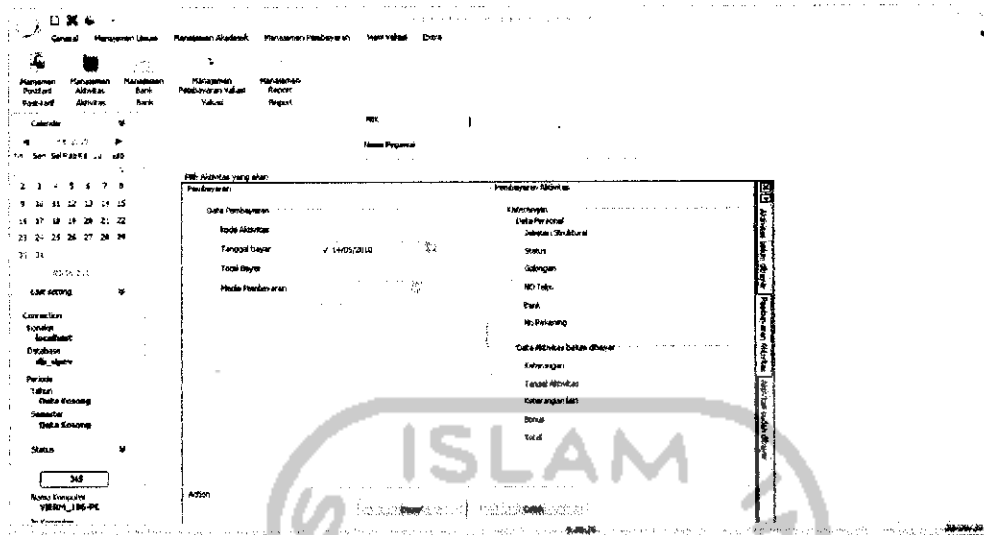
Halaman ini digunakan untuk mengolah data bank yang dimiliki personal, dimana bank ini akan digunakan untuk pembayaran vakasi secara transfer kepada dosen, dan pegawai. Halaman ini merupakan hak akses dari Divisi Administrasi Keuangan (DAK), pada halaman dapat melakukan penambahan data bank, tetapi tidak hanya proses penambahan yang berada dalam halaman ini, terdapat pula proses lainnya, yaitu proses pengubahan data dan penghapusan data bank. Tampilan halaman manajemen aktivitas dapat dilihat pada Gambar 5-21 Halaman Manajemen Bank Gambar 5-21.



Gambar 5-21 Halaman Manajemen Bank

5.3.17 Halaman Manajemen Pembayaran Vakasi

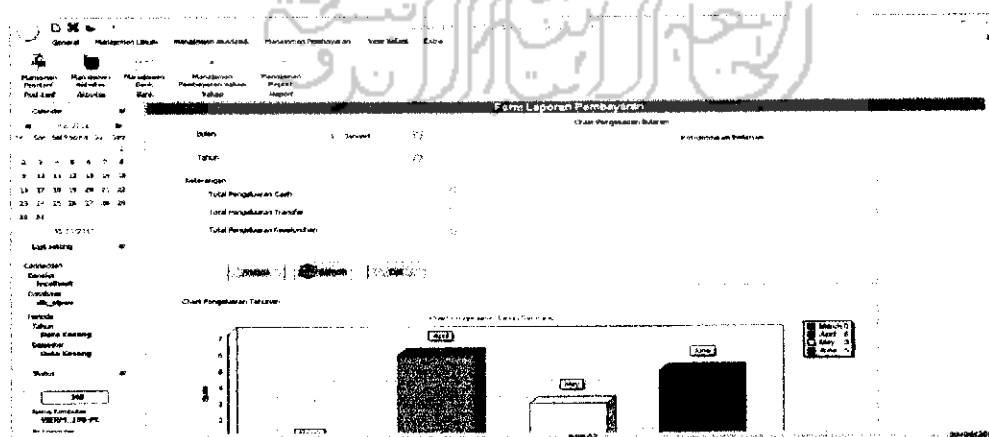
Halaman ini digunakan untuk proses pembayaran atas aktivitas yang telah dilakukan. Proses pembayaran nantinya akan mempengaruhi pada pelaporan (report). Halaman manajemen pembayaran vakasi ini merupakan sebuah halaman yang digunakan untuk menambah data pembayaran, dan merupakan hak akses dari Divisi Administrasi Keuangan (DAK). Tidak hanya proses penambahan yang berada dalam halaman ini tetapi terdapat proses lainnya, yaitu proses pengubahan data personal dan penghapusan data pembayaran. Pada proses ini terdapat proses cetak yang digunakan untuk mendokumentasikan hasil pembayaran, selain itu dalam halaman manajemen pembayaran vakasi juga terdapat 3 proses yaitu tambah, ubah, delete data. Konsep kerja dalam halaman ini adalah menampilkan semua aktivitas yang belum dibayarkan berdasarkan nip. Tampilan halaman manajemen pembayaran vakasi dapat dilihat pada Gambar 5-22.



Gambar 5-22 Halaman Manajemen Pembayaran Vakasi

5.3.18 Halaman Manajemen Report (Laporan Pembayaran)

Halaman ini untuk melihat vakasi yang keluar baik dalam 1 tahun atau bisa ditampilkan secara perbulan. Vakasi akan ditampilkan dalam bentuk diagram batang untuk *report* 1 tahun dan diagram pie (*pie chart*) untuk *report* perbulannya. *Report* perbulan akan menampilkan perhitungan total vakasi yang dibayarkan ke bank dan yang dibayarkan tunai dalam bulan tersebut. Tampilan halaman manajemen report (laporan pembayaran) dapat dilihat pada Gambar 5-23.



Gambar 5-23 Halaman Manajemen report

5.3.19 Halaman Cetak *Report* Bulanan Personal

Halaman ini digunakan untuk mencetak, dan mengekspor data kedalam format pdf, excel, atau format lain yang diinginkan berdasarkan aktivitas bulan tertentu dan vakasi yang didapat oleh masing-masing personal. Tampilan halaman cetak *report* bulanan personal dapat dilihat pada Gambar 5-24.

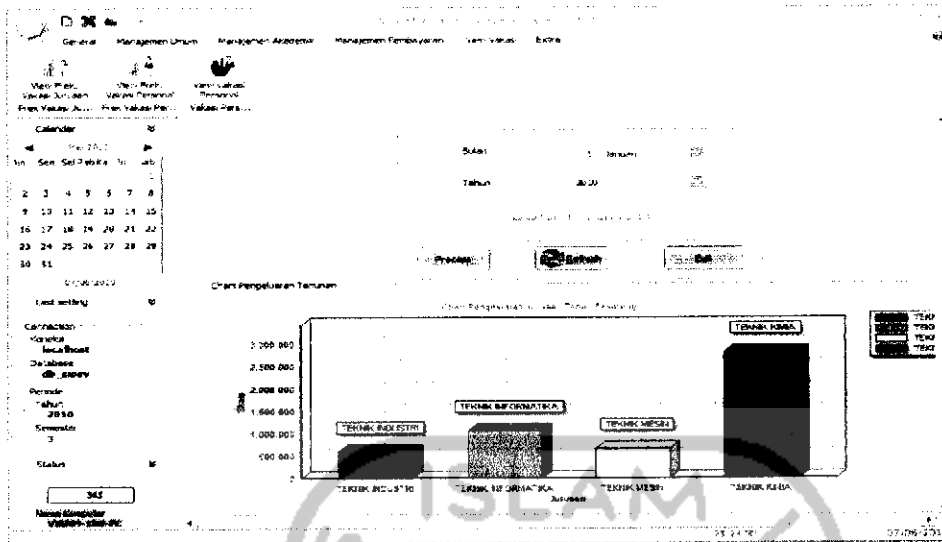
The screenshot displays a web application interface for printing a personal monthly report. The main content area is titled 'Forma Cetak Report Bulanan Personal'. It features a navigation menu on the left with options like 'Manajemen Umum', 'Manajemen Akademik', and 'Manajemen Perencanaan'. Below the menu is a calendar for the month of June 2010. The main area contains a form for selecting the month and year, and a table of personal data. The table has the following columns: Bulan, Tahun, Id, Kode Akt. vikasi, Nama, Taruhan, Total, Media, Dibayar, no, name, and no_rek. The data rows are as follows:

Bulan	Tahun	Id	Kode Akt. vikasi	Nama	Taruhan	Total	Media	Dibayar	no	name	no_rek
6	2010	1	50	Mengajar	2700000	1411200	manusfe	525	RIGDI		
6	2010	1	50	Mengajar	1700000	1411200	caaf	525	RISDI		
6	2010	6	49	Sekret	1800000	1200000	caaf	152	METDI		
6	2010	7	52	Ichtylogipid	1800000	600000	caaf	521	DIADI		
6	2010	8	53	gihadi	0700000	1050000	caaf	252	VAWI		
6	2010	9	54	Mengajar	0700000	530000	caaf	524	TITDI		

