

TUGAS AKHIR

**Pengembangan Aplikasi Pelayanan Izin Mendirikan  
Bangunan (Studi Kasus Pada Badan Pelayanan Perijinan  
Terpadu Kota Tangerang Selatan)**



Oleh:

**WAHYU ARI KURNIAWAN**

**106091002975**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH**

**JAKARTA**

**2011**

**Pengembangan Aplikasi Pelayanan Izin Mendirikan Bangunan  
(Studi Kasus Pada Badan Pelayanan Perijinan Terpadu Kota  
Tangerang Selatan)**

Oleh :

**WAHYU ARI KURNIAWAN**  
**106091002975**

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar

Sarjana Komputer

Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UIN SYARIF HIDAYATULLAH**  
**JAKARTA**  
**2011 M / 1432 H**

**Pengembangan Aplikasi Pelayanan Izin Mendirikan Bangunan  
(Studi Kasus Pada Badan Pelayanan Perijinan Terpadu Kota  
Tangerang Selatan)**

Skripsi  
Sebagai Salah Satu Syarat Untuk Memperoleh Gelar  
Sarjana Komputer  
Pada Fakultas Sains dan Teknologi UIN Jakarta

Oleh :  
WAHYU ARI KURNIAWAN  
106091002975

Menyetujui,

Pembimbing 1

Pembimbing 2

Viva Arifin, MMSI  
NIP 19730810 200604 2 001

Imam M. Shofi, MT  
NIP. 19720205 200801 1 020

Mengetahui,  
Ketua Program Studi Teknik Informatika

Yusuf Durachman, M.Sc, MIT.  
NIP: 19710522 200604 1 002

# **Pengembangan Sistem Informasi Manajemen Surat Keputusan**

## **Studi Kasus: Subbagian Administrasi Kepegawaian Pusat**

**UIN Syarif Hidayatullah Jakarta**

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar

Sarjana Komputer

Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

Oleh:

Sarika

106091002966

Menyetujui,

Pembimbing I

Pembimbing II

Yusuf Durachman M.Sc, MIT

Husni Teja Sukmana, Ph.D

NIP. 19710522 200604 1 002

NIP. 19771030 200112 1 003

Mengetahui,  
Ketua Program Studi Teknik Informatika,

**Yusuf Durrachman, M.sc., M.I.T.**  
**NIP.197110522 200604 1 002**

## PENGESAHAN UJIAN

Skripsi yang berjudul "Pengembangan Aplikasi Pelayanan Izin Mendirikan Bangunan (Studi Kasus Pada Badan Pelayanan Perijinan Terpadu Kota Tangerang Selatan) telah diuji dan dinyatakan lulus dalam Sidang Munaqosah Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta, pada hari Kamis tanggal 27 Mei 2011. Skripsi ini telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Teknik Informatika

Jakarta, 27 Mei 2011

Tim Penguji,

Penguji I

Husni Teja Sukmana, Ph.D  
NIP. 19771030 200112 1 003

Penguji II

Yusuf Durrachman M.Sc, MIT  
NIP. 19710522 200604 1 002

Tim Pembimbing,

Pembimbing I

Viva Arifin, MMSI  
NIP. 19730810 200604 2 001

Pembimbing II

Imam M. Shofi, MT  
NIP. 19720205 200801 1 010

Mengetahui,

Dekan  
Fakultas Sains Dan Teknologi

DR. Syopiansyah Jaya Putra, M.Sis  
NIP. 19680117 200112 1 001

Ketua Program Studi  
Teknik Informatika

Yusuf Durrachman M.Sc, M.I.T.  
NIP. 19710522 200604 1 002

## **PERNYATAAN**

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SENDIRI YANG BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN.

Jakarta, Mei 2011

Wahyu Ari Kurniawan

## ABSTRAK

**Wahyu Ari Kurniawan (106091002975).** Pengembangan Aplikasi Pelayanan Izin Mendirikan Bangunan (Studi Kasus Pada Badan Pelayanan Perijinan Terpadu Kota Tangerang Selatan). Di bawah bimbingan Viva Arifin, M.MSI dan Imam M. Shofi, MT.

BP2T Kota Tangerang Selatan Selama masa kerjanya BP2T masih menggunakan aplikasi semi manual dalam mengelola perijinan serta dalam hal penerbitan laporan-laporan yang dibutuhkan serta pembayaran perijinan (MS. Excel 2003). Oleh karena itu sering terjadinya kesalahan-kesalahan yang ditemui yaitu kesalahan hitung karena perbedaan perhitungan dan koefisien-koefisien yang dibutuhkan pada Izin Mendirikan Bangunan, sering ditemukan ketidaksesuaian data mulai proses pendaftaran hingga proses penerbitan ijin pada masing - masing bagian, pengulangan input data pemohon pada masing - masing bagian (bagian pendaftaran, pengawasan dan pengendalian, penetapan, validasi) karena tidak terkomputerisasi dengan baik. Rumusan masalah dalam penelitian ini adalah membangun suatu aplikasi pelayanan Ijin Mendirikan Bangunan (IMB) yang sesuai tujuan/ sasaran dalam penyelesaian perijinan di BP2T Kota Tangerang Selatan yang dapat mempercepat proses pelayanan perijinan, baik pada bagian pendaftaran, pengawasan dan pengendalian, penetapan, pembayaran, penomoran sampai pada penyerahan izin. Tujuan penelitian ini adalah mengembangkan aplikasi Ijin Mendirikan Bangunan yang merupakan solusi dari kelemahan-kelemahan sistem yang ada dan berbasis komputerisasi. Studi kasus penelitian ini adalah di BP2T Kota Tangerang Selatan. Metodologi dalam mengembangkan aplikasi ini menggunakan Extreme Programming yang terdiri dari Planning, Design, Coding dan Testing. Hasil yang diperoleh dari pengembangan aplikasi ini adalah memudahkan masing-masing bagian terkait dalam hal membuat laporan-laporan karena telah terformat dan terstruktur secara dinamis, mulai dari pendaftaran sampai ke penerbitan ijin mendirikan bangunan (IMB).

**Kata Kunci:** aplikasi imb, *web*, Extreme Programming

## KATA PENGANTAR

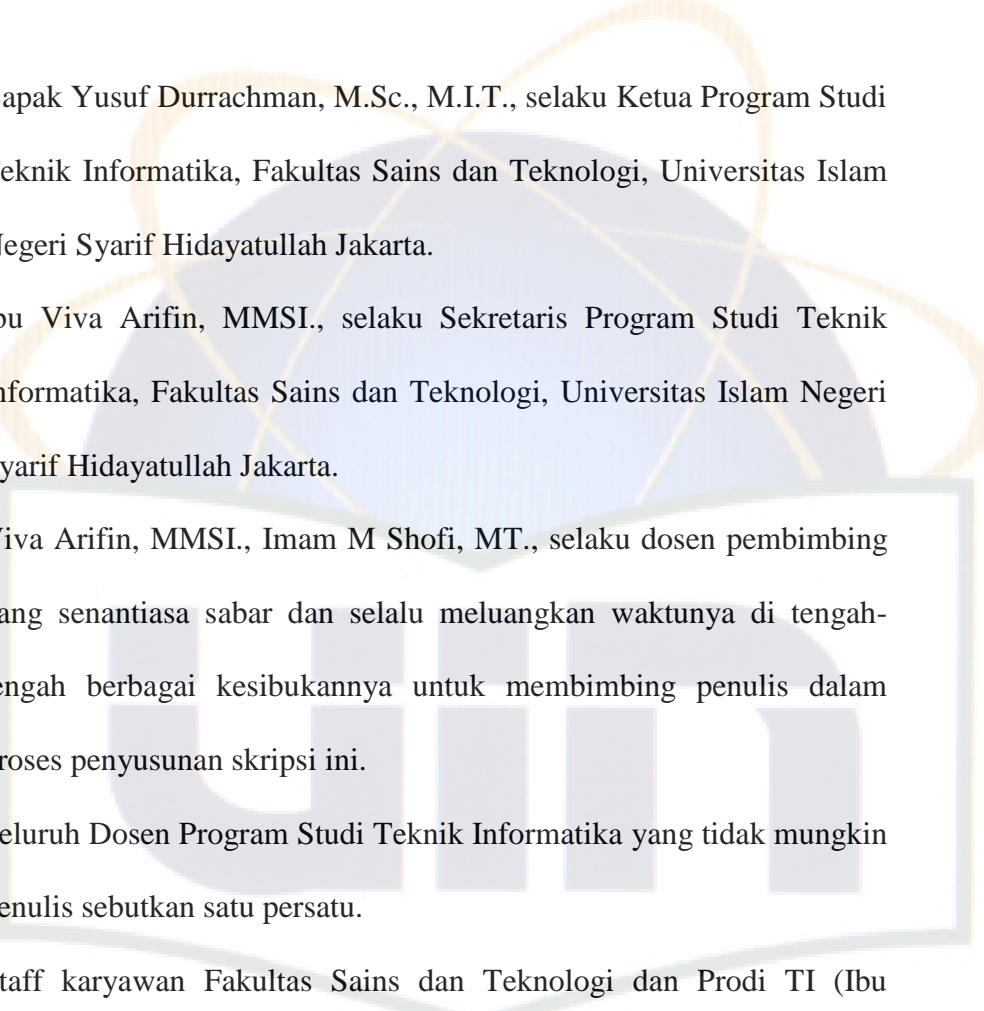
*Bismillaahirrahmaanirrahiim . . . . .*

Segala puji dan syukur penulis panjatkan kehadiran Allah SWT atas segala karunia, rahmat dan kekuatan, juga segala petunjuk dan kemudahan sehingga penulis dapat menyelesaikan penulisan skripsi ini dengan judul Pengembangan Aplikasi Pelayanan Izin Mendirikan Bangunan (Studi Kasus Pada Badan Pelayanan Perijinan Terpadu Kota Tangerang Selatan). Shalawat serta salam selalu kita haturkan kepada junjungan kita Nabi besar Muhammad SAW, beserta keluarganya, para sahabatnya, dan para pengikutnya.

Skripsi ini berjudul ***“Pengembangan Aplikasi Pelayanan Izin Mendirikan Bangunan (Studi Kasus Pada Badan Pelayanan Perijinan Terpadu Kota Tangerang Selatan)”***, yang disusun untuk memenuhi salah satu syarat dalam menyelesaikan program S1 pada Program Studi Teknik Informatika di Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Pada kesempatan yang berbahagia ini, penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam menyelesaikan skripsi ini. Mereka yang berdedikasi tinggi diantaranya:

1. Bapak DR. Syopiansyah Jaya Putra, M.Sis., selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.

- 
2. Bapak Yusuf Durrachman, M.Sc., M.I.T., selaku Ketua Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.
  3. Ibu Viva Arifin, MMSI., selaku Sekretaris Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.
  4. Viva Arifin, MMSI., Imam M Shofi, MT., selaku dosen pembimbing yang senantiasa sabar dan selalu meluangkan waktunya di tengah-tengah berbagai kesibukannya untuk membimbing penulis dalam proses penyusunan skripsi ini.
  5. Seluruh Dosen Program Studi Teknik Informatika yang tidak mungkin penulis sebutkan satu persatu.
  6. Staff karyawan Fakultas Sains dan Teknologi dan Prodi TI (Ibu Fitroh, Ibu Tari, Pak Amin, Pak Tommy, Niki, dan semuanya).

Akhirnya, penulis berharap semoga skripsi ini dapat bermanfaat. Saran dan kritik untuk kesempurnaan skripsi ini sangat penulis harapkan.

Jakarta, Mei 2011

Penulis

Wahyu Ari Kurniawan

## Teruntuk

Skripsi ini terkhusus penulis persembahkan kepada mereka yang telah mendukung, baik moril maupun materil, baik melalui doa ataupun sua dalam menyelesaikan skripsi ini.

1. Teruntuk Mama dan Bapa. Semoga Allah SWT selalu melimpahkan rahmat, rahim dan ampunan-Nya kepada mereka. Amin.
2. Teruntuk kakak-kakakku, Tri Purwanto, Indah Susanti, dan Ninik Handayani. Dukungan kalianlah yang selalu memberikan penulis motivasi untuk terus maju dan bertahan. Semoga kalian tidak pernah lelah untuk terus memberi penulis motivasi untuk menjadi yang lebih baik. Amin.
3. Teruntuk Pranaria Kartikaningsatya, SKG., selaku orang terdekat penulis. Terimakasih atas semangat, perhatian, motivasi dan pengorbanannya ^\_^.
4. Teruntuk teman-teman GAST, Andhika Sumanda, Indra Lesmana, Dino, Sentiya, Eti Nurdianti, Arief, Ari, Boim, Fasya, Hari, Alunk, Ozi, Alex, Mika dan Ayu. Senyum kalianlah yang selalu memberikan penulis semangat. Semoga penulis dapat menjadi teman terbaik, terimakasih semangatnya..
5. Teruntuk teman-teman satu perjuangan, Iche Berlyanti, Dimas Sony D, Lulu Mitha Sari, Anam, dan Ferry S. Terima kasih atas segala bantuan dan semangatnya.

6. Teruntuk Sarika, yang telah banyak membantu penulis dalam penggerjaan skripsi ini. Terimakasih untuk waktu dan ilmunya.
7. Teman-teman seperjuangan Teknik Informatika dan Sistem Informasi UIN angkatan 2006. Terima kasih untuk semua kenangan terindahnya.
8. Dan seluruh pihak yang telah membantu penulis, baik langsung maupun tidak langsung dalam menyelesaikan skripsi ini.



## DAFTAR ISI

Halaman

HALAMAN JUDUL .....	ii
Lembar Persetujuan Pembimbing .....	iii
Lembar Persetujuan Penguji .....	iv
Lembar Pernyataan .....	v
Abstrak .....	vi
Kata Pengantar .....	vii
Lembar Persembahan .....	ix
Daftar Gambar.....	xvii
Daftar Tabel.....	xx
Daftar Simbol.....	xxiv
Daftar Istilah.....	xxxii

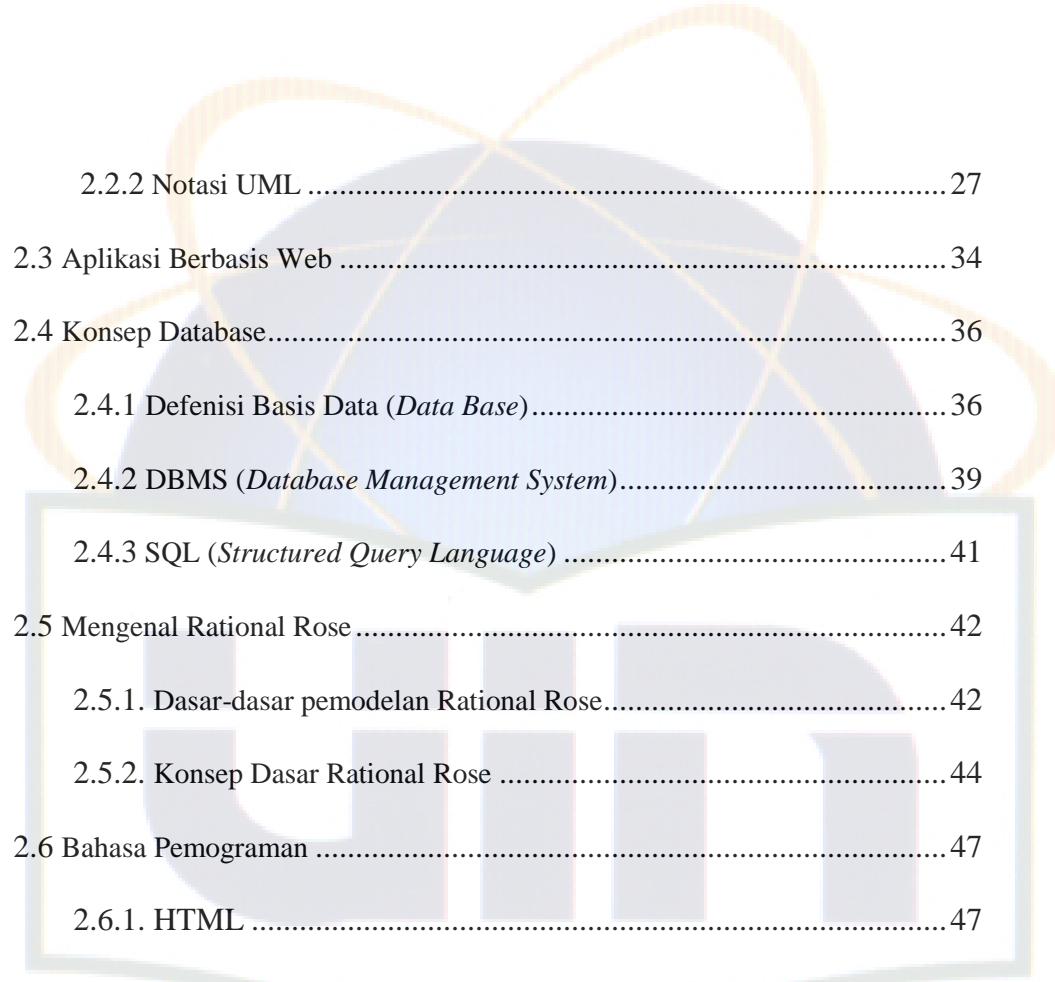
## BAB I PENDAHULUAN

1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian.....	4

1.5 Manfaat Penelitian .....	5
1.5.1 Bagi Pelayanan Perizinan Terpadu Kota Tangerang .....	5
1.5.2 Bagi Penulis.....	5
1.5.3 Bagi Akademik.....	6
1.6. Metodologi Penelitian .....	7
1.6.1 Metode Pengumpulan Data .....	7
1.6.2 Metode Pengembangan Aplikasi.....	8
1.7. Sistematika Penulisan .....	9

## BAB II LANDASAN TEORI

2.1. Badan Pelayanan Perizinan Terpadu (BP2T) Kota Tangerang Selatan .....	10
2.1.1. Gambaran Umum Badan Pelayanan Perizinan Terpadu (BP2T) Kota Tangerang Selatan.....	10
2.1.2. Dasar Hukum BP2T.....	11
2.1.3. Struktur Organisasi BP2T.....	12
2.1.4. Tugas Pokok dan Fungsi BP2T .....	13
2.1.5. Visi dan Misi BP2T .....	14
2.1.6. Pelayanan Perizinan oleh BP2T .....	15
2.1.7 Izin Mendirikan Bangunan (IMB) .....	16
2.2. UML (Unified Modelling Language).....	22
2.2.1 Definisi .....	22



2.2.2 Notasi UML .....	27
2.3 Aplikasi Berbasis Web .....	34
2.4 Konsep Database.....	36
2.4.1 Defenisi Basis Data ( <i>Data Base</i> ).....	36
2.4.2 DBMS ( <i>Database Management System</i> ).....	39
2.4.3 SQL ( <i>Structured Query Language</i> ) .....	41
2.5 Mengenal Rational Rose .....	42
2.5.1. Dasar-dasar pemodelan Rational Rose.....	42
2.5.2. Konsep Dasar Rational Rose .....	44
2.6 Bahasa Pemograman .....	47
2.6.1. HTML .....	47
2.6.2. PHP .....	48
2.6.3. CSS.....	50
2.6.4 JavaScript.....	51
2.6.5 <i>Firebird</i> .....	51
2.7 Mengenal Version Control.....	53
2.7.1 Subversion .....	54
2.8 Metode Pengembangan Sistem Extreme Programming .....	55
2.8.1. Index Card .....	61
2.8.2. Class Responsibility Colaborator .....	62
2.9. Pengujian Perangkat Lunak.....	62
2.10. Flowchart .....	65

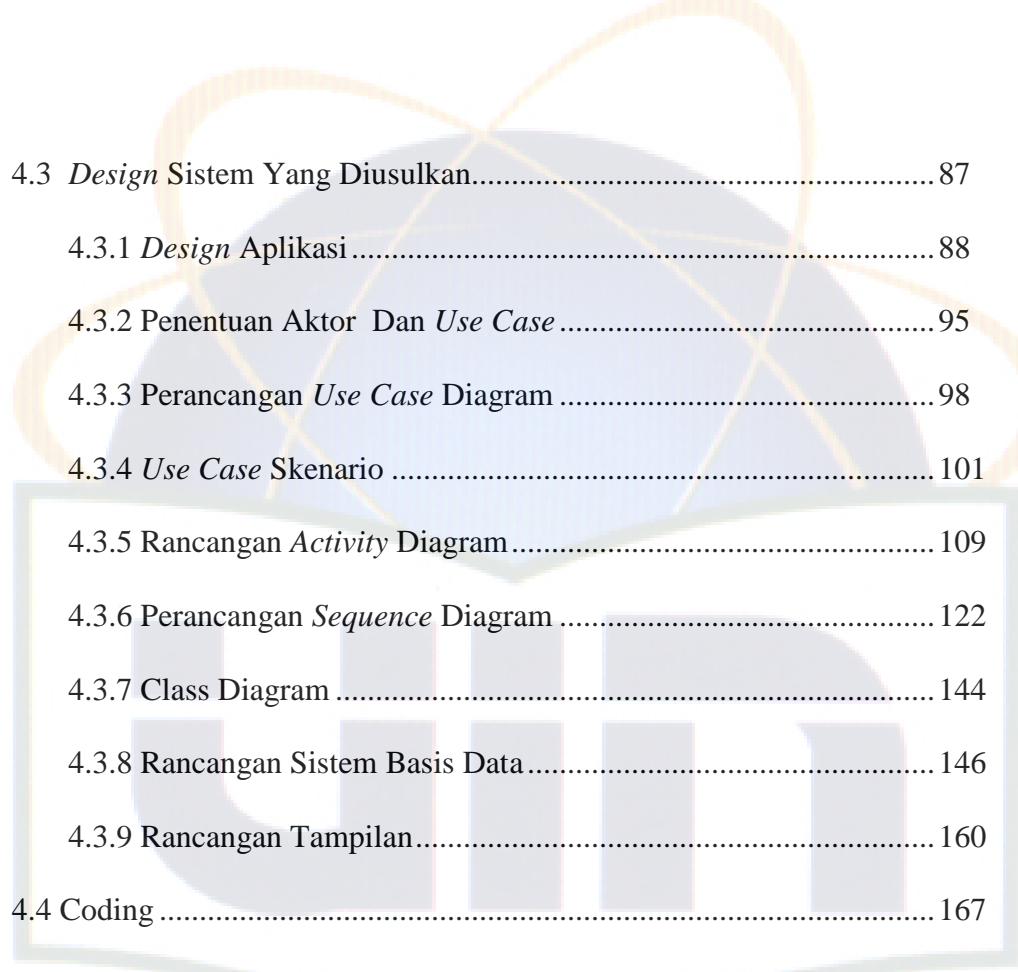
2.11. Studi Literatur Sejenis .....	66
-------------------------------------	----

### **BAB III METODOLOGI PENELITIAN**

3.1. Metode Pengumpulan Data.....	69
3.1.1. Observasi .....	69
3.1.2. Wawancara.....	70
3.1.3. Studi Pustaka.....	70
3.2. Metodologi Pengembangan Sistem.....	71
3.2.1 <i>Planning</i> .....	72
3.2.2 <i>Design</i> .....	72
3.2.3 <i>Coding</i> .....	74
3.2.4 <i>Testing</i> .....	75
3.3. Kerangka Berpikir.....	76

### **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

4.1. Sistem Yang Berjalan Pada Badan Pelayanan Perijinan Terpadu Kota Tangerang Selatan .....	77
4.1.1 Alur Kerja Sistem Berjalan.....	77
4.1.2 Identifikasi Masalah.....	79
4.2. Perencanaan ( <i>Planning</i> ) .....	79
4.2.1 Permintaan User ( <i>User Stories</i> ).....	79
4.2.2 Uraian Singkat Sistem yang Diusulkan.....	84



4.3 <i>Design</i> Sistem Yang Diusulkan.....	87
4.3.1 <i>Design</i> Aplikasi .....	88
4.3.2 Penentuan Aktor Dan <i>Use Case</i> .....	95
4.3.3 Perancangan <i>Use Case</i> Diagram .....	98
4.3.4 <i>Use Case</i> Skenario .....	101
4.3.5 Rancangan <i>Activity</i> Diagram.....	109
4.3.6 Perancangan <i>Sequence</i> Diagram .....	122
4.3.7 Class Diagram .....	144
4.3.8 Rancangan Sistem Basis Data .....	146
4.3.9 Rancangan Tampilan.....	160
4.4 Coding .....	167
4.4.1 Bahasa Pemrograman Dan Komponen .....	167
4.4.2 Implementasi <i>Coding</i> .....	167
4.5 Pengujian ( <i>Testing</i> ) .....	169
4.5.1 Pengujian Mandiri.....	170
4.5.2 Pengujian <i>All Users</i> .....	178

## BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan.....	179
5.2 Saran .....	180

## DAFTAR PUSTAKA .....

181

## LAMPIRAN-LAMPIRAN

## DAFTAR GAMBAR

Halaman

Gambar 2.1 Struktur Organisasi.....	13
Gambar 2.2 Notasi Aktor .....	28
Gambar 2.3 Notasi Kelas .....	29
Gambar 2.4 Notasi <i>Use Case</i> .....	30
Gambar 2.5 Notasi <i>Interaction</i> .....	31
Gambar 2.6 Notasi <i>Interface</i> .....	31
Gambar 2.7 Notasi <i>Packgae</i> .....	32
Gambar 2.8 Notasi <i>Note</i> .....	32
Gambar 2.9. Notasi <i>Dependency</i> .....	33
Gambar 2.10. Notasi Association .....	34
Gambar 2.11. Jenjang Dari Data.....	37
Gambar 2.12. Tampilan Dasar dalam Rational Rose.....	46
Gambar 2.13. Tahapan-tahapan Extreme Programming.....	58
Gambar 2.14. Simbol Flowchart .....	34
Gambar 3.1 Kerangka Berpikir .....	76
Gambar 4.1. Alur Kerja Sistem yang Berjalan pada proses pembuatan IMB .....	78
Gambar 4.2. Alur Kerja Sistem yang diusulkan pada proses pembuatan IMB. .	85
Gambar 4.3. <i>Use Case Diagram</i> Pendaftaran Pengajuan IMB.....	98
Gambar 4.4. <i>Use case Diagram</i> Pengawasan dan Pengendalian.....	99

Gambar 4.5. <i>Use case Diagram</i> Bendahara.....	99
Gambar 4.6. <i>Use case Diagram</i> Penomoran dan Penyerahan.....	100
Gambar 4.7. <i>Activity Diagram</i> pendaftaran .....	110
Gambar 4.8. <i>Activity Diagram</i> edit pendaftaran .....	111
Gambar 4.9. <i>Activity Diagram</i> pengawasan dan pengendalian .....	112
Gambar 4.10. <i>Activity Diagram</i> edit pengawasan dan pengendalian.....	123
Gambar 4.11. <i>Activity Diagram</i> penolakan .....	114
Gambar 4.12. <i>Activity Diagram</i> penetapan biaya .....	115
Gambar 4.13. <i>Activity Diagram</i> edit penetapan biaya.....	116
Gambar 4.14. <i>Activity Diagram</i> pembayaran .....	118
Gambar 4.15. <i>Activity Diagram</i> penomoran .....	119
Gambar 4.16. <i>Activity Diagram</i> Mencetak Surat Izin.....	120
Gambar 4.17. <i>Activity Diagram</i> penyerahan.....	121
Gambar 4.18. <i>Sequence Diagram</i> pendaftaran .....	123
Gambar 4.19. <i>Sequence Diagram</i> edit pendaftaran .....	125
Gambar 4.20. <i>Sequence Diagram</i> Pengawasan dan Pengendalian .....	127
Gambar 4.21. <i>Sequence Diagram</i> edit Pengawasan dan Pengendalian .....	129
Gambar 4.22. <i>Sequence Diagram</i> Penolakan .....	131
Gambar 4.23. <i>Sequence Diagram</i> Penetapan.....	133
Gambar 4.24. <i>Sequence Diagram</i> edit penetapan.....	135
Gambar 4.25. <i>Sequence Diagram</i> pembayaran.....	137
Gambar 4.26. <i>Sequence Diagram</i> penomoran .....	139

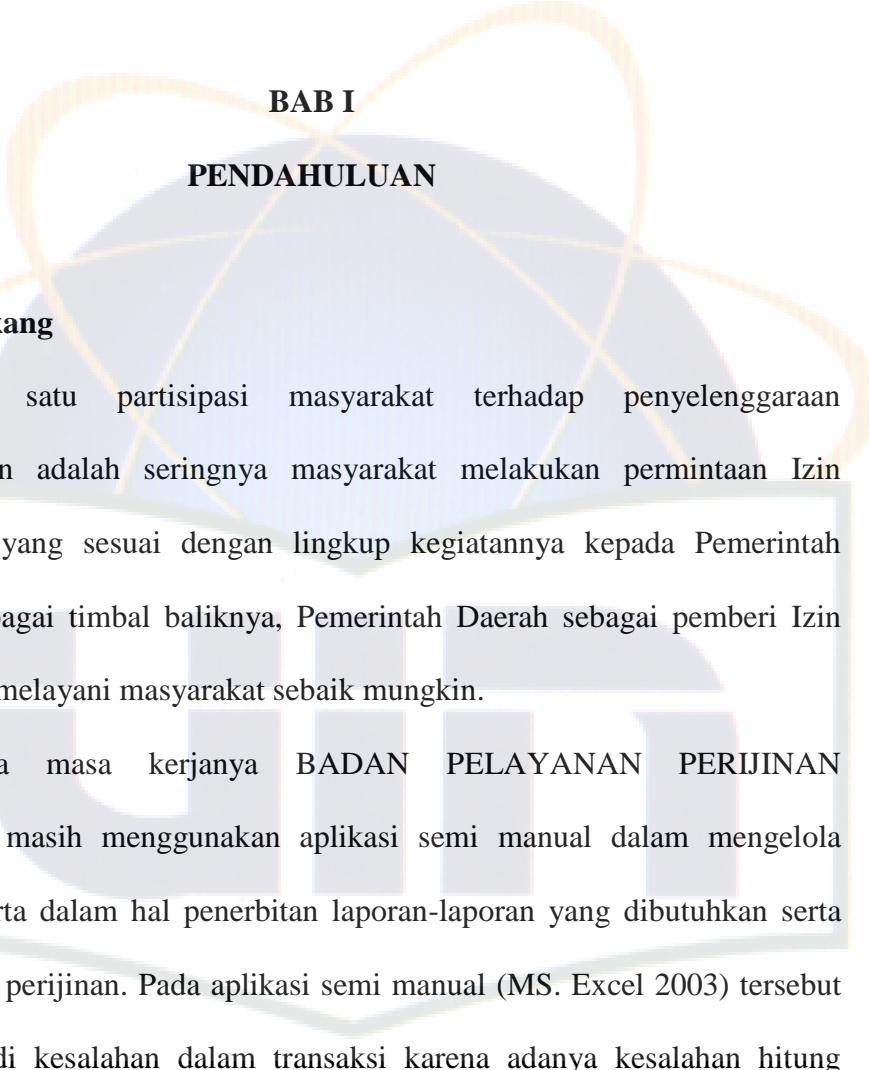
Gambar 4.27. <i>Sequence Diagram</i> mencetakn surat izin .....	141
Gambar 4.28. <i>Sequence Diagram</i> penyerahan.....	143
Gambar 4.29. <i>Class Diagram</i> untuk Sistem yang diusulkan .....	145
Gambar 4.30. <i>Database Relational (Physical Databasse Schema)</i> .....	147
Gambar 4.31. Rancangan Halaman <i>Login</i> .....	160
Gambar 4.32. Rancangan Halaman Utama.....	161
Gambar 4.33. Rancangan Pendaftaran tab permohonan.....	161
Gambar 4.34. Rancangan Pendaftaran tab pemohon .....	162
Gambar 4.35. Rancangan Pendaftaran tab komponen bangunan.....	162
Gambar 4.36. Rancangan Pendaftaran tab lokasi bangunan .....	163
Gambar 4.37. Rancangan Pemeriksaan.....	163
Gambar 4.38. Rancang Data Lapangan.....	164
Gambar 4.39. Rancangan Penetapan.....	165
Gambar 4.40. Rancangan Penomoran.....	165
Gambar 4.41. Rancangan Halaman Penyerahan.....	166
Gambar 4.42. Rancangan Halaman Pembayaran.....	166

## DAFTAR TABEL

	Halaman
Tabel 2.1. Jenis-jenis Perijinan .....	15
Tabel 3.1. Tabel Perbandingan Studi Literatur Sejenis dan Penulis.....	69
Tabel 4.1. <i>Index Card</i> Untuk Pendaftaran .....	80
Tabel 4.2. <i>Index Card</i> Untuk Pengawasan dan Pengednalian .....	82
Tabel 4.3. <i>Index Card</i> Untuk Penetapan Pembayaran Retribusi .....	83
Tabel 4.4. <i>Index Card</i> Untuk Penomoran dan Penyerahan.....	84
Tabel 4.5. <i>CRC Card Class</i> pendaftar.....	88
Tabel 4.6. <i>CRC Card Class</i> pemeriksaan.....	89
Tabel 4.7. <i>CRC Card Class</i> dataLapangan .....	90
Tabel 4.8. <i>CRC Card Class</i> rincianbangunan .....	91
Tabel 4.9. <i>CRC Card Class</i> user.....	92
Tabel 4.10. <i>CRC Card Class</i> penetapan.....	92
Tabel 4.11. <i>CRC Card Class</i> pembayaran.....	93
Tabel 4.12. <i>CRC Card Class</i> penomoran.....	93
Tabel 4.13. <i>CRC Card Class</i> penyerahan.....	94
Tabel 4.14. <i>CRC Card Class</i> report.....	94
Tabel 4.15. Tabel requirement dan use case pendaftaran IMB.....	95
Tabel 4.16. Tabel requirement dan use case Pengawasan dan Pengendalian.....	96
Tabel 4.17. Tabel requirement dan use case pada Bendahara .....	97

Tabel 4.18. Tabel requirement dan use case Penomoran dan Penyerahan .....	97
Tabel 4.19. <i>Use case scenario</i> Melakukan Pendataan Permohonan.....	101
Tabel 4.20. <i>Use case scenario</i> Lampiran Status .....	102
Tabel 4.21. <i>Use case scenario</i> Mengisi Data Pemohon.....	102
Tabel 4.22. <i>Use case scenario</i> Menentukan Komponen.....	103
Tabel 4.23. <i>Use case scenario</i> Menentukan Lokasi .....	103
Tabel 4.24. <i>Use case scenario</i> Mencetak Berkas Pemohon .....	104
Tabel 4.25. <i>Use case scenario</i> Memeriksa Permohonan .....	104
Tabel 4.26. <i>Use case scenario</i> Data Komponen Bangunan.....	104
Tabel 4.27. <i>Use case scenario</i> menolak permohonan.....	105
Tabel 4.28. <i>Use case scenario</i> Cetak Berkas Pemeriksaan.....	105
Tabel 4.29. <i>Use case scenario</i> Menetapkan Permohonan .....	106
Tabel 4.30. <i>Use case scenario</i> Mendata Pembayaran.....	106
Tabel 4.31. <i>Use case scenario</i> Cetak SKRD dan Nota Perhitungan .....	107
Tabel 4.32. <i>Use case scenario</i> Mengentry Penomoran.....	107
Tabel 4.33. <i>Use case scenario</i> Mencetak Surat Izin .....	108
Tabel 4.34. <i>Use case scenario</i> Mencatat Penyerahan.....	108
Tabel 4.35. Tabel M03_01 .....	148
Tabel 4.36. Tabel M03_02 .....	149
Tabel 4.37. Tabel M03_02_1 .....	150
Tabel 4.38. Tabel M03_03 .....	151
Tabel 4.39. Tabel M03_03_1 .....	151

Tabel 4.40. Tabel M04_04.....	152
Tabel 4.41. Tabel Pembayaran.....	153
Tabel 4.42. Tabel Penomoran .....	154
Tabel 4.43. Tabel Penyerahan.....	155
Tabel 4.44. Tabel MSETUP3_1.....	155
Tabel 4.45. Tabel MSETUP3_2.....	156
Tabel 4.46. Tabel MSETUP3_3.....	156
Tabel 4.47. Tabel MSETUP3_4.....	157
Tabel 4.48. Tabel MSETUP3_5.....	157
Tabel 4.49. Tabel master _pejabat .....	158
Tabel 4.50. Tabel master_rekening.....	158
Tabel 4.51. Tabel master_kecamatan.....	159
Tabel 4.52. Tabel master_desa.....	159
Tabel 4.53. Tabel Pengujian Mandiri.....	170



## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Salah satu partisipasi masyarakat terhadap penyelenggaraan pemerintahan adalah seringnya masyarakat melakukan permintaan Izin (Perizinan) yang sesuai dengan lingkup kegiatannya kepada Pemerintah Daerah. Sebagai timbal baliknya, Pemerintah Daerah sebagai pemberi Izin harus dapat melayani masyarakat sebaik mungkin.

Selama masa kerjanya BADAN PELAYANAN PERIJINAN TERPADU masih menggunakan aplikasi semi manual dalam mengelola perijinan serta dalam hal penerbitan laporan-laporan yang dibutuhkan serta pembayaran perijinan. Pada aplikasi semi manual (MS. Excel 2003) tersebut sering terjadi kesalahan dalam transaksi karena adanya kesalahan hitung (perhitungan masih manual diatas kertas) karena melihat dari perbedaan dalam perhitungan dan juga koefisien-koefisien yang dibutuhkan dari pada perjinian IMB (Izin Mendirikan Bangunan), sering ditemukan ketidaksesuaian data mulai proses pendaftaran hingga proses penerbitan ijin pada masing - masing bagian, pengulangan input data pemohon pada masing - masing bagian (bagian pendaftaran, pengawasan dan pengendalian, penetapan, validasi) karena tidak terkomputerisasi dengan baik. Dari hasil wawancara dengan pihak terkait, salah satunya dengan Kabid Pelayanan Perijinan Bidang Pembangunan yaitu Bapak Severul Cahyo Kuntadi S.Pd, maka perlu dibuat

suatu aplikasi untuk mengatasi permasalahan. Aplikasi tersebut tersebut terkomputerisasi dengan baik yang dapat mengurangi kekurangan pada pelayanan secara semi manual sehingga dapat meningkatkan kualitas pelayanan yang lebih baik.

Dalam rangka kualitas pelayanan yang lebih baik, maka penulis tertarik membuat judul **“Pengembangan Aplikasi Pelayanan Izin Mendirikan Bangunan (Studi Kasus Pada Badan Pelayanan Perijinan Terpadu Kota Tangerang Selatan)”**.

Alasan pemilihan pada Kota Tangerang Selatan karena Kota Tangerang Selatan masih terbilang baru dan merupakan daerah pemekaran dari Kabupaten Tangerang yang diresmikan pada 29 Oktober 2008 sedangkan BADAN PELAYANAN PERIJINAN TERPADU Kota Tangerang Selatan merupakan bentuk organisasi baru, berdasarkan Peraturan Walikota Tangerang Selatan No. 32 Tahun 2009, sehingga penulis ingin membantu mengatasi permasalahan – permasalahan tersebut pada Izin Mendirikan Bangunan.

## **1.2. Perumusan Masalah**

Atas dasar permasalahan yang dipaparkan pada latar belakang, maka perumusan masalah penelitian ini adalah :

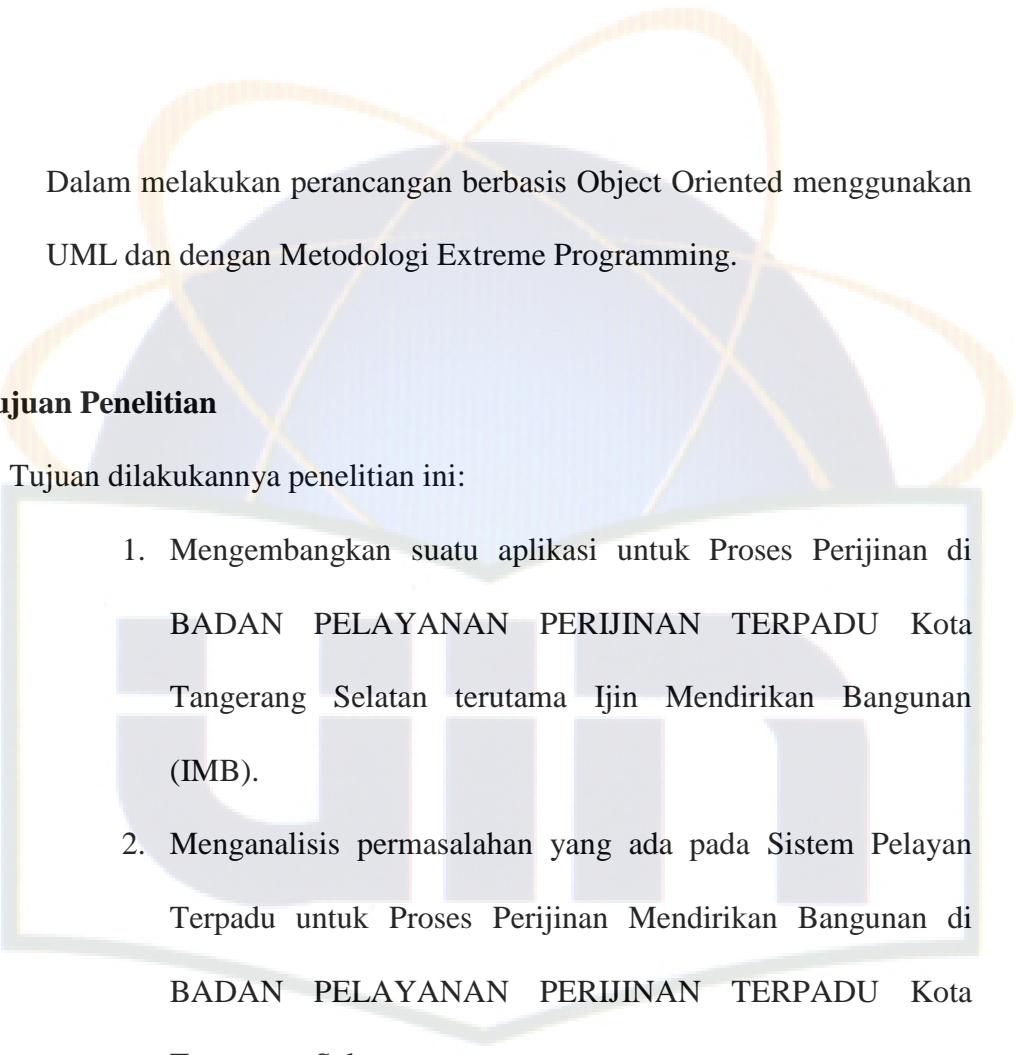
1. Bagaimana mengembangkan suatu aplikasi pelayanan Ijin Mendirikan Bangunan (IMB) yang sesuai tujuan/ sasaran dalam penyelesaian perijinan di BADAN PELAYANAN PERIJINAN TERPADU Kota

Tangerang Selatan yang dapat mempercepat proses pelayanan perijinan, baik pada bagian pendaftaran, pengawasan dan pengendalian, penetapan, pembayaran, penomoran sampai pada penyerahan izin.

### 1.3. Pembatasan Masalah

Penulis membatasi masalah yang ada pada penelitian ini pada beberapa hal berikut :

1. Perancangan database dengan menggunakan *firebird* 2.0 sebagai *tools*-nya.
2. Perancangan aplikasi berbasis *web* menggunakan bahasa pemrograman HTML dan PHP versi 5.2.6
3. Perancangan dan pembuatan fitur atau modul aktifitas untuk pendaftaran, pemeriksaan, penetapan, modul daftar perijinan, modul pembayaran, modul penomoran SK.
4. Penyajian Laporan (*report*) yang dibuat disesuaikan dengan kebutuhan BADAN PELAYANAN PERIJINAN TERPADU adalah Kartu Daftar, Berita Acara Pemeriksaan Lapangan (BAPL), Nota Perhitungan, Surat Ketetapan Retribusi Daerah (SKRD), Surat Perijinan.
5. Pengguna sistem ini dibagi menjadi level pengguna aplikasi atau hak aksesnya hanya diberikan kepada bagian pelayanan, pemeriksaan, bendahara, validasi dan administrator.
6. Desain *Interface* aplikasi ini menggunakan YUI (*Yahoo User Interface Library*) versi 2.5.1

- 
7. Dalam melakukan perancangan berbasis Object Oriented menggunakan UML dan dengan Metodologi Extreme Programming.

#### **1.4. Tujuan Penelitian**

Tujuan dilakukannya penelitian ini:

1. Mengembangkan suatu aplikasi untuk Proses Perijinan di BADAN PELAYANAN PERIJINAN TERPADU Kota Tangerang Selatan terutama Ijin Mendirikan Bangunan (IMB).
2. Menganalisis permasalahan yang ada pada Sistem Pelayan Terpadu untuk Proses Perijinan Mendirikan Bangunan di BADAN PELAYANAN PERIJINAN TERPADU Kota Tangerang Selatan.
3. Membantu BADAN PELAYANAN PERIJINAN TERPADU Kota Tangerang Selatan dalam pengintegrasian perijinan mendirikan bangunan.
4. Tersedianya Aplikasi Ijin Mendirikan Bangunan yang sesuai dengan keinginan user.

## 1.5. Manfaat Penelitian

### 1.5.1. Bagi Pelayanan Perijinan Terpadu Kota Tangerang

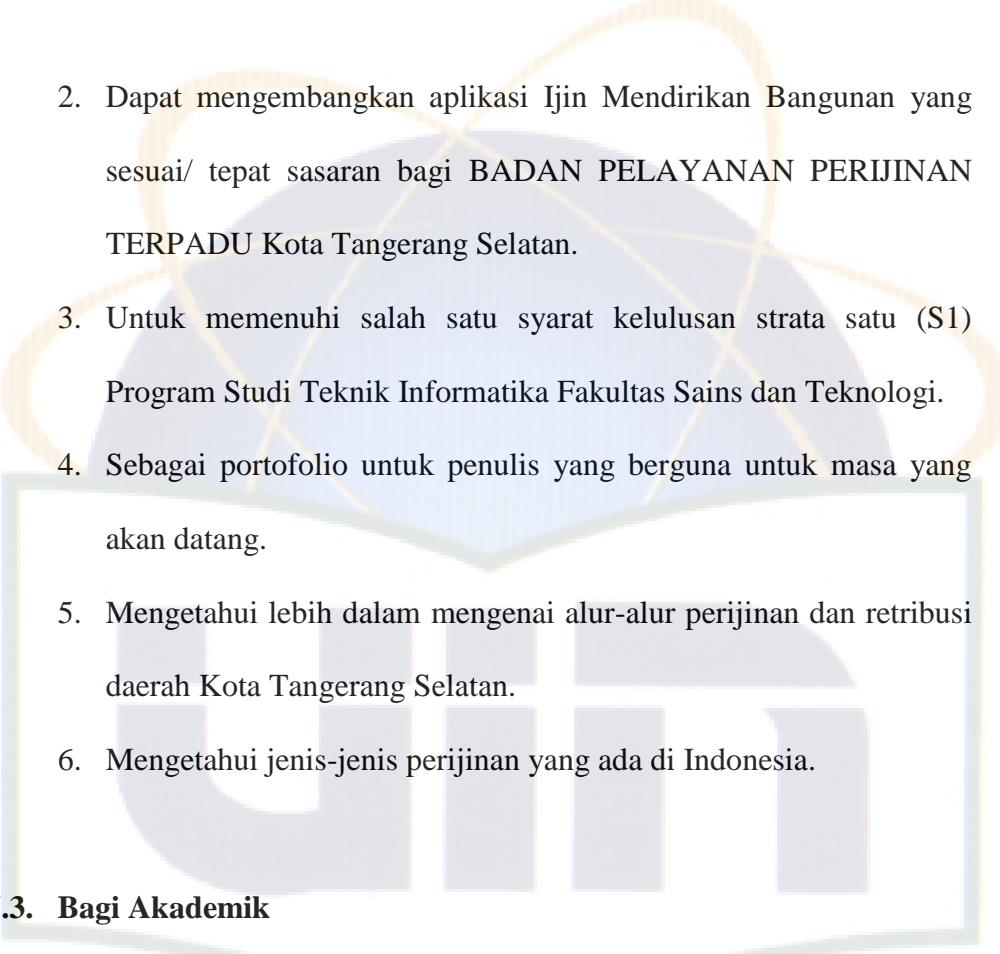
Manfaat yang dapat dipetik oleh Bagian Pelayanan Terpadu Kota Tangerang dari penelitian ini antara lain:

1. Sebagai model atau contoh aplikasi yang kompleks dan solusi yang efektif dalam menyelesaikan permasalahan perijinan.
2. Sebagai bahan evaluasi dalam implementasi sistem informasi pada pelayanan terpadu perijinan.
3. Mempermudah BPPT dalam mengelola perijinan secara terkomputerisasi.
4. Jalannya sistem dapat terkoordinasi dengan baik antara tiap-tiap bagian (Pelayanan, Pemeriksaan, Penetapan, Bendahara, Penomoran, Validasi).
5. Mempermudah dalam penerbitan Ijin maupun laporan-laporan lainnya.
6. BPPT dapat meningkatkan pelayanan kepada pelanggannya.

### 1.5.2. Bagi Penulis

Banyak sekali manfaat yang dapat penulis petik dalam penelitian skripsi ini sebagai berikut :

1. Memahami proses kerja aplikasi dan implementasinya yang sesuai dengan sasaran/ tujuan.

- 
2. Dapat mengembangkan aplikasi Ijin Mendirikan Bangunan yang sesuai/ tepat sasaran bagi BADAN PELAYANAN PERIJINAN TERPADU Kota Tangerang Selatan.
  3. Untuk memenuhi salah satu syarat kelulusan strata satu (S1) Program Studi Teknik Informatika Fakultas Sains dan Teknologi.
  4. Sebagai portofolio untuk penulis yang berguna untuk masa yang akan datang.
  5. Mengetahui lebih dalam mengenai alur-alur perijinan dan retribusi daerah Kota Tangerang Selatan.
  6. Mengetahui jenis-jenis perijinan yang ada di Indonesia.

#### **1.5.3. Bagi Akademik**

1. Mengetahui kemampuan mahasiswa dalam penguasaan materi dan penerapan ilmu yang telah di dapat di bangku kuliah.
2. Memberikan gambaran tentang kesiapan mahasiswa dalam menghadapi dunia kerja dari hasil yang diperoleh selama pembelajaran pada masa kuliah.

## 1.6. Metodologi Penelitian

Metode penelitian yang digunakan dan diterapkan dalam penulisan tugas akhir ini meliputi :

### 1.6.1. Metode Pengumpulan Data

Dalam proses penulisan skripsi ini, bentuk metodologi pelaksanaan yang akan dilakukan antara lain :

1. Observasi, Observasi adalah suatu penyelidikan yang dijalankan secara sistematis dan sengaja diadakan dengan menggunakan alat indra terutama mata terhadap kejadian-kejadian yang langsung (Bimo Walgito, 1987:54)
2. Wawancara, yakni mengumpulkan data melalui tanya jawab dan diskusi dengan pihak-pihak yang terlibat (Jogiyanto, 2005).
3. Studi Pustaka, yakni penulisan kepustakaan yang bertujuan untuk mengumpulkan data dan informasi dengan bantuan bermacam-macam material yang terdapat dalam ruang pustaka, misalnya berupa buku-buku, majalah-majalah, naskah-naskah, catatan, dokumen, internet skripsi, tesis dan lain-lain (Kartono, 1990).
4. Studi Literatur, merupakan merupakan identifikasi, lokasi dan analisis dari dokumen yang berisi informasi yang berhubungan dengan permasalahan penelitian secara sistematis (Kuncoro, 2003: 28).

### 1.6.2. Metode Pengembangan Aplikasi

Metode pengembangan sistem yang penulis gunakan dalam penelitian ini adalah metode *Extreme Programming* (XP) oleh *Kent Back*. Tahapan-tahapan pengembangan aplikasi ini adalah :

1. *Planning*, tahapan ini adalah tahapan perencanaan pada sistem yang akan dikembangkan. Pada tahapan ini penulis melakukan:

Pengumpulan permintaan *user (user stories)*, Pengamatan terhadap sistem yang sedang berjalan di Badan Pelayanan Perijinan Terpadu (BP2T), menentukan alur bisnis dan aplikasi serta wilayah persoalan data yang akan didukung oleh sistem yang akan dikembangkan serta ditentukan pula jangkauan atau batasan sistem.

2. *Design*, Setelah mengetahui definisi aplikasi yang akan dikembangkan maka tahapan berikutnya adalah melakukan perancangan (*design*). *Design* ini meliputi *design* aplikasi, *design* basis data, dan *design* tampilan.
3. *Coding*, Pada tahapan *coding*, sebelum dilakukan *developing* aplikasi berdasarkan tahapan-tahapan sebelumnya, *unit test* untuk setiap *user stories*, kemudian dilakukan pembangunan aplikasi berdasarkan design yang sebelumnya telah dilakukan.
4. *Testing*, testing dilakukan pada *unit test* yang sebelumnya telah disiapkan, yaitu dengan mengetes *source code* yang digunakan. Setelah unit test dilakukan dilakukan *acceptance test* dengan

melakukan *deployment* terhadap aplikasi yang telah dikembangkan ke target server tertentu.

### **1.7. Sistematika Penulisan**

Dalam skripsi ini, pembahasan yang penulis sajikan terbagi dalam lima bab, yang secara singkat akan diuraikan sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini membahas tentang latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

#### **BAB II LANDASAN TEORI**

Bab ini membahas secara singkat teori yang diperlukan dalam penelitian skripsi.

#### **BAB III METODOLOGI PENELITIAN**

Pada bab ini akan dijelaskan metodologi yang digunakan penulis dalam melakukan penelitian.

#### **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Dalam bab ini diuraikan hasil analisis dan perancangan sistem yang dibuat.

#### **BAB V KESIMPULAN DAN SARAN**

Bab ini adalah bab terakhir yang menyajikan kesimpulan serta saran dari apa yang telah diterangkan dan diuraikan pada bab-bab sebelumnya



## BAB II

### LANDASAN TEORI

#### 2.1. Badan Pelayanan Perizinan Terpadu (BP2T) Kota Tangerang Selatan

##### 2.1.1. Gambaran Umum Badan Pelayanan Perizinan Terpadu (BP2T) Kota Tangerang Selatan

Badan Pelayanan Perizinan Terpadu (BP2T) adalah kegiatan penyelenggaraan perijinan dan non perijinan yang proses pengelolaannya mulai dari tahap permohonan sampai ke tahap terbitnya dokumen dilakukan dalam satu tempat. Atau dengan kata lain terjadinya pemangkasan tahapan dan prosedure, Transparansi biaya, Penyederhanaan Persyaratan, Pengurangan waktu rata-rata dalam proses perijinan dan Pemberian hak kepada masyarakat untuk memperoleh informasi dalam kaitannya dengan penyelenggaraan pelayanan (PERATURAN MENTERI DALAM NEGERI NOMOR, 2008).

Birokasi perijinan merupakan salah satu permasalahan yang menjadi kendala bagi perkembangan usaha di Indonesia. Kondisi pelayanan perijinan saat ini masih dihadapkan pada sistem yang belum efektif dan efisien serta belum sesuai dengan tuntutan masyarakat. Hal ini dapat dilihat dari banyaknya pengaduan dan keluhan dari masyarakat baik secara langsung maupun tidak

langsung mengenai kinerja aparatur dan banyaknya peraturan yang tumpang tindih, prosedur yang berbelit-belit, tidak ada kepastian jangka waktu penyelesaian, tingginya biaya yang harus dikeluarkan, banyaknya persyaratan yang harus dipenuhi, sikap petugas yang kurang *responsive*, sarana yang kurang menunjang dan lain-lain, sehingga menimbulkan citra yang kurang baik terhadap kinerja Pemerintah Daerah. Untuk mengatasi kondisi tersebut perlu dilakukan upaya perbaikan kualitas berkesinambungan demi mewujudkan pelayanan publik yang prima. Upaya perbaikan kualitas pelayanan perijinan dilakukan melalui serangkaian regulasi kebijakan sebagai wujud reformasi birokrasi pelayanan publik yang telah dicanangkan oleh Pemerintah Kota Tangerang Selatan. Oleh sebab itu, dibentuklah Badan Pelayanan Perijinan Terpadu (BP2T, 2010).

