MEMBANGUN SISTEM INFORMASI MANAJEMEN SATU ATAP E-GOVERNMENT DENGAN ARSITEKTUR MODEL-VIEW-CONTROLLER BERBASIS JOOMLA

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika

Oleh:

Nama : **Choirul Ihwan**
No. Mahasiswa : **03 523 245**

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007

LEMBAR PENGESAHAN PEMBIMBING

MEMBANGUN SISTEM INFORMASI MANAJEMEN SATU ATAP E-GOVERNMENT DENGAN ARSITEKTUR MODEL-VIEW-CONTROLLER BERBASIS JOOMLA

TUGAS AKHIR

Oleh :

Nama : **Choirul Ihwan**

No. Mahasiswa : **03 523 245**

Yogyakarta, 2 Oktober 2007

Pembimbing I	**Pembimbing II**
(Fathul Wahid, ST. MS.c)	**(Hendrik, ST.)**

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Yang bertanda tangan di bawah ini
Nama : Choirul Ihwan
NIM : 03 523 245
Tugas Akhir dengan judul :

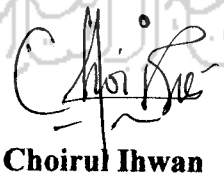
MEMBANGUN SISTEM INFORMASI MANAJEMEN SATU ATAP E-GOVERNMENT DENGAN ARSITEKTUR MODEL-VIEW-CONTROLLER BERBASIS JOOMLA

Dengan ini saya menyatakan dengan sesungguhnya bahwa dalam Tugas Akhir ini tidak terdapat keseluruhan tulisan atau karya yang saya ambil dengan menyalin, meniru dalam bentuk rangkaian kalimat atau simbol atau algoritma atau program yang menunjukkan gagasan atau pendapat atau pemikiran orang lain, yang saya aku seolah-olah sebagai tulisan atau karya saya sendiri.

Apabila saya melakukan hal tersebut di atas, baik sengaja atau tidak, dengan ini saya menyatakan menarik Tugas Akhir yang saya ajukan sebagai hasil karya saya sendiri ini. Bila di kemudian hari terbukti bahwa saya melakukan tindakan di atas, gelar dan ijazah yang telah diberikan oleh Universitas Islam Indonesia batal saya terima.

Yogyakarta, 2 Oktober 2007

Yang Membuat Pernyataan



Choirul Ihwan

LEMBAR PENGESAHAN PENGUJI

MEMBANGUN SIMTAP E-GOVERNMENT
DENGAN ARSITEKTUR MVC MENGGUNAKAN JOOMLA

TUGAS AKHIR

Oleh:

Nama : Choirul Ihwan

NIM : 03 523 245

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

Yogyakarta, 29 Oktober 2007

Tim Penguji

Fathul Wahid, ST, M.Sc
Ketua

Hendrik, ST
Anggota I

Lizda Iswari, ST
Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

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

PERSEMBAHAN

Rasa puji syukur saya panjatkan kepada Allah SWT atas karunia dan rahmat-Nya saya bisa menyelesaikan tugas akhir ini dengan baik dan semoga akan dapat bermanfaat dikemudian hari atau bagi orang lain.

Sholawat dan Salam tak lupa saya haturkan kepada Nabi Muhammad SAW, karena beliau semoga saya menjadi orang yang selalu benar langkahnya dan diridloi oleh Allah...

To Almarhum Ayah, semoga senantiasa bahagia disisiNya,
dan Ibu yang telah mencurahkan kasih sayang yang tak terhingga ..
Kakak-kakakku..Kak Niam, Kak Anam, dan mbak Luluk,
Keponakan-keponakanku, Ikhul, Didin, Ali, dan Ibnu yang imut dan selalu lucu
Dan semua keluarga serta famili yang lain yang tidak bisa disebutkan satu persatu disini

Teman" Lab Sirkel n Lab Terpadu,
Santri" Pondok Pesantren UII,khususnya Mustofa dan Ari yang setia menemaniku
Teman-teman GamaTechno, yang senantiasa berdiskusi tentang kemajuan dunia TI
Thank's for support, do'a dan semua dukungannya

Buat mas hendrik, makasih atas bimbingannya,
dan sorry mahasiswa bimbinganmu yang satu ini memang jarang kekampus,
Muzid dan Uji yang sering tak tanyain urusan Tugas Akhir
Dan Nu2nk, thanks for giving me information about recent news in campus

Semua kawan, sahabat, serta rival"ku,
baik yang aku kenal maupun yang belum aku kenal,
terima kasih atas persahabatan n motivasinya, Great to meet u all guys...

For All Every One I mentioned above as a part or a unity.....
I am nothing withour you All.....and the last
I just want to say once more time Thank's a lot .....

MOTTO

"... Allah akan meninggikan orang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat... ..."

(QS.Al-Mujaadilah ayat 11)

"Bacalah.... bacalah dengan menyebutkan nama Tuhanmu yang menciptakan. "

(QS. Al-Alaq ayat 1-2)

"Manusia tidak akan mendapatkan sesuatu kecuali dari apa yang ia usahakan ".

(Q.S. Al Baqarah ayat 153)

" Sesungguhnya sesudah kesulitan itu ada kemudahan ; Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain ".

(Q.S. Asy Syarh ayat 6 dan 7)

KATA PENGANTAR

بسم الله الرحمن الرحيم

Assalamu'alaikum Wr.Wb

Alhamdulillah, segala puji bagi Allah SWT atas segala rahmat, hidayah dan inayah yang diberikanNya, sehingga penulisan laporan tugas akhir yang berjudul ***MEMBANGUN SIMTAP E-GOVERNMENT DENGAN ARSITEKTUR MVC MENGGUNAKAN JOOMLA*** dapat diselesaikan dengan baik. Sholawat serta salam juga tak lupa senantiasa dipanjatkan kepada Nabi Muhammad SAW beserta para kerabat, sahabat – sahabatnya dan seluruh keturunannya hingga hari kiamat nanti.

Laporan tugas akhir ini disusun sebagai salah satu syarat guna memperoleh gelar Sarjana Teknik Informatika pada Universitas Islam Indonesia. Dan juga sebagai sarana untuk mempraktekkan secara langsung ilmu dan teori yang telah diperoleh selama menjalani masa studi di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia..

Dalam pembuatan laporan tugas akhir ini, penulis mendapat banyak bimbingan, pengarahan, bantuan maupun motivasi yang sangat bermanfaat bagi penulis khususnya dari berbagai pihak. Oleh karena itu dalam kesempatan ini dengan segala kerendahan hati, ijinkanlah penulis untuk menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

a. Ibu yang senantiasa memberikan do'a restunya, serta kasih sayang yang tak pernah habis bagi penulis.
b. Bapak Rektor dan seluruh jajaran Rektoriat Universitas Islam Indonesia.
c. Para pendiri Pondok Pesantren UII, yang telah memberikan saya kesempatan untuk mengenyam pendidikan di Universitas yang tercinta ini.
d. Bapak Fathul Wahid, ST., M.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
e. Bapak Yudi Prayudi, S.Si., M.Kom, selaku Ketua Jurusan Teknik Informatika.

f. Bapak Hendrik, dan Bapak Fathul Wahid selaku dosen pembimbing yang telah memberikan pengarahan, bimbingan, serta masukan selama pelaksanaan tugas akhir dan penulisan laporan.
g. Dosen-dosen Jurusan Teknik Informatika. Terima kasih atas ilmu pengetahuan dan kerjasamanya.
h. Kakak-kakakku tersayang, Cak Niam, Cak Anam, Mbak Luluk. Terima kasih atas semua dukungannya, baik moral maupun material.
i. Keponakan-keponakanku, Ikhul, Didin, Ali, Ibnu, yang selalu menumbuhkan inspirasi baru pada setiap tatapan matanya.
j. Teman-teman asisten laboratorium SIRKEL, dan juga seluruh keluarga besar laboratorium Terpadu Informatika atas dukungan yang diberikan.
k. Teman-teman Santri Pondok Pesantren UII, khususnya angkatan 2003, atas kerjasama dan motivasinya, dan juga Supriyadi, Mustofa, Ari wibowo atas kesabaran dan dorongannya.
l. Teman-teman GamaTechno, khususnya Gosonx, Ronald, mas Triasmono, mas Nanang, dan TamRizal atas bantuan yang diberikan, kegilaan yang ditularkan, dan kebersamaan yang tak pernah terlupakan.
m. Teman-teman angkatan 2003 semuanya atas kebersamaan dan kekompakannya.
n. Sahabat-sahabat kami yang belum tersebut di manapun mereka berada.
o. Semua pihak yang telah memberikan bantuan dan dorongan.

Semoga Allah SWT melimpahkan segala rahmat, hidayah dan inayahnya kepada semua pihak yang telah membantu terselesaikannya penulisan laporan tugas akhir ini.

Penulis menyadari bahwa dalam penyusunan laporan tugas akhir ini masih banyak terdapat kekeliruan dan kekurangan. Untuk itu penulis menyampaikan permohonan maaf sebelumnya serta sangat diharapkan kritik dan saran yang sifatnya membangun untuk penyempurnaan di masa mendatang.

Akhir kata semoga laporan ini dapat bermanfaat bagi penulis dan semua pembaca. Amien ya robbal-alamien.....

Wassalamu'alaikum Wr.Wb.

Yogyakarta, 04 Oktober 2007

Choirul Ihwan

SARI

Sistem Informasi Manajemen Satu Atap (SIMTAP) adalah suatu sistem informasi yang diterapkan dalam pemerintahan untuk menangani pemberian layanan kepada masyarakat yang berhubungan dengan pengurusan dokumen. SIMTAP merupakan bagian dari penerapan *E-Government*.

Joomla merupakan sebuah *Content Management System* (CMS) yang dapat digunakan untuk berbagai keperluan sesuai dengan tujuan *developer* yang *menggunakannya*.

Model-View-Controller (MVC) merupakan suatu bentuk arsitektur dalam mendesain sistem informasi yang dibangun dengan memisahkan *data model* dari *user interface*. Penggunaan MVC akan memudahkan dalam pengembangan program berikutnya.

Pembuatan SIMTAP menggunakan Joomla dengan arsitektur MVC dapat membantu proses perizinan di suatu wilayah sehinga pelayanan pemerintah terhadap masyarakat akan lebih efektif dan efisien. Selain itu dalam pengembangan program, SIMTAP yang dibangun akan menjadi lebih mudah untuk dimodifikasi karena *data model* dan *user interface* telah dipisahkan, sehingga perubahan yang terjadi pada *user interface* tidak mempengaruhi pada data, begitu juga *graphic designer* dapat berkreasi seindah mungkin dalam membuat tampilan tanpa harus kuatir akan merusak tampilan yang sudah ada.

Kata kunci: SIMTAP, e-Government, CMS, Joomla, MVC.

TAKARIR

bug, error	kesalahan program
user requirements	kebutuhan-kebutuhan pengguna
user interface	tampilan program
fleksibilitas	kemudahan penggunaan
reusable	penggunaan kembali
content	isi
free	bebas
developer	pengembang
flowchart	diagram alir
workflow	alur kerja
system requirements	kebutuhan sistem
system analyst	analis sistem
breakdown	dipecah
package	paket
dependency	bergantung
paging	pengaturan halaman
record	baris data
wizard	serangkaian langkah
request	permintaan data
response	tanggapan

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING i
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR ii
LEMBAR PENGESAHAN PENGUJI iii
PERSEMBAHAN iii
MOTTO iv
SARI v
TAKARIR ix
DAFTAR ISI x
DAFTAR TABEL xi
DAFTAR GAMBAR xiii
DAFTAR GAMBAR xiv
BAB I xiv
PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 1
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 2
1.5 Manfaat Penelitian 2
1.6 Metode Penelitian 2
1.6.1 Metode Pengumpulan Data 3
1.6.2 Metode Pengembangan Sistem 3
1.7 Sistematika Penulisan 3
BAB II 4
LANDASAN TEORI 6
2.1 Sistem Informasi 6
2.2 SIMTAP 6
2.3 Model-View-Controller 7
2.4 Joomla CMS 9
BAB III 10
METODOLOGI 12
3.1 Analisis Kebutuhan 12
3.1.1 Masukan Sistem 12
3.1.2 Keluaran Sistem 14
3.1.3 Perangkat Lunak yang Dibutuhkan 15
3.1.4 Perangkat Keras yang Dibutuhkan 15
3.2 Desain Sistem 15
3.2.1 Use case Diagram 16
3.2.2 Sequence Diagram 16
3.2.3 Class Diagram 18
3.2.4 Tabel Relasional 23
3.2.5 Rancangan Antar Muka 24
3.3 Implementasi 34
..... 43

3.3.1 Perangkat Keras 43
3.3.2 Perangkat Lunak 44
3.3.3 Implementasi Sistem 44
3.3.3.1 Model-View-Controller SIMTAP 44
3.3.3.2 Grafik SIMTAP 50
3.3.3.3 Manajemen User 51
3.3.3.4 Library lain 52
3.3.4 Struktur File 52
BAB IV 54
HASIL DAN PEMBAHASAN 54
4. 1. Hasil dan Pembahasan 54
4.1.1 Halaman Muka 54
4.1.2 Halaman Download 55
4.1.3 Halaman setelah login 55
4.1.4 Halaman Kecamatan 56
4.1.5 Halaman Pemohon 58
4.1.6 Halaman Pemberi Izin 61
4.1.7 Halaman Daftar Layanan 62
4.1.8 Halaman Daftar Permohonan 69
4.1.9 Halaman Surat Survey 72
4.1.10 Halaman Surat Izin 73
4.1.11 Halaman Slip Pembayaran 75
BAB V 78
KESIMPULAN DAN SARAN 78
5.1 Kesimpulan 78
5.2 Saran 79
DAFTAR PUSTAKA 80

DAFTAR TABEL

Tabel 3.1 Struktur tabel ref_kecamatan 26
Tabel 3.2 Struktur tabel jos_ref_kewarganegaraan 26
Tabel 3.3 Struktur tabel jos_ref_status_proses 26
Tabel 3.4 Struktur tabel jos_ref_status_izin 27
Tabel 3.5 Struktur tabel jos_ref_prosedur_layanan 27
Tabel 3.6 Struktur tabel jos_ref_jenis_identitas 27
Tabel 3.7 Struktur tabel jos_ref_kategori _layanan 28
Tabel 3.8 Struktur tabel jos_ref_instansi 28
Tabel 3.9 Struktur tabel jos_ref_kop_surat 28
Tabel 3.10 Struktur tabel jos_ref_syarat_layanan 29
Tabel 3.11 Struktur tabel jos_jenis_layanan 29
Tabel 3.12 Struktur tabel jos_ref_pemberi_izin 30
Tabel 3.13 Struktur tabel jos_survey 30
Tabel 3.14 Struktur tabel jos_pemohon 31
Tabel 3.15 Struktur tabel jos_permohonan 32
Tabel 3.16 Struktur tabel jos_kuitansi 32
Tabel 3.17 Struktur tabel jos_kuitansi_rincian 33
Tabel 3.18 Struktur tabel jos_surat_izin 33
Tabel 3.19 Struktur tabel jos_surat_izin_data 34
Tabel 3.20 Struktur tabel jos_survey_data 34

DAFTAR GAMBAR

Gambar 2.1 Prosedur pengurusan perizinan SIMTAP 9
Gambar 3.1 Activity diagram bisnis proses SIMTAP 13
Gambar 3. 2 Use case diagram pelayanan perizinan 17
Gambar 3. 3 Sequence diagram data referensi 19
Gambar 3. 4 Sequence diagram data master 20
Gambar 3.5 Sequence diagram pengajuan permohonan 21
Gambar 3.6 Sequence diagram pemrosesan permohonan 22
Gambar 3.7 Class diagram package daftar layanan 23
Gambar 3.8 Class diagram package permohonan 24
Gambar 3.9 Tabel relasional database SIMTAP 25
Gambar 3.10 Rancangan antar muka halaman utama SIMTAP 35
Gambar 3.11 Rancangan antar muka backoffice pasca login 36
Gambar 3.12 Rancangan antar muka halaman view content 37
Gambar 3.13 Rancangan antar muka halaman tambah data 37
Gambar 3.14 Rancangan antar muka halaman konfirmasi penghapusan 38
Gambar 3.15 Rancangan antar muka halaman tambah pemohon 39
Gambar 3.16 Rancangan antar muka halaman tambah pemberi izin 39
Gambar 3.17 Rancangan antar muka halaman tambah layanan 40
Gambar 3.18 Rancangan antar muka halaman tambah syarat layanan 41
Gambar 3.19 Rancangan antar muka halaman tambah prosedur layanan 41
Gambar 3.20 Rancangan antar muka langkah I pembuatan draft 42
Gambar 3.21 Rancangan antar muka langkah II pembuatan draft 42
Gambar 3.22 Rancangan antar muka langkah III pembuatan draft 43
Gambar 3.23 Struktur file komponen-komponen SIMTAP 53
Gambar 3.24 Struktur file komponen instansi 53
Gambar 4.1 Halaman muka SIMTAP 54
Gambar 4.2 Halaman unduh formulir 55
Gambar 4.3 Halaman setelah login 56
Gambar 4.4 Halaman kecamatan 57
Gambar 4.5 Halaman edit kecamatan 57
Gambar 4.6 Halaman konfirmasi penghapusan kecamatan 58
Gambar 4.7 Halaman pemohon 59
Gambar 4.8 Halaman tambah pemohon 60
Gambar 4.9 Halaman detail pemohon 61
Gambar 4.10 Halaman pemberi izin 61
Gambar 4.11 Halaman tambah pemberi izin 62
Gambar 4.12 Halaman jenis layanan 63
Gambar 4.13 Halaman tambah jenis layanan 63
Gambar 4.14 Halaman tambah syarat layanan 64
Gambar 4.15 Halaman tambah prosedur layanan 65
Gambar 4.16 Halaman detail jenis layanan 66
Gambar 4.17 Halaman langkah I pembuatan draft surat izin 67
Gambar 4.18 Halaman langkah II pembuatan draft surat izin 68

Gambar 4.19 Halaman langkah III pembuatan draft surat izin 69
Gambar 4.20 Halaman daftar permohonan..... 70
Gambar 4.21 Halaman tambah permohonan 71
Gambar 4.22 Halaman detail permohonan 71
Gambar 4.23 Form pembuatan surat survey..... 72
Gambar 4.24 Halaman preview surat survey..... 73
Gambar 4.25 Form penulisan surat izin..... 74
Gambar 4.26 Halaman preview surat izin 75
Gambar 4.27 Form penulisan slip pembayaran 76
Gambar 4.28 Form rincian biaya slip pembayaran..... 76
Gambar 4.29 Halaman detail slip pembayaran..... 77

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan dan keuntungan yang didapatkan dari penerapan teknologi informasi mendorong berbagai macam organisasi untuk membuat sistem informasi dalam organisasinya. Begitu juga dengan organisasi/instansi pemerintah. Instansi-instansi tersebut berlomba-lomba untuk membangun teknologi informasi di wilayahnya masing-masing.

Berdasarkan Inpres No 3/2003 tentang Kebijakan Dan Strategi Nasional Pengembangan *e-Government*, "Setiap Gubernur dan Bupati/Walikota diamanatkan untuk mengambil langkah-langkah yang diperlukan sesuai dengan tugas, fungsi dan kewenangannya masing-masing guna terlaksananya pengembangan *e-Government* secara nasional" [HAR05].

Salah satu bentuk *e-Government* adalah SIMTAP (Sistem Informasi Manajemen Satu Atap). Penerapan SIMTAP dimaksudkan untuk meningkatkan kualitas layanan publik lembaga pemerintahan. Misalnya dalam hal pengurusan izin. Sebelum adanya *e-Government*, pengurusan izin terkesan berbelit-belit dan membutuhkan waktu lama. Faktor ini seringkali menjadikan para investor takut untuk menanamkan modalnya di wilayah tersebut. Selain itu panjangnya proses birokrasi menimbulkan potensi terjadinya penyelewengan oleh instansi pemerintahan terkait.

Sistem informasi perizinan satu atap umumnya bersifat regional dan diterapkan pada wilayah tertentu saja. Oleh karena itu perlu adanya SIMTAP yang memuat hal-hal yang umum dalam regulasinya, sehingga suatu daerah tidak memerlukan usaha yang besar dan mahal untuk menerapkan SIMTAP di daerahnya. Selain itu, pemerintah daerah juga dapat dengan mudah menambahkan jenis layanan ke dalam SIMTAP.

Joomla merupakan salah satu Content Management System (CMS) berbasis

web yang dapat digunakan untuk mengatur dan mempublikasikan dokumen di internet. Karena gratis dan bersifat *open source* Joomla banyak digunakan di seluruh dunia. Selain itu, karena dibangun dengan menggunakan PHP, Joomla juga relatif lebih ringan dibanding dengan aplikasi yang lain. Karena beberapa faktor itulah pembangunan SIMTAP *e-Government* ini menggunakan Joomla sebagai *framework* aplikasi.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah bagaimana membuat SIMTAP berbasis web dengan menggunakan PHP, MySQL dan Joomla yang dapat diterapkan pada pemerintah daerah di Indonesia.

1.3 Batasan Masalah

Tujuan pembatasan masalah di sini adalah untuk menyederhanakan persoalan yang dihadapi dan untuk memastikan bahwa persoalan tersebut tidak melebar sehingga menyimpang dari tujuan yang diinginkan. Batasan masalah pada penelitian ini antara lain :

1. Aplikasi yang dibangun berbasis web
2. Ruang lingkup SIMTAP adalah pemerintah kabupaten.
3. Permasalahan yang diselesaikan adalah penanganan pelayanan masyarakat terkait dengan perizinan atau dokumentasi kependudukan.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah membuat SIMTAP yang dapat digunakan oleh pemerintah kabupaten di Indonesia dan dapat memberikan layanan-layanan kepada publik berbasis web dengan menggunakan PHP, MySql dan Joomla.

1.5 Manfaat Penelitian

Ada beberapa manfaat yang dapat diambil dari penelitian ini, antara lain :

1. Pemkab/Pemkot dapat menerapkan SIMTAP di wilayahnya masing-masing.

2. Efektifitas dan efisiensi pemberian layanan semakin meningkat.
3. Mengetahui bagaimana menambahkan komponen, modul maupun *plug-in* pada Joomla.
4. Mengetahui bagaimana penerapan arsitektur Model-View-Controller (MVC) pada aplikasi.

1.6 Metode Penelitian

Metode penelitian merupakan metode yang digunakan dalam pengumpulan data dan pengembangan sistem. Ada beberapa metode yang digunakan pada penelitian ini, antara lain :

1.6.1 Metode Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan studi pustaka, yaitu pengumpulan data dengan cara melakukan studi dan analisis literatur, dan sumber catatan lain yang berkaitan dengan permasalahan yang dibahas. Adapun sumber data digali dari internet, jurnal hasil prosiding, buku dan sumber data lain yang berhubungan dengan tema penelitian. .

1.6.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *iteratif incremental*. Pada metode ini pembuatan aplikasi dilakukan dengan membuat bagian sistem yang paling sederhana, kemudian secara bertahap dikembangkan dan dimodifikasi dengan menambahkan fungsionalitas tertentu sampai aplikasi secara total selesai dibuat. Pemilihan metode *iteratif incremental*, didasarkan pada karakteristik metode ini yang lebih adaptif terhadap perubahan *user requirements* karena satu siklus membutuhkan waktu yang tidak terlalu lama dan perbaikan kesalahan-kesalahan (*bug, error*) yang ditemui dan perubahan-perubahan dari *user requirements* akan segera dapat dilakukan pada siklus berikutnya[AZI05]. Dalam metode *iteratif incremental*, ada beberapa siklus pengembangan yang dilakukan, yaitu : Perencanaan, Analisis Kebutuhan,

Perancangan Sistem, Implementasi, Pengujian, Evaluasi dan Instalasi [WIK07].

1. Perencanaan

   Tahap perencanaan digunakan untuk merencanakan pembuatan SIMTAP.

2. Analisis kebutuhan

   Pada tahap ini analisis dilakukan untuk menggambarkan bisnis proses dari SIMTAP. Analisis menghasilkan data untuk keperluan input, output yang diharapkan serta kebutuhan perangkat keras dan perangkat lunak untuk menjalankan SIMTAP.

3. Perancangan sistem

   Perancangan sistem dilakukan untuk merancang arsitektur SIMTAP yang akan dibangun. Perancangan sistem akan menghasilkan diagram-diagram arsitektur sistem.