Gambar 5-24 Halaman Cetak Report Bulanan Personal

5.3.20 Halaman *View* (Lihat) Frekwensi Vakasi Jurusan

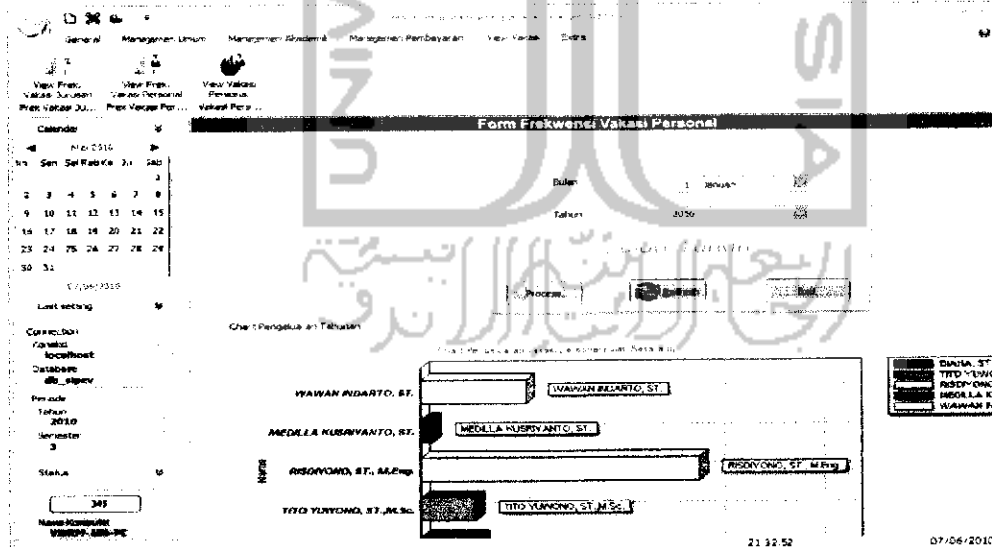
Halaman *view* (lihat) frekwensi vakasi jurusan merupakan salah satu hasil akhir atau tujuan dibuatnya SIPEV dari 3 halaman yang lain yaitu Halaman *View* (Lihat) Frekwensi Vakasi Personal dan Halaman *View* (Lihat) Vakasi Personal. Halaman ini digunakan untuk melihat frekwensi vakasi yang telah dibayarkan berdasarkan jurusan masing-masing. Tampilan halaman *view* (lihat) frekwensi vakasi jurusan dapat dilihat pada Gambar 5-25.



Gambar 5-25 Halaman *View* (Lihat) Frekwensi Vakasi Jurusan.

5.3.21 Halaman *View* (Lihat) Frekwensi Vakasi Personal

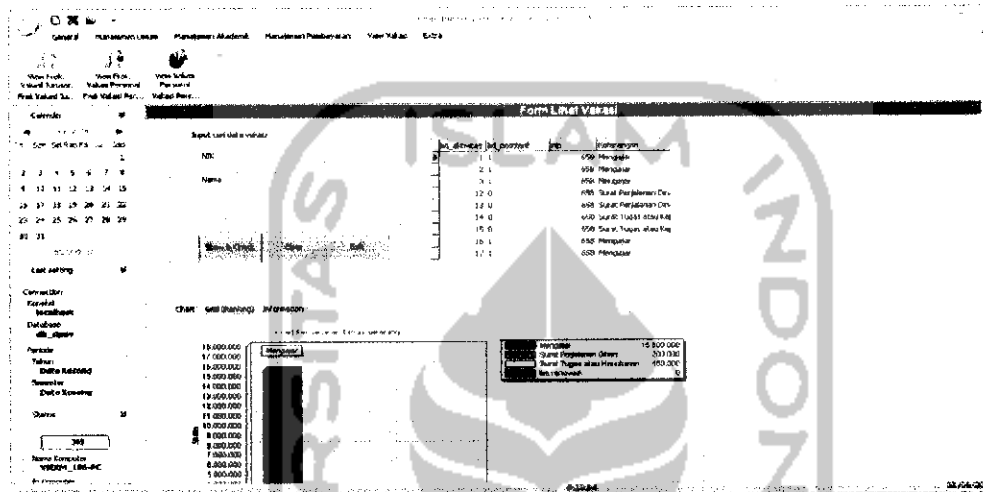
Halaman ini digunakan untuk menampilkan frekwensi vakasi masing-masing personal. Pada bulan dan tahun yang di ingin ditampilkan oleh *user*. Tampilan halaman ini dapat dilihat pada Gambar 5-26.



Gambar 5-26 Halaman *View* (Lihat) Frekwensi Vakasi Personal

5.3.22 Halaman *View* (Lihat) Vakasi Personal

Manajemen *view* (lihat) vakasi personal digunakan untuk mengetahui aktivitas yang dilakukan oleh nip tertentu, aktivitas ditampilkan dalam 2 *form*, yang pertama dalam diagram batang yang kedua dalam data grid. Tampilan halaman *view* (lihat) vakasi personal dapat dilihat pada Gambar 5-27.



Gambar 5-27 Halaman *View* (Lihat) Vakasi Personal

5.3.23 Halaman Manajemen Profil

Halaman manajemen personal merupakan sebuah halaman yang digunakan untuk menambah data profil instansi dalam hal ini adalah FTI, dan merupakan hak akses dari Divisi Administrasi Umum (DAU). Tidak hanya proses penambahan yang berada dalam halaman ini tetapi terdapat proses lainnya, yaitu proses perubahan data dan penghapusan data profil. Data tersebut nantinya digunakan untuk proses mencetak, atau digunakan untuk pengisian alamat dalam SPJ. Tampilan halaman manajemen profil dapat dilihat pada Gambar 5-28.

Form Manajemen Profil Instansi

Nama Instansi

Telephone

Faximile

Kota

Alamat Instansi

Nomor Telephone dan Facs akan disipkan pada alamat setelah karakter "

Batal Ok

Kops

UNIVERSITAS ISLAM INDONESIA

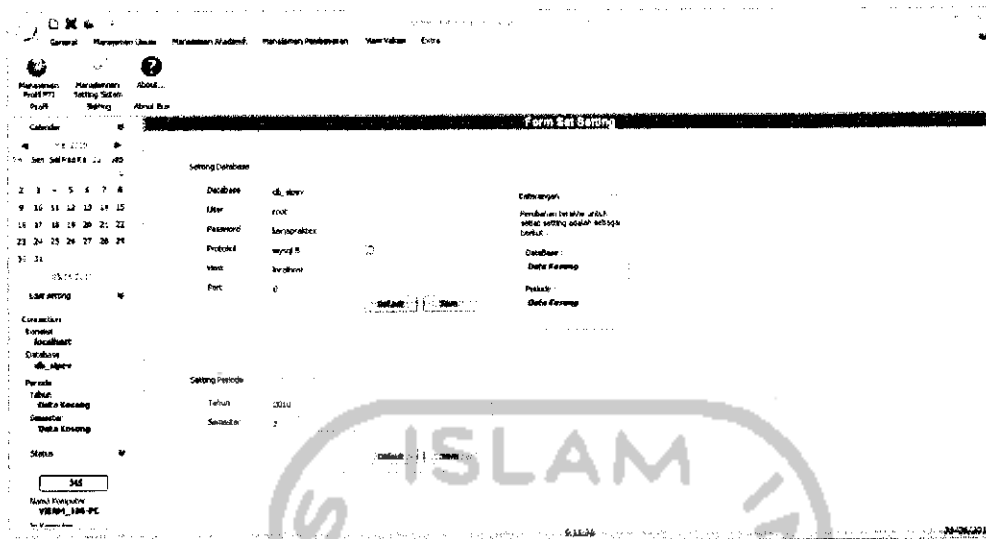
FAKULTAS TEKNOLOGI INDUSTRI

Jalan Kalurung KM 14,4 Telp.(0274) 89287, 895007 Facs.(0274) 895007 Ext. 148;
Kotak Pos 75 Sleman 55901

Gambar 5-28 Halaman Manajemen Profil.

5.3.24 Halaman Set Setting

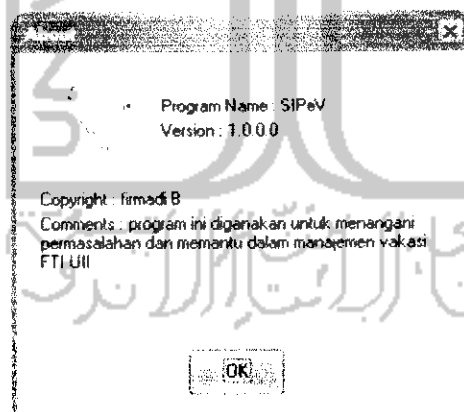
Halaman manajemen personal merupakan sebuah halaman yang digunakan untuk menambah data *setting*, dan merupakan hak akses dari Administrator. Tidak hanya proses penambahan yang berada dalam halaman ini tetapi terdapat proses lainnya, yaitu proses pengubahan data dan penghapusan data *setting* yang berjalan di aplikasi SIPeV. Halaman ini bertujuan untuk mengatur *setting* koneksi database, serta *setting* waktu di SIPeV. Dalam hal ini nilai *setting* tidak disimpan ke dalam database melainkan disimpan dalam bentuk *filetext* yang berekstensi *.spv*. Tampilan halaman *set setting* dapat dilihat pada Gambar 5-29.