BP2T Kota Tangerang Selatan memiliki motto : “*Kepastian Ijin dengan Tidak Mempermudah dan Tidak Mempersulit*”

### **2.1.2. Dasar Hukum BP2T**

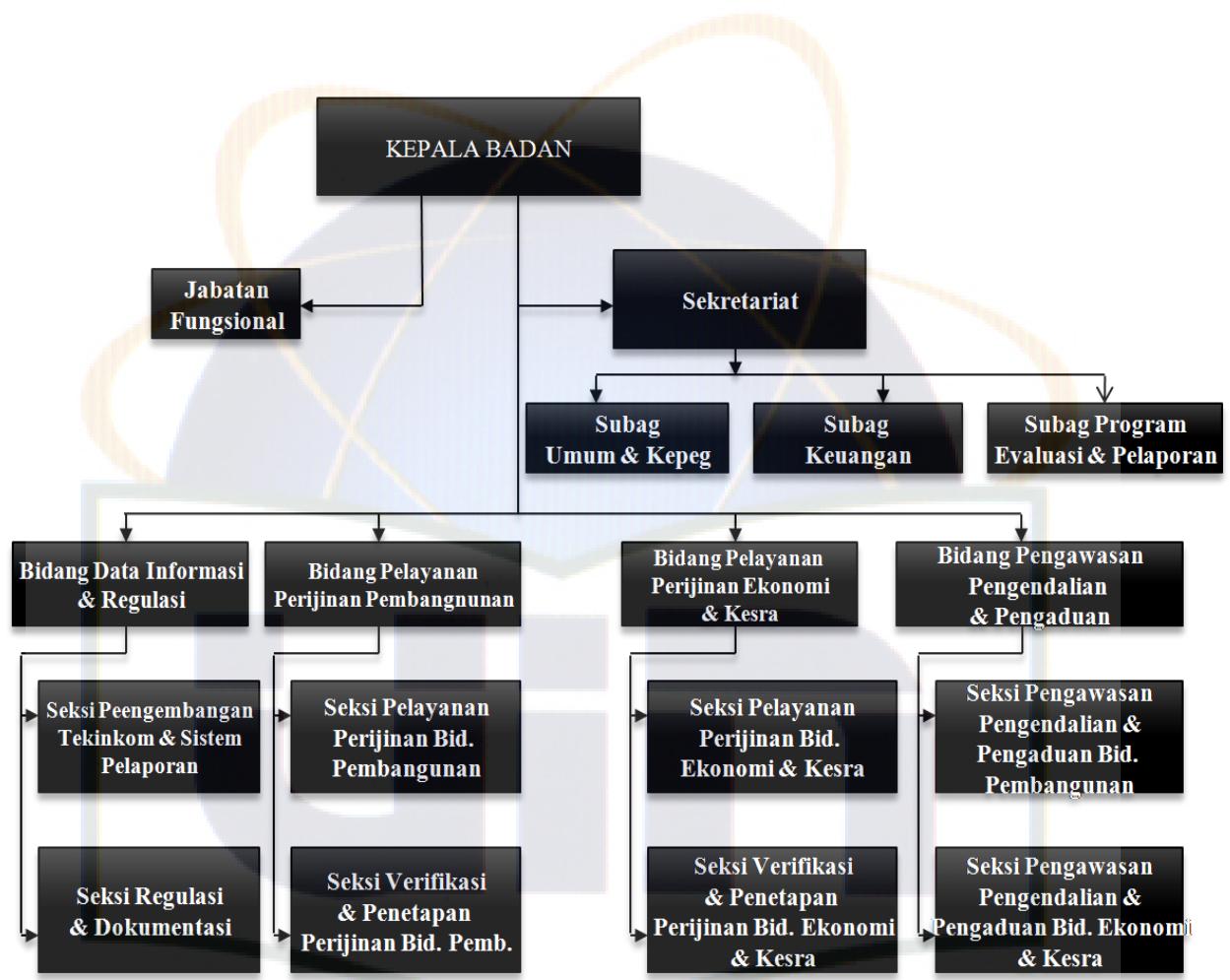
1. Undang-Undang Nomor 51 Tahun 2008 Tentang Pembentukan Kota Tangerang Selatan.
2. Peraturan Walikota Tangerang Selatan Nomor 07 Tahun 2009 Tentang perubahan Atas Peraturan Walikota Tangerang Selatan

Nomor 01 Tahun 2009 Tentang Organisasi Perangkat Daerah  
Kota Tangerang Selatan.

3. Peraturan Walikota Tangerang Selatan Nomor 32 Tahun 2009  
Tentang Rincian Tugas dan Fungsi Badan Pelayanan Perijinan  
Terpadu Kota Tangerang Selatan.
4. Peraturan Walikota Tangerang Selatan Nomor 46 Tentang  
Pemberlakuan Peraturan Daerah Kabupaten Tangerang dan  
Peraturan Bupati Tangerang di Kota Tangerang Selatan.
5. Peraturan Walikota Tangerang Selatan Nomor 47 Tahun 2009  
Tentang Penyelenggaraan Perijinan.
6. Surat Menteri Dalam Negeri No. 61/2671/SJ Tanggal 17 Juli  
2009 Tentang Pelaksanaan Penyelenggaraan Perijinan Kota  
Tangerang Selatan.

### **2.1.3. Struktur Organisasi BP2T**

Peraturan Walikota No. 7 Tahun 2009 tentang Perubahan Atas  
Peraturan Walikota tentang Organisasi Perangkat daerah Kota  
Tangerang Selatan.



**Gambar 2.1.** Struktur Organisasi

#### 2.1.4. Tugas Pokok dan Fungsi BP2T

Peraturan Walikota Tangerang Selatan No. 32 Tahun 2009 tentang Rincian Tugas Fungsi dan Tata Kerja Badan Pelayanan Perijinan Terpadu Kota Tangerang Selatan.

Tugas Pokok BP2T Tangerang Selatan yaitu Melaksanakan koordinasi dan penyelenggaraan serta pelayanan administrasi dibidang perijinan dan non perijinan secara terpadu dengan prinsip

koordinasi, integrasi, sinkronisasi, simplifikasi, keamanan dan kepastian.

Adapun Fungsi BP2T Tangerang Selatan Yaitu :

1. Penyusunan program Badan Pelayanan Perijinan Terpadu;
2. Penyelenggaraan Perijinan;
3. Pelaksanaan koordinasi proses pelayanan perijinan;
4. Pelaksanaan administrasi pelayanan perijinan;
5. Pengawasan dan evaluasi pelayanan perijinan.

#### **2.1.5. Visi dan Misi BP2T**

Visi BP2T Tangerang Selatan: “**Terwujudnya Pelayanan Prima Tahun 2015**”

Misi BP2T Tangerang Selatan :

1. Meningkatkan Kualitas Pelayanan Perijinan Yang Cepat, Tepat, Akurat Dan Akuntabel.
2. Meningkatkan Kualitas Aparatur Yang Profesional Dalam Melayani Masyarakat Di Bidang Perijinan.
3. Meningkatkan Sarana Dan Prasarana Perijinan Yang Memadai.
4. Meningkatkan Koordinasi Dan Kerjasama Antar SKPD Terkait Guna Mempercepat Proses Perijinan.
5. Meningkatkan Kesadaran Masyarakat Dalam Perijinan.

### 2.1.6. Pelayanan Perizinan oleh BP2T

Dasar Hukum Jenis Perijinan yang dilayani oleh BP2T  
Tangerang Selatan :

1. Peraturan Wali Kota Tangerang Selatan Nomor 46 Tahun 2009 tentang Pemberlakuan Peraturan Daerah Kabupaten Tangerang dan Peraturan Bupati Tangerang di Kota Tangerang Selatan ;
2. Peraturan Wali Kota Tangerang Selatan Nomor 47 Tahun 2009 tentang Penyelenggaraan Perijinan ;
3. Sejak Agustus 2009 Pelayanan Perijinan Terpadu mulai dikelola Badan Pelayanan Perijinan terpadu, baru 20 ijin yang dikelola ;
4. Sebanyak 64 ijin masih dikelola 7 (tujuh) SKPD Lainnya.

Adapun Jenis Perijinan dan Non Perijinan yang dikelola pada BP2T Tangerang Selatan yaitu:

**Tabel 2.1.** Jenis-jenis perizinan

<b>Perijinan</b>	<b>Non Perijinan</b>
<ol style="list-style-type: none"> <li>1. <u>Ijin Mendirikan Bangunan (IMB)</u></li> <li>2. <u>Ijin Pengadaan Ruang (IPR)</u></li> <li>3. <u>Upaya Pemantauan Lingkungan/Kelola Lingkungan (UPL/UKL).</u></li> <li>4. <u>Rencana Tapak (Site Plane).</u></li> </ol>	<ol style="list-style-type: none"> <li>1. Tanda Daftar Perusahaan CV (TDP).</li> <li>2. <u>Surat Keterangan Pengesahan Pendirian Koperasi.</u></li> <li>3. <u>Surat Ijin Usaha Konstruksi (SIUK).</u></li> <li>4. <u>Ijin Usaha Waralaba.</u></li> <li>5. <u>Ijin Usaha Perdagangan (IUP).</u></li> </ol>

<p>5. <u>Ijin Lokasi (IL).</u></p> <p>6. <u>Ijin Layak Huni (ILH).</u></p> <p>7. <u>Ijin Gangguan (HO).</u></p>	<p>6. <u>Ijin Usaha Kepariwisataan (Baru).</u></p> <p>7. <u>Ijin Usaha Kepariwisataan (Daftar Ulang).</u></p> <p>8. <u>Ijin Usaha Industri.</u></p> <p>9. <u>Ijin Usaha Industri Kecil.</u></p> <p>10. <u>Ijin Reklame.</u></p> <p>11. <u>Ijin Penyelenggaraan Kursus dan Kelembagaan.</u></p> <p>12. <u>Ijin Penyelenggaraan Parkir.</u></p> <p>13. <u>Ijin Lembaga Bursa Kerja.</u></p> <p>14. <u>Ijin Operasional Perusahaan Penyediaan Jasa Pekerja.</u></p> <p>15. <u>Ijin Gudang.</u></p>
---	---

#### **2.1.7. Izin Mendirikan Bangunan (IMB)**

Ijin Mendirikan Bangunan (IMB) adalah izin yang diberikan oleh Pemerintah Daerah kepada orang pribadi atau badan yang dimaksudkan agar desain, pelaksanaan pembangunan dan bangunan sesuai dengan Rencana Tata Ruang yang berlaku, sesuai dengan Koefisien Dasar Bangunan (KDB), Koefisien Lantai Bangunan (KLB), Ketinggian Bangunan yang ditetapkan dan sesuai dengan syarat-syarat keselamatan bangunan yang menempati bangunan tersebut (PERATURAN DAERAH TENTANG PERUBAHAN ATAS PERATURAN DAERAH NOMOR 10 TAHUN 2001 TENTANG IZIN MENDIRIKAN BANGUNAN, 2006).

Adapun Dasar Hukum Ijin Mendirikan Bangunan Pada Kota Tangerang Selatan yaitu :

1. Undang-Undang Nomor 18 Tahun 1997 tentang Pajak Daerah dan Retribusi Daerah (Lembaran Negara Republik Indonesia Tahun 1997 Nomor 41, Tambahan Lembaran Negara Republik Indonesia Nomor 3685) sebagaimana telah diubah dengan Undang-Undang Nomor 34 Tahun 2000 tentang Perubahan Atas Undang-undang Republik Indonesia Nomor 18 Tahun 1997 tentang Pajak Daerah dan Retribusi Daerah (Lembaran Negara Republik Indonesia Tahun 1997 Nomor 34, Tambahan Lembaran Negara Republik Indonesia Nomor 4048);
2. Undang-Undang Nomor 23 Tahun 1997 tentang Pengelolaan Lingkungan Hidup (Lembaran Negara Republik Indonesia Tahun 997 Nomor 68 Tambahan Lembaran Negara Republik Indonesia 699);
3. Undang-Undang Nomor 23 Tahun 2000 tentang Pembentukan Ropinsi Banten (Lembaran Negara Republik Indonesia Tahun 2000 Nmor 182 Tambahan Lembaran Negara Republik Indonesia Nomor 010);
4. Undang-undang Nomor 28 Tahun 2002 tentang Bangunan Gedung Lembaran Negara Republik Indonesia Tahun 2002 Nomor 34, ambahan Lembaran Negara Republik Indonesia Tahun 2002 Nomor 247);
5. Undang-Undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah Lembaran Negara Republik Indonesia Tahun 2004

- Nomor 125, ambahan Lembaran Negara Republik Indonesia 4437) sebagaimana elah dua kali diubah terakhir dengan Undang-undang Nomor 12 ahun 2008 tentang Perubahan Kedua atas Undang-undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah (Lembaran Negara Republik Indonesia Tahun 2008 Nomor 59, Tambahan Lembaran Negara Republik Indonesia Nomor 4844);
6. Undang-Undang Nomor 26 Tahun 2007 tentang Penataan Ruang Lembaran Negara Republik Indonesia Tahun 2007 Nomor 68 Tambahan Lembaran Negara Republik Indonesia Nomor 4725);
  7. Undang-Undang Nomor 51 Tahun 2008 tentang Pembentukan Kota Tangerang Selatan di Provinsi Banten (Lembaran Negara Republik Indonesia Tahun 2008 Nomor 188, Tambahan Lembaran Negara Republik Indonesia 4935);
  8. Peraturan Pemerintah Nomor 66 Tahun 2001 tentang Retribusi Daerah (Lembaran Negara Republik Indonesia Tahun 2001 Nomor 119; Tambahan Lembaran Negara Republik Indonesia Nomor 4139);
  9. Peraturan Pemerintah Nomor 38 Tahun 2007 tentang Pembagian Urusan Pemerintah, Pemerintah Daerah Provinsi dan Pemerintahan Daerah Kabupaten/ Kota (Lembaran Negara

- Republik Indonesia Tahun 2007 Nomor 82, Tambahan Lembaran Negara Republik Indonesia Nomor 4737);
10. Peraturan Daerah Kabupaten Daerah Tingkat II Tangerang Nomor 3 Tahun 1996 tentang Rencana Tata Ruang Wilayah Kabupaten Daerah Tingkat II Tangerang (Lembaran Daerah Kabupaten Daerah Tingkat II Tangerang Tahun 1996 Nomor 1 Seri C) sebagaimana telah dua kali diubah terakhir dengan Peraturan Daerah Kabupaten Tangerang Nomor 3 Tahun 2008 tentang Perubahan Kedua Atas Peraturan Daerah Kabupaten Daerah Tingkat II Tangerang Nomor 3 Tahun 1996 tentang Rencana Tata Ruang Wilayah Kebupaten Daerah Tingkat II Tangerang (Lembaran Daerah Kabupaten Tangerang Tahun 2008 Nomor 3, Tambahan Lembaran Daerah Kabupaten Tangerang Nomor 0308);
11. Peraturan Daerah Kabupaten Tangerang Nomor 10 Tahun 2001 tentang Izin Mendirikan bangunan (Lembaran Daerah Kabupaten Tangerang Tahun 2001 Nomor 10, Tambahan Lembaran Daerah Nomor 1001) sebagaimana telah diubah dengan Peraturan Daerah Kabupaten Tangerang Nomor 10 Tahun 2006 (Lembaran Daerah Kabupaten Tangerang Tahun 2006, Tambahan Lembaran Daerah Kabupaten Tangerang Nomor 1006);

12. Peraturan Daerah Kabupaten Tangerang Nomor 9 Tahun 2006 tentang Rencana Tapak (Lembaran Daerah Kabupaten Tangerang Tahun 2006 Nomor 9, Tambahan Lembaran Daerah Kabupaten Tangerang Nomor 0906);
13. Peraturan Daerah Kabupaten Tangerang Nomor 11 Tahun 2006 tentang Izin Pemanfaatan Ruang (Lembaran Daerah Kabupaten Tangerang Tahun 2006 Nomor 11, Tambahan Lembaran Daerah Kabupaten Tangerang Nomor 1106);
14. Peraturan Daerah Kabupaten Tangerang Nomor 12 Tahun 2006 tentang Garis Sempadan (Lembaran Daerah Kabupaten Tangerang Tahun 2006 Nomor 12, Tambahan Lembaran Daerah Kabupaten Tangerang Nomor 1206);
15. Peraturan Walikota Tangerang Selatan Nomor 01 Tahun 2009 tentang Organisasi Perangkat Daerah Kota Tangerang Selatan (Berita Daerah Kota Tangerang Selatan Tahun 2009 Nomor 01) sebagaimana telah diubah dengan Peraturan Walikota Tangerang Selatan Nomor 07 Tahun 2009 tentang Perubahan Atas Peraturan Walikota Tangerang Selatan Nomor 01 Tahun 2009 tentang Organisasi Perangkat Daerah Kota Tangerang Selatan (Berita Daerah Kota Tangerang Selatan Tahun 2009 Nomor 07);
16. Peraturan Walikota Tangerang Selatan Nomor 46 Tahun 2009 tentang Pemberlakuan Peraturan Daerah dan Peraturan Bupati

Tangerang di Kota Tangerang Selatan (Berita Daerah Kota Tangerang Selatan Tahun 2009 Nomor 46);

Persyaratan – persyaratan yang harus dipenuhi untuk memperoleh Ijin Mendirikan Bangunan di BP2T Tangerang Selatan :

1. Formulir Permohonan;
2. Foto Copy KTP;
3. Foto Copy PBB;
4. Foto Copy Surat Tanah;
5. Foto Copy Akte Perusahaan;
6. Foto Copy NPWP;
7. Foto Copy Ijin Lokasi;
8. Foto Copy Amdal/UKL/UPL;
9. Foto Copy Peil Banjir;
10. Perhitungan Kontruksi untuk > 2 Lantai dan Bentangan > 15 Meter;
11. Foto Copy IPR;
12. Foto Copy Site Plan;
13. Foto Copy Ijin Lingkungan;
14. Amdal Lalu lintas;
15. Rekomendasi dari Instansi Terkait;
16. Gambar Bangunan 3 (tiga) Rangkap.
17. Surat kuasa bagi yang dikuasakan.

## 2.2. UML (Unified Modelling Language)

### 2.2.1. Defenisi

UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak tergantung pada bahasa dan teknologi, pemanfaatan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak. Standar UML dikelola oleh OMG (Object Management Group). (Heriyanto,2004)

UML adalah bahasa permodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem. Kelebihan UML diantaranya:

1. Di dalam (*system intensive process*), metode diterapkan sebagai proses untuk menurunkan atau mengevolusikan sistem.
2. Sebagai bahasa UML digunakan untuk komunikasi, yaitu alat untuk menangkap pengetahuan (semantik) mengenai suatu subjek dan mengapresiasikan yang memperdulikan subjek sebagai suatu komunikasi. Subjek merupakan sistem yang dibahas.
3. Sebagai bahasa permodelan, UML focus pada pemahaman subjek melalui formulasi model dari subyeks ( dan konsep yang terhubung). Model memuat pengetahuan pada subyek, dan aplikasi dari pengetahuan ini berkaitan dengan intelejensi.

4. Berkaitan dengan unifikasi, UML memadukan praktik rekayasa terbaik sistem informasi dan industry, meliputi beragam tipe sistem (perangkat lunak dan non lunak), domain (bisnis, perangkat lunak) dan proses siklus hidup.
5. Ketika diterapkan untuk menspesifikasikan sistem, UML dapat digunakan untuk mengkomunikasikan “apa” yang diperlukan dari sistem dan “bagaimana” sistem dapat direalisasikan.
6. Ketika diterapkan untuk memvisualisasikan sistem, UML dapat digunakan untuk menjelaskan sistem secara visual sebelum direalisasikan.
7. Ketika diterapkan untuk membangun sistem, UML dapat digunakan untuk memandu realisasi sistem serupa dengan “blueprint”.
8. Ketika diterapkan untuk mendokumentasikan sistem, UML dapat digunakan untuk menangkap pengetahuan mengenai sistem pada seluruh siklus hidup. (Bambang Heriyanto, 2004)

Diagram-diagram yang terdapat pada permodelan UML sebagai berikut :

1. *Use Case Diagram*

*Use case* diagram merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing-masing diagram use case menunjukkan sekumpulan *use case*, *actor* dan hubungannya.

Diagram *use case* digunakan untuk memvisualisasikan, menspesifikasikan dan mendokumentasikan kebutuhan perilaku sistem. Diagram-diagram *use case* merupakan pusat permodelan perilaku sistem, subsistem dan kelas.

## 2. Class Diagram

*Class diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

*Class diagram* memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

## 3. Statechart Diagram

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada

sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

#### 4. *Activity Diagram*

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

#### 5. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap

waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang menjadi *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

#### 6. *Collaboration Diagram*

*Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.

Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.

#### 7. *Component Diagram*

*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary*

*code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

#### 8. *Deployment Diagram*

*Deployment/physical diagram* menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Sebuah node adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini. (Dharwiyanti, 2003:4)

#### 2.2.2. Notasi UML

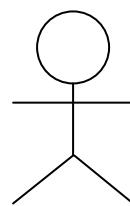
Notasi UML diturunkan dari tiga notasi yang sudah ada sebelumnya yaitu Grady Booch OOD (*Object Oriented Design*), Jim Rumbough OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*). Notasi

UML yang digunakan sekarang merupakan penggabungan dari tiga notasi tersebut, yang terdiri dari:

1. Aktor (*actor*)

Dalam pemodelan sistem dengan UML, aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan. Aktor berada diluar lingkup sistem atau perangkat lunak yang sedang kita kembangkan; bersifat eksternal (Hermawan, 2006).

Secara prinsip dapat kita kenali 3 jenis aktor untuk hampir semua sistem atau perangkat lunak yang kita kembangkan: para pengguna sistem atau perangkat lunak lain yang berinteraksi dengan sistem atau perangkat lunak yang kita kembangkan serta waktu. Jadi aktor ini bisa berupa orang, perangkat keras atau mungkin juga objek lain dalam sistem yang sama. Biasanya yang dilakukan oleh aktor adalah memberikan informasi pada sistem dan atau memerintahkan sistem untuk melakukan sesuatu.



**Gambar 2.2** Notasi Aktor

**Sumber:** (Nugroho, 2005)

Kelas seperti juga objek, adalah sesuatu yang membungkus (*encapsulate*) informasi dan perilaku dalam dirinya (Hermawan, 2006). Dalam pengembangan sistem tradisional, kita mengadakan pendekatan dengan cara memisahkan informasi-informasi pada sisi basis data dan perilaku yang mengaksesnya di sisi aplikasi pemasup atau pengakses. Pendekatan berorientasi objek berbeda, yaitu menggabungkan potongan-potongan informasi dengan perilaku yang akan mengaksesnya dalam apa yang dinamakan kelas.



**Gambar 2.3** Notasi Kelas

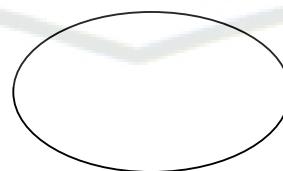
**Sumber:** (Nugroho, 2005)

## 2. Use Case

*Use Case* adalah peringkat tertinggi dari fungsional yang dimiliki sistem (Hermawan, 2006). Dengan kata lain, *use case* menggambarkan bagaimana seseorang akan menggunakan sistem. *Use case* menjelaskan suatu urutan

kegiatan yang dilakukan oleh aktor dan sistem untuk mencapai tujuan tertentu walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan tersebut.

Keunggulan dari cara memandang sistem sebagai kumpulan *use case* adalah kemampuannya untuk memisahkan implementasi sistem dari alasan mengapa sistem harus ada. Ia akan membantu kita untuk berfokus pada apa yang paling penting, yaitu menentukan apa yang dibutuhkan serta apa harapan pengguna terhadap sistem atau perangkat lunak yang sedang dikembangkan.



**Gambar 2.4** Notasi *Use Case*

**Sumber:** (Nugroho, 2005)

### 3. *Interaction*

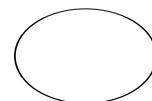
Menurut Hermawan (2004) *interaction* digunakan untuk menunjukkan baik aliran pesan atau informasi antar objek maupun hubungan antar objek. Biasanya *interaction* ini dilengkapi juga dengan teks bernama *operation*

*signature* yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.

**Gambar 2.5 Notasi *Interaction***  
**Sumber :** ( Nugroho, 2005)

#### 4. *Interface*

*Interface* merupakan kumpulan operasi tanpa implementasi dari suatu *class*. Implementasi operasi dalam *interface* dijabarkan dalam operasi dalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang mengimplementasikan operasinya (Hermawan, 2004). *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.



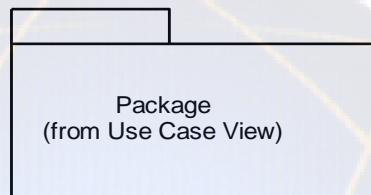
**Gambar 2.6 Notasi *Interface***

**Sumber:** (Nugroho, 2005)

#### 5. *Package*

*Package* adalah *container* atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari suatu sistem yang sedang dibangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah

untuk mempermudah penglihatan (*visibility*) dari suatu model yang sedang dibangun.



**Gambar 2.7 Notasi Package**  
**Sumber:** (Nugroho, 2005)

#### 6. Note

Note dibangun untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. Note ini bisa ditempelkan ke semua elemen notasi yang lain.

```

graph TD
    Note[Note]

```

**Gambar 2.8 Notasi Note**  
**Sumber:** (Nugroho, 2005)

#### 7. Dependency

Merupakan relasi yang menunjukkan bahwa perubahan pada suatu elemen memberikan pengaruh pada elemen yang lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa ada tanda panah.

Terdapat dua *stereotype* dari *dependency*, yaitu *include* dan *extend*. *Include* menunjukan bahwa suatu bagian dari elemen (yang ada di garis tanpa panah) memicu eksekusi bagian dari elemen yang lain (yang ada di garis dengan panah), misalnya untuk notasi  $A \rightarrow B$  operasi yang ada di *class A* memicu dieksekusinya operasi yang berada di *class B*.

*Extend* menunjukan bahwa suatu bagian dari elemen di garis tanpa panah bisa disiapkan ke dalam elemen yang ada di gari dengan panah, misalnya untuk notasi  $A \rightarrow B$  suatu fungsi dari *use case A* bisa disisipkan ke dalam *use case B* atau dengan kata lain A optional untuk B.

Ke dua *stereotype* ini di representasikan dengan menambahkan *text include* atau *extend* di notasi *dependency*.

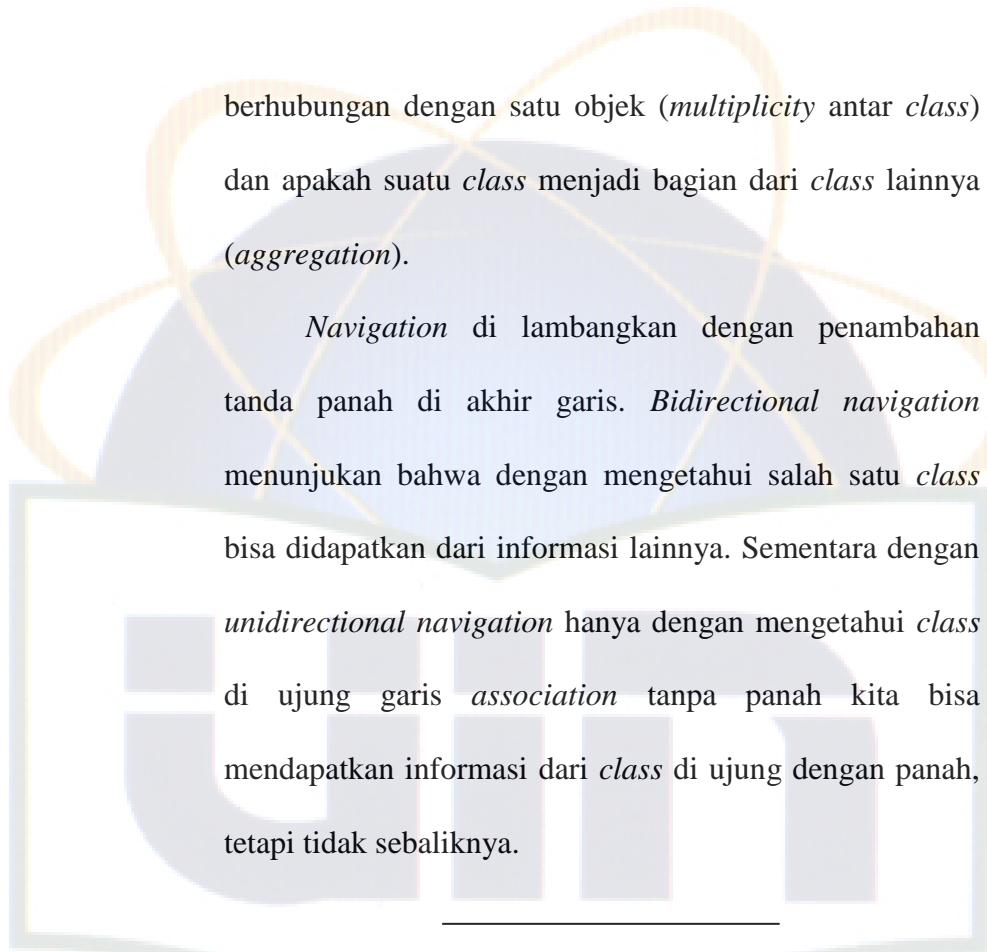


**Gambar 2.9** Notasi Dependency

**Sumber:** (Nugroho, 2005)

## 8. Association

*Association* menggambarkan navigasi antar *class* (*navigation*), berapa banyak objek lain yang bisa



**Gambar 2.10** notasi association

**Sumber:** (Nugroho, 2005)

Pada penelitian kali ini penulis menggunakan visual paradigm for UML 7.0 edition dan Rational Rose untuk merancang aplikasi. Visual paradigm for UML 7.0 edition dan rational rose adalah perangkat lunak yang digunakan untuk perancangan aplikasi dengan tools UML.

### 2.3. Aplikasi Berbasis Web

Aplikasi berbasis web (*web based application*) adalah aplikasi yang dapat dijalankan langsung melalui web browser bisa menggunakan internet

ataupun intranet dan tidak tergantung pada sistem operasi yang digunakan (Rizky, 2010).

Unsur- unsur dalam web adalah sebagai berikut:

### 1. Internet

Internet merupakan kepanjangan dari *Interconnection Networking*. internet merupakan rangkaian jaringan terbesar di dunia dimana semua jaringan yang berada pada semua organisasi dihubungkan dengan suatu jaringan terbesar melalui telefon, satelit dan sistem-sistem komunikasi yang lain sehingga dapat saling berkomunikasi (Mulyanto, 2009: 113).

Untuk dapat bertukar informasi, digunakan protocol standar yaitu Transmision Control Protocol dan Internet Protocol yang lebih dikenal sebagai TCP/IP. Sedangkan intranet merupakan jaringan komputer didalam suatu organisasi yang menggunakan teknologi internet sehingga memungkinkan saling berbagi informasi, komunikasi, kerja sama, dan dukungan bagi proses bisnis.

### 2. Nama domain/ URL

Nama domain atau URL adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah website. Nama domain memudahkan user dalam mengingat alamat IP. Layanan yang bertugas menerjemahkan alamat IP ke sebuah nama domain adalah DNS (Domain Name Service).

### 3. Web browser

Web browser merupakan aplikasi di pihak client yang berfungsi menerjemahkan dan menampilkan informasi dari server secara grafis kepada client.

### 4. Web server

Sebuah komputer (server) dan software yang menyimpan dan mendistribusikan data komputer lainnya melalui jaringan internet.

### 5. Web hosting

Web hosting yaitu sebagai ruangan yang terdapat dalam harddisk tempat menyimpan berbagai data, file-file, gambar, dan lain-lain yang akan ditampilkan di website.

Aplikasi berbasis web memiliki beberapa kelebihan sebagai berikut:

- a. *Platform independent* yaitu aplikasi dapat dijalankan di berbagai sistem operasi seperti Windows, Linux, dan Mac OS.
- b. Untuk menjalankan aplikasi di banyak komputer tidak perlu *instal* aplikasi di setiap komputer, cukup *copy script* programnya ke server atau salah satu komputer. Untuk komputer lain yang ingin menjalankan program cukup membuka alamat *host server* dimana program disimpan melalui *browser*.
- c. Aplikasi dapat dijalankan dari jarak jauh dengan menggunakan koneksi internet.

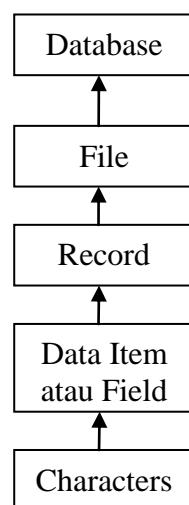
## 2.4. Konsep Database

### 2.4.1. Defenisi Basis Data (*Data Base*)

James F. Courtney Jr. dan David B. Paradice dalam buku “Database System for Management” menjelaskan sistem database adalah sekumpulan database yang dapat dipakai secara bersama-sama, personal-personal yang merancang dan mengelola database, teknik-teknik untuk merancang dan mengelola database, serta komputer untuk mendukungnya (Sutabri, 2005 : 161).

Dari definisi diatas, penulis menyimpulkan bahwa sistem database mempunyai beberapa elemen penting, yaitu database sebagai inti sistem database, perangkat lunak untuk mengelola database, perangkat keras sebagai pendukung operasi pengolahan data, serta manusia mempunyai peran penting dalam sistem tersebut.

Sampai dengan membentuk database, data mempunyai jenjang yang dapat dilihat dalam gambar 2.11.



**Gambar 2.11.** Jenjang dari data (Sumber: Jogianto, 2005)

### a. *Characters*

*Characters* adalah bagian data yang terkecil, dapat berupa karakter numeric, huruf ataupun karakter-karakter khusus yang membentuk suatu *item data* atau *field*.

### b. *Field*

*Field* menggambarkan suatu atribut dari record yang menunjukkan suatu item dari data, seperti nama, jenis kelamin, dan lain-lain.

Kumpulan dari *field* membentuk suatu *record*.

#### 1) Nama field (*field name*)

*Field* harus diberi nama untuk membedakan *field* yang satu dengan *field* yang lain.

#### 2) Representasi dari field (*field representation*)

Representasi dari *field* menunjukkan tipe dari *field* (*field type*) dapat berupa tipe numeric, karakter, tanggal, dan lain-lain. Serta lebar dari *field* menunjukkan ruang maksimum dari *field* yang dapat diisi dengan karakter-karakter data.

#### 3) Nilai dari field (*field value*)

Nilai dari *field* menunjukkan isi dari *field* untuk masing-masing *record*.

### c. *Record*

*Record* adalah kumpulan dari *field* yang membentuk suatu *record*.

Kumpulan dari *record* membentuk file. Misalnya file pegawai, tiap-tiap *record* dapat mewakili data tiap-tiap pegawai.

#### d. *File*

*File* terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya *file* pangkat berisi tentang semua pangkat yang ada.

#### 2.4.2. DBMS (*Database Management System*)

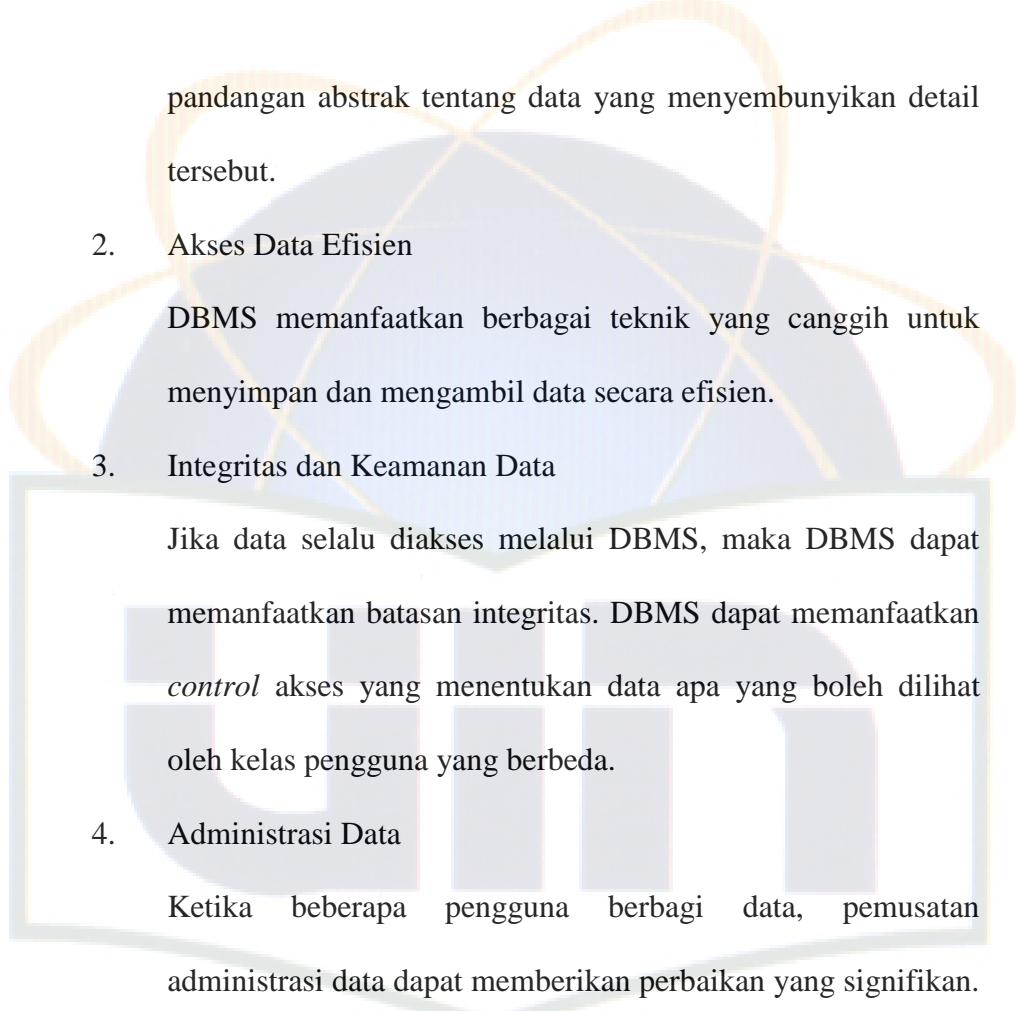
Inti dari suatu basis data adalah *database management system* (DBMS), yang membolehkan pembuatan, modifikasi, dan pembaharuan basis data. *Database Management System* (DBMS) adalah paket perangkat lunak yang komplek digunakan untuk memanipulasi *database*.

Lebih lanjut lagi, DBMS merupakan koleksi terpadu dari database dan program-program komputer (*utilitas*) yang digunakan untuk mengakses dan memelihara database. Program-program tersebut menyediakan berbagai fasilitas operasi untuk memasukkan, melacak, dan memodifikasi data kedalam database, mendefinisikan data baru, serta mengolah data menjadi informasi yang dibutuhkan (Ladjamudin, 2005 : 130).

Beberapa keunggulan DBMS untuk mengelola data (Ramakrishnan & Gehrke, 6: 2003) :

##### 1. Kemandirian data

Program aplikasi idealnya tidak diekspos pada detail representasi dan penyimpanan data. DBMS menyediakan satu



pandangan abstrak tentang data yang menyembunyikan detail tersebut.

## 2. Akses Data Efisien

DBMS memanfaatkan berbagai teknik yang canggih untuk menyimpan dan mengambil data secara efisien.

## 3. Integritas dan Keamanan Data

Jika data selalu diakses melalui DBMS, maka DBMS dapat memanfaatkan batasan integritas. DBMS dapat memanfaatkan *control* akses yang menentukan data apa yang boleh dilihat oleh kelas pengguna yang berbeda.

## 4. Administrasi Data

Ketika beberapa pengguna berbagi data, pemusatan administrasi data dapat memberikan perbaikan yang signifikan.

Para profesional yang berpengalaman yang memahami sifat data yang akan dikelola, dan memahami bagaimana kelompok pengguna yang berbeda menggunakan data tersebut, dapat memegang tanggung jawab untuk mengatur representasi data untuk meminimalkan redudansi dan untuk mengatur penyimpanan data guna melakukan pengambilan data yang efisien.

## 5. Akses Konkuren dan Crash recovery

DBMS menjadwalkan akses konkuren pada data dalam cara tertentu sehingga pengguna dapat memandang data sebagai

data yang diakses oleh hanya satu pengguna pada satu waktu.

Lebih lanjut, DBMS memproteksi pengguna dari efek kegagalan sistem (konkurensi).

#### 6. Waktu Pengembangan Aplikasi Terkurangi

DBMS mendukung fungsi penting bagi banyak aplikasi untuk mengakses data dalam DBMS dan memfasilitasi pengembangan aplikasi yang cepat dalam hal ini berkaitan dengan bantuan interface untuk mengatur data.

#### 2.4.3. SQL (*Structured Query Language*)

SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Umumnya, setiap software-software RDBMS menyediakan 4 perintah DML dari SQL. DML (*Data Manipulation Language*) adalah perintah yang digunakan untuk mengoperasikan atau memanipulasi isi *database*. Empat perintah DML tersebut diantaranya:

- *select* : digunakan untuk mengambil data dari database.
- *delete*: digunakan untuk menghapus data pada database.
- *insert* : menambahkan data ke database.
- *update*: memodifikasi data pada database.

## 2.5. Mengenal Rational Rose

Pengguna UML secara manual untuk melakukan analisis masalah dan perancangan sistem atau perangkat lunak sungguh sudah memadai. Namun, para pengguna merasa pemodelan dengan UML secara manual masih terlalu lambat dan tingkat akurasinya rendah. Selain itu, proses rekayasa balik (*reverse engineering*), yaitu mengubah kode-kode program kemudian melakukan pemodelan kembali dengan cara *iteratif*, akan sangat membosankan jika dilakukan dengan cara manual.

Sebab itu para pengembang perangkat lunak berorientasi objek membutuhkan perangkat lunak yang dapat mengotomatisasi semua hal itu. Di antaranya adalah *Rational Rose*, yang dapat mengotomatisasi pemodelan dengan UML, mempermudah rekayasa balik, serta *Rational Rose* dapat menghasilkan kode-kode program dari beberapa jenis pemrograman, diantaranya C++, Java serta Visual Basic.

### 2.5.1. Dasar-dasar pemodelan Rational Rose

Pemodelan UML dengan menggunakan Rational Rose didasarkan pada beberapa konsep, di antaranya sebagai berikut:

a. *Visual Modelling*

Menurut Quatrani *visual modelling* atau pemodelan visual adalah suatu cara berpikir tentang persoalan menggunakan model-model yang diorganisasikan seputar dunia nyata (Sholiq, 2006). Model tersebut berguna untuk memahami persoalan,

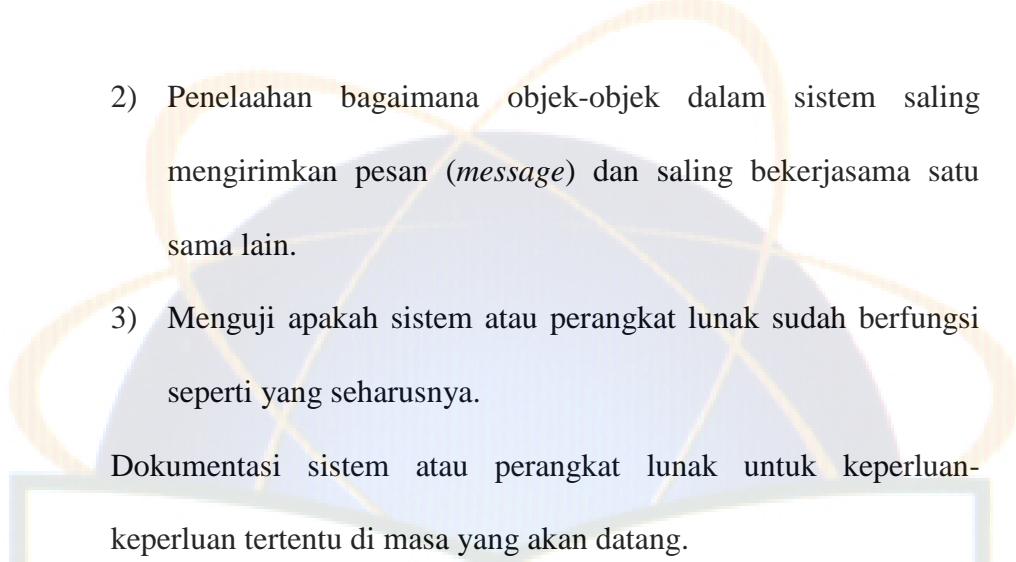
mengkomunikasikan dengan orang-orang yang terlibat di dalam proyek, memodelkan perusahaan, menyiapkan dokumentasi, merancang program dan merancang basis data.

*Visual Modelling* dapat membantu kita untuk menampilkan elemen-elemen yang penting secara detil dari suatu masalah yang kompleks dan menyaring untuk kemudian membuang elemen-elemen yang tidak penting. *Rational Rose* menggunakan UML sebagai bahasa pemodelannya. Semua semantik dan notasi dalam UML dibuat untuk digunakan dalam *visual modelling*.

#### b. Model dalam Rekayasa *Software*

Sebagaimana telah diketahui dalam mendesain sebuah model dalam proses rekayasa *software* sangat penting sebagaimana pentingnya memiliki cetak biru untuk membangun suatu bangunan yang besar. Untuk melakukan pemodelan sistem atau perangkat lunak maka akan digunakan notasi-notasi UML (*Unified Modeling Language*) yang akan digambarkan secara elektronik (dengan bantuan komputer) lewat sarana perangkat lunak *Rational Rose*. Dengan pemodelan menggunakan UML ini, para pengembang sistem dapat melakukan (Nugroho, 2005):

- 1) Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.

- 
- 2) Penelaahan bagaimana objek-objek dalam sistem saling mengirimkan pesan (*message*) dan saling bekerjasama satu sama lain.
  - 3) Menguji apakah sistem atau perangkat lunak sudah berfungsi seperti yang seharusnya.

Dokumentasi sistem atau perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.

### 2.5.2. Konsep Dasar Rational Rose

*Rational Rose* adalah kertas (*tools*) pemodelan visual untuk pengembangan sistem berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem (Nugroho, 2005).

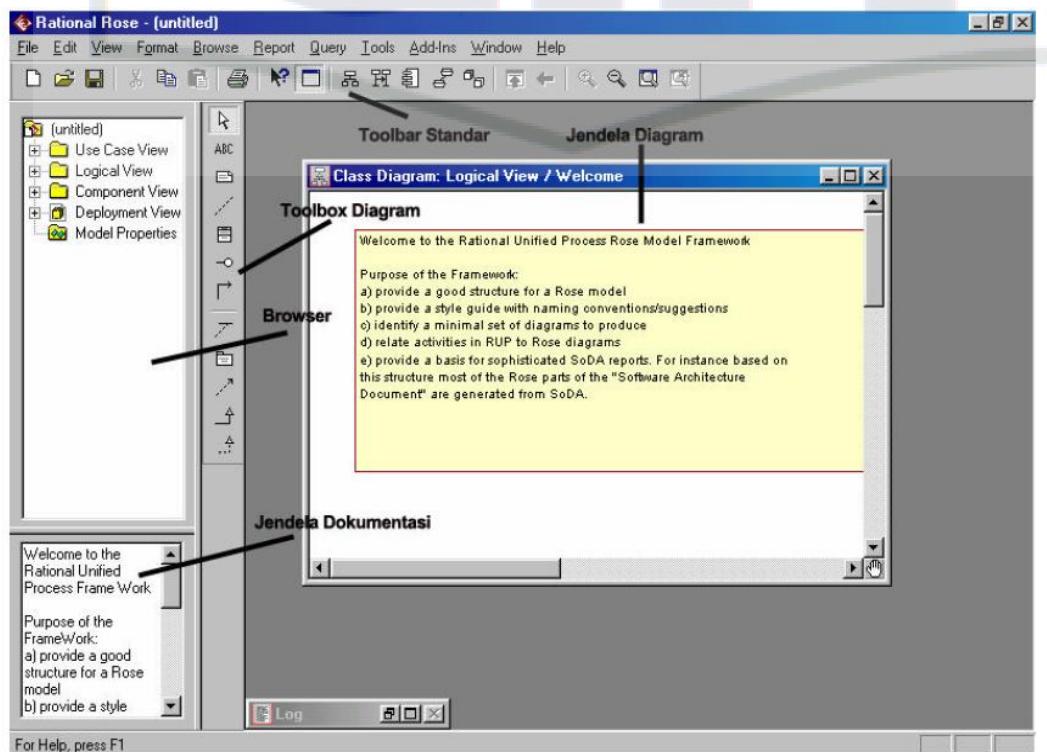
*Rational Rose* yang dikembangkan oleh tiga pendekar teknologi objek dan pendeklarasi UML Booch, Rumbaugh, Jacobson melalui perusahaan Rational (yang sejak Februari 2003 menjadi anak perusahaan IBM) yang sudah menjadi alat bantu yang digunakan oleh industri pengembang perangkat lunak aplikasi berorientasi objek di seluruh dunia untuk melakukan analisis dan desain visual (Hermawan, 2004).

Selain itu, Rational Rose juga memungkinkan pengembang untuk mendokumentasikan kegiatan-kegiatannya. Adapun beberapa fungsi dari dokumentasi adalah sebagai berikut (Nugroho, 2005):

1. Para pengembang dapat menggunakan diagram *usecase* untuk mendapatkan pemahaman yang komprehensif tentang sistem dan lingkungan luar yang meliputi sistem
2. Para calon pengguna dan manajer proyek dapat menggunakan diagram *usecase* untuk mendapatkan pandangan peringkat paling atas tentang sistem dan untuk menentukan persetujuan tentang lingkup proyek.
3. Manajer proyek dapat menggunakan diagram *usecase* untuk membagi sistem secara keseluruhan menjadi bagian-bagian yang dapat dikelola dengan seksama.
4. Calon pengguna dan analis sistem dapat melihat diagram *usecase* untuk dapat memahami fungsionalitas sistem yang diharapkan
5. Analis sistem dan pengembang dapat melihat *sequence diagram* untuk memahami bagaimana logika sistem berjalan, objek-objek yang terlibat dalam sistem, serta pesan-pesan (*message*) yang dikirimkan suatu objek ke objek-objek lainnya.
6. Para pengembang dapat menggunakan diagram *class* dan *statechart diagram* untuk mendapatkan rincian sistem bagaimana objek-objek berhubungan dan bekerja.

Elemen-elemen dasar GUI dalam *Rational Rose* terdiri dari empat jendela (*window*) utama, yang deskripsinya adalah sebagai berikut (Nugroho, 2005):

- 1) *Browser*. Jendela ini berfungsi untuk secara cepat bergerak dalam model.
- 2) Jendela *Diagram*. Jendela ini berfungsi untuk membuat, menampilkan (*display*), serta menyunting (*edit*) satu atau lebih diagram UML.
- 3) Jendela *Dokumentasi*. Jendela ini berguna untuk melihat atau memperbarui (*update*) dokumentasi unsur-unsur model.
- 4) Jendela *Log*. Jendela ini berguna untuk melihat kesalahan (*error*) dan melaporkan (*report*) hasil-hasil dari berbagai perintah yang diberikan pada Rational Rose.



**Gambar 2.12.** Tampilan dasar dalam *Rational Rose*  
**(Sumber:** Nugroho, 2005)

Ada tiga edisi *Rational Rose*, yaitu:

- a) *Rose Modeler*, tidak mendukung bahasa pemrograman apapun.
- b) *Rose Profesional*, mendukung satu bahasa pemrograman.
- c) *Rose Enterprise*, mendukung banyak bahasa pemrograman, yaitu CORBA, VC ++, *Visual Basic*, *Java* dan sebagainya.

Penulisan skripsi ini akan menggunakan edisi ***Rational Rose Enterprise***.

## 2.6. Bahasa Pemograman

### 2.6.1. HTML

HTML (Hyper Text Markup Language) adalah sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang dimaksudkan untuk menampilkan halaman pada web browser. Tag-tag tadi memberitahu browser bagaimana menampilkan halaman web dengan lengkap kepada pengguna (Astamal, 2006). HTML hanya berisi berisi data saja (content), dan untuk mengatur tampilan (layout) halaman digunakan CSS.

Cara kerja HTML sangat sederhana, yaitu berawal dari client yang memanggil berdasarkan URL (Uniform Resource Locator) melalui browser, kemudian browser mendapat alamat dari web server, yang nantinya akan memberikan segala informasi yang dibutuhkan web browser. Web browser yang sudah mendapat informasi segera melakukan proses penterjemahan kode HTML dan menampilkannya ke layar pemakai.

### Contoh Skrip HTML

```
<html>
<head>
<title>Web Programming </title>
</head>
<body bgcolor="#FFFFFF"> <h1>ini adalah heading 1</h1>
ini adalah bagian tubuh dokumen. Semua yang ditulis
disini akan ditampilkan ke layar browser
</body>
</html>
```

#### 2.6.2. PHP

PHP merupakan hasil kerja seorang bernama Rasmus Lerdorf pada 1995. Namun kemudian PHP berkembang dan tidak hanya merupakan proyek pribadi Rasmus. PHP ditulis ulang dan dengan banyak menambahkan fungsi-fungsi baru oleh Zeev Suraski dan Andi Gutmants (disingkat Zend) dan lahirlah PHP 3 pada 1998 (Astamal, 2006).

PHP adalah bahasa server-side scripting yang didesain khusus untuk web. Pada halaman HTML dapat ditempelkan (*embed*) kode PHP. Kode PHP dieksekusi di sisi server bukan di komputer klien. Dan hasil yang ditampilkan adalah kode HTML (Astamal, 2006).

Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang di berikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada dokumen HTML biasa. Pembuatan *web* ini merupakan kombinasi antara PHP sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman *web*. PHP

dikenal sebagai bahasa scripting yang menyatu dengan tag HTML, dieksekusi di server dan digunakan untuk membuat halaman web yang dinamis. PHP adalah merupakan *software* yang *Open Source* dan mampu lintas *platform*.

PHP mampu berjalan di Windows NT dan beberapa versi UNIX, dan PHP dapat dibangun sebagai modul pada *web server* Apache. PHP dapat mengirim HTTP *header*, dapat mengeset *cookies*, mengatur *authentication* dan *redirect users*. PHP menawarkan konektifitas yang baik dengan beberapa basis data antara lain Oracle, Firebird, Sybase, MySQL, PostgreSQL, dan tak terkecuali semua database berinterface ODBC. Dan juga integrasi dengan beberapa *library* eksternal yang dapat membuat programmer melakukan segalanya dari dokumen PDF hingga mem-parse XML. PHP juga mendukung komunikasi dengan layanan lain melalui protokol SNMP, POP3 atau bahkan HTTP.

Konsep kerja PHP hanya perlu penterjemahan khusus untuk kode-kode PHP yang nantinya akan diterjemahkan oleh mesin PHP ke kode HTML terlebih dahulu sebelum diterjemahkan browser untuk ditampilkan di layar klien.

Aturan penulisan script PHP adalah:

1. Semua script PHP harus diapit oleh tanda:

<?php dan ?> , atau

<script language='php'> dan </script> , atau

<? dan ?>, atau

<% dan %>

2. Tetapi tanda yang resmi dan paling banyak digunakan adalah yang pertama, yaitu <?php dan ?>
3. Pada setiap akhir perintah, diakhiri dengan tanda titik koma ( ; ) (Yuliano: 2003).

Berikut ini contoh sederhana pemakaian bahasa PHP dalam halaman *web* :

```
<html>
<head>
<title>Example</title>
</head>
<body>
<? Echo "Hello World!"; ?>
</body>
</html>
```

### 2.6.3. CSS

CSS atau Cascading Style Sheet adalah suatu cara untuk membuat format atau layout halaman web menjadi lebih menarik dan mudah dikelola. Beberapa hal yang dapat dilakukan dengan CSS adalah :

- a. Mendefinisikan tampilan halaman web yang dibuat dalam satu tempat khusus, lebih baik daripada menulis berulang-ulang dalam beberapa halaman web.
- b. Kemudahan mengubah tampilan halaman web setelah halaman itu selesai dibuat.

- c. Mendefinisikan ukuran huruf dan atribut-atribut serupa yang memiliki akurasi setingkat word-processor.
- d. Mendefinisikan style sesuai kebutuhan untuk link
- e. Mendefinisikan layer yang dapat diletakkan diatas elemen lain (pop-up).

#### **2.6.4. JavaScript**

*JavaScript* Adalah bahasa skrip (bahasa yang kodennya ditulis menggunakan teks biasa) yang ditempelkan pada dokumen HTML dan diproses pada sisi klien (Kadir, 2009: 8) . Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi semakin luas. Menggunakan JavaScript memungkinkan mengimplementasikan tugas yang bersifat interaktif tanpa berhubungan dengan server.

#### **2.6.5. Firebird**

*InterBase* adalah RDBMS (*Relational Data Management System*) yang dibuat oleh *Borland Software Corp.*, perusahaan pembuat Delphi. *InterBase* memiliki semua fitur-fitur utama yang harus dimiliki oleh sebuah RDBMS

*InterBase* pertama kali ditulis oleh Jim Starkey dengan perusahaannya *Groton Database System*. *Groton Database System* kemudian dibeli oleh Ashton-Tate pembuat dBasis. Ashton-Tate kemudian mengembangkan *InterBase* dan selalu menyertakan

*InterBase* dengan jumlah lisensi terbatas pada produk Delphi yang mereka jual selain dijual terpisah.