4. Implementasi

   Pada tahap inilah SIMTAP dibangun dengan menggunakan bahasa pemrograman PHP, DBMS MySQL dan CMS Joomla.

5. Pengujian

   Pengujian pada tahap ini merupakan pengujian unit, dalam artian pengujian dilakukan pada unit-unit terkecil dari sistem.

6. Evaluasi

   Evaluasi dilakukan dengan melakukan pengujian pada keseluruhan sistem untuk menentukan apakah sistem yang dibangun telah sesuai dengan harapan.

7. Instalasi

   Tahap instalasi dilakukan untuk memasang sistem pada komputer yang akan menggunakan sistem tersebut.

1.7 Sistematika Penulisan

Untuk memudahkan pemahaman penelitian, laporan penelitian ini disusun menjadi 5 bab dengan sistematika sebagai berikut :

BAB I merupakan pengantar dan gambaran terhadap permasalahan yang akan dibahas. Ada beberapa hal yang dibahas pada bab I, antara lain: latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II membahas tentang landasan teori yang digunakan dalam pemecahan masalah, dalam hal ini teori yang berkaitan dengan SIMTAP, Joomla dan MVC.

BAB III membahas tentang metodologi yang digunakan dalam pembuatan sistem, mulai dari tahap analisis, desain, implementasi, dan pengujian.

BAB IV membahas tentang hasil SIMTAP, jalannya sistem serta gambaran detail dari sistem yang dibangun.

BAB V berisi kesimpulan yang diperoleh dari penelitian dan saran-saran yang diperlukan sebagai masukan untuk perbaikan sistem di masa yang akan datang.

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

Sistem informasi adalah suatu cara untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan[WAH04].

Sistem informasi merupakan suatu sistem yang dibuat manusia untuk menyajikan informasi-informasi tertentu. Sistem informasi dalam suatu organisasi digunakan untuk pemenuhan kebutuhan pengolahan transaksi, operasional, manajerial dan kegiatan strategis dari suatu organisasi serta menyediakan laporan tertentu untuk pihak-pihak yang membutuhkan informasi organisasi.

Sistem informasi menerima input berupa data atau instruksi, memproses input sesuai instruksi kemudian mengeluarkan hasil pemrosesan berupa output. Selain itu dalam sistem informasi juga diperlukan adanya media penyimpanan file (*data file storage*) untuk menyimpan hasil pengolahan sebelumnya sehingga dapat digunakan kembali ketika diperlukan.

Sistem informasi terdiri dari beberapa komponen, antara lain :

- Blok Masukan

  Berupa metode-metode dan media untuk menangkap data yang akan dimasukkan yang biasanya berupa dokumen-dokumen dasar.

- Blok Model

  Berupa kombinasi prosedur, logika dan model matematik yang berfungsi memanipulasi data untuk keluaran tertentu.

- Blok Output

  Berupa data-data keluaran misalnya informasi atau laporan-laporan tertentu.

- Blok Teknologi

  Berupa teknologi yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan output serta

membantu pengendalian sistem secara keseluruhan. Blok teknologi ini merupakan komponen yang membantu memperlancar proses pengolahan data di dalam sistem.

- Blok Basis Data

  Berupa kumpulan data yang saling berhubungan yang tersimpan di perangkat keras dan perangkat lunak komputer.

- Blok Kendali

  Berupa mekanisme yang berfungsi untuk mencegah dan menangani kesalahan/kegagalan dalam sistem.

Seiring kemajuan zaman, perkembangan sistem informasi semakin pesat. Hal ini karena manfaat sistem informasi yang sangat besar dalam membantu aktivitas organisasi. Sekarang hampir semua jenis organisasi berlomba-lomba untuk menerapkan sistem informasi dalam setiap aktivitasnya. Bermacam-macam istilah pun muncul, mulai dari *e-Learning* yang digunakan untuk pembelajaran sampai *e-Government* yang digunakan untuk pemerintahan.

2.2 SIMTAP

e-Government merupakan istilah dalam penggunaan teknologi informasi oleh pemerintah untuk pertukaran informasi dan pelayanan dengan penduduk, pelaku bisnis, dan penyelenggaran pemerintahan yang lain. Layanan utama yang diberikan oleh *e-Government* adalah *Government-to-Citizen (G2C)*, *Government-to-Business* (G2B), Government-*to-Government* (G2G) dan *Government-to-Employees* (G2E) [WIK07].

Salah satu bentuk layanan *Government-to-Citizen (G2C)*, *Government-to-Business* (G2B) adalah SIMTAP. SIMTAP (Sistem Informasi Manajemen Satu Atap) merupakan sistem informasi yang mengelola data perizinan dan layanan masyarakat lainnya yang terdapat dalam suatu wilayah pemerintahan, secara terpusat dan terintegrasi di dalam satu lokasi (satu atap) dengan memanfaatkan Teknologi Informasi [IND07].

Ada beberapa hal yang menjadi pertimbangan diperlukannya SIMTAP sebagai solusi efisiensi dan efektifitas pelayanan masyarakat oleh pemerintah daerah maupun pemerintah kota, antara lain :

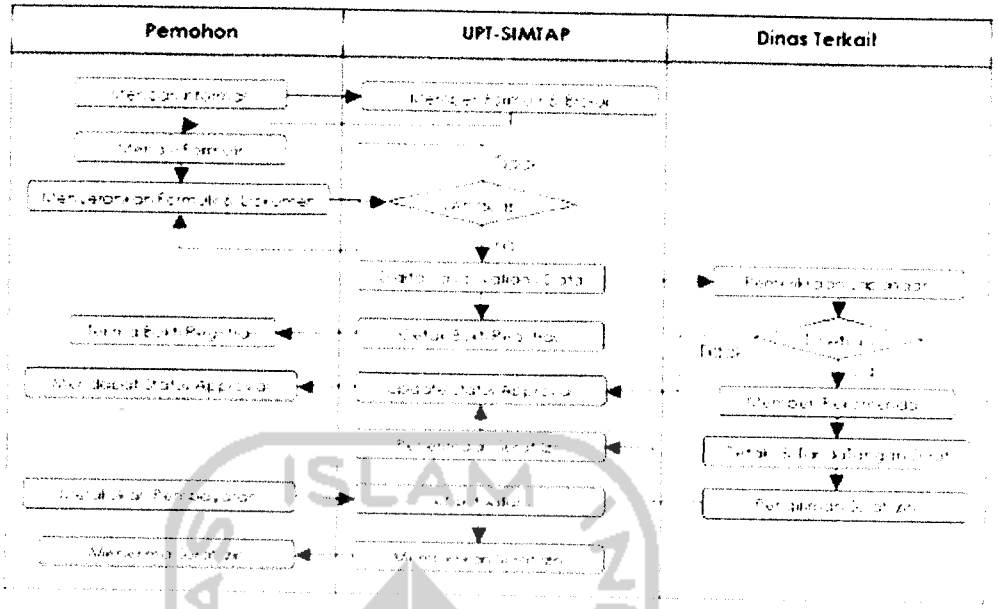
1. Banyaknya jenis perizinan.
2. Banyaknya institusi pengelola perizinan
3. Prosedur/mekanisme perizinan yant sering berbelit-belit dan melibatkan banyak institusi.
4. Persyaratan perizinan yang banyak, berat dan seringkali berulang-ulang untuk berbagai izin.
5. Waktu penyelesaian izin relatif lama.
6. Jarak jangkau jauh, waktu tempuh lama, dan biaya transportasi mahal.

Sedangkan tujuan dibuatnya SIMTAP sendiri antara lain :

1. Meningkatkan mutu pelayanan publik.
2. Meningkatkan partisipasi masyarakat untuk melengkapi berbagai perizinan yang dibutuhkan sesuai perundang-undangan yang berlaku.
3. Meningkatkan kinerja pelayanan pemerintah kepada masyarakat, khususnya dalam hal perizinan.
4. Mengembangkan basisdata sebagai sarana penyedia informasi agar dapat dimanfaatkan dalam perencanaan, pelaksanaan dan pengawasan pembangunan.
5. Mengembangkan basis data yang dapat diakses oleh institusi – institusi pemerintah atau swasta terkait sehingga diperoleh informasi yang integral dan terpercaya.

Umumnya alur SIMTAP antara satu daerah dengan daerah yang lain sama. Mulai dari pengambilan formulir sampai dengan pencetakan surat izin permohonan.

Alur yang mendeskripsikan prosedur pengurusan perizinan di SIMTAP, mulai dari pemohon mencari informasi sampai dengan penerimaan surat izin ditunjukkan pada gambar 2.1 berikut :



Gambar 2.1 Prosedur pengurusan perizinan SIMTAP

Proses administrasi layanan publik terjadi di kantor UPTS sementara proses pemeriksaan lapangan dan pengesahan surat izin terjadi di instansi terkait. Petugas yang berada di UPTS adalah perwakilan dari masing-masing instansi [KUT07]. Mereka bertugas memverifikasi setiap permohonan yang masuk.

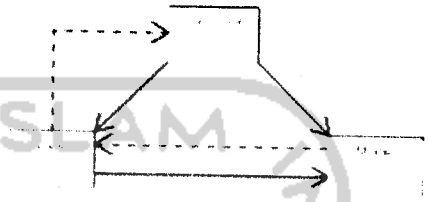
2.3 Model-View-Controller

MVC *(Model-View-Controller)* merupakan suatu teknik untuk desain sistem dalam proses pengembangan perangkat lunak. MVC memisahkan data (*model*) dan *user interface* (*view*), sehingga perubahan pada data tidak akan mengakibatkan perubahan pada *user interface* dan data dapat di manipulasi tanpa merubah *user interface*. Konsep ini dilakukan dengan melepaskan keterhubungan langsung *data access* dan *business logic* dari *data presentation* dengan membuat suatu komponen perantara yang dikenal dengan *controller*.

Model merepresentasikan informasi pada aplikasi. *Model* merupakan gabungan antara *domain logic*(data) dengan *data access layer* yang mengakses data tersebut. *View* memparsing *model* kedalam form yang sesuai. Sebuah *data*

model dapat memiliki banyak *view* ataupun sebaliknya sesuai dengan kebutuhan. *Controller* menerima *request* dan merespon *request* yang diberikan oleh pengguna.

Diagram yang menggambarkan hubungan antara *model*, *view* dan *controller* ditunjukkan pada gambar 2.2 berikut :



Gambar 2.2 Hubungan model, view, controller

Pada MVC, pengguna melakukan interaksi dengan *user interface* (melalui klik tombol dan lain-lain). *Controller* kemudian menangani *request* dari pengguna. Selanjutnya *controller* mengakses *model* sesuai dengan *request* dari pengguna. *View* kemudian melakukan generate *user interface* sesuai dengan *response* dari model.

Ada beberapa keuntungan dengan menggunakan MVC, antara lain :

1. Kompleksitas dalam arsitektur perangkat lunak dapat dikurangi
2. Meningkatkan *fleksibilitas* aplikasi
3. Memudahkan dalam *reusable*.

2.4 Joomla CMS

Content Management Sytem (CMS) adalah perangkat lunak komputer yang dapat digunakan untuk mengorganisir dan memfasilitasi pembuatan dokumen maupun *content* yang lain. Umumnya CMS digunakan untuk aplikasi web. Joomla merupakan salah satu CMS yang *free* dan bersifat *open source* yang dapat digunakan untuk mempublikasikan informasi pada internet maupun intranet.

Ada beberapa fitur yang disediakan oleh Joomla untuk *end user* setelah instalasi, antara lain : *page caching*, *RSS feed*, halaman yang bisa dicetak, *blog*, polling, pencarian web site, dan mendukung banyak bahasa.

Sedangkan untuk *developer* dapat menambahkan beberapa *extension* dengan mudah pada Joomla. Hal ini dikarenakan Joomla didesain secara modular. Ada beberapa *extension* yang bisa ditambahkan antara lain : *plugin (mambots)*, *components*, *modules* dan *templates* [WHI06].

Plugin merupakan suatu komponen kecil yang dapat memberikan fungsionalitas tertentu yang biasanya digunakan untuk memanipulasi *content*. Contoh plugin pada Joomla adalah {mosimage}.

Components merupakan elemen utama Joomla yang ditampilkan pada halaman utama setiap template menu. Ada beberapa komponen bawaan Joomla yaitu *banners*, *contacts*, *news feeds*, *polls*, dan *web links*.

Modules adalah bagian dari *content* Joomla yang dapat diletakkan pada posisi yang berbeda-beda di dalam *template*. Ada beberapa *modules* bawaan Joomla, yaitu : *main menu*, *top menu*, *template chooser* dan lain-lain.

Templates merupakan bagian Joomla yang memberikan tampilan visual web site yang diletakkan terpisah dari *content* web site. Template yang biasanya terinstall langsung dengan Joomla adalah *madeyourweb* dan *rhuk_solaflare_ii*.

BAB III

METODOLOGI

Untuk pemodelan sistem, penelitian ini menggunakan bahasa pemodelan *Unified Modelling Language (UML)*. UML merupakan salah satu bahasa standar untuk pembuatan perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menspesifikkan, membangun dan mendokumentasikan perangkat lunak.[BOO98]

UML terdiri dari beberapa komponen yang berupa diagram. Diagram tersebut dibuat berdasarkan sudut pandang yang berbeda-beda terhadap sistem. Diagram-diagram UML yang digunakan untuk pemodelan sistem antara lain : *Activity Diagram, Use Case Diagram, Class Diagram* dan *Sequence Diagram.*

3.1 Analisis Kebutuhan

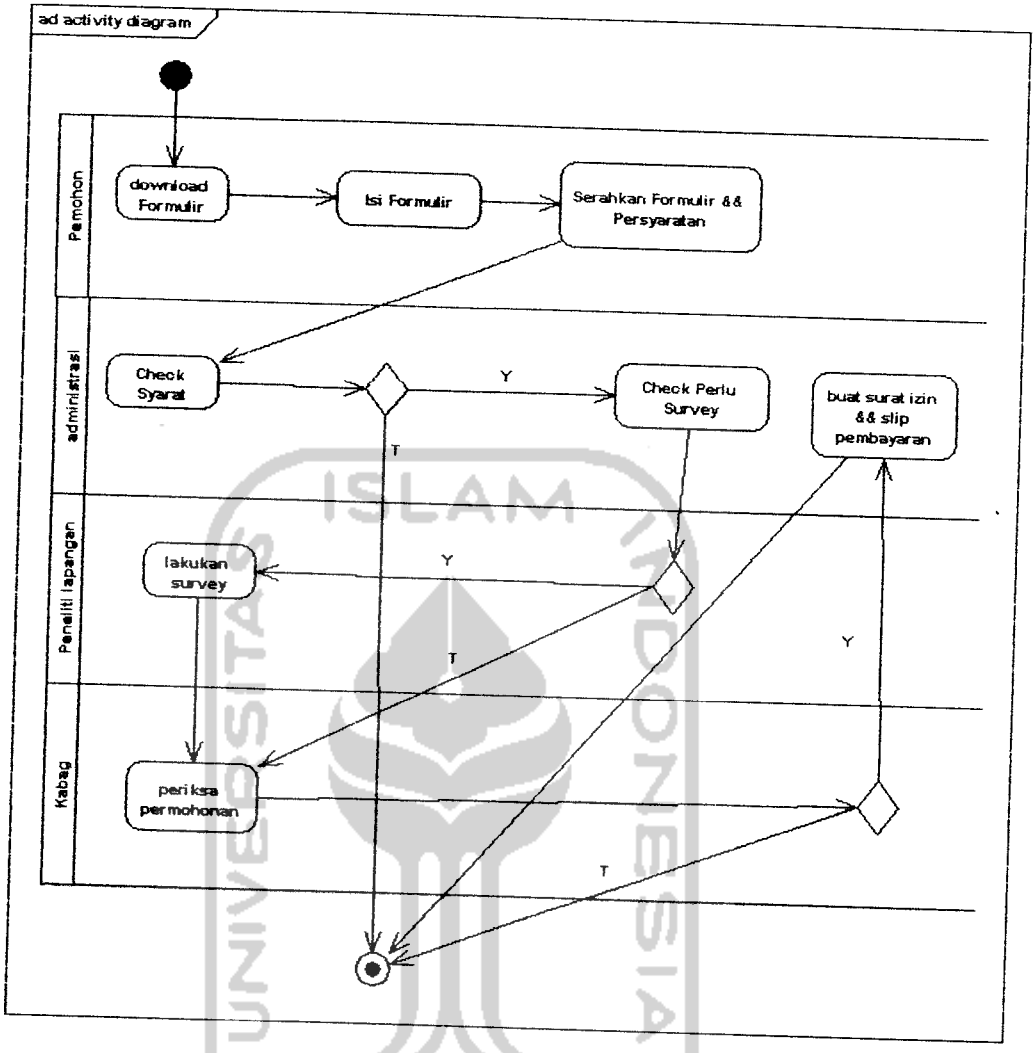
Analisis kebutuhan dilakukan dengan mengidentifikasi bisnis proses yang terjadi dalam SIMTAP. Pada tahap analisis, digunakan *activity diagram* untuk menggambarkan proses apa saja yang terjadi di dalam sistem.

Activity diagram merupakan salah satu diagram yang digunakan untuk menggambarkan alur kerja sistem. *Activity diagram* serupa dengan *flowchart* yang mendeskripsikan rangkaian perubahan dari satu aktivitas ke aktivitas yang lain[BOO98].

Karena menggambarkan *workflow* aktifitas, *activity diagram* sangat cocok untuk menggambarkan bisnis proses dari suatu sistem. Selain itu, *activity diagram* juga dapat menggambarkan aktivitas yang dilakukan oleh operator dalam hal ini manusia, maupun aktivitas yang dilakukan oleh mesin.

Ada beberapa hal yang dapat ditarik dari *activity diagram*, antara lain : algoritma bisnis proses, aktor yang menjadi pelaku dalam setiap proses dan objek-objek yang berinteraksi satu dengan yang lainnya.

Gambar 3.1 menunjukkan gambaran bisnis proses SIMTAP menggunakan *activity diagram* :



Gambar 3.1 *Activity diagram* bisnis proses SIMTAP

Pada SIMTAP terdapat 4 pengguna sistem, yaitu pemohon, bagian administrasi, peneliti lapangan dan kepala bagian. Pertama pemohon mengunduh blanko/formulir perizinan yang diperlukan. Kemudian mengisi formulir tersebut dan menyerahkan formulir yang disertai syarat-syarat permohonan pada bagian administrasi institusi tempat pelayanan permohonan.

Bagian administrasi melakukan pengecekan formulir dan kelengkapan persyaratan yang diperlukan. Jika syarat terpenuhi dan formulir telah diisi maka bagian administrasi melakukan pengecekan apakah permohonan memerlukan

survey atau tidak. Apabila permohonan yang diajukan memerlukan survey, selanjutnya bagian peneliti lapangan melakukan survey ke lokasi pemohon, tetapi apabila tidak memerlukan survey, selanjutnya bagian administrasi menyerahkan berkas permohonan kepada kepala bagian yang berhak memberikan surat izin permohonan.

Setelah melakukan survey ke lokasi, bagian peneliti lapangan kemudian menyerahkan hasil survey ke kepala bagian untuk dicek apakah permohonan diterima atau tidak. Kepala bagian yang menentukan suatu permohonan diterima atau tidak. Jika permohonan diterima, selanjutnya bagian administrasi membuat surat izin dan slip pembayaran untuk permohonan yang diajukan.

Setelah surat izin selesai, pemohon dapat mengambil perizinan yang diajukannya ke bagian administrasi instansi terkait.

Dari gambaran di atas, dapat diketahui masukan sistem, keluaran sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak serta antar muka sistem yang akan dibuat, sehingga sistem yang dibuat nantinya sesuai dengan apa yang diharapkan

3.1.1 Masukan Sistem

Masukan pada SIMTAP ini berupa data-data yang diperlukan oleh sistem. Data-data tersebut dapat dikelompokkan kedalam beberapa kategori, antara lain :

a. Data Referensi

   Data referensi berisi data-data induk yang menjadi referensi untuk data-data yang lain yang dibutuhkan oleh keseluruhan sistem. Data referensi dalam SIMTAP ini mencakup data Instansi pemerintahan, data jenis identitas penduduk, data kategori layanan SIMTAP, data jenis kewarganegaraan penduduk, data prosedur pengajuan permohonan, data syarat permohonan, data status izin, data status proses dan data kecamatan.

b. Data Master

   Data master merupakan data induk yang dibutuhkan sistem untuk pemrosesan dan pemberian informasi terhadap layanan yang diberikan.

Data master mencakup data jenis layanan, data pemohon, data pemberi izin.

c. Data Layanan

Data layanan digunakan untuk menjalankan proses pelayanan kepada pemohon. Data layanan meliputi : pembuatan surat izin, slip pembayaran, dan surat pemberitahuan survey.

3.1.2 Keluaran Sistem

Adapun keluaran SIMTAP berupa informasi kemajuan permohonan, informasi permohonan, formulir permohonan, hasil cetakan surat permohonan, hasil cetakan surat survey lapangan, dan hasil cetakan slip pembayaran.

3.1.3 Perangkat Lunak yang Dibutuhkan

SIMTAP dibuat menggunakan *framework* Joomla yang berbasis web. Oleh karena itu untuk menjalankan SIMTAP, ada beberapa perangkat lunak yang harus terinstall di komputer, antara lain :

1. Web Server Apache 1.3, atau versi yang lebih baru.
2. PHP 4.2.x atau versi yang lebih baru dengan library Mysql, XML, dan Zlib yang diaktifkan.
3. DBMS MySQL 4.2.x atau versi yang lebih baru yang mendukung *innodb storage engine*.

3.1.4 Perangkat Keras yang Dibutuhkan

Agar dapat berjalan dengan baik, komputer yang menjalankan SIMTAP ini harus memenuhi *system requirements* minimal yang dibutuhkan. Spesifikasi dari *system requirements* tersebut adalah :

1. Processor Pentium III atau setara dengan kecepatan 667 MHz
2. Memori (RAM) dengan kapasitas 256 MB
3. Ruangan kosong Harddisk sebesar 30 MB
4. Monitor VGA atau SVGA dengan resolusi minimum 800 x 600.

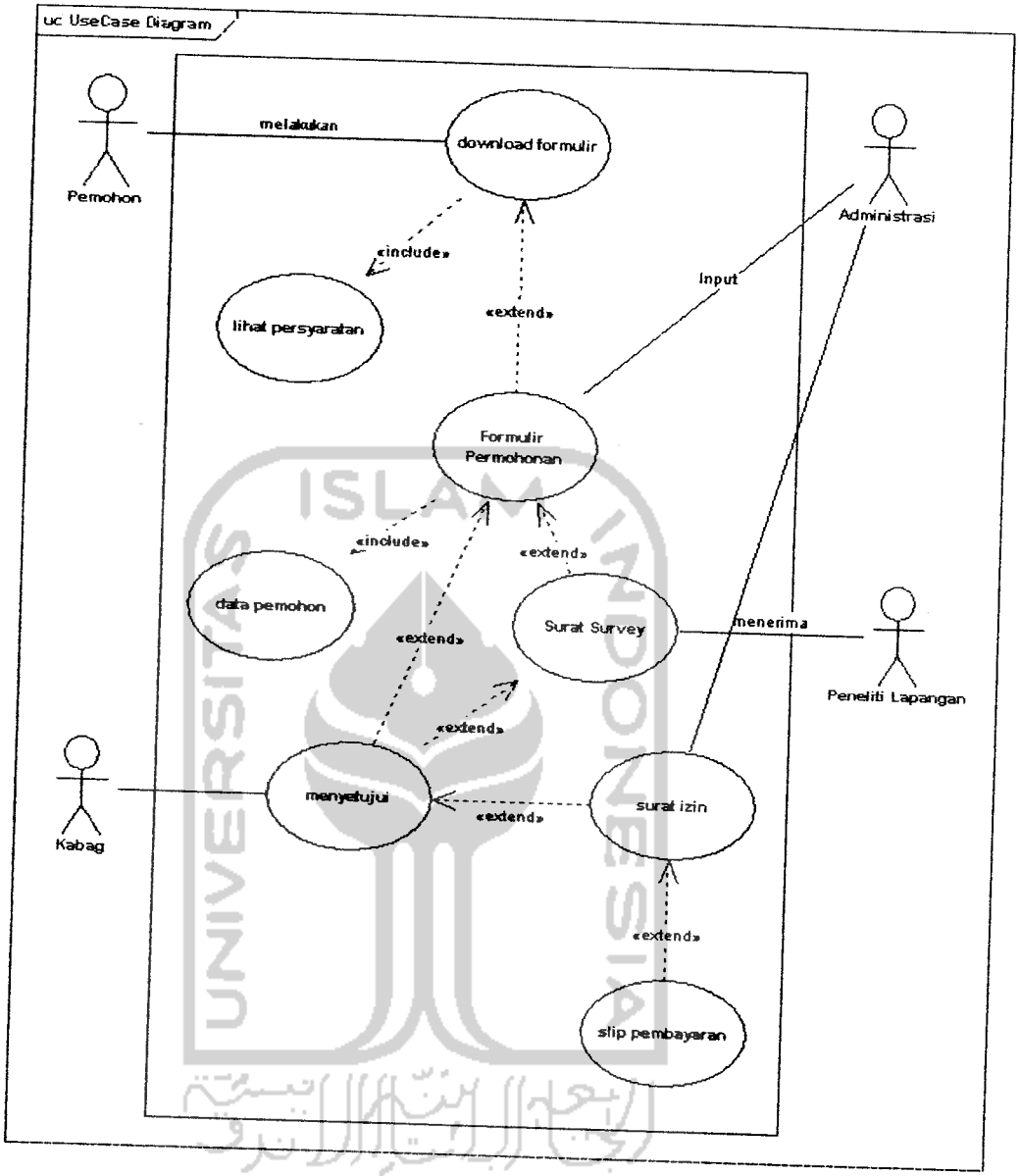
3.2 Desain Sistem

3.2.1 Use case Diagram

Dari analisis yang telah dilakukan di atas, selanjutnya dibuat pemodelan dari sudut pandang user menggunakan *use case diagram*.