Gambar 5-29 Halaman *Set Setting*

5.3.25 Halaman *About*

Halaman ini berfungsi sebagai halaman informasi mengenai aplikasi SIPeV, dan tujuan dibangunnya aplikasi SIPeV. Halaman *about* akan ditampilkan seperti Gambar 5-30.



Gambar 5-30 Halaman *About*

5.3.26 Halaman *Pop Up*

Halaman ini digunakan untuk mengisikan data ke dalam *form* dengan mudah. Terdapat 3 halaman *pop up* yaitu:

1. Halaman *pop up* personal

Untuk memudahkan pengisian *form* berdasarkan NIP maka terdapat fasilitas untuk mencari data NIP dan kemudian meletakkan data tersebut kedalam *form* yang membutuhkan data tersebut. Caranya adalah dengan cara menekan tombol yang mempunyai icon seperti Gambar 5-31. Setelah menekan tombol yang mempunyai icon tersebut kemudian akan muncul *form pop up* pencarian data personal seperti Gambar 5-32.

NIK

Gambar 5-31 Form Untuk Memunculkan *Pop up* Personal

nip	nama	polar	pendidikan	golongan	t_akademik	status	tempat_lahir	tg_lahir	ins_bekasmin	alamat
656	hnmadi	asdf	SMA				asdf	31/03/1980	P	d'esd
657	asdf		SMA					31/03/1980		

Gambar 5-32 Halaman *Pop up* Data Personal

2. Halaman *pop up* matakuliah

Untuk memudahkan pengisian *form* berdasarkan kode matakuliah maka terdapat fasilitas untuk mencari data kode matakuliah dan kemudian meletakkan data tersebut kedalam *form* yang membutuhkan data tersebut.

Caranya adalah dengan cara menekan tombol yang mempunyai icon seperti Gambar 5-33. Setelah menekan tombol yang mempunyai icon tersebut kemudian akan muncul *form pop up* pencarian data matakuliah Gambar 5-34.

Kode Matakuliah

Gambar 5-33 Form Untuk Memunculkan *Pop up* Matakuliah

Kode Jurusan	Nama	Kode	Nama	Kelas	SKS	nm Jurusan	semester	tahun
523410	523	Ferrogram	C	2	523523410	TEKNIK	2	2010
523440	523	Ferrogram	A	2	523523440	TEKNIK	2	2010
523440	523	Ferrogram	B	2	523523440	TEKNIK	2	2010
523910	523	Optimasi Db	B	2	523523910	TEKNIK	2	2010
523910	523	Optimasi Db	C	2	523523910	TEKNIK	2	2010

Gambar 5-34 Halaman *Pop up* Data Matakuliah

3. Halaman *pop up* post-tarif

Untuk memudahkan pengisian *form* berdasarkan kode post-tarif maka terdapat fasilitas untuk mencari data kode post-tarif dan kemudian meletakkan data tersebut kedalam *form* yang membutuhkan data tersebut. Caranya adalah dengan cara menekan tombol yang mempunyai icon seperti Gambar 5-35. Setelah menekan tombol yang mempunyai *icon* tersebut kemudian akan muncul *form pop up* pencarian data personal Gambar 5-36.

Kode Postarif

Gambar 5-35 Form Untuk Memunculkan *Pop up* Data Post-tarif

Form Pencarian Data Postarif

Pencarian Data Postarif

Filter

Nama

id_postarif	keterangan	tarif	satuan	ket_sacuan	parak	semester	tahun
1	Menzisir	50000		1 losok		0	1 2010
2	Jalan Dinas	100000		1 hari		0	2 2010

selected Kode Postarif 1 Tarif 50000

Gambar 5-36 Halaman *Pop up* Data Post-tarif

5.3.27 Halaman *Log* aktivitas

Halaman ini berfungsi untuk memberikan informasi aktivitas *user* terhadap sistem. Aktivitas ini dicatat untuk membedakan pengolahan data antar *user* dan dapat memberikan informasi *user* yang telah melakukan pengolahan data di halaman post-tarif. Tampilan halaman *log* aktivitas dapat dilihat pada Gambar 5-37.

Form data log Aktivitas

id_log	id_postarif	nama	waktu	keterangan
1	1	Menzisir	2010-06-03 10:00:00	Menzisir Post-Tarif
2	2	Jalan Dinas	2010-06-03 10:00:00	Jalan Dinas Post-Tarif

Gambar 5-37 Halaman *log* aktivitas

BAB 6

ANALISIS KINERJA PERANGKAT LUNAK

Pada bab ini akan dijelaskan tentang analisis kinerja perangkat lunak. Menganalisa kinerja aplikasi SIPeV ini menggunakan cara pengujian normal dan tidak normal terhadap aplikasi SIPeV.

6.1 Batasan Pengujian

Pengujian adalah suatu proses yang digunakan untuk menjamin bahwa perangkat lunak yang dihasilkan dapat berjalan sesuai dengan keinginan dan dapat dilihat seberapa besar kemampuan aplikasi dalam menangani kesalahan (*error*). Pengujian ini dilakukan untuk menemukan kesalahan – kesalahan yang mungkin terjadi dikarenakan kesalahan penulisan sintaks, fungsi maupun implementasi lainnya. Pengujian dilakukan dalam dua tahapan, yaitu tahapan pengujian normal dan tahapan pengujian tidak normal. Pada tahapan pengujian normal dilakukan pengujian dengan procedure sesuai dengan ketentuan dan data yang dibutuhkan, sedangkan pada tahapan pengujian tidak normal dilakukan pengujian dengan menjalankan program yang tidak sesuai dengan ketentuan dan data yang dibutuhkan.

6.2 Pengujian Normal

Pengujian normal dilakukan dengan menjalankan perangkat lunak sesuai dengan ketentuan dan data yang dibutuhkan sehingga dapat dilihat apakah aplikasi berjalan dengan baik sesuai dengan tujuan yang diinginkan.

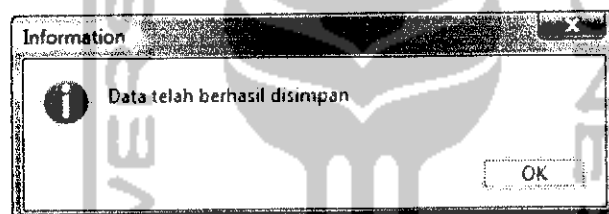
6.2.1 Proses Login

Pengujian normal untuk proses login adalah jika *username* dan *password* sesuai dengan data yang berada di dalam *database*, maka langsung membuka *form* utama (*main form*). Dengan langsung membuka *main form* dan tidak

menampilkan pesan dahulu jika berhasil login, maka akan mempercepat waktu tunggu untuk membuka *main form*.

6.2.2 Proses Penambahan data

Pengujian normal untuk proses penambahan data dilakukan dengan mengisikan semua input yang berada dalam *form* manajemen telah diisi. Pesan yang muncul apabila semua pengisian data telah sesuai dengan prosedur, sama dengan pesan yang akan ditampilkan jika proses penambahan berhasil dilakukan, yang akan ditampilkan pada Gambar 6-1. Pesan ini terdapat dalam proses penambahan data perbankan, proses penambahan data post-tarif, proses penambahan data aktivitas, proses penambahan data jurusan, proses penambahan data mengajar, proses penambahan data matakuliah.

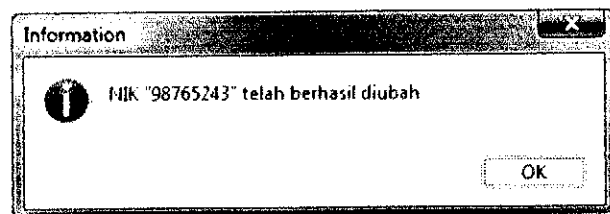


Gambar 6-1 Pesan Berhasil Menambah Data Personal

6.2.3 Proses Pengubahan Data

Pengujian normal untuk proses pengubahan data adalah dengan mengubah data yang telah dipilih untuk diubah dan menekan tombol *save*. Pesan yang ditunjukkan untuk proses pengubahan data apabila berhasil, sama dengan pesan yang ditunjukkan pada proses penambahan yang lain hanya saja untuk atribut atau parameter keterangan berubah, untuk Form yang berhubungan dengan data personal akan ditampilkan NIP, untuk data bank ditampilkan kode bank, untuk data yang berhubungan dengan data matakuliah ditampilkan kode matakuliah. Untuk pesan pengujian proses pengubahan data yang berhasil akan ditampilkan pada Gambar 6-2. Pesan ini terdapat dalam proses pengubahan data perbankan, proses pengubahan data post-tarif, proses pengubahan data aktivitas, proses

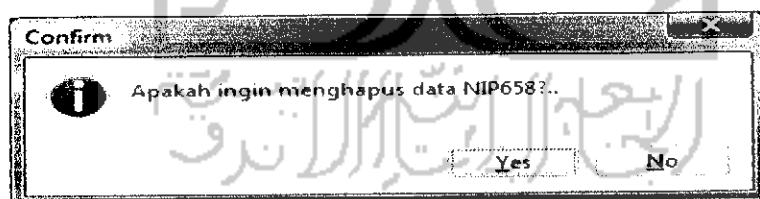
pengubahan data jurusan, proses pengubahan data mengajar, proses pengubahan data matakuliah.