Di tahun 2000 Borland (ketika itu bernama Inprise Corp.) me-release source-code *InterBase 6.0* ke public di bawah *InterBase Public License 1.0*. Karena suatu hal, Borland meneruskan menjual produk *InterBase* sebagai produk *proprietary*. Di sisi lain, komunitas *Open Source* mengembangkan sebuah RDBMS baru berdasarkan source-code *InterBase 6.0*. RDBMS baru ini diberi nama *Firebird*. Beberapa orang pembuat versi pertama *InterBase* kini termasuk dalam tim pengembangan *Firebird* seperti Jim Starkey dan Ann W. Harrison.

*Firebird* dan *InterBase* tersedia dalam berbagai system operasi termasuk *Windows*, *Linux* dan beberapa variant *Unix*.

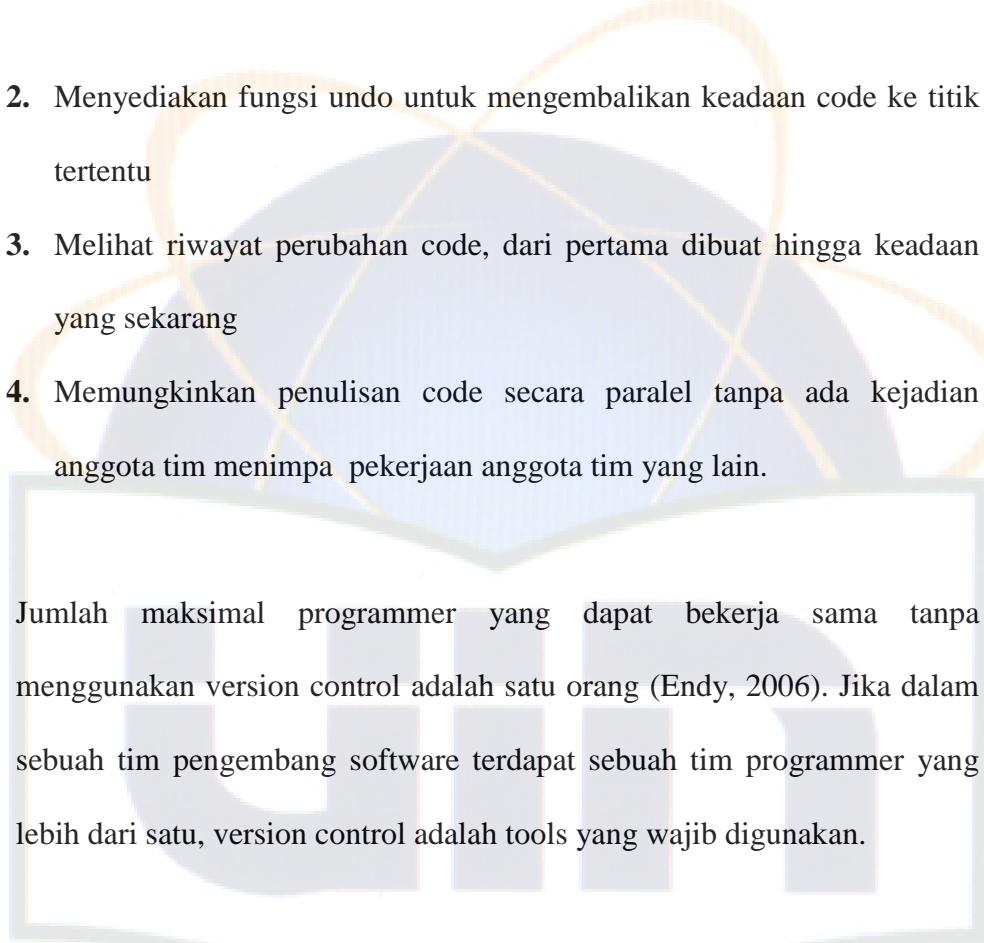
## 2.7. Mengenal Version Control

Kemampuan menggunakan version control (Endy Muhardin, Endy.muhardin@gmail.com) merupakan hal yang masih langka ditemukan di kalangan programmer Indonesia. Tidak banyak buku, tutorial ataupun tempat pelatihan yang menyediakan bahan pembelajaran dalam bahasa indonesia. Bahkan di banyak perusahaan software consultant, version control ini masih jarang digunakan.

Padahal kemampuan menggunakan version control adalah kemampuan wajib yang harus dimiliki oleh tim programmer. Di seluruh dunia, ribuan programmer terlibat dalam pengembangan proyek opensouce, kolaborasi dalam skala raksasa seperti ini mustahil dilaksanakan tanpa adanya version control. Akan terjadi bencana dan kekacauan jika kode sumber hanya diletakkan di website, kemudian dibagi ke semua programmer. Jika dua orang programmer bekerja dalam file yang sama, mereka akan saling menimpa pekerjaan temannya, dan jika terjadi kesalahan maka tidak akan mungkin mengembalikan keadaan kode ke keadaan sebelumnya tanpa usaha yang cukup besar.

Version control dapat membantu sebuah tim pengembang perangkat lunak dengan menyediakan akses kepada setiap anggota tim tanpa harus saling menimpa pekerjaan anggota tim yang lain, seperti yang terjadi jika sebuah tim pengembang menggunakan sharing folder. Version control mampu :

1. Mencatat perubahan code dan membuat perubahan

- 
2. Menyediakan fungsi undo untuk mengembalikan keadaan code ke titik tertentu
  3. Melihat riwayat perubahan code, dari pertama dibuat hingga keadaan yang sekarang
  4. Memungkinkan penulisan code secara paralel tanpa ada kejadian anggota tim menimpa pekerjaan anggota tim yang lain.

Jumlah maksimal programmer yang dapat bekerja sama tanpa menggunakan version control adalah satu orang (Endy, 2006). Jika dalam sebuah tim pengembang software terdapat sebuah tim programmer yang lebih dari satu, version control adalah tools yang wajib digunakan.

### 2.7.1. Subversion

Subversion adalah aplikasi version control. Aplikasi ini mirip dengan file server, artinya dia bisa digunakan untuk menyimpan file, dan bisa diakses jika ingin mengambil file tersebut. Bedanya dengan file server biasa (FTP server, Samba, atau Windows Sharing), aplikasi version control menyimpan riwayat perubahan semua yang kita simpan di dalamnya. Kalau kita menyimpan satu file, kemudian isinya kita modifikasi (tambah baris, hapus, ganti nama, dan sebagainya), maka setiap perubahan tersebut dicatat oleh version control.

Tidak hanya dicatat, kita juga bisa mengembalikan kondisi file sesuai keinginan. Kita bisa melihat versi awal dari file tersebut, atau nama file sebelum diganti, dan semua titik penting lain di masa lalu.

## 2.8. Metode Pengembangan Sistem Extreme Programming

Pada penelitian ini dalam mengembangkan aplikasi penulis menggunakan *Extreme Programming*. *Extreme Programming (XP)* adalah metode pengembangan perangkat lunak yang ringan dan termasuk salah satu *agile methods* yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. *XP* merupakan *agile methods* yang paling banyak digunakan dan menjadi sebuah pendekatan yang sangat terkenal. *Extreme Programming (XP)* adalah sebuah pendekatan pengembangan perangkat lunak yang mencoba meningkatkan efisiensi dan fleksibilitas dari sebuah proyek pengembangan perangkat lunak dengan mengkombinasikan berbagai ide sederhana (Pressman, 2010).

*Extreme Programming* merupakan salah satu metodologi dalam rekayasa perangkat lunak dan juga merupakan satu dari beberapa *agile software development methodologies* yang berfokus pada *coding* sebagai aktivitas utama di semua tahap pada siklus pengembangan perangkat lunak (*software development lifecycle*). Metodologi ini mengedepankan proses pengembangan yang lebih responsive terhadap kebutuhan customer ("agile") dibandingkan dengan metode-metode tradisional sambil membangun suatu software dengan kualitas yang lebih baik. *Extreme*

Programming muncul menawarkan sebuah disiplin baru dalam pengembangan software secara *agile*. Nilai dasar yang terkandung di dalam Extreme Programming adalah: Komunikasi (***Communication***), Kesederhanaan (***Simplicity***), Umpam balik (***Feedback***) Berikut adalah nilai-nilai mendasar yang menjadi roh dari XP pada setiap tahapan proses pengembangan perangkat lunak:

### 1. *Communication*

XP mengfokuskan pada hubungan komunikasi yang baik antar anggota tim. Para anggota tim harus membangun saling pengertian, mereka juga wajib saling berbagi pengetahuan dan keterampilan dalam mengembangkan perangkat lunak. Ego dari para programer yang biasanya cukup tinggi harus ditekan dan mereka harus membuka diri untuk bekerjasama dengan programer lain dalam menuliskan kode program.

### 2. *Simplicity*

Lakukan semua dengan sederhana. Hal tersebut adalah salah satu nilai dasar dari XP. Gunakan *method* yang pendek dan simpel, jangan terlalu rumit dalam membuat desain, hilangkan fitur yang tidak ada gunanya, dan berbagai proses penyederhanaan lain akan selalu menjadi nilai utama dari setiap aspek XP.

### 3. *Feedback*

Berikan selalu *feedback* kepada sesama anggota tim maupun pihak-pihak lain yang terlibat dalam pengembangan perangkat lunak.

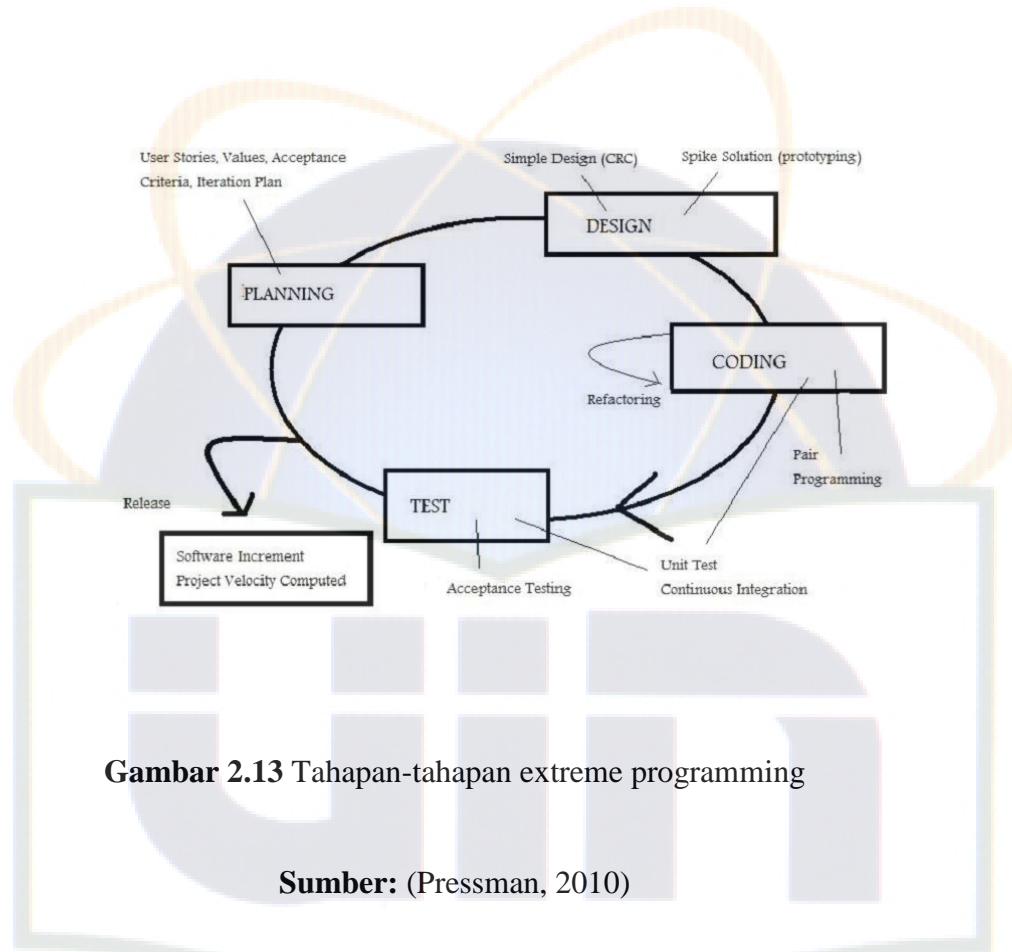
Utarakan selalu pikiran anda dan diskusikan kesalahan-kesalahan yang muncul selama proses pengembangan. Dengarkan selalu pendapat rekan yang lain, dengan adanya *feedback* inilah seringkali kita menyadari bagian mana yang salah atau bisa ditingkatkan lagi dari perangkat lunak yang dikembangkan.

Kelebihan *Extreme Programming* diantaranya adalah:

1. Sistem yang dikembangkan adalah sistem yang sesuai dengan sistem yang diinginkan user, karena pada extreme programming ada keterlibatan user selama pembangunan sistem.
2. Testing yang dilakukan maksimal, karena pada extreme programming mengutamakan coding dan testing pada pengembangannya, testing dilakukan oleh programmer, dan user.
3. Adanya komunikasi yang baik antara user dan pengembang aplikasi.
4. Terciptanya hubungan komunikasi yang baik antar anggota tim, karena adanya prinsip pair programming.

Selain kelebihan terdapat beberapa kelemahan pada metode pengembangan *extreme programming*:

1. Developer harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
2. Tidak bisa membuat kode yang detail di awal (prinsip simplicity dan juga anjuran untuk melakukan apa yang diperlukan).



Tahapan-tahapan pada metode pengembangan *extreme programming*:

1. **Planning.** Kebutuhan awal user atau biasa disebut user stories, ditulis pada fase ini. User stories menjelaskan rincian perkiraan awal untuk mengidentifikasi proses pengembangan dan faktor resiko yang mungkin muncul. User stories umumnya ditulis pada index card. Tim XP dan *Customer* bekerja bersama untuk menentukan *group stories* kedalam rilis berikutnya (*software increment*) untuk dikembangkan oleh tim XP
2. **Design.** Desain XP ketat mengikuti prinsip *Keep it Simple (KIS)*. desain sederhana selalu lebih disukai daripada representasi yang lebih kompleks. Selain itu, desain menyediakan petunjuk

pelaksanaan dari *story* seperti apa adanya tertulis tidak kurang dan tidak lebih. XP mendorong penggunaan *CRC card* sebagai sebuah Mekanisme yang efektif untuk digunakan pada perangkat lunak dalam konteks berorientasi objek. *CRC card* mengidentifikasi dan mengatur kelas-kelas obyek oriented yang relevan dengan pengembangan perangkat lunak. hanya *CRC card* merupakan produk yang hanya digunakan dalam *design work* yang merupakan bagian dari proses xp. Jika kesulitan mendesain ditemui sebagai bagian dari desain *story*, xp merekomendasikan pembuatan sebuah prototipe operasional yang merupakan bagian dari desain. Yang disebut *Spike Solution*, prototipe perancangan diimplementasikan dan dievaluasi. Tujuannya adalah untuk menurunkan risiko pada saat mulai pelaksanaan yang sebenarnya dan untuk memvalidasi perkiraan untuk *story* yang mengandung masalah pada desain. Karena desain xp menggunakan notasi hampir tidak ada, jika produk pekerjaan apapun selain *CRC card* dan *spike solution*, desain dipandang sebagai artefak sementara yang bisa dan harus terus diubah sebagai hasil konstruksi.

3. **Coding.** XP merekomendasikan bahwa setelah *stories* dikembangkan dan desain awal dilakukan, tim mulai mengembangkan serangkaian tes unit yang akan digunakan pada setiap *stories* yang akan disertakan (pengembangan perangkat lunak) . setelah unit test telah dibuat, pengembang berfokus pada

apa yang harus diimplementasikan untuk diuji.setelah kode selesai, bisa langsung diuji sehingga memberikan umpan balik seketika kepada para pengembang.

Konsep kunci selama kegiatan coding adalah pair programming. XP merekomendasikan bahwa dua orang bekerja bersama-sama pada satu komputer workstation untuk membuat kode dari *stories*. Ini menyediakan mekanisme untuk memecahkan masalah pengembangan secara real-time dan jaminan kualitas real-time. Juga membuat para pengembang berfokus pada masalah yang dihadapi. Pada prakteknya, setiap orang mengambil peran yang sedikit berbeda. misalnya, satu orang mungkin mengerjakan rincian tentang bagian pengkodean tertentu dari desain, sementara yang lain membuat standar coding dan coding yang dihasilkan akan "sesuai" ke dalam desain *story*.

4. **Refactoring** adalah proses mengubah sistem perangkat lunak sedemikian rupa sehingga tidak mengubah perilaku eksternal kode sebelum memperbaiki struktur internal.itu adalah cara disiplin untuk membersihkan kode (dan memodifikasi / menyederhanakan desain internal) yang meminimalkan kemungkinan adanya bug. Pada dasarnya, ketika Anda merefactore Anda meningkatkan desain kode setelah ditulis. Maksud refactoring adalah untuk mengontrol modifikasi dengan menyarankan perubahan desain

kecil yang "umumnya dapat meningkatkan desain". perlu dicatat, bagaimanapun upaya yang diperlukan untuk refactoring dapat tumbuh secara dramatis sebagai ukuran dari pengembangan aplikasi.

5. **Testing.** Kita mencatat bahwa penciptaan tes unit sebelum coding dimulai merupakan elemen kunci dari pendekatan xp. Unit test yang dibuat harus dilaksanakan dengan menggunakan kerangka kerja yang memungkinkan mereka untuk otomatis (sehingga, mereka dapat dieksekusi dengan mudah dan berulang-ulang). XP *acceptance test*, juga disebut *customer tests*, ditetapkan oleh *customer* dan fokus pada fitur-fitur sistem secara keseluruhan dan fungsionalitas yang terlihat dan ditinjau kembali oleh *customers*. *acceptance test* berasal dari *user stories* yang telah diimplementasikan sebagai bagian dari rilis software.

### 2.12.1. Index Card

Kartu ini adalah media yang baik untuk menangani cerita pengguna untuk sejumlah alasan (clin ton Keith, 2010):

1. Ukuran kartu membatasi jumlah detail dalam cerita. Kami tidak mau cerita menjadi dokumen besar yang mencakup setiap diperlukan desain detail. Sebuah kartu kecil mencegah hal ini terjadi.

2. Kartu dapat fisik dimanipulasi (disortir, diedit, diganti, dan lulus) oleh banyak tangan dalam pengaturan kolaboratif (scrum sehari-hari dan perencanaan pertemuan).

### 2.12.1. Class Responsibility Colaborator

Model Class Responsibility Collaborator (Wir, 1990)

menyediakan cara sederhana untuk mengidentifikasi dan mengorganisir kelas-kelas yang relevan dengan persyaratan sistem atau produk. Ambler (Ambler, 1995) Menggambarkan pemodelan CRC bahwa CRC merupakan sebuah koleksi dari index card yang menggambarkan kelas-kelas. CRC dibagi menjadi tiga bagian. Dibagian atas kartu anda dapat menuliskan nama kelas, Dalam badan kartu mencatat *responsibility* dari class pada bagian kiri dan *Collaborator* pada bagian kanan.

*Responsibility* merupakan attribute dan operasi yang relevan untuk class. *Collaborator* adalah kelas-kelas yang diperlukan untuk melengkapi kelas dengan informasi yang dibutuhkan untuk *Responsibility* pada umumnya, *Collaborator* menyiratkan baik request informasi atau request untuk beberapa tindakan.

## 2.9. Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk

memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan antara hasil yang diharapkan dengan hasil yang terjadi (Hariyanto, 2004).

Sasaran pengujian adalah penemuan semaksimum mungkin kesalahan dengan usaha yang dapat dikelola pada rentang waktu realistik. Pengujian perangkat lunak merupakan tahap kritis dalam penjaminan kualitas perangkat lunak dan merupakan *review* menyeluruh terhadap spesifikasi perancangan dan pengkodean (Hariyanto, 2004).

Glen Myers menyatakan tiga sasaran pengujian, yaitu (Hariyanto, 2004):

- a. Pengujian adalah proses mengeksekusi program dengan hasrat menemukan kesalahan.
- b. Kasus uji yang bagus adalah mempunyai peluang tinggi menemukan kesalahan yang sebelumnya belum ditemukan.
- c. Pengujian yang berhasil adalah pengujian yang menyingkap kesalahan yang sebelumnya belum ditemukan.

Manfaat pengujian (Hariyanto, 2004):

- a. Pengujian akan menyingkap kesalahan di perangkat lunak.
- b. Pengujian mendemonstrasikan fungsi-fungsi perangkat lunak bekerja sesuai spesifikasi, kebutuhan serta terpenuhi perilaku dan sejenisnya.

Terdapat dua teknik pengujian berdasarkan ketersediaan logik sistem, yaitu *black box testing* dan *white box testing* (Hariyanto, 2004).

Pengujian *white-box* adalah sebuah pengujian yang dilakukan lebih dekat lagi untuk menguji prosedur-prosedur yang ada. Lintasan logik yang dilalui oleh setiap bagian prosedur yang diuji dengan memberikan kondisi/loop spesifik. Proses yang terjadi pada pengujian *white-box* yaitu:

- a. Menjamin pengujian terhadap semua lintasan yang tidak bergantungan minimal satu kali.
- b. Mencoba semua keputusan logik dari sisi ‘*true*’ dan ‘*false*’.
- c. Eksekusi semua loop dalam batasan kondisi dan batasan operasionalnya
- d. Pengujian validasi struktur data internal.

Konsep *black box testing* digunakan untuk merepresentasikan sistem yang cara kerja di dalamnya tidak tersedia untuk diinspeksi. Di dalam kotak hitam, item-item yang diuji dianggap “gelap” karena logiknya tidak diketahui, yang diketahui hanya apa yang masuk dan apa yang keluar. Pada *black box testing*, kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Pada *black box testing*, dicobakan beragam masukan dan memeriksa keluaran yang dihasilkan. Teknik *black box testing* juga dapat digunakan untuk pengujian berbasis skenario di mana isi dalam sistem mungkin tidak tersedia untuk diinspeksi tapi masukan dan keluaran yang didefinisikan dengan *usecase* dan informasi analisis yang lain (Hariyanto, 2004). *Black box testing* berusaha menemukan kesalahan dalam kategori, sebagai berikut (Pressman, 2002):

- a. Fungsi-fungsi yang tidak benar atau hilang.
- b. Kesalahan antarmuka.

- c. Kesalahan dalam struktur data atau akses *database*.
- d. Kesalahan kinerja.

Pengujian *black box testing* merupakan pengujian yang dilakukan oleh pengembang sistem. Karena pengguna akhir sistem memiliki pemahaman tentang sistem informasi dengan tingkatan yang berbeda, maka seberapa jauh pengguna akhir dapat memahami dan menerima sistem harus diuji. Pengujian inilah yang dinamakan dengan *user acceptance test*. Pengujian ini dilakukan untuk menjamin bahwa sistem telah melayani kebutuhan organisasi.

## 2.10. Flowchart

Menurut Husni Iskandar Pohan (1997) Flowchart berfungsi memodelkan masukan, keluaran, referensi, master, proses ataupun transaksi dalam simbol-simbol tertentu. Pada dasarnya tidak berorientasi pada fungsi, waktu ataupun aliran data, tetapi lebih ke arah proses.

Pembuatan flowchart harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi, dalam memodelkannya digunakan simbol-simbol tertentu.

Simbol	Arti
	Menandakan dokumen dapat berupa surat, formulir, buku, bundel, berkas atau cetakan
	Proses manual
	Proses yang dilakukan oleh komputer
	Menandakan dokumen yang diarsipkan (arsip manual)
	Data penyimpanan ( <i>Data Storage</i> )
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
	Terminasi yang menandakan awal dan akhir dari suatu aliran
	Pengambilan keputusan ( <i>decision</i> )
	Layar peraga (monitor)
	Pemasukan data secara manual

**Gambar 2.11.** Simbol Flowchart

## 2.11. Studi Sejenis

Pada tahapan pengumpulan data dengan cara studi literatur, penulis mengumpulkan data melalui literatur penelitian yang terkait dengan penulisan. Dalam perbandingan dan sebagai acuan dalam pembuatan aplikasi Pelayanan Izin Mendirikan Bangunan maka perlu dilakukan pengamatan terhadap penelitian sejenis yang telah dilakukan sebelumnya. Dari hasil pengamatan penulis didapatkan beberapa literatur penelitian sejenis diantaranya dengan judul Perancangan aplikasi pelayanan izin mendirikan bangunan pada kota Tangerang (Indra, 2010) mempunyai kelebihan yang terdapat pada sistem ini memberikan kemudahan dalam proses entry dan *update* data-data permohonan izin mendirikan bangunan.

Mempermudah pembuatan report-report yang dibutuhkan secara otomatis. Kekurangan diantaranya tidak dapat membuat user dan pemberian hak akses, pengembangan basis data masih menggunakan Microsoft Access 2003 dan pengembangan aplikasi menggunakan Microsoft Visual Basic 6.0. Secara garis besar terdapat beberapa kekurangan yang dimiliki VB diantaranya bersifat komersial dan file VB sering menjadi target serangan virus.

Pada judul Analisis Perancangan Sistem Informasi Pelayanan Terpadu Berbasis Web Services di Pemerintah Kota Pekalongan (Adi, 2008) memiliki kelebihan pada sistem ini sudah berbasis web sehingga memudahkan pengguna dalam men-*sharing* data kepada pengguna yang lain dan memiliki sub system pengaduan izin. Kekurangan pada sistem ini belum ada template yang terstruktur sehingga terkadang mempersulit dan tidak rapi. Belum adanya sub sistem untuk mengetahui posisi suatu izin.

Pada judul Implementasi Aplikasi Sistem Informasi Pembuatan Ijin Mendirikan Bangunan pada Unit Pelayanan Terpadu Kodya Denpasar Berbasis Web (Hardi, 2010) memiliki kelebihan pada sistem ini sudah berbasis web sehingga memudahkan pengguna dalam men-*sharing* data kepada pengguna yang lain, kemudahan dalam *input* dan *edit data*, kemudahan dalam pencetakan report-report yang dibutuhkan.

Sedangkan kekurangan sistem ini belum memiliki template yang memudahkan pengembang dalam mengembangkan aplikasi tersebut untuk diintegrasikan dengan izin-izin lainnya.

Pada judul Sistem Informasi Manajemen Pelayanan Perizinan Terpadu Satu Pintu Berbasis GIS (Heru, 2010). Sistem yang di bangun sudah dilakukan dengan terkomputerisasi dan berbasis web dan terintegrasi dengan letak dan kondisi geografis serta kemudahan dalam mencetak report-report yang dibutuhkan. Namun memiliki kekurangan pada database yang digunakan, karena menggunakan MS Sql Sever 2005D.E yang hanya bisa berjalan pada Sistem Operasi Windows yang sifatnya komersial.

Pada judul Perancangan Sistem Informasi Pendapatan Asli Daerah (PAD) Berbasis Database Pada Instansi Pemerintah (Almilia, 2006), memiliki kekurangan Lisensi berbayar untuk Sistem Operasi dan database yang digunakan (MS. Access), Tidak memiliki menu pengaturan untuk penambahan user dan pemberian hak akses.

Pada Judul Sistem Informasi Izin Mendirikan Bangunan (Imb) Pada Kantor Unit Pelayanan Perizinan Satu Atap (Uppsa) Kota Gorontalo (Nento, 2011). Kelebihan yang terdapat pada sistem ini memberikan kemudahan dalam proses entry dan *update* data-data permohonan izin mendirikan bangunan. Mempermudah pembuatan report-report yang dibutuhkan secara otomatis. Kekurangan dari sistem ini diantaranya tidak dapat membuat user dan pemberian hak akses, pengembangan basis data masih menggunakan Microsoft Access 2003 dan pengembangan aplikasi menggunakan Microsoft Visual Basic 6.0 yang sifatnya komersial dan belum memiliki template yang baik.

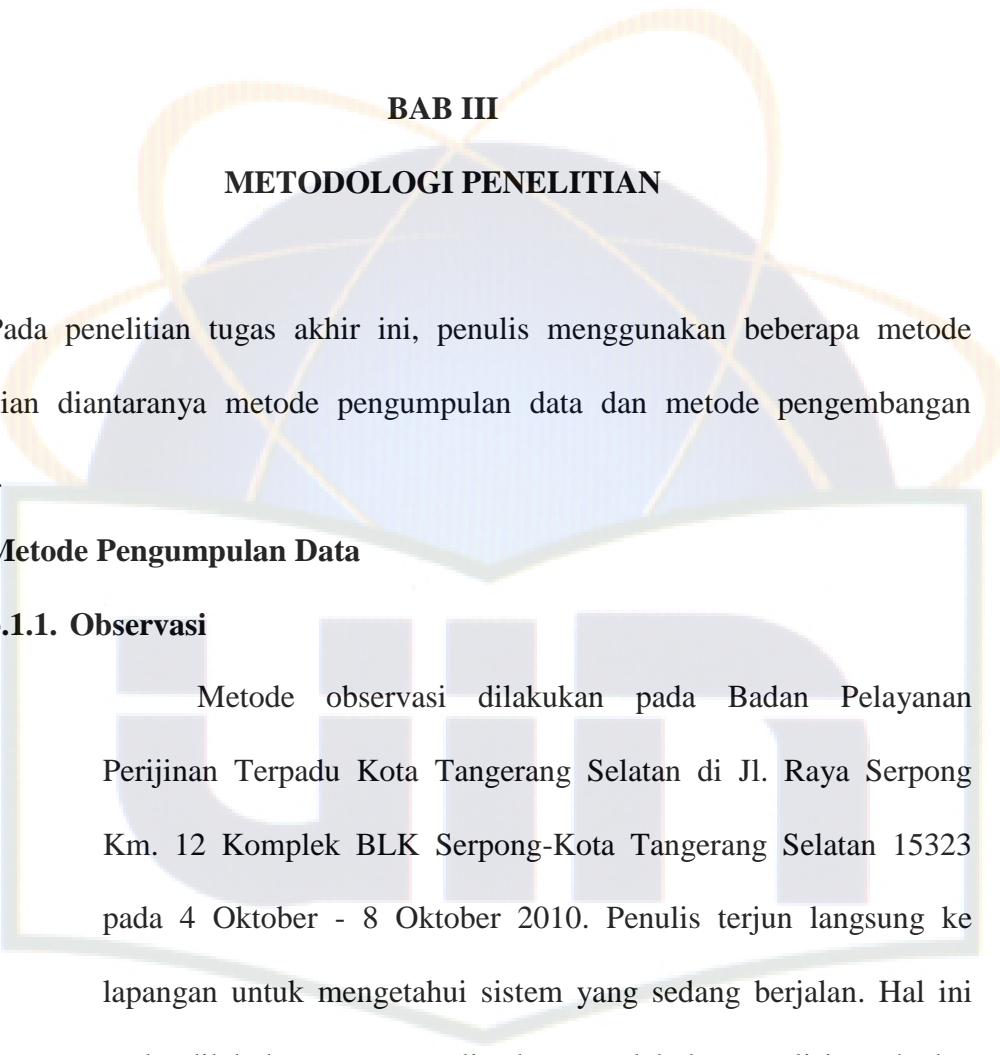
Pada Judul Sistem Sistem Informasi Pembuatan Surat Ijin Tempat Usaha Pada Unit Pelayanan Perijinan Satu Atap Walikota Gorontalo (Ester, 2011). Kelebihan yang terdapat pada sistem ini memberikan kemudahan pembuatan report-report yang dibutuhkan secara otomatis. Kekurangan dari sistem ini diantaranya tidak dapat membuat user dan pemberian hak akses, pengembangan basis data masih menggunakan Microsoft Access 2003 dan pengembangan aplikasi menggunakan Microsoft Visual Basic 6.0 yang sifatnya komersial dan belum memiliki template yang baik.

Pada Judul Sistem Informasi Pengurusan Izin Trayek Angkutan Antar Kota di Provinsi Gorontalo (Fadli Umar, 2010). Kelebihan yang terdapat pada sistem ini memberikan kemudahan pembuatan report-report yang dibutuhkan secara otomatis. Kekurangan dari sistem ini diantaranya tidak dapat membuat user dan pemberian hak akses, pengembangan basis data masih menggunakan Microsoft Access 2003 dan pengembangan aplikasi menggunakan Microsoft Visual Basic 6.0 yang sifatnya komersial dan belum memiliki template yang baik.

Pada Aplikasi Pengolahan Data Izin Usaha Jasa Konstruksi (IUJK) pada Dinas PU Bina Marga dan PSDA Kota Palembang (Juliansyah, 2010). Kelebihan pada sistem ini sudah berbasis web sehingga memudahkan pengguna dalam men-*sharing* data kepada pengguna yang lain. Kelemahannya belum memiliki template yang baik untuk pengembangan dan diintegrasikan dengan izin yang lain.

Pada Judul Sistem Informasi Catatan Sispil pada Unit Pelayanan Perizinan Satu Atap Kota Gorontalo (Zanial, 2010). Kelebihan pada sistem ini memberikan kemudahan dalam proses entry dan *update* data-data. Mempermudah pembuatan report-report yang dibutuhkan secara otomatis. Kekurangan dari pengembangan aplikasi menggunakan Microsoft Fisual Foxpro 9.0. yang sifatnya komersia.





## BAB III

### METODOLOGI PENELITIAN

Pada penelitian tugas akhir ini, penulis menggunakan beberapa metode penelitian diantaranya metode pengumpulan data dan metode pengembangan sistem.

#### 3.1. Metode Pengumpulan Data

##### 3.1.1. Observasi

Metode observasi dilakukan pada Badan Pelayanan Perijinan Terpadu Kota Tangerang Selatan di Jl. Raya Serpong Km. 12 Komplek BLK Serpong-Kota Tangerang Selatan 15323 pada 4 Oktober - 8 Oktober 2010. Penulis terjun langsung ke lapangan untuk mengetahui sistem yang sedang berjalan. Hal ini perlu dilakukan agar penulis dapat melakukan analisis terhadap sistem yang telah berjalan.

Dari hasil observasi yang ada, pada sistem yang berjalan terdapat masalah yaitu ditemukan ketidaksesuaian data mulai dari pemasukan data sampai proses penerbitan ijin, sulitnya Perhitungan biaya retribusi Ijin mendirikan Bangunan, sehingga kemungkinan dapat terjadi kesalahan-kesalahan dalam perhitungan, Rata-rata sehari permohonan IMB sebanyak 30 permohonan, sehingga membutuhkan waktu rata-rata 10 hari dalam melakukan proses

pembuatan laporan-laporan (Masing-masing bagian memiliki tugas dalam pembuatan laporan-laporan).

Penulis mengamati alur proses perijinan mulai dari bagian pendaftaran, bagian pengawasan dan pengendalian, bendahara sampai dengan terbitnya Ijin serta mengumpulkan data-data dari masing-masing bagian tersebut. Bukti observasi dapat dilihat di lampiran.

### **3.1.2. Wawancara**

Penulis melakukan wawancara kepada pihak-pihak yang berhubungan dengan aplikasi yang dibuat. Penulis melakukan wawancara dengan bapak Drs. Iskandar sebagai Kepala Seksi Pelayanan Perijinan Bidang Pembangunan untuk mengetahui proses-proses secara umum dalam ijin mendirikan bangunan. Mulai dari pendaftaran sampai dengan terbitnya Ijin, Faktor-faktor yang mempengaruhi perhitungan Ijin Mendirikan Bangunan. Hasil wawancara disajikan pada bagian Lampiran.

### **3.1.3. Studi Pustaka**

Pada tahapan pengumpulan data dengan cara studi pustaka, penulis mencari referensi-referensi yang relevan dengan objek yang diteliti. Pencarian referensi dilakukan di perpustakaan, toko buku, maupun secara *online* melalui internet. Setelah mendapatkan

referensi-referensi yang relevan, penulis lalu mencari informasi-informasi yang dibutuhkan dalam penulisan. Informasi yang didapatkan digunakan dalam penyusunan landasan teori, metodologi penelitian serta pembuatan aplikasi. Pustaka-pustaka yang dijadikan acuan dapat dilihat di Daftar Pustaka.

### 3.2. Metodologi Pengembangan Sistem

Metodologi pengembangan sistem yang penulis gunakan dalam penelitian ini adalah metode *Extreme Programming* (XP). Pemilihan metode ini dilakukan dengan alasan :

1. requirement sistem yang berubah dengan cepat, karena sistem yang digunakan disesuaikan dengan kebutuhan user, sehingga ketika ada perubahan yg diminta oleh *client/user* perubahan langsung bisa dilakukan.
2. Aplikasi yang dikembangkan berfokus pada *coding* dan *testing*. Selain itu, metode ini digunakan karena aplikasi yang dikembangkan dikerjakan secara tim sebanyak 2 orang.
3. Alasan lain pemilihan metode ini adalah adanya keterlibatan user dalam pembangunan aplikasi misalnya pada tahap planning dengan mengumpulkan user stories.
4. Selain itu, extreme programming dipilih karena waktu penggerjaan sistem yang singkat.

Tahapan-tahapan yang dilakukan oleh penulis dalam melakukan penelitian pengembangan aplikasi ini adalah:

### **3.2.1. *Planning***

Tahapan ini adalah tahapan perencanaan pada sistem yang akan dikembangkan. Pada tahapan ini penulis melakukan:

- a. Pengumpulan permintaan user (*user stories*) dari pendaftaran, pengawasan dan pengendalian, bendahara, penomoran dan penyerahan SK, sehingga output yang dihasilkan dari sistem sesuai dengan keinginan para user.
- b. Pengamatan terhadap sistem yang sedang berjalan di Pelayanan Perijinan Terpadu (BP2T) Kota Tangerang Selatan mulai dari Pendaftaran sampai terbitnya surat izin.
- c. Melakukan identifikasi masalah yang terjadi pada sistem yang sedang berjalan.
- d. Menentukan alur bisnis dan aplikasi serta wilayah persoalan data yang akan didukung oleh sistem yang akan dikembangkan serta ditentukan pula jangkauan atau batasan sistem.

### **3.2.2. *Design***

Setelah mengetahui definisi aplikasi yang akan dikembangkan maka tahapan berikutnya adalah melakukan perancangan (*design*). Perancangan di sini dimaksudkan untuk membuat pemodelan terhadap

aplikasi baru yang dapat mewakili sistem yang berjalan saat ini di Pelayanan Perijinan Terpadu (BP2T) Tangerang. Design yang dimaksud meliputi perancangan aplikasi dan perancangan database.

#### a. *Design Aplikasi*

Untuk perancangan aplikasi, penulis menggunakan alat bantu (*tools*) yaitu CRC (*Class – Responsibility – Collaborator*), pada bagian ini penulis mendefinisikan *class-class* yang akan digunakan pada sistem. Selain itu, aplikasi juga didesain menggunakan *Unified Modelling Language (UML)* sebagai bahan untuk melakukan pendokumentasian dalam pengembangan sistem. Hal ini dilakukan untuk memudahkan memberikas sketsa awal aplikasi, selain itu penggunaan UML lebih cocok digunakan dalam perancangan aplikasi yang bersifat *object oriented*. Perancangan aplikasi yang penulis lakukan dengan menggunakan *tools* UML ini meliputi:

1. Penentuan *Actor* (subbab 4.3.2)
2. Perancangan *Use Case Diagram* (subbab 4.3.3)
3. Perancangan *Use Case Scenario* (subbab 4.3.4)
4. Perancangan *Activity Diagram* (subbab 4.3.5)
5. Perancangan *Sequence Diagram* (subbab 4.3.6)
6. Perancangan *Class Diagram* (subbab 4.3.7)

Dalam perancangan dengan UML ini, penulis menggunakan *software Rational Rose*. Secara detail design aplikasi dapat dibaca di subbab 4.3.

#### **b. *Design Basis Data***

Data-data yang digunakan dalam aplikasi ini akan disimpan ke dalam database. Pada tahapan ini penulis melakukan Penerjemahan *class diagram* ke dalam bentuk *entity* (subbab 4.3.8)

Secara detail design basis data dapat dibaca disubbab 4.3.8

#### **c. *Design Tampilan***

Pada tahap ini, penulis melakukan perancangan terhadap *user interface* dari aplikasi ini. Perancangan yang dilakukan meliputi halaman-halaman yang ada di dalam sistem. Terdapat Tampilan login, pendaftaran, pemeriksaan, penetapan, pembayaran, penomoran dan penyerahan. Secara detail dapat dibaca disubbab 4.3.9.

### **3.2.3. *Coding***

Pada tahapan coding, sebelum dilakukan *developing* aplikasi berdasarkan tahapan-tahapan sebelumnya, *unit test* untuk setiap *user stories* disiapkan. penulis melakukan *deployment* terhadap aplikasi yang telah dikembangkan ke target *server* tertentu.

Setelah *unit test* disiapkan, kemudian baru dilakukan pembangunan aplikasi berdasarkan *design* yang sebelumnya telah dilakukan.

Pada implementasi aplikasi, penulis melakukan pengembangan aplikasi dengan mengacu pada *design* aplikasi yang telah dilakukan. Pada *coding* karena sistem dikerjakan secara tim maka diterapkan konsep *pair programming* yaitu konsep dimana programmer duduk bersama untuk saling membantu. Selain itu pada tahapan ini, dilakukan *refactoring* untuk meningkatkan kualitas dari struktur program. Pada tahapan ini juga *build* program dilakukan sesering mungkin sehingga eror pada program dapat terdeteksi secepat mungkin. Secara detail dapat dilihat dilampiran.

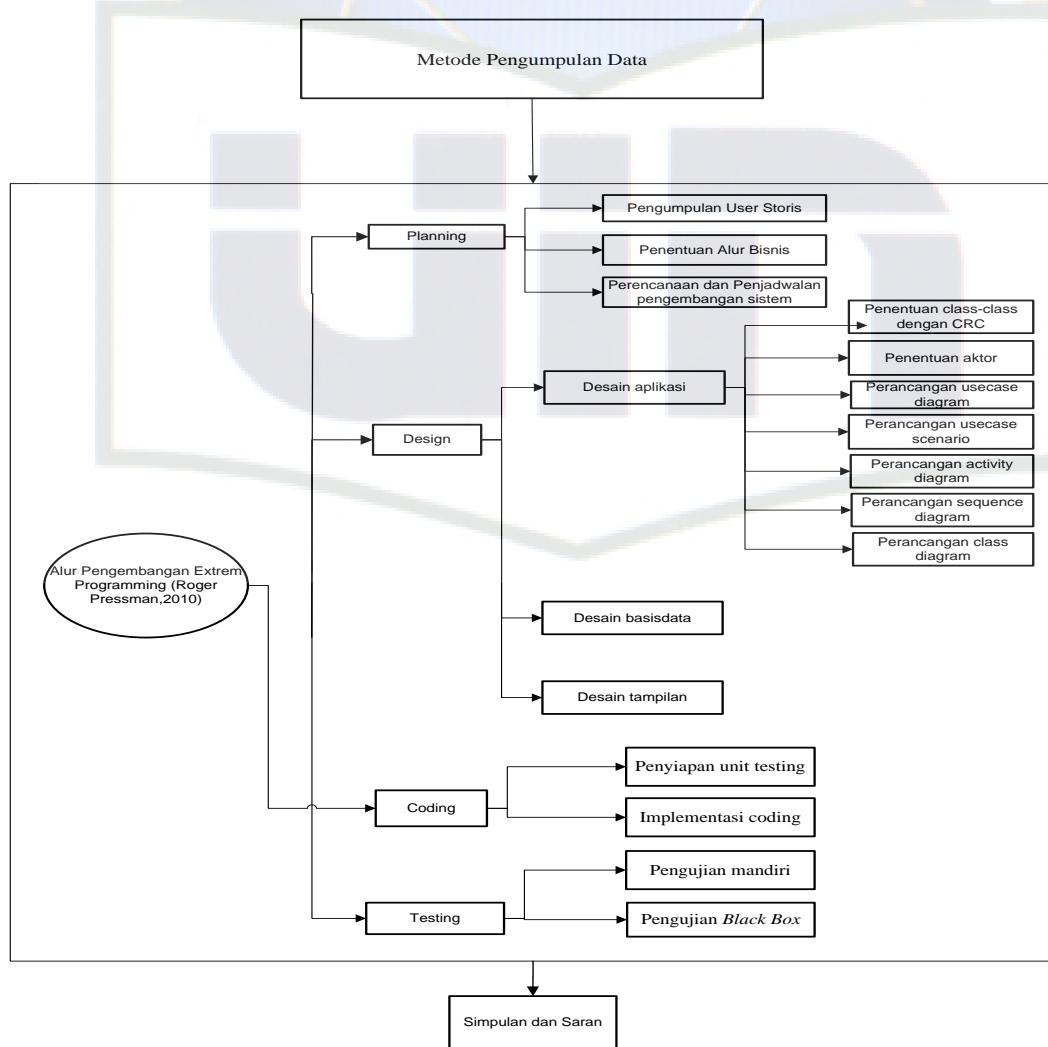
### 3.2.4. *Testing*

Pada tahapan *testing*, *testing* dilakukan pada unit test yang sebelumnya telah disiapkan, yaitu dengan menguji *source code* yang digunakan. Setelah unit test, dilakukan acceptance test dengan melakukan *deployment* terhadap aplikasi yang telah dikembangkan ke target *server* tertentu. Pengujian secara pengujian fungsional aplikasi yang meliputi tampilan data, pemasukan, perubahan data serta pencetakan.

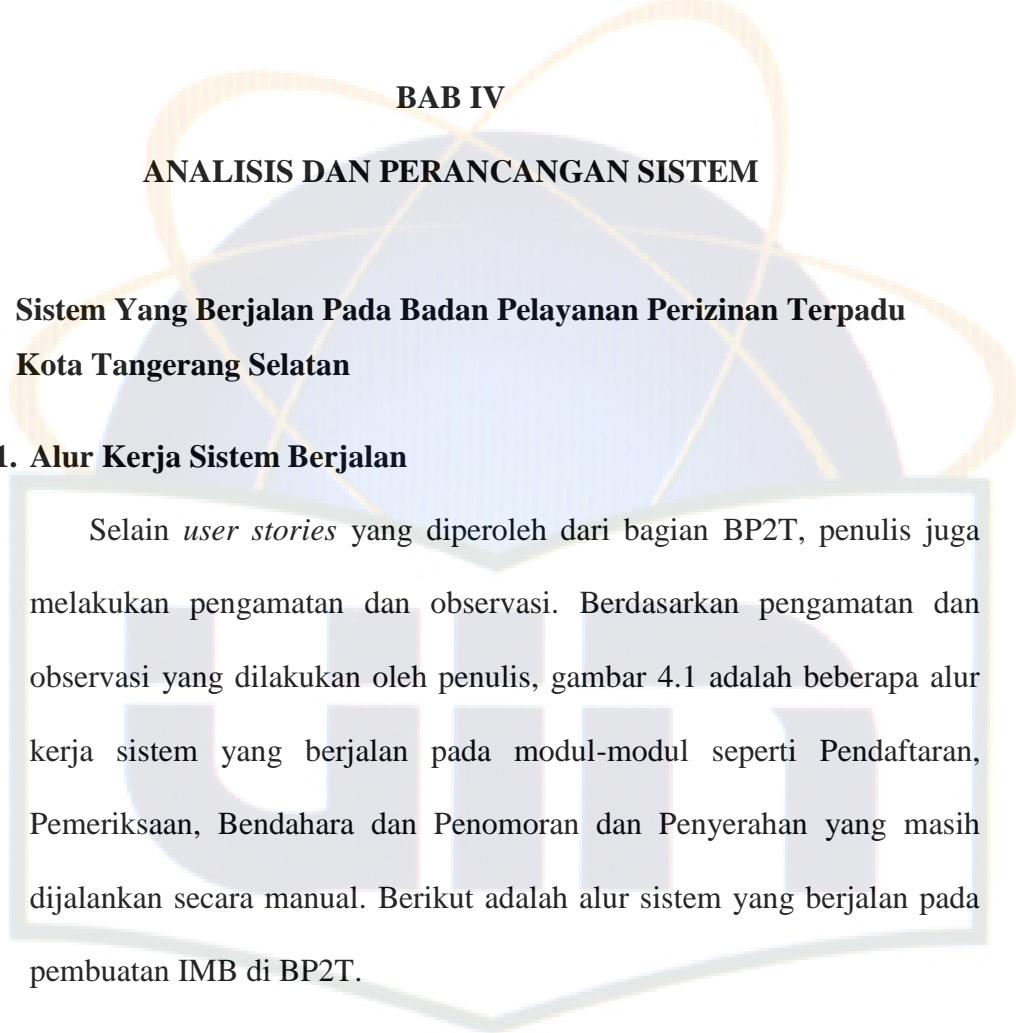
Untuk pengujian ini, penulis menerapkan 2 (dua) macam pengujian yaitu pengujian mandiri oleh penulis dan pengujian oleh user pada masing-masing bagian dengan metode *blackbox*.

### 3.3. Kerangka Berpikir

Dalam melakukan penelitian ini, penulis melakukan tahapan-tahapan kegiatan dengan mengikuti rencana kegiatan yang tertuang dalam kerangka berpikir penelitian ini



**Gambar 3.1** Kerangka Berpikir



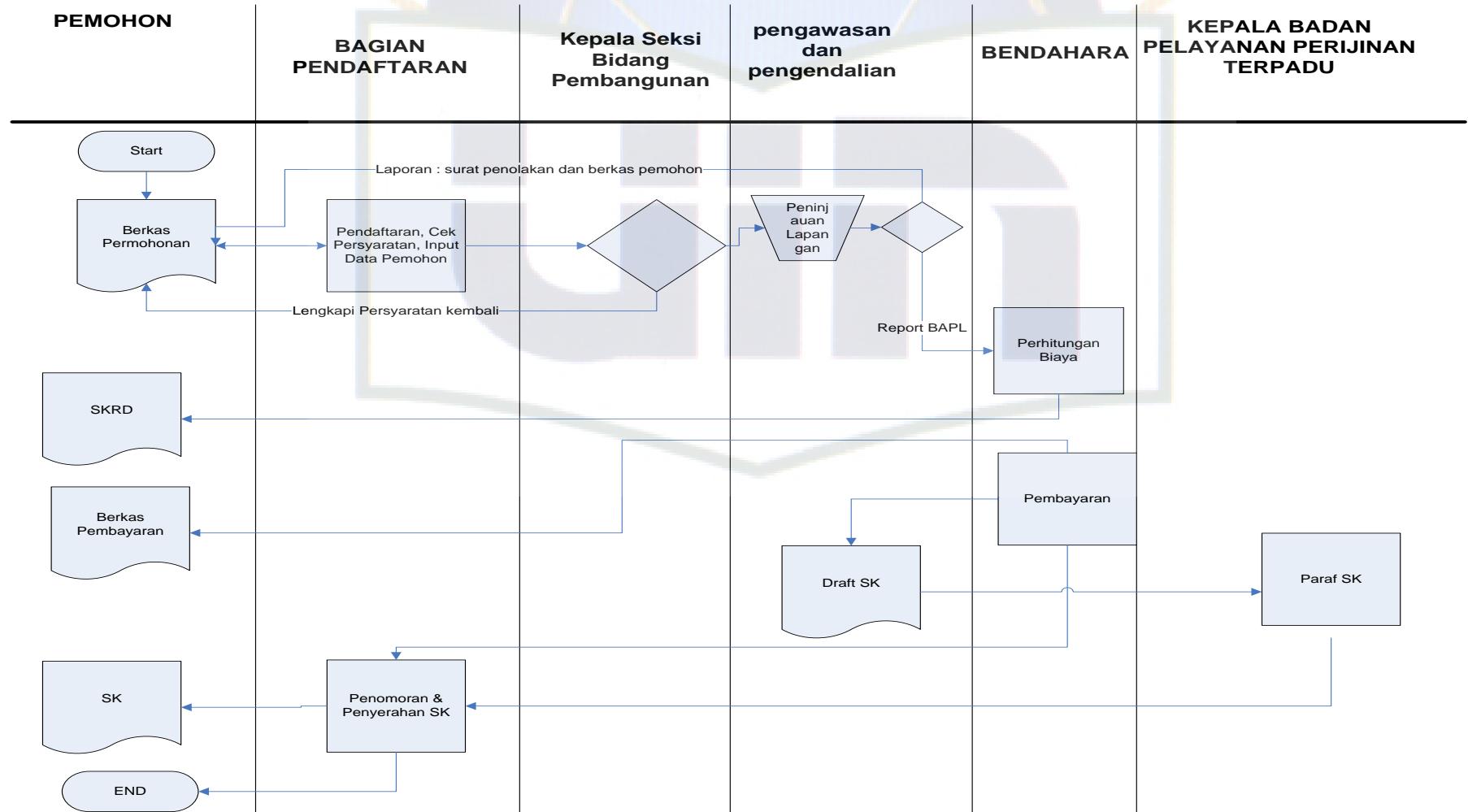
## BAB IV

### ANALISIS DAN PERANCANGAN SISTEM

#### 4.1. Sistem Yang Berjalan Pada Badan Pelayanan Perizinan Terpadu Kota Tangerang Selatan

##### 4.1.1. Alur Kerja Sistem Berjalan

Selain *user stories* yang diperoleh dari bagian BP2T, penulis juga melakukan pengamatan dan observasi. Berdasarkan pengamatan dan observasi yang dilakukan oleh penulis, gambar 4.1 adalah beberapa alur kerja sistem yang berjalan pada modul-modul seperti Pendaftaran, Pemeriksaan, Bendahara dan Penomoran dan Penyerahan yang masih dijalankan secara manual. Berikut adalah alur sistem yang berjalan pada pembuatan IMB di BP2T.



Gambar 4.1 Alur kerja sistem yang berjalan pada proses pembuatan IMB.

#### 4.1.2. Identifikasi Masalah

Berdasarkan gambar alur kerja sistem yang sedang berjalan, maka dapat diidentifikasi kelemahan-kelemahan dari sistem yang sedang berjalan ini, yaitu:

- a. Sering ditemukan ketidaksesuaian mulai dari pemasukan data sampai proses penerbitan izin.
- b. Sulitnya Perhitungan biaya retribusi Izin mendirikan Bangunan, sehingga kemungkinan dapat terjadi kesalahan-kesalahan dalam perhitungan.
- c. Rata-rata sehari permohonan IMB 200 permohonan sehingga membutuhkan waktu 10 hari dalam proses pembuatan laporan-laporan (Masing-masing bagian memiliki tugas dalam pembuatan laporan-laporan).
- d. Tidak Adanya Integrasi data antar masing-masing bagian.
- e. Sulitnya mengetahui posisi suatu berkas permohonan hingga tahap mana yanh sudah dikerjakan.

### 4.2. Perencanaan (*Planning*)

#### 4.2.1. Permintaan User (*User Stories*)

Pada tahap ini penulis melakukan pengumpulan cerita user, *user stories* yang dikumpulkan diantaranya dari pihak Badan Pelayanan Perizinan Terpadu BP2T sebagai pengguna utama dari sistem yang ada, wawancara juga dilakukan kepada pengguna yang terkait langsung dengan penggunaan sistem ini seperti bagian pendaftaran, pengelolaan,

bendahara, penomoran dan penyerahan dan pihak admin selaku admin utama.

### 1. Pendaftaran IMB.

**Tabel 4.1** index card untuk pendaftaran

Project	: Pendaftaran IMB
Interview Date	: Oktober 2010
Interviewer	: Peneliti
Interviewee	: bagian pendaftaran IMB.
Summary	: Pada sistem diharapkan dapat merekam data pemohon yang mengajukan permohonan IMB dengan cepat dan langsung terhubung untuk ke tahapan berikutnya.
No.	Description
1.	<p>Bagian pendaftaran mencatat berkas-berkas data yang menjadi syarat-syarat pemohon. Yaitu</p> <ol style="list-style-type: none"> <li>1. Formulir Permohonan (harus terlampir)</li> <li>2. Foto Copy KTP (harus terlampir)</li> <li>3. Foto Copy Surat Tanah (harus terlampir)</li> <li>4. Foto Copy Site Plan(harus terlampir)</li> <li>5. Gambar Bangunan 2 (dua) Rangkap (harus terlampir)</li> <li>6. Foto Copy SPPT dan STTS Tahun Terakhir (harus terlampir)</li> <li>7. Foto Copy IPR (jika dibutuhkan)</li> </ol>

	<p>8. IMB Lama (jika dibutuhkan)</p> <p>9. Foto Copy Izin Lingkungan/ Tetangga (jika dibutuhkan)</p> <p>10. Surat Keterangan Persetujuan Warga (jika dibutuhkan)</p> <p>11. Foto Copy Pengesahan Site Plan (jika dibutuhkan)</p> <p>12. Kartu Keluarga (KK) (jika dibutuhkan)</p> <p>13. Foto Copy NPWP (jika dibutuhkan)</p> <p>14. Foto Copy Akte Perusahaan (jika dibutuhkan)</p> <p>15. Perhitungan Kontruksi untuk &gt; 2 Lantai dan Bentangan &gt; 15 Meter (jika dibutuhkan)</p> <p>16. Foto Copy Amdal/ UKL/ UPL (jika dibutuhkan)</p> <p>17. Foto Copy Peil Banjir (jika dibutuhkan)</p> <p>18. Foto Copy Izin Lokasi (jika dibutuhkan)</p> <p>19. Amdal Lalu Lintas (jika dibutuhkan)</p> <p>20. Surat Kuasa (jika dibutuhkan)</p>
2.	Pendaftar merekam data kedalam Sistem yang dapat langsung terhubung ketahapan berikutnya.
3.	Dapat melihat data berkas pengaju permonan IMB.
4.	Dapat mencetak berkas dalam bentuk pdf.