Use case diagram merupakan diagram yang dibuat dengan menggunakan sudut pandang pengguna. Pada *use case diagram*, pengguna seolah-olah dilibatkan pada tahap analisis dan desain sistem. Dalam hal ini, *developer* berperan sebagai aktor yang akan menggunakan sistem. *Use case* diagram merupakan suatu konstruksi yang membantu *system analyst* untuk menentukan bagaimana kegunaan sistem. *Use case* diagram akan menggambarkan sistem dengan berpijak pada apa yang akan dilakukan user terhadap sistem. *Use case* diagram merupakan kumpulan skenario dalam sebuah sistem yang menggambarkan serangkaian kejadian pada sistem yang dilakukan oleh aktor pengguna sistem. [SCH04]

Use case diagram untuk pemberian pelayanan pada SIMTAP ditunjukkan pada gambar 3.2 :



Gambar 3. 2 *Use case diagram* pelayanan perizinan

Use case diagram di atas menggambarkan aktor-aktor dan aksi yang dilakukan oleh masing-masing aktor pada sistem. *Use case* masih bersifat umum dan belum menjelaskan transisi yang terjadi di dalam sistem. Oleh karena itu, use case diagram di atas perlu di*breakdown* lagi menjadi *sequence diagram*.

3.2.2 Sequence Diagram

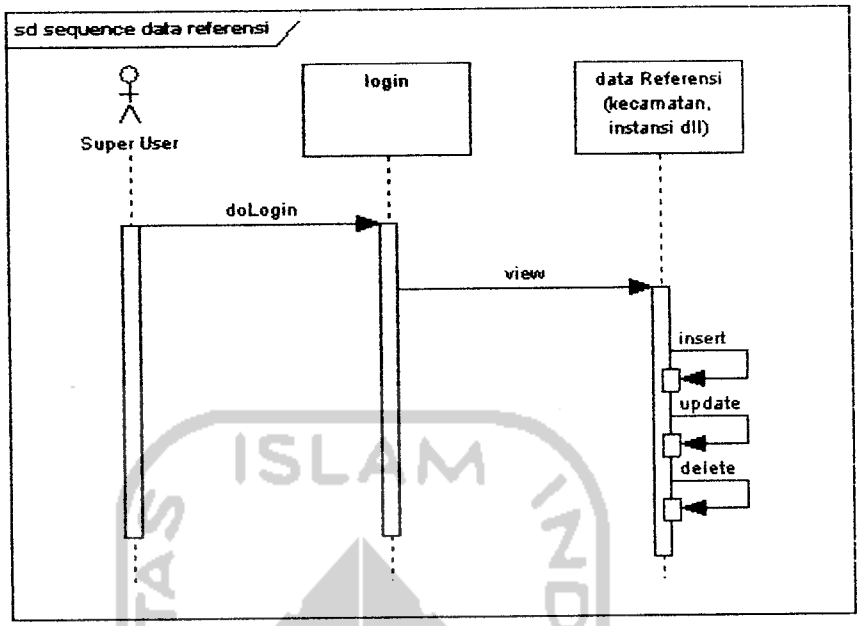
Sequence diagram digunakan untuk menggambarkan urut-urutan kondisi di mana elemen-elemen sistem saling berinteraksi antara satu dengan yang lain.[ALH03]

Sequence diagram pada SIMTAP merupakan bentuk penjabaran detail dari *use case diagram*. Pada *use case diagram*, menggambarkan tentang aktor dan aktivitas aktor tersebut. Pada *sequence* aktivitas tersebut dijabarkan dengan menggambarkan urutan waktu terjadinya aktivitas tersebut.

Use case diagram pada gambar III.2 di atas hanya menjelaskan aktivitas proses layanan sebagai proses utama dalam SIMTAP. Kenyataannya pada SIMTAP, selain bisnis proses utama, juga ada proses lain yang diperlukan oleh bisnis proses utama yaitu manajemen data referensi sistem. Dalam hal ini data tersebut dapat dikelompokkan ke dalam 2 jenis, data referensi dan data master.

Data referensi merupakan data pokok yang digunakan oleh semua instansi. Data referensi bersifat sangat umum. Hanya super user (administrator sistem) yang memiliki hak penuh dalam mengatur data referensi. Keberadaan data referensi menjadi suatu keharusan. Oleh karena itu, sebelum melangkah ke bisnis proses SIMTAP, data referensi harus terisi terlebih dahulu.

Sequence Diagram untuk data referensi digambarkan pada gambar 3.3 berikut :

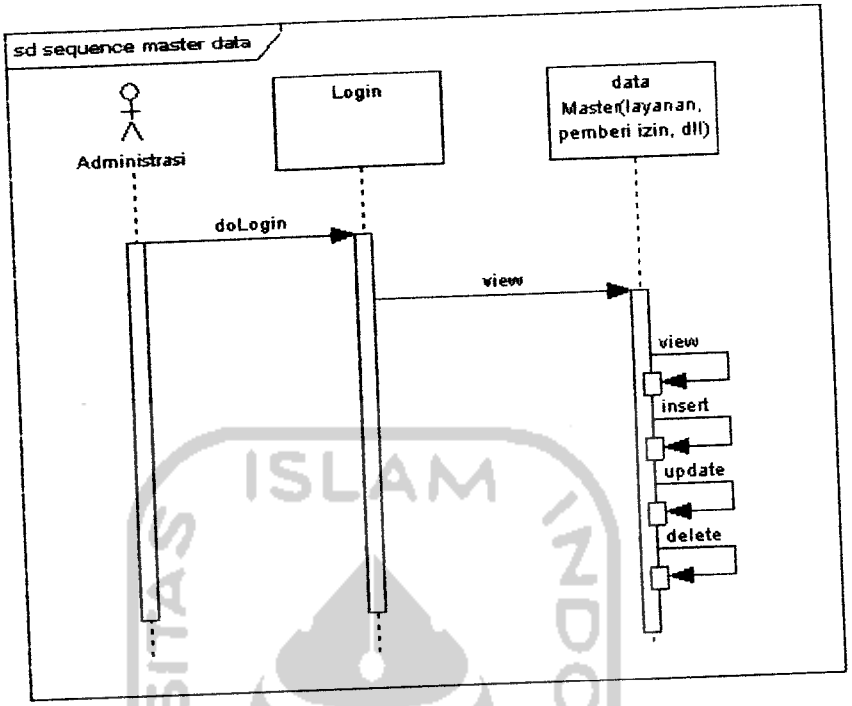


Gambar 3. 3 *Sequence diagram* data referensi

Super user/Administrator melakukan login, kemudian administrator menginputkan data referensi yang diperlukan. Selain itu, administrator juga dapat melakukan update dan menghapus data referensi.

Data master hampir sama dengan referensi, hanya saja berada pada level di bawah data referensi. Data master umumnya berbeda-beda antar instansi. Pengguna yang memiliki hak penuh terhadap data master adalah bagian administrasi instansi terkait.

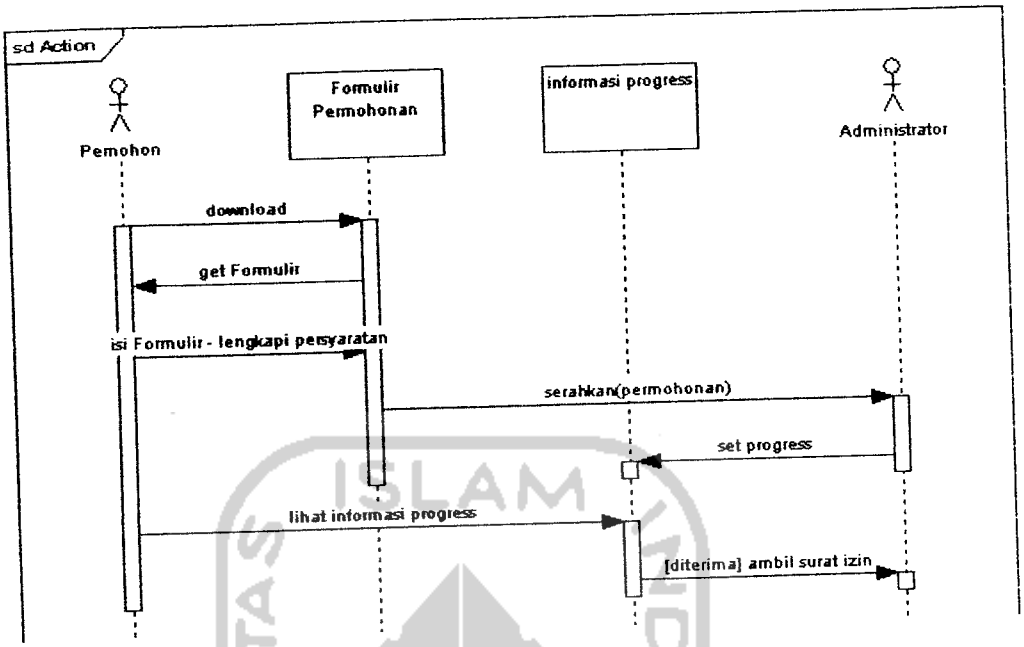
Sequence Diagram untuk data master digambarkan dengan gambar 3.4 berikut :



Gambar 3. 4 *Sequence diagram* data master

Setelah data master terbuat, maka pada halaman publik, formulir jenis layanan dapat diunduh oleh pemohon yang akan mengajukan permohonan perizinan.

Sequence diagram untuk pengajuan permohonan ditunjukkan pada gambar 3.5 berikut :

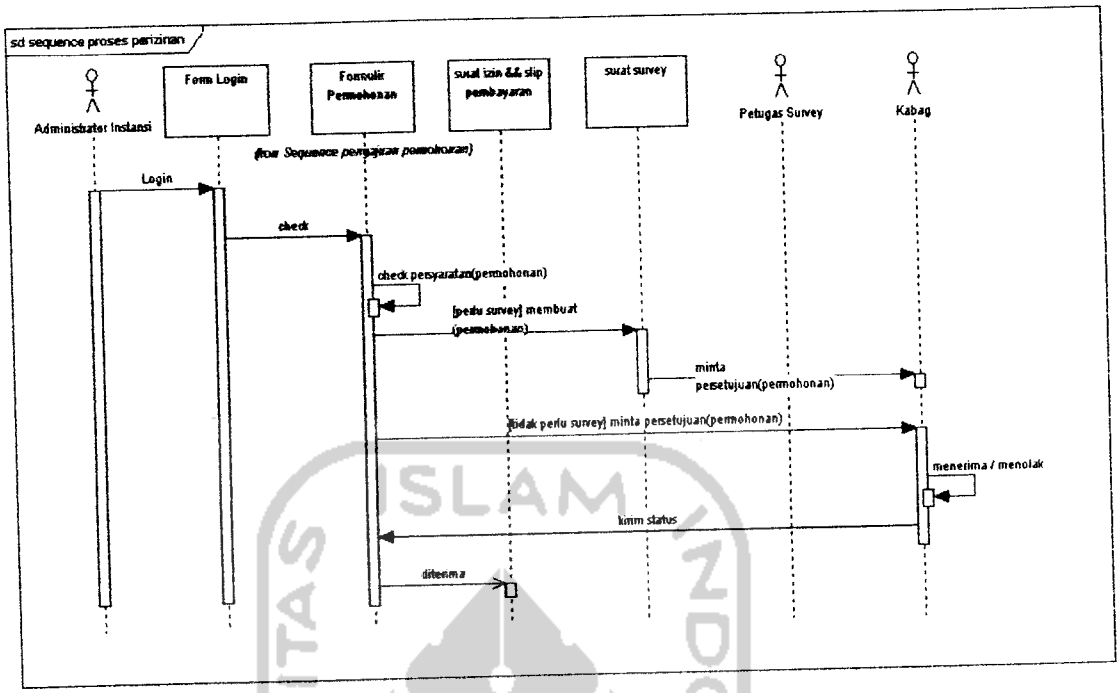


Gambar 3.5 *Sequence diagram* pengajuan permohonan

Pemohon mula-mula mendownload formulir permohonan dari web site SIMTAP. Formulir kemudian diisi dan diserahkan beserta persyaratan-persyaratan yang diperlukan ke bagian administrasi. Permohonan kemudian diproses di bagian administrasi. Setiap perubahan status proses perizinan, akan dimunculkan di halaman publik sehingga pemohon dapat melihat sejauh mana permohonan yang diajukannya diproses.

Apabila status permohonan diterima, pemohon kemudian mengambil surat izin dari bagian administrasi instansi yang memproses permohonannya.

Sequence diagram yang menggambarkan serangkaian langkah dalam memproses permohonan yang diajukan ditunjukkan pada gambar 3.6 berikut :



Gambar 3.6 *Sequence diagram* pemrosesan permohonan

Bagian administrasi instansi melakukan login ke SIMTAP. Kemudian melakukan pengecekan persyaratan permohonan yang diajukan. Apabila semua persyaratan telah terpenuhi, selanjutnya dilakukan pengecekan terhadap jenis permohonan untuk menentukan perlu/tidaknya survey dilakukan. Jika survey tidak perlu dilakukan, bagian administrasi kemudian meminta persetujuan kepada kabag atas permohonan yang diajukan, tetapi apabila survey harus dilakukan, bagian administrasi membuat surat pengantar survey untuk peneliti lapangan dan surat pemberitahuan survey untuk pemohon.

Petugas survey kemudian melakukan survey berdasarkan persetujuan kabag. Hasil survey kemudian diserahkan kembali kepada kabag untuk ditinjau apakah permohonan diterima atau ditolak. Kabag selanjutnya menentukan untuk menerima/menolak permohonan.

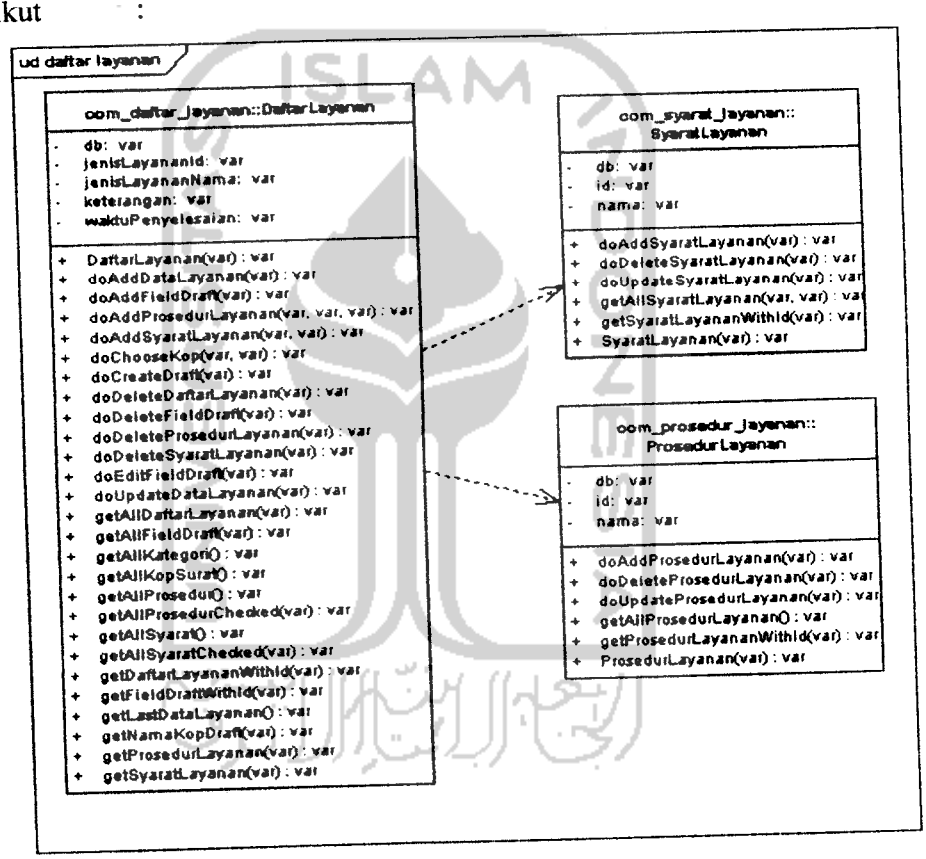
Bagian administrasi kemudian mengeset status permohonan terakhir, apakah diterima atau ditolak. Informasi status permohonan itulah yang dilihat oleh pemohon untuk mendapatkan surat perizinan atas permohonan yang diajukan.

3.2.3 Class Diagram

Class diagram digunakan untuk menggambarkan *domain* permasalahan sistem. Oleh karena itu, *class diagram* bersifat spesifik pada sistem yang dibangun[ALH03].

Untuk SIMTAP ada beberapa *class* yang dibuat untuk keperluan sistem. Masing-masing *class* memiliki hubungan dengan *class-class* yang lain.

Class diagram pada SIMTAP ditunjukkan oleh gambar 3.7 dan gambar 3.8 berikut :

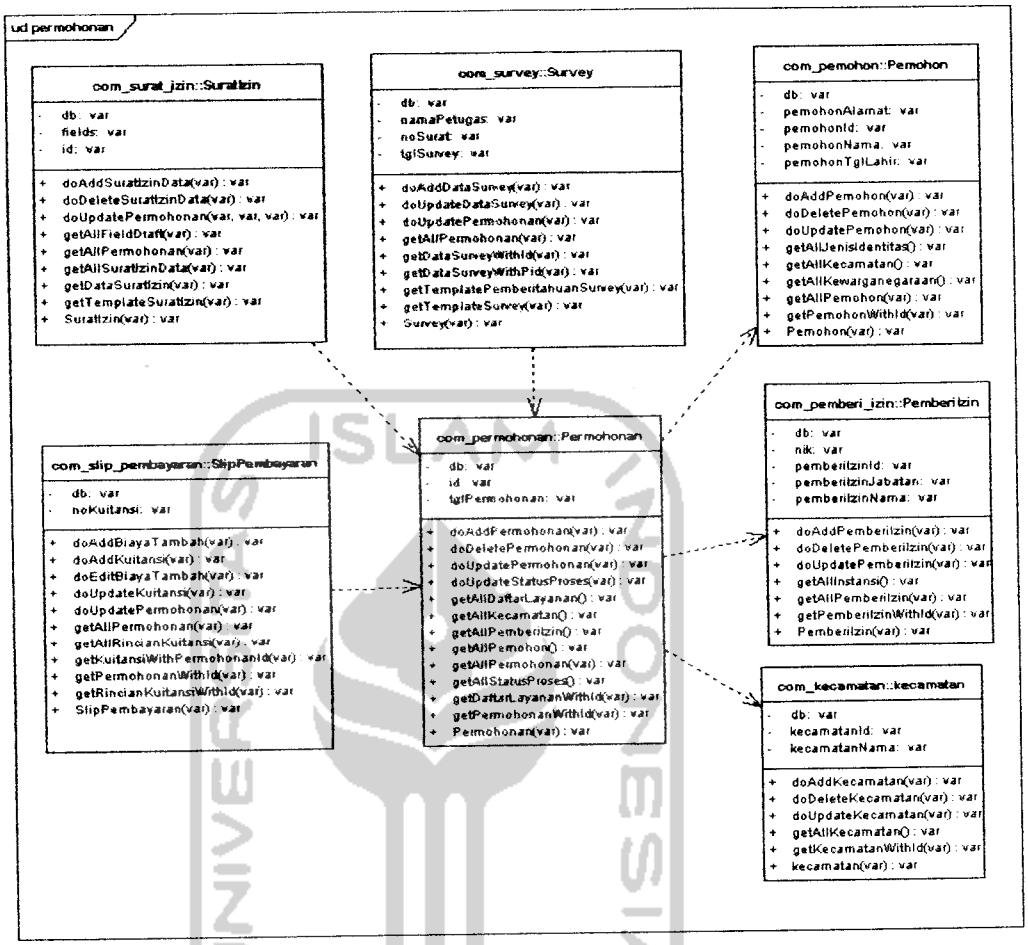


Gambar 3.7 ***Class diagram*** package daftar layanan

Gambar 3.7 di atas merupakan *package* daftar layanan yang terdiri dari *class* daftar layanan, syarat layanan dan prosedur layanan. *Class* daftar layanan *dependen* ke *class* syarat layanan dan *class* prosedur layanan.

Selain *package* daftar layanan, juga terdapat *package* permohonan yang

ditunjukkan pada gambar 3.8 berikut :



Gambar 3.8 *Class diagram* package permohonan

Gambar 3.8 di atas menggambarkan kumpulan *class* pada *package* permohonan. *Class* permohonan *dependency* ke *class* pemohon, *class* pemberiIzin dan *class* kecamatan. Sedangkan *class* slipPembayaran, *class* suratIzin dan *class* survey *dependency* pada *class* permohonan.

3.2.4 Tabel Relasional

Dari desain sistem di atas, selanjutnya dibuat tabel-tabel database relasional SIMTAP.