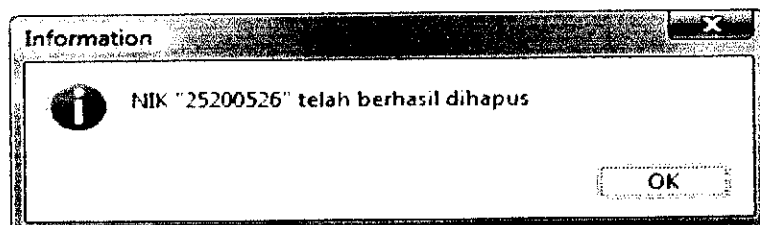
Gambar 6-2 Pesan Berhasil Mengubah Data Personal.

6.2.4 Proses Penghapusan Data

Pengujian yang dilakukan adalah dengan memilih data dan menekan tombol hapus. Pada proses ini akan muncul pesan *dialog* “apakah anda ingin menghapus data..?” jika *user* akan menghapus data yang salah atau tidak diperlukan. Tampilan *dialog* tersebut dapat dilihat pada Gambar 6-3, jika *user* memilih “yes” maka data akan dihapus dari *data base* dan muncul pesan seperti Gambar 6-4. Pesan ini terdapat dalam proses penghapusan data bank, proses penghapusan data post-tarif, proses penghapusan data aktivitas, proses penghapusan data jurusan, proses penghapusan data mengajar dan proses penghapusan data matakuliah.



Gambar 6-3 Pesan Dialog Penghapusan Data.



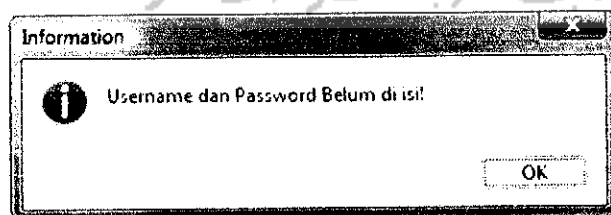
Gambar 6-4 Pesan Dialog Data Telah Berhasil Dihapus.

6.3 Pengujian Tidak Normal

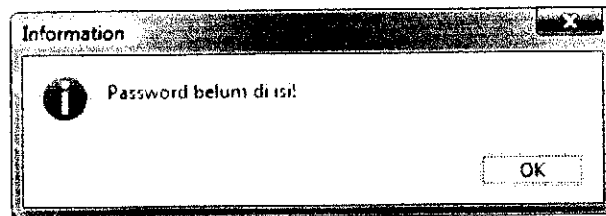
Pengujian tidak normal dilakukan dengan menjalankan perangkat lunak dengan data yang tidak sesuai dengan ketentuan dan data yang dibutuhkan sehingga akan muncul pesan – pesan kesalahan yang memberitahukan kepada pengguna agar menjalankan sistem sesuai dengan ketentuan yang ada dan data yang dibutuhkan.

6.3.1 Penanganan Kesalahan Proses *Login*

Pengujian dilakukan dengan tidak mengisi semua data yang wajib diisi maka, akan muncul pesan kesalahan seperti Gambar 6-5. Gambar 6-6 menampilkan pesan kesalahan apabila *user* belum memasukkan atau mengisikan *password*, tetapi terdapat pulan penanganan kesalahan apabila *user* belum mengisikan *username* untuk proses *login* maka akan tampil pesan kesalahan seperti pada Gambar 6-7, sedangkan data yang diisikan tidak sesuai dengan format yang telah ditentukan maka, akan muncul pesan – pesan kesalahan seperti Gambar 6-8 apabila *username* yang dimasukkan tidak terdapat dalam *database* atau dengan kata lain *username* belum terdaftar sebagai pengguna aplikasi SIPeV. *Username* yang diketikan tidak mempunyai hak akses maka akan muncul kalimat “*user* tidak mempunyai hak akses!” seperti gambar Gambar 6-9, dan apabila *password* tidak sesuai dengan *username* yang dimasukkan akan muncul pesan *error* seperti pada Gambar 6-10.



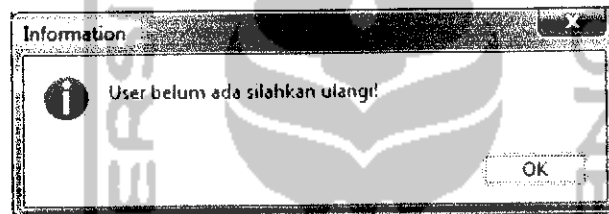
Gambar 6-5 Pesan Kesalahan Jika *Username* dan *Password* Belum Diisi



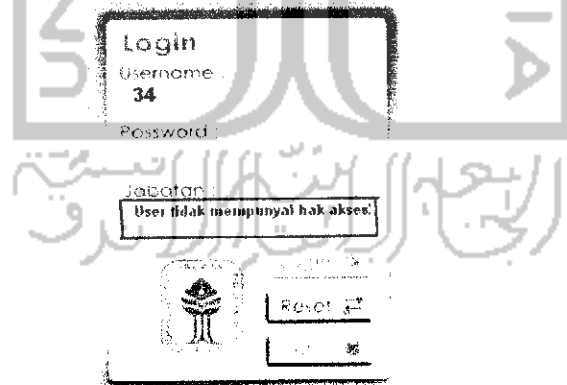
Gambar 6-6 Pesan Kesalahan Jika *Password* Belum Diisi



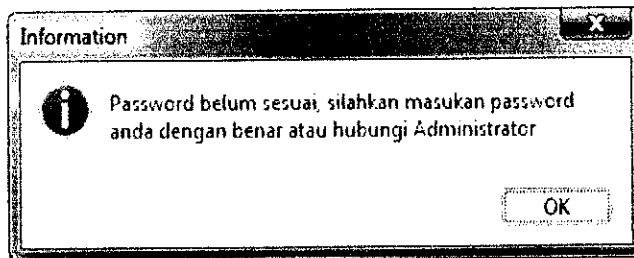
Gambar 6-7 Kesalahan Jika *Username* Belum Diisi



Gambar 6-8 Kesalahan Jika *Username* Tidak terdapat dalam *Database*.



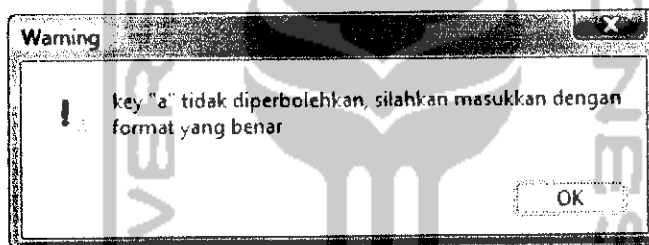
Gambar 6-9 Kesalahan Jika *Username* Tidak Mempunyai Hak Akses.



Gambar 6-10 Kesalahan Jika *Username* tidak sesuai dengan password.

6.3.2 Penanganan Kesalahan Pengetikan Pada *Form*.

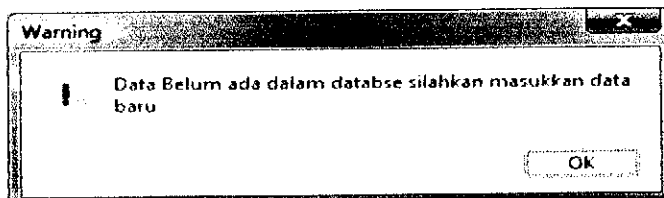
Pengujian dilakukan dengan meng-*input*-kan atau mengetikan pada *form* yang secara logika mempunyai aturan yang berlawanan, sebagai contoh data yang diketikan harus angka dan simbol yang diijinkan, tetapi *form* diisi dengan huruf, maka akan tampil pesan *error* seperti Gambar 6-11.