2. Pemeriksaan Permohonan IMB.

**Tabel 4.2** index card untuk pengawasan dan pengendalian

Project : Pemeriksaan berkas	
Project	: Verifikasi berkas IMB
Interview Date	: Oktober 2010
Interviewer	: Peneliti
Interviewee	: Bagian Pengelolaan
Summary	: Pada sistem diharapkan dapat melihat berkas pendaftaran pemohon yang ada dan kemudian bagian pengawasan dan pengendalian merekam hasil dari peninjauan lapangan permohonan IMB.
No.	Description
1.	Bagian pengelolaan melakukan verifikasi dari data permohonan dan memeriksa kelengkapan-kelengkapan apakah telah sesuai dan valid kemudian merekam hasil peninjauan lapangan.
2.	Bagian pengelolaan mengisi komponen-komponen bangunan, luas serta fungsi bangunan dan menentukan apakah izin permohonan tersebut berhak layak Iizin Mendirikan Bangunan.
3.	Mencetak hasil peninjauan lapangan berkas dalam bentuk pdf

3. Penetapan biaya dan Pembayaran Retribusi.

**Tabel 4.3** index card untuk Penetapan Pembayaran Retribusi

Project	: Penetapan Pembayaran Retribusi
Interview Date	: Oktober2010
Interviewer	: Peneliti
Interviewee	: Bendahara
Summary	: Proses Penetapan biaya kemudian pembayaran retribusi jika data dari pihak kelola telah diproses.
No.	Description
1.	Bendahara melakukan penghitungan biaya retribusi..
2.	Mengeluarkan Surat Ketetapan Retribusi Daerah dan Nota Penghitungan
3.	Bendahara melakukan pendataan pembayaran retribusi.

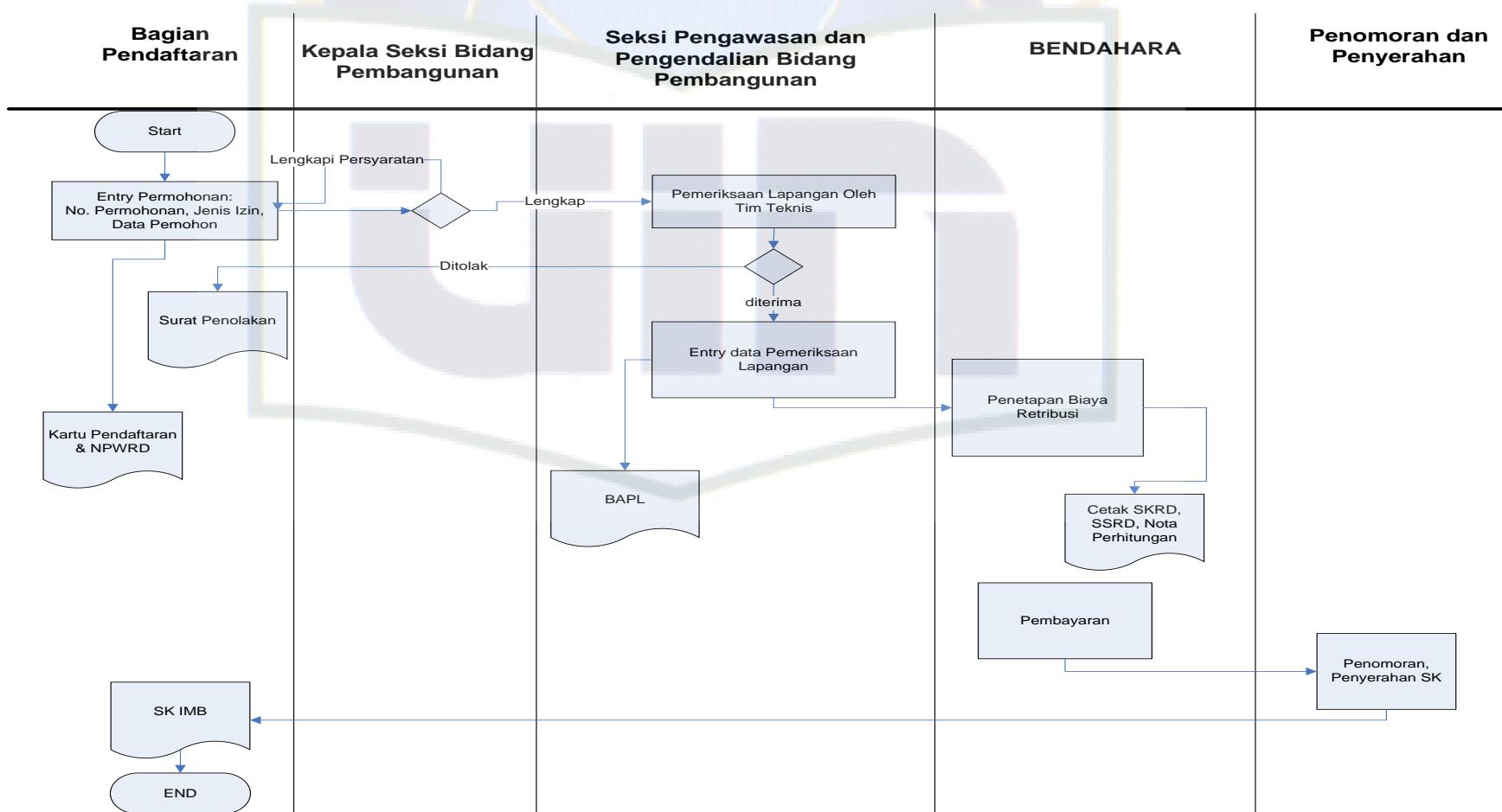
#### 4. Penomoran dan Penyerahan.

**Tabel 4.4** index card Untuk Penomoran dan Penyerahan

Project	: Penomoran dan Penyerahan
Interview Date	: Oktober 2010
Interviewer	: Peneliti
Interviewee	: Penomoran dan Penyerahan
Summary	: Proses penomoran berkas apabila telah melakukan pembayaran retribusi dan penyerahan berkas yang telah dibayar.
No.	Description
1.	Penomoran pada berkas yang telah dibayar.
2.	Memasukan data yang mengambil IMB seperti nama dan nomor KTP.
3.	Mencetak surat izin dalam bentuk pdf.

#### 4.2.2. Uraian Singkat Sistem yang Diusulkan

Untuk menjawab permasalahan-permasalahan yang dihadapi, penulis bermaksud mengusulkan pembuatan aplikasi memudahkan pengorganisasian data. Pada pengembangannya, penulis mengusulkan adanya integrasi ke Bagian yang terkait dalam melakukan proses pengurusan IMB.



Gambar 4.2 Alur kerja sistem yang diusulkan pada proses pembuatan IMB

Alur kerja sistem yang diusulkan pada permohonan IMB adalah pemohon mengajukan permohonan pembuatan IMB pada bagian pendaftaran. Pemohon menyiapkan lampiran syarat-syarat kelengkapan yang dibutuhkan dalam pengisian form penginputan data yang ada pada bagian pendaftaran. Lampiran syarat-syarat yang dibutuhkan yaitu fotokopi KTP, PBB terakhir, bukti kepemilikan atau penguasaan tanah, akte pendirian perusahaan bagi yang berbadan hukum, NPWP bagi yang berbadan hukum, IPR, Site Plan, Rekomendasi Dinas terkait, Gambar bangunan tiga rangkap, pengisian data diri, pengisian data komponen-komponen bangunan, dan pengisian lokasi yang akan dibuat permohonan izin.

Setelah kelengkapan berkas-berkas lengkap dan sudah di proses pada bagian pendaftaran maka data tersebut telah tersimpan di dalam database. Berkas tersebut diverifikasi kembali oleh Kepala Seksi Bidang Pembangunan, Kemudian bagian Pengawasan dan Pengendalian melakukan pemeriksaan ke lapangan untuk menyesuaikan apakah data yang telah diajukan pemohon sesuai atau tidak. Bagian Pengawasan dan Pengendalian dapat menolak pengajuan permohonan IMB apabila data yang diajukan tidak valid. Setelah pemeriksaan lapangan selesai maka akan diproses pada sistem dan kemudian bagian data-data di verifikasi dan disimpan. Bagian Pengawasan dan Pengendalian kemudian dapat mencetak berita acara pemeriksaan lapangan dalam bentuk pdf.

Proses sistem kemudian berlanjut pada bagian Bendahara. Bendahara menghitung penetapan biaya retribusi yang harus dibayar oleh pemohon. Bendahara kemudian mencetak SKRD dan nota penghitungan apabila biaya retribusi yang terbilang untuk pemohon telah dibayarkan oleh pemohon itu sendiri. Bukti SKRD ini kemudian diserahkan untuk pemohon dan nota perhitungan sebagai pertinggalan untuk bagian bendahara.

Setelah biaya retribusi dibayarkan maka pemohon dapat mengambil IMB setelah izin dicetak dan diberi nomor oleh pada bagian Penomoran dan Penyerahan. Bagian Penomoran dan Penyerahan mengentri nomor IMB untuk pemohon dan kemudian mencatat data diri pemohon yang mengambil IMB tersebut.

Pada sistem yang diusulkan ini, penulis menggunakan CRC card, dan *Unified Modelling Language* (UML) dalam perancangannya. Diagram-diagram UML yang digunakan yaitu *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence diagram*.

### **4.3. Design Sistem yang Diusulkan**

Desain atau perancangan sistem didefinisikan sebagai tugas yang fokus pada spesifikasi solusi detail berbasis komputer. Jika analisis sistem menekankan pada masalah bisnis, maka sebaliknya desain sistem fokus pada segi teknis atau implementasi sebuah sistem.

Perancangan berorientasi objek menekankan penggambaran model sistem untuk mendokumentasikan aspek teknis dan implementasi dari sebuah sistem. Untuk itu dalam perancangan IMB ini menggunakan CRC card *dan Rational Rose.*

#### 4.3.1. *Design* aplikasi

##### CRC Card

CRC-card adalah daftar *class-class* yang akan digunakan pada saat coding.

Berikut ini 9 *class* yang digunakan:

1. *Class* pendaftar

**Tabel 4.5** *CRC card class* pendaftar

<i>Class</i> : pendaftar	
<i>Description</i> : menerangkan tentang entitas pendaftar	
Resposibilities	Collaborators
1. Mendefenisikan nomor Pendaftaran 2. Mendefinisikan Tanggal pendaftaran 3. Mendefinisikan persyaratan kelengkapan. 4. Mendefiniskan rekomendasi 5. Mendefiniskan Nomor KTP pemohon 6. Mendefiniskan Nama Pemohon 7. Mendefinisikan NPWP pemohon	1. User

8. Mendefinisikan Nama Contact Person	
9. Mendefiniskan Nomor Telepon	
10. Mendefiniskan Nomor HP	
11. Mendefiniskan Tempat lahir	
12. Mendefiniskan Tanggal lahir	
13. Mendefiniskan Alamat	
14. Mendefiniskan Nama Perusahaan	
15. Mendefiniskan Jabatan	
16. Mendefenisikan Alamat Pemohon	
17. Mendefinisikan nomor id imb	

## 2. Class pemeriksaan

**Tabel 4.6 CRC card class pemeriksaan**

<i>Class : pemeriksaan</i>	
<i>Description:</i> menerangkan entitas dari pemeriksaan dan atributnya.	
Resposibilities	Collaborators
1. Mendefinisikan Nomor Pemeriksaan 2. Mendefiniskan Tanggal Pemeriksaan 3. Mendefiniskan Petugas lapangan 4. Menentukan Status Pemeriksaan	1. Pendaftar 2. User 3. Datalapangan 4. rincianbangunan

### 3. Class dataLapangan

**Tabel 4.7 CRC card class datalapangan**

<i>Class : datalapangan</i> <i>Description:</i> menerangkan data lapangan hasil pemeriksaan	
<b>Responsibilities</b>	
Collaborators	
1. Mendefinisikan batas utara 2. Mendefinisikan batas timur 3. Mendefinisikan batas utara 4. Mendefinisikan batas selatan 5. Mendefinisikan GSB 6. Mendefinisikan GSP 7. Mendefinisikan KDB 8. Mendefinisikan KLB 9. Mendefinisikan GSP Samping 10. Mendefinisikan GSB samping 11. Mendefinisikan Fisik Terbangun 12. Mendefinisikan Jumlah lantai 13. Mendefinisikan Lebar Bentangan, 14. Mendefinisikan Struktur Konstruksi (Pondasi, Dinding, atap, lantai, Rangka Kap, Struktur kolom dan Balok.	1. pendaftar

#### 4. Class rincianbangunan

**Tabel 4.8 CRC card class rincianbangunan**

<i>Class : rincianbangunan</i> <i>Description:</i> menerangkan data rincianbangunan hasil pemeriksaan	
Responsibilities	Collaborators
1. Mendefinisikan nama detail bangunan 2. Mendefinisikan luas bangunan 3. Mendefinisikan fungsi bangunan 4. Mendefinisikan koefisien bangunan 5. Mendefinisikan jenis bangunan 6. Mendefinisikan ketetapan biaya 7. Mendefinisikan guna bangunan 8. Mendefinisikan jumlah unit 9. Mendefiniskan satuan 10. Mendefiniskan tarif	1. Pendaftar

### 5. Class user

**Tabel 4.9 CRC card class user**

<i>Class : user</i>	
Resposibilities	Collaborators
<ol style="list-style-type: none"> <li>1. Mendefinisikan nama operator</li> <li>2. Mendefinisikan NIP operator</li> <li>3. Mendefinisikan jabatan</li> <li>4. Mendefinisikan username</li> <li>5. Mendefinisikan password</li> <li>6. Mendefinisikan Email</li> </ol>	

### 6. Class penetapan

**Tabel 4.10 CRC card class penetapan**

<i>Class : penetapan</i>	
Resposibilities	Collaborators
<ol style="list-style-type: none"> <li>1. Mendefinisikan nomor SKRD</li> <li>2. Mendefinisikan tanggal penetapan</li> <li>3. Mendefinisikan npwpd</li> <li>4. Mendefinisikan biaya retribusi</li> </ol>	<ol style="list-style-type: none"> <li>1. Pendaftar</li> <li>2. User</li> <li>3. rincianbangunan</li> </ol>

7. *Class pembayaran*

**Tabel 4.11 CRC card class pembayaran**

<i>Class : pembayaran</i> <i>Description:</i> menerangkan data pembayaran hasil penetapan biaya	
Resposibilities	Collaborators
1. Mendefinisikan nomor bukti pembayaran 2. Mendefinisikan tanggal pembayaran 3. Pembayar	1. Pendaftar 2. User

8. *Class penomoran*

**Tabel 4.12 CRC card class penomoran**

<i>Class : penomoran</i> <i>Description:</i> menerangkan data Penomoran Surat Izin Mendirikan Bangunan	
Resposibilities	Collaborators
1. Mendefinisikan nomor surat izin 2. Mendefinisikan tanggal penomoran	1. Pendaftar 2. User

9. *Class* penyerahan

**Tabel 4.13** CRC card class penyerahan

<i>Class</i> : penyerahan <i>Description</i> : menerangkan data Penyerahan Surat Izin	
Resposibilities	Collaborators
1. Mendefinisikan nama pengambil surat izin 2. Mendefinisikan Nomor Identitas 3. Mendefinisikan tanggal penyerahan	1. Pendaftar 2. User

10. *Class* report

**Tabel 4.14** CRC card class penyerahan

<i>Class</i> : report <i>Description</i> : untuk mencetak Surat report-report	
Resposibilities	Collaborators
1. Menerangkan data pemohon 2. Menerangkan data rincian bangunan 3. Menerangkan pejabat penandatangan	1. Pendaftar 2. rincianbangunan 3. Penomoran

### 4.3.2 Penentuan Aktor dan *use case*

Identifikasi aktor dan *use case* ini didasari pada kebutuhan fungsi-fungsi sistem. Kebutuhan akan fungsi ini digambarkan di *use case*. Selanjutnya *use case* menyediakan nilai hasil kepada aktor.

#### 4. Pendaftaran dan Pengajuan IMB

Tabel 4.15 adalah *requirement* dan *use case* pada pendaftaran dan pengajuan IMB.

**Tabel 4.15** Tabel *requirement* dan *use case* pendaftaran IMB

<i>Requirement</i>	Aktor	<i>Use case</i>
1. Mengisi data permohonan seperti tanggal , jenis permohonan, nomor permohonan.	Petugas Pendaftaran.	Melakukan permohonan.
2. Mengisi isian lampiran status.	Petugas Pendaftaran	Lampiran status.
3. Pendaftaran memasukan data diri pemohon.	Petugas Pendaftaran	Mengisi data pemohon
4. Pendaftaran mengisi komponen data-data jenis bangunan	Petugas Pendaftaran	Menentukan komponen
5. Pendaftar memasukan lokasi pengajuan IMB yang diajukan oleh pemohon	Petugas Pendaftaran	Menentukan lokasi
6. Pendaftar dapat mencetak berkas pemohon	Petugas Pendaftaran	Mencetak berkas pemohon

## 2. Pengawasan dan Pengendalian

Tabel 4.16 adalah *requirement* dan *use case* pada Pengawasan dan Pengendalian.

**Tabel 4.16** Tabel *requirement* dan *use case* Pengawasan dan Pengendalian

<i>Requirement</i>	Aktor	<i>Use case</i>
1. Pengawas mengambil data pemohon dari database	Petugas Pengawasan dan Pengendalian	Memeriksa permohonan.
2. Memverifikasi data komponen bangunan.	Petugas Pengawasan dan Pengendalian	Data komponen bangunan.
3. Pengawas dapat menolak permohonan pengajuan IMB.	Petugas Pengawasan dan Pengendalian	Menolak permohonan.
4. Pengawas mencetak berkas pemeriksaan	Petugas Pengawasan dan Pengendalian	Mencetak berkas pemeriksaan

### 3. Bendahara (Pembayaran Retribusi)

Tabel 4.17 adalah *requirement* dan *use case* pada Bendahara (Pembayaran Retribusi).

**Tabel 4.17** Tabel requirement dan *use case* pada Bendahara

<i>Requirement</i>	Aktor	<i>Use case</i>
1. Menghitung biaya retribusi	Bendahara	Menetapkan permohonan
2. Bendahara mencetak ketetapan retribusi untuk pemohon berupa SKRD dan Nota Penghitungan	Bendahara	Cetak SKRD dan Nota Penghitungan
3. Bendahara menerima dan mencatat Informasi Pembayaran.	Bendahara	Pencatatan Informasi Pembayaran

### 4. Penomoran dan Penyerahan

Tabel 4.18 adalah *requirement* dan *use case* Penomoran dan Penyerahan.

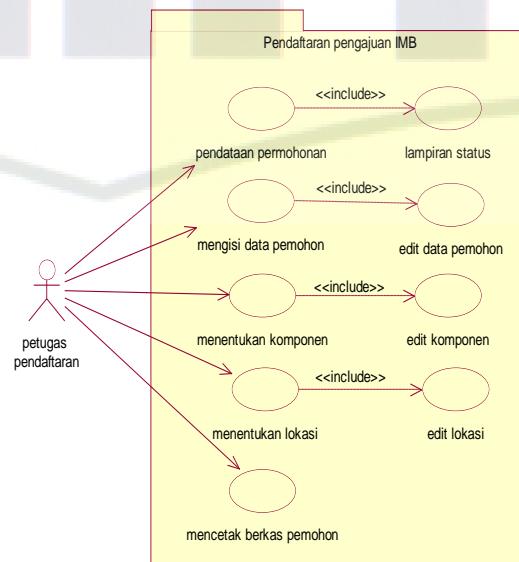
**Tabel 4.18** Tabel *requirement* dan *use case* pada Penomoran dan Penyerahan.

<i>Requirement</i>	Aktor	<i>Use case</i>
1. Melakukan penomoran IMB.	Petugas Penomoran dan Penyerahan	Memasukan penomoran
2. Mencatat data pemohon yang mengambil IMB	Petugas Penomoran dan Penyerahan	Mencatat penyerahan

### 4.3.3 Perancangan *Use Case Diagram*

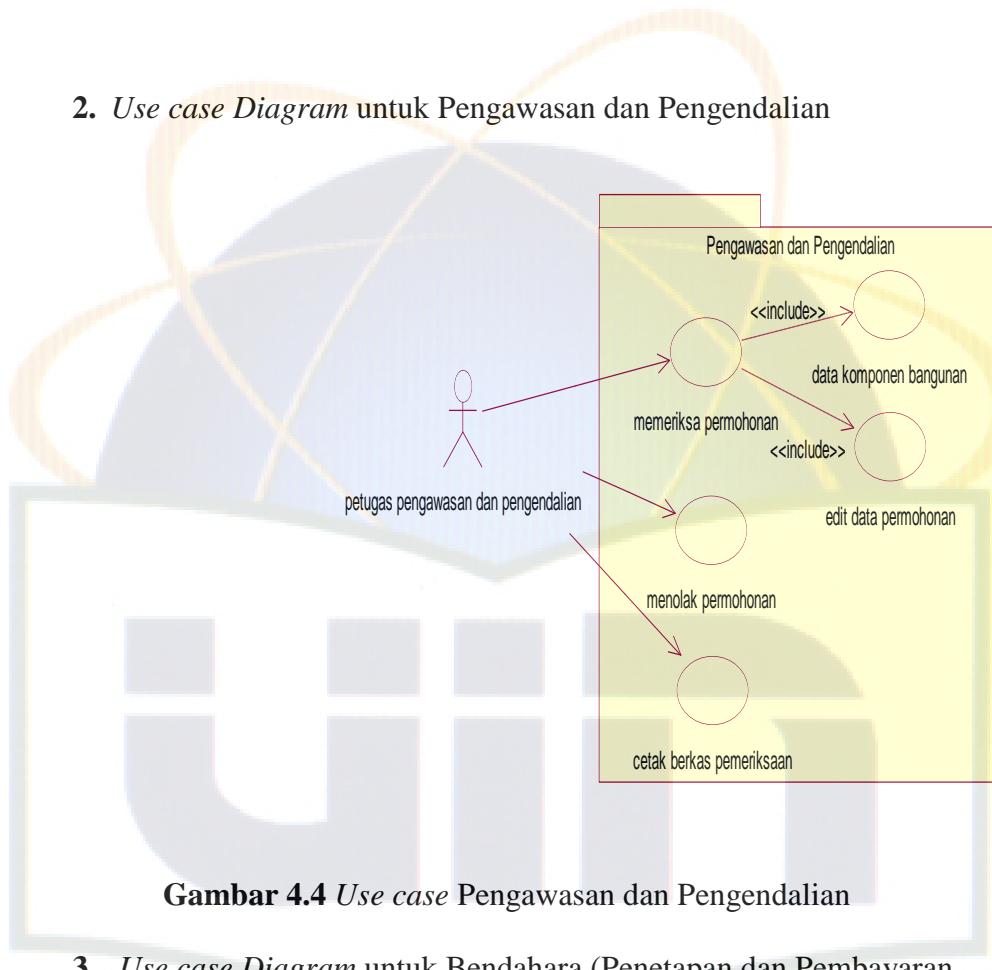
*Use Case Diagram* digunakan untuk menjelaskan apa yang dilakukan oleh sistem serta aktor-aktor yang akan berhubungan dengan proses-proses yang ada pada sistem. Meskipun telah digunakan user stories, tetapi pada pengembangannya, usecase ini diperlukan untuk kebutuhan dokumentasi dan arah pengembangan selanjutnya. Dibawah ini adalah *use case* untuk beberapa sistem yang diusulkan.

1. *Use case Diagram* untuk pendaftaran pengajuan IMB.



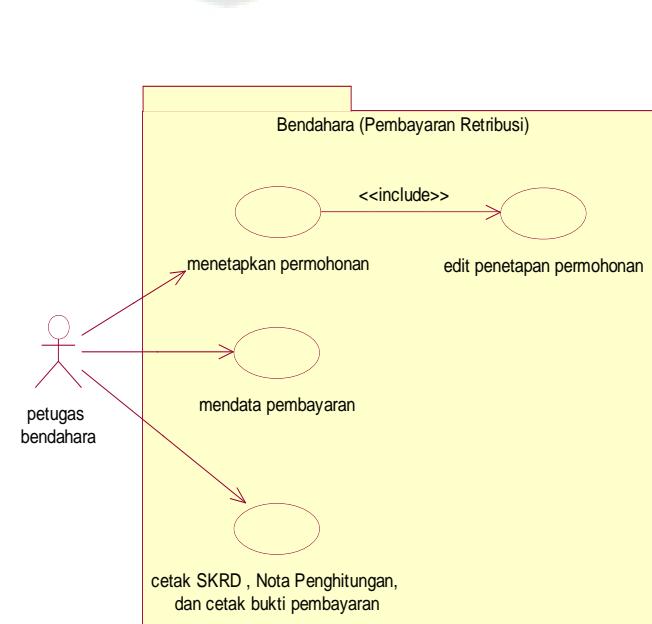
**Gambar 4.3** *Use case* Pendaftaran Pengajuan IMB

## 2. Use case Diagram untuk Pengawasan dan Pengendalian



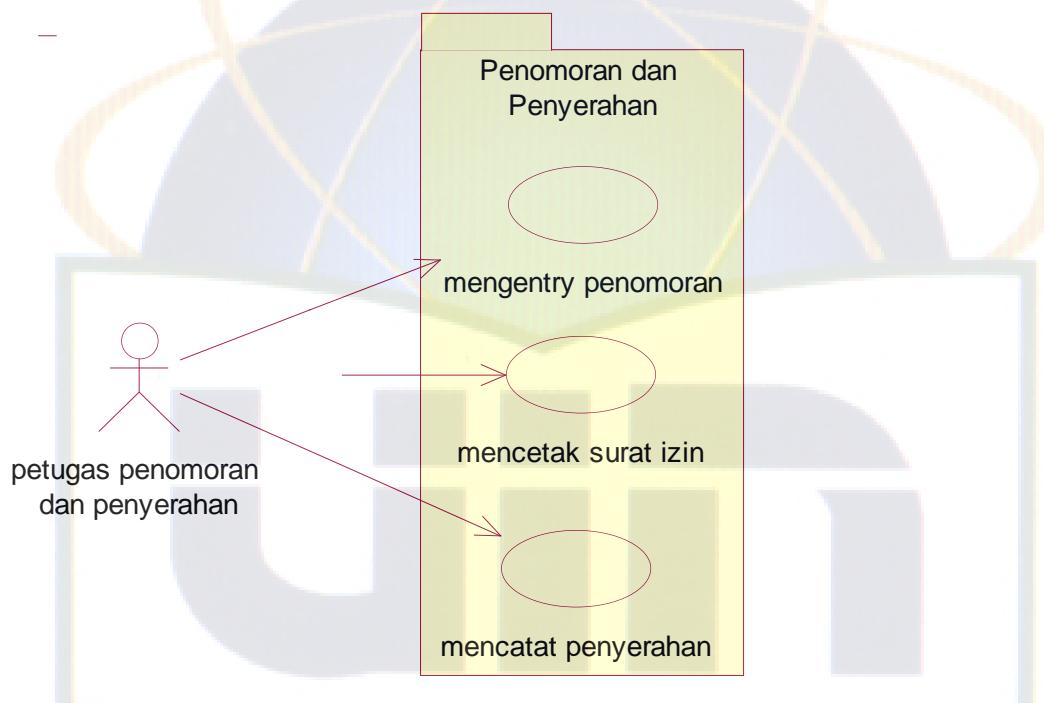
**Gambar 4.4** Use case Pengawasan dan Pengendalian

## 3. Use case Diagram untuk Bendahara (Penetapan dan Pembayaran Retribusi)



**Gambar 4.5** Use case Bendahara (Penetapan dan Pembayaran Retribusi)

#### 4. Use case Diagram untuk Penomoran dan Penyerahan



**Gambar 4.6** Use case Penomoran dan Penyerahan

#### 4.3.4 Use case Scenario

*Use case scenario* merupakan penjelasan yang lebih terperinci mengenai masing-masing *use case* yang terjadi di dalam sistem. Use case scenario ini terdiri dari:

1. Nama *use case* adalah nama *use case* yang akan dideskripsikan.
2. Aktor yang terlibat.
3. Trigger.
4. *Precondition* yang penting bagi *use case* untuk memulai.
5. Action.
6. *Postcondition* yang menjelaskan state dari sistem setelah *use case* berakhir.

Setelah menjelaskan *use case* pada bahasan sebelumnya, maka berikut ini akan dijelaskan spesifikasi *use case* yang telah ditentukan.

##### 1. Pendaftaran dan Pengajuan IMB

- a. Melakukan pendataan permohonan

**Tabel 4.19** *Use case scenario* Melakukan pendataan permohonan

<b>Nama Use case</b>	Melakukan pendataan permohonan.
<b>Aktor yang terlibat</b>	Bagian Pendaftaran.
<b>Trigger</b>	Aktor memasukan data nomor permohonan dan jenis permohonan.
<b>Pre condition</b>	Login sebagai pendaftaran.
<b>Action</b>	Memasukan data.
<b>Post condition</b>	Data telah dimasukan

b. Lampiran status

**Tabel 4.20** Use case scenario Lampiran status

<b>Nama Use case</b>	Lampiran status.
<b>Aktor yang terlibat</b>	Bagian Pendaftaran
<b>Trigger</b>	Aktor memverifikasi lampiran status yang telah diajukan pemohon.
<b>Pre condition</b>	Login sebagai pendaftaran.
<b>Action</b>	Proses data lampiran status.
<b>Post condition</b>	Data sukses diproses.

c. Mengisi data pemohon

**Tabel 4.21** Use case scenario Mengisi data pemohon

<b>Nama Use case</b>	Mengisi data pemohon.
<b>Aktor yang terlibat</b>	Bagian Pendaftaran
<b>Trigger</b>	Aktor memasukan data pemohon berupa data diri.
<b>Pre condition</b>	Login sebagai pendaftaran.
<b>Action</b>	Memasukan data
<b>Post condition</b>	Data telah dimasukan

d. Menentukan Komponen

**Tabel 4.22** Use case scenario Menentukan Komponen

<b>Nama Use case</b>	Menentukan Komponen
<b>Aktor yang terlibat</b>	Bagian Pendaftaran
<b>Trigger</b>	Aktor menentukan komponen bahan bangunan yang diajukan
<b>Pre condition</b>	Login sebagai pendaftaran.
<b>Action</b>	Memasukan data
<b>Post condition</b>	Data telah dimasukan

e. Menentukan lokasi

**Tabel 4.23** *Use case scenario* Menentukan lokasi

<b>Nama Use case</b>	Menetukan lokasi.
<b>Aktor yang terlibat</b>	Bagian Pendaftaran
<b>Trigger</b>	Aktor mengisi tempat atau lokasi bangunan.
<b>Pre condition</b>	Login sebagai pendaftaran.
<b>Action</b>	Simpan data
<b>Post condition</b>	Data Tersimpan dalam database

f. Mencetak berkas pemohon

**Tabel 4.24** *Use case scenario* mencetak berkas pemohon

<b>Nama Use case</b>	Mencetak berkas pemohon.
<b>Aktor yang terlibat</b>	Bagian Pendaftaran
<b>Trigger</b>	Pendaftar mencetak berkas data diri pemohon.
<b>Pre condition</b>	Data sudah harus tersimpan didalam database
<b>Action</b>	Cetak data.
<b>Post condition</b>	Data berhasil tercetak

## 2. Pengawasan dan Pengendalian

### a. Memeriksa permohonan

**Tabel 4.25** Use case scenario Memeriksa permohonan

<b>Nama Use case</b>	Memeriksa permohonan
<b>Aktor yang terlibat</b>	Bagian Pengawasan dan pengendalian
<b>Trigger</b>	Aktor mengambil dari nomor pendaftaran
<b>Pre condition</b>	Mengambil data yang telah tersimpan di dalam database
<b>Action</b>	Mengisi data pemeriksaan kemudian menyimpan data pemeriksaan
<b>Post condition</b>	Data terupdate.

### b. Data komponen bangunan

**Tabel 4.26** Use case scenario Data komponen bangunan

<b>Nama Use case</b>	Data komponen bangunan
<b>Aktor yang terlibat</b>	Bagian Pengawasan dan pengendalian
<b>Trigger</b>	Aktor mengambil data pemohon
<b>Pre condition</b>	Data lapangan dimasukan.
<b>Action</b>	Proses data yang telah di input
<b>Post condition</b>	Data tersimpan didalam data base.

- c. Menolak permohonan

**Tabel 4.27** *Use case scenario* menolak permohonan

<b>Nama Use case</b>	menolak permohonan
<b>Aktor yang terlibat</b>	Bagian Pengawasan dan pengendalian
<b>Trigger</b>	Aktor dapat menolak permohonan pemohon
<b>Pre condition</b>	Data pemeriksaan telah ada di dalam database.
<b>Action</b>	Mengisi Form Penolakan dan Menyimpan data penolakan
<b>Post condition</b>	Permohonan IMB ditolak

- d. Cetak berkas pemeriksaan

**Tabel 4.28** *Use case scenario* Cetak berkas pemeriksaan

<b>Nama Use case</b>	Cetak berkas pemeriksaan
<b>Aktor yang terlibat</b>	Bagian Pengawasan dan pengendalian
<b>Trigger</b>	Aktor mencetak berkas pemeriksaan
<b>Pre condition</b>	Database sudah tersimpan didalam database.
<b>Action</b>	Cetak berkas
<b>Post condition</b>	Pdf

3. Bendahara (Pembayaran Retribusi)

- a. Menetapkan permohonan

**Tabel 4.29** *Use case scenario* Menetapkan permohonan

<b>Nama Use case</b>	Menetapkan permohonan
<b>Aktor yang terlibat</b>	Bagian Bendahara
<b>Trigger</b>	Bendahara menghitung biaya retribusi.
<b>Pre condition</b>	Mengambil data pemohon dari database
<b>Action</b>	Mengisi data komponen, guna bangunan, nilai ketetapan dan Hitung biaya retribusi kemudian disimpan.
<b>Post condition</b>	Simpan data kedalam database

- b. Mendaata Pembayaran

**Tabel 4.30** *Use case scenario* Mendaata Pembayaran

<b>Nama Use case</b>	Mendaata Pembayaran
<b>Aktor yang terlibat</b>	Bagian Bendahara
<b>Trigger</b>	Aktor mencatat Informasi Pembayaran
<b>Pre condition</b>	Mengambil data pemohon dari database
<b>Action</b>	Mengisi dan Menyimpan Informasi Pembayaran
<b>Post condition</b>	Simpan data kedalam database

c. Cetak SKRD dan Nota Penghitungan

**Tabel 4.31 Use case scenario Cetak SKRD dan Nota Penghitungan**

<b>Nama Use case</b>	Cetak SKRD dan Nota Penghitungan
<b>Aktor yang terlibat</b>	Bagian Bendahara
<b>Trigger</b>	Aktor dapat mencetak nota penghitungan dan SKRD
<b>Pre condition</b>	Data penetapan biaya retribusi telah ada di dalam database.
<b>Action</b>	Cetak data
<b>Post condition</b>	Pdf

4. Penomoran dan Penyerahan

a. Mengentry penomoran

**Tabel 4.32 Use case scenario Mengentry penomoran**

<b>Nama Use case</b>	Mengentry penomoran
<b>Aktor yang terlibat</b>	Bagian Penomoran dan Penyerahan
<b>Trigger</b>	Mengambil dari nomor pendaftaran
<b>Pre condition</b>	Data sudah ada didalam database
<b>Action</b>	Memasukan form penomoran dan disimpan
<b>Post condition</b>	Data di update.

- b. Mencetak Surat Izin

**Tabel 4.33** *Use case scenario* Mencetak Surat Izin

<b>Nama Use case</b>	Mencetak Surat Izin
<b>Aktor yang terlibat</b>	Bagian Penomoran dan Penyerahan
<b>Trigger</b>	Aktor menekan tombol preview surat izin dan memilih Pejabat penanda tangan
<b>Pre condition</b>	Data Penomoran telah diisi
<b>Action</b>	Aktor mencetak Surat Izin Mendirikan Bangunan
<b>Post condition</b>	Surat Izin telah di cetak

- c. Mencatat penyerahan

**Tabel 4.34** *Use case scenario* Mencatat penyerahan

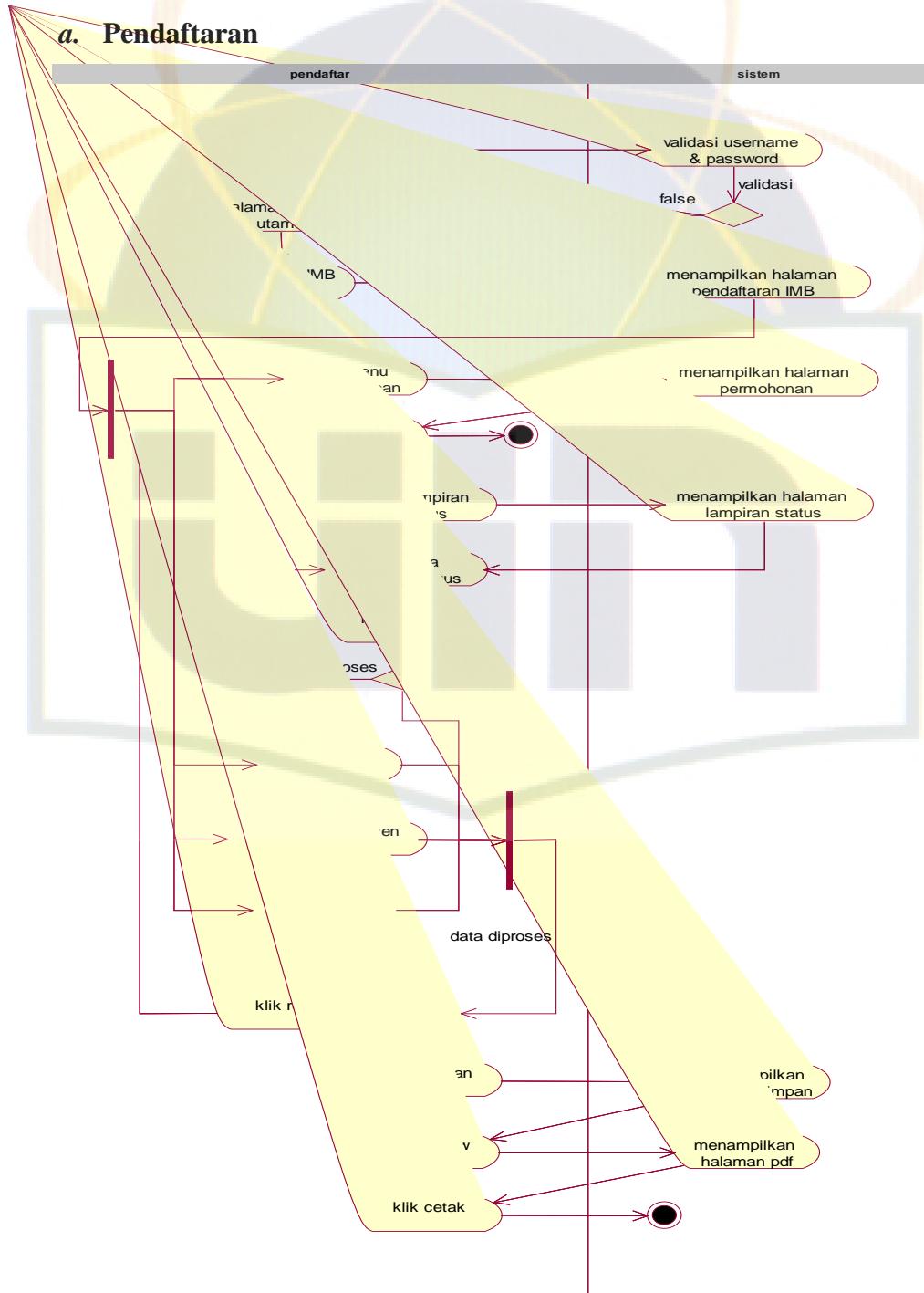
<b>Nama Use case</b>	Mencatat penyerahan
<b>Aktor yang terlibat</b>	Bagian Penomoran dan Penyerahan
<b>Trigger</b>	Aktor mencatat data pengambil IMB
<b>Pre condition</b>	Data pemohon di input.
<b>Action</b>	Mengisi form penyerahan dan data disimpan kedalam database
<b>Post condition</b>	Data Penyerahan tersimpan ke database

#### 4.3.5 Perancangan *Activity Diagram*

*Activity diagram* memodelkan alur kerja (*work flow*) sebuah urutan aktivitas pada suatu proses. Diagram ini sangat mirip dengan *flow chart* karena kita dapat memodelkan proses logika, proses bisnis dan laur kerja. Perbedaan utamanya adalah *flow chart* dibuat untuk menggambarkan alur kerja dari sebuah sistem, sedangkan *activity diagram* dibuat untuk menggambarkan aktivitas aktor.

Berikut akan digambarkan satu persatu *activity diagram* untuk masing-masing *use case*.

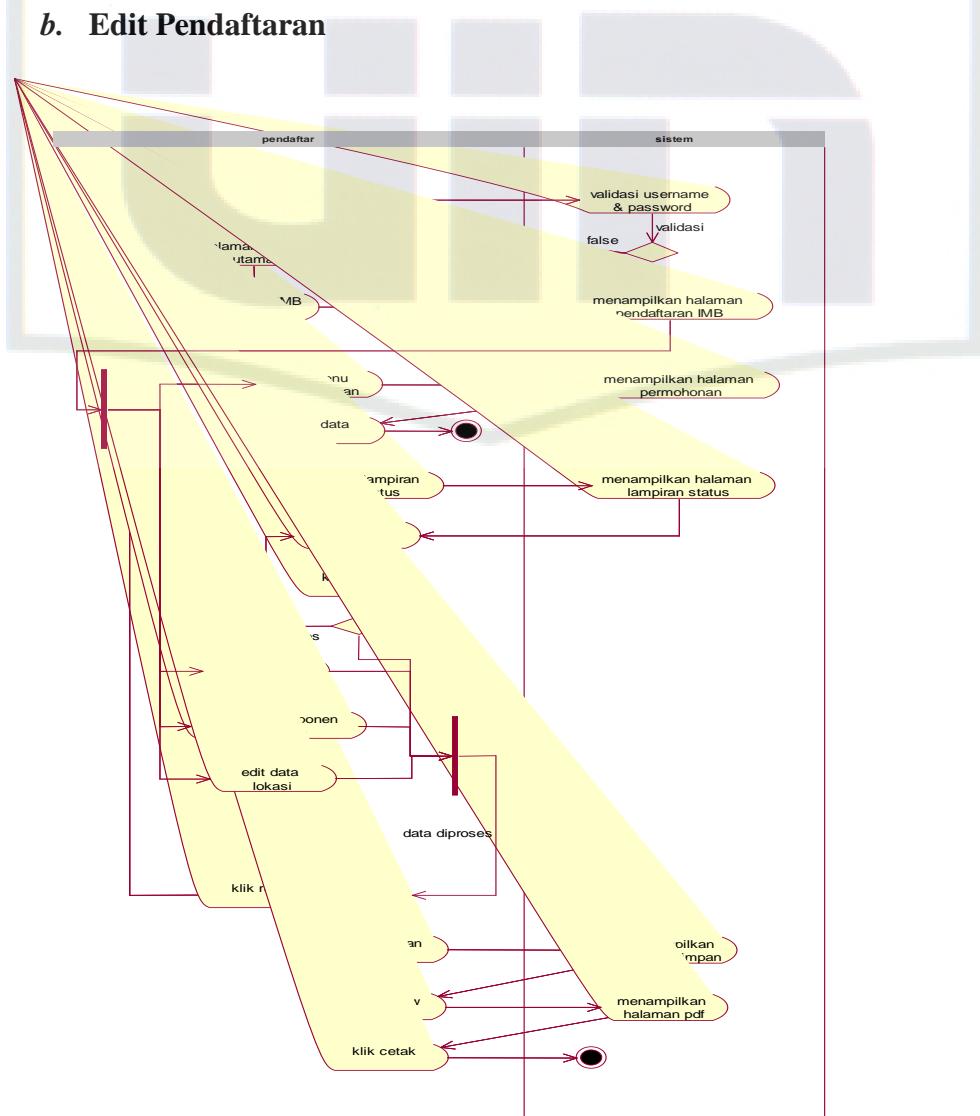
## 1. Pendaftaran



Gambar 4.7 Activity diagram pendaftaran

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu IMB dan sistem akan menampilkan form halaman pendaftaran untuk permohonan. Kemudian *user* memasukkan data pemohon serta kelengkapan berkas-berkas kebutuhan yang diperlukan dalam tahapan permohonan pengajuan IMB.

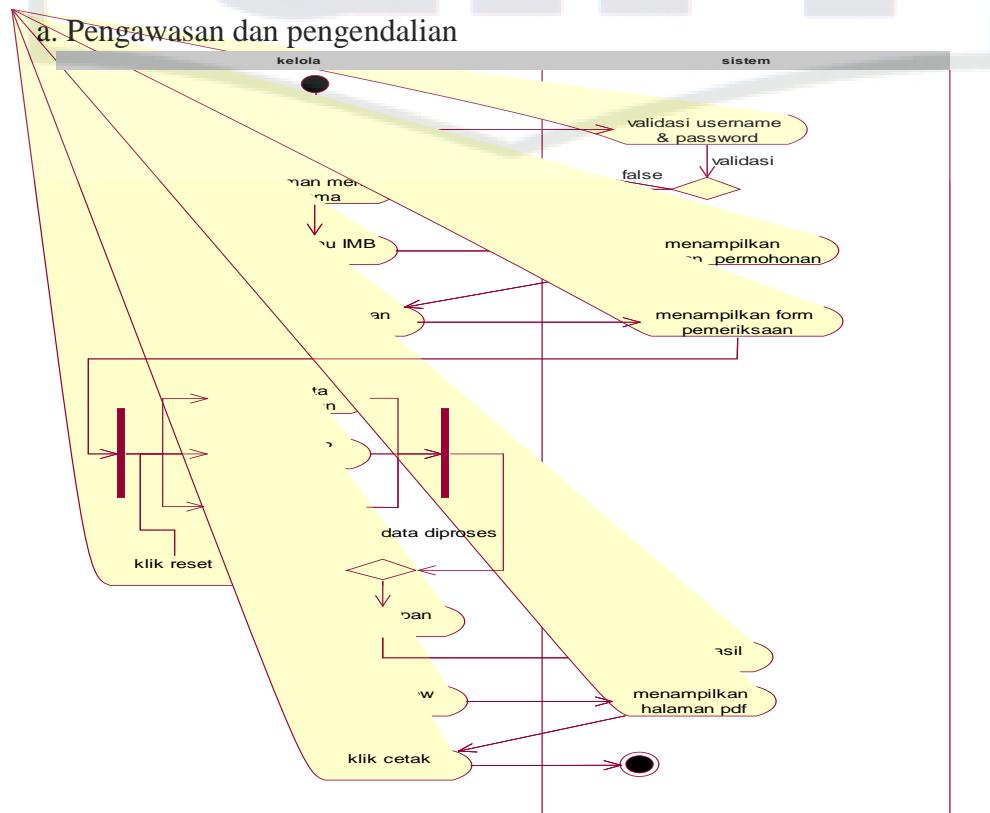
### b. Edit Pendaftaran



**Gambar 4.8** Activity diagram Edit Pendaftaran

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu daftar perizinan IMB dan sistem akan menampilkan form halaman data pemohon yang telah terdaftar sebelumnya, kemudian user memilih data permohonan yang ingin diedit dan mengklik edit. Sistem akan menampilkan form edit pendaftaran. Kemudian *user* dapat merubah data pemohon serta kelengkapan berkas-berkas kebutuhan yang ingin dirubah, kemudian user dapat menyimpan data pendaftaran dan mencetak kartu daftar kembali.

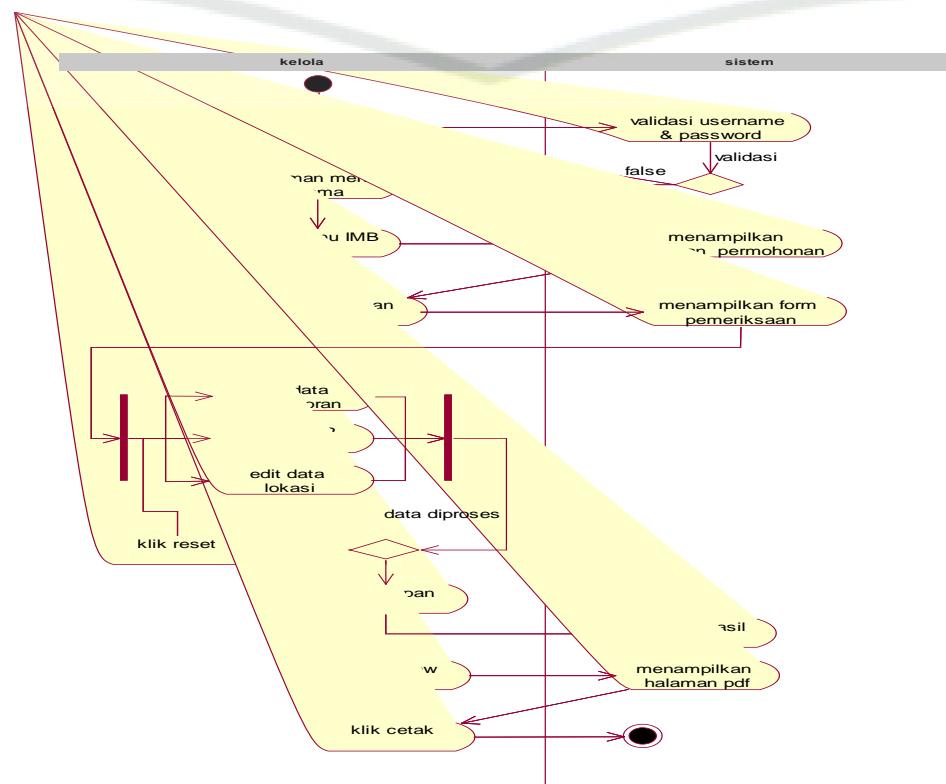
## 2. Pengawasan dan pengendalian



Gambar 4.9 *Activity diagram* pengawasan dan pengendalian

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu IMB dan sistem akan menampilkan form halaman kelola untuk pemeriksaan lapangan. Kemudian *user* memasukkan nomor pemeriksaan (BAPL) yang akan diperiksa tim lapangan serta kelengkapan berkas-berkas. Kemudian tim lapangan mengecek kebenaran berkas yang telah masuk. Tim lapangan kemudian melaporkan data-data yang telah diperiksa. Setelah berkas diperiksa dan alamat lokasi sudah jelas data dapat disimpan dan user dapat mencetak Berita Acara Hasil Pemeriksaan Lapangan.

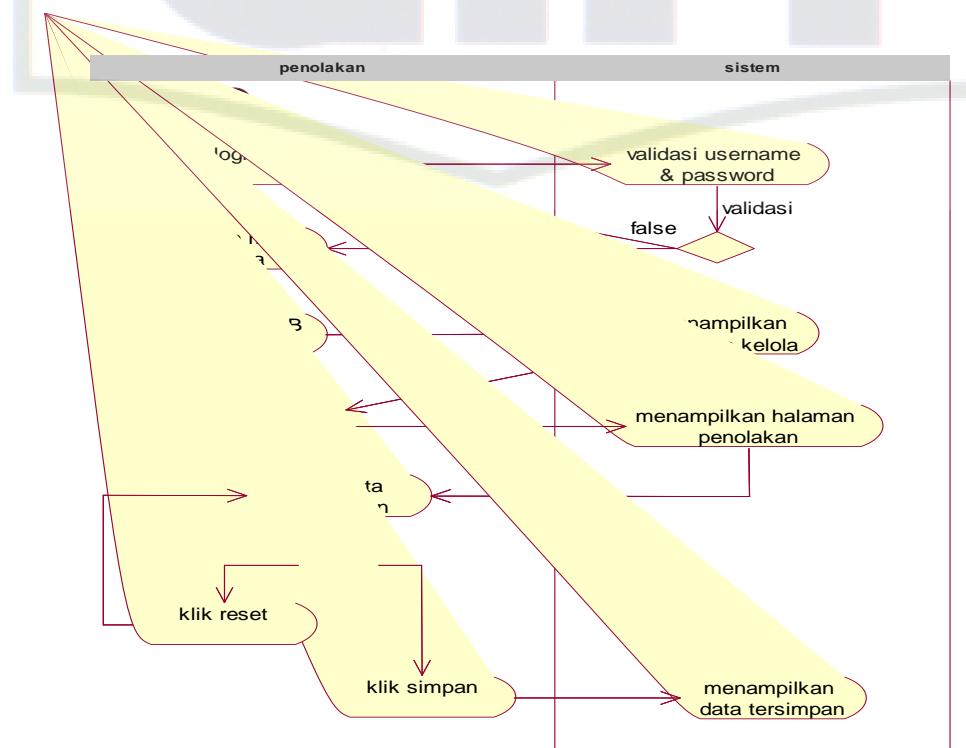
#### b. Edit Pengawasan dan pengendalian



**Gambar 4.10** Activity diagram edit pengawasan dan pengendalian

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu daftar perizinan IMB dan mengkil menu pemeriksaan, sistem akan menampilkan data yang ingin diedit. Kemudian user memilih data yang akan diedit. Sistem akan menampilkan form halaman edit untuk pemeriksaan data lapangan. Kemudian *user* Merubah data hasil pemeriksaan lapangan. Setelah berkas diperiksa dan telah melakukan perubahan user dapat kembali menyimpan dan mencetak Berita Acara Hasil Pemeriksaan Lapangan.

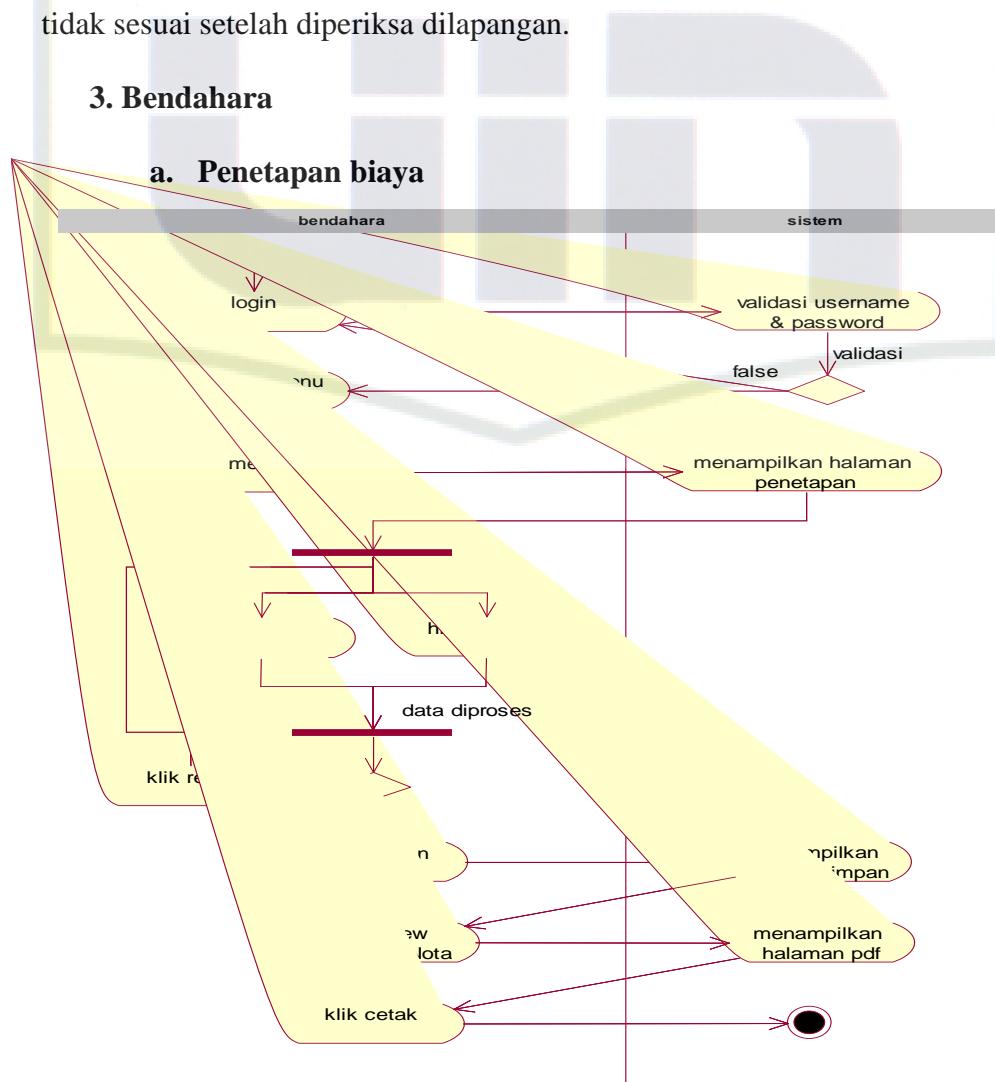
### c. Penolakan



**Gambar 4.11** *Activity diagram* penolakan

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu IMB dan sistem akan menampilkan form halaman kelola untuk pemeriksaan berkas. Aktivitas penolakan terjadi apabila berkas yang diajukan tidak disetujui untuk di lanjutkan ketahap selanjutnya. Dapat terjadi apabila berkas tidak lengkap, atau data yang di ajukan pemohon tidak sesuai setelah diperiksa dilapangan.