Tabel database relasional SIMTAP ditunjukkan pada gambar 3.9 berikut :

Tabel 3.1 Struktur tabel ref_kecamatan

Field	Type	Keterangan
refKecamatanId	smallint(6)	Primary Key
refKecamatanNama	Varchar(50)	(Null)

b. jos_ref_kewarganegaraan,

Berisi data kewarganegaraan penduduk, contoh data : WNI (Warga Negara Indonesia). Struktur tabel jos_ref_kewarganegaraan ditunjukkan pada tabel 3.2 berikut :

Tabel 3.2 Struktur tabel jos_ref_kewarganegaraan

Field	Type	Keterangan
refKewarganegaraanId	smallint(6)	Primary Key
refKewarganegaraanNama	Varchar(75)	(Null)

c. jos_ref_status_proses

Berisi data status proses untuk menjelaskan progress dari permohonan, contoh data : sedang diproses dibagian survey lapangan. struktur tabel jos_ref_status_proses ditunjukkan pada tabel berikut :

Tabel 3.3 Struktur tabel jos_ref_status_proses

Field	Type	Keterangan
refStatusProsesId	smallint(6)	Primary Key
refStatusProsesNama	Varchar(50)	(Null)

d. jos_ref_status_izin

Berisi data status permohonan yang diajukan, contoh data : diterima, ditolak. Struktur tabel jos_ref_status_izin ditunjukkan pada tabel berikut :

Tabel 3.4 Struktur tabel jos_ref_status_izin

Field	Type	Keterangan
refStatusIzinId	smallint(6)	Primary Key
refStatusIzin	Varchar(50)	(Null)

e. jos_ref_prosedur_layanan

Berisi data prosedur pengajuan permohonan, contoh data : mendownload formulir permohonan. Struktur tabel jos_ref_prosedur_layanan ditunjukkan pada tabel berikut :

Tabel 3.5 Struktur tabel jos_ref_prosedur_layanan

Field	Type	Keterangan
refProsedurLayananId	smallint(6)	Primary Key
refProsedurLayananNama	Varchar(50)	(Null)

f. jos_ref_jenis_identitas

Berisi data jenis identitas yang digunakan pemohon, contoh data : SIM, KTP. Struktur tabel jos_ref_jenis_identitasditunjukkan pada tabel berikut

Tabel 3.6 Struktur tabel jos_ref_jenis_identitas

Field	Type	Keterangan
refJenisIdentitasId	smallint(6)	Primary Key
refJenisIdentitasNama	Varchar(50)	(Null)

g. jos_ref_kategori_layanan

Berisi data kategori layanan yang diberikan, contoh data : Industri, Umum. Struktur tabel jos_ref_kategori_layanan ditunjukkan pada tabel berikut :

Tabel 3.7 Struktur tabel jos_ref_kategori _layanan

Field	Type	Keterangan
refJenisIdentitasId	smallint(6)	Primary Key
refJenisIdentitasNama	Varchar(50)	(Null)

h. jos_ref_instansi

Berisi data instansi yang ada pada SIMTAP, contoh data : dinas kependudukan. Struktur tabel jos_ref_instansi ditunjukkan pada tabel berikut :

Tabel 3.8 Struktur tabel jos_ref_instansi

Field	Type	Keterangan
refInstansiId	smallint(6)	Primary Key
refInstansiNama	Varchar(50)	(Null)
refInstansiKeterangan	varchar(255)	(Null)

i. jos_ref_kop_surat

Berisi data kop surat yang digunakan untuk pelayanan perizinan. Data kop mencakup nama kop dan format kop yang digunakan. Format kop dibuat dalam bentuk html. Struktur tabel jos_ref_kop_surat ditunjukkan pada tabel berikut :

Tabel 3.9 Struktur tabel jos_ref_kop_surat

Field	Type	Keterangan
refKopId	smallint(6)	Primary Key
refKopNamaKop	Varchar(50)	(Null)
refKopKeterangan	varchar(255)	(Null)
refKopIsi	Text	(Null)

j. jos_ref_syarat_layanan

Berisi data syarat permohonan yang diajukan. Contoh data : fotocopy KTP. Struktur tabel jos_ref_syarat_layanan ditunjukkan pada tabel berikut :

Tabel 3.10 Struktur tabel jos_ref_syarat_layanan

Field	Type	Keterangan
refSyaratLayananId	smallint(6)	Primary Key
refSyaratLayananNama	Varchar(50)	(Null)

2. Data Master

Berikut tabel-tabel yang berisi data master :

a. jos_jenis_layanan

Berisi data jenis layanan yang diberikan oleh SIMTAP per instansi. Tabel ini memuat data : nama layanan, waktu penyelesaian, formulir, keterangan layanan, kategori layanan, satuan berlaku, lama berlaku, kop, format nomor surat, instansi, draft layanan, tarif. Struktur tabel jos_jenis_layanan ditunjukkan pada tabel berikut :

Tabel 3.11 Struktur tabel jos_jenis_layanan

Field	Type	Keterangan
jenisLayananId	smallint(6)	Primary Key
jenisLayananNama	Varchar(50)	(Null)
jenisLayananWaktuPenyelesaian	smallint(6)	
jenisLayananFormulirPathFile	varchar(255)	
jenisLayananKeterangan	varchar(255)	
jenisLayananRefKategoriLayananId	int(6)	Foreign key
jenisLayananSatuanLamaBerlaku	enum('Hari','Bulan','Tahun')	
jenisLayananWaktuLamaBerlaku	int(4)	
jenisLayananRefKopId	smallint(6)	Foreign key

jenisLayananFormatNomorSurat	varchar(20)	
jenisLayananRefInstansiId	smallint(6)	Foreign key
jenisLayananStringTemplate	Text	
jenisLayananTarif	decimal(10,0)	

b. jos_ref_pemberi_izin.

Berisi data pemberi izin untuk setiap permohonan yang diberikan. Tabel ini memuat data nama, nik , jabatan, instansi pemberi izin. Struktur tabel jos_ref_pemberi_izin ditunjukkan pada tabel berikut :

Tabel 3.12 Struktur tabel jos_ref_pemberi_izin

Field	Type	Keterangan
refPemberiIzinId	smallint(6)	Primary Key
refPemberiIzinNama	Varchar(50)	(Null)
refPemberiIzinNik	varchar(15)	(Null)
refPemberiIzinJabatan	varchar(255)	
refPemberiIzinRefInstansiId	smallint(6)	Foreign Key

c. jos_survey

Berisi data template untuk pembuatan surat pengantar dan surat pemberitahuan survey. Template dibuat dengan format HTML. Struktur tabel jos_survey ditunjukkan pada tabel berikut :

Tabel 3.13 Struktur tabel jos_survey

Field	Type	Keterangan
surveyId	smallint(6)	Primary Key
surveyNama	Varchar(50)	(Null)
surveyTemplate	varchar(15)	(Null)
pemberitahuanTemplate	varchar(255)	

3. Data Permohonan

Berikut tabel-tabel untuk data permohonan :

a. jos_pemohon

Berisi data pemohon yang mengajukan permohonan. Tabel ini memuat data : Nama, Jenis Identitas, Nomor Identitas, Kewarganegaraan Alamat, Kecamatan, Nomor Telepon, Nomor HP. Struktur tabel jos_pemohon ditunjukkan pada tabel berikut :

Tabel 3.14 Struktur tabel jos_pemohon

Field	Type	Keterangan
pemohonId	smallint(6)	Primary Key
pemohonNama	varchar(50)	(Null)
pemohonRefJenisIdentitasId	smallint(6)	Foreign Key
pemohanNomorIdentitas	varchar(50)	
pemohonTglLahir	date	
pemohonRefKewarganegaraanId	smallint(6)	Foreign key
pemohonAlamat	varchar(255)	
pemohonRefKecamatanId	smallint(6)	Foreign key
pemohonNomorTelepon	varchar(20)	
pemohonNomorHP	varchar(20)	

b. jos_permohonan

Berisi data permohonan perizinan. Tabel ini memuat data : Pemohon, Alamat Usaha, Kecamatan lokasi usaha, Tanggal Permohonan, Tanggal Perizinan, Tanggal Akhir Perizinan, Status Proses, Status Izin, Jenis Izin, perlu/tidaknya Survey, Nomor Surat Izin, Pemberi Izin, Jumlah Cetak Surat Izin, Tanggal Diserahkan. Struktur tabel jos_pemohonan ditunjukkan pada tabel berikut :

Tabel 3.15 Struktur tabel jos_permohonan

Field	Type	Keterangan
permohonanId	bigint(20)	Primary Key
permohonanPemohonId	bigint(20)	Foreign Key
permohonanAlamat	varchar(255)	
permohonanRefKecamatanId	smallint(6)	Foreign key
permohonanTglPermohonan	date	
permohonanTglDiberiIzin	date	
permohonanTglAkhirIzin	date	
permohonanRefStatusProsesId	smallint(6)	Foreign key
permohonanRefStatusIzinId	smallint(6)	Foreign key
permohonanJenisIzinId	smallint(6)	Foreign Key
permohonanIsPerluSurvey	enum('0','1')	
permohonanNomorSurat	varchar(100)	
permohonanRefPemberiIzinId	smallint(6)	Foreign Key
permohonanJumlahCetakSuratIzin	smallint(6)	
permohonanTglDiserahkan	date	
permohonanSurveyId	Tinyint(4)	Foreign Key

c. *jos_kuitansi*

Berisi data kuitansi dan slip pembayaran. Tabel ini memuat data : Nomor Kuitansi, Permohonan, Operator, Tanggal pembuatan, Tanggal penyerahan. Struktur tabel jos_kuitansi ditunjukkan pada tabel berikut :

Tabel 3.16 Struktur tabel jos_kuitansi

Field	Type	Keterangan
kuitansiId	bigint(20)	Primary Key
kuitansiNomor	varchar(50)	
kuitansiPermohonanId	bigint(20)	Foreign key
kuitansiUserId	smallint(6)	Foreign key

kuitansiTanggalBuat	date	
kuitansiTanggalDiserahkan	date	

d. jos_kuitansi_rincian

Berisi data rincian biaya untuk suatu permohonan. Tabel ini memuat data Kuitansi, Nama Rincian Biaya, Besar Rincian Biaya. Struktur tabel jos_kuitansi_rincian ditunjukkan pada tabel berikut :

Tabel 3.17 Struktur tabel jos_kuitansi_rincian

Field	Type	Keterangan
rinKuitansiId	bigint(20)	Primary key
rinKuitansiKuitansiId	bigint(20)	Foreign key
rinKuitansiNamaRincian	varchar(50)	Foreign key
rinKuitansiBesarRincian	decimal(20,2)	Foreign key

e. jos_surat_izin

Berisi data surat izin dan field-field apa yang diperlukan oleh suatu surat izin. Tabel ini berisi data : jenis layanan, Field, Nomor urut field. Struktur tabel jos_surat_izin ditunjukkan pada tabel berikut :

Tabel 3.18 Struktur tabel jos_surat_izin

Field	Type	Keterangan
suratIzinId	bigint(20)	Primary key
suratIzinJenisLayananId	smallint(6)	Foreign key
suratIzinField	varchar(50)	
suratIzinFieldNomorUrut	int(4)	

f. jos_surat_izin_data

Merupakan detail dari tabel jos_surat_izin. Tabel ini berisi data yang diperlukan pada suatu layanan perizinan. Tabel ini memuat data : surat

izin, permohonan, data field surat izin. Struktur tabel jos_surat_izin_data ditunjukkan pada tabel berikut :

Tabel 3.19 Struktur tabel jos_surat_izin_data

Field	Type	Keterangan
suratIzinDataId	bigint(20)	Primary key
suratIzinDataSuratIzinId	bigint(20)	
suratIzinDataPermohonanId	bigint(20)	Foreign key
suratIzinDataValue	varchar(50)	

g. jos_survey_data

Merupakan detail dari tabel jos_survey. Tabel ini berisi data yang diperlukan untuk pembuatan surat survey. Tabel ini memuat : Nama Petugas, Nomor Surat, Tanggal Survey, Permohonan. Struktur tabel jos_survey_data ditunjukkan pada tabel berikut :

Tabel 3.20 Struktur tabel jos_survey_data

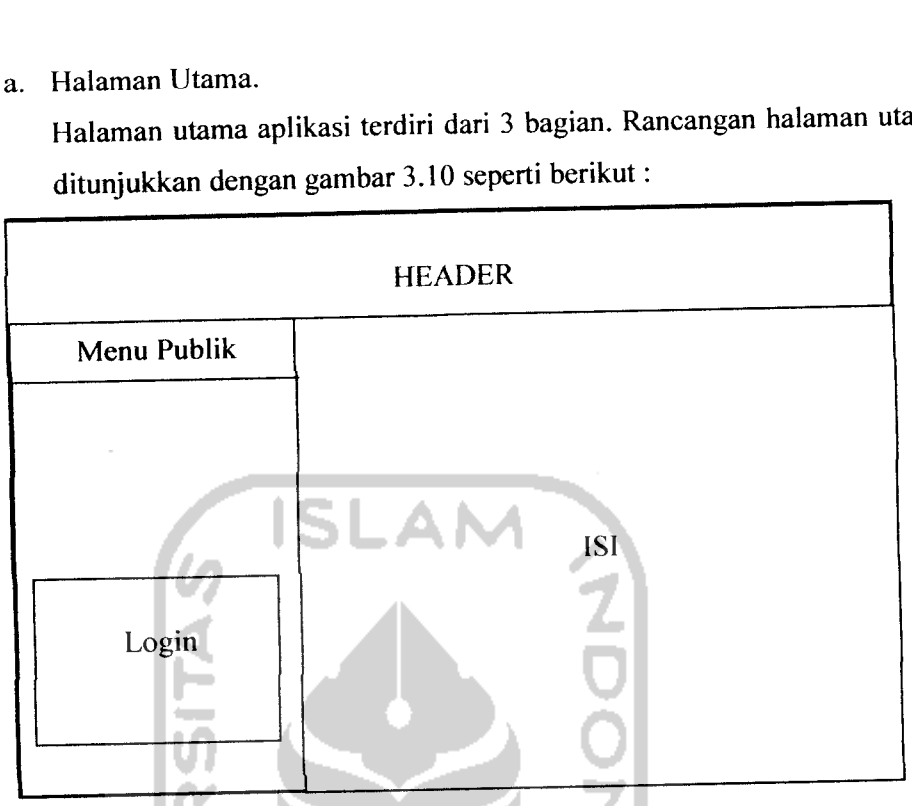
Field	Type	Keterangan
surveyDataId	bigint(20)	Primary key
surveyDataNamaPetugas	varchar(50)	
surveyDataNoSurat	varchar(50)	
surveyDataTanggalSurvey	date	
surveyDataPermohonanId	bigint(20)	Foreign key

3.2.5 Rancangan Antar Muka

Antarmuka merupakan media interaksi antara pengguna dengan aplikasi. Melalui tampilan antar muka pengguna dapat dengan mudah menjalankan aplikasi dan mendapatkan hasil pemrosesan aplikasi.

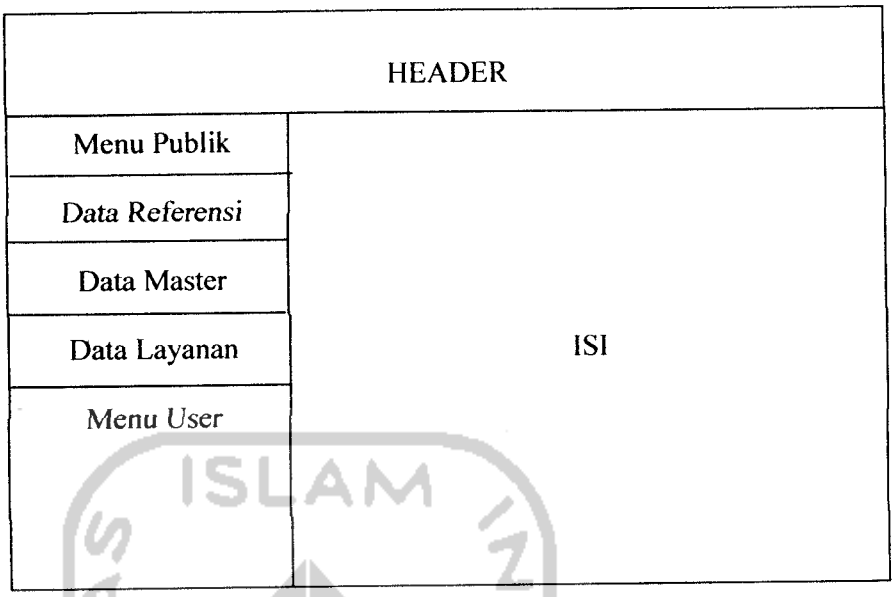
a. Halaman Utama.

Halaman utama aplikasi terdiri dari 3 bagian. Rancangan halaman utama ditunjukkan dengan gambar 3.10 seperti berikut :



Gambar 3.10 Rancangan antar muka halaman utama SIMTAP

Halaman utama di atas diperuntukkan untuk pengguna umum yang tidak memiliki account ke *backoffice* SIMTAP. Sedangkan untuk pengguna yang memiliki account ke *backoffice*, memiliki halaman yang berbedan dari gambar 3.10 diatas. Rancangan antar muka halaman utama setelah login ditunjukkan dengan gambar 3.10 seperti berikut :

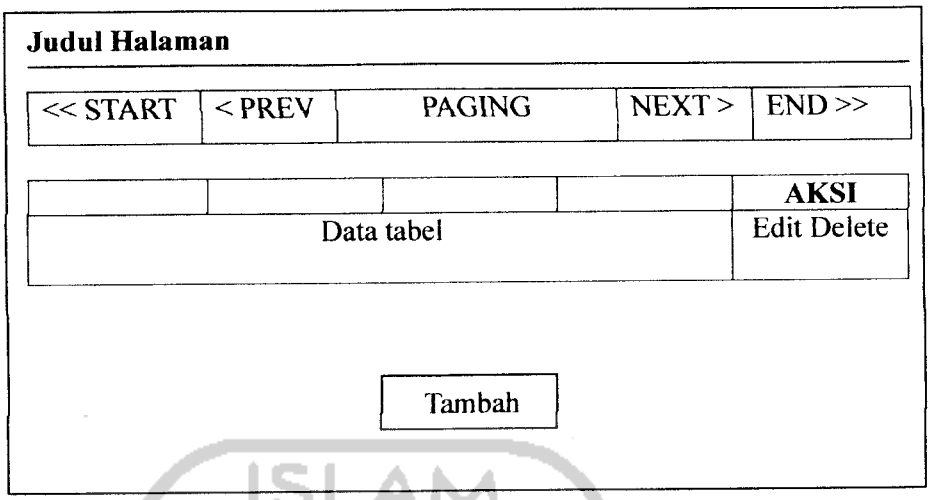


Gambar 3.11 Rancangan antar muka backoffice pasca login

Semua pengguna yang sudah login dapat melihat menu dan isi dari menu yang ditampilkan. Hanya saja masing-masing pengguna memiliki hak akses yang berbeda terhadap isi dari menu sesuai dengan group pengguna tersebut.

b. Tampilan Pertama Isi Menu

Tampilan pertama isi menu merupakan tampilan yang akan keluar setelah suatu menu diklik. Rancangan antar muka tampilan pertama isi menu ditunjukkan pada gambar 3.12 berikut :

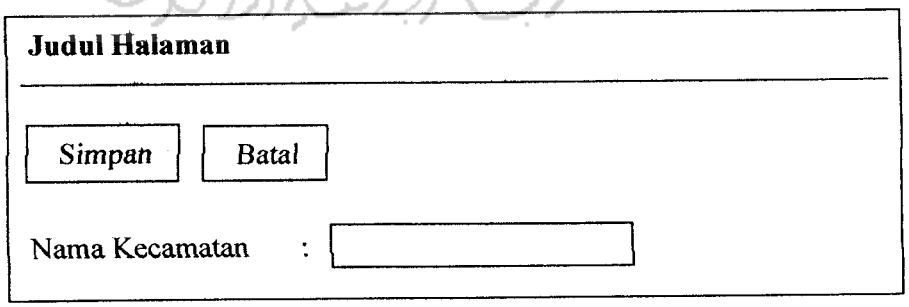


Gambar 3.12 Rancangan antar muka halaman *view content*

Tampilan pertama terdiri dari 3 bagian, *paging* dan daftar data. *Paging* akan muncul apabila data yang didapatkan lebih dari 20 *record*. Tombol tambah, edit dan delete hanya muncul untuk pengguna yang memiliki akses penuh terhadap informasi yang ditampilkan.

c. Halaman tambah data

Halaman tambah data berupa form untuk memasukkan data pada sistem. Ada 2 aksi yang dapat dilakukan pada halaman ini, yaitu simpan dan batal. Rancangan antar muka halaman tambah data ditunjukkan pada gambar 3.13 berikut :



Gambar 3.13 Rancangan antar muka halaman tambah data

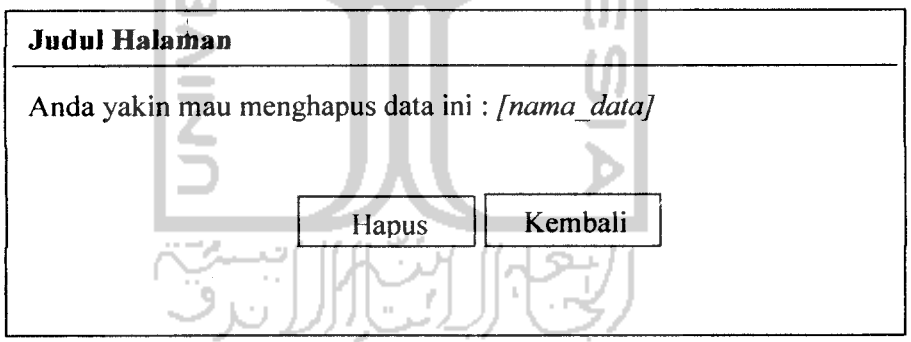
Rancangan antar muka halaman tambah seperti di atas diterapkan untuk data referensi kecamatan, data instansi, data syarat layanan, dan data prosedur layanan.

d. Halaman edit data

Halaman edit data berupa form untuk melakukan proses pengeditan data. Rancangan antar muka edit data pada dasarnya sama dengan antar muka tambah data. Hanya saja pada edit data terdapat proses pengambilan data dengan id yang dijadikan parameter.

e. Halaman konfirmasi penghapusan

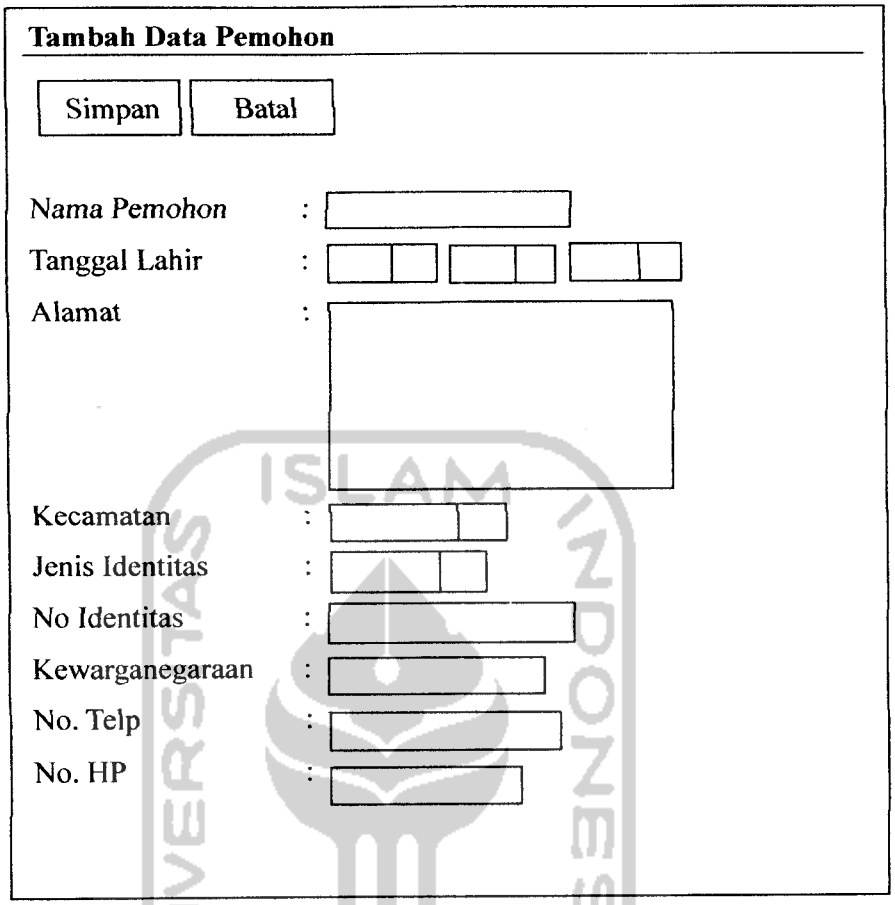
Setiap proses penghapusan data, akan memunculkan konfirmasi untuk meyakinkan apakah pengguna benar-benar yakin akan menghapus data tersebut. Rancangan antar muka halaman konfirmasi penghapusan data ditunjukkan pada gambar 3.14 berikut :



Gambar 3.14 Rancangan antar muka halaman konfirmasi penghapusan

f. Halaman tambah pemohon

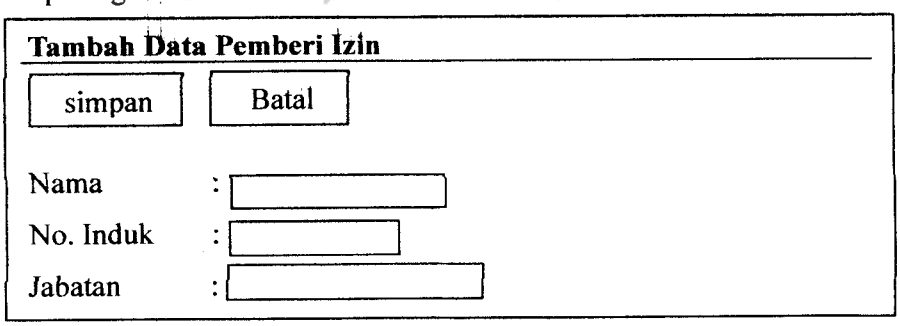
Halaman tambah pemohon berupa form untuk penambahan pemohon. Rancangan antar muka halaman tambah pemohon ditunjukkan pada gambar 3.15 berikut :