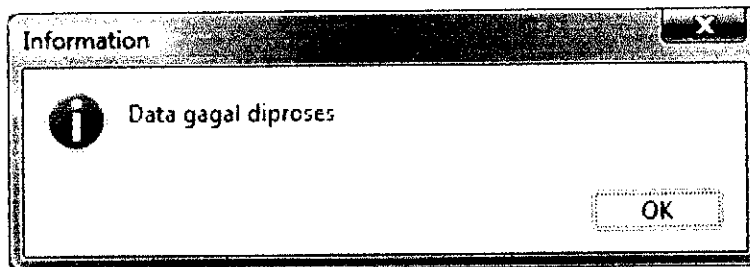
Gambar 6-11 Kesalahan pengetikan data pada *form*

6.3.3 Penanganan Kesalahan Proses Penambahan Data

Pengujian dilakukan dengan memasukkan data yang belum ada di dalam *database*, apabila data belum ada di dalam *database* maka akan muncul pesan *error* seperti Gambar 6-12. Pesan yang muncul jika data gagal dimasukkan kedalam *database* dapat dilihat pada Gambar 6-13.



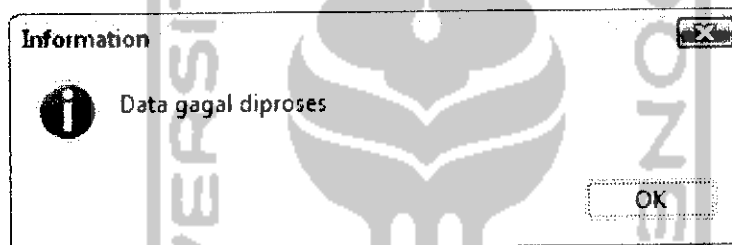
Gambar 6-12 Pesan Data Baru.



Gambar 6-13 Proses Data Gagal Dimasukkan Kedalam *Database*.

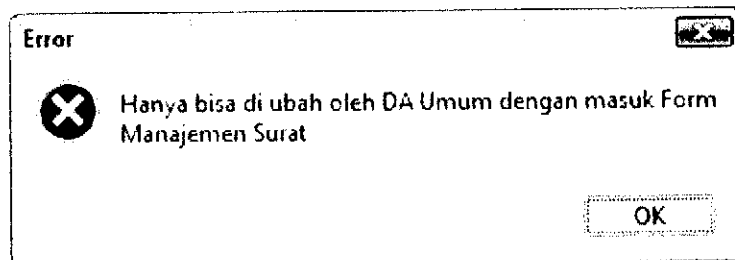
6.3.4 Penanganan Kesalahan Proses Perubahan Data

Pengujian dilakukan dengan tidak memasukkan data pada *form* atau proses perubahan data gagal dilakukan, maka akan tampil pesan kesalahan seperti pada Gambar 6-14.



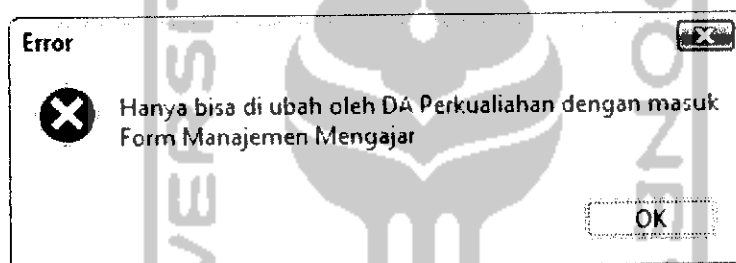
Gambar 6-14 Proses Data Gagal Diubah.

Khusus untuk manajemen aktivitas yang merupakan halaman dengan sekumpulan data mengajar, data surat keluar dan data aktivitas yang lain atau aktivitas yang diinputkan melalui halaman aktivitas. Perubahan data yang dilakukan harus sesuai procedure perubahan yaitu untuk data aktivitas surat hanya dapat diubah dengan masuk ke halaman manajemen surat sedangkan data aktivitas mengajar hanya dapat diubah dengan masuk ke halaman manajemen mengajar. Penanganan kesalahan yang terjadi ketika *user* akan mengubah data surat melalui halaman manajemen aktivitas yang seharusnya perubahan data surat hanya dapat dilakukan melalui halaman manajemen surat. Pesan kesalahan yang muncul ketika hal tersebut dilakukan dapat dilihat pada Gambar 6-15.



Gambar 6-15 Penanganan Kesalahan Ubah Data Surat

Penanganan kesalahan yang terjadi ketika *user* akan mengubah data mengajar melalui halaman manajemen aktivitas yang seharusnya perubahan data mengajar hanya dapat dilakukan melalui halaman manajemen mengajar. Pesan kesalahan yang muncul ketika hal tersebut dilakukan dapat dilihat pada Gambar 6-16.



Gambar 6-16 Penanganan Kesalahan Ubah Data Mengajar

6.3.5 Penanganan Kesalahan Pada Proses Pencarian

Pengujian dilakukan dengan mengetikkan kata pada *form* nama, apabila tidak terdapat data yang mengandung kata tersebut maka pada *data grid* akan muncul pesan "<no data to display>" seperti gambar Gambar 6-17.

6.3.6 Penanganan Kesalahan Proses Pelaporan (*Reporting*)

Penanganan kesalahan terjadi dalam halaman manajemen *report* saat *user* memilih data berdasarkan bulan dan tahun, apabila data tersebut belum ada di dalam *database* maka akan muncul pesan *error* seperti Gambar 6-18.

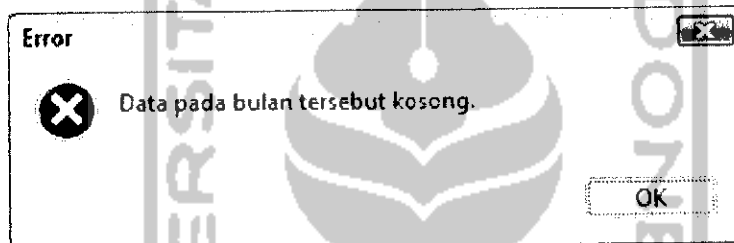
Form data Mengajar

Data Dosen Pengajar

Pencarian: Masukkan data berupa Nama Untuk menampilkan dan mengubah data HR pada grid akan Menampilkan yang dapat diolah atau pada grid akan Edit yang dapat diolah

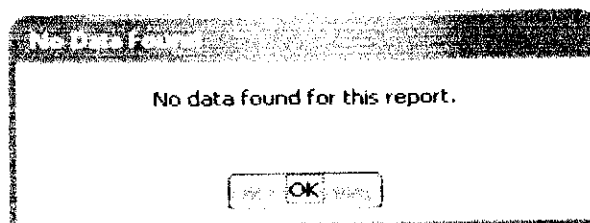
No	Nama	Gelar	Pendidik	Golongan	Jabatan Akademik	Status	Alamat

Gambar 6-17 Penangan Kesalahan Proses Pencarian.

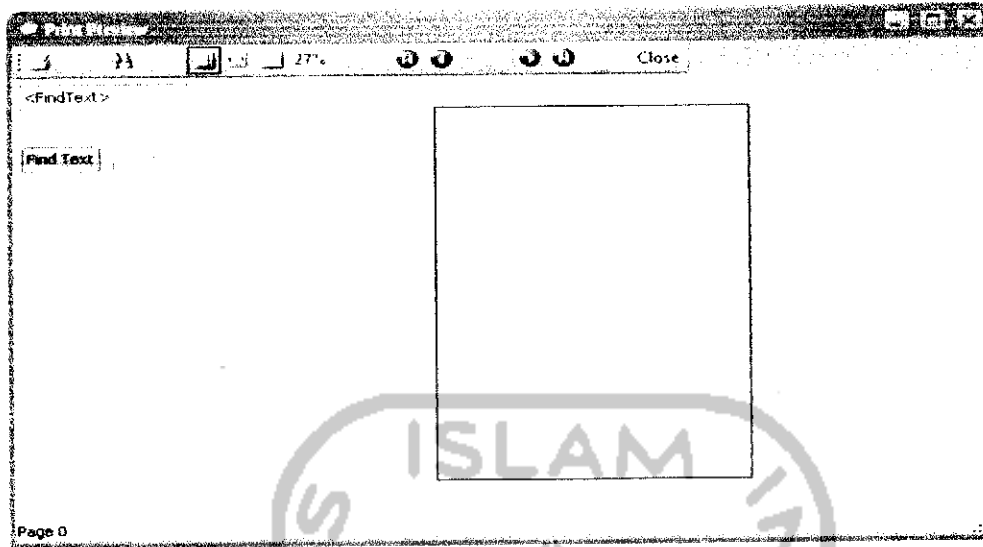
Gambar 6-18 Penanganan Kesalahan Data *Report* Kosong

6.3.7 Penanganan Kesalahan Pada cetak *Report*

Penanganan kesalahan terjadi apabila tidak terdapat data tetapi dicetak dengan kata lain *user* melakukan pencetakan data kosong. Data yang dipilih menurut bulan dan tahun ternyata kosong tetapi *user* melakukan proses pencetakan maka akan muncul pesan *error* seperti Gambar 6-19 dan halaman cetak akan menampilkan *form* kosong seperti pada Gambar 6-20.