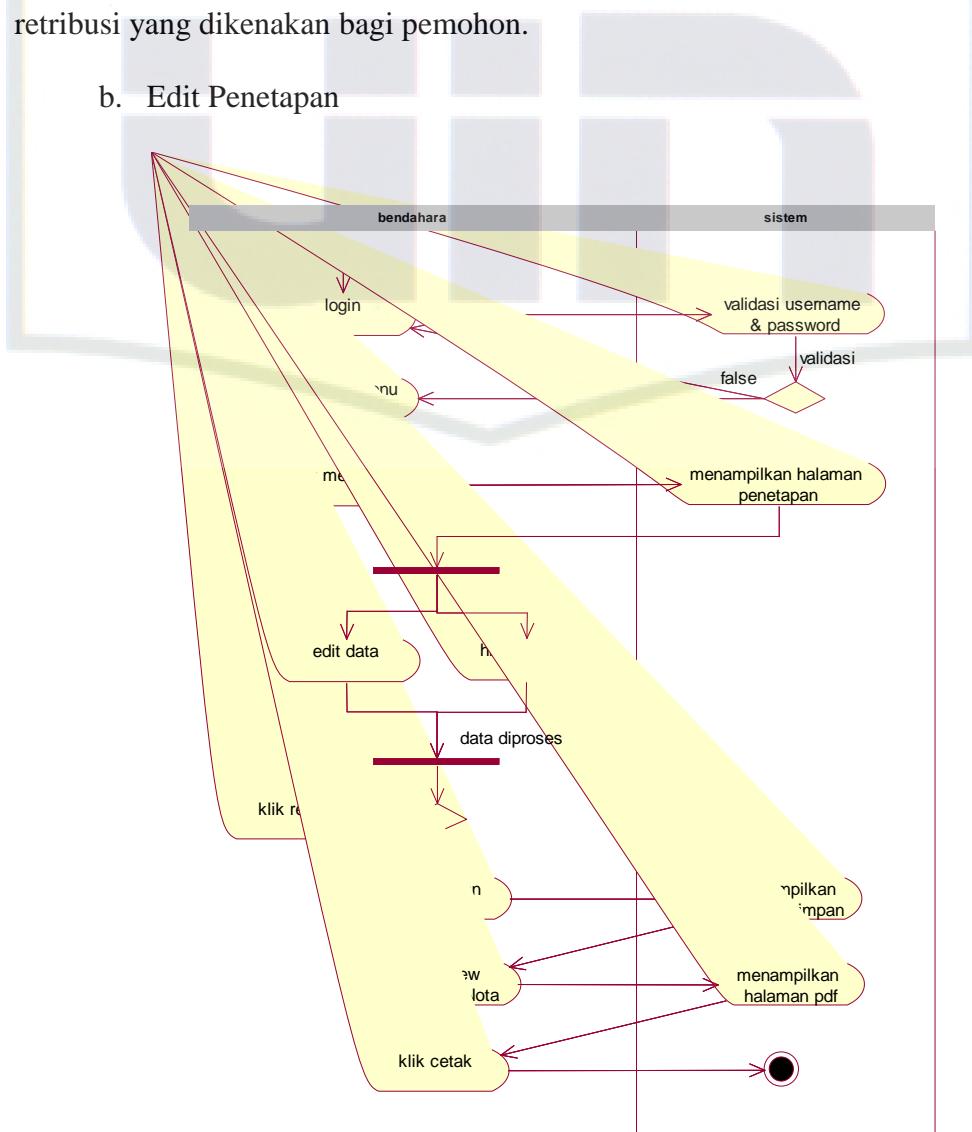
### 3. Bendahara



**Gambar 4.12** *Activity diagram* penetapan biaya

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik penetapan dan sistem akan menampilkan form halaman penetapan biaya retribusi. *User* sebagai bendahara memasukan data nomer pemohon dan kemudian menghitung menetukan nilai ketetapan retribusi pada masing-masing komponen-komponen bangunan untuk mendapatkan besaran biaya retribusi yang dikenakan bagi pemohon.

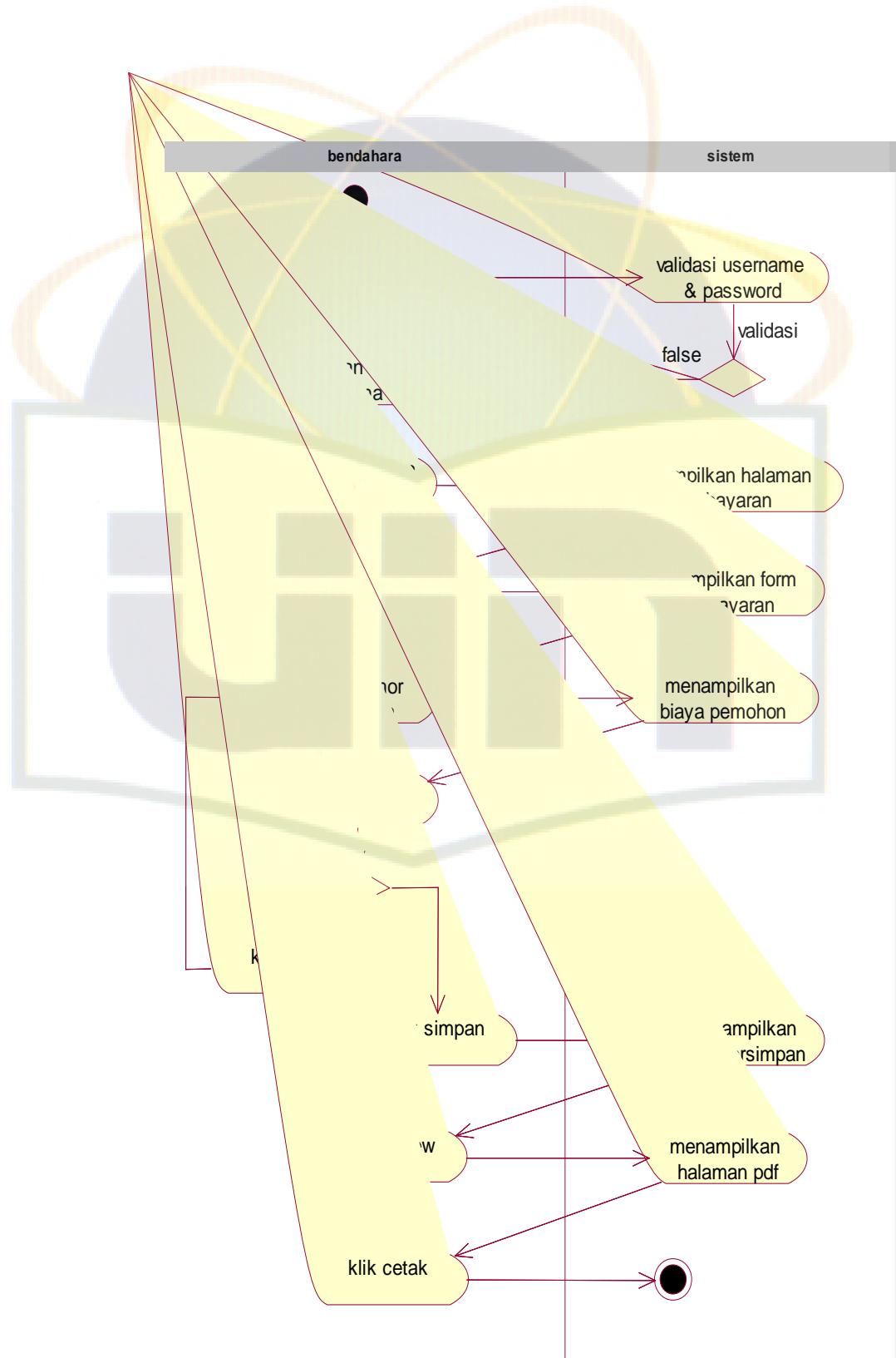
#### b. Edit Penetapan



**Gambar 4.13** Activity diagram edit penetapan biaya

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik daftar perizinan IMB dan mengklik menu penetapan dan sistem akan menampilkan data permohonan yang telah ditetapkan sebelumnya, kemudian user memilih data yang ingin diedit lalu menekan tombol edit. Sistem akan menampilkan Form halaman Edit penetapan biaya sesuai dengan data yanng dipilih. *User* melakukan perubahan dan dapat kembali menyimpan data yang telah dirubah tersebut.

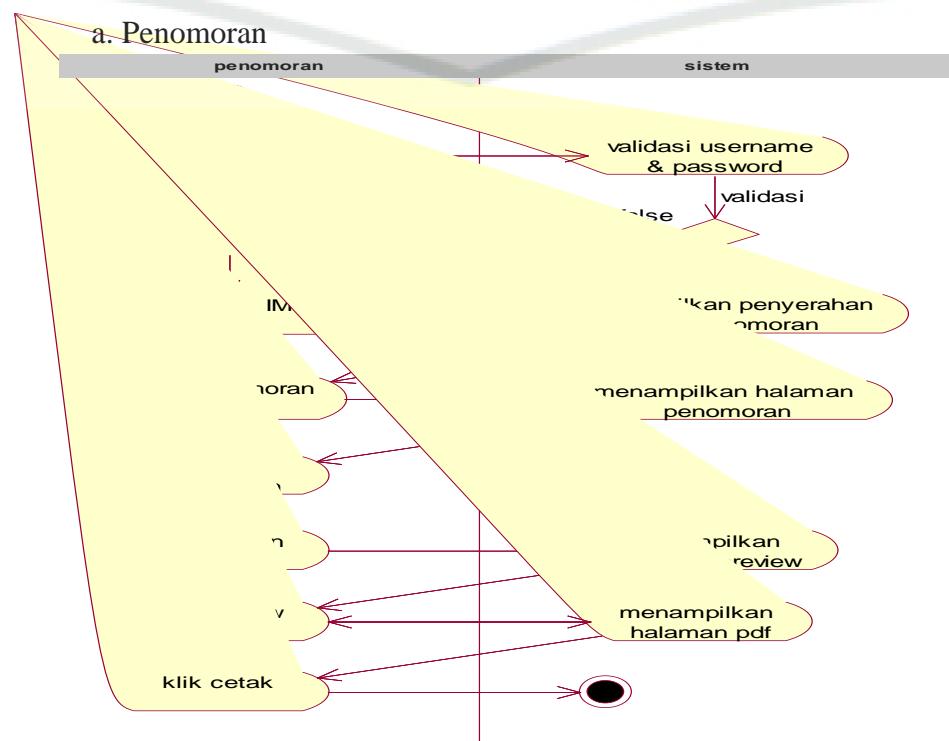
c. Pembayaran



**Gambar 4.14** Activity diagram pembayaran

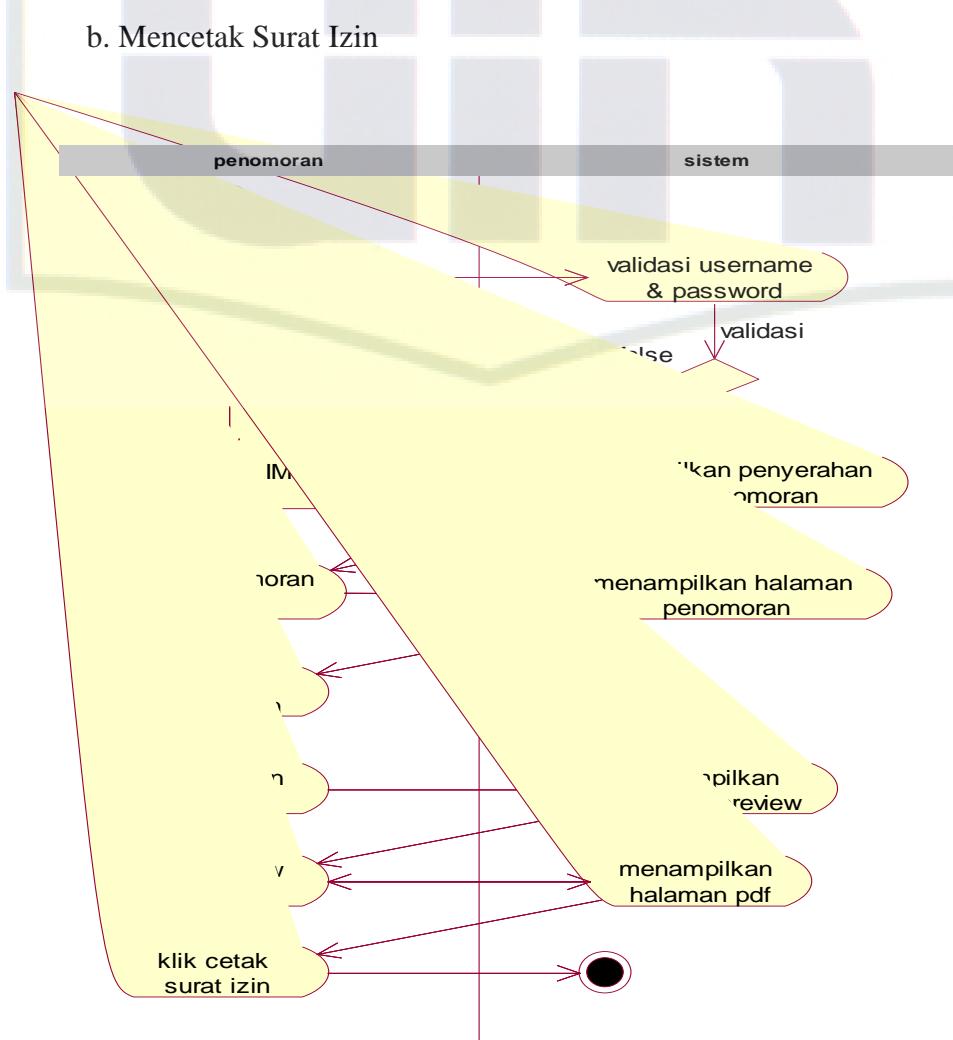
Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu pembayaran dan sistem akan menampilkan form halaman pembayaran untuk pembayaran retribusi. Pemohon diharuskan membayarkan sejumlah retribusi yang sesuai dengan Surat Ketetapan Retribusi Daerah yang di berikan BP2T Kota Tangerang Selatan. *User* sebagai bendahara memasukan data nomer pemohon, system akan menampilkan besar nilai retribusi sesuai dengan SKRD yang diterima pemohon, dan kemudian user mengisi informasi pembayaran lalu menyimpannya.

#### 4. Penomoran dan Penyerahan



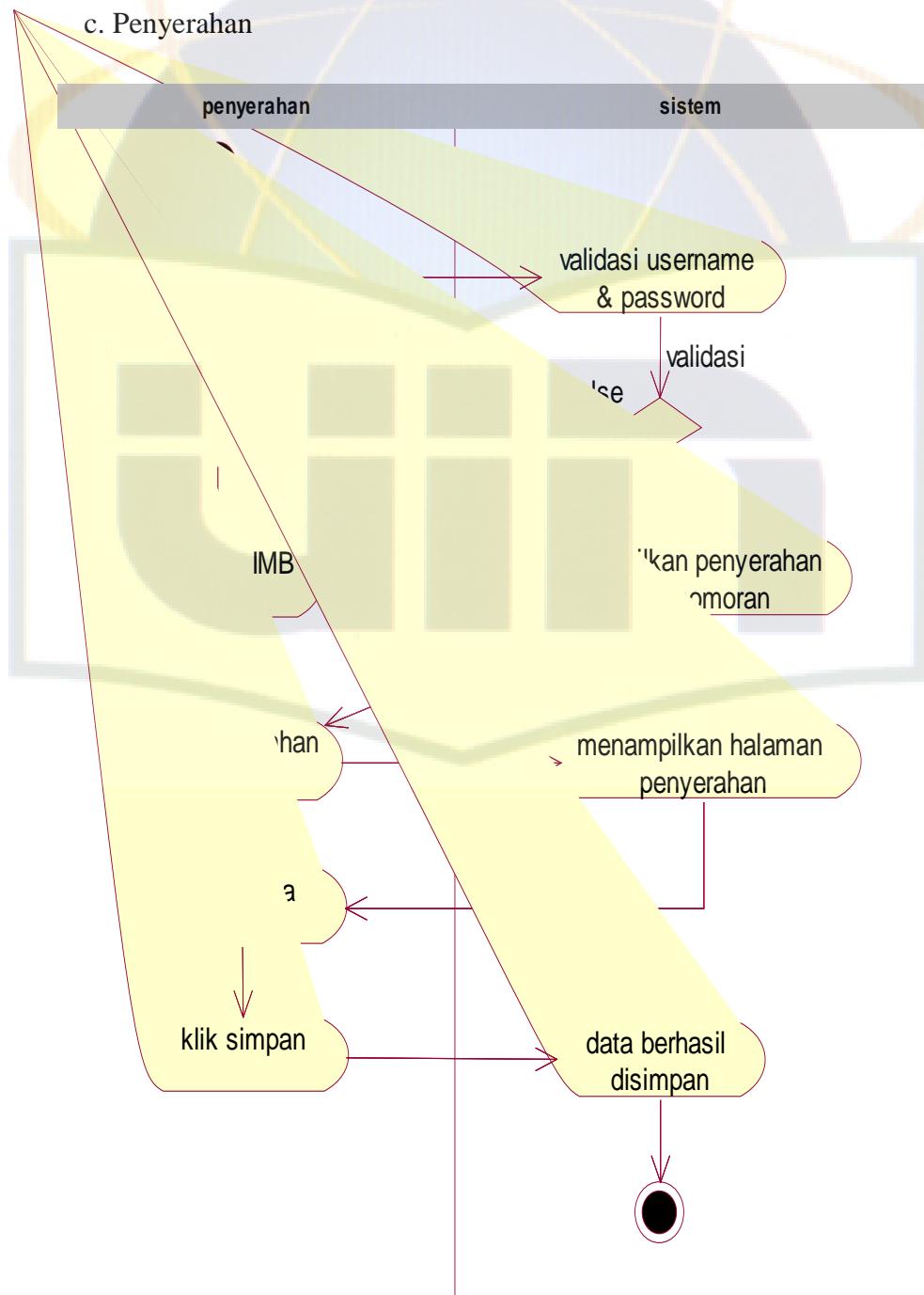
Gambar 4.15 Activity diagram Penomoran

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu IMB dan sistem akan menampilkan form halaman penomoran dan penyerahan untuk penyerahan berkas persetujuan IMB. Setelah pembayaran diselesaikan, maka pemohon mengambil penomoran IMB. *User* memasukan penomoran IMB dan kemudian mencetaknya.



**Gambar 4.16** Activity diagram Mencetak Surat Izin

Pada *activity diagram* ini *user* dapat mencetak surat ijin dengan menekan tombol Cetak Surat Izin setelah mengisi form penomoran.



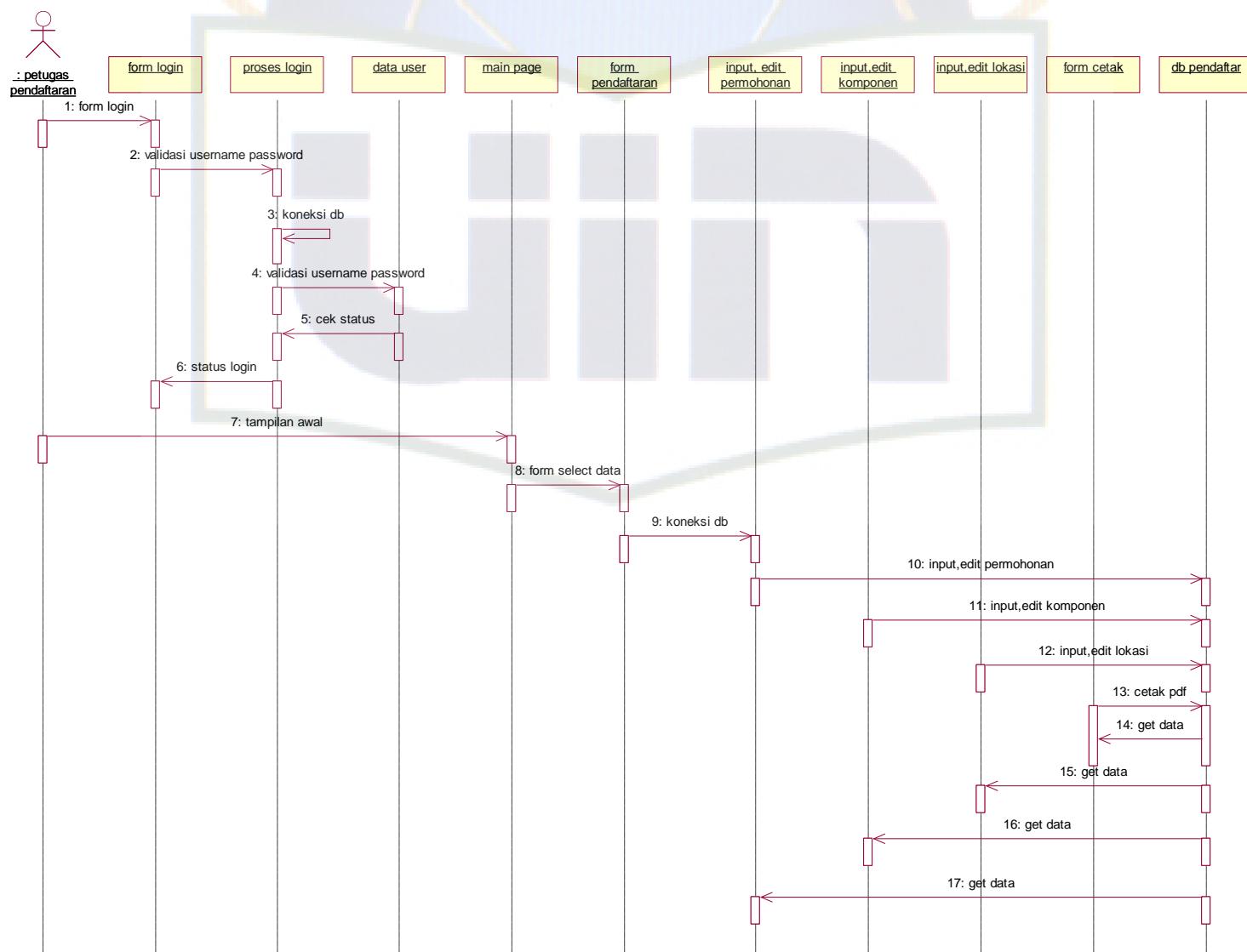
**Gambar 4.17** *Activity diagram* penyerahan

Pada *activity diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu IMB dan sistem akan menampilkan form halaman penomoran dan penyerahan untuk penyerahan berkas persetujuan IMB. Setelah berkas pemohon di nomori, kemudian *user* memasukan data pemohon yang mengambil perizinan ini.

#### 4.3.6 Perancangan Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Dibawah ini adalah sequence diagram untuk masing-masing modul.

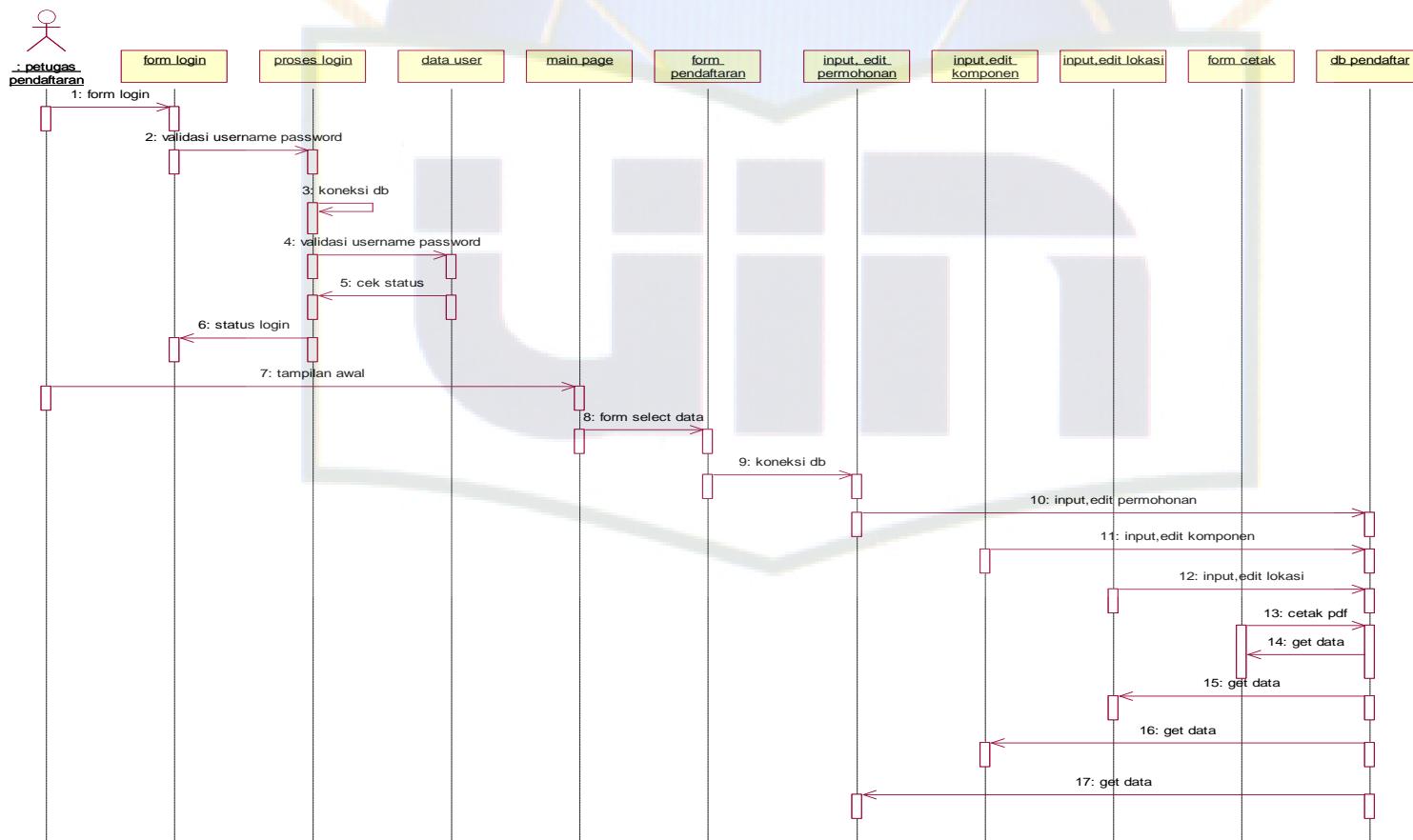
##### a. Pendaftaran



Gambar 4.18 Sequence diagram pendaftaran

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu aktivitas izin mendirikan bangunan dan sistem akan menampilkan form untuk input data pemohon, komponen dan lokasi. *User* juga dapat mencetak data yang telah disimpan.

b. Edit Pendaftaran

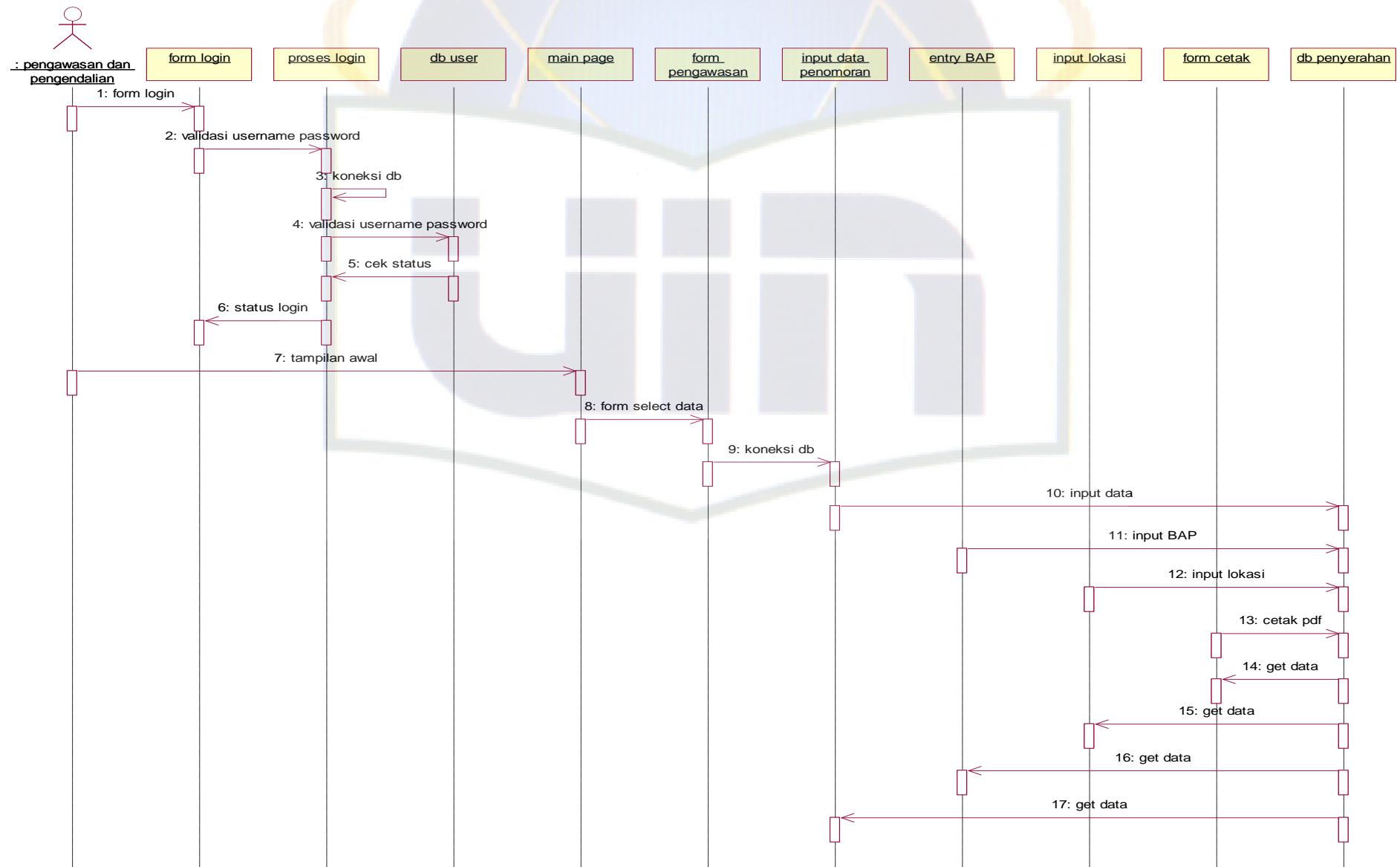


**Gambar 4.19** Sequence diagram edit pendaftaran

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu daftar permohonan izin mendirikan bangunan dan sistem akan menampilkan form data yang pernah disimpan sebelumnya untuk di edit, kemudian user memilih data pemohon yang ingin di edit dan sistem akan menampilkan data pemohon yang telah diinput sebelumnya untuk dirubah. *User* juga dapat mencetak kembali data setelah melakukan perubahan.

## 2. Pengawasan dan Pengelolaan

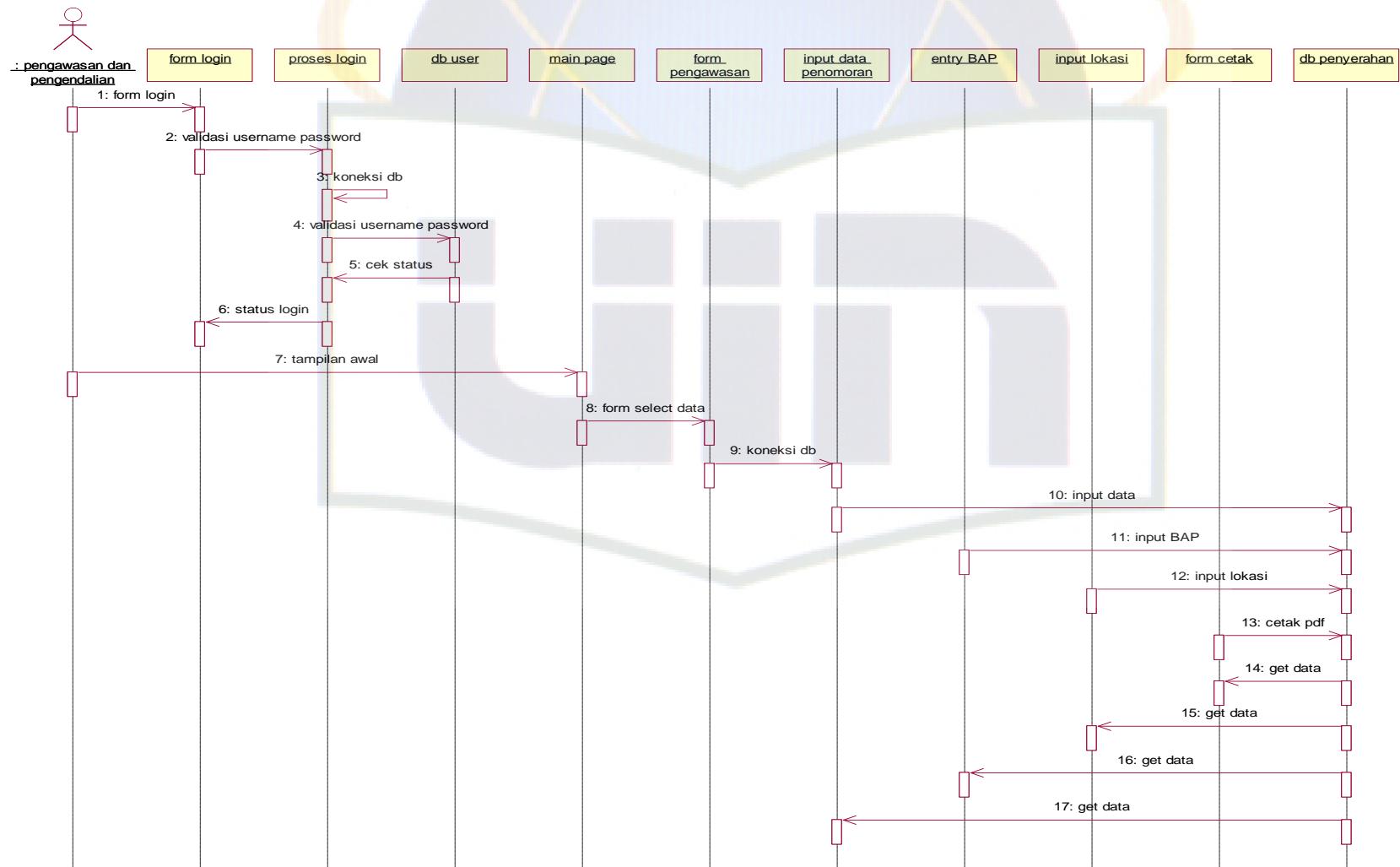
### a. Pengawasan dan Pengendalian



Gambar 4.20 Sequence diagram pengawasan dan pengendalian

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik form pemeriksaan dan sistem akan menampilkan form untuk input data penomoran, memsaukan data BAP dan input lokasi. *User* juga dapat mencetak data yang telah disimpan berupa Berita Acara Peninjauan Lapangan.

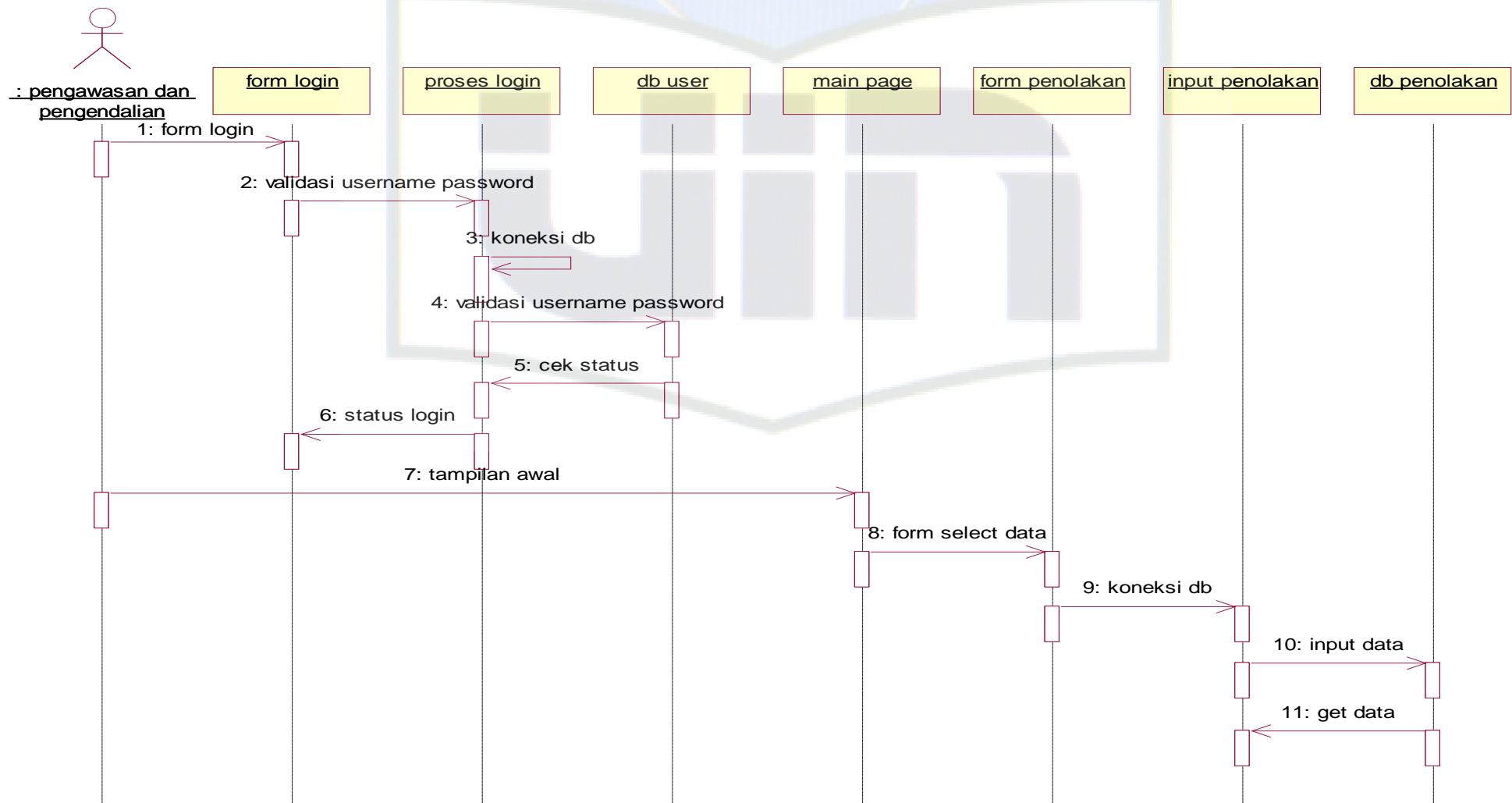
- b. Edit Pengawasan dan Pengendalian



Gambar 4.21 Sequence diagram pengawasan dan pengendalian

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik form daftar permohonan pemeriksaan dan sistem akan menampilkan data pemeriksaan yang sudah disimpan sebelumnya kemudian user memilih data pemohon yang akan diedit dan sistem akan menampilkan form data edit sesuai dengan data pemeriksaan yang dipilih. Setelah melakukan perubahan *user* juga dapat mencetak kembali Berita Acara Peninjauan Lapangan.

c. Penolakan

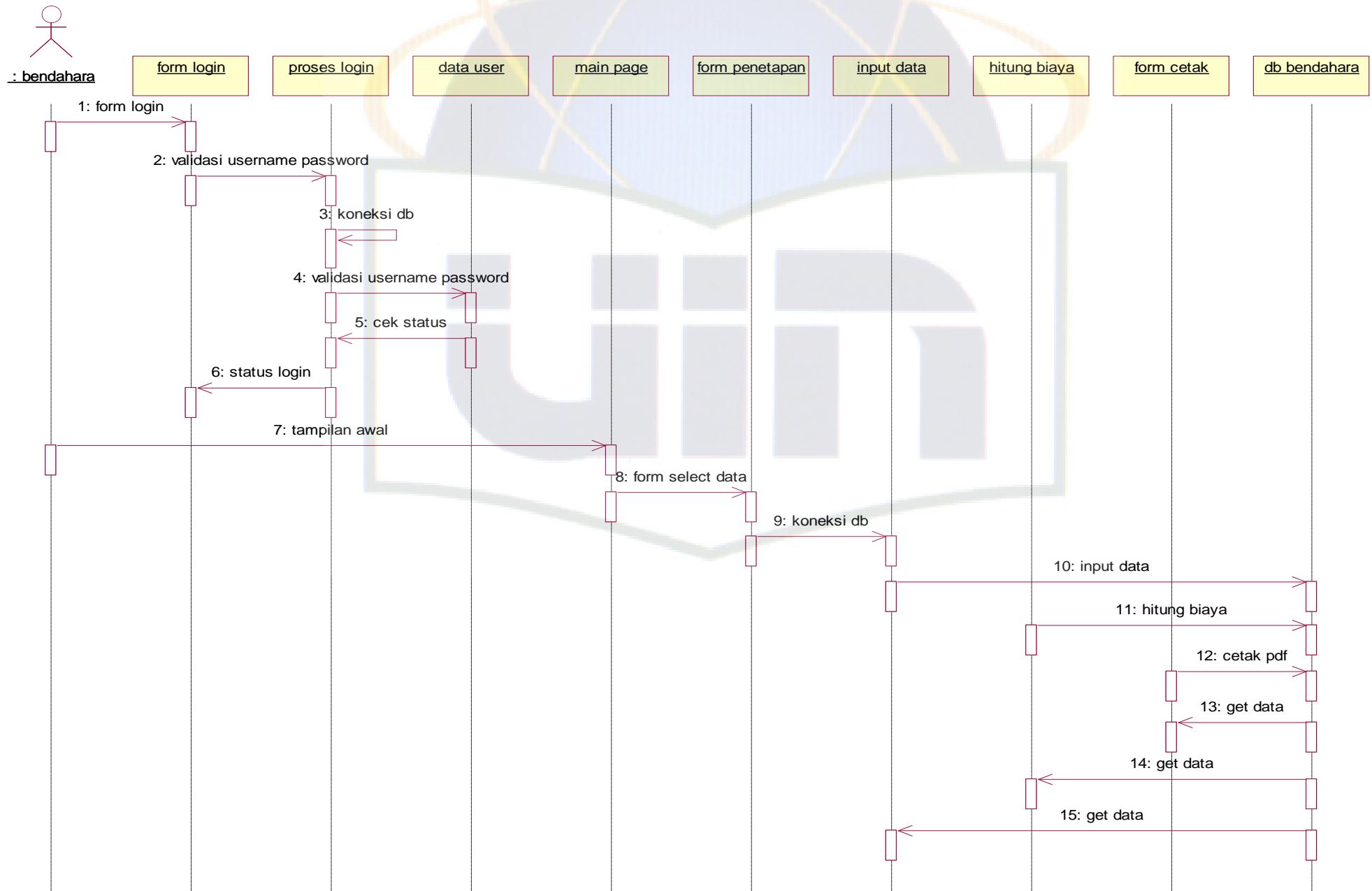


Gambar 4.22 Sequence diagram penolakan

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik form penolakan dan form untuk input data penolakan. *User* juga dapat mencetak data yang telah disimpan.

### **3. Penetapan Biaya dan Pembayaran**

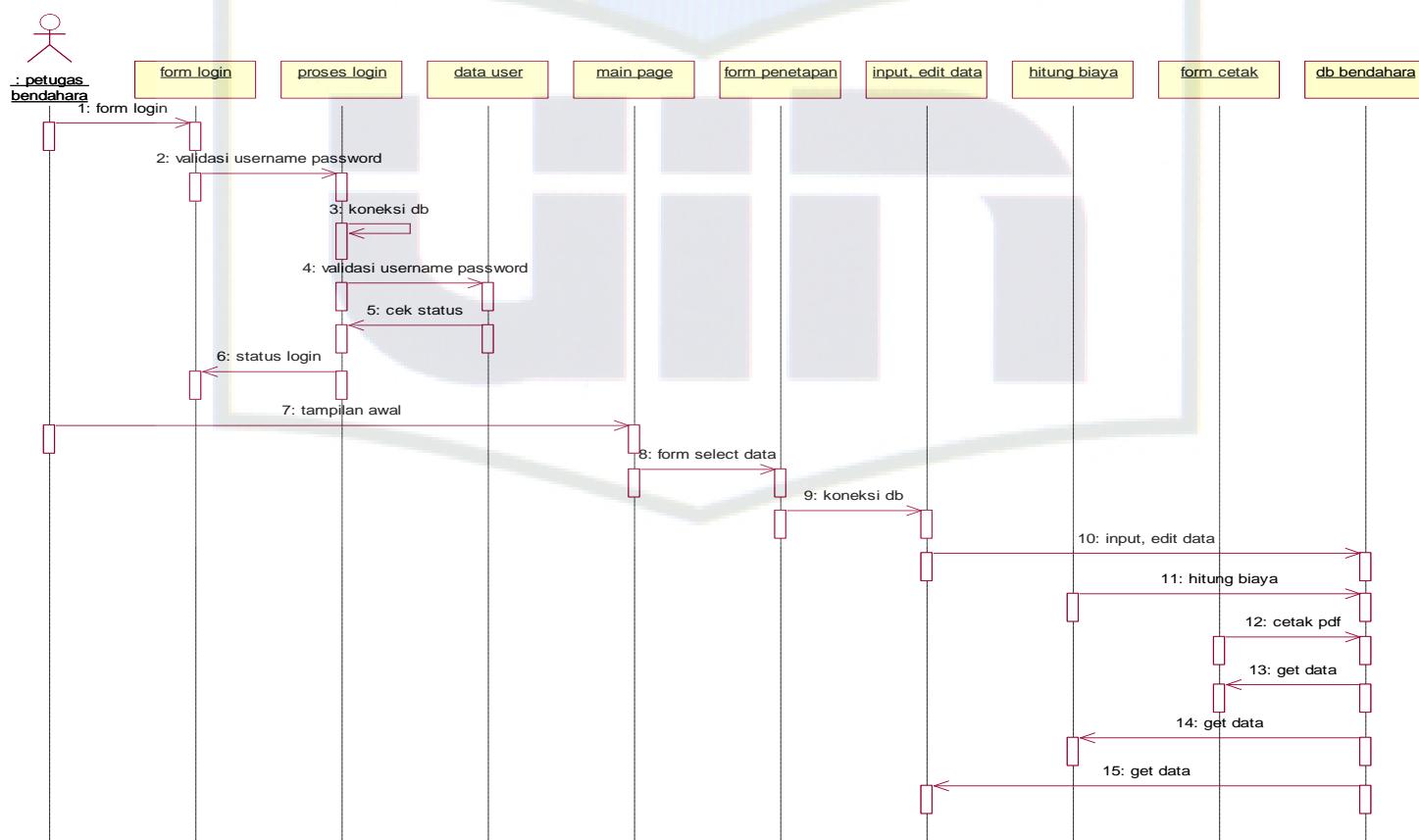
#### a. Penetapan Biaya



Gambar 4.23 Sequence diagram penetapan

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik form aktivitas penetapan dan sistem akan menampilkan form untuk input data dan penghitungan biaya retribusi. *User* juga dapat mencetak nota perhitungan dan SKRD yang telah disimpan.

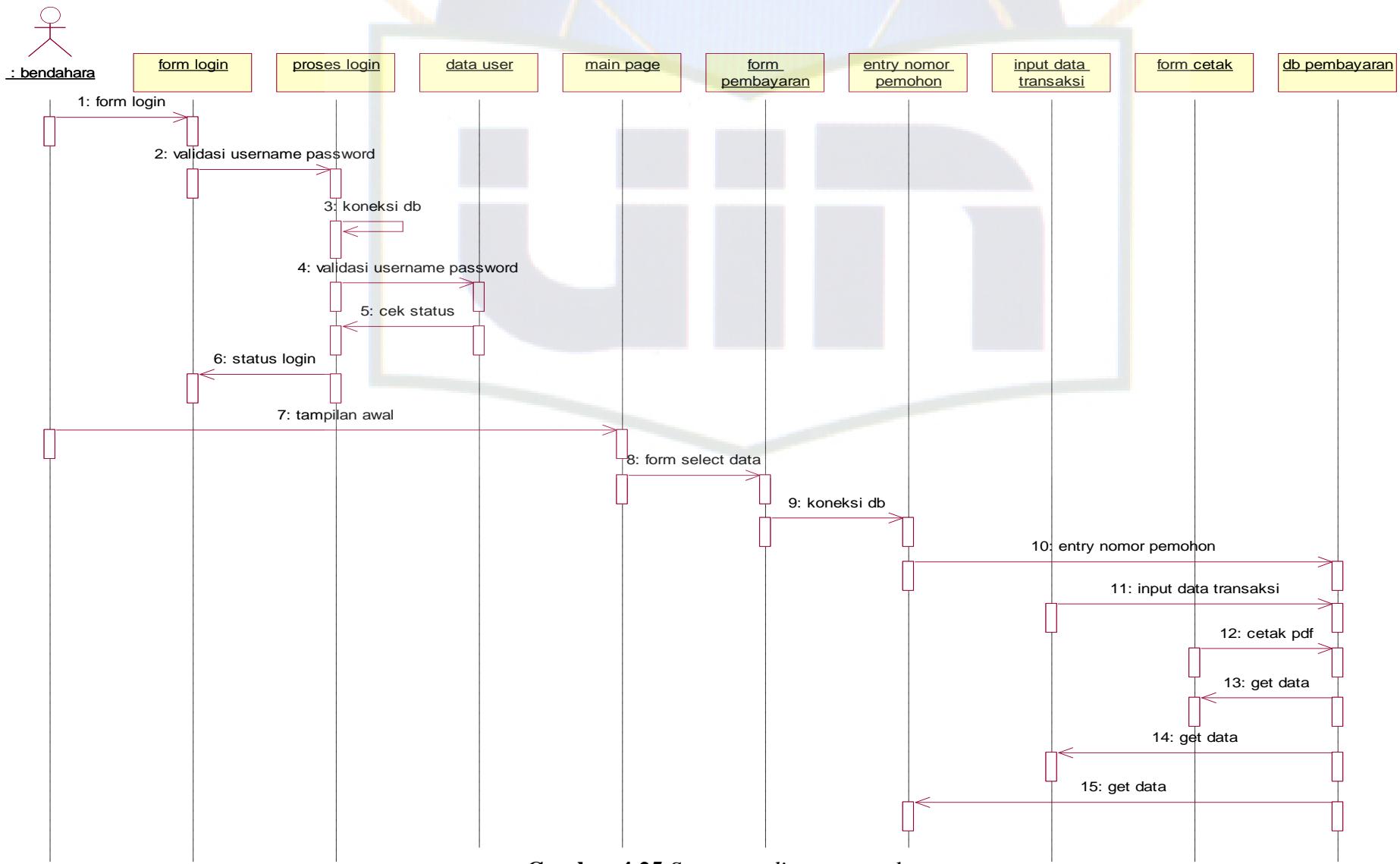
b. Edit Penetapan



Gambar 4.24 Sequence diagram edit penetapan

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik form daftar perizinan penetapan Izin Mendirikan Bangunan dan sistem akan menampilkan data penetapan biaya yang pernah ditetapkan sebelumnya, kemudian user memilih penetapan mana yang ingin diedit untuk melakukan perubahan baik data ataupun penghitungan biaya retribusi. *User* juga dapat kembali mencetak data nota perhitungan dan SKRD setelah melakukan perubahan.

c. Pembayaran

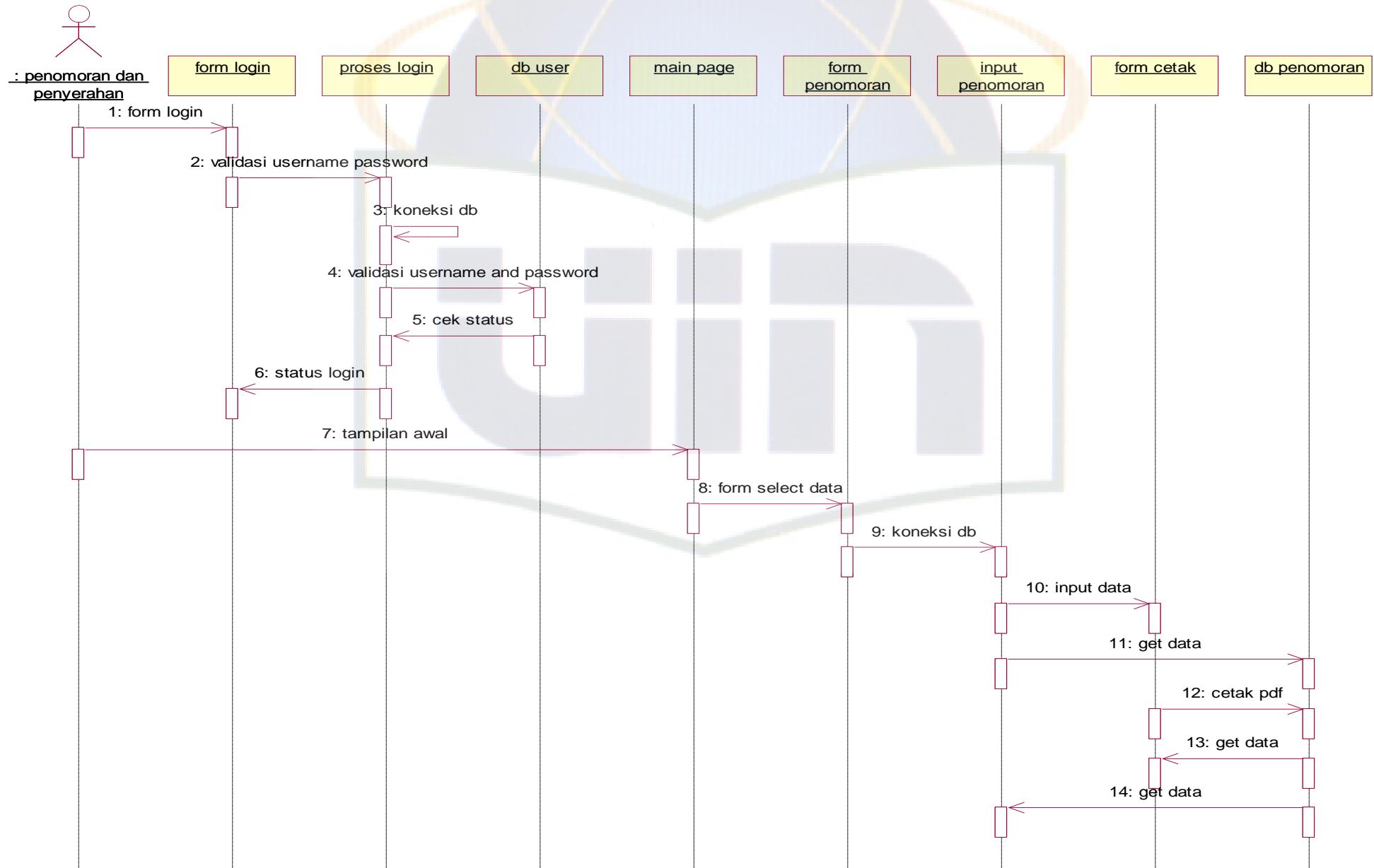


Gambar 4.25 Sequence diagram pembayaran

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik menu pembayaran dan sistem akan menampilkan form pendataan pembayaran beserta informasi-informasi pembayraran. *User* juga dapat mencetak bukti pembayaran data yang telah disimpan.