Gambar 3.15 Rancangan antar muka halaman tambah pemohon

g. Halaman tambah pemberi izin

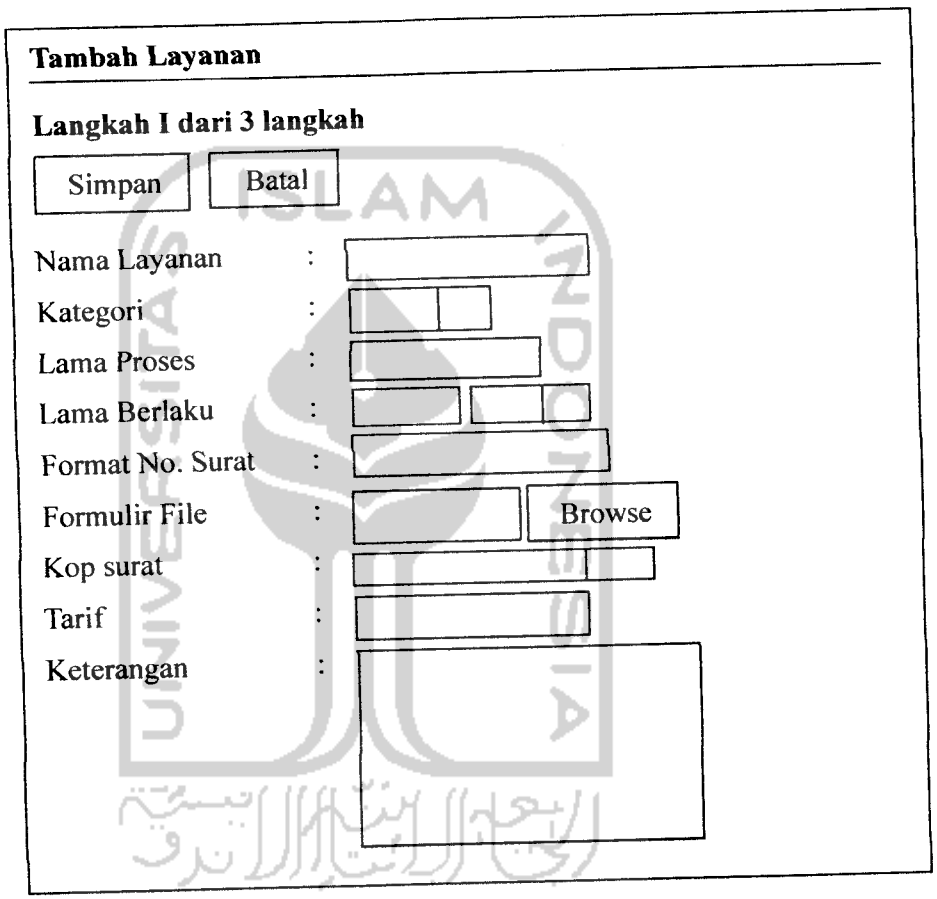
Halaman tambah pemberi izin berupa form untuk penambahan pemberi izin. Rancangan antar muka halaman tambah pemberi izin ditunjukkan pada gambar 3.16 berikut :



Gambar 3.16 Rancangan antar muka halaman tambah pemberi izin

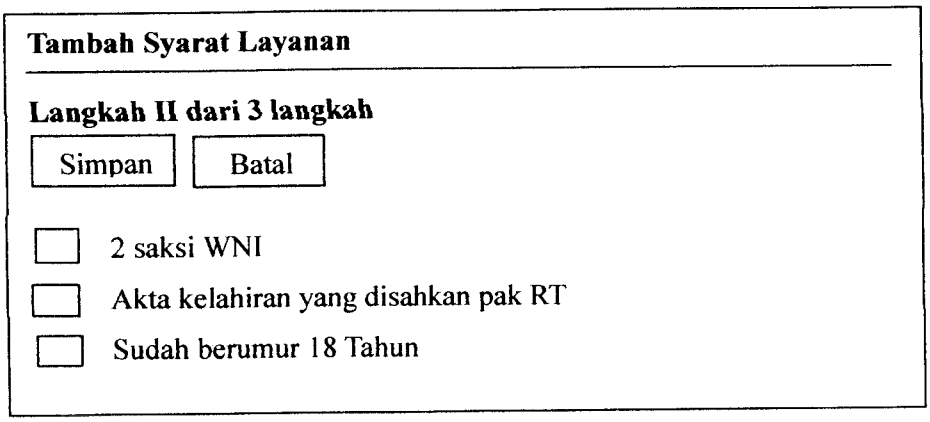
h. Halaman tambah jenis layanan

Halaman tambah jenis layanan berupa form untuk penambahan jenis layanan. Halaman ini berupa *wizard* dengan tiga form input. Rancangan antar muka langkah pertama tambah jenis layanan ditunjukkan pada gambar 3.17 berikut :



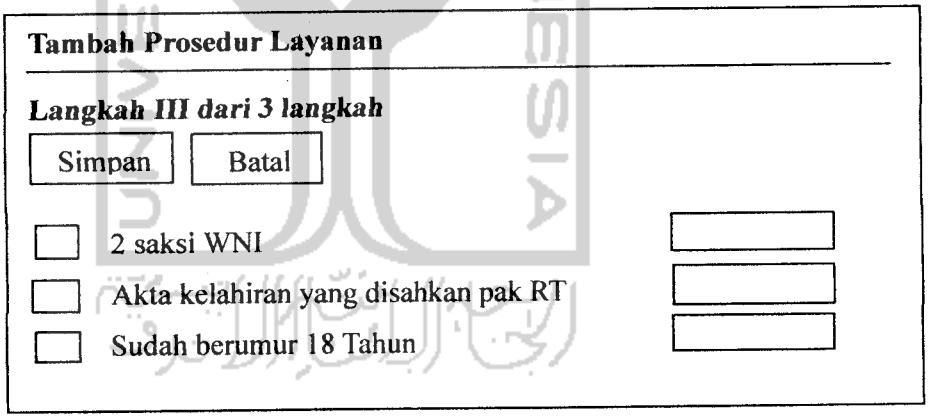
Gambar 3.17 Rancangan antar muka halaman tambah layanan

Langkah kedua digunakan untuk memasukkan syarat layanan pada jenis layanan. Rancangan antar muka langkah kedua tambah jenis layanan ditunjukkan pada gambar 3.18 berikut :



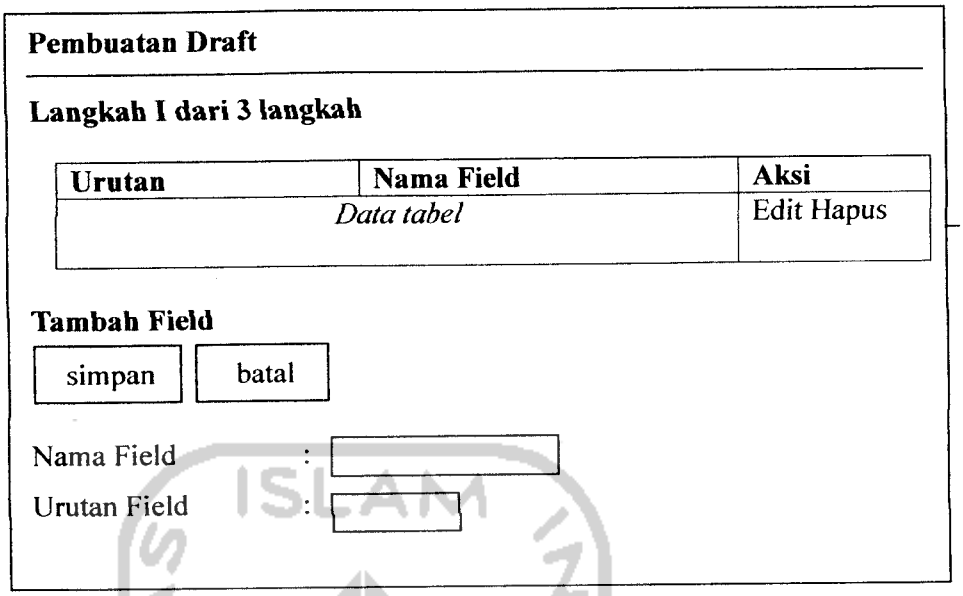
Gambar 3.18 Rancangan antar muka halaman tambah syarat layanan

Langkah ketiga digunakan untuk memasukkan data prosedur layanan yang disertai dengan data urutan pelaksanaan prosedur layanan tersebut. Rancangan antar muka langkah ketiga tambah jenis layanan ditunjukkan pada gambar 3.19 berikut :



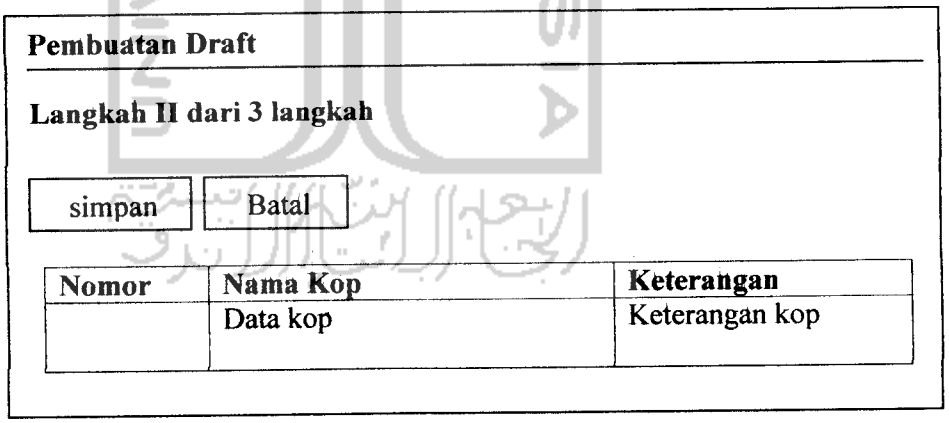
Gambar 3.19 Rancangan antar muka halaman tambah prosedur layanan

Aksi lain yang dilakukan pada saat penambahan layanan adalah pembuatan draft surat izin layanan. Pembuatan draft juga menggunakan *wizard*, Rancangan antar muka langkah pertama pembuatan draft surat izin ditunjukkan pada gambar 3.20 berikut :



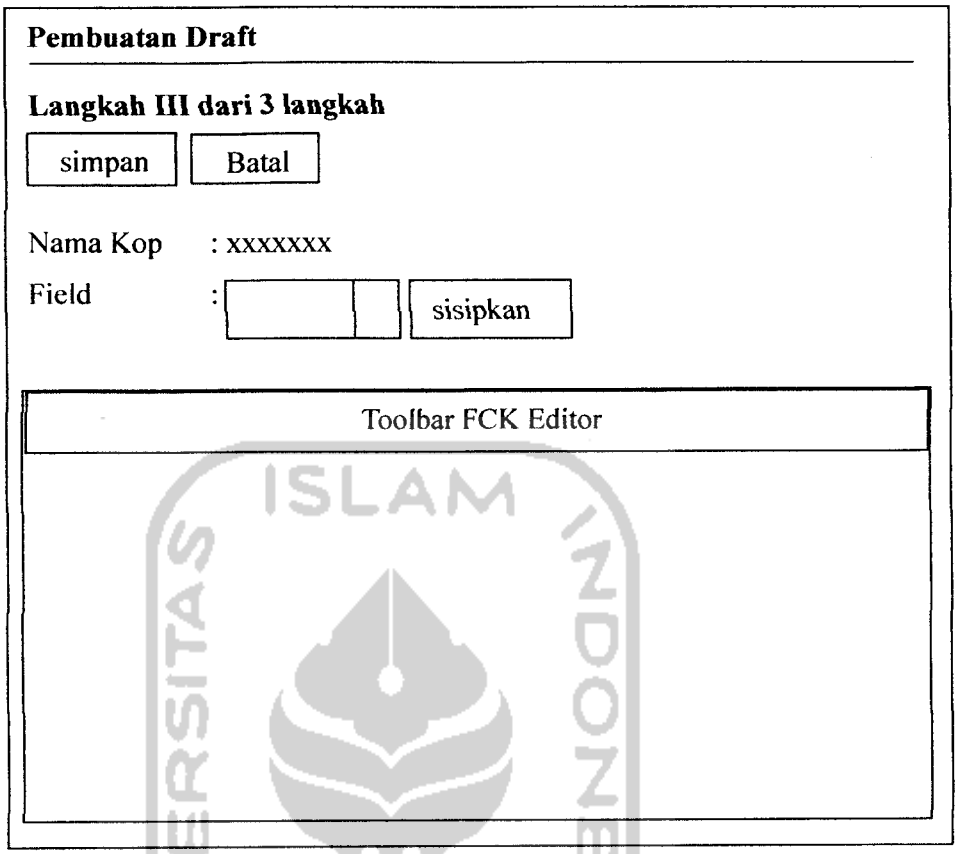
Gambar 3.20 Rancangan antar muka langkah I pembuatan draft

Langkah kedua digunakan untuk pemilihan kop surat izin, Rancangan antar muka langkah kedua pembuatan draft surat izin ditunjukkan pada gambar 3.21 berikut :



Gambar 3.21 Rancangan antar muka langkah II pembuatan draft

Langkah ketiga digunakan untuk membuat template draft surat izin, Rancangan antar muka langkah ketiga pembuatan draft surat izin ditunjukkan pada gambar 3.22 berikut :



Gambar 3.22 Rancangan antar muka langkah III pembuatan draft

Pembuatan template menggunakan tool FCK Editor agar template dapat diformat dengan menggunakan HTML.

3.3 Implementasi

3.3.1 Perangkat Keras

Perangkat keras yang digunakan pada saat implementasi program adalah:

1. Komputer dengan prosesor Intel Pentium Centrino 2.3 GHz
2. RAM 512 MB
3. Hardisk 40 GB

3.3.2 Perangkat Lunak

Perangkat lunak yang digunakan pada saat implementasi program adalah:

1. Linux ubuntu Feisty Fawn
2. Xampp 1.6.3 (apache 2.0.55, php 5.2.3, mysql 5)

3.3.3 Implementasi Sistem

SIMTAP dibuat dengan melakukan kustomisasi Joomla. Kustomisasi dilakukan dengan menambahkan beberapa komponen dan modul untuk keperluan SIMTAP. Selain itu untuk pengaturan hak akses data pada *core* aplikasi Joomla juga ditambahkan satu alur bisnis proses untuk memasukkan pengguna pada instansi tertentu. Hal ini perlu dilakukan untuk membatasi hak akses pengguna hanya pada instansi pengguna tersebut berada.

3.3.3.1 Model-View-Controller SIMTAP

Komponen dan modul dibuat dengan menggunakan *design pattern* MVC (*Model-View-Controller*). Setiap komponen yang dibuat masing-masing memiliki data model, *controller*, dan tampilan yang dipisahkan.

Berikut daftar modul yang dibuat :

a. Modul menu data referensi

   Modul ini menampilkan menu-menu data referensi.

b. Modul menu master data

   Modul ini menampilkan menu-menu master data.

c. Modul menu data layanan

   Modul ini menampilkan menu-menu data layanan.

Modul-modul tersebut dibuat sebagai ekstensi joomla yang diinstall oleh super administrator melalui fasilitas yang disediakan oleh joomla. Sebuah modul joomla terdiri dari 2 file, yaitu file modul yang bertipe php dan file installasi modul yang bertipe xml. Agar bisa dijadikan sebagai ekstensi joomla, kedua file tersebut harus di jadikan sebagai *archive* dengan tipe file zip.

Berikut contoh skrip file installasi untuk modul menu master data :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<mosinstall type="module" version="4.5.2">
  <name>Master Data</name>
  <author>Choirul Ihwan</author>
  <creationDate>Agustus 2007</creationDate>
  <copyright>(C) 2007 Choirul Ihwan</copyright>
  <license>http://www.gnu.org/copyleft/gpl.html
GNU/GPL</license>
  <version>1.0</version>
  <description>module untuk menampilkan menu master
data</description>
  <authorEmail>aristhu_oracle@yahoo.com</authorEmail>
  <authorUrl>www.aristhu03.wordpress.com</authorUrl>
  <files>
    <filename
module="mod_master_data_menu">mod_master_data_menu.php</filenam
e>
  </files>
  <params />
</mosinstall>
```

Tag-tag xml yang digunakan untuk membuat file installasi modul joomla sudah disediakan oleh joomla sebagai *Application Programming Interface (API)* untuk joomla. *Developer* kemudian menggunakan API tersebut untuk memberikan informasi modul yang dibuat.

Berikut merupakan *source code* modul master data :

```
<?php
defined( '_VALID_MOS' ) or
    die( 'Direct Access to this location is not allowed.' );

$params->def('menutype', 'master_data');
mosShowVIMenu( $params );
?>
```

Karena modul menu master data digunakan untuk menampilkan menu, maka pembuatan modul cukup dilakukan dengan memanggil fungsi yang sudah ada dari modul main menu, dengan mengeset parameter *menutype* dengan menu yang ada di*database*.

Selain modul, implementasi pembuatan SIMTAP juga dengan menambahkan komponen. Berikut komponen-komponen yang dibuat :

a. Komponen daftar layanan
b. Komponen instansi

c. Komponen kecamatan
d. Komponen pemberi Izin
e. Komponen pemohon
f. Komponen permohonan
g. Komponen prosedur layanan
h. Komponen slip pembayaran
i. Komponen surat izin
j. Komponen survey
k. Komponen syarat layanan
l. Komponen download
m. Komponen draft survey
n. Komponen jenis identitas
o. Komponen kategori layanan
p. Komponen kewarganegaraan
q. Komponen kop surat
r. Komponen laporan
s. Komponen layanan_public
t. Komponen progress
u. Komponen status izin
v. Komponen status proses

Komponen-komponen diatas dibuat dengan menggunakan konsep MVC. Untuk satu modul paling tidak terdiri dari 1 file *Model,* 1 file *Controller* dan beberapa file *View.*

File Model berupa *class* yang berhubungan dengan pengambilan data dari database. *Class* tersebut memiliki beberapa *method* yang digunakan untuk melakukan proses manipulasi tabel database.

Berikut contoh *source code* file *Model* untuk komponen kecamatan.

```
class kecamatan {
   private $db;
   private $kecamatanNama;
   private $kecamatanId;

   function kecamatan($db){
      $this->db = $db;
   }

   function getAllKecamatan(){
      $query = '
               SELECT refKecamatanId AS Id,
               refKecamatanNama AS Nama
               FROM ref_kecamatan ORDER BY refKecamatanNama ';

      $this->db->setQuery( $query );
      $rows = $this->db->loadObjectList();
      return $rows;
   }
}
```

Class diatas digunakan untuk memanipulasi data pada tabel kecamatan. *Properti* db pada digunakan sebagai *accessor* pada koneksi database yang didefinisikan pada file konfigurasi. Selain itu *properti* db juga digunakan untuk mengakses *method-method* dari *class* database pada framework joomla. Method-method tersebut merupakan method-method yang digunakan untuk manipulasi database. Hasil dari manipulasi database, baik yang berupa *result set* maupun *boolean* kemudian diambil oleh *controller.*

Berikut *source code* untuk *controller* pada komponen kecamatan :

```
require_once 'kecamatan.class.php';
class kecamatanScreens {
private $db;

function kecamatanScreens($db, $my){
      $this->model = new kecamatan($db);
      $this->user = $my;
}

function &createTemplate() {
      global $option, $mosConfig_absolute_path;
      require_once( $mosConfig_absolute_path .
'/includes/patTemplate/patTemplate.php' );
      $tmpl =& patFactory::createTemplate( $option, true, false
);
      $tmpl->setRoot( dirname( __FILE__ ) . '/tmpl' );
            return $tmpl;
}

function viewKecamatan($task = "") {
      $rows = $this->model->getAllKecamatan();
      $tmpl =& kecamatanScreens::createTemplate();

      $tmpl->readTemplatesFromInput( 'kecamatan.html' );


      foreach($rows as $value){
         ++$no;

         $tmpl->addVar( 'rows', 'NO', $no );
         $tmpl->addVar( 'rows', 'NAMA', $value->Nama );
         $tmpl->addVar( 'rows', 'ID', $value->Id );

         $tmpl->parseTemplate( 'rows', 'a' );
         $tmpl->parseTemplate( 'canEdit', 'a' );
         $tmpl->clearTemplate( 'canEdit' );
      }

      $tmpl->displayParsedTemplate( 'content' );
 }
```

Konstruktor *class controller* meng*inisialisasi model* dengan membuat objek dari *class* kecamatan. Pada arsitektur MVC, untuk mem*parsing* data model ke view harus dilakukan menggunakan *template engine*. Joomla sendiri telah menyediakan *library patTemplate* sebagai *template engine*nya.

File *View* ditentukan pada *method* CreateTemplate. V*iew* yang dipake adalah file html. Berikut *source code* untuk *view* kecamatan :

```
<mos:tmpl name="content">
   <div class="componentheading"> Daftar Kecamatan</div>

   <mos:tmpl name="canAdd" visibility="hidden">
   <div class="pageBar">
   <!--paging section -->

   <div class="toolbar">
      <a href="{URL_ADD}"><img
src="includes/js/ThemeOffice/button-add.gif" alt="">
Tambah</a>
   </div>
   </div>
   </mos:tmpl>

   <table class="table-common" width="100%">

      <tr>
         <th width="5%">NO</th>
         <th>Nama</th>
         <mos:tmpl name="headerCanEdit" visibility="hidden">
            <th width="10%">Aksi</th>
         </mos:tmpl>
      </tr>
      <mos:tmpl name="rows">
         <tr>
            <td>{NO}</td>
            <td>{NAMA}</td>
            <mos:tmpl name="canEdit" visibility="hidden">
               <td class="links">
                  <a href="{URL_EDIT}"><img
src="includes/js/ThemeOffice/edit.png" border="0" alt="Edit"
title="Edit" /></a>
                  <a href="{URL_DELETE}"><img
src="includes/js/ThemeOffice/trash.png" border="0"
alt="Hapus" title="Hapus"/></a>
               </td>
            </mos:tmpl>
         </tr>
      </mos:tmpl>
   </table>

</mos:tmpl>
```

Tampilan yang dimunculkan pada browser adalah *skrip* yang ada diantara *<mos:tmpl name="content">* dan *</mos:tmpl>*. Sedangkan *string* yang ada diantara kurung kurawal, akan diisi dengan data hasil *parsing* pada *controller*.

Selain file *Model, View,* dan *Controller* diatas, juga ada satu file *dispatcher* yang berguna untuk menerima *request* dan *response* dari klien. Pada Joomla,

aturan penamaan file ini harus sama dengan nama komponen tanpa menggunakan *prefix* com. Pada komponen kecamatan, nama file *dispatcher* yang dipakai adalah kecamatan.php.

Berikut *source code* file kecamatan.php :

```
<?php

defined( '_VALID_MOS' ) or die( 'Direct Access to this location
is not allowed.' );
require_once( $mainframe->getPath( 'front_html' ) );
$task = mosGetParam( $_REQUEST, 'task', '' );
$kecamatanScreens = new kecamatanScreens($database, $my);

switch ($task) {
   default:
    $kecamatan = $kecamatanScreens->viewKecamatan();
   break;
}

?>
```

Variable task digunakan untuk menerima *request* dari klien. *Method* yang dieksekusi pada awal *request* adalah *viewKecamatan* yang digunakan untuk menampilkan daftar kecamatan pada table kecamatan.

3.3.3.2 Grafik SIMTAP

Untuk kebutuhan pembuatan laporan, pada SIMTAP memerlukan grafik. Oleh karena itu php yang digunakan harus mendukung *library gd.* Pada joomla sendiri dukungan untuk menampilkan informasi dalam bentuk grafik belum disediakan pada *core* aplikasi. Oleh karena itu pada penelitian ini untuk menampilkan grafik ditambahkan jpgraph.

Penambahan jpgraph pada core aplikasi berimplikasi apabila komponen yang dibuat dipindahkan ke aplikasi joomla yang lain yang tidak menyediakan jpgraph.

Untuk penanganan grafik, perlu dibuat *class image handler.* Berikut *source code class image handler* :