Gambar 6-19 Penanganan Kesalahan Cetak Data Kosong



Gambar 6-20 Penanganan Kesalahan Cetak Data Dalam Bentuk *Form*.

6.4 Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan diatas dan pengujian terhadap level operasional yang meliputi DAU, DAK dan DAP serta pada level manajerial yang meliputi jurusan dan fakultas, maka dapat diambil hasil dari kinerja sistem informasi pembayaran vakasi ini menjadi hasil analisis yang telah didapat, yaitu:

1. Level operasional, dari sisi level operasioanl dengan sistem ini memudahkan pengintputan data dengan fasilitas-fasilitas dan mempercepat pengerjaan yang dulunya masih *konvensional* menjadi terkomputerisasi, selain itu dengan adanya proses penanganan kesalahan dapat meminimalisir kesalahan data di dalam *database*.
2. Level manajerial, dari sisi level manajerial dengan sistem ini dapat melihat frekwensi vakasi dan detil vakasi setiap personal dalam bentuk diagram sehingga memudahkan level operasional dalam pengambilan keputusan.

Berdasarkan hasil pengujian dapat diambil beberapa kelebihan dan kelemahan system sebagai berikut.

1. Kelebihan :

- a) Sistem ini menggunakan *optimasi database* sehingga pertukaran data lebih cepat, dan aplikasi dapat berjalan lebih ringan.
- b) Desain dari *halaman* dan *form* lebih *user friendly* sehingga memudahkan dalam pengoperasian sistem ini.
- c) Sistem ini dapat merubah dapat pembayaran kedalam beberapa file menurut kebutuhan antara lain : *excel, pdf, rich text file, dan html*.
- d) Informasi yang diberikan adalah dalam bentuk diagram sehingga nantinya dapat digunakan untuk referensi level manajerial dalam menentukan kebijakan-kebijakan.
- e) Informasi selalu berubah berdasarkan data vakasi maupun aktivitas yang diolah oleh level operasional.

2. Kekurangan :

- a) Sistem harus di-*restart* (ditutup dan dibuka lagi), jika terjadi kesalahan sistem yang tidak didefinisikan dalam penanganan, misalakan terjadi kerusakan koneksi jaringan yang mengakibatkan aplikasi ini *hung!* tidak berjalan sesuai dengan mestinya.
- b) Sedikit rumit penggunaan dikarenakan menu yang terlalu banyak.
- c) Sistem tidak dapat diakses oleh setiap personal.

BAB 7

SIMPULAN DAN SARAN

Pada bab ini akan disampaikan kesimpulan yang bisa diambil selama melakukan proses pengembangan perangkat lunak Sistem Informasi Pembayaran Vakasi (SIPeV). Bab ini juga terdapat beberapa saran yang dapat dipergunakan untuk pengembangan lebih lanjut yang lebih baik.

5.1 Simpulan

Berdasarkan hasil penelitian dan hasil pembahasan yang telah dilakukan, dapat disimpulkan bahwa:

1. SIPeV dapat membantu pengarsipan dan pertukaran data yang dulunya masih konvensional sekarang pertukaran dapat melewati sistem.
2. Menggunakan *optimasi database (view, trigger, stored procedure)* yang menjadikan pertukaran data lebih cepat, dan meringankan kerja aplikasi
3. Memberikan informasi vakasi baik per personal sampai per jurusan, sehingga dapat sistem ini dapat menjadi referensi pihak jurusan atau fakultas dalam mengambil kebijakan.
4. SIPeV dapat memberikan efektif dan efisiensi kerja khususnya pada level operasional

5.2 Saran

Mengingat berbagai keterbatasan yang dialami, maka penulis menyarankan untuk pengembangan SIPeV Fakultas Teknologi Industri Universitas Islam Indonesia (FTI UII) di masa mendatang sebagai berikut.

1. Penambahan proses penanganan kesalahan yang lebih mendalam sehingga semua bentuk kesalahan dapat teratasi dengan baik.
2. Penyederhanaan *form* tetapi dapat menampilkan informasi yang lengkap agar sistem tidak dianggap rumit.
3. Pembuatan sistem informasi lain untuk memberikan informasi vaksinasi personal sehingga vaksinasi setiap personal dapat diakses sesuai kepentingan personal masing-masing.



DAFTAR PUSTAKA

- Badriyah Tussy. 2007. *Database Trigger*. <http://lecturer.eepis-its.edu/~tessy/lecturenotes/db2/bab5.pdf>, (diakses pada tanggal 27 Februari 2010)
- Darmayuda, K. 2007. *Program aplikasi client/server: pengolahan data akademik dan sistem penjualan terpadu dengan visual basic 6.0 dan borland delphi 7.0*. Bandung: Informatika Bandung.
- Hermawan, J. 2004. *UML dan VB.NET*. Yogyakarta: Andi Offset.
- Howland, G. 1996. *Development of an Open and Flexible Payment System*. <http://www.systemics.com/docs/sox/overview.html>, (diakses pada tanggal 17 Februari 2010)
- Lanusse, A. 2009. *Reasons to Migrate from Delphi 7 to Delphi 2009*. <http://info.danysoft.com/free/razonesDelphi2009.pdf>, (diakses pada tanggal 17 Februari 2010).
- Mulyanto, A. 2009. *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta: Pustaka Pelajar.
- Munawar. 2005. *Pemodelan Visual dengan UML*. Yogyakarta: Graha Ilmu.
- Nugroho, B. 2005. *Database Relational dengan MySQL*. Yogyakarta: Andi Offset.
- Ramakrishnan, R., & Gehrke, J. 2003. *Database Management System – Third Edition*. USA: McGraw-Hill Companies
- Rucky, A.S. 2001. *Manajemen Penggajian & Pengupahan Untuk Karyawan Perusahaan*. Jakarta: Gramedia Pustaka Utama
- Simarmata, J., & Prayudi, I. 2006. *Basis Data*. Yogyakarta: Andi Offset.

Sirkel. 2008, *Modul Praktikum Rekayasa Perangkat Lunak 2008/2009*. Yogyakarta.

Witarto. 2004. *Memahami sistem Informasi*, Penerbit. Bandung: Informatika.





LAMPIRAN

الجامعة الإسلامية
الاندونيسية



Universitas Islam Indonesia
Jurusan Teknik Informatika FTI

No.

USULAN PENELITIAN TUGAS AKHIR

Nama : Firmadi Biastiko
Contact : 085227012569 / vierm_186@yahoo.com

No. Mhs : 06 523 186

A. JUDUL / TOPIK

SISTEM INFORMASI PEMBAYARAN VAKASI FAKULTAS TEKNOLOGI
INDUSTRI UNIVERSITAS ISLAM INDONESIA

B. MASALAH

1. Latar Belakang

Teknologi informasi memegang peranan penting dalam segala bidang kehidupan, kemampuan dalam memperoleh dan mendistribusikan informasi berkembang semakin cepat dan semakin luas. Perkembangan pengolahan data secara digital menggunakan komputer yang semakin meluas secara cepat menandai semakin cepatnya perkembangan teknologi informasi. Kemajuan teknologi informasi tidak bisa dipungkiri sudah merambah ke semua bidang kehidupan, tidak terkecuali bidang pendidikan. SIAT (Sistem Informasi Akutansi Terpadu) adalah salah satu contoh penerapan teknologi informasi dalam bidang pendidikan.

Sebagai lembaga pendidikan yang berkecimpung dalam bidang teknologi industri, tentu saja Fakultas Teknologi Industri UII tidak mau tertinggal dalam hal ini. Hanya saja, di tengah-tengah maraknya penggunaan teknologi informasi di bidang pendidikan ini, ternyata masih terdapat berbagai pekerjaan di fakultas ini yang dikerjakan belum optimal, yaitu manajemen pembayaran vakasi menggunakan SIAT (Sistem Informasi Akutansi Terpadu). Walaupun menggunakan Sistem Informasi, penggunaannya tidak berbeda jauh hanya sebatas sebagai pelaporan tanpa adanya kemampuan lebih dari monitoring pada level managerial.

Dari kelemahan SIAT tersebut tidak menutup kemungkinan berdampak kepada individu setiap staf yang terkait didalam lembaga pendidikan ini. Pemerataan pekerjaan dan kegiatan diluar tugas pokok yang berimbas pada vakasi yang didapat, sangat berpengaruh pada totalitas masing-masing personal untuk memajukan lembaga pendidikan ini khususnya pada fakultas teknologi industri. Sehingga tidak ada anggapan dari setiap staf bahwa dirinya tidak dianggap dan tidak diikuti dalam pekerjaan padahal mempunyai kompeten diri di bidang tersebut.

Sistem informasi pembayaran vakasi salah merupakan satu hal yang penting. Untuk level managerial dapat digunakan sebagai tolak ukur pemerataan pekerjaan sesuai dengan kompeten diri setiap personal. Tetapi aplikasi ini lebih di khususkan untuk level operasional yang bertujuan mempermudah pengentrian, pelaporan data sehingga meningkatkan kualitas pelayanan.