#### 4. Penomoran dan Penyerahan

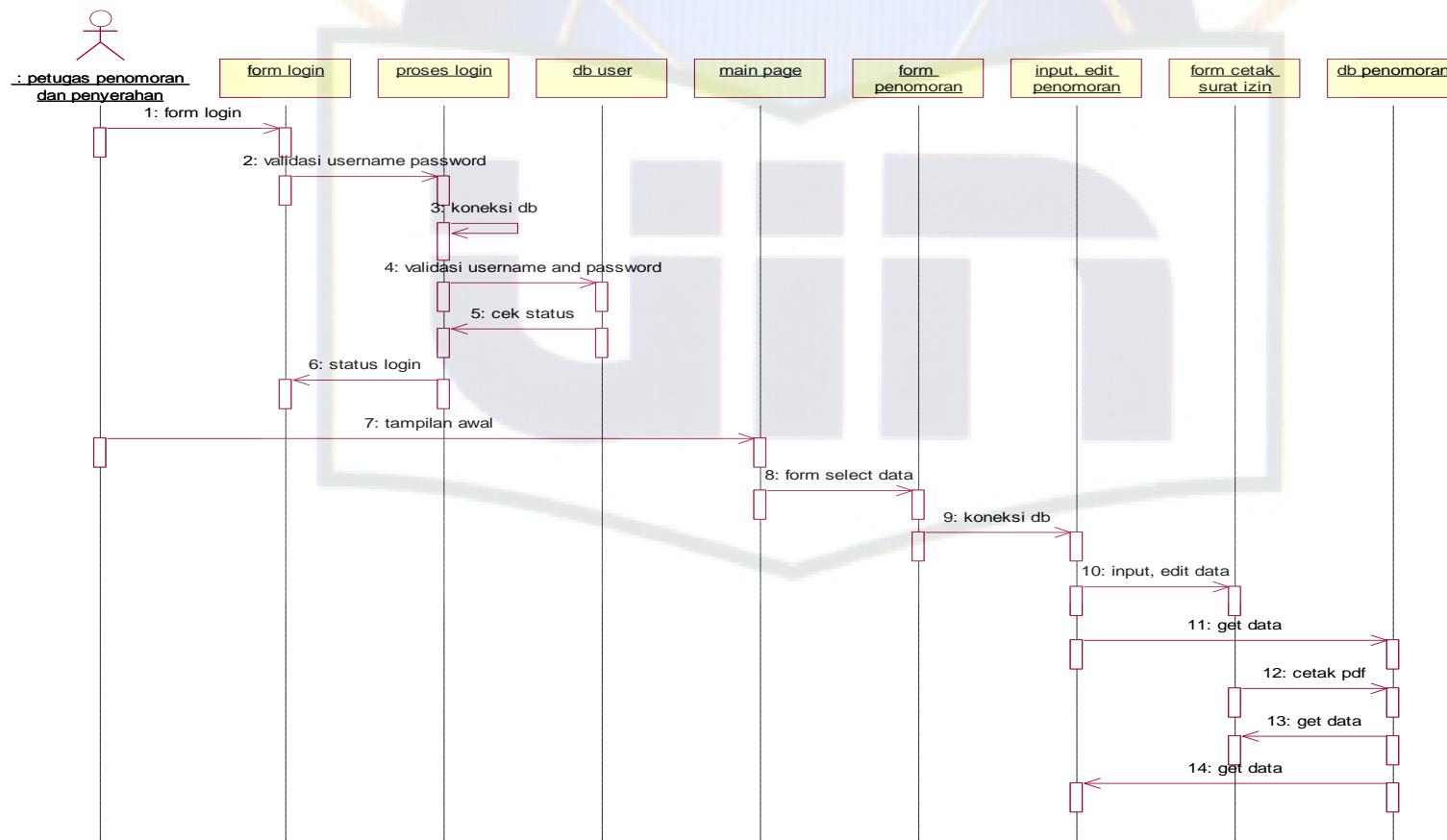
##### a. Penomoran



Gambar 4.26 Sequence diagram penomoran

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik form penomoran dan sistem akan menampilkan form untuk input data penomoran. *User* mengisi form penomoran kemudian menekan tombol simpan. Data penomoran akan tersimpan kedalam database.

b. Mencetak Surat Izin

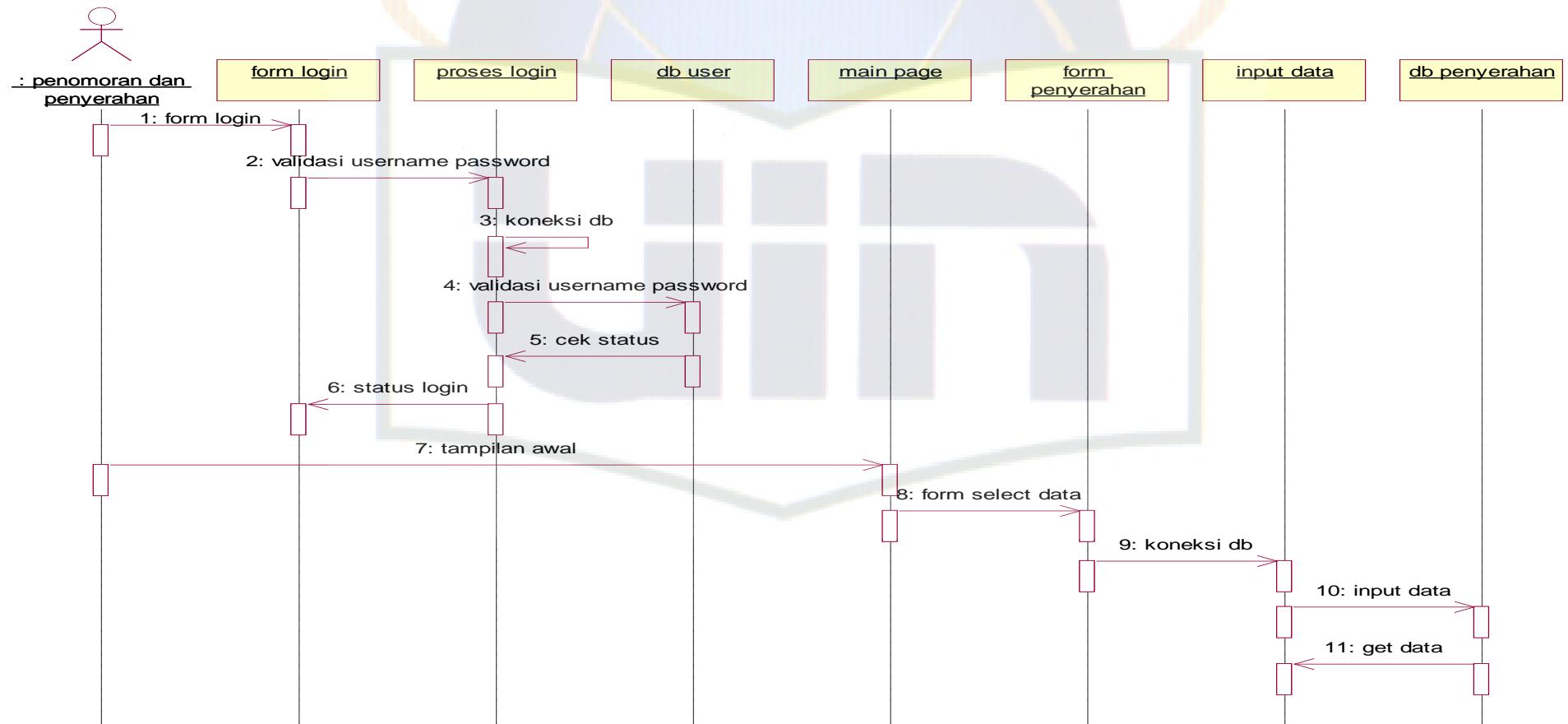


Gambar 4.27 Sequence Mencetak Surat Izin

Pada *Sequence diagram* ini *user* mencetak Surat Izin yang telah diberi nomor pada data yang telah disimpan dengan cara menekan tombol Cetak Surat Izin.

c. Penyerahan



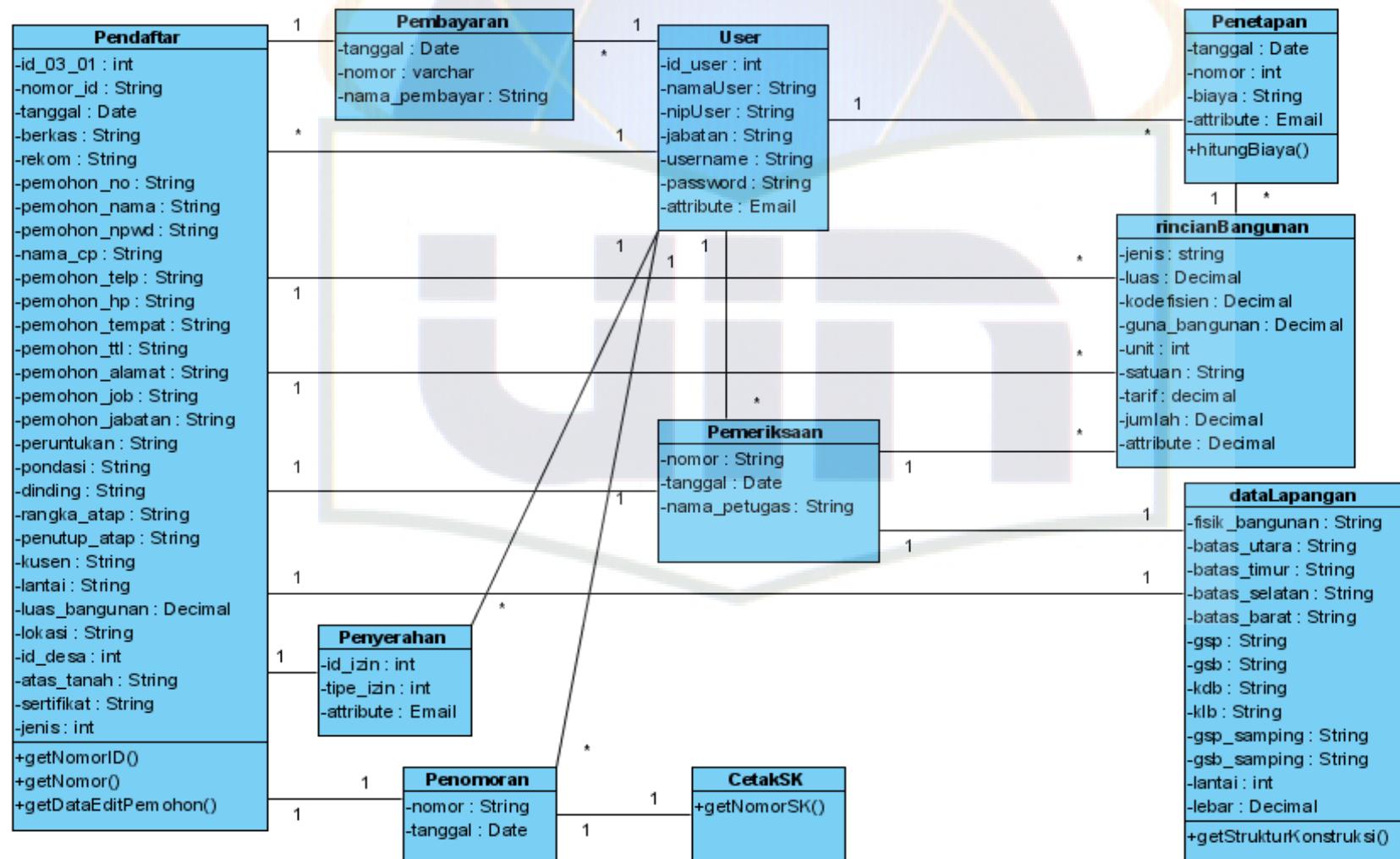


Gambar 4.28 Sequence diagram penyerahan

Pada *Sequence diagram* ini *user* melakukan *login*, kemudian sistem memvalidasi *username* dan *password* yang dimasukan. Jika *username* dan *password* sesuai dengan data yang terdapat didatabase, *user* mengklik form penyerahan dan sistem akan menampilkan form untuk input data.

#### 4.3.7 *Class Diagram*

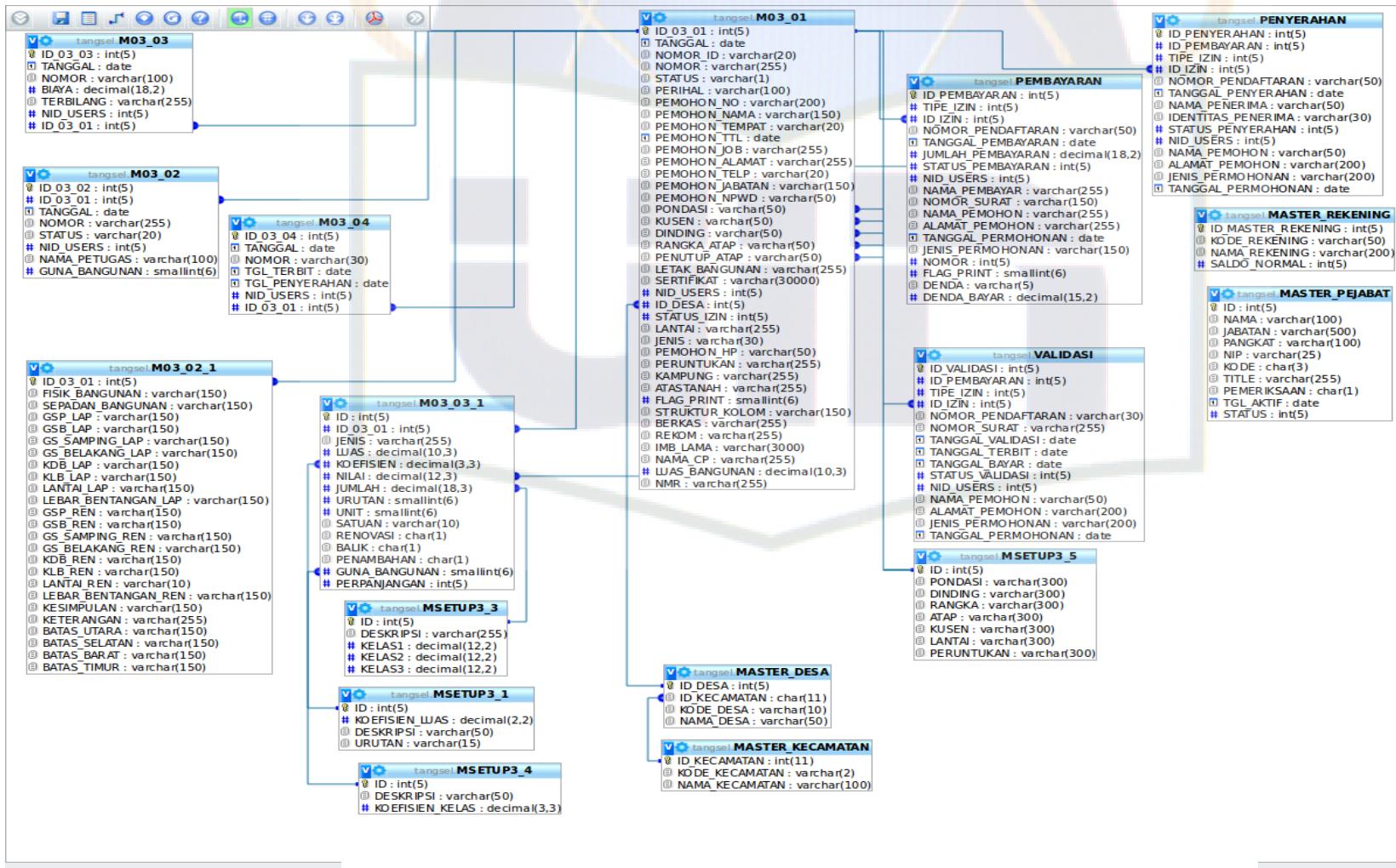
*Class diagram* ini digunakan untuk menggambarkan kumpulan dari class dan hubungannya. Diagram ini merupakan diagram yang paling umum ditemukan dalam pemodelan sistem berorientasi objek. Class menggambarkan keadaan suatu sistem, sekaligus layanan untuk memanipulasi keadaaan metode atau fungsi sehingga class memiliki tiga area pokok, yaitu: nama, atribut, dan metode. Selain itu setiap class yang ada dapat menjadi sebuah form saat pembuatan program. *Class diagram* sistem yang diusulkan dapat dilihat pada gambar berikut.



Gambar 4.29 Class Diagram untuk sistem yang diusulkan

#### 4.3.8 Rancangan Sistem Basis Data

Rancangan sistem basis data (*database*) ini merupakan rancangan sistem informasi berbasis *web* yang mengintegrasikan kumpulan data yang saling berhubungan satu dengan yang lainnya. *Database* ini didapat dari pemetaan (*mapping*) *class entity* yang telah digambarkan dalam *class diagram* sebelumnya. Pemetaan data-data yang berhubungan dalam sistem dijabarkan dalam bentuk tabel. Berikut adalah tabel-tabel *database* yang dihubungkan oleh *foreign key* untuk menunjukkan relasi atau disebut juga *Relational Database Management System* (RDBMS). Pada database ini terdapat nama-nama yang digunakan guna memudahkan para pengembang dalam membangun aplikasi dan diintegrasikan dengan izin-izin lainnya (SIUP, TDP, HO, dan lain-lain). IMB misalnya, merupakan aktivitas ke tiga dari banyaknya izin yang terdapat pada Badan Pelayanan Perizinan Terpadu Kota Tangerang Selatan. Maka untuk memudahkan, tabel izin tersebut disandikan dengan angka 3 sesuai dengan urutan izin. Untuk tabel pendaftaran IMB yaitu M03\_01, tabel pemeriksaan IMB M03\_02, Tabel Penetapan M03\_03. Setiap tabel yang berhubungan dengan tabel IMB disandikan dengan angka 3. Ada 18 tabel yang akan di gambarkan di bawah ini.



Gambar 4.30 Database Relational (Physical Database Schema)

## 1. Tabel Pendaftaran

Nama tabel : m03\_01

Primary key: id\_03\_01

Foreign key : nomor\_id

Tabel 4.35 Tabel m03\_01

Field	Type	NULL	Default	Description
ID_03_01	Integer	not null		No description [ <a href="#">edit</a> ]
TANGGAL	Date	not null		No description [ <a href="#">edit</a> ]
NOMOR_ID	Varchar(20)	not null		No description [ <a href="#">edit</a> ]
NOMOR	Varchar(255)	not null		No description [ <a href="#">edit</a> ]
STATUS	Varchar(1)	not null		No description [ <a href="#">edit</a> ]
PERIHAL	Varchar(100)			No description [ <a href="#">edit</a> ]
PEMOHON_NO	Varchar(200)	not null		No description [ <a href="#">edit</a> ]
PEMOHON_NAMA	Varchar(150)	not null		No description [ <a href="#">edit</a> ]
PEMOHON_TEMPAT	Varchar(20)			No description [ <a href="#">edit</a> ]
PEMOHON_TTL	Date			No description [ <a href="#">edit</a> ]
PEMOHON_JOB	Varchar(255)			No description [ <a href="#">edit</a> ]
PEMOHON_ALAMAT	Varchar(255)			No description [ <a href="#">edit</a> ]
PEMOHON_TELP	Varchar(20)			No description [ <a href="#">edit</a> ]
PEMOHON_JABATAN	Varchar(150)			No description [ <a href="#">edit</a> ]
PEMOHON_NPWD	Varchar(50)			No description [ <a href="#">edit</a> ]
PONDASI	Varchar(50)			No description [ <a href="#">edit</a> ]
KUSEN	Varchar(50)			No description [ <a href="#">edit</a> ]
DINDING	Varchar(50)			No description [ <a href="#">edit</a> ]
RANGKA_ATAP	Varchar(50)			No description [ <a href="#">edit</a> ]
PENUTUP_ATAP	Varchar(50)			No description [ <a href="#">edit</a> ]
LETAK_BANGUNAN	Varchar(255)			No description [ <a href="#">edit</a> ]
SERTIFIKAT	Varchar(30000)			No description [ <a href="#">edit</a> ]
NID_USERS	Integer	not null		No description [ <a href="#">edit</a> ]
ID_DESA	Integer			No description [ <a href="#">edit</a> ]
STATUSIZIN	Integer			No description [ <a href="#">edit</a> ]
LANTAI	Varchar(255)			No description [ <a href="#">edit</a> ]
JENIS	Varchar(30)	not null		No description [ <a href="#">edit</a> ]
PEMOHON_HP	Varchar(50)			No description [ <a href="#">edit</a> ]
PERUNTUKAN	Varchar(255)			No description [ <a href="#">edit</a> ]
KAMPUNG	Varchar(255)			No description [ <a href="#">edit</a> ]
ATASTANAH	Varchar(255)			No description [ <a href="#">edit</a> ]
FLAG_PRINT	Smallint	not null		No description [ <a href="#">edit</a> ]
STRUKTUR_KOLOM	Varchar(150)			No description [ <a href="#">edit</a> ]
BERKAS	Varchar(255)			No description [ <a href="#">edit</a> ]
REKOM	Varchar(255)			No description [ <a href="#">edit</a> ]
IMB_LAMA	Varchar(3000)			No description [ <a href="#">edit</a> ]
NAMA_CP	Varchar(255)			No description [ <a href="#">edit</a> ]
LUAS_BANGUNAN	Decimal(10,3)			No description [ <a href="#">edit</a> ]
NMR	Varchar(255)			No description [ <a href="#">edit</a> ]

## 2. Tabel Pemeriksaan

Nama tabel : m03\_02

Primary key: id\_03\_02

Uniq key: id\_03\_01

**Tabel 4.36** Tabel m03\_02

Field	Type	NULL	Default	Description
ID_03_02	Integer	not null		No description [ <a href="#">edit</a> ]
ID_03_01	Integer			No description [ <a href="#">edit</a> ]
TANGGAL	Date	not null		No description [ <a href="#">edit</a> ]
NOMOR	Varchar(255)	not null		No description [ <a href="#">edit</a> ]
STATUS	Varchar(20)	not null		No description [ <a href="#">edit</a> ]
NID_USERS	Integer			No description [ <a href="#">edit</a> ]
NAMA_PETUGAS	Varchar(100)			No description [ <a href="#">edit</a> ]
GUNA_BANGUNAN	Smallint			No description [ <a href="#">edit</a> ]

### Tabel Detail Pemeriksaan hasil pemeriksaan

Nama tabel : m03\_02 \_1

**Tabel 4.37 Tabel m03\_02\_1**

Field	Type	NULL	Default	Description
ID_03_01	Integer	not null		No description [ <a href="#">edit</a> ]
FISIK_BANGUNAN	Varchar(150)			No description [ <a href="#">edit</a> ]
SEPADAN_BANGUNAN	Varchar(150)			No description [ <a href="#">edit</a> ]
GSP_LAP	Varchar(150)			No description [ <a href="#">edit</a> ]
GSB_LAP	Varchar(150)			No description [ <a href="#">edit</a> ]
GS_SAMPING_LAP	Varchar(150)			No description [ <a href="#">edit</a> ]
GS_BELAKANG_LAP	Varchar(150)			No description [ <a href="#">edit</a> ]
KDB_LAP	Varchar(150)			No description [ <a href="#">edit</a> ]
KLB_LAP	Varchar(150)			No description [ <a href="#">edit</a> ]
LANTAI_LAP	Varchar(150)			No description [ <a href="#">edit</a> ]
LEBAR_BENTANGAN_LAP	Varchar(150)			No description [ <a href="#">edit</a> ]
GSP_REN	Varchar(150)			No description [ <a href="#">edit</a> ]
GSB_REN	Varchar(150)			No description [ <a href="#">edit</a> ]
GS_SAMPING_REN	Varchar(150)			No description [ <a href="#">edit</a> ]
GS_BELAKANG_REN	Varchar(150)			No description [ <a href="#">edit</a> ]
KDB_REN	Varchar(150)			No description [ <a href="#">edit</a> ]
KLB_REN	Varchar(150)			No description [ <a href="#">edit</a> ]
LANTAI_REN	Varchar(10)			No description [ <a href="#">edit</a> ]
LEBAR_BENTANGAN_REN	Varchar(150)			No description [ <a href="#">edit</a> ]
KESIMPULAN	Varchar(150)			No description [ <a href="#">edit</a> ]
KETERANGAN	Varchar(255)			No description [ <a href="#">edit</a> ]
BATAS_UTARA	Varchar(150)			No description [ <a href="#">edit</a> ]
BATAS_SELATAN	Varchar(150)			No description [ <a href="#">edit</a> ]

### 3. Tabel Penetapan Biaya

Nama tabel : m03\_03

Primary key: id\_03\_03

Foreign key : id\_03\_01

**Tabel 4.38** Tabel m03\_03

Field	Type	NULL	Default	Description
ID_03_03	Integer	not null		No description [edit]
TANGGAL	Date	not null		No description [edit]
NOMOR	Varchar(100)	not null		No description [edit]
BIAYA	Decimal(18,2)	not null		No description [edit]
TERBILANG	Varchar(255)			No description [edit]
NID_USERS	Integer	not null		No description [edit]
ID_03_01	Integer			No description [edit]

### 4. Tabel Ketetapan Biaya dan Koefisien Bangunan

Nama tabel : m03\_03\_1

Primary key: id\_03\_01

**Tabel 4.39** Tabel m03\_03\_1

Field	Type	NULL	Default	Description
ID	Integer	not null		No description [edit]
ID_03_01	Integer			No description [edit]
JENIS	Varchar(255)			No description [edit]
LUAS	Decimal(10,3)			No description [edit]
KOEFISIEN	Decimal(3,3)			No description [edit]
NILAI	Numeric(12,3)			No description [edit]
JUMLAH	Decimal(18,3)			No description [edit]
URUTAN	Smallint			No description [edit]
UNIT	Smallint			No description [edit]
SATUAN	Varchar(10)			No description [edit]
RENOVASI	Char(1)	not null		No description [edit]
BALIK	Char(1)	not null		No description [edit]
PENAMBAHAN	Char(1)	not null		No description [edit]
GUNA_BANGUNAN	Smallint			No description [edit]
PERPANJANGAN	Integer			No description [edit]

## 5. Tabel Penolakan

Nama tabel : m03\_04

Primary key: id\_03\_04

Foreign key : id\_03\_01,nid\_users

**Tabel 4.40** Tabel m03\_04

	<b>Field</b>	<b>Type</b>	<b>NULL</b>	<b>Default</b>	<b>Description</b>
	ID_03_04	Integer	<b>not null</b>		No description [edit]
	TANGGAL	Date	<b>not null</b>		No description [edit]
	NOMOR	Varchar(30)	<b>not null</b>		No description [edit]
	TGL_TERBIT	Date			No description [edit]
	TGL_PENYERAHAN	Date			No description [edit]
	NID_USERS	Integer	<b>not null</b>		No description [edit]
	ID_03_01	Integer	<b>not null</b>		No description [edit]

## 6. Tabel Pembayaran

Nama tabel : Pembayaran

Primary key: id\_pembayaran

Foreign key : nid\_user

**Tabel 4.41** Tabel Pembayaran

	Field	Type	NULL	Default	Description
1	ID PEMBAYARAN	Integer	not null		No description [ <a href="#">edit</a> ]
2	TIPE_IZIN	Integer	not null		No description [ <a href="#">edit</a> ]
3	ID_IZIN	Integer	not null		No description [ <a href="#">edit</a> ]
4	NOMOR_PENDAFTARAN	Varchar(50)	not null		No description [ <a href="#">edit</a> ]
5	TANGGAL PEMBAYARAN	Date			No description [ <a href="#">edit</a> ]
6	JUMLAH PEMBAYARAN	Decimal(18,2)	not null		No description [ <a href="#">edit</a> ]
7	STATUS PEMBAYARAN	Integer	not null		No description [ <a href="#">edit</a> ]
8	NID_USERS	Integer	not null		No description [ <a href="#">edit</a> ]
9	NAMA PEMBAYAR	Varchar(255)			No description [ <a href="#">edit</a> ]
10	NOMOR SURAT	Varchar(150)			No description [ <a href="#">edit</a> ]
11	NAMA PEMOHON	Varchar(255)			No description [ <a href="#">edit</a> ]
12	ALAMAT PEMOHON	Varchar(255)			No description [ <a href="#">edit</a> ]
13	TANGGAL PERMOHONAN	Date			No description [ <a href="#">edit</a> ]
14	JENIS PERMOHONAN	Varchar(150)			No description [ <a href="#">edit</a> ]
15	NOMOR	Integer			No description [ <a href="#">edit</a> ]
16	FLAG_PRINT	Smallint	not null		No description [ <a href="#">edit</a> ]
17	DENDA	Varchar(5)			No description [ <a href="#">edit</a> ]
18	DENDA_BAYAR	Decimal(15,2)			No description [ <a href="#">edit</a> ]

## 7. Tabel Penomoran

Nama tabel : Validasi

Primary key: id\_validasi

Foreign key : id\_pembayaran, nid\_users

**Tabel 4.42** Tabel Validasi

Field	Type	NULL	Default	Description
ID_VALIDASI	Integer	not null		No description [ <a href="#">edit</a> ]
ID_PEMBAYARAN	Integer			No description [ <a href="#">edit</a> ]
TIPE_IZIN	Integer	not null		No description [ <a href="#">edit</a> ]
ID_IZIN	Integer	not null		No description [ <a href="#">edit</a> ]
NOMOR_PENDAFTARAN	Varchar(30)	not null		No description [ <a href="#">edit</a> ]
NOMOR_SURAT	Varchar(255)			No description [ <a href="#">edit</a> ]
TANGGAL_VALIDASI	Date			No description [ <a href="#">edit</a> ]
TANGGAL_TERBIT	Date			No description [ <a href="#">edit</a> ]
TANGGAL_BAYAR	Date			No description [ <a href="#">edit</a> ]
STATUS_VALIDASI	Integer	not null		No description [ <a href="#">edit</a> ]
NID_USERS	Integer			No description [ <a href="#">edit</a> ]
NAMA PEMOHON	Varchar(50)			No description [ <a href="#">edit</a> ]
ALAMAT PEMOHON	Varchar(200)			No description [ <a href="#">edit</a> ]
JENIS_PERMOHONAN	Varchar(200)			No description [ <a href="#">edit</a> ]
TANGGAL_PERMOHONAN	Date			No description [ <a href="#">edit</a> ]

## 8. Tabel Penyerahan

Nama tabel : penyerahan

Primary key: id\_penyerahan

Foreign key : id\_pembayaran, nid\_users

**Tabel 4.43** Tabel penyerahan

Field	Type	NULL	Default	Description
ID_PENYERAHAN	Integer	not null		No description [ <a href="#">edit</a> ]
ID_PEMBAYARAN	Integer			No description [ <a href="#">edit</a> ]
TIPE_IZIN	Integer	not null		No description [ <a href="#">edit</a> ]
ID_IZIN	Integer	not null		No description [ <a href="#">edit</a> ]
NOMOR_PENDAFTARAN	Varchar(50)	not null		No description [ <a href="#">edit</a> ]
TANGGAL_PENYERAHAN	Date			No description [ <a href="#">edit</a> ]
NAMA_PENERIMA	Varchar(50)			No description [ <a href="#">edit</a> ]
IDENTITAS_PENERIMA	Varchar(30)			No description [ <a href="#">edit</a> ]
STATUS_PENYERAHAN	Integer	not null		No description [ <a href="#">edit</a> ]
NID_USERS	Integer			No description [ <a href="#">edit</a> ]
NAMA PEMOHON	Varchar(50)			No description [ <a href="#">edit</a> ]
ALAMAT PEMOHON	Varchar(200)			No description [ <a href="#">edit</a> ]
JENIS_PERMOHONAN	Varchar(200)			No description [ <a href="#">edit</a> ]
TANGGAL_PERMOHONAN	Date			No description [ <a href="#">edit</a> ]

## 9. Tabel Setup Koefisien Luas Bangunan

Nama tabel : msetup3\_1

Primary key: id

**Tabel 4.44** Tabel msetup3\_1

Field	Type	NULL	Default	Description
ID	Integer	not null		No description [ <a href="#">edit</a> ]
KOEFISIEN_LUAS	Decimal(2,2)			No description [ <a href="#">edit</a> ]
DESKRIPSI	Varchar(50)			No description [ <a href="#">edit</a> ]
URUTAN	Varchar(15)			No description [ <a href="#">edit</a> ]

## 10. Tabel Jenis Objek Bangunan

Nama tabel : msetup3\_2

Primary key: id

**Tabel 4.45** Tabel msetup3\_2

	Field	Type	NULL	Default	Description
	ID	Integer	<b>not null</b>		No description [ <a href="#">edit</a> ]
	JENIS_OBJEK	Varchar(50)			No description [ <a href="#">edit</a> ]
	SATUAN	Varchar(10)			No description [ <a href="#">edit</a> ]
	NILAI	Decimal(12,3)			No description [ <a href="#">edit</a> ]
	NAMA_SATUAN	Varchar(25)			No description [ <a href="#">edit</a> ]
	FLAG_BANGUNAN	Char(1)			No description [ <a href="#">edit</a> ]
	FLAG_TERAS	Char(1)			No description [ <a href="#">edit</a> ]

## 11. Tabel Jenis Komponen Bangunan

Nama tabel : msetup3\_3

Primary key: id\_03\_03

Foreign key : nid\_users,id\_03\_01

**Tabel 4.46** Tabel msetup3\_3

	Field	Type	NULL	Default	Description
	ID	Integer	<b>not null</b>		No description [ <a href="#">edit</a> ]
	DESKRIPSI	Varchar(255)			No description [ <a href="#">edit</a> ]
	KELAS1	Decimal(12,2)			No description [ <a href="#">edit</a> ]
	KELAS2	Decimal(12,2)			No description [ <a href="#">edit</a> ]
	KELAS3	Decimal(12,2)			No description [ <a href="#">edit</a> ]

## 12. Tabel Peruntukan Bangunan

Nama tabel : msetup3\_4

Primary key: id

**Tabel 4.47** Tabel msetup3\_4

	Field	Type	NULL	Default	Description
	ID	Integer	<b>not null</b>		No description [ <a href="#">edit</a> ]
	DESKRIPSI	Varchar(50)			No description [ <a href="#">edit</a> ]
	KOEFISIEN_KELAS	Decimal(3,3)			No description [ <a href="#">edit</a> ]

## 13. Tabel Pondasi Bangunan

Nama tabel : msetup3\_5

Primary key: id

**Tabel 4.48** Tabel msetup3\_5

	Field	Type	NULL	Default	Description
	ID	Integer	<b>not null</b>		No description [ <a href="#">edit</a> ]
	PONDASI	Varchar(300)			No description [ <a href="#">edit</a> ]
	DINDING	Varchar(300)			No description [ <a href="#">edit</a> ]
	RANGKA	Varchar(300)			No description [ <a href="#">edit</a> ]
	ATAP	Varchar(300)			No description [ <a href="#">edit</a> ]
	KUSEN	Varchar(300)			No description [ <a href="#">edit</a> ]
	LANTAI	Varchar(300)			No description [ <a href="#">edit</a> ]
	PERUNTUKAN	Varchar(300)			No description [ <a href="#">edit</a> ]

#### 14. Tabel Master Pejabat

Nama tabel : master\_pejabat

Primary key: id

**Tabel 4.49** Tabel master\_pejabat

	Field	Type	NULL	Default	Description
	ID	Integer	<b>not null</b>		No description [ <a href="#">edit</a> ]
	NAMA	Varchar(100)			No description [ <a href="#">edit</a> ]
	JABATAN	Varchar(500)			No description [ <a href="#">edit</a> ]
	PANGKAT	Varchar(100)			No description [ <a href="#">edit</a> ]
	NIP	Varchar(25)			No description [ <a href="#">edit</a> ]
	KODE	Char(3)	<b>not null</b>		No description [ <a href="#">edit</a> ]
	TITLE	Varchar(255)			No description [ <a href="#">edit</a> ]
	PEMERIKSAAN	Char(1)			No description [ <a href="#">edit</a> ]
	TGL_AKTIF	Date			No description [ <a href="#">edit</a> ]
	STATUS	Integer			No description [ <a href="#">edit</a> ]

#### 15. Tabel Rekening

Nama tabel : master\_rekening

Primary key: id\_master\_rekening

**Tabel 4.50** Tabel master\_rekening

	Field	Type	NULL	Default	Description
	ID_MASTER_REKENING	Integer	<b>not null</b>		No description [ <a href="#">edit</a> ]
	KODE_REKENING	Varchar(50)	<b>not null</b>		No description [ <a href="#">edit</a> ]
	NAMA_REKENING	Varchar(200)	<b>not null</b>		No description [ <a href="#">edit</a> ]
	SALDO_NORMAL	Integer	<b>not null</b>		No description [ <a href="#">edit</a> ]

## 16. Tabel Master Kecamatan

Nama tabel : master\_kecamatan

Primary key: id\_kecamatan

**Tabel 4.51** Tabel master\_kecamatan

	Field	Type	NULL	Default	Description
	ID_KECAMATAN	Integer	not null		No description [ <a href="#">edit</a> ]
	KODE_KECAMATAN	Varchar(10)	not null		No description [ <a href="#">edit</a> ]
	NAMA_KECAMATAN	Varchar(50)	not null		No description [ <a href="#">edit</a> ]

## 17. Tabel Master Kelurahan

Nama tabel : master\_desa

Primary key: id\_desa

Foreign key : id\_kecamatan

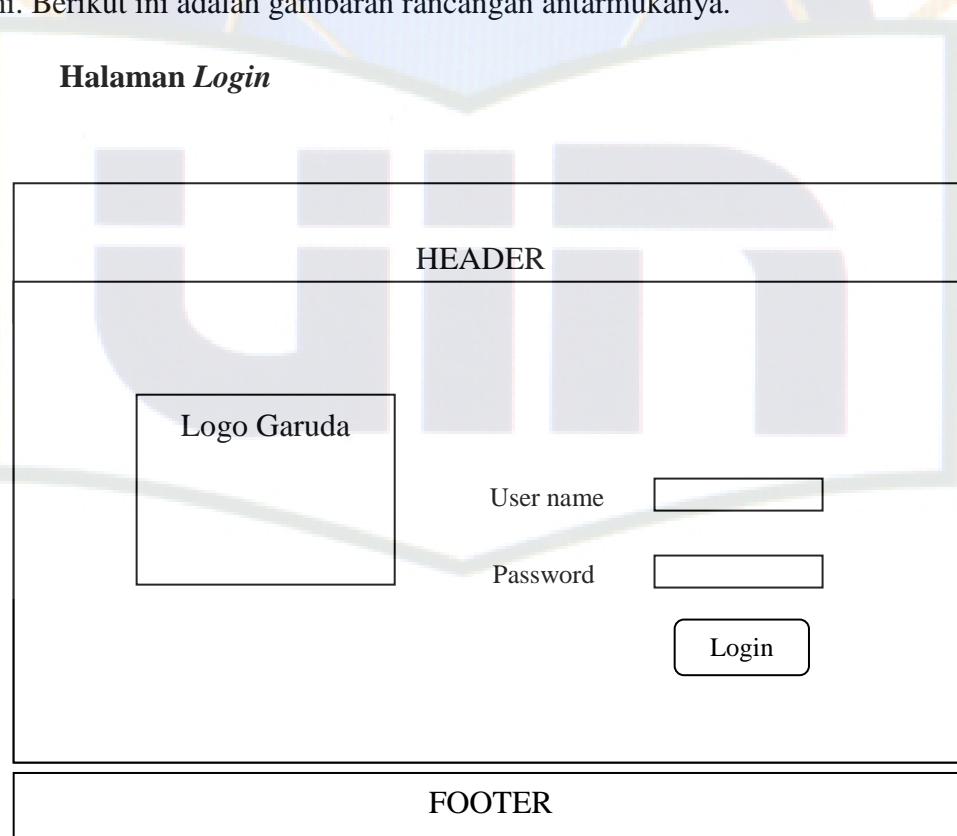
**Tabel 4.52** Tabel master\_desa

	Field	Type	NULL	Default	Description
	ID_DESA	Integer	not null		No description [ <a href="#">edit</a> ]
	ID_KECAMATAN	Char(11)	not null		No description [ <a href="#">edit</a> ]
	KODE_DESA	Varchar(10)	not null		No description [ <a href="#">edit</a> ]
	NAMA_DESA	Varchar(50)	not null		No description [ <a href="#">edit</a> ]

#### 4.3.9 Perancangan Tampilan

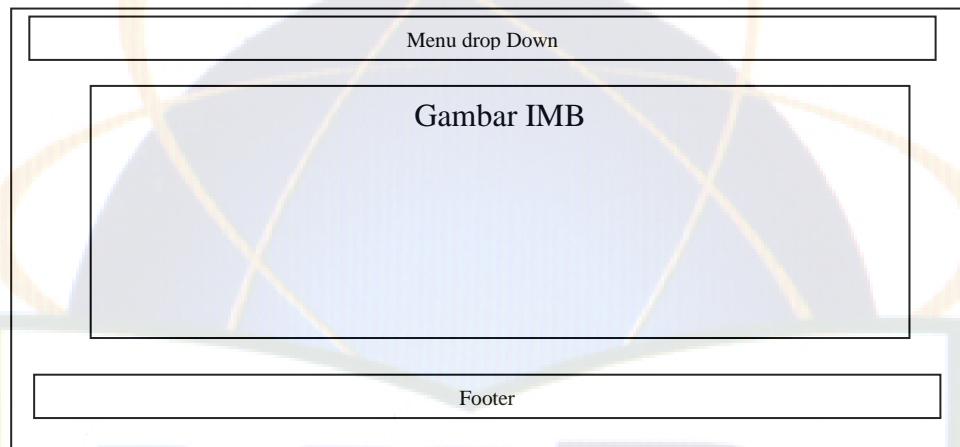
Pada tahap ini dilakukan perancangan tampilan (antarmuka) halaman-halaman web, dimana dari beberapa menu, yaitu *formlogin*, *home*, *permohonan*, *pemeriksaan*, *penetapan*, *penomoran* penyerahan dan pembayaran. Perancangan antarmuka ini diharapkan dapat memudahkan pengguna dalam menjalankan sistem ini. Berikut ini adalah gambaran rancangan antarmukanya.

##### a. Halaman *Login*



Gambar 4.31 Rancangan Halaman *Login*

**b. Halaman Utama**



**Gambar 4.32 Rancangan Halaman Utama**

**c. Halaman Pendaftaran**

The diagram shows the registration form for the application tab. It includes a header section labeled "Menu drop Down" and a footer section labeled "Footer". A navigation bar at the top right contains four tabs: "Permohonan", "Pemeriksaan", "Penetapan", and "Penolakan". Below this, another row of tabs includes "Permohonan", "Pemohon", "Komponen", and "Lokasi Bangunan". The main content area contains five input fields: "Tgl. Permohonan", "Jenis Permohonan", "No. Permohonan", "Perihal", and "Lampiran Status". Each input field is accompanied by a small square checkbox or button element to its right.

**Gambar 4.33 Rancangan Pendaftaran tab permohonan**

Menu drop Down

Permohonan	Pemeriksaan	Penetapan	Penolakan
Permohonan	Pemohon	Komponen	Lokasi Bangunan
No. ktp NPWRD Contact Person No. Telp Tempat lahir Alamat	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	Nama No. HP Tgl. Lahir <input type="text"/>
Nama Perusahaan	<input type="text"/>	Jabatan	<input type="text"/>
Footer			

**Gambar 4.34** Rancangan Pendaftaran tab pemohon

Menu drop Down

Permohonan	Pemeriksaan	Penetapan	Penolakan
Permohonan	Pemohon	Komponen	Lokasi Bangunan
Tgl. Permohonan	<input type="text"/>	Dinding	<input type="text"/>
Pondasi	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
Rangka Kap	<input type="text"/>	<input type="checkbox"/>	Atap <input type="text"/>
Kusen	<input type="text"/>	<input type="checkbox"/>	Lantai <input type="text"/>
Luas Bangunan	<input type="text"/>	M2	
Footer			

**Gambar 4.35** Rancangan Pendaftaran tab komponen bangunan

Menu drop Down

Permohonan	Pemeriksaan	Penetapan	Penolakan
Permohonan	Pemohon	Komponen	Lokasi Bangunan
Kampung/ Komplek Perumahan Jalan Kecamatan Kelurahan Berada diatas tanah Nomor			
<input type="button" value="Simpan"/> <input type="button" value="Reset"/>			
Footer			

**Gambar 4.36 Rancangan Pendaftaran tab lokasi bangunan**

**d. Pemeriksaan**

Menu drop Down

Permohonan	Pemeriksaan	Penetapan	Penolakan
No.Pendaftaran Tgl. Pemeriksaan No. BA. Pemeriksaan Nama Petugas Status Pemeriksaan			
Komponen Bangunan			
<input type="button" value="Simpan"/> <input type="button" value="Reset"/>			
Footer			

**Gambar 4.37 Rancangan Halaman Pemeriksaan**

### e. Form Data Pemeriksaan Lapangan

<b>Data Lapangan</b>			
Fisik Terbangun	<input type="text"/> %	Sepadan Bangunan	<input type="text"/>
GSP	<input type="text"/>	GSB	<input type="text"/>
GSP Samping	<input type="text"/>	GSB Samping	<input type="text"/>
KDB	<input type="text"/>	KLB	<input type="text"/>
Jumlah Lantai	<input type="text"/> Lantai	Lebar Bentangan	<input type="text"/>
Batas Lahan / Bangunan			
- Sebelah Utara	<input type="text"/> Lantai	- Sebelah Selatan	<input type="text"/>
- Sebelah Barat	<input type="text"/> Lantai	- Sebelah Timur	<input type="text"/>

<b>Data Gambar</b>	
<ul style="list-style-type: none"> <li>● Gambar Rencana</li> </ul>	
GSP	<input type="text"/> GSB
GSP Samping	<input type="text"/> GSB Samping
KDB	<input type="text"/> KLB
Jumlah Lantai	<input type="text"/> Lantai
Lebar Bentangan	

<ul style="list-style-type: none"> <li>● Struktur Konstruksi</li> </ul>	
Pondasi	<input type="text"/> Dinding
Rangka Kap	<input type="text"/> Atap
Struktur Kolom & Balok	<input type="text"/> Lantai

<ul style="list-style-type: none"> <li>● Kesimpulan</li> </ul>	
Kesimpulan	<input type="button" value="dapat diterbitkan"/>
Dengan alasan	<input type="text"/>

**Gambar 4.38** Rancangan Data Lapangan

**f. Penetapan**

The diagram illustrates the layout of the 'Penetapan' (Approval) page. At the top is a horizontal menu bar labeled 'Menu drop Down'. Below it is a horizontal navigation bar with four tabs: 'Permohonan', 'Pemeriksaan', 'Penetapan' (which is highlighted in blue), and 'Penolakan'. The main content area contains several input fields for data entry:

- 'No.Pendaftaran' (Registration Number) with an adjacent checkbox.
- 'Nomor SKRD' (SKRD Number)
- 'Tgl. Penetapan' (Approval Date)
- 'Alamat Pemohon' (Applicant Address)
- 'Pemohon Lokasi' (Applicant Location)
- 'NPWRD'

A section titled 'Komponen Biaya' (Cost Components) contains an input field for 'Biaya' (Fees).

At the bottom of the page are two buttons: 'Simpan' (Save) and 'Reset'. A footer bar at the very bottom is labeled 'Footer'.

**Gambar 4.39** Rancangan Halaman Penetapan

**g. Penomoran**

The diagram illustrates the layout of the 'Penomoran' (Numbering) page. At the top is a horizontal menu bar labeled 'Menu drop Down'. Below it is a horizontal navigation bar with two tabs: 'Entri Penomoran' and 'Sudah Diproses' (which is highlighted in blue). The main content area contains several input fields for data entry:

- 'No. Pendaftaran' (Registration Number)
- 'Nama Pemohon' (Applicant Name)
- 'Jenis Izin/ Akta' (Type of Permit/ Document)
- 'Tgl. Permohonan' (Application Date)
- 'Tgl. Pembayaran' (Payment Date)
- 'Tgl. Penomoran' (Numbering Date)
- 'Tanggal Penerbitan' (Issuance Date)
- 'Nomor Surat' (Document Number)

At the bottom of the page are two buttons: 'Simpan' (Save) and 'Reset'. A footer bar at the very bottom is labeled 'Footer'.

**Gambar 4.40** Rancangan Halaman Penomoran

### **h. Penyerahan**

The diagram illustrates the layout of the 'Penyerahan' (Handover) page. At the top is a header section with two buttons: 'Penyerahan' and 'Sudah diserahkan'. Below this is a large form area containing several input fields and labels. The labels are: 'No. Pendaftaran', 'Nama Pemohon', 'Jenis Izin', 'Nomor Surat', 'Tgl. Penomoran', 'Tgl. Penyerahan', 'Nama Pengambil', and 'Nomor KTP'. To the right of these labels are corresponding input fields. At the bottom of the form area are two buttons: 'Simpan' and 'Reset'. A 'Footer' section is located at the very bottom of the page.

**Gambar 4.41** Rancangan Halaman Penyerahan

### **i. Pembayaran**

The diagram illustrates the layout of the 'Pembayaran' (Payment) page. At the top is a header section with two buttons: 'Pembayaran' and 'Sudah dibayar'. Below this is a large form area containing several input fields and labels. The labels are: 'No. Pendaftaran', 'Nama Pemohon', 'Jenis Izin', 'Alamat Pemohon', 'Tgl. Permohonan', 'Tgl. Pembayaran', 'Jumlah', and 'Nomor Bukti'. To the right of these labels are corresponding input fields. At the bottom of the form area are two buttons: 'Simpan' and 'Reset'. A 'Footer' section is located at the very bottom of the page.

**Gambar 4.42** Rancangan Halaman Pembayaran

#### 4.4. Coding

Pada tahap ini dilaksanakan implementasi dari rancangan-rancangan, baik rancangan basis data, rancangan aplikasi, maupun rancangan tampilan.

##### 4.4.1. Bahasa Pemrograman dan Komponen

Bahasa pemrograman yang digunakan pada pengembangan aplikasi ini adalah menggunakan PHP 5.2. Untuk pembuatan laporan digunakan tool FPDF.

Firebird 2.0 digunakan sebagai database untuk menyimpan data dengan menggunakan tool Flamerobin sebagai database manajemen editor.

Untuk menjalankan codingan diperlukan *application server*, pada pengembangan sistem ini digunakan apache2. Untuk editor dan unit test digunakan eclipse editor.

Untuk *tools pair programming* yang dilakukan menggunakan aplikasi subversion yang berfungsi sebagai salah satu media komunikasi sesama developer sistem.

##### 4.4.2. Implementasi Coding

Pada implementasi coding terdapat beberapa module yang dibuat, diantaranya adalah :

a. Modul Aktivitas Izin Mendirikan Bangunan

Modul ini digunakan dalam aktivitas pendaftaran, pemeriksaan dan penetapan izin. Modul ini terletak di dalam folder /var/www/simyandu/extensions/simatap/izin3. Didalam folder tersebut terdapat file-file source code php dan javascript, dimana

file-file javascript yang ada digunaakan untuk fungsi tanggal, fungsi data grid dan fungsi lainnya. Untuk source code lengkap terdapat dilampiran.

b. Modul Pembayaran

Modul ini digunakan untuk melayani perekaman pembayaran yang dilakukan oleh wajib retribusi. Modul ini terletak di dalam folder /var/www/simyandu/extensions/simatap/pembayaran.

Untuk source code lengkap terdapat pada lampiran.

c. Modul Penomoran

Modul ini digunakan untuk melayani perekaman penomoran surat izin yang dilakukan oleh petugas penomoran. Modul ini terletak di dalam folder /var/www/simyandu/extensions/simatap/validasi.

Untuk source code lengkap terdapat pada lampiran.

d. Modul Penyerahan

Modul ini digunakan untuk melayani perekaman penyerahan surat izin yang dilakukan oleh petugas penyerahan. Modul ini terletak di dalam folder /var/www/simyandu/extensions/simatap/penomoran. Untuk source code lengkap terdapat pada lampiran.

Untuk Jenis Report penulis menggunakan FPDF untuk-report Berita Acara peninjauan Lapangan, Surat Izin, SKRD, Nota Perhitungan, dll. Sedangkan untuk desain tampilan agar lebih baik penulis menggunakan *Yahoo user Interface (YUI)*. Untuk source code penggunaan YUI dan FPDF terdapat pada lampiran.

#### 4.5. Pengujian (*Testing*)

Setelah selesai melakukan pengembangan, maka aplikasi ini harus diujicoba yang bertujuan untuk mengetahui sejauh mana aplikasi ini dapat bekerja dengan baik dan apakah aplikasi ini dapat memenuhi tujuan yang ingin diperoleh sebelum diserahkan kepada *end user*. Pengujian dilakukan dengan *deployment* terhadap aplikasi yang telah dikembangkan ke target server tertentu, url yang bisa diakses untuk testing adalah <http://192.168.10.1/simyandu>. Pengujian dilakukan dalam 2 (dua) tahap yaitu pengujian mandiri dan pengujian oleh *user* secara *blackbox*.