```
<?php
require_once ($_SERVER['DOCUMENT_ROOT'] .
"/skripsi/includes/jpgraph/jpgraph.php");
require_once ($_SERVER['DOCUMENT_ROOT'] .
"/skripsi/includes/jpgraph/jpgraph_bar.php");
require_once 'laporan.class.php';

class ImageLaporan {
   function ImageLaporan($data, $judul, $labelx){
      $this->data   = $data;
      $this->judul  = $judul;
      $this->color  = array("red", "blue", "darkblue", "green",
"orange", "black", "brown", "pink", "darkred", "yellow");
      $this->labelx = $labelx;
   }

   function createImage(){
      $graph = new Graph(300,200,"auto");
      $graph->SetScale("textlin");
      $graph->SetShadow();

      $graph->img->SetMargin(40,30,20,40);

      $bplot = new BarPlot($this->data);

      $bplot->value->Show();
      $bplot->value->SetFormat('%d');
      $bplot->value->SetFont(FF_FONT1,FS_BOLD);

      $bplot->SetFillColor($this->color);

      $graph->Add($bplot);

      $graph->title->Set($this->judul."Per bulan September");

      $graph->xaxis->SetTickLabels($this->labelx);
      $graph->xaxis->SetFont(FF_FONT1,FS_NORMAL,8);

      $graph->yaxis->scale->SetGrace(20);
      $graph->title->SetFont(FF_FONT1,FS_BOLD);
      $graph->yaxis->title->SetFont(FF_FONT1,FS_BOLD);
      $graph->xaxis->title->SetFont(FF_FONT1,FS_BOLD);