2. Rumusan Masalah

Bagaimana sebuah sistem informasi pembayaran vakasi yang terkomputerisasi dan terintegrasi antara bagian – bagian yang berada dalam fakultas teknologi industri dan juga dapat digunakan sebagai monitoring atas pemerataan pekerjaan yang berdampak pada vakasi yang didapat oleh masing – masing staf. Serta sistem yang familiar di level operasional sehingga mudah dan cepat dalam pengoperasiannya

3. Batasan Masalah

Dalam sistem informasi pembayaran vakasi ini, terdapat beberapa batasan masalah antara lain:

1. Sistem hanya menangani masalah keuangan.
2. Analisis system keuangan berdasarkan prosedur – prosedur kerja atau kebijakan yang ada dalam kepala bagian dan berkaitan dengan keuangan lembaga pendidikan ini khususnya di fakultas teknologi industri.
3. Sistem terpisah dari system yang sudah ada dalam lembaga pendidikan ini.

C. PENYELESAIAN MASALAH

1. Usulan

Merancang dan mengimplementasikan suatu sistem informasi pembayaran vakasi Fakultas Teknologi Industri Universitas Islam Indonesia untuk memudahkan monitoring pada level managerial dan pelaporan serta entry data dan cepat dalam pertukaran data antar bagian – bagian yang terkait pada level operasional

2. Langkah Penyelesaian

1. Analisa kasus
2. Mengumpulkan data
3. Merancang sistem
4. Membuat sistem
5. Melakukan pengujian sistem
6. Menyelesaikan sistem
7. Penulisan laporan

D. DAFTAR PUSTAKA UTAMA

- Arbie. *Manajemen Database dengan MySQL*. Yogyakarta : Andi Offset. 2004
- Noerlina, dkk. *Perancangan sistem informasi: berbasis object oriented, Studi kasus*. Jakarta: Mitra Wacana Media. 2007
- Sunarto, Rumono B. *Membangun Sistem Akuisisi Data berbasis database dengan Delphi*. Jakarta : PT Elex Media Komputindo.2004

E. DOSEN PEMBIMBING YANG DIUSULKAN

Zainudin Zuhri, ST, M,Sc

Yogyakarta, 8 Desember 2009



Firmadi Biastiko

Nilai Kerja Praktek : A

Dosen Pembimbing Kerja Praktek : Ahmad Munasir Rafi'e Pratama, ST

Diisi oleh Notulis Rapat Dewan Dosen

1. Hasil Evaluasi : Disetujui / ~~Disetujui dengan perbaikan~~ / Ditolak *)

Dengan dosen Pembimbing :

a. _____ (Payung / Tunggal)

b. 22 (Pelaksana)

2. Catatan :

Alasan Penolakan Usulan Tugas Akhir :

- Proyek TA terlalu mudah.
- Pernah ada topik sejenis.
- Metode utama telah banyak dipakai dalam topik lainnya.
- Metode yang dipakai tidak jelas.
- Masalah terlalu sempit.
- Referensi tidak relevan dengan topik.
-
-

3. Usulan penelitian yang disetujui ini, harus dilampirkan pada Laporan Tugas Akhir

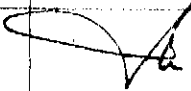



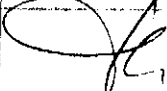
Yogyakarta, 12 - 2009

Pimpinan Rapat Dewan Dosen
Jurusan Teknik Informatika

*) Coret yang tidak perlu

**PRESENSI PRESENTASI KEMAJUAN TA
 LAMPIRAN USULAN TA**

Nama : Firmadi Biastiko
 No.Mahasiswa : 06 523186

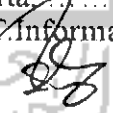
Tanggal	Judul TA	No.Mahasiswa	Nama Mhs Yg Presentasi	T.T.Dosen *
13 / 10 - 2009	Aplikasi Patroli kendaraan Sekolah berbasis Multimedia	05 523186	Yuli Marisna	
13 / 10 - 2009	Ensiklopedia Fenomena kehidupan sehari-hari berbasis multimedia	05 523 225	Annindya Ayu	
13 / 10 - 2009	Belajar menghitung dengan matematika Perkalian dan pembagian berbasis multimedia	05 523 229	Ramadhan Tri Aditama	
13 / 10 - 2009	Implementasi steeganografi Pada media audio visual	03 523 125	Hasc Adi P	
13 / 10 - 2009	SI 6 " Jalur Bus kota Yogyakarta"	05 523 060	Andi Kurnianto	

Catatan :

.....

Jogyakarta, 13 Oktober 2009

Ka. Jur. T. Informatika


 Yudi Prayudi, SSI.M.Kom

Tanda – Tangan Dosen pada saat proesestasi di laksanakan dan kertas ini jangan sampai hilang **

Surat Pernyataan

Yang bertanda tangan di bawah ini saya

Nama : F.i.r.m.a.d.i.....B.i.a.s.t.i.k.o

No.Mahasiswa :04.533.186.....

Menyatakan kesangupau untuk menyelesaikan tugas akhir selama
....., Terhitung Sejak usulan penelitian tugas akhir saya di
setujui.

Apabila saya tidak dapat menyelesaikan Tugas akhir sesuai dengan kesangupan
tsb di atas , maka saya bersedia untuk mengajukan judul baru

Jogjakarta 8 Desember 2009

Yang membuat pernyataan


(..... F.i.r.m.a.d.i...B.i.a.s.t.i.k.o)

Nama lengkap & Tanda -Tangan

UNIVERSITAS INDONESIA
الجامعة الإسلامية
الابدية الأندلسية

Tgl Mulai TA :
Tgl Batas Akhir TA :UNIVERSITAS ISLAM INDONESIA
Jurusan Teknik Informatika FTI UII
Jl. Kallurang Km 14,5 Sleman Jogjakarta

LEMBAR KONSULTASI TUGAS AKHIR (SKRIPSI)

Nama Mhs : FIRMADI BIASTIKO No.Mhs: 06523186
Judul TA : SISTEM INFORMASI PEMBAYARAN VAKASI
Dosen Pembimbing : 1. ZAINUDDIN ZUKHRU, ST, M.IT
2. _____

A. Konsultasi TA

Gunakan halaman belakang jika form ini tidak cukup **

Tanggal	Uraian	Paraf Pembimbing
11-12-2008	Konsultasi (atas belakang) rkn Bab I (Pendahuluan) Skripsi	<i>[Signature]</i>
24-12-2009	Desain database dan aturan - aturan Pembayaran Vakasi	<i>[Signature]</i>
18-1-2010	Desain database dan Rahn database aplikasi SIPEV	<i>[Signature]</i>
26-1-2010	Perancang SIPEV dan saran antara DFD dan UML	<i>[Signature]</i>
30-1-2010	Bab 2, Lanjutan Teori	<i>[Signature]</i>
8-2-2010	Uraian, sistem struktur, dan UML	<i>[Signature]</i>
10-3-2010	Bab 3, Analisis kebutuhan	<i>[Signature]</i>
20-3-2010	Bab 4, Perancangan Sistem	<i>[Signature]</i>
6-05-2010	demo Program dan Bab 5, Bab 6	<i>[Signature]</i>
12-8-2010	Bab 6 (analisis kebutuhan, Bab 7 (kesimpulan, saran)	<i>[Signature]</i>

Jogjakarta,
Dosen Pembimbing,

x

Form Permohonan Presentasi Kemajuan Tugas Akhir Lebih Awal *)

Kepada Yth, Ketua Jurusan Teknik Informatika FTI L II

Assalamu'alaikum Wr.Wb

Yang bertanda tangan di bawah ini :

Nama : _____ No, Mhs _____

Bermaksud untuk mengajukan permohonan untuk dapat melakukan presentasi kemajuan tugas akhir lebih awal yaitu pada periode bulan _____

Demikian permohonan ini,

Wassala'mualaikum Wr.Wb

Dosen Pembimbing,

Yogyakarta, _____
Hormat Kami,

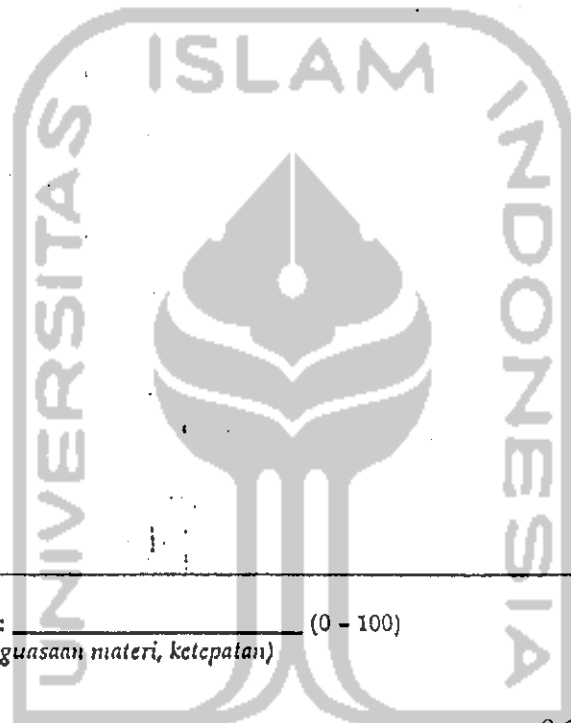
*) Hanya diisi bagi Mahasiswa yang akan melakukan presentasi lebih awal dan pada jadwal yang ditentukan jurusan.



SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Firmadi
No. Mhs. : 06 523 (B)
Judul TA : SIM Vakasi FTI - UII.

Lampirkan !!!



Nilai kemajuan Tugas Akhir: _____ (0 - 100)
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 09.03.2016.

Dosen,

Hermano

(nama terang)

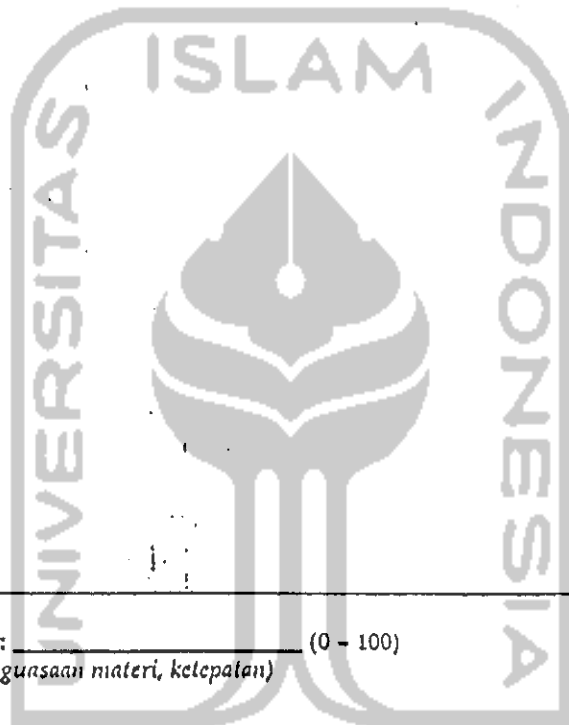
Dilampirkan pada Laporan TA yang diajukan untuk pendadaran



SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Firma N Bastiko
No. Mhs. : 06523186
Judul TA : ilm vakasi #Ti Uii


Lanjutkan !!



Nilai kemajuan Tugas Akhir: _____ (0 - 100)
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 09 Maret 2010

Dosen,


(Azmi Ansyah)
(nama terang)

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran

Nama : Azlifatul

Keterangan : Jurusan Teknik Informatika

LEMBAR LAPORAN KESIMPULAN PENGGUNAAN

Sistem Informasi Pembayaran Vaksinasi (SIPeV) FTI UII

1. Apakah anda ketahui tentang Sistem Informasi Pembayaran Vaksinasi?

Ya, tapi hanya garis besarnya saja.

2. Perlukah FTI UII memiliki Sistem Informasi Pembayaran Vaksinasi? Alasannya?

Perlu, untuk proses pembayaran di bank/langsung sudah sesuai apa belum.

User Interface dan menu

1. Bagaimana menurut anda penggunaan SIPeV (dari segi kemudahan penggunaan, fungsi dari sistem dan tampilan)?

Untuk penggunaan dari segi kemudahan relatif mudah, tapi perlu adanya training penggunaan lebih lanjut.

Tampilan dan fungsi sudah bagus, tapi perlu disederhanakan agar tidak rumit dalam pemakaiannya.

2. Apa yang perlu diperbaiki atau tambahan ke dalam sistem ini?

Perlu ditambah sistem utk pengguna dosen/personal, agar mereka juga bisa mengecek pembayaran yang diterima.

3. Perbandingan menurut anda sebelum dan sesudah adanya SIPeV ini?

Lebih tertata dlm pembayaran baik dalam penggunaan maupun hasilnya.

Nama : Wahyu Bp. S.M.Eng.

Keterangan : Wabek FTI

LEMBAR LAPORAN KESIMPULAN PENGGUNAAN

Sistem Informasi Pembayaran Vaksinasi (SIPeV) FTI UII

1. Apakah anda ketahui tentang Sistem Informasi Pembayaran Vaksinasi?

Ya. Saya pernah melihat perbedaannya.

2. Perlukah FTI UII memiliki Sistem Informasi Pembayaran Vaksinasi? Alasannya?

Sangat perlu, untuk mempermudah dan mempercepat serta transparansi penerimaan keuangan karyawan di FTI

User Interface dan menu

1. Bagaimana menurut anda penggunaan SIPeV (dari segi kemudahan penggunaan, fungsi dari sistem dan tampilan)?

mudah dan user friendly.

2. Apa yang perlu diperbaiki atau tambahan ke dalam sistem ini?

- harapan saya, semua karyawan dapat melihat besarnya nilai yg diterima.

- ke depan ~~akan~~ semua permintaan / disposisi dari satu unit ke unit lain dapat dilakukan dari sistem ini (Paperless)

3. Perbandingan menurut anda sebelum dan sesudah adanya SIPeV ini?

- Mudah - mudahan akan lebih efisien

Nama : ..MASIRAH..AMD...

Keterangan : ..KEUANGAN.....

LEMBAR LAPORAN KESIMPULAN PENGGUNAAN

Sistem Informasi Pembayaran Vakasi (SIPeV) FTI UII

1. Apakah anda ketahui tentang Sistem Informasi Pembayaran Vakasi?

Sistem yang memberikan informasi kepada Dosen / Karyawan tentang berapa jumlah honorarium yg mereka terima.

2. Perlukah FTI UII memiliki Sistem Informasi Pembayaran Vakasi? Alasannya?

Perlu sekali karena untuk mempermudah penyampaian informasi kepada Dosen / Karyawan juga efisiensi pekerjaan karena selama ini pemberitahuan ke dosen diadak manual.

User Interface dan menu

1. Bagaimana menurut anda penggunaan SIPeV (dari segi kemudahan penggunaan, fungsi dari sistem dan tampilan)?

Jumayan Bagus, sudah informatif.

2. Apa yang perlu diperbaiki atau tambahan ke dalam sistem ini?

Format Print Out berupa surat pemberitahuan resmi.

3. Perbandingan menurut anda sebelum dan sesudah adanya SIPeV ini.?

- Lebih aktif dan efisien.

Nama : Junati

Keterangan : Div - Adm - Umum

LEMBAR LAPORAN KESIMPULAN PENGGUNAAN

Sistem Informasi Pembayaran Vakasi (SIPeV) FTI UII

1. Apakah anda ketahui tentang Sistem Informasi Pembayaran Vakasi?

Tahu, namun belum secara detail

2. Perlukah FTI UII memiliki Sistem Informasi Pembayaran Vakasi? Alasannya?

Sangat perlu, alasan : guna transparansi keuangan, efektif dan efisien kerja.

User Interface dan menu

1. Bagaimana menurut anda penggunaan SIPeV (dari segi kemudahan penggunaan, fungsi dari sistem dan tampilan)?

Sedikit rumit saat penggunaannya, tetapi output yang memiliki kelengkapan tampilan sudah bagus.

2. Apa yang perlu diperbaiki atau tambahan ke dalam sistem ini?

Belum sistem ini personal tidak bisa mengakses

3. Perbandingan menurut anda sebelum dan sesudah adanya SIPeV ini?

Sebelum ada SIPeV pelaporan masih konvensional, setelah menggunakan SIPeV lebih mudah dan pelaporan manajemen.

Nama : Eni Uswatun

Keterangan : Kp/Ita

LEMBAR LAPORAN KESIMPULAN PENGGUNAAN

Sistem Informasi Pembayaran Vakasi (SIPeV) FTI UII

1. Apakah anda ketahui tentang Sistem Informasi Pembayaran Vakasi?

Sedikit tahu - (secara garis besar)

2. Perlukah FTI UII memiliki Sistem Informasi Pembayaran Vakasi? Alasannya?

Perlu sekali, supaya ada kesamaan dalam print out dan pelaksanaannya lebih lancar

User Interface dan menu

1. Bagaimana menurut anda penggunaan SIPeV (dari segi kemudahan penggunaan, fungsi dari sistem dan tampilan)?

Agak rumit, karena sangat detail untuk sistem tampilan lumayan bagus.

2. Apa yang perlu diperbaiki atau tambahan ke dalam sistem ini?

Dalam hal sistem ini hanya dipergunakan secara resmi ~~pariwisata~~ karena harus menggunakan password khusus

3. Perbandingan menurut anda sebelum dan sesudah adanya SIPeV ini?

Sebelum sistem ini ada laporan kevakasan belum tertata rapi.

Setelah ada sistem ini laporan dapat terbaca - mudah & dipahami