      $graph->Stroke();
   }
}
```

3.3.3.3 Manajemen User

Pada Joomla ada beberapa *group* yang sudah didefinisikan, yaitu : Super Administrator, Administrator, Manajer, Editor. Group-group tersebut tidak bisa dimanipulasi melalui aplikasi. Oleh karena itu, untuk keperluan SIMTAP, group-

group tersebut perlu diganti. Mekanisme penggantian dilakukan dengan memasukkan data langsung pada database. Tabel yang menyimpan group ada pada *jos_core_acl_aro_groups*. Ditabel tersebut masukkan group Administrasi, Kabag, Surveyor.

Berikutnya, yang perlu dilakukan adalah pembuatan table jos_groups_modules. Tabel ini digunakan untuk mengasosiasikan menu dengan suatu group tertentu. Tujuan dibuatnya tabel ini untuk memberikan hak akses modul dan hak akses menu pada user.

Selain itu, setiap user perlu diasosiasikan dengan instansi masing-masing untuk pembatasan hak akses data. Hal ini perlu dilakukan agar user pada suatu instansi hanya mendapatkan data-data dari instansi user tersebut. Pembatasan hak akses data disimpan pada tabel jos_users_instansi. Sebagaimana jos_groups_modules, tabel ini juga merupakan tabel tambahan dari sistem yang tidak disertakan pada joomla secara default.

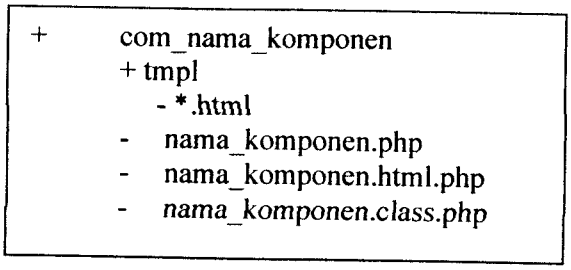
3.3.3.4 Library lain.

Untuk keperluan aplikasi, juga dibuat sebuah class library yang diletakkan pada *core* aplikasi. *Class* tersebut diletakkan pada file /includes/functions.php. Ada 2 *method* yang dibuat pada class library, yaitu Terbilang dan toIndonesianFormatDate.

Method terbilang digunakan untuk mengkonversi data bilangan ke bentuk kalimat dalam bahasa indonesia. Sedangkan *method* toIndonesianFormatDate digunakan untuk mengkonversi tanggal ke bentuk penanggalan indonesia dengan format dd-mm-yyyy.

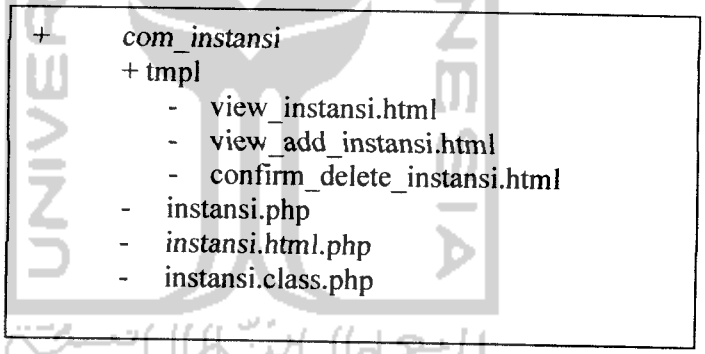
3.3.4 Struktur File

Pembuatan komponen dengan menggunakan MVC pada Joomla harus sesuai dengan aturan penyusunan dan penamaan struktur file. Struktur file untuk setiap komponen yang dibuat ditunjukkan pada gambar 3.23 berikut :



Gambar 3.23 Struktur file komponen-komponen SIMTAP

Dari susunan di atas, **nama_komponen.php** merupakan *dispatcher* yang menerima *request* dan *respons*. Selanjutnya *request* dan *response* dimasukkan kedalam *controller* pada **nama_komponen.html.php**. *Controller* memanggil data model dari **nama_komponen.class.php.** Data model yang didapatkan berupa *result set* atau *boolean. Result set* selanjutnya diparsing ke bentuk HTML dengan menggunakan view yang sesuai dibawah direktori tmpl. Misal untuk komponen instansi, struktur filenya ditunjukkan pada gambar 3.24 berikut :



Gambar 3.24 Struktur file komponen instansi

BAB IV

HASIL DAN PEMBAHASAN

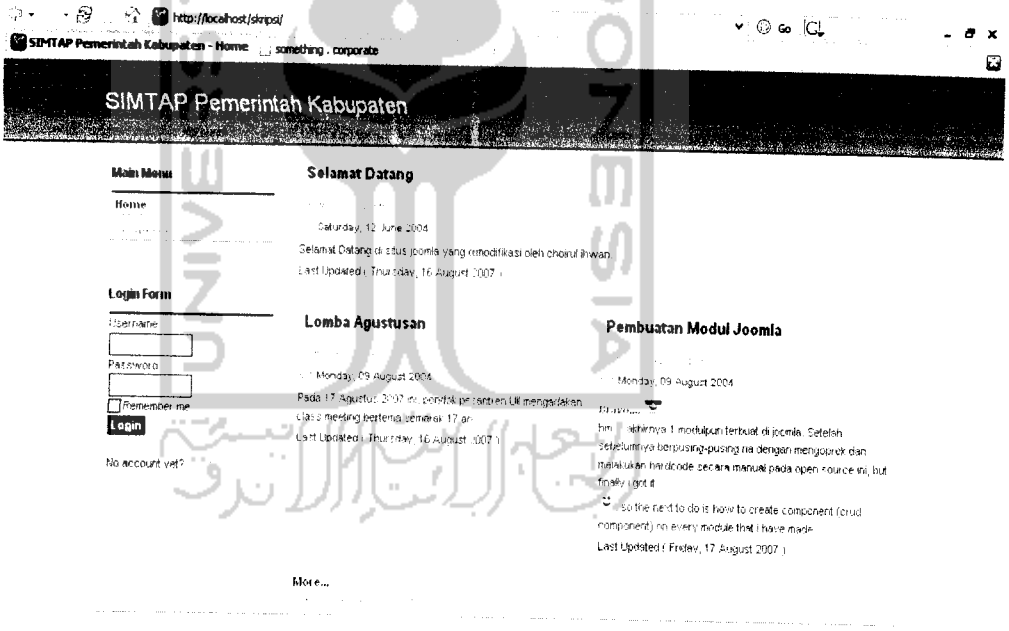
4. 1. Hasil dan Pembahasan

Pada bagian ini akan dibahas tentang hasil implementasi SIMTAP. Disini akan diuji apakah SIMTAP yang dibangun sesuai dengan fungsi dan tujuan di kembangkannya SIMTAP tersebut.

Berikut hasil dari implementasi SIMTAP :

4.1.1 Halaman Muka

Tampilan halaman muka SIMTAP ditunjukkan pada gambar 4.1 berikut :

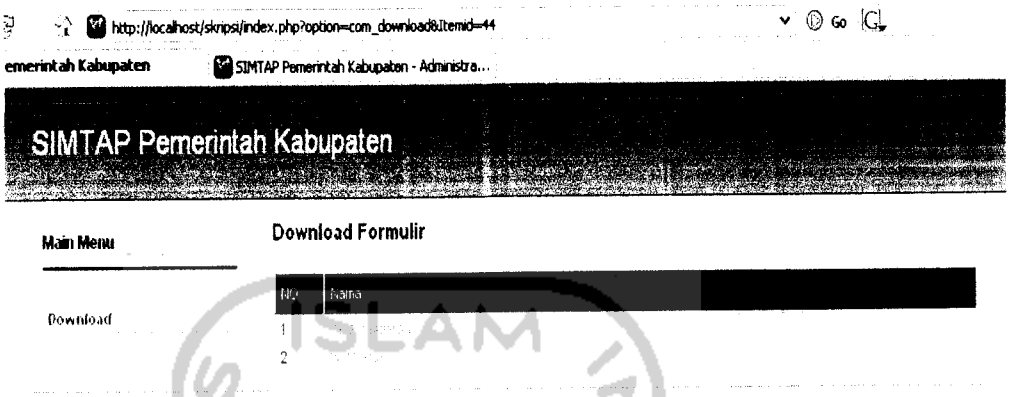


Gambar 4.1 Halaman muka SIMTAP

Halaman muka SIMTAP berisi informasi-informasi seputar SIMTAP yang diisikan oleh Administrator. Selain informasi juga terdapat form login untuk login ke *backoffice* SIMTAP dan menu utama yang berisi link ke halaman download dan halaman awal.

4.1.2 Halaman Download

Halaman *download* formulir ditunjukkan pada gambar 4.2 berikut :



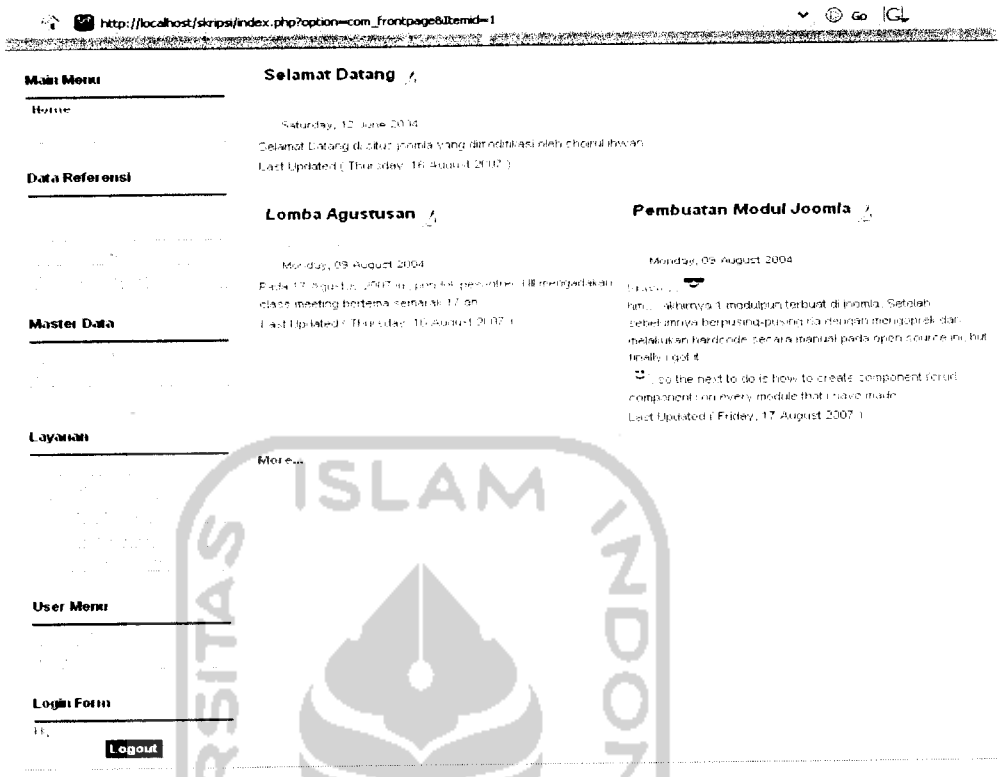
Gambar 4.2 Halaman unduh formulir

Halaman *download* berisi link ke formulir permohonan yang disediakan oleh SIMTAP. Pemohon dapat mengunduh setiap formulir permohonan yang ingin diajukan.

4.1.3 Halaman setelah login

Untuk pengguna selain pemohon dapat melakukan login ke *backoffice* SIMTAP. Format penamaannya adalah *nama_jabatan@nama_instansi.* Untuk contoh login, gunakan account user admin yang merupakan account user administrator.

Halaman setelah pengguna melakukan login ditunjukkan pada gambar 4.3 berikut :



Gambar 4.3 Halaman setelah login

Untuk pengguna yang terautentikasi (administrator, bag. Administrasi, peneliti lapangan, kabag) terdapat menu data referensi, data master, dan data layanan. Administrator adalah super user yang memiliki hak akses penuh terhadap data referensi. Sedangkan pengguna yang memiliki hak akses penuh terhadap data master dan data layanan adalah bag. Administrasi instansi.

4.1.4 Halaman Kecamatan

Halaman kecamatan ditunjukkan pada gambar 4.4 berikut :

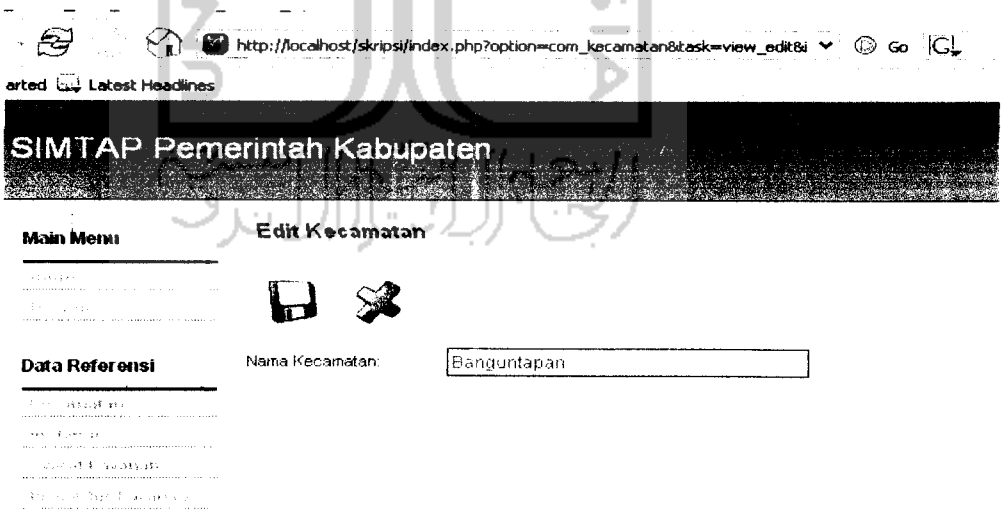
:



Gambar 4.4 Halaman kecamatan

Halaman kecamatan berisi data kecamatan di suatu kabupaten. Pada halaman kecamatan link untuk manipulasi data (*insert, update, dan delete*) muncul. Hal ini karena *session* pengguna yang sedang aktif adalah administrator yang memiliki hak akses penuh terhadap data kecamatan.

Halaman edit kecamatan ditunjukkan pada gambar 4.5 berikut ::



Gambar 4.5 Halaman edit kecamatan

Tampilan tambah dan edit kecamatan memiliki view yang sama, hanya saja ketika *request* yang dilakukan adalah tambah, sistem tidak mengambil data model dari database, tetapi ketika *request* edit dengan membawa parameter id kecamatan, *controller* kemudian mengambil data kecamatan berdasarkan id dan menampilkan hasilnya pada form edit kecamatan.

Untuk proses tambah dan update tidak terdapat formulir tersendiri, tetapi di*redirect*kan ke halaman awal suatu modul dengan membawa pesan sukses apabila proses berhasil dan gagal apabila proses gagal dilakukan.

Selain bisa menambah dan mengedit, bag. Administrasi juga dapat menghapus data kecamatan. Untuk menghapus klik icon hapus. Maka muncullah halaman konfirmasi penghapusan yang ditunjukkan pada gambar 4.6 berikut :



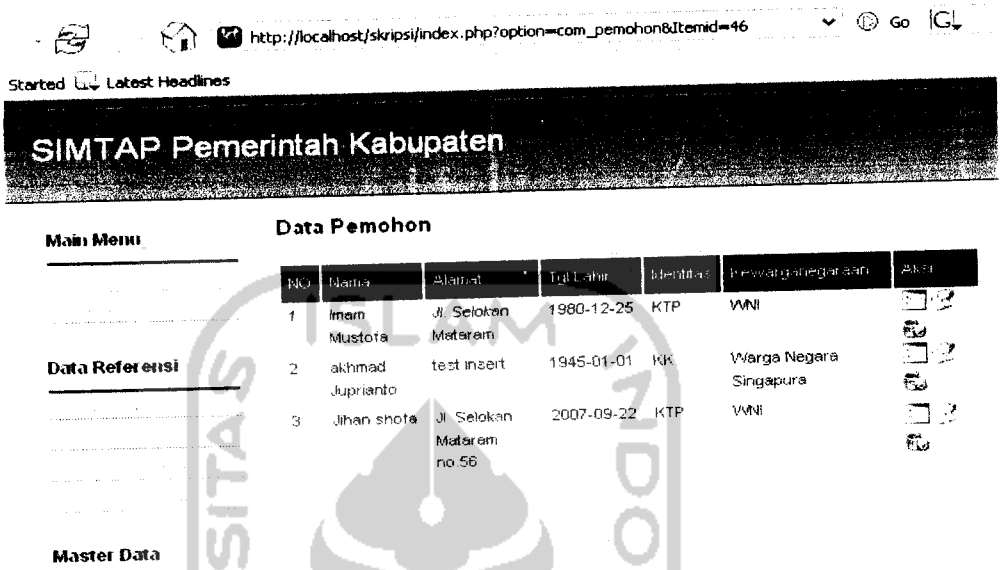
Gambar 4.6 Halaman konfirmasi penghapusan kecamatan

Untuk menghapus klik link hapus, maka sistem akan menghapus data kecamatan yang ingin dihapus.

4.1.5 Halaman Pemohon

Pengguna yang memiliki hak akses penuh pada halaman pemohon adalah bag. Administrasi. Oleh karena itu login dengan account administrasi. Misalnya admin@kependudukan untuk bagian administrasi dinas kependudukan. Halaman

pemohon dengan *session* pengguna admin@kependudukan ditunjukkan pada gambar 4.7 berikut :



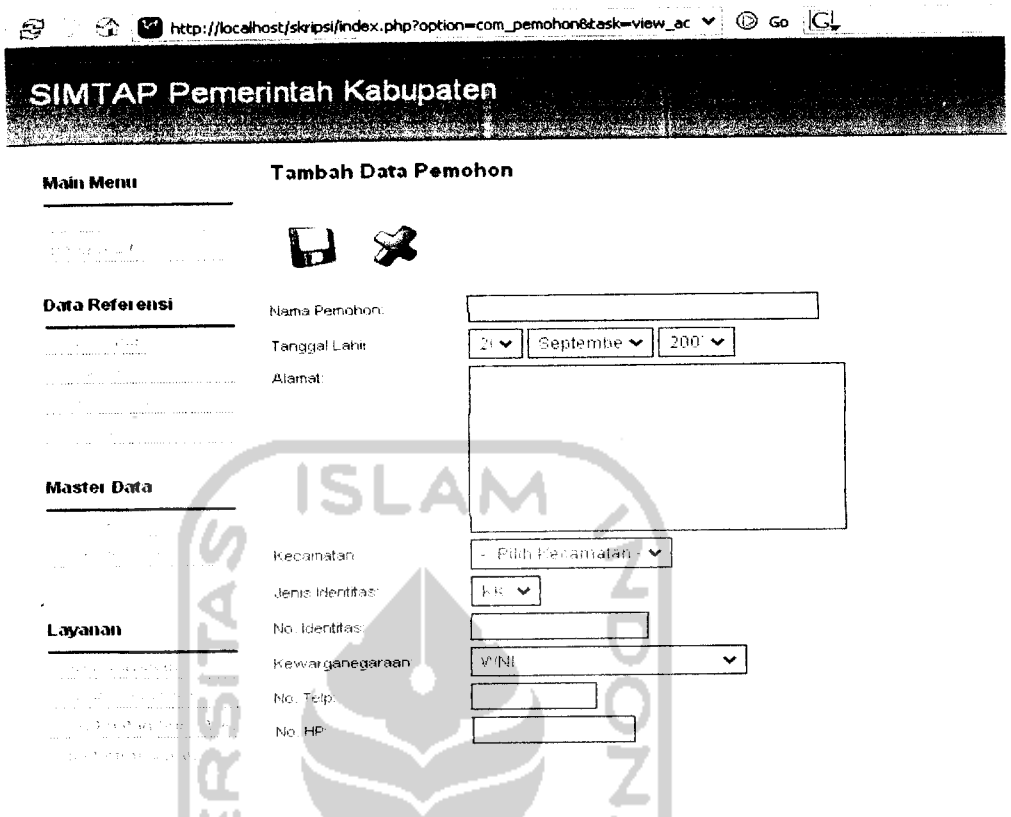
Gambar 4.7 Halaman pemohon

Halaman pemohon memberikan informasi nama, alamat, tanggal lahir, jenis identitas dan kewarganegaraan pemohon permohonan. Pada halaman permohonan terdapat 3 icon pada kolom aksi. Icon pertama digunakan untuk melihat detail pemohon. Icon kedua digunakan untuk mengedit data pemohon, dan icon yang ketiga digunakan untuk menghapus data pemohon. Selain ketiga icon aksi tersebut, pada form pemohon juga terdapat link untuk melakukan tambah.

Penambahan pemohon dilakukan oleh bag. administrasi apabila ada pemohon yang telah menyerahkan formulir permohonan dan melengkapi semua persyaratan permohonan.

Halaman tambah pemohon ditunjukkan pada gambar 4.8 berikut :

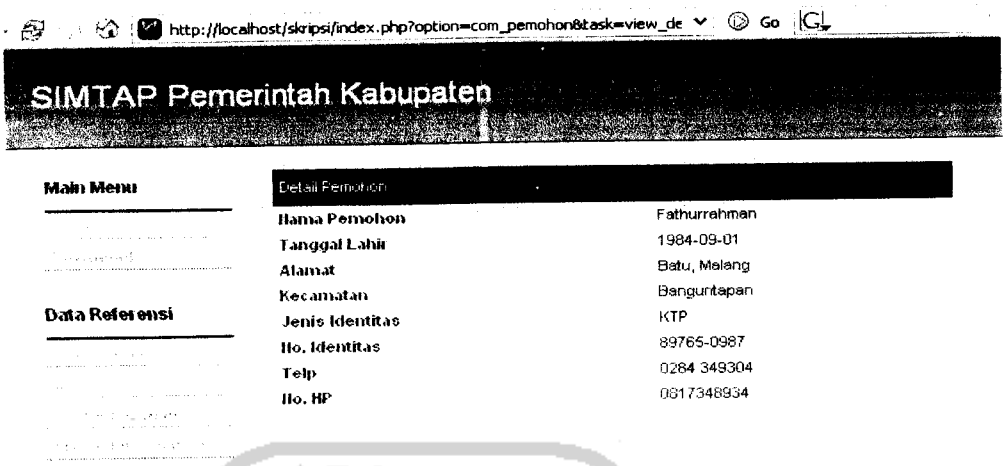
:



Gambar 4.8 Halaman tambah pemohon

Isikan data pemohon pada form sesuai dengan kondisi pemohon yang mengajukan permohonan. View form tambah pemohon sama dengan form edit pemohon, hanya saja pada tambah pemohon *controller* tidak memanggil model yang menghasilkan result set informasi pemohon.

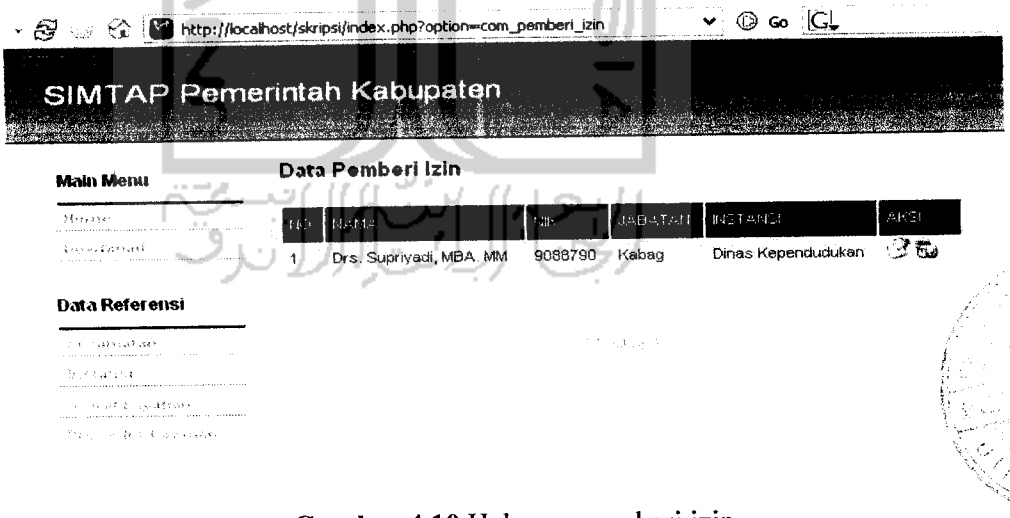
Untuk melihat detail pemohon, klik icon detail. Detail pemohon ditunjukkan pada gambar 4.9 berikut :



Gambar 4.9 Halaman detail pemohon

4.1.6 Halaman Pemberi Izin

Halaman pemberi izin digunakan untuk menampilkan informasi yang berkaitan dengan pemberi izin. Halaman pemberi izin ditunjukkan pada gambar 4.10 berikut :

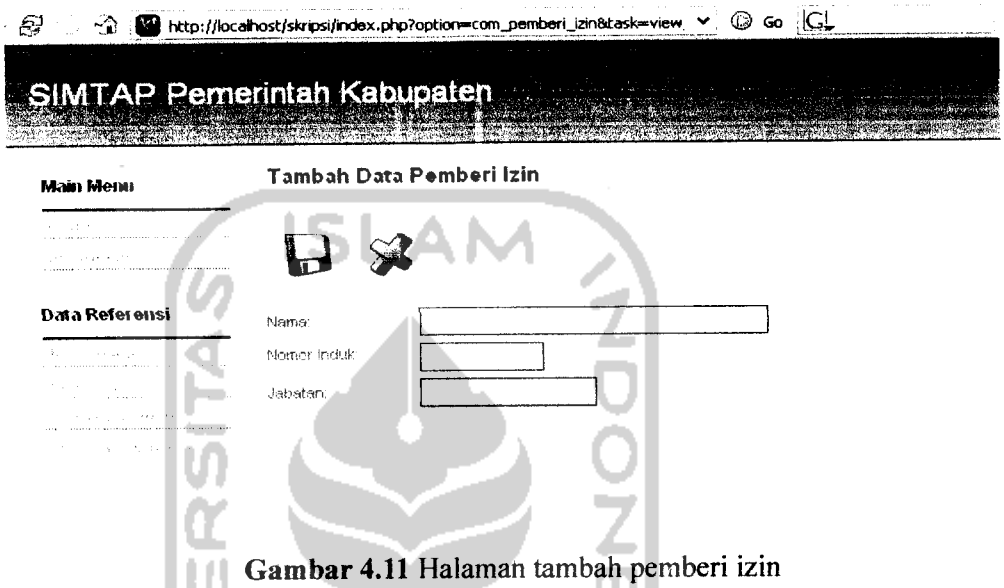


Gambar 4.10 Halaman pemberi izin

Umumnya pemberi izin untuk suatu jenis perizinan adalah kepala bagian instansi terkait. Hal ini telah diakomodir oleh sistem dengan memasukkan instansi

pada session yang melekat pada user, sehingga ketika terjadi proses penambahan pemberi izin, bagian administrasi tidak dapat menambahkan pemberi izin untuk selain instansinya.

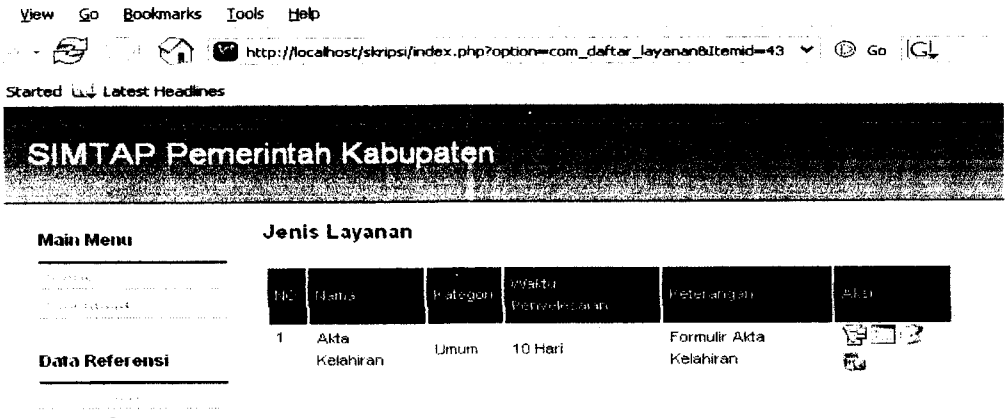
Halaman tambah pemberi izin ditunjukkan pada gambar 4.11 berikut :



Gambar 4.11 Halaman tambah pemberi izin

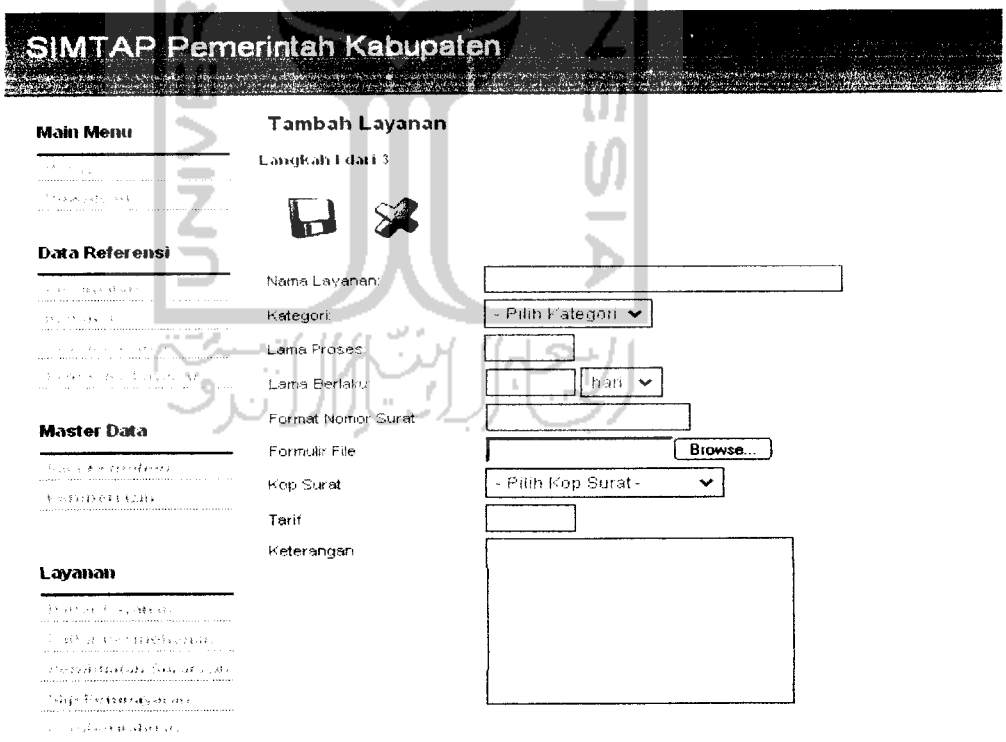
4.1.7 Halaman Daftar Layanan

Halaman daftar layanan menampilkan list jenis layanan yang disediakan oleh suatu instansi. Sebagaimana pada data master, pengguna yang memiliki hak akses penuh terhadap halaman daftar layanan adalah bagian administrasi. Halaman daftar layanan ditunjukkan pada gambar 4.12 berikut :



Gambar 4.12 Halaman jenis layanan

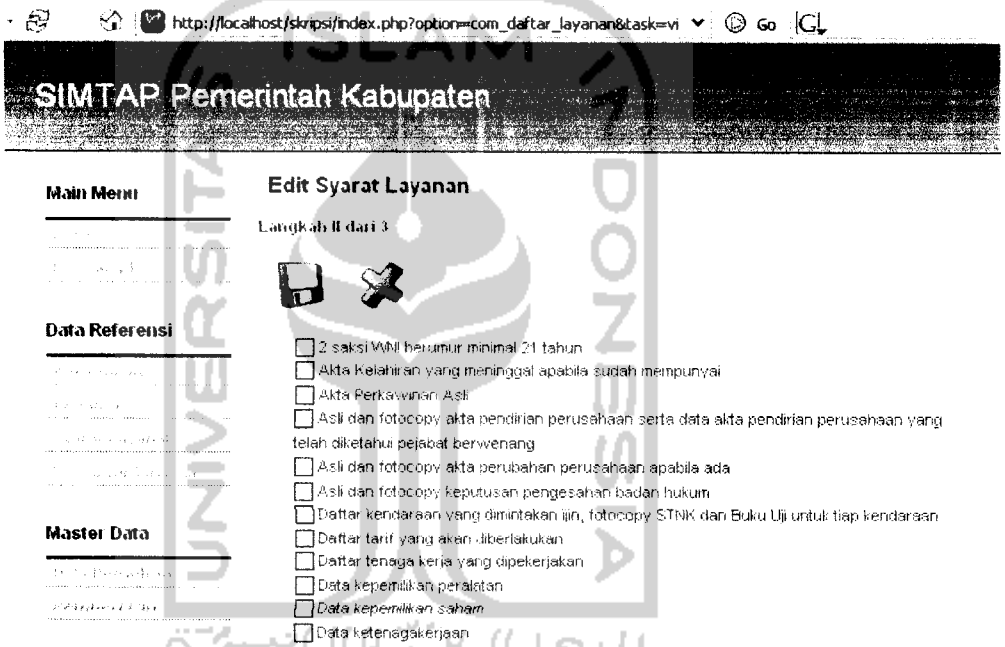
Informasi yang diberikan pada halaman layanan mencakup nama, kategori, lama waktu penyelesaian dan keterangan. Untuk data layanan, terdapat 5 macam aksi yang bisa dilakukan, yaitu : tambah layanan, tulis draft layanan, detail layanan, edit layanan dan hapus layanan. Halaman tambah layanan ditunjukkan pada gambar 4.13 berikut :



Gambar 4.13 Halaman tambah jenis layanan

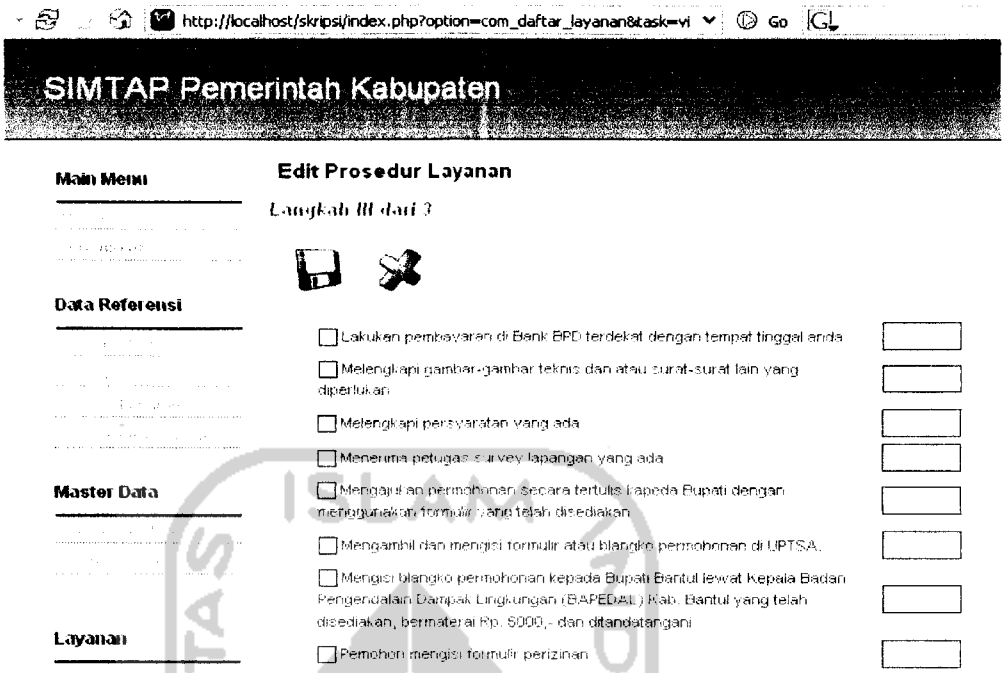
Proses penambahan layanan terdiri dari beberapa langkah. Untuk menangani hal ini dibuatlah semacam *wizard* yang memberikan panduan pada user untuk menjalankan masing-masing langkah.

Langkah pertama yaitu dengan mengisikan data yang berhubungan dengan informasi layanan itu sendiri, sebagaimana tampilan di atas. Pada form di atas, isikan data yang sesuai dengan kondisi layanan yang akan diberikan. Selanjutnya muncul *wizard* kedua untuk menambahkan syarat pada jenis layanan seperti yang ditunjukkan pada gambar 4.14 berikut :



Gambar 4.14 Halaman tambah syarat layanan

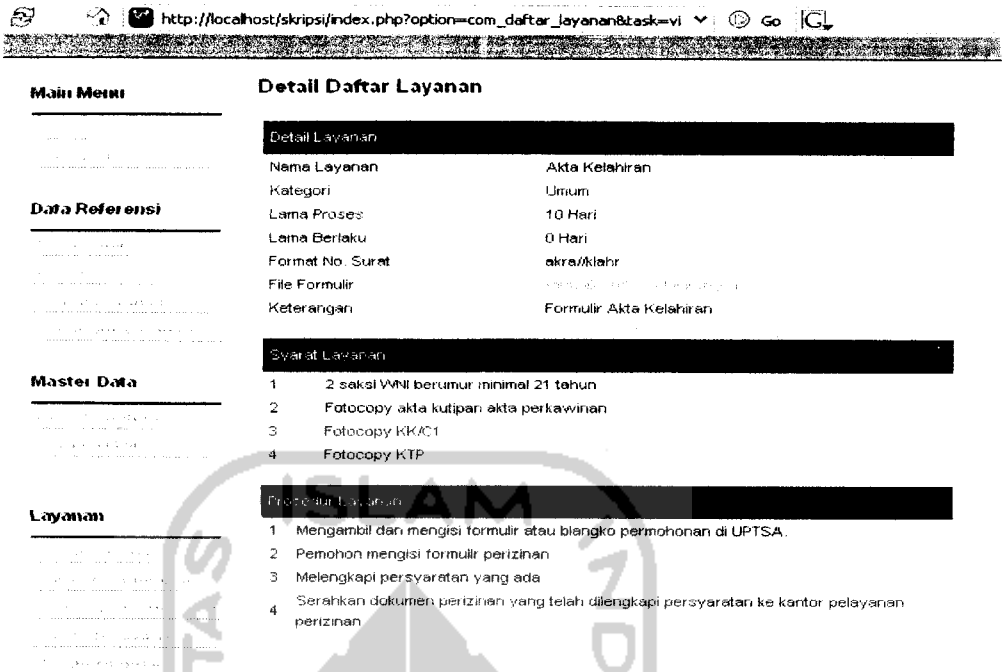
Wizard kedua digunakan untuk memasukkan syarat layanan pada jenis layanan yang diberikan. Hal ini penting, karena pemohon harus memenuhi syarat-syarat layanan yang diajukan. Klik checkbox syarat layanan, kemudian klik icon simpan, selanjutnya muncul *wizard* ketiga untuk menambahkan prosedur layanan seperti yang ditunjukkan pada gambar 4.15 berikut :



Gambar 4.15 Halaman tambah prosedur layanan

Wizard ketiga digunakan untuk memasukkan prosedur layanan dan urutan pelaksanaan prosedur layanan. Klik checkbox prosedur layanan untuk menambahkan prosedur layanan dan isikan urutan prosedur tersebut harus dijalankan pada kotak input yang ada di sebelah kanan prosedur layanan.

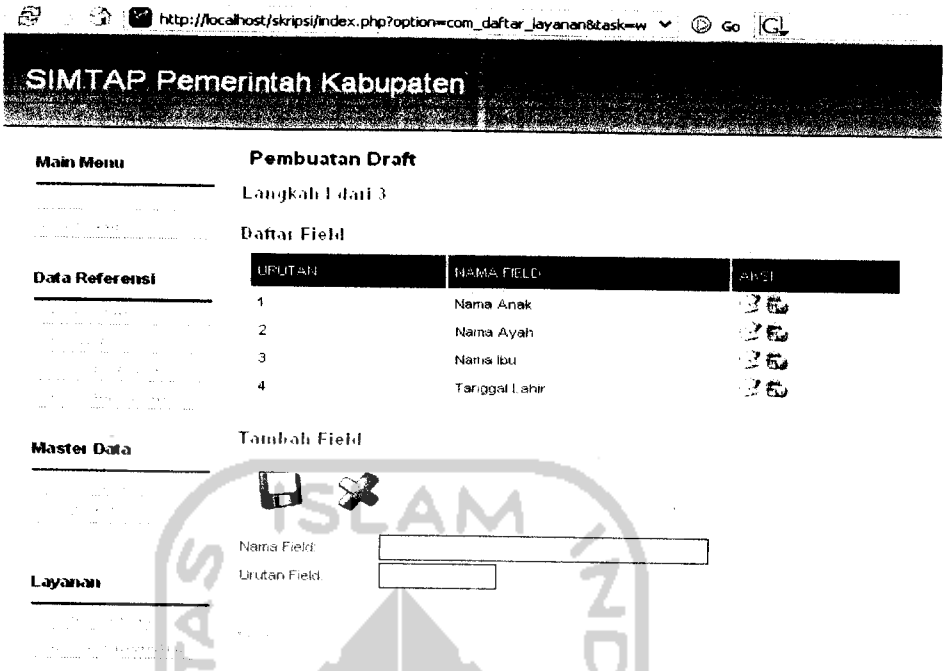
Untuk melihat detail layanan, klik tombol detail, muncul halaman detail layanan seperti yang ditunjukkan pada gambar 4.16 berikut :



Gambar 4.16 Halaman detail jenis layanan

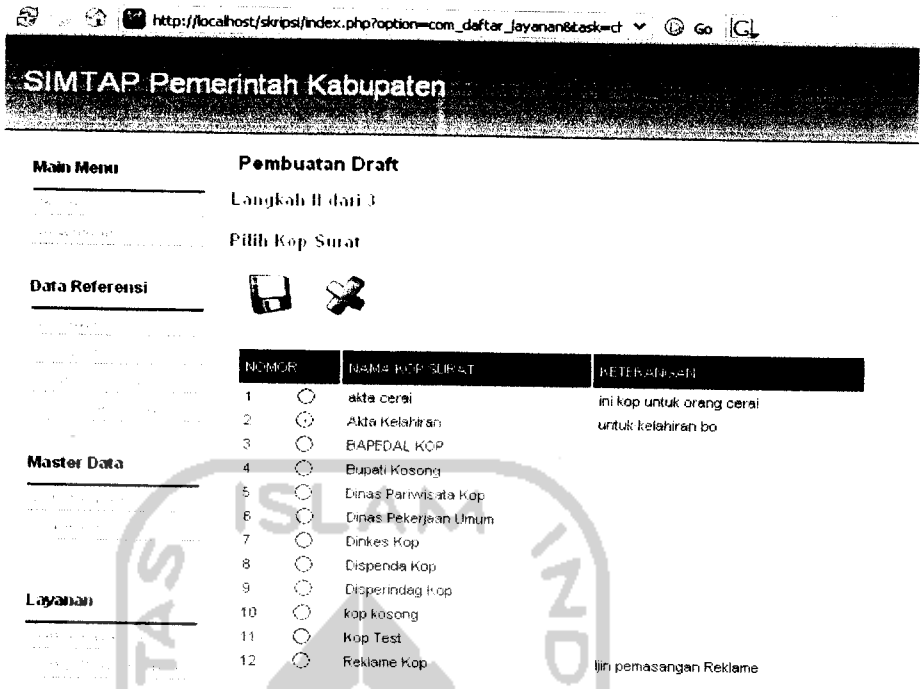
Halaman detail layanan menampilkan informasi layanan yang diberikan beserta syarat-syarat layanan dan urutan prosedur yang harus dijalankan.

Selain beberapa aksi di atas, pada halaman jenis layanan juga tersedia aksi untuk menulis draft layanan. Sebagaimana penambahan jenis layanan, pembuatan draft menggunakan *wizard*. Ada beberapa langkah yang harus dilakukan untuk pembuatan draft. Langkah pertama adalah menambahkan field-field yang diperlukan untuk perizinan. Halaman langkah pertama penulisan draft surat izin ditunjukkan pada gambar 4.17 berikut :



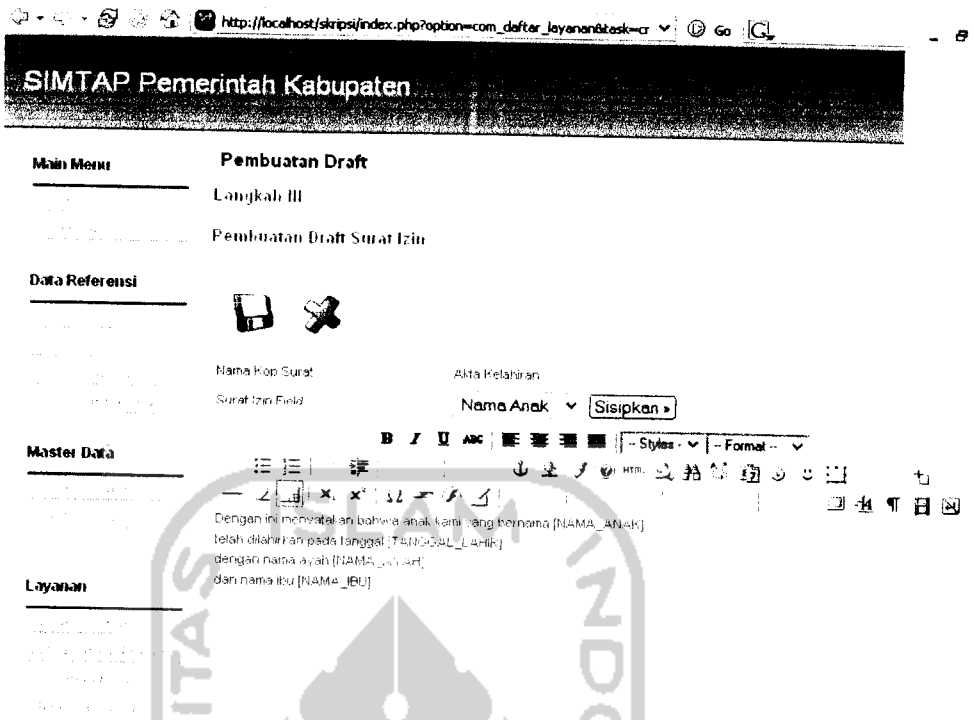
Gambar 4.17 Halaman langkah I pembuatan draft surat izin

Setelah penambahan field, *wizard* kedua digunakan untuk pemilihan kop. Klik salah satu kop untuk memilih kop yang akan digunakan. Halaman pemilihan kop ditunjukkan pada gambar 4.18 berikut :



Gambar 4.18 Halaman langkah II pembuatan draft surat izin

Selanjutnya muncul *wizard* ketiga yang digunakan untuk penulisan draft surat izin. Halaman *wizard* ketiga ditunjukkan pada gambar 4.19 berikut :

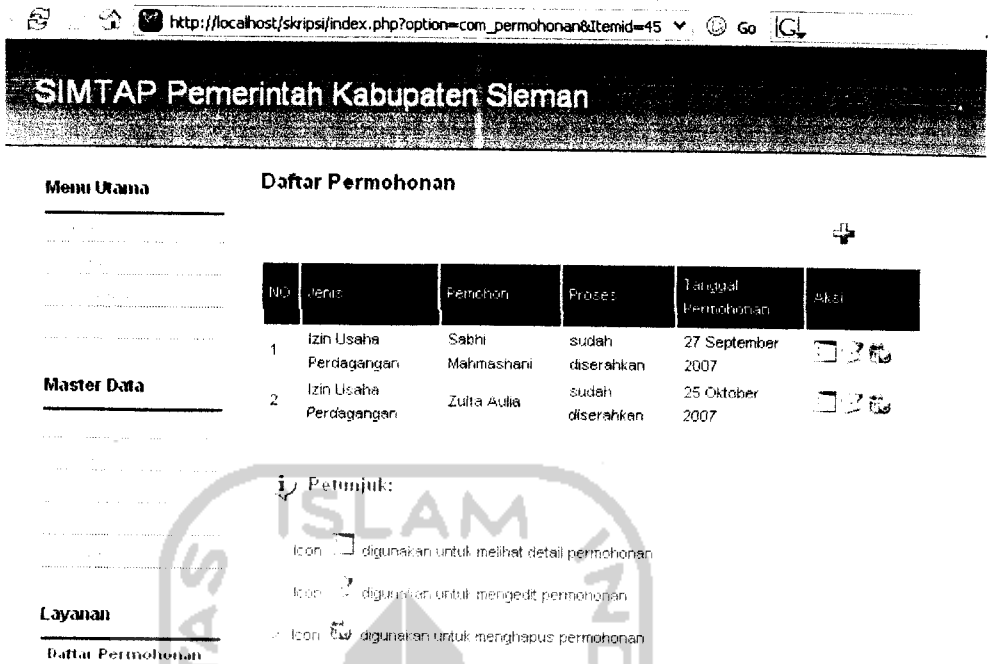


Gambar 4.19 Halaman langkah III pembuatan draft surat izin

Wizard ketiga digunakan untuk membuat draft perizinan. Nama-nama field yang telah dibuat pada langkah sebelumnya disisipkan pada draft. Format nama field yang digunakan adalah *[nama_field]*. Ketika pembuatan surat izin, nama field tersebut akan di *replace* dengan data permohonan perizinan.

4.1.8 Halaman Daftar Permohonan

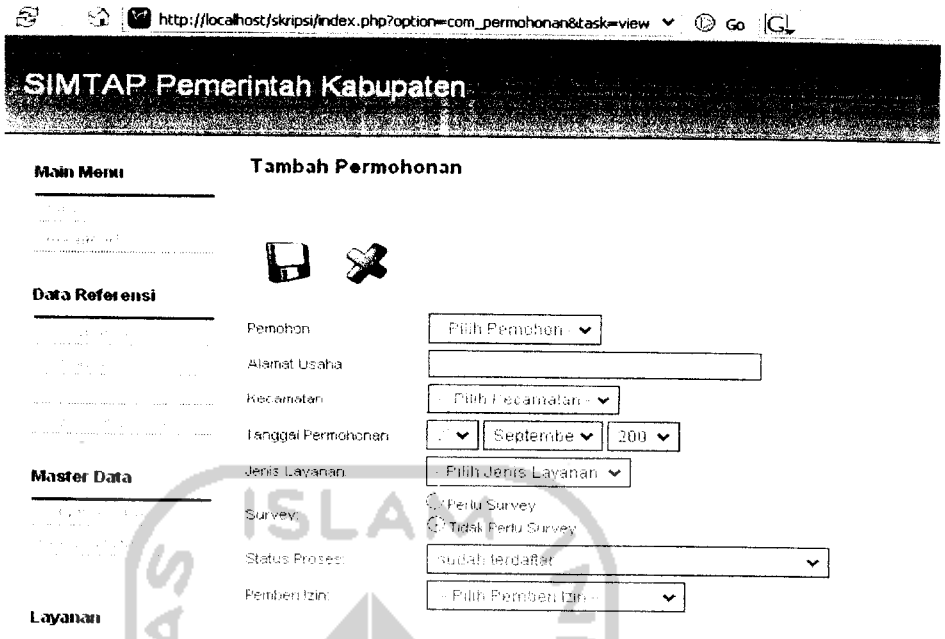
Halaman daftar permohonan menampilkan informasi seluruh permohonan yang masuk pada suatu instansi. Halaman daftar permohonan ditunjukkan pada gambar 4.20 berikut :



Gambar 4.20 Halaman daftar permohonan

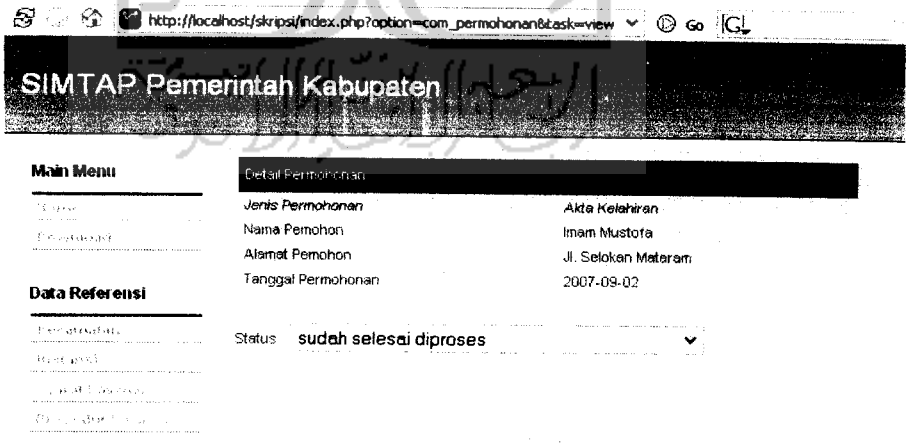
Informasi yang ditampilkan pada halaman permohonan meliputi jenis permohonan, nama pemohon, status progress permohonan dan tanggal permohonan. Selain itu, beberapa aksi yang bisa dilakukan oleh bagian administrasi pada halaman permohonan.

Untuk menambahkan permohonan klik link tambah. Penambahan permohonan dilakukan apabila pemohon telah memenuhi syarat-syarat pengajuan permohonan dan telah mengisi blanko permohonan. Tampilan halaman tambah permohonan ditunjukkan pada gambar 4.21 berikut :



Gambar 4.21 Halaman tambah permohonan

Sebagaimana disebutkan sebelumnya, status proses permohonan akan senantiasa mengalami perubahan seiring dengan berjalannya dokumen permohonan. Pengesetan status proses permohonan ada pada halaman detail permohonan seperti ditunjukkan pada gambar 4.22 berikut :

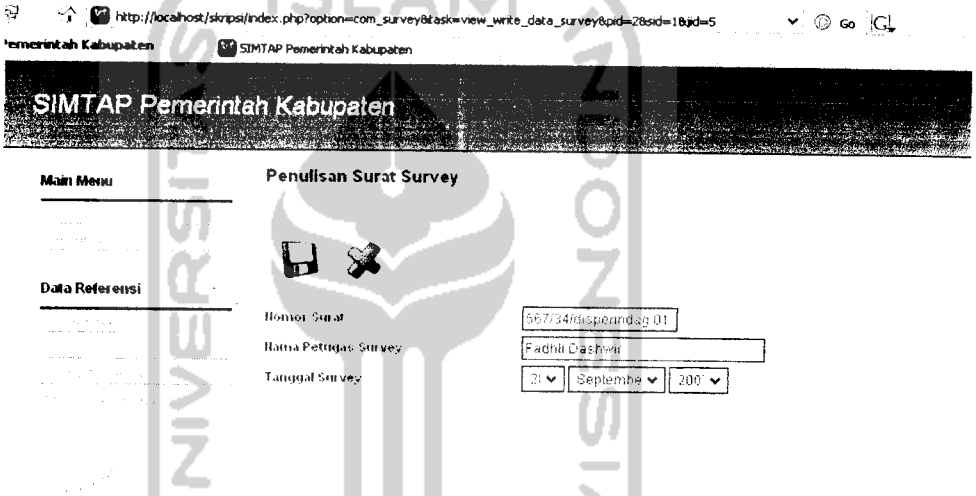


Gambar 4.22 Halaman detail permohonan

4.1.9 Halaman Surat Survey

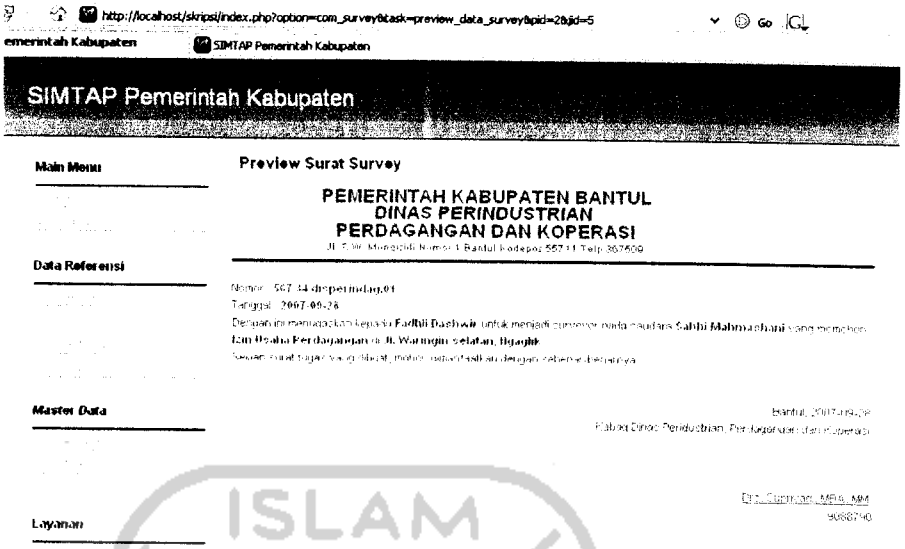
Halaman surat survey digunakan untuk membuat surat pengantar untuk dan surat pemberitahuan survey lapangan. Surat pengantar survey lapangan digunakan oleh petugas survey ketika melakukan survey. Sedangkan surat pemberitahuan survey diberikan kepada pemohon untuk memberitahukan waktu pelaksanaan dan petugas survey lapangan.

Untuk pembuatan surat survey lapangan, perlu data petugas survey, tanggal survey dan nomor surat yang ditampilkan dalam bentuk form seperti ditunjukkan pada gambar 4.23 :



Gambar 4.23 Form pembuatan surat survey

Setelah data di atas dimasukkan selanjutnya bagian administrasi dapat melihat preview surat pengantar survey lapangan dan surat pemberitahuan survey lapangan seperti yang ditunjukkan pada gambar 4.24 berikut :

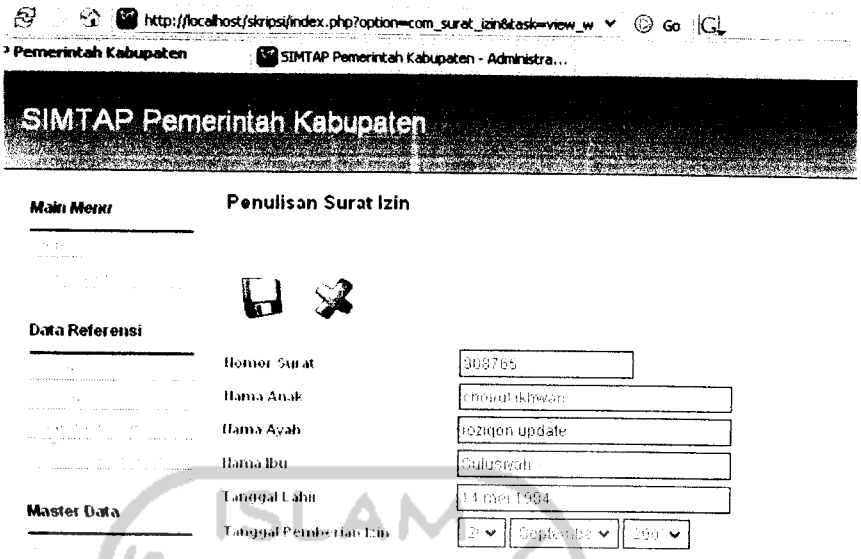


Gambar 4.24 Halaman preview surat survey

4.1.10 Halaman Surat Izin

Halaman surat izin digunakan untuk menulis surat izin dari permohonan yang diajukan. Halaman pertama untuk surat izin, hampir sama dengan halaman permohonan. Ada beberapa aksi yang dapat dilakukan pada halaman surat izin, yaitu tulis surat izin dan preview surat izin.

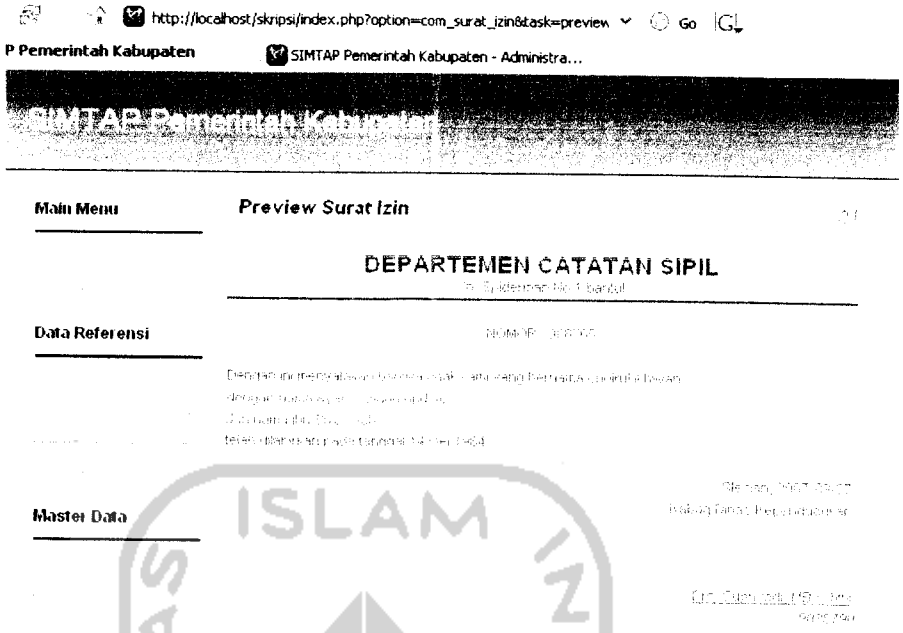
Halaman tulis surat izin ditunjukkan pada gambar 4.25 :



Gambar 4.25 Form penulisan surat izin

Penulisan surat izin dilakukan dengan mengambil field-field yang telah didefinisikan pada draft. Untuk gambar di atas, field nama anak, nama ayah, nama ibu, dan tanggal lahir adalah field yang diambil dari draft. Selanjutnya field-field tersebut diisi dengan data yang sesuai untuk dibuatkan surat izin.

Untuk melihat preview surat izin sebelum diprint klik icon preview surat izin. Muncul halaman preview surat izin seperti ditunjukkan pada gambar 4.26 berikut :



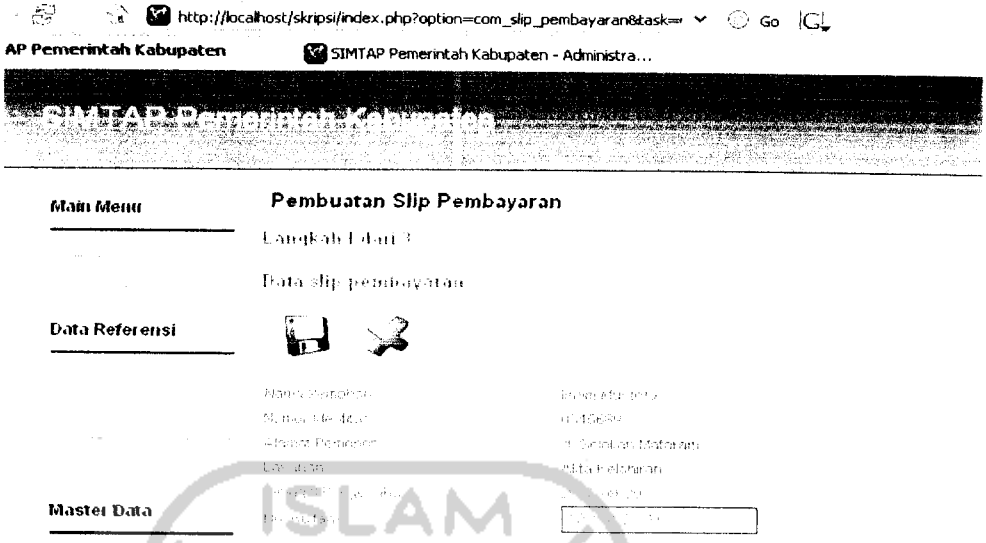
Gambar 4.26 Halaman preview surat izin

Untuk mencetak surat izin, klik icon print pada sudut kanan atas halaman preview surat izin.

4.1.11 Halaman Slip Pembayaran

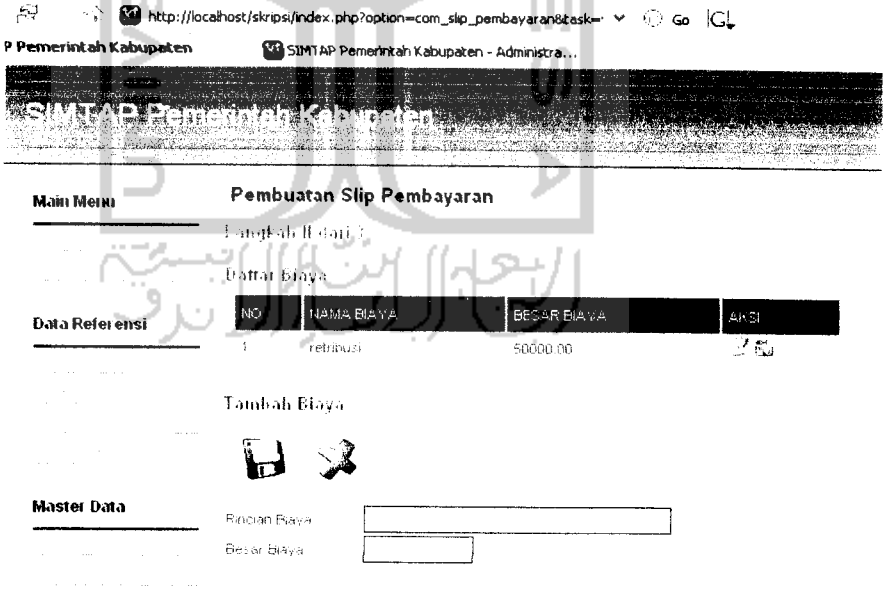
Halaman slip pembayaran merupakan halaman terakhir dalam proses penanganan perizinan SIMTAP. Halaman ini digunakan untuk mencetak kuitansi pembayaran untuk suatu permohonan. Ada 2 aksi yang bisa dilakukan pada halaman slip pembayaran, yaitu pembuatan slip dan preview slip pembayaran.

Halaman pembuatan slip pembayaran ditunjukkan pada gambar 4.27 berikut :



Gambar 4.27 *Form penulisan slip pembayaran*

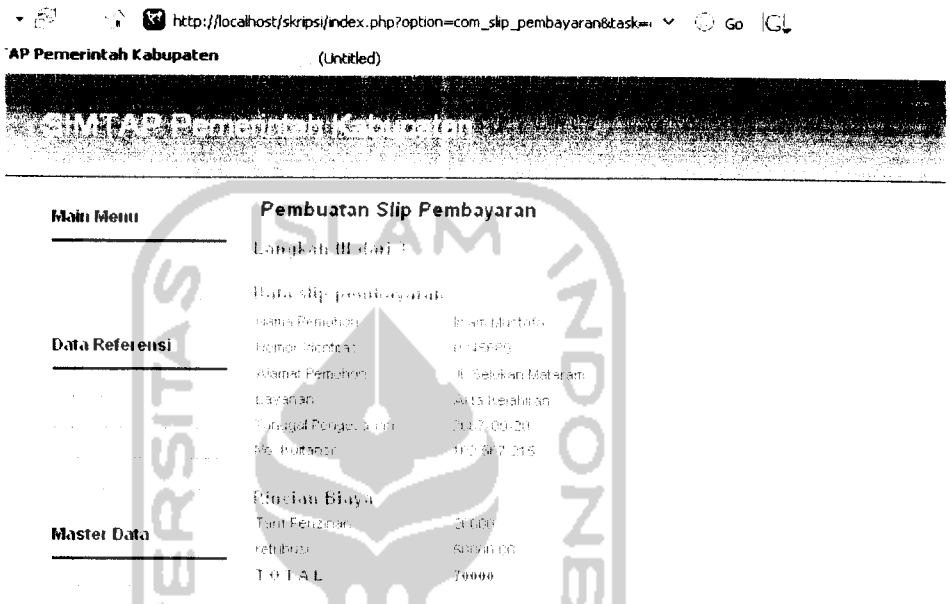
Pembuatan slip pembayaran dilakukan dengan menggunakan *wizard.* Langkah pertama memasukkan no. Kuitansi pembayaran. Klik icon simpan setelah memasukkan no. Kuitansi, muncul *wizard* kedua seperti yang ditunjukkan pada gambar 4.28 berikut :



Gambar 4.28 Form rincian biaya slip pembayaran

Wizard kedua digunakan untuk memasukkan biaya-biaya perizinan. Biaya-

biaya tersebut nantinya akan ditambahkan dengan biaya tarif dasar perizinan. Total biaya dan rincian biaya yang harus dikeluarkan oleh pemohon ditampilkan pada halaman detail slip pembayaran seperti yang ditunjukkan pada gambar 4.29 berikut :



Gambar 4.29 Halaman detail slip pembayaran

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melalui tahapan analisis, desain, implementasi dan pengujian maka dapat diambil beberapa kesimpulan bahwa :

a. Penggunaan SIMTAP dapat memperbaiki pelayanan pemerintah daerah kepada masyarakat.
b. Untuk membuat SIMTAP yang dapat mengakomodir berbagai jenis layanan, perlu dibuat tabel khusus yang mencatat *field-field* yang diperlukan untuk pembuatan perizinan.
c. Penggunaan desain arsitektur MVC dalam implementasi program akan memudahkan pembuatan program .
d. Implementasi *modular* akan memudahkan dalam *maintenance* kode program selanjutnya.
e. Pembuatan *component-component* SIMTAP sebagai komponen yang berdiri sendiri dan dapat diinstall sulit untuk direalisasikan karena adanya *dependency* antara satu *class* dengan *class* yang lain.
f. Template untuk joomla yang banyak tersedia di internet, ternyata tidak semuanya memiliki *class* css yang sama. Hal ini mengakibatkan tampilan web site yang dibangun dengan joomla sering tidak sesuai dengan harapan.

5.2 Saran

Saran untuk pengembangan SIMTAP berikutnya adalah:

a. Diharapkan program berikutnya dapat menambahkan berbagai fasilitas dan fungsionalitas.

b. Diharapkan dalam pengembangan perangkat lunak selanjutnya juga mempertimbangkan faktor interoperabilitas perangkat yang mengakses SIMTAP.



DAFTAR PUSTAKA

[ALH03] Alhir, Sinan. 2003. *Learning UML*. O'Reilly

[AZI05] Azis, M.Farid. 2005. *Object Oriented Programming dengan PHP 5*. Jakarta:PT. Elex Media Komputindo

[BOO98] Booch, et.al. 1998. *The Unified Modelling Language User Guide*. Addison Wesley.

[HAR05] Harijadi, Djoko Agung. 2005. *Blueprint Aplikasi E-Government Pemerintah Daerah. Bandung*. Prosiding Konferensi Nasional Teknologi Informasi dan Komunikasi Indonesia ITB, 3-4 Mei 2005.

[IND07] *SIMTAP* (online) . http://www.indonux.co.id. Diakses tanggal 27 september 2007

[KUT07] *Metode Perizinan* (online) http://www.kutaikartanegarakab.go.id Diakses tanggal 27 september 2007.

[SCH04] Schmuller, 2004. *Sams Teach Yourself UML in 24 Hours, Third Edition*. Sams Publishing.

[WAH04] Wahyono, Teguh. 2004. *Sistem Informasi Konsep Dasar, Analisis Desain dan Implementasi*. Yogyakarta:Penerbit Graha Ilmu.

[WHI06] White, Steve dan Wallace, Andy. 2006. *Joomla! 1.0.11 User Manual*.

[WIK05] *E-Government*. (online) http://en.wikipedia.org/wiki/E-Government. Diakses tanggal 27 september 2007.

Iterative and Incremental Development (online) http://en.wikipedia.org/wiki/Incremental_and_iterative_development. Diakses tanggal 18 Juni 2007