#### 4.5.1. Pengujian Mandiri

**Tabel 4.53** Hasil pengujian mulai dari pendaftaran sampai terbit Surat Ijin

No	Kasus Uji	Prasyarat dan Langkah-langkah	Metode	Hasil yang diharapkan	Hasil Aktual	Hasil uji coba
1	Input Data Pendaftaran.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai <i>pendaftar</i></li> <li>- Klik menu <i>aktivitas</i> pilih menu <i>izin Mendirikan bangunan</i></li> <li>- Masukan data-data pemohon dan komponen bangunan.</li> <li>- Klik tombol Simpan</li> </ul>	Demo, Inspeksi	Data pendaftaran Tersimpan ke dalam database	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan Data Tersimpan dan Tombol Preview Kartu Daftar</li> </ul>	OK

2	Edit Data Pendaftaran.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai <i>pendaftar</i></li> <li>- Klik menu <i>daftar perizinan</i> pilih menu <i>izin Mendirikan bangunan</i></li> <li>- Rubah data pemohon atau komponen bangunan yang ingin dirubah.</li> <li>- Klik tombol Simpan</li> </ul>	Demo, Inspeksi	Bagian pendaftaran dapat melakukan edit pada data pendaftar permohonan izin mendirikan bangunan kedalam database.	<ul style="list-style-type: none"> <li>- Data pada database telah terupdate</li> <li>- Pesan Data Terupdate dan Tombol Preview Kartu Daftar</li> </ul>	OK
---	------------------------	--	----------------	---	--	----

3	Input Data Pemeriksaan.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai pengendalian dan Data pendaftaran sebelumnya sudah ada di database</li> <li>Data sudah ada di database</li> <li>- Klik menu <i>aktivitas</i> dan pilih <i>izin mendirikan bangunan</i>, kemudian klik tab <i>pemeriksaan</i></li> <li>- Isi data hasil pemeriksaan lapangan berupa, komponen-komponen bangunan,</li> </ul>	Demo, Inspeksi	<p>Bagian pengawasan dan pengendalian dapat memasukan data detail hasil pemeriksaan kedalam database.</p>	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan data pemeriksaan berhasil tersimpan dan tombol preview Berita Acara Pemeriksaan</li> </ul>	OK
---	-------------------------	--	----------------	---	--	----

		<p>Guna Bangunan, dll</p> <ul style="list-style-type: none"> <li>- Klik tombol Simpan</li> </ul>				
4	Edit Data Pemeriksaan.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai pengendalian dan Data pemeriksaan sebelumnya sudah ada di database Data sudah ada di database</li> <li>- Klik menu <i>perizinan</i> dan pilih <i>izin mendirikan bangunan</i>, kemudian klik tab <i>pemeriksaan</i> dan pilih data yang ingin dirubah</li> </ul>	Demo, Inspeksi	Pengendalian dapat mengedit data pemeriksaan hasil lapangan yang sudah tersimpan sebelumnya pada database.	<ul style="list-style-type: none"> <li>- Data pada database terupdate</li> <li>- Pesan data pemeriksaan berhasil terupdate dan tombol preview Berita</li> </ul>	OK

		<ul style="list-style-type: none"> <li>- Rubah data hasil pemeriksaan lapangan berupa atau komponen-komponen bangunan, Guna Bangunan, dll</li> <li>- Klik tombol Simpan</li> </ul>	Demo, Inspeksi	Bendahara dapat melakukan perhitungan dan memasukan data biaya penetapan ke dalam database.	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan data berhasil disimpan dan tombol preview nota perhitungan dan SKRD</li> </ul>	OK
5	Input data Penetapan.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai bendahara, Data Hasil Pemeriksaan sudah ada dalam database.</li> <li>- Klik <i>aktivitas</i> pilih <i>izin mendirikan bangunan</i> dan klik tab <i>penetapan</i></li> <li>- Isi nomor SKRD dan</li> </ul>				

		<p>tentukan nilai dasar bangunan, lalu tekan tombol hitung</p> <ul style="list-style-type: none"> <li>- Tekan tombol Simpan</li> </ul>				
6	Edit data penetapan.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai bendahara, Data Hasil Pemeriksaan sudah ada dalam database.</li> <li>- Klik <i>Daftar Perizinan</i> pilih <i>izin mendirikan bangunan</i> dan klik tab <i>penetapan</i> dan pilih data yang ingin di edit</li> <li>- Rubah nomor SKRD</li> </ul>	Demo, Inspeksi	Bendahara dapat memasukan data biaya penetapan ke dalam database.	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan data berhasil diupdate dan tombol preview nota perhitungan dan SKRD</li> </ul>	OK

		<p>atau nilai dasar bangunan, lalu tekan tombol hitung</p> <ul style="list-style-type: none"> <li>- Tekan tombol Simpan</li> </ul>				
7	Input data pembayaran.	<ul style="list-style-type: none"> <li>- <i>Telah Login sebagai bendahara, Data Penetapan biaya sudah ada dalam database.</i></li> <li>- Klik menu Pembayaran</li> <li>- Pilih nomor pendaftaran, masukan nama pembayar</li> <li>- Klik tombol Simpan</li> </ul>	Demo, Inspeksi	Bendahara dapat memasukan data pembayaran retribusi ke dalam database.	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan Data Berhasil disimpan dan tombol preview bukti pembayaran</li> </ul>	OK

8	Input Data Penomoran.	<ul style="list-style-type: none"><li>- Telah <i>Login</i> sebagai penomoran, data sudah sampai ke pembayaran.</li><li>- Klik menu penomoran</li><li>- Pilih nomor pendaftaran, masukan nomor Surat Ijin</li><li>- Klik tombol Simpan</li></ul>	Demo, Inspeksi	Bagian penomoran dapat memasukan penomoran surat izin kedalam database.	<ul style="list-style-type: none"><li>- Data tersimpan kedalam database</li><li>- Pesan Data Penomoran Berhasil Disimpan</li></ul>	OK
---	-----------------------	---	----------------	---	--	----

9	Input Data penyerahan.	<ul style="list-style-type: none"><li>- Telah <i>Login</i> sebagai penyerahan, data sudah sampai ke penomoran.</li><li>- Klik menu penyerahan</li><li>- Masukan nama dan nomor KTP pengambil izin.</li><li>- Klik tombol Simpan</li></ul>	Demo, Inspeksi	Bagian penyerahan dapat memasukan data penyerahan surat izin kedalam database.	<ul style="list-style-type: none"><li>- Data tersimpan kedalam database</li><li>- Pesan Bahwa Izin Telah Sampai ketahap Penyerahan ke Pemohon</li></ul>	OK
---	------------------------	---	----------------	--	---	----

#### 4.5.2. Pengujian oleh *user*

Pada pengujian kedua dilakukan pengujian oleh *user*. Bukti terlampir uji coba yang dilakukan dapat dilihat pada bagian lampiran. Secara ringkas, hasil pengujian yang dapat penulis simpulkan adalah sebagai berikut:

Tabel 4.52 Pengujian Lapangan

No	Pengujian	Penilaian
1.	Fitur aplikasi secara keseluruhan	Baik
2.	Fitur untuk Admin	Baik
3.	Fitur untuk pendaftaran	Baik
4.	Fitur untuk pemeriksaan	Baik
5.	Fitur untuk penetapan	Baik
6.	Fitur untuk Pembayaran	Baik
6.	Fitur untuk penomoran	Baik
7.	Fitur untuk penyerahan	Baik
8.	Kestabilan Aplikasi	Baik
9.	Keamanan Aplikasi	Baik
10.	Kesesuaian dengan Kebutuhan	Baik

## BAB V

### KESIMPULAN DAN SARAN

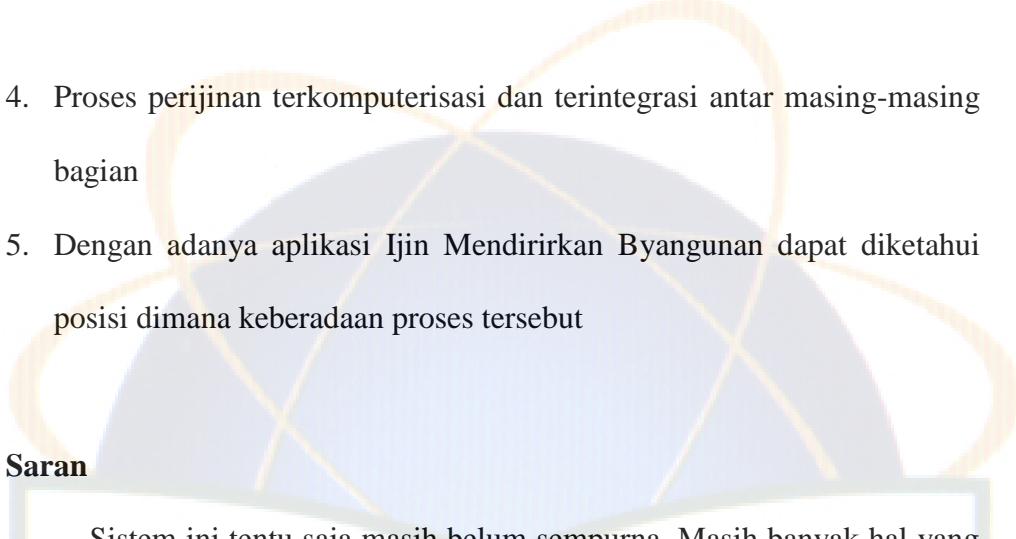
Aplikasi Ijin Mendirikan Bangunan ini diharapkan dapat mengurangi permasalahan-permasalahan yang berhubungan dengan pengolahan data, perhitungan nilai retribusi serta dokumentasi pada Izin Mendirikan Bangunan (IMB) Badan Pelayanan Perijinan Terpadu Tangerang Selatan.

Setelah melakukan serangkaian penelitian, maka pada bab ini penulis akan menguraikan kesimpulan dan saran yang dapat diambil dari rangkaian penelitian tersebut. Saran yang diberikan diharapkan dapat bermanfaat bagi pihak-pihak yang akan melanjutkan pengembangan penelitian ini.

#### 5.1. Kesimpulan

Dari penelitian yang telah peneliti uraikan, maka dapat ditarik kesimpulan sebagai berikut:

1. Dapat menghindarkan kesalahan dalam perhitungan berasarkan koefisien-koefisien yang didapat.
2. Aplikasi yang dibuat memudahkan masing-masing bagian terkait dalam hal membuat laporan-laporan karena telah terformat dan terstruktur secara dinamis.
3. Memudahkan petugas terkait dalam mendata semua data pemohon yang mengajukan IMB.

- 
4. Proses perijinan terkomputerisasi dan terintegrasi antar masing-masing bagian
  5. Dengan adanya aplikasi Ijin Mendirikan Bangunan dapat diketahui posisi dimana keberadaan proses tersebut

## 5.2. Saran

Sistem ini tentu saja masih belum sempurna. Masih banyak hal yang dapat dilakukan untuk mengembangkan sistem ini agar menjadi lebih baik lagi, antara lain:

1. Diharapkan pengembangan dapat berjalan secara online, tetapi user (masyarakat maupun petugas BP2T) dapat melihat proses maupun posisi izin mulai dari pendaftaran sampai ke tahap penerbitan SK bisa dilakukan secara online dan tidak hanya pada Izin Mendirikan Bangunan. Di samping itu user juga dapat mengakses informasi secara transparan, dapat berjalan sesuai dengan regulasi unit user serta dapat menciptakan akuntabilitas publik yang baik.
2. Informasi terhadap bagi pendaftar (waktu pembayaran retribusi, waktu pengambilan ijin, dll) dapat diinformasikan secara realtime melalui sms gateway atau web online.

## DAFTAR TABEL

Halaman

Tabel 4.1. <i>Cause and Effect Analysis</i> .....	99
Tabel 4.2. <i>System Improvement Objectives</i> .....	100
Tabel 4.3. <i>Nonfunctional Requirement</i> .....	103
Tabel 4.4. <i>Requirement</i> Aktor dan <i>Use case</i> .....	106
Tabel 4.5. Spesifikasi Naratif untuk <i>Use Case</i> menambah S.K pada data master Tingkat Perancangan.....	110
Tabel 4.6. Spesifikasi Naratif untuk <i>Use Case</i> Melihat S.K pada data master Tingkat Perancangan .....	111
Tabel 4.7. Spesifikasi Naratif untuk <i>Use Case</i> Mengedit data master Surat Keputusan Tingkat Perancangan .....	112
Tabel 4.8. Spesifikasi Naratif untuk <i>Use Case</i> Menghapus S.K pada data master Perancangan.....	113
Tabel 4.9. Spesifikasi Naratif untuk <i>Use Case</i> Menambah S.K pada menu mutasi pegawai Tingkat Perancangan.....	114
Tabel 4.10. Spesifikasi Naratif untuk <i>Use Case</i> Melihat S.K pada menu Mutasi Pegawai Tingkat Perancangan.....	115
Tabel 4.11. Spesifikasi Naratif untuk <i>Use Case</i> Mencetak S.K pada Tingkat Perancangan.....	116
Tabel 4.12. Spesifikasi Naratif untuk <i>Use Case</i> Mengedit S.K pada	

Tingkat Perancangan.....	117
 Tabel 4.13. Spesifikasi Naratif untuk <i>Use Case</i> Menghapus S.K pada Tingkat Perancangan.....	118
 Tabel 4.14. Spesifikasi Naratif untuk <i>Use Case</i> Melihat S.K pada pegawai di tingkat Perancangan.....	119
 Tabel 4.15. Spesifikasi Naratif untuk <i>Use Case</i> Mencetak S.K pada Tingkat Perancangan.....	120
 Tabel 4.16. Spesifikasi Naratif untuk <i>Use Case</i> Level User S.K pada Tingkat Perancangan.....	121
 Tabel 4.17. Tabel Pegawai.....	158
 Tabel 4.18. Tabel riwayat_pangkat.....	159
 Tabel 4.19. Tabel unit_kerja .....	159
 Tabel 4.20. Tabel naik_pangkat.....	160
 Tabel 4.21. Tabel user.....	160
 Tabel 4.22. Tabel konten .....	161
 Tabel 4.23. Tabel hubungi .....	161
 Tabel 4.24. Tabel golongan.....	162
 Tabel 4.25. Tabel leveluser .....	162
 Tabel 4.26. Tabel naik_pangkat.....	163
 Tabel 4.27. Tabel penetapan .....	164
 Tabel 4.28. Tabel pindah .....	165
 Tabel 4.29. Tabel pns_pegawai .....	166

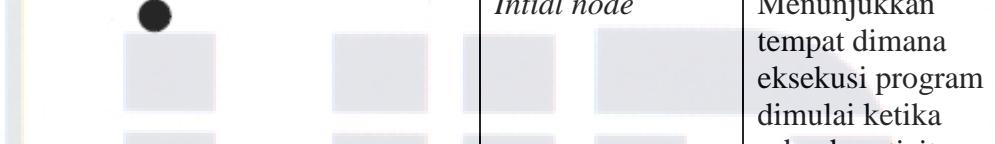
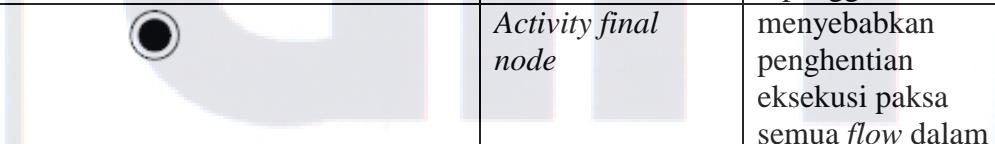
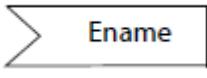
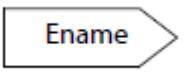
Tabel 4.30. Tabel tanda_tangan .....	166
Tabel 4.31. Tabel tembusan .....	167
Tabel 4.32. Tabel manimbang.....	167
Tabel 4.33. Tabel membaca .....	168
Tabel 4.34. Tabel memperhatikan.....	168
Tabel 4.35. Tabel mengingat.....	168
Tabel 4.36. Daftar <i>Tools</i> Pengembangan Perangkat Lunak Sistem.....	172
Tabel 4.37. Pengujian Mandiri.....	174
Tabel 4.38. Pengujian Penerimaan Sistem.....	179

## Simbol Use Case Diagram

Simbol	Nama	Fungsi
	<i>Actor</i>	<i>User</i> di luar sistem
	<i>Use case</i>	Sebuah spesifikasi perilaku suatu entitas dalam interaksi dengan agen di luar
	<i>Subject Boundary</i>	<i>Classifier</i> yang sifatnya dideskripsikan dengan <i>use case</i>
	<i>Communication path</i>	Penjelasan tentang sambungan antara <i>class instances</i>
	<i>Generalization</i>	Hubungan antara lebih spesifik dan deskripsi yang lebih umum, digunakan untuk <i>inheritance</i> dan deklarasi tipe polimorfik
	<i>Extend</i>	menghubungkan sebuah perluasan <i>use case</i> ke <i>use case</i> dasar
	<i>Include</i>	menghubungkan <i>use case</i> dasar ke <i>use case</i> tambahan

(Sumber: Rumbaugh, Jacobson, Booch, 2006: 48-49, 695, 351, 388, 627)

### Simbol Activity Diagram

Simbol	Nama	Fungsi
	Action atau activity	perilaku spesifikasi yang menggambarkan sekuensial serta langkah-langkah prosedur komputasi.
	Fork atau join	Node dengan satu input dan beberapa output atau sebaliknya
	Initial node	Menunjukkan tempat dimana eksekusi program dimulai ketika sebuah <i>activity</i> dipanggil
	Activity final node	menyebabkan penghentian eksekusi paksa semua <i>flow</i> dalam <i>activity</i> dan berakhirnya pelaksanaan <i>activity</i>
	Flow	Mendeskripsikan sumber dan target dari berbagai jenis informasi
	Accept action	Suatu aksi yang melakukan eksekusi sampai suatu jenis kejadian tertentu dikenal oleh objek pelaksana
	Send signal action	Spesifikasi dari blok informasi yang dikomunikasikan dengan <i>asynchronous</i> antar objek

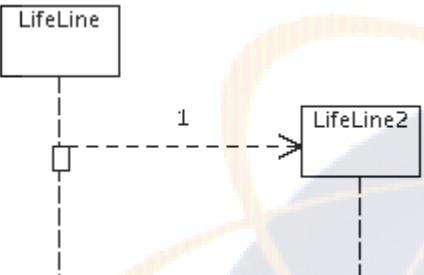
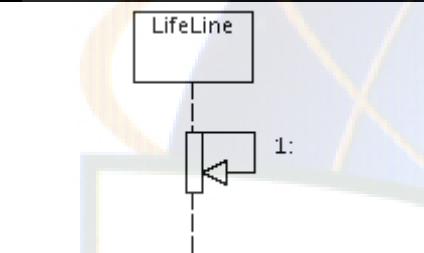
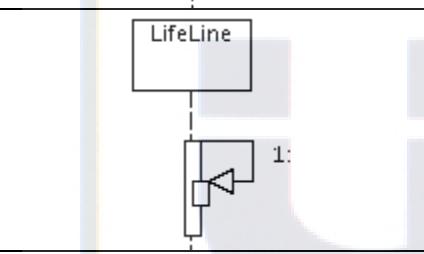
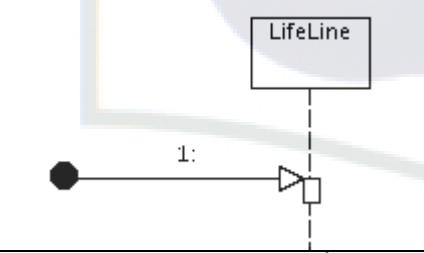
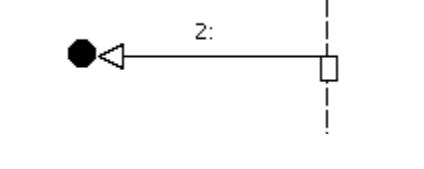
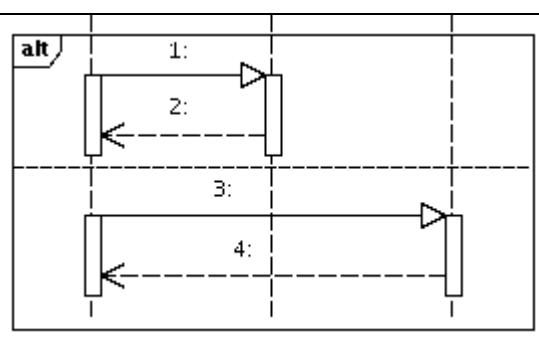
	<i>Accept time action</i>	Suatu aksi yang dilakukan dalam waktu tertentu
	<i>Branch</i> atau <i>merge</i>	Situasi dimana <i>flow</i> input aksi dapat menuju satu atau beberapa kemungkinan <i>flow</i> output atau sebaliknya
	<i>Pin</i>	Merepresentasikan koneksi untuk nilai input atau output suatu aksi
	<i>Object node</i>	Merepresentasikan adanya suatu objek yang diproduksi oleh aksi
	<i>Exception handler</i>	Kontrol yang melakukan asumsi jika ada pengecualian dari suatu kejadian yang terjadi
	<i>Interruptible activity region</i>	Suatu daerah yang dapat dilakukan proses interupsi
	<i>Expansion region</i>	Suatu daerah yang mengeksekusi satu dari setiap elemen yang diinput
	<i>Sub-activity</i>	Activity diagram utama terlalu kompleks sehingga ditulukan <i>activity</i> diagram tersendiri untuk proses tertentu

(Sumber: <http://www.holub.com/> dan Rumbaugh, Jacobson, Booch, 2006: 698, 169, 385,

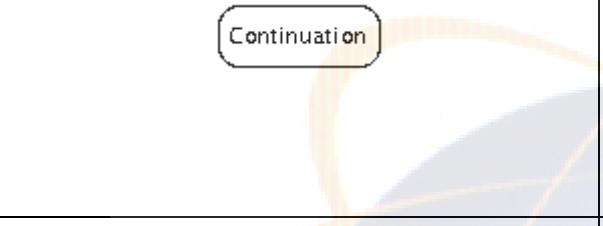
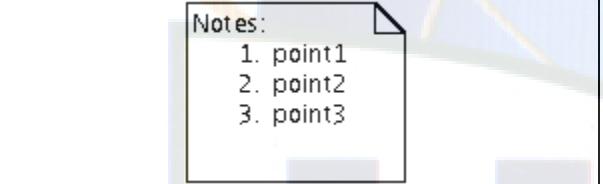
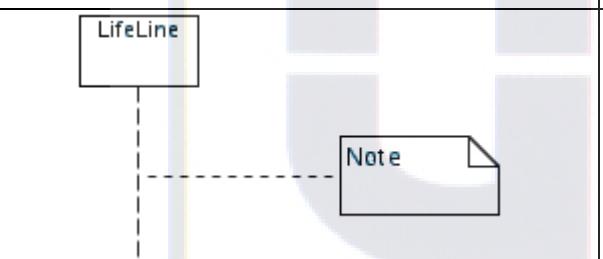
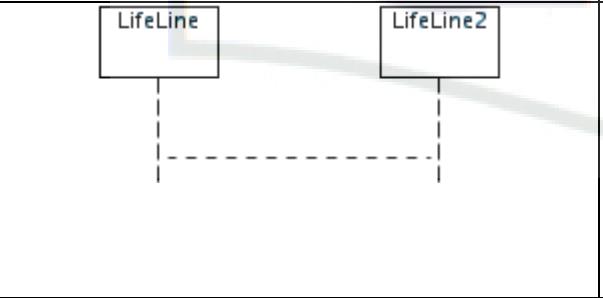
448, 412, 178, 383, 155, 601, 669, 219, 469, 540, 507, 361, 443, 367 )

## Simbol Sequencial Diagram

Simbol	Nama	Fungsi
	<i>LifeLine</i>	Partisipasi individu dalam interaksi
	<i>Boundary LifeLine</i>	Jenis penghubung yang didefinisikan sebagai stereotip Batas.
	<i>Control LifeLine</i>	Jenis penghubung yang didefinisikan sebagai stereotip Kontrol.
	<i>Entity LifeLine</i>	Sebuah jenis penghubung yang didefinisikan sebagai stereotip entitas.
	<i>asynchronous message</i>	Komunikasi antar lifeline
	<i>synchronous call</i>	Tipe message yang merupakan tipe operasi call
	<i>return message</i>	Tipe message yang merupakan akhir dari eksekusi dan kembali kepada pemanggilnya
	<i>Duration Message</i>	Tipe pesan yang bergantung pada waktu panggilan pesan
	<i>stop (destruction)</i>	Penghapusan objek

	<i>Create Message</i>	Model penciptaan objek.
	<i>Self Message</i>	Jenis pesan yang merupakan pelaksanaan atau operasi memanggil dalam garis hidup yang sama.
	<i>Recursive Message</i>	Jenis pesan diri yang dijalankan secara rekursif.
	<i>Found Message</i>	Jenis pesan mana terjadinya peristiwa penerima dikenal.
	<i>Lost Message</i>	Jenis pesan mana terjadinya peristiwa pengiriman dikenal.
	<i>Alternative Combined Fragment</i>	Sebuah fragmen gabungan alternatif menetapkan alternatif fragmen interaksi.

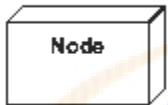
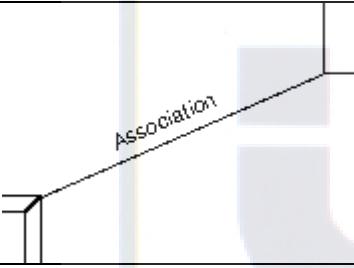
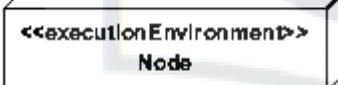
	<i>Loop Combined Fragment</i>	Sebuah fragmen yang mendefinisikan perulangan fragmen interaksi.
	<i>Interaction Use</i>	Sebuah cara singkat untuk menyalin isi dari interaksi yang dirujuk.
	<i>Frame</i>	Sebuah unit yang mempunyai perilaku untuk berfokus pada pengamatan pertukaran informasi
	<i>Actor</i>	Aktor menentukan peran yang dimainkan oleh pengguna manusia, perangkat keras eksternal, atau subjek lain yang berinteraksi dengan subjek.
	<i>Concurrent</i>	Membuat timeline eksesuki baru bersamaan

	<i>Continuation</i>	Mendefinisikan kelanjutan dari cabang <i>Alternative Combined Fragment</i> yang berbeda
	<i>Gate</i>	Titik sambungan antara pesan internal dan eksternal dari Frame
	<i>Note</i>	Sebuah notasi tekstual yang dapat melekat pada elemen (s).
	<i>Anchor</i>	Sebuah konektor antara elemen dan catatan.
	<i>Constraint</i>	kendala adalah suatu kondisi atau pembatasan, yang dapat disajikan dalam kedua bahasa natual atau bahasa mesin yang dapat dibaca.

(Sumber: <http://resource.visual-paradigm.com/> dan Rumbaugh, Jacobson, Booch, 2006:

700, 701, 319)

## Simbol *Deployment Diagram*

Simbol	Nama	Fungsi
	<i>Device Node</i>	sumber daya komputasi fisik dengan kemampuan pemrosesan untuk mengeksekusi artefak
	<i>Artifact</i>	Artefak adalah informasi yang digunakan atau dihasilkan oleh proses pengembangan perangkat lunak.
	<i>Association</i>	Sebuah asosiasi yang menetapkan hubungan semantik yang dapat terjadi antar instansi
	<i>Execution Environment Node</i>	Sebuah <i>execution environment</i> adalah node yang menawarkan lingkungan eksekusi untuk tipe komponen tertentu komponen sebagai syarat untuk mengeksekusi artefak

(Sumber : <http://www.visual-paradigm.com>)

## DAFTAR ISTILAH

<i>SKRD</i>	Surat Ketetapan Retribusi Daerah.
<i>BAPL</i>	Berita Acara Pemeriksaan Lapangan.
<i>BP2T</i>	Badan Pelayanan Perijinan Terpadu.
<i>Blackbox</i>	Suatu jenis pengujian perangkat lunak berdasarkan dengan input, output pemakai.
<i>Coding</i>	Tahapan dimana pengembangan software melakukan pembangunan aplikasi.
<i>Design</i>	Tahapan dimana pengembangan software melakukan perancangan.
<i>Testing</i>	Tahapan dimana pengembangan software melakukan uji coba terhadap aplikasi.
<i>UML</i>	<i>Unified Modelling Language</i>
<i>User</i>	Pengguna dari sistem
<i>User Friendly</i>	Bersahabat untuk digunakan, sehingga user mudah untuk menggunakan.
<i>XP</i>	<i>Extreme Programming</i>

## DAFTAR PUSTAKA

- Walgitto, Bimo. 1987. *Psikologi Sosial (Suatu Pengantar)*. Yogyakarta: Yayasan Penerbit Fakultas Psikologi Universitas Gajah Mada.
- Kuncoro, Mudrajad. 2003. Metoda Riset Bisnis dan Ekonomi: Bagaimana Meneliti dan Menulis Tesis (Research Methods for Business & Economics: How to Research & Write a Thesis). Jakarta: Erlangga.
- Astamal, Rio. 2006. *Menjadi Web Master dalam 30 Hari*. [Online] Tersedia: [www.rahasia-webmaster.com](http://www.rahasia-webmaster.com). [15 Mei 2010]
- Aswandi. 2006. *Aplikasi Berbasis Web*. [Online] Tersedia: <http://aswandi.or.id/2006/01/20/aplikasi-berbasis-web>. [10 April 2010]
- Hakim, Lukmanul. 2009. *Trik Rahasia Master PHP Terbongkar Lagi*. Yogyakarta: Lokomedia.
- Hariyanto, Bambang. 2004. *Sistem Manajemen Basis Data : Pemodelan, Perancangan, dan Terapannya*. Bandung : Informatika.
- Jogianto, HM. 2005. *Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi.
- Juju, Dominikus. 2007. *Buku Latihan Dreamweaver CS3*. Jakarta: PT. Elex Media Komputindo.
- Kadir, Abdul. 2003. *Pengenalan Sistem Informasi*. Yogyakarta: Andi.
- Kendall, Kenneth E. dan Kendall, Julie E. 2003. *Analisis dan Perancangan Sistem*. Jakarta : Pearson Education Asia Pte. Ltd dan PT. Prenhallindo.

- Ladjamudin, Albahra. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Mulyanto, Agus. 2009. *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta: Pustaka Pelajar.
- Pramono, Andi dan M. Syafii. 2006. *Kolaborasi Flash, Dreamweaver, dan PHP untuk Aplikasi Website*. Yogyakarta: Andi.
- Sutabri, Tata. 2005. *Sistem Informasi Manajemen*. Yogyakarta: Andi.
- Whitten L, Jeffrey. 2007. *System Analysis and Design Methods E. 7th*. Yogyakarta: McGraw-Hill.
- Marko Publishing McLeod, Raymond Jr., dan George Schell. 2008. *Sistem Informasi Manajemen Jilid 10*. PT. Prenhallindo dan Index: Jakarta
- Nugroho, Adi. 2002. *Analisis & Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*. Informatika: Bandung.
- Hariyanto, Bambang. 2005. *Rational Rose untuk Pemodelan Berorientasi Objek*. Informatika: Bandung.



## **LAMPIRAN-LAMPIRAN**

- 1. LAMPIRAN A : HASIL WAWANCARA**
- 2. LAMPIRAN B : HASIL TAMPILAN ANTARMUKA**
- 3. LAMPIRAN C : SOURCE CODE**
- 4. LAMPIRAN D : PENGUJIAN PENERIMAAN SISTEM**
- 5. LAMPIRAN E : PERATURAN DAERAH**

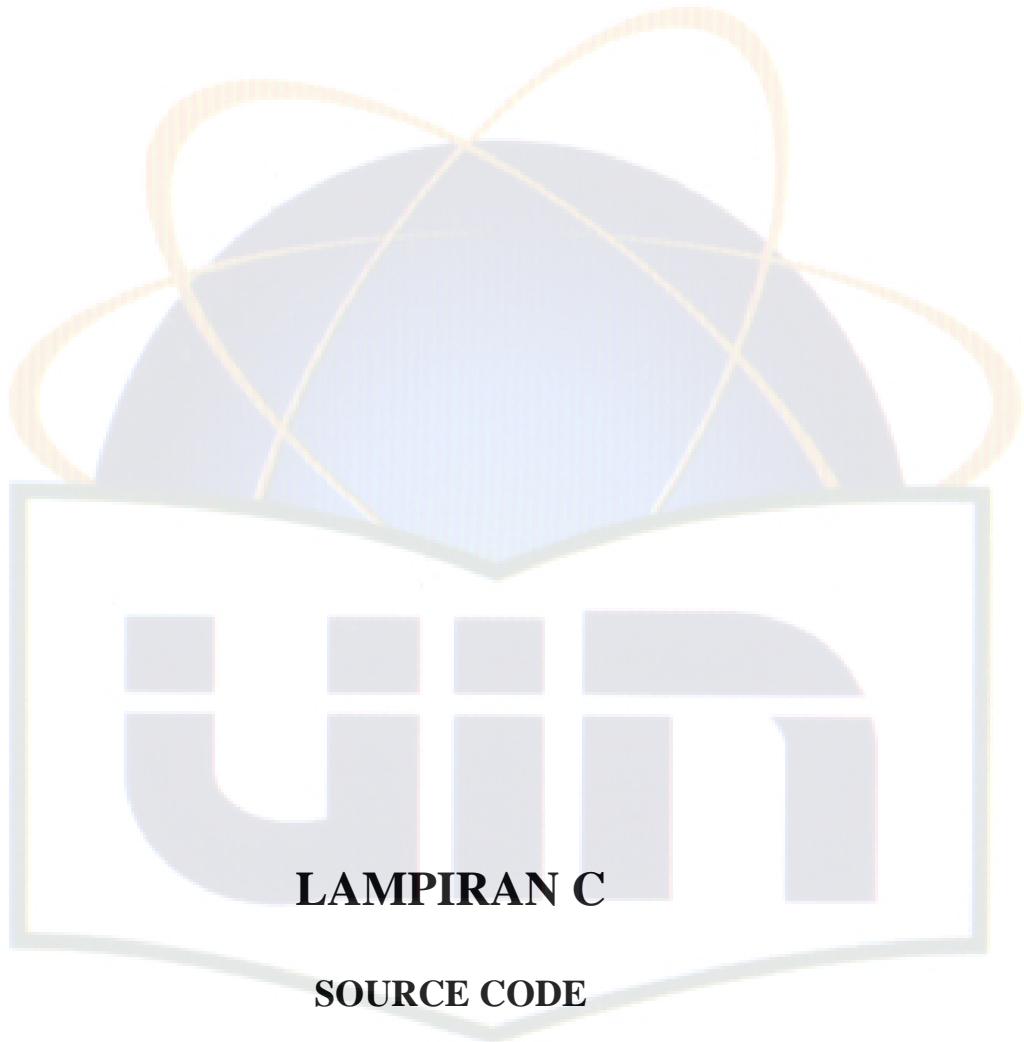
6.





**LAMPIRAN B**

**HASIL TAMPILAN ANTARMUKA**





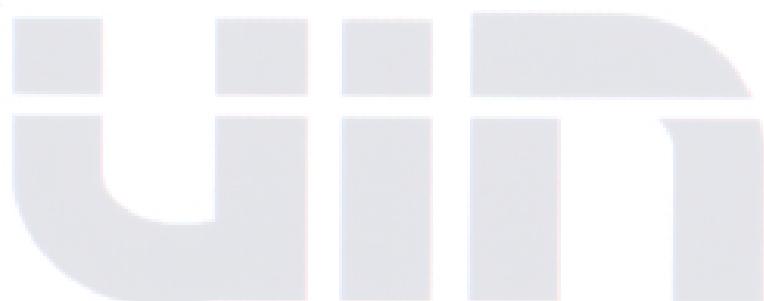
## LAMPIRAN D

### PENGUJIAN PENERIMAAN SISTEM



## **LAMPIRAN E**

### **PERATURAN DAERAH KABUPATEN TANGERANG**



## HASIL TESTING APLIKASI

11 MEI 2011, 12.00 wib, BP2T TANGERANG SELATAN

### Peserta :

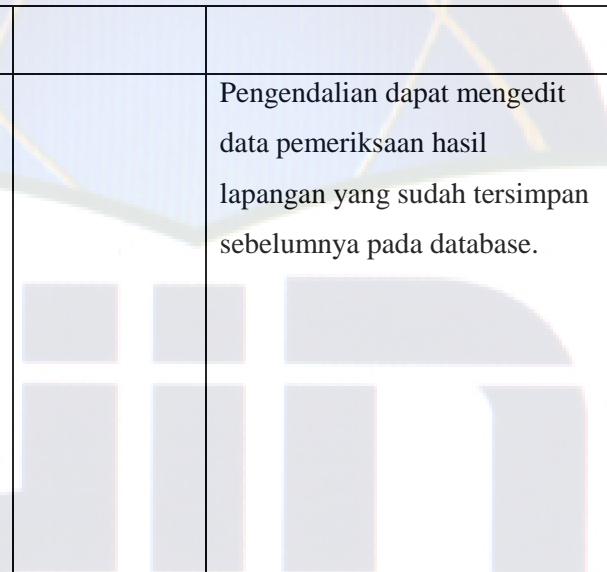
#### No. Unit Perusahaan

1. Pendaftaran
2. Pengawasan dan Pengendalian
3. Bendahara
4. Validasi

No	Kasus Uji	Prasyarat dan Langkah-langkah	Metode Kualifikasi	Hasil yang diharapkan	Hasil Aktual	Hasil uji coba
1	Input Data Pendaftaran.	- Telah <i>Login</i> sebagai <i>pendaftar</i>		Data pendaftaran Tersimpan ke dalam database	- Data Telah tersimpan kedalam database	

		<ul style="list-style-type: none"> <li>- Klik menu <i>aktivitas</i> pilih menu <i>izin Mendirikan bangunan</i></li> <li>- Masukan data-data pemohon dan komponen bangunan.</li> <li>- Klik tombol Simpan</li> </ul>			<ul style="list-style-type: none"> <li>- Pesan Data Tersimpan dan Tombol Preview Kartu Daftar</li> </ul>	
2	Edit Data Pendaftaran.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai <i>pendaftar</i></li> <li>- Klik menu <i>daftar perijinan</i> pilih menu <i>izin Mendirikan bangunan</i></li> <li>- Rubah data pemohon atau komponen bangunan yang ingin dirubah.</li> </ul>		Bagian pendaftaran dapat melakukan edit pada data pendaftar permohonan izin mendirikan bangunan kedalam database.	<ul style="list-style-type: none"> <li>- Data pada database telah terupdate</li> <li>- Pesan Data Terupdate dan Tombol Preview Kartu Daftar</li> </ul>	

		<ul style="list-style-type: none"> <li>- Klik tombol Simpan</li> </ul>				
3	Input Data Pemeriksaan.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai pengendalian dan Data pendaftaran sebelumnya sudah ada di database Data sudah ada di database</li> <li>- Klik menu <i>aktivitas</i> dan pilih <i>ijin mendirikan bangunan</i>, kemudian klik tab <i>pemeriksaan</i></li> <li>- Isi data hasil pemeriksaan lapangan berupa, komponen-komponen bangunan, Guna Bangunan, dll</li> </ul>		<p>Bagian pengawasan dan pengendalian dapat memasukan data detail hasil pemeriksaan kedalam database.</p>	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan data pemeriksaan berhasil tersimpan dan tombol preview Berita Acara Pemeriksaan</li> </ul>	

		<ul style="list-style-type: none"> <li>- Klik tombol Simpan</li> </ul>				
4	Edit Data Pemeriksaan.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai pengendalian dan Data pemeriksaan sebelumnya sudah ada di database Data sudah ada di database</li> <li>- Klik menu <i>perijinan</i> dan pilih <i>ijin mendirikan bangunan</i>, kemudian klik tab <i>pemeriksaan</i> dan pilih data yang ingin dirubah</li> <li>- Rubah data hasil pemeriksaan lapangan berupa atau komponen-komponen bangunan,</li> </ul>		Pengendalian dapat mengedit data pemeriksaan hasil lapangan yang sudah tersimpan sebelumnya pada database.	<ul style="list-style-type: none"> <li>- Data pada database terupdate</li> <li>- Pesan data pemeriksaan berhasil terupdate dan tombol preview Berita Acara Pemeriksaan</li> </ul>	

		<p>Guna Bangunan, dll</p> <ul style="list-style-type: none"> <li>- Klik tombol Simpan</li> </ul>				
5	Input data Penetapan.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai bendahara, Data Hasil Pemeriksaan sudah ada dalam database.</li> <li>- Klik <i>aktivitas pilih izin mendirikan bangunan</i> dan klik tab <i>penetapan</i></li> <li>- Isi nomor SKRD dan tentukan nilai dasar bangunan, lalu tekan tombol hitung</li> <li>- Tekan tombol Simpan</li> </ul>		<p>Bendahara dapat melakukan perhitungan dan memasukan data biaya penetapan ke dalam database.</p>	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan data berhasil disimpan dan tombol preview nota perhitungan dan SKRD</li> </ul>	
6	Edit data	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai</li> </ul>		<p>Bendahara dapat memasukan data biaya penetapan ke dalam</p>	<ul style="list-style-type: none"> <li>- Data tersimpan</li> </ul>	

	penetapan.	bendahara, Data Hasil Pemeriksaan sudah ada dalam database.  - Klik <i>Daftar Perijinan</i> pilih <i>izin mendirikan bangunan</i> dan klik tab <i>penetapan</i> dan pilih data yang ingin di edit  - Rubah nomor SKRD atau nilai dasar bangunan, lalu tekan tombol hitung  - Tekan tombol Simpan		database.	kedalam database  - Pesan data berhasil diupdate dan tombol preview nota perhitungan dan SKRD	
7	Input data pembayaran.	- <i>Telah Login</i> sebagai bendahara, Data Penetapan biaya sudah ada		Bendahara dapat memasukan data pembayaran retribusi ke dalam database.	- Data tersimpan kedalam database  - Pesan Data Berhasil	

		<p>dalam database.</p> <ul style="list-style-type: none"> <li>- Klik menu Pembayaran</li> <li>- Pilih nomor pendaftaran, masukan nama pembayar</li> <li>- Klik tombol Simpan</li> </ul>			<p>disimpan dan tombol preview bukti pembayaran</p>	
8	Input Data Penomoran.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai penomoran, data sudah sampai ke pembayaran.</li> <li>- Klik menu penomoran</li> <li>- Pilih nomor pendaftaran, masukan nomor SK</li> <li>- Klik tombol SImpan</li> </ul>		<p>Bagian penomoran dapat memasukan penomoran surat izin kedalam database.</p>	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan Data Penomoran Berhasil Disimpan</li> </ul>	
9	Input Data penyerahan.	<ul style="list-style-type: none"> <li>- Telah <i>Login</i> sebagai penyerahan, data sudah sampai ke penomoran.</li> </ul>		<p>Bagian penyerahan dapat memasukan data penyerahan surat izin kedalam database.</p>	<ul style="list-style-type: none"> <li>- Data tersimpan kedalam database</li> <li>- Pesan Bahwa Izin</li> </ul>	

		<ul style="list-style-type: none"><li>- Klik menu penyerahan</li><li>- Masukan nama dan nomor KTP pengambil ijin.</li><li>- Klik tombol Simpan</li></ul>			Telah Sampai ketahap Penyerahan ke Pemohon	
--	--	--	--	--	--	--

## **ACCEPTANCE TESTING**

### **Pengujian Penerimaan Aplikasi**

Nama User Penguji : .....

Jabatan : .....

1. Menurut anda, apakah aplikasi ini mudah dijalankan (*user friendly*) ?

a. Mudah .....

b. Cukup mudah, karena .....  
.....

c. Kurang mudah, karena .....  
.....

2. Menurut anda bila dilihat dari segi pemilihan warna *Layout*, jenis, maupun ukuran *Font*, bagaimana tampilan aplikasi ini secara keseluruhan?

a. Bagus

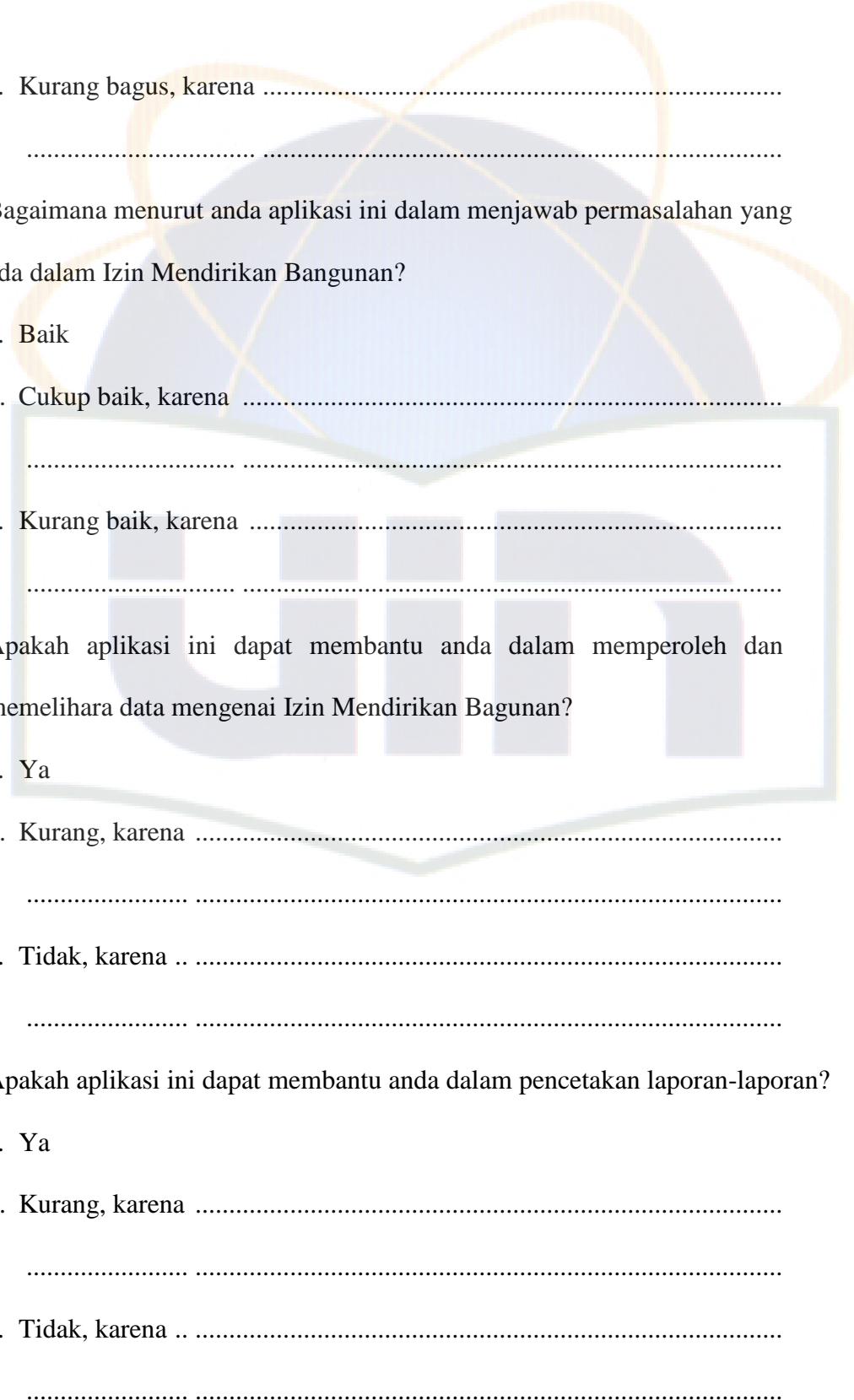
b. Cukup bagus, karena .....  
.....

c. Kurang bagus, karena .....  
.....

3. Apa pendapat anda setelah melihat aplikasi ini secara keseluruhan?

a. Bagus

b. Cukup bagus, karena .....  
.....

- 
- c. Kurang bagus, karena .....  
.....
4. Bagaimana menurut anda aplikasi ini dalam menjawab permasalahan yang ada dalam Izin Mendirikan Bangunan?
- a. Baik
- b. Cukup baik, karena .....  
.....
- c. Kurang baik, karena .....  
.....
5. Apakah aplikasi ini dapat membantu anda dalam memperoleh dan memelihara data mengenai Izin Mendirikan Bagunan?
- a. Ya
- b. Kurang, karena .....  
.....
- c. Tidak, karena .. .....  
.....
6. Apakah aplikasi ini dapat membantu anda dalam pencetakan laporan-laporan?
- a. Ya
- b. Kurang, karena .....  
.....
- c. Tidak, karena .. .....  
.....

7. Menurut anda, apakah aplikasi ini bila dikembangkan akan lebih membantu anda?

- a. Sangat membantu .....
- b. Kurang membantu, karena .....
- c. Tidak membantu, karena .....

Keterangan Penilaian:

Sangat Baik : 80-100

Baik : 70-80

Cukup : 60-70

Buruk : 50-60

Sangat Buruk : <50

Serpong Utara,

Score Jawaban:

Jawaban A : 15

Jawaban B : 10

Jawaban C : 7

Dalam pengumpulan data (*Collecting Data*), penulis melakukan wawancara dengan Bapak Drs. Iskandar sebagai Kepala Seksi Pelayanan Perijinan Bidang Pembangunan di Badan Pelayanan Perijinan Terpadu. Berikut beberapa petikan hasil wawancara yang dilakukan oleh penulis:

1. Apa tugas dan fungsi Badan Pelayanan Perijinan Terpadu?

Badan Pelayanan Perijinan Terpadu bertugas Melaksanakan koordinasi dan penyelenggaraan serta pelayanan administrasi dibidang perijinan dan non perijinan secara terpadu. Ada pun fungsinya salah satunya yaitu Penyelenggaraan Perijinan dan Pengawasan Dan Evaluasi Pelayanan Perijinan.

2. Bagaimanakah Alur Perijinan pengurusan Izin Mendirikan Bangunan yang saat ini sedang berjalan?

Alur pembuatan Ijin Mendirikan Bangunan dimulai dari pemohon mengajukan permohonan pembuatan IMB pada bagian pendaftaran dengan mengisi form pendaftaran dan melampirkan berkas-berkas persyaratan dibutuhkan. Berkas tersebut diverifikasi kembali oleh Kepala Seksi Bidang Pembangunan, Kemudian bagian Pengawasan dan Pengendalian melakukan pemeriksaan ke lapangan untuk menyesuaikan apakah data yang telah diajukan pemohon sesuai atau tidak. Kemudian berlanjut pada bagian Bendahara yang bertugas menghitung penetapan biaya retribusi yang harus dibayar oleh pemohon. Bendahara kemudian mencetak SKRD. Bukti SKRD ini kemudian diserahkan untuk pemohon dan pemohon harus membayar sesuai dengan jumlah yang tertera pada SKRD. Setelah biaya retribusi dibayarkan maka pemohon dapat mengambil IMB setelah izin dicetak dan diberi nomor.

3. Apakah sebelumnya sudah terkomputerisasi terutama pada pengurusan Ijin Mendirikan Bangunan?

Belum, pengerjaan masih menggunakan *Microsoft Word* atau di *Microsoft Excel*.

4. Apakah dengan aplikasi tersebut cukup membantu?

Cukup membantu membantu namun masih banyak kekurangan apalagi ketika masing-masing bagian diminta membuat report-report keseluruhan permohonan ijin, karena pengerjaan masih sendiri-sendiri.

5. Apakah sering terjadi ketidaksesuaian data pada masing-masing bagian khususnya IMB?

Sering, karena cukup banyak data yang didaftarkan perharinya untuk Ijin Mendirikan Bangunan (belum termasuk ijin lainnya), sehari bisa mencapai 30 pemohon. Untuk IMB sendiri, ijin yang paling banyak permohonan dalam sehari.

6. Apakah membutuhkan waktu yang lama dalam pembuatan Surat Ijin dan laporan-laporan lainnya?

Ya, karena masing-masing bagian tidak hanya mengelola IMB tetapi ijin-ijin yang lainnya belum lagi data pemohon terpisah (masih menggunakan MS. Word dan MS. Excel) sehingga membutuhkan waktu lama untuk pembuatan laporan-laporan.

7. Bagaimanakah cara menghitung nilai retribusi Ijin Mendirikan Bangunan?

IMB memiliki perbedaan perhitungan untuk setiap komponen-komponen bangunan, ada beberapa faktor yang mempengaruhi perhitungan seperti guna bangunan, jenis komponen bangunan, luas komponen bangunan Koefisien Tingkat Bangunan.

```
izin3_permohonan.php
<?php
    $xpath =
".".str_replace("\\", "/",
str_replace(realpath("."), "", dirname(__FILE__)))."/";
?>
<script type="text/javascript">
    $(document).ready(function()
{
    $('[@name="cpermohonan"]').focus();

    $('#blampiran').click(function()
{
        var
cekLampiran=
$('#blampiran').attr('checked')
;

        if(cekLampiran == true)
{
            openBerkas('form1');
        }
    });

    $('#cjabatan').keypress(function(event)
{
        var
selectedItem= $(this).val();
        if
(event.keyCode == 9 &&
selectedItem !== null) {

```

```
// User
pressed enter key.

removeClass();
YAHOO.util.Dom.addClass('komponen', 'selected');

var
tabView = new
YAHOO.widget.TabView('demo');

document.getElementById('cperuntukan').focus();
}

});

// User
pressed enter key.

$('#cluasbangunan').keypress(function(event) {
    var
selectedItem= $(this).val();
    if
(event.keyCode == 9 &&
selectedItem !== null) {
        // User
pressed enter key.

removeClass();
YAHOO.util.Dom.addClass('lokasi_bangunan', 'selected');

var
tabView = new
YAHOO.widget.TabView('demo');

document.getElementById('kampung').focus();
```

```
        }
    });

$( '#cbatastimur' ).keypres
s(function(event) {
    var
selectedItem= $(this).val();
    if
(event.keyCode == 9 &&
selectedItem !== null) {

        tombolSimpan(this.form);
    }
});

$( '#cpanjangbangunan' ).ke
yup(function() {
    var panjang=
$(this).val();
    var lebar=
$( '#clebarbangunan' ).val();
    if (lebar != ''
) {
        var
luas= panjang * lebar;
        $ ("#cluasbangunan" ).val(l
uas);
    }
});

$( '#clebarbangunan' ).keyu
p(function() {
```

```
        var panjang=
$(this).val();
        var lebar=
$('#cpanjangbangunan').val();
        if (lebar != ''
) {
            var
luas= panjang * lebar;
            $('#cluasbangunan').val(luas);
}
        $('#jenis').change(function() {
            var value=
$(this).val();
            munculBerkas(value);
})
        $('[@name="cpermohonan"]').blur(function(){
            var pEdit=
$('#pEdit').val();
            var nomor=
$(this).val();
            if(pEdit=='') { //begin if
$.ajax({
            type: "POST",
            url:
"<?=$xpath?>cekNoPendaftaran.php",
            data: "nomor="+nomor,
            success: function(data){
if(parseInt(data) > 0){
                $('#errorMessage').css('display','block');
                $('#errorMessage').html('
<span
class="headerMessage">Pesan
Error</span> :<br><span
class="headerMessage">NOMOR
PENDAFTARAN</span> Sudah Ada');
}
            }
            //end if
        });
}
        var nilai=
$('#jenis').val();
        munculBerkas(nilai);
})
        function removeEnter(str)
{
    return
str.replace(/\n/g,"");
}
        function
munculBerkas(valu){
    if(valu != ''){
        if
(parseInt(valu) > 1) {
            $('#titleIMB').css('display','block');
            $('#fieldIMB').css('display','block');
        } else{
            $('#titleIMB').css('display','none');
            $('#fieldIMB').css('display','none');
        }
    }
}
//function openBerkas()
```

```
//  
gcms_open_form('<?=$expat  
h?>form1.php',' ',800,600)  
//}  
  
function  
openBerkas(fname){  
  
    gcms_open_form('<?=$expat  
h?>' + fname + '.php',' ',800,600)  
}  
  
function  
bukaTabPemohon(){  
    removeClass();  
    YAHOO.util.Dom.addClass('pemoho  
n','selected');  
    var tabView = new  
    YAHOO.widget.TabView('demo');  
  
    document.getElementById('  
cktp').focus();  
}  
  
function simpanData(){  
    $.ajax({  
        type: 'POST',  
        url:  
'<?=$expath?>proses.php',  
        data:  
"dpermohonan+=" + $('#name="dperm  
ohonan"]').val() +  
  
"cpermohonan+=" + $('#name  
="cpermohonan"]').val() +  
  
"blampiran+=" + $('#name="blampiran  
").val() +  
"cperihal+=" + $('#name="c  
perihal"]').val() +  
"cktp+=" + $('#name="cktp  
').val() +  
"cnama+=" + $('#name="cnam  
a"]').val() +  
"clahir+=" + $('#name="cla  
hir"]').val() +  
"dlahir+=" + $('#name="dla  
hir"]').val() +  
"cpekerjaan+=" + $('#name="c  
pekerjaan"]').val() +  
"alamat+=" + $('#name="ca  
lamat"]').val() +  
"cjabatan+=" + $('#name="c  
jabatan"]').val() +  
"cnpwd+=" + $('#name="cnpw  
d"]').val() +  
"cpondasi+=" + $('#name="c  
pondasi"]').val() +  
"kusen+=" + $('#name="cran  
gkabangunan"]').val() +  
"cdinding+=" + $('#name="c  
dinding"]').val() +  
"crangkaatap+=" + $('#name  
="crangkaatap"]').val() +  
"cpenuutupatap+=" + $('#name  
="cpenuutupatap"]').val() +  
"cluasbangunan+=" + $('#name  
="cluasbangunan"]').val() +  
"cletakbangunan+=" + $('#name  
="cletakbangunan"]').val() +  
"csertifikatpersil+=" + $('#  
name="csertifikatpersil"]').v  
al() +  
"jenis+=" + $('#name="jeni  
s"]').val() +  
"nid_login+=" + $('#name="ni  
d_login"]').val() +  
"ciddesa+=" + $('#name="ci  
ddesa"]').val() +  
"clantai+=" + $('#name="cl  
antai"]').val() +
```

```
"&kampung)+"+$('@name="kampung"]').val()+
"&atastanah)+"+$('@name="atastanah"]').val()+
"&cperuntukan)+"+$('@name="cperuntukan"]').val()+
"&pemohon_hp)+"+$('@name="pemohon_hp"]').val()+
"&berkas)+"+$('@name="berkas"]').val()+
"&rekom)+"+$('@name="rekom"]').val()+
"&imb_lama)+"+$('@name="imb_lama"]').val()+
"&no_telp)+"+$('@name="no_telp"]').val()+
"&nama_cp)+"+$('@name="nama_cp"]').val()+
"&cupdate)+"+$('@name="cupdate"]').val()+
"&nmrnmr)+"+$('@name="nrmr"]').val()+
"&pTampungNomorID)+"+$('@
```



```
name="pTampungNomorID"]').val()+
+ ds + '<font size="3">' +expl[2] +'</font></b></td>' +
"&tab=permohonan",
success:
function(data) {
    if(data!='kosong') {
        var expl=
data.split('|');
        var ds = '';
        if(expl[2]=='Nomor Pendaftaran Tidak Sesuai Format')
            ds = '';
        else
            ds = 'Pemohonan berhasil dibuat dengan Nomor Pendaftaran : ';
        $('#tombol').html('');
        $('#demo').html('<form><div id="demo" class="yui-navset"><div class="yui-content">' +
'<table><tr>' +
'<td><br><b>' +
'<span id="pushbuttonPreview" class="yui-button yui-push-button"><em class="first-child"><button name="cetak" type="button" onclick="javascript:printReport('daftar',''+expl[1]','', '03');\\" value="Preview">Preview</button></em></span>' +
'</td><br>' +
'</tr><tr>' +
'<td><br>' +
'"<span id="pushbuttonPreview" class="yui-button yui-push-button"><em class="first-child"><button name="cetak" type="button" onclick="javascript:printReport('daftar',''+expl[1]','', '03');\\" value="Preview">Preview</button></em></span>' +
'</td><br>' +
'</tr></table></div></div></form>');
        pesanData(expl[2]);
        $.ajax({
            type: 'POST',
            url:
'<?=$expath?>trackIzin.php',
            data:

```

```

'id_izin='+expl[1]+'&tipe_izin=
3&from=layan&to=kelola&sent=<?=
date("Y-m-d, H:i:s")?>' 

    })
}

    });

function
pesanData(noPend){
    var chatuser=
'kelola';
    var pesan= 'Izin
'+noPend+' telah di proses pada
<?=date("h:i:s A")?>';

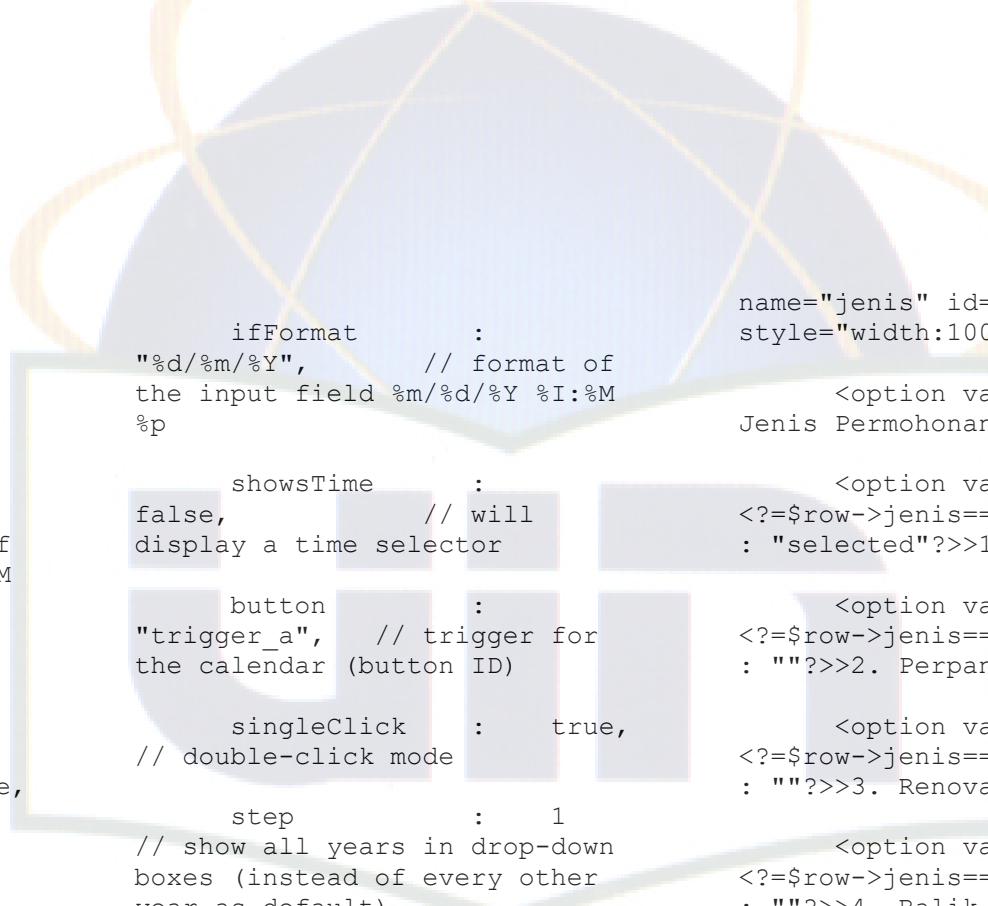
    $.post(""+pathChat+"chat.
php?action=sendchat", {to:
chatuser, message: pesan});
}
</script>
<?
if(basename( __FILE__
)==basename(
$_SERVER['PHP_SELF'])) die();

$success=false;
if(!$success){
    if($_GET['edit']=='1'){
        $vsq1 = "select *
from m03_01 where
id_03_01=".$_GET['id'];
        $rest =
gcms_query($vsq1);
        $row =
gcms_fetch_object($rest);
    }
    <form name="izin3"
method="post"
action="<?=menu_urlpage()?>&amp
;cpagesub=<?=$_GET['cpagesub']?>">
<input name="cupdate"
type="hidden" value="<?=$row-
>id_03_01?>"/>
<div id="demo" class="yui-
navset">
    <ul class="yui-nav">
        <li id="permohonan"
class="selected"><a
href="#tab1"><em>Permohonan</em>
</a></li>
        <li id="pemohon"><a
href="#tab2"><em>Pemohon</em></
a></li>
        <li id="komponen"><a
href="#tab3"><em>Komponen</em><
/a></li>
        <li
id="lokasi_bangunan"><a
href="#tab4"><em>Lokasi
Bangunan</em></a></li>
    </ul>
    <div class="yui-content">
        <div id="tab1">
            <table>
                <tr>
                    <td
colspan="2" style="text-align:center;
font-weight:bold;">Permohonan<
/h1></td>
                </tr>
                <tr>
                    <td
width="150" style="vertical-align:top;">Tgl.
Permohonan </td>
                    <td
width="280" style="vertical-align:top;">
<?php $date= date('d-m-
Y'); ?>
<input type="text"
id="date_permohonan"
name="dpermohonan"
value="<?!=empty($row->tanggal)
? formatDate($row->tanggal) :
date('d/m/Y')?>" size="8"
maxlength="10"
onKeyUp="fdate(this)"
onBlur="validateDate(this)"
title="Tanggal Permohonan-
Permohonan">
<a id="trigger_a"></a>
<script
type="text/javascript">

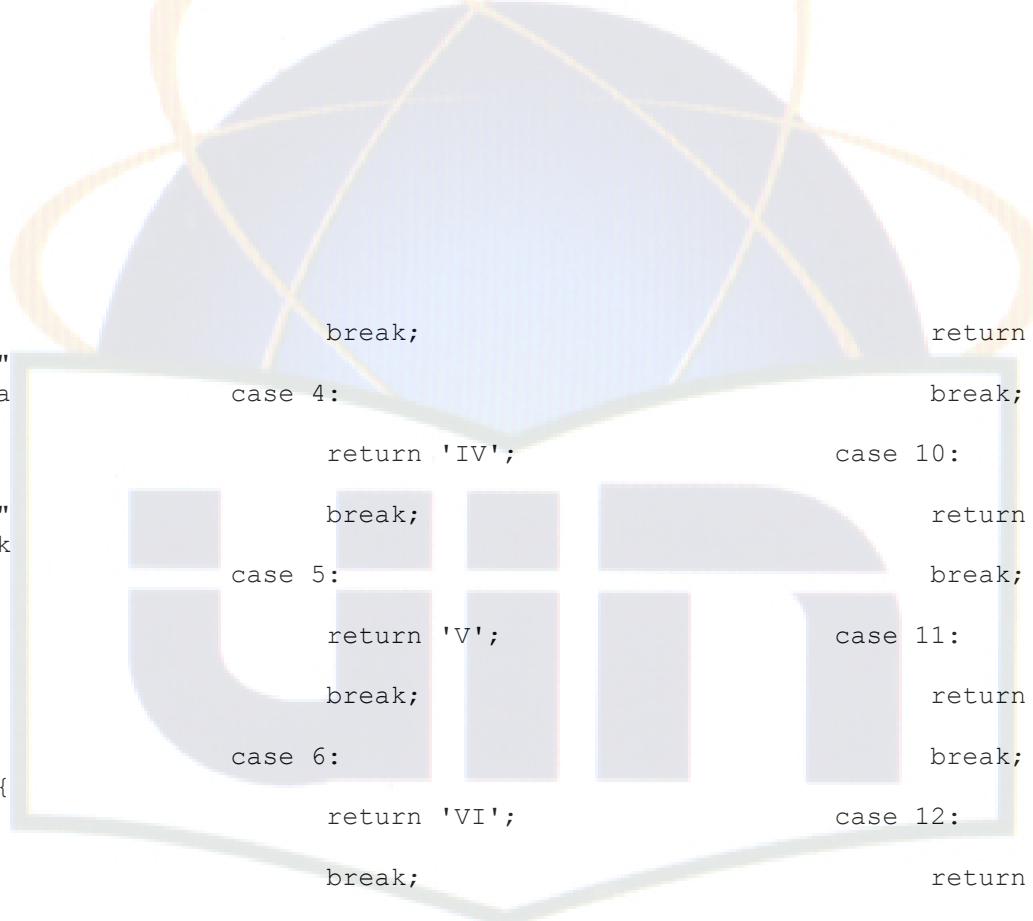
```

```
Calendar.setup({  
    inputField : "date_permohonan", // id  
    of the input field  
  
    ifFormat : "%d/%m/%Y", // format of  
    the input field %m/%d/%Y %I:%M  
    %p  
  
    showsTime : false, // will  
    display a time selector  
  
    singleClick : true,  
    // double-click mode  
  
    eventName : "focus",  
  
    step : 1  
    // show all years in drop-down  
    boxes (instead of every other  
    year as default)  
});  
  
Calendar.setup({  
    inputField : "date_permohonan", // id  
    of the input field
```

```
    ifFormat : "%d/%m/%Y", // format of  
    the input field %m/%d/%Y %I:%M  
    %p  
  
    showsTime : false, // will  
    display a time selector  
  
    button : "trigger_a", // trigger for  
    the calendar (button ID)  
  
    singleClick : true,  
    // double-click mode  
  
    step : 1  
    // show all years in drop-down  
    boxes (instead of every other  
    year as default)  
});  
  
</script>  
</td>  
</tr>  
  
<td>Jenis Permohonan</td>  
<td>  
    <select title="Jenis  
    Permohonan-Permohonan"
```



```
name="jenis" id="jenis"  
style="width:100%">  
  
    <option value=''>Pilih  
    Jenis Permohonan</option>  
  
    <option value="1"  
    <?=$row->jenis==1 ? "selected"  
    : "selected"?>>1. Baru</option>  
  
    <option value="8"  
    <?=$row->jenis==8 ? "selected"  
    : ""?>>2. Perpanjangan</option>  
  
    <option value="2"  
    <?=$row->jenis==2 ? "selected"  
    : ""?>>3. Renovasi</option>  
  
    <option value="3"  
    <?=$row->jenis==3 ? "selected"  
    : ""?>>4. Balik Nama</option>  
  
    <option value="4"  
    <?=$row->jenis==4 ? "selected"  
    : ""?>>5. Penambahan</option>  
  
    <option value="5"  
    <?=$row->jenis==5 ? "selected"  
    : ""?>>6. Balik Nama &  
    Penambahan</option>  
  
    <option value="6"  
    <?=$row->jenis==6 ? "selected"  
    : ""?>>7. Renovasi &  
    Penambahan</option>
```



```
<option value="7"
<?=$row->jenis==7 ? "selected"
: ""?>>8. Renovasi, Balik Nama
& Penambahan</option>

<option value="9"
<?=$row->jenis==9 ? "selected"
: ""?>>9. Renovasi & Balik
Nama</option>

</select>
      </td>
    </tr>
    <?php

function romawi($input) {
  switch(intVal($input)) {
    case 1:
      return 'I';
      break;
    case 2:
      return 'II';
      break;
    case 3:
      return 'III';
      break;
    case 4:
      return 'IV';
      break;
    case 5:
      return 'V';
      break;
    case 6:
      return 'VI';
      break;
    case 7:
      return 'VII';
      break;
    case 8:
      return 'VIII';
      break;
    case 9:
      return 'IX';
      break;
    case 10:
      return 'X';
      break;
    case 11:
      return 'XI';
      break;
    case 12:
      return 'XII';
      break;
  }
}

$sqlJumData= "SELECT
COUNT(*) as jumbaris FROM
M03_01";
$resultJumData=
gcms_query($sqlJumData);

$arrJumData=
```

```

gcms_fetch_object($resultJumData);
if($arrJumData->jumbaris > 0){
    $sqlNoP= "SELECT NOMOR
    FROM M03_01 WHERE
    ID_03_01=(SELECT MAX(ID_03_01)
    FROM M03_01 WHERE NOMOR_ID LIKE
    '%".date("y").'')";
    $resultNoP=
    gcms_query($sqlNoP);
    $arrNoP=
    gcms_fetch_object($resultNoP);
    $noP= $arrNoP->NOMOR;
    if(!empty($row->nomor)){
        $noPermohonan= $row->nomor;
    }else{
        $expl= explode('/', $noP);
        $nomor1=
        trim($expl[0])+1;
        $noPermohonan=
        $nomor1.'/IMB-

```

```

        Bid.Pel/.romawi(date('m')). '/'
        .date('Y'));
    }
    }else{
        $noPermohonan= '160/IMB-
        Bid.Pel/.romawi(date('m')). '/'
        .date('Y'));
    }
    ?>
    <tr>
        <td>No.
        Permohonan </td>
        <td><input title="Nomor
        Permohonan-Permohonan"
        type="text" name="cpermohonan"
        style="width:100%"
        value=<?=$noPermohonan?>>/></td>
    </tr>
    <tr>
        <td>Perihal </td>
        <td><input
        title="Perihal-Permohonan"
        type="text" name="cperihal"
        style="width:100%"
        value=<?=!empty($row->perihal)
        ? $row->perihal : "Permohonan
        Ijin Mendirikan Bangunan
        (IMB)"?>>/></td>
    </tr>
    <tr>
        <td>Lampiran
        Status </td>
        <td><input
        title="Lampiran Status-
        Permohonan" id="blampiran"
        name="blampiran"
        type="checkbox" value="1"
        <?=$row->status==1 ? "checked"
        : ""?>/>&ampnbsp<a
        onclick="javascript:openBerkas(
        'form1');"
        style="cursor:pointer;"></a></td>
    </tr>
</table>
</div>
<div id="tab2">
<Table>
<tr>
        <td colspan="4"><br
        /><h1>Pemohon</h1></td>
    </tr>
    <tr>
        <td>No.
        KTP</td>
        <td><input title="Nomor
        KTP-Pemohon" type="text"
        id="cktp" name="cktp"
        value=<?=$row-
        >pemohon_no?>>/></td>

```

<td>&nbsp;Nama&nbsp;</td>	<td><input title="Nama-Pemohon" type="text" name="cnama" value="<?=\$row->pemohon_nama?"/></td>	<td><input type="text" name="pemohon_telp" value="<?=\$row->pemohon_telp?"/></td>	<td><input type="text" name="pemohon_hp" value="<?=\$row->pemohon_hp?"/></td>	<td><input type="text" name="pemohon_tempat" value="<?=\$row->pemohon_tempat?"/></td>	<td><input type="text" id="date_lahir" name="dlahir" value="<?!=empty(\$row->pemohon_ttl) ? formatDate(\$row->pemohon_ttl) : \$_POST['dlahir']?>" size="8" maxlength="10" onKeyUp="fdate(this)" onBlur="validateDate(this)" title="Tanggal Lahir-Pemohon">
<td>NPWRD</td>	<td><input type="text" name="cnpwd" title="NPWRD-Pemohon" value="<?=\$row->pemohon_npwd?"/></td>	<td>&nbsp;No. HP&nbsp;</td>	<td><input type="text" name="pemohon_hp" title="Nomor Handphone-Pemohon" value="<?=\$row->pemohon_hp?"/></td>	<td>Tempat Lahir&nbsp;</td>	<a id="trigger_b"></a>
<td>Contact Person</td>	<td><input type="text" name="nama_cp" title="Contact Person-Pemohon" value="<?=\$row->nama_cp?"/></td>	<td><input title="Tempat Lahir-Pemohon" type="text" name="clahir" value="<?=\$row->pemohon_tempat?"/></td>	<td align="right">&nbsp;Tanggal Lahir&nbsp;</td>	<td><?php \$date= date('d-m-Y');?>	<script type="text/javascript">
	<!-- onKeyUp="number(this,event)"-->	<td><input type="text" id="date_lahir" name="dlahir" value="<?!=empty(\$row->pemohon_ttl) ? formatDate(\$row->pemohon_ttl) : \$_POST['dlahir']?>" size="8"	<td align="right">&nbsp;</td>		Calendar.setup({
	<tr>				inputField : "date_lahir", // id of the input field
					ifFormat : "%d/%m/%Y", // format of the input field %m/%d/%Y %I:%M %p
					showsTime : false, // will display a time selector
					button : "trigger_b", // trigger for the calendar (button ID)
					singleClick : true, // double-click mode

```

step : 1
// show all years in drop-down
boxes (instead of every other
year as default)

});

</script>
</td>
</tr>
<tr>
<td
valign="top">Alamat</td>
<td
colspan="3"><textarea
title="Alamat-Pemohon"
name="alamat"
style="width:100%"><?=$row-
>pemohon_alamat?></textarea><t
d>
</tr>
<tr>

<td>Nama
Perusahaan&ampnbsp</td>

<td><input title="Nama
Perusahaan-Pemohon" type="text"
name="cpekerjaan"
value="<?=$row->pemohon_job?>">
/></td>

<td>&ampnbspJabatan&ampnbsp</
td>
<td>
<script>
step : 1
// show all years in drop-down
boxes (instead of every other
year as default)

};

</script>
</td>
</tr>
<tr>
<td
valign="top">Alamat</td>
<td
colspan="3"><input
title="Jabatan" type="text"
id="cjabatan" name="cjabatan"
value="<?=$row-
>pemohon_jabatan?>"></td>
</tr>
<tr>
<td
colspan="4"><br /></td>
<td
colspan="4">&ampnbsp
</td>
</tr>
<td
colspan="4"><?php
$sqlBahan= "SELECT * FROM
MSETUP3_6";
$resultBahan=
gcms_query($sqlBahan);
$arrBahan=
gcms_fetch_object($resultBahan)
;
$peruntukan=
explode(',',$arrBahan-
>PERUNTUKAN);
$pondasi=
explode(',',$arrBahan-
>PONDASI);
$dinding=
explode(',',$arrBahan-
>DINDING);
$rangka_atap=
explode(',',$arrBahan->RANGKA);
$atap=
explode(',',$arrBahan->ATAP);
$kusen=
explode(',',$arrBahan->KUSEN);
$lantai=
explode(',',$arrBahan->LANTAI);
?>
<div id="tab3">
<Table>
<tr>
<td
colspan="4"><br /><h1>Komponen
Bangunan</h1></td>
</tr>
<tr>
<td>Peruntukan
Bangunan&ampnbsp</td>
<td>
<input type="text"
name="cperuntukan"
id="cperuntukan"
title="Peruntukan Bangunan"
value="<?=$row-
>peruntukan?>"/>&ampnbsp<a
onclick="javascript:openBerkas(
'form2');"

```

```

        style="cursor:pointer;"></a>
      </td>
    </tr>
    <tr>

      <td>Pondasi </td>
      <td>

        <input type="text"
name="cpondasi" id="cpondasi"
title="Pondasi-Komponen
Bangunan" value=<?=$row-
>pondasi?>/>&nbsp;<a
onclick="javascript:openBerkas(
'form_pondasi');"
style="cursor:pointer;"></a>

      <!--<select id="cpondasi"
name="cpondasi" title="Pondasi-
Komponen Bangunan"
style="width:100%">

        <option value="">-- Pilih
Pondasi --</option>

      <?php foreach($pondasi as
$keyBahan=>$valueBahan) {

        trim($row->pondasi)
== trim($valueBahan) ?
$selected="SELECTED" :
$selected= "";

```

'). The code uses PHP variables like \$row, \$keyBahan, and \$valueBahan, and HTML entities like '<?php' and '<?='."/>

```

        ?>
      <option value=<?=
$valueBahan; ?>">
<?=$selected;?>><?=
$valueBahan; ?></option>
<?php } ?>
</select>-->
      <td>&nbsp;Dinding&nbsp;</td>
      <td>

        <input type="text"
name="cdinding" id="cdinding"
title="Dinding-Komponen
Bangunan" value=<?=$row-
>dinding?>/>&nbsp;<a
onclick="javascript:openBerkas(
'form_dinding');"
style="cursor:pointer;"></a>

      <!--<select
name="cdinding" title="Dinding-
Komponen Bangunan"
style="width:100%">

        <option value="">-- Pilih
Dinding --</option>

```

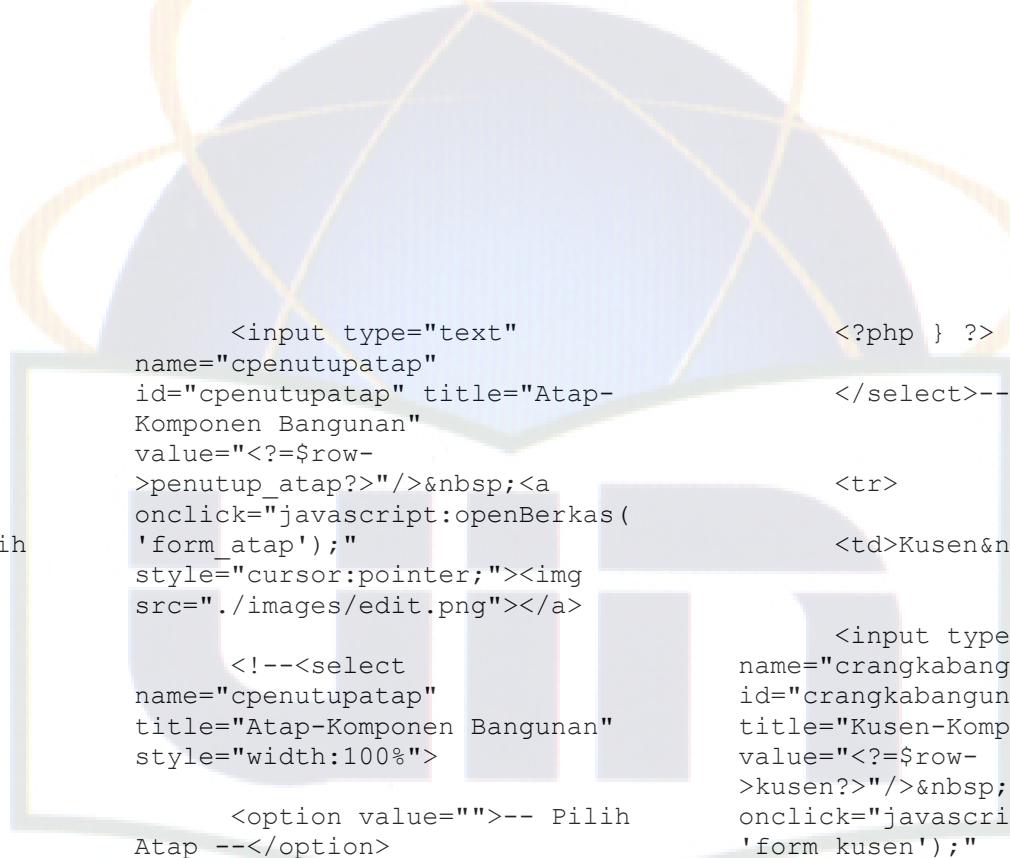
```

        <?php foreach($dinding as
$keyBahan=>$valueBahan) {
          trim($row->dinding)
== trim($valueBahan) ?
$selected="SELECTED" :
$selected= "";
        ?>
      <option value=<?=
$valueBahan; ?>">
<?=$selected;?>><?=
$valueBahan; ?></option>
<?php } ?>
</select>-->
      </td>
    </tr>
    <tr>

      <td>Rangka Kap </td>
      <td>

        <input type="text"
name="crangkaatap"
id="crangkaatap" title="Rangka
Atap-Komponen Bangunan"
value=<?=$row-
>rangka_atap?>/>&nbsp;<a
onclick="javascript:openBerkas(
'form_rangka');"

```



```

style="cursor:pointer;"></a>

<!--<select
name="crangkaatap"
title="Rangka Atap-Komponen
Bangunan" style="width:100%">

<option value="">-- Pilih
Rangka Kap --</option>

<?php
foreach($rangka_atap as
$keyBahan=>$valueBahan) {

    trim($row-
>rangka_atap) ==
trim($valueBahan) ?
$selected="SELECTED" :
$selected= "";

?>

<option value="<?=
$valueBahan; ?>">
<?=$selected;?>><?=
$valueBahan; ?></option>

<?php } ?>

</select>-->
</td>

<td>&nbsp;Atap&nbsp;</td>
<td>

```

```

<input type="text"
name="cpenutupatap"
id="cpenutupatap" title="Atap-
Komponen Bangunan"
value="<?=$row-
>penutup_atap?>"/>&nbsp;<a
onclick="javascript:openBerkas(
'form_atap');"
style="cursor:pointer;"></a>

<!--<select
name="cpenutupatap"
title="Atap-Komponen Bangunan"
style="width:100%">

<option value="">-- Pilih
Atap --</option>

<?php foreach($atap as
$keyBahan=>$valueBahan) {

    trim($row-
>penutup_atap) ==
trim($valueBahan) ?
$selected="SELECTED" :
$selected= "";

?>

<option value="<?=
$valueBahan; ?>">
<?=$selected;?>><?=
$valueBahan; ?></option>

```

```

<?php } ?>

</select>-->
</td>

<td>Kusen&nbsp;</td>
<td>

<input type="text"
name="crangkabangunan"
id="crangkabangunan"
title="Kusen-Komponen Bangunan"
value="<?=$row-
>kusen?>"/>&nbsp;<a
onclick="javascript:openBerkas(
'form_kusen');"
style="cursor:pointer;"></a>

<!--<select
name="crangkabangunan"
title="Kusen-Komponen Bangunan"
style="width:100%">

<option value="">-- Pilih
Kusen --</option>

<?php foreach($kusen as
$keyBahan=>$valueBahan) {

    trim($row->kusen)
== trim($valueBahan) ?

```

```

$selected="SELECTED" :
$selected= "";
?>

<option value=<?
$valueBahan; ?>
<?=$selected;?><?=
$valueBahan; ?></option>

<?php } ?>

</select>-->
</td>

<td>&ampnbspLantai&ampnbsp</t
d>
<td>

<input type="text"
name="clantai" id="clantai"
title="Lantai-Komponen
Bangunan" value=<?=$row-
>lantai?>/>&ampnbsp<a
onclick="javascript:openBerkas(
'form_lantai');"
style="cursor:pointer;"></a>

<!--<select
name="clantai" title="Lantai-
Komponen Bangunan"
style="width:100%">

```

<option value="">-- Pilih  
Lantai --</option>

<?php foreach(\$lantai as  
\$keyBahan=>\$valueBahan) {

    trim(\$row->lantai)  
== trim(\$valueBahan) ?  
\$selected="SELECTED" :  
\$selected= "";

?>

    <option value=<?
\$valueBahan; ?>>
<?=\$selected;?><?=
\$valueBahan; ?></option>

<?php } ?>

</select>-->

<td>Luas  
Bangunan&ampnbsp</td>

<td><input type="text"
id="cluasbangunan"
name="cluasbangunan"
onKeyUp="number(this,event)"
title="Luas Bangunan-Komponen"
value=<?=\$row-
>kampung?>/></td>

<td><br /><div id="tab4">
<Table>
<tr>

<td colspan="4"><br /></td>
</tr>

<td colspan="2"><br /><h1>Lokasi</h1></td>
</tr>

<td colspan="2" width="150">Kampung/ Komplek/
Perumahan&ampnbsp</td>

<td width="250"><input type="text"
id="kampung" name="kampung"
style="width:100%"
title="Kampung-Komponen"
value=<?=\$row-
>kampung?>/></td>

<tr>

<td align="top">Jalan&ampnbsp</td>
<td>

<textarea>

```

>luas_bangunan?>" style="text-
align:right;"/> m2</td>
</tr>

<tr>
<td colspan="4"><br /></td>
</tr>

</table>
</div>
<div id="tab4">
<Table>
<tr>
<td colspan="2"><br /><h1>Lokasi</h1></td>
</tr>
<td colspan="2" width="150">Kampung/ Komplek/
Perumahan&ampnbsp</td>
<td width="250"><input type="text"
id="kampung" name="kampung"
style="width:100%"
title="Kampung-Komponen"
value=<?=$row-
>kampung?>/></td>
</tr>

<tr>
<td align="top">Jalan&ampnbsp</td>
<td>
```

```

        id="cletakbangunan"
        name="cletakbangunan"
        style="width:100%" title="Letak
Bangunan-Komponen"><?=$row-
>letak_bangunan?></textarea>
      </td>
    </tr>
    <?php

    if($_GET['edit']==1){

      $csq1 = "select * from
master_desa where
id_desa='".$row->id_desa."';

      $nresult=
gcms_query($csq1);

      $arr=
gcms_fetch_object($nresult);
    }
    ?>
    <tr>

      <td>Kecamatan&ampnbsp;</td>
      <td>
        <select
id="idKecamatan"
style="width:100%"
onchange="ajaxdo_trigDesa(this.
value)">

          <option value="">Pilih
Kecamatan</option>

```

```

        <?
      $csq1 =
"select * from master_kecamatan
order by kode_kecamatan asc";
      $nresult=gcms_query($csq1
);
      while($rox =
gcms_fetch_object($nresult)) {
        ?>
        <option value=<?=$rox-
>kode_kecamatan?>"<?
      trim($arr-
>id_kecamatan)==trim($rox-
>kode_kecamatan) ? "selected" :
""?><?=$rox->kode_kecamatan."-
".$rox-
>nama_kecamatan?></option>;
        <? } ?>
      </select>
      </td>
    </tr>
    <tr>
      <td>Desa/Kelurahan&ampnbsp;</td>
      <td
id="fieldDesa">
<?php
      if($_GET['edit']==1){

        $qry= "select *
from master_desa where
id_kecamatan='".$arr-
>id_kecamatan." order by
kode_desa asc";
        $result=
gcms_query($qry);
      }
      <?>
      <select
title="Desa/Kelurahan-Lokasi
Bangunan" name="ciddesa"
style="width:100%">
      <?php
        while($rox =
gcms_fetch_object($result)) {
          ?>
          <option
value=<?=$rox->id_desa?>"<?
      trim($arr-
>id_desa==$rox->id_desa
? "selected" : "")?><?=$rox-
>kode_desa." - ".$rox-
>nama_desa?></option>;
        <? } ?>
      </select>
      <?php } else{ ?>

```

```

        <select
title="Desa/Kelurahan-Lokasi
Bangunan" name="ciddesa"
style="width:100%">

        <option
value="">Pilih
Desa/Kelurahan</option>

    </select>
    <?php } ?>
        </td>
    <tr>
        <td>Berada diatas
Tanah </td>

        <td><input title="Berada
diatas Tanah-Lokasi Bangunan"
type="text" name="atastanah"
style="width:100%"
value=<?=$row-
>atastanah?>"></td>
        </tr>
    <tr>

        <td>Nomor </td>

        <td><input title="Nomor-
Lokasi Bangunan" type="text"
name="csertifikatpersil"
style="width:100%">

```

```

            value=<?=$row-
>sertifikat?>"></td>
        </tr>
        <tr>

            <td><span id="titleIMB"
style="display:none;">IMB
Lama  </span></td>
            <td><span id="fieldIMB"
style="display:none;"><textarea
name="imb_lama"
style="width:100%"><?=$row-
>imb_lama?></textarea></span></
td>
        </tr>
        <tr>
            <td
colspan="2"><br /></td>
        </tr>
    </table>
    </div>
</div>
<div id="tombol">
    <table>
        <tr>
            <td><br />
                <span
id="pushbutton5" class="yui-
button yui-push-button"><em
class="first-child"><button
type="button"
onclick="tombolSimpan(this.form
)">Simpan</button></em></span><
!--

```

```

            onclick="tombolSimpan(this.form
)"-->
            <span
id="pushbutton6" class="yui-
button yui-push-button"><em
class="first-child"><button
type="reset">Reset</button></em
></span>
        </tr>
    </table>
    <input name="csubmit"
type="hidden" value="Simpan"/>
    <!--<input name="noId"
type="hidden"
value=<?=$no_id?>" />-->
    <input
name="nid_login" type="hidden"
value=<?php echo
$_SESSION['nid_login']; ?>" />
    <input
type="hidden" name="berkas"
id="berkas" value=<?=$row-
>berkas?>"/>
    <input
type="hidden" name="rekom"
id="rekom" value=<?=$row-
>rekom?>"/>
    <input
type="hidden" name="pEdit"
id="pEdit"
value=<?=$_GET['edit']?>"/>
    <input
type="hidden" name="pTampungNomorID"

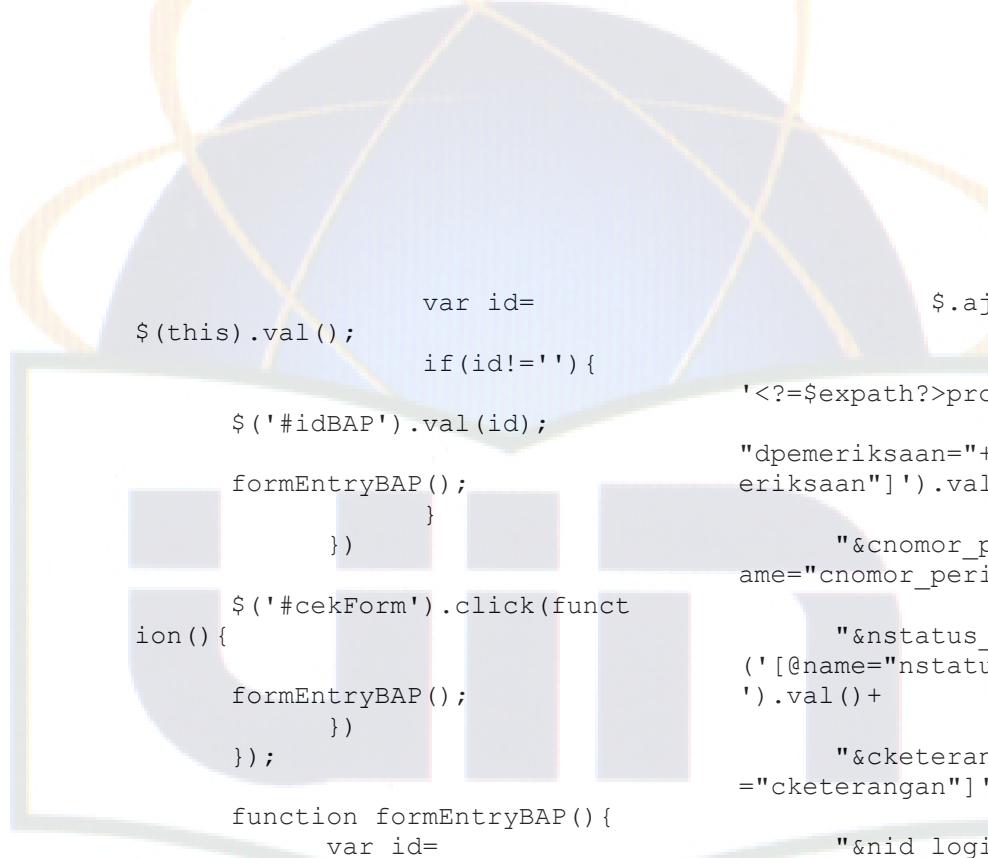
```

```
id="pTampungNomorID"
value="<?=$row->nomor_id?"/>
<input
type="hidden" name="nmrnmr"
id="nmrnmr" value="<?=$row-
>nmr?"/>
</div>
</form>
<script>
(function() {
    var tabView = new
YAHOO.widget.TabView('demo');
})();
</script>
<?
}
?>
```

### Izin3\_pemeriksaan.php

```
<?php
    $xpath =
".".str_replace("\\\\", "/",
str_replace(realpath("."), "", dirname(__FILE__)))."/";
?>

<script type="text/javascript">
    $(document).ready(function() {
        $('#cidentitas').change(function() {
            // alert('oke');idBAP
        })
    })
}
```



```
var id=
$(this).val();
if(id!=''){
    $('#idBAP').val(id);
    formEntryBAP();
}
}

$('#cekForm').click(function() {
    formEntryBAP();
});

function formEntryBAP() {
    var id=
    $('#idBAP').val();
    GB_show("Data
Pemeriksaan
Lapangan",'<?=$xpath?>form.php
?id='+id,550,750);
}

function formLokasi() {
//         var id=
    $('#idBAP').val();
    GB_show("Edit BAP
Lokasi",'<?=$xpath?>form4.php'
,650,750);
}

function simpanData() {
$.ajax({
    type: 'POST',
    url:
'<?=$xpath?>proses.php',
    data:
"dpemeriksaan="+$('[@name="dpem
eriksaan"]').val()+
"&cnomor_periksa="+$('[@n
ame="cnomor_periksa"]').val()+
"&nstatus_pemeriksaan="+$(
'[@name="nstatus_pemeriksaan"]
').val()+
"&cketerangan="+$('[@name
="cketerangan"]').val()+
"&nid_login="+$('[@name="nid
_login"]').val()+
"&cnama_petugas="+$('[@na
me="cnama_petugas"]').val()+
// "&cnip="+$('[@name="cni
p"]').val()+
// "&cjabatan="+$('[@name="c
jabatan"]').val()+
"&cidentitas="+$('[@name=
"cidentitas"]').val()+
});
```

```
"&editaj="+$('@name="editaj"]').val()+
    "&tab=pemeriksaan",
    success:
function(data) {
    if(data!='kosong') {
        var expl=
data.split('|');
        $('#tomboll').html('');
        $('#demo').html('<form><div id="demo" class="yui-navset"><div class="yui-content">' +
            '<table><tr>' +
                '<td><br><b>Pemeriksaan permohonan berhasil dibuat dengan status ke '+expl[1]+' dan Nomor Pemeriksaan : <font size="+3">'+expl[2]+'</font></b></td>' +
            '</tr><tr>' +
                '<td><br>' +
                    "<span
```

```
id=\"pushbuttonPreview\""
class=\"yui-button yui-push-
button\">><em class=\"first-
child\">><button name=\"cetak\"
type=\"button\"
onclick=\"javascript:printRepor-
t('bapl','"+expl[3]+"' , '03');\"
value=\"Preview\">>Preview</but-
ton></em></span>" +
```

' </td></tr>' +

' </table>' +

' </div></div></form>') ;

pesanData(expl[2]);

\$.ajax({

type: 'POST',

url:

'<?=\$expath?>trackIzin.php',

data:

'id\_izin='+expl[1]+'&tipe\_izin=
3&from=kelola&to=bendahara&sent=
=<?=date("Y-m-d, H:i:s")?>'

})

}

}

});

}

```

        function
pesanData(noPend) {
    var chatuser=
'bendahara';
    var pesan= 'Izin
'+noPend+' telah di periksa
pada <?=date("h:i:s A")?>';

    $.post(""+pathChat+"chat.
php?action=sendchat", {to:
chatuser, message: pesan});
}
</script>
<?php
if(basename( __FILE__ )
==basename(
$_SERVER['PHP_SELF'])) die();

$success=false;

if(!$success){
    if($_GET['edit']=='1') {
        $vsq1 = "select *
from m03_02 where
id_03_01=".$_GET['id'];
        $rest =
gcms_query($vsq1);
        $row =
gcms_fetch_object($rest);

        $key_date=b_fetch("select
tangal from m03_01 where
id_03_01=".$_row->id_03_01);
    }
}

```

```

?>
<form name="izin3_prs"
method="POST"
action=<?=menu_urlpage() ?>&cpa
gesub=<?=$_GET['cpagesub'] ?>">
<input name="cupdate"
type="hidden" value=<?=$row-
>id_03_02?>"/>
<input name="limit_start"
id="limit_start" type="hidden"
value=<?!=empty($key_date) ?
formatDate($key_date) : ""?>"/>
<input name="go_table"
id="go_table" type="hidden">
<div id="demo" class="yui-
navset">
    <div class="yui-content">
        <?php
            if($_GET['id']!=''){
                $sqlId=
"SELECT NOMOR_ID, ID_03_01 FROM
M03_01 WHERE
ID_03_01=". $_GET['id']. "";
                $resultId=
gcms_query($sqlId);
                $arrId=
gcms_fetch_object($resultId);
            }
        ?>
        <table>
            <tr>
                <td colspan="4"><h1>Pemeriksaan
Permohonan</h1></td>
            </tr>
            <tr>
                <td width="150">No.
Pendaftaran&ampnbsp</td>
                <td width="220">
                    <input
id="no_pendaftaran"
style="width:70%"
value=<?=$_GET['id'] != '' ?
$arrId->NOMOR_ID : ""?>
                    readonly="true">
                <?php if($_GET['id']=='')
{ ?>
                    &nbsp;') title="No.
Pendaftaran" >
                    <!--&nbsp;', getElementById('cidentita
s').value)" title="Cek Data" >-
                <?php } ?>
                    <!--width="20px"
height="20px"-->
                    <input type="hidden"
name="cidentitas"
id="cidentitas"
value=<?=$_GET['id'] != '' ?
$arrId->ID_03_01 : ""?>"/>
                <!--
name="cidentitas"
style="width:100%" title="Nomor
Pendaftaran" id="cidentitas"
onChange="ajaxdo_getDate(this.v
alue,1)">
                    <option
value="">Pilih Nomor
Pendaftaran</option>
                    <? if(!empty($row-
>id_03_02)) {
                        $xres=gcms_query("select
nomor_id,nomor from m03_01
where id_03_01=". $row-
>id_03_01 ." ORDER BY nomor_id,
nomor DESC");
                        $rox
=gcms_fetch_object($xres);
                        echo
                    }
                <?php
            }
        </table>
    </div>
</div>

```

```

"<option value='".$row->id_03_01'
selected>$rox-
>nomor_id</option>";
} else{
    $nresultuntuk=gcms_query(
    "select * from m03_01 where
    status_izin = '0' ORDER BY
    nomor_id, nomor DESC");
    while
    ($row =
    gcms_fetch_object($nresultuntuk
    )){ ?>
        <option
        value=<?=$row-
        >id_03_01?>><?=$row-
        >nomor_id?></option>;
    } ?>
        </select>-
    </td>
</tr>
<tr>
    <td>Tgl.
Pemeriksaan&ampnbsp</td>

```

```

<td>
    <?php $date= date('d-m-
Y') ;?>
        <input type="text"
        id="date_pemeriksaan"
        name="dpemeriksaan"
        value=<?!=empty($row->tanggal)
        ? formatDate($row->tanggal) :
        date('d/m/Y')?>" size="8"
        maxlength="10"
        onKeyUp="fdate(this)"
        onBlur="validateDate(this);chec
        k_limit_date(this,limit_start);
        " title="Tanggal Pemeriksaan">
            <a id="trigger_a"></a>
        <script
        type="text/javascript">
            Calendar.setup({
                inputField :
                "date_pemeriksaan", // id
                of the input field
                ifFormat :
                "%d/%m/%Y", // format of
                the input field %m/%d/%Y %I:%M
                %p
            });
            Calendar.setup({
                inputField :
                "date_pemeriksaan", // id
                of the input field
                ifFormat :
                "%d/%m/%Y", // format of
                the input field %m/%d/%Y %I:%M
                %p
            });
        </script>
    </td>

```

```

"trigger_a", // trigger for
the calendar (button ID)

singleClick : true,
// double-click mode

step : 1
// show all years in drop-down
boxes (instead of every other
year as default)

});

</script>
</td>
</tr>
<tr>
<td>No.
BA Pemeriksaan&ampnbsp</td>
<td
colspan="3"><input type="text"
name="cnomor_periksa"
style="width:100%" title="Nomor
Pemeriksaan" value="<?=$row-
nomor?>"></td>

</tr>
<?php

$sqlKoordinator= "SELECT
* FROM MASTER_PEJABAT WHERE
PEMERIKSAAN!=1 and status =
'1'";

```

`$resultKoordinator=`  
`gcms_query($sqlKoordinator);`  
`//`  
`$arrKoordinator=`  
`gcms_query($resultKoordinator);`  
`?>`  
`<tr>`  
`<td>Nama`  
`Petugas&ampnbsp</td>`  
`colspan="3">`  
`<select`  
`name="cnama_petugas"`  
`style="width:100%" title="Nama`  
`Petugas">`  
`<option value="">-- Pilih`  
`Koordinator --</option>`  
`<?php`  
`while($rowKoordinator=`  
`gcms_fetch_object($resultKoordinator)) {`  
`$row-`  
`>nama_petugas==$rowKoordinator-`  
`>ID ? $selected="SELECTED" :`  
`$selected="" ;`  
`?>`

```

<option
value="<?=$rowKoordinator-
>ID?>">
<?=$selected?><><?=$rowKoordinat-
or->NAMA?></option>

<?php } ?>

</select>
<!--
<input type="text"
name="cnama_petugas"
style="width:100%" title="Nama
Petugas" value="<?=$row-
>nama_petugas?>"> -->
</td>
</tr>
<tr>

<td>NIP&ampnbsp</td>
<td
colspan="3"><input type="text"
name="cnip" style="width:100%" title="NIP Petugas"
value="<?=$row-
>nip_petugas?>"></td>

</tr>
<tr>

<td>Jabatan dalam
TIM&ampnbsp</td>
<td
colspan="3"><input type="text"

```

```

name="cjabatan"
style="width:100%"
title="Jabatan TIM"
value="php=$row-&gt;jabatan??"&gt;&lt;/td&gt;
</pre

```

<!--
<tr>
<td>SKPD </td>
<td colspan="3"><input type="text" name="cskp" style="width:100%" title="SKPD" value="php=$row-&gt;skpd??"&gt;&lt;/td&gt;
</pre

```

<tr>
<td>Status Pemeriksaan </td>
<td colspan="3">
<select name="nstatus_pemeriksaan" id="nstatus_pemeriksaan" style="width:100%" title="Status Pemeriksaan">
<option value="">Pilih Status Pemeriksaan</option>
<option value="1" &gt;status==1 || $row->

```



```

>status=='1' ? "selected" : "">&gt;1. Lengkap / Memenuhi Syarat</option>
<option value="2" &gt;status==2 ? "selected" : "">&gt;2. Tidak Memenuhi Syarat</option>
</select>

```



```

</td>
</tr>
<tr>
<td>Form Entry BAP </td>
<td colspan="3"><input type="checkbox" id="cekForm" &gt;$_GET['id']!='' ? "checked" : ""&gt;&gt;<a onclick="javascript:formEntryBAP()"></a></td>

```



```

</tr>
<tr>
<td>Edit Lokasi &amp; Jenis Permohonan </td>
<td colspan="3"><a onclick="javascript:formLokasi()";>

```



```

style="cursor:pointer;"></a></td>

```



```

</tr>
<tr>
<td colspan="4" id="title">
<a onclick="javascript:tambahData();">&nbsp;Tambah</a>&nbsp;&nbsp;<a onclick="deleteData()";>&nbsp;Hapus</a>&nbsp;&nbsp;<a onclick="javascript:tambahDataKe();">&nbsp;Tambah Ke</a>

```



```

</td>
</tr>
<tr>
<td colspan="4" id="field">
<div id="gridbox" width="300%" height="150px" style="background-color:white;overflow:hidden"></div>

```


```


```

```
<textarea  
name="cketerangan"  
style="display:none;width:100%;  
"  
id="cketerangan"><?!=empty($row  
->deskripsi) ? $row->deskripsi  
: ""?></textarea>  
                </td>  
            </tr>  
        </table>  
    </div>  
<script type="text/javascript">  
    $(document).ready(function()  
    {  
        /* Maset Combo Box  
        di grid jenis */  
  
        $('#nstatus_pemeriksaan').  
change(function(){  
        $nilai=  
        $(this).val();  
  
        if($nilai==2){  
  
            $('#title').html('Keteran  
gan');  
  
            $('#gridbox').css('displa  
y','none');  
  
            $('#cketerangan').css('di  
splay','block');  
        }  
    })
```

```
$('#guna_bangunan').attr(  
'disabled','true');  
        }else{  
  
            $('#title').html('<a  
onclick="javascript:tambahData(  
);"  
style="cursor:pointer;">&nbsp;  
ambah</a>&nbsp;&nbsp;<a  
onclick="deleteData()"  
style="cursor:pointer;">&nbsp;  
Hapus</a>&nbsp;&nbsp;<a  
onclick="javascript:tambahDataK  
e();"  
style="cursor:pointer;">&nbsp;  
ambah Ke</a>');  
  
            $('#title').css('display'  
, 'block');  
  
            $('#gridbox').css('displa  
y','block');  
  
            $('#cketerangan').css('di  
splay','none');  
  
            $('#guna_bangunan').remov  
eAttr('disabled');  
        }  
    })
```

```
//  
//  
$.ajax({  
    type: "POST",  
    url:  
    '<?=$xpath?>post_setup.php',  
    data:  
    'data=obyek',  
    //  
    success:function(data){  
        //  
        var  
        explode= data.split("+");  
        //  
        for(var  
i=0;i<explode.length;i++){  
        //  
        if(explode[i] != ''){  
        //  
        var baris=  
        explode[i].split("|");  
        //  
        var no= i + 1;  
        //  
        mygrid.getCombo(1).put(ba  
ris[0],baris[1]);  
        //  
        }  
        //  
        }  
    }  
});  
  
$.ajax({  
    type: "POST",  
    url:  
    '<?=$xpath?>post_setup.php',  
    data:  
    'data=koef',  
    success:function(data){
```

```
        var
explode= data.split("+");
        for(var
i=0;i<explode.length;i++){
            if(explode[i] != ''){
                var baris=
explode[i].split("|");
                var no= i + 1;
                mygrid.getCombo(3).put(baris[0],baris[1]);
            }
        });
        var satuan= new
Array();
satuan[0]= "bh";
satuan[1]= "m3";
satuan[2]= "m2";
satuan[3]= "m1";
satuan[4]= "lot";
satuan[5]= "m";
for(var
i=0;i<satuan.length;i++){
    mygrid.getCombo(5).put(satuan[i],satuan[i]);
}
        mygrid.getCombo(6).put(1,
'Ya');
        mygrid.getCombo(6).put(0,
'Tidak');
        mygrid.getCombo(7).put(1,
'Ya');
        mygrid.getCombo(7).put(0,
'Tidak');
        mygrid.getCombo(8).put(1,
'Ya');
        mygrid.getCombo(8).put(0,
'Tidak');
        mygrid.getCombo(9).put(1,
'Ya');
        mygrid.getCombo(9).put(0,
'Tidak');
        mygrid.cells(mygrid.getSelectedId(),6).setValue(0);
        mygrid.cells(mygrid.getSelectedId(),7).setValue(0);
        mygrid.cells(mygrid.getSelectedId(),8).setValue(0);
        mygrid.cells(mygrid.getSelectedId(),9).setValue(0);
    }
    mygrid = new
dhtmlXGridObject('gridbox');
mygrid.setImagePath("./gr
id/dhtml/imgs/");
mygrid.setHeader("No.,Ura
ian Jenis,Luas,Koefisiensi,Jumlah
Unit,Satuan,Renovasi,Balik
Nama,Penambahan,Perpanjangan");
mygrid.setSizes();
mygrid.setInitWidths("50,
*,100,150,100,100,100,100,1
00");
mygrid.setColAlign("cente
r,left,right,left,lef
t,lef
t,lef
t");
mygrid.setColTypes("ro,ed
,ed,coro,ed,coro,coro,coro
,coro");
//
mygrid.attachEvent("onEdi
tCell",doOnCellEdit);

mygrid.setColSorting("str
,str,int")
mygrid.init();
//
mygrid.setMultiSelect(!my
grid.selMultiRows);
mygrid.setSkin("gray");
// mygrid.loadXML("<?php
echo $xpath; ?>grid.xml");
```

```
</script>
<div id="tombol1">
    <table>
        <tr>
            <td
height="45"><br>
                <span
id="pushbutton5" class="yui-
button yui-push-button"><em
class="first-child"><button
type="button">Simpan</button></em></span>
                <span
id="pushbutton6" class="yui-
button yui-push-button"><em
class="first-child"><button
type="reset">Reset</button></em>
            </span>
        </td>
    </tr>
</table>
</div>
<input type="hidden"
name="csubmit" value="Simpan">
<input type="hidden"
name="editaj" id="editaj"
value="<?=$row->ID_03_01?>">
<input type="hidden"
name="nid_login"
value="<?=$_SESSION['nid_login'
]?>">
<input type="hidden"
id="jenisTab"
value="pemeriksaan">
```

```
<input type="hidden"
id="delete" value="">
<input type="hidden"
id="input" value="">
<input type="hidden"
id="idBAP"
value="<?=$_GET['id']!=' ' ?
$arrId->ID_03_01 : ""?>">
<input type="hidden"
id="tambahKe" value="0">
<input type="hidden"
id="lokasi" name="lokasi">
<input type="hidden"
id="jenis" name="jenis">
</form>
<? } ?>
```

## 1. Halaman Login

The screenshot shows a login interface. At the top left is the Indonesian national emblem (Garuda Pancasila). Below it is a blue-themed form containing fields for 'Username' (with 'admin' entered) and 'Password' (with several dots indicating the password). A 'Login' button is at the bottom right of the form. The background of the page features a large, faint watermark-like image of a building's interior.

## 2. Halaman Utama Admin

The screenshot shows the main administration dashboard. At the top is a horizontal navigation menu with items: Start ▾, Aktivitas ▾, Daftar Perizinan ▾, Pembayaran ▾, Penomoran ▾, Penyerahan ▾, Laporan ▾, Setup ▾, Tools ▾, and NSR ▾. Below the menu is a blue header bar with the text "Aplikasi Izin Mendirikan Bangunan". The main content area features a large, blurred image of a modern office lobby with a reception desk and chairs. At the bottom of the screen is a status bar with the text "Logged as Administrator on from IP 127.0.0.1" and "Last Login on until from IP 127.0.0.1".

### 3. Pendaftaran Izin Mendirikan Bangunan

#### Tab Permohonan

Start ▾ Aktivitas ▾ Daftar Perizinan ▾ Pembayaran ▾ Penomoran ▾ Penyerahan ▾ Laporan ▾ Setup ▾ Tools ▾ NSR ▾

#### Aktivitas 3 • Izin Mendirikan Bangunan / IMB

Permohonan Pemeriksaan Permohonan Penetapan Permohonan Penolakan Permohonan

Permohonan Pemohon Komponen Lokasi Bangunan

**Permohonan**

Tgl. Permohonan: 08/05/2011  
Jenis Permohonan: 1. Baru  
No. Permohonan: 951/IMB-Bid.Pel/V/2011  
Perihal: Permohonan Ijin Mendirikan Bangunan (IMB)  
Lampiran Status:

Logged as Administrator on Sun, May 08, 2011 20:19:58 from IP :1  
Last Logon on Sun, May 08, 2011 19:53:28 until Thu, January 01, 1970 07:00:00 from IP ::1

#### Tab Pemohon

Start ▾ Aktivitas ▾ Daftar Perizinan ▾ Pembayaran ▾ Penomoran ▾ Penyerahan ▾ Laporan ▾ Setup ▾ Tools ▾ NSR ▾

#### Aktivitas 3 • Izin Mendirikan Bangunan / IMB

Permohonan Pemeriksaan Permohonan Penetapan Permohonan Penolakan Permohonan

Permohonan Pemohon Komponen Lokasi Bangunan

**Pemohon**

No. KTP:  Nama:   
NPW RD:   
Contact Person:   
No. Telepon:  No. HP:   
Tempat Lahir:  Tanggal Lahir:   
Alamat:   
Nama Perusahaan:  Jabatan:

Logged as Administrator on Tue, May 10, 2011 08:53:22 from IP :1  
Last Logon on Sun, May 08, 2011 20:19:58 until Mon, May 09, 2011 13:16:10 from IP ::1

#### Tab Komponen

Start ▾ Aktivitas ▾ Daftar Perizinan ▾ Pembayaran ▾ Penomoran ▾ Penyerahan ▾ Laporan ▾ Setup ▾ Tools ▾ NSR ▾

#### Aktivitas 3 • Izin Mendirikan Bangunan / IMB

Permohonan Pemeriksaan Permohonan Penetapan Permohonan Penolakan Permohonan

Permohonan Pemohon Komponen Lokasi Bangunan

**Komponen Bangunan**

Peruntukan Bangunan:   
Pondasi:  Dinding:   
Rangka Kap:  Atap:   
Kusen:  Lantai:   
Luas Bangunan:  m<sup>2</sup>

Logged as Administrator on Tue, May 10, 2011 08:53:22 from IP :1  
Last Logon on Sun, May 08, 2011 20:19:58 until Mon, May 09, 2011 13:16:10 from IP ::1

## Tab Lokasi Bangunan

Start ▾ Aktivitas ▾ Daftar Perizinan ▾ Pembayaran ▾ Penomoran ▾ Penyerahan ▾ Laporan ▾ Setup ▾ Tools ▾ NSR ▾

Aktivitas 3 • Izin Mendirikan Bangunan / IMB

Permohonan Pemeriksaan Permohonan Penetapan Permohonan Penolakan Permohonan

Permohonan Pemohon Komponen Lokasi Bangunan

**Lokasi**

Kampung/ Komplek  
Perumahan  
Jalan  
Kecamatan  
Desa/Kelurahan  
Berada diatas Tanah  
Nomor

Logged as **Administrator** on Tue, May 10, 2011 08:53:22 from IP ::1  
Last Login on Sun, May 08, 2011 20:19:58 until Mon, May 09, 2011 13:16:10 from IP ::1

## 4. Pengawasan dan Pengendalian (Pemeriksaan)

Start ▾ Aktivitas ▾ Daftar Perizinan ▾ Pembayaran ▾ Penomoran ▾ Penyerahan ▾ Laporan ▾ Setup ▾ Tools ▾ NSR ▾

Aktivitas 3 • Izin Mendirikan Bangunan / IMB

Permohonan Pemeriksaan Permohonan Penetapan Permohonan Penolakan Permohonan

**Pemeriksaan Permohonan**

No. Pendaftaran

Tgl. Pemeriksaan

No. BA Pemeriksaan

Nama Petugas

Status Pemeriksaan

Form Entry BAP

Edit Lokasi & Jenis Permohonan

| No. | Uraian Jenis | Luas | Koefisien | Jumlah Unit | Satuan | Renovasi | Balik Nama | Penambahan | Perpanjangan |
|-----|--------------|------|-----------|-------------|--------|----------|------------|------------|--------------|
|     |              |      |           |             |        |          |            |            |              |

Logged as **Administrator** on Sun, May 08, 2011 20:19:58 from IP ::1  
Last Login on Sun, May 08, 2011 19:53:28 until Thu, January 01, 1970 07:00:00 from IP ::1

**Data Lapangan**

Faslik Terbangun  %  
Sepadan Bangunan   
GSP  GSB  
GSP Samping  GSB Samping  
KDB  KLB  
Jumlah Lantai  Lantai  
Lebar Bentangan

Batas Lahan / Bangunan

- Sebelah Utara  Lantai - Sebelah Selatan   
- Sebelah Barat  Lantai - Sebelah Timur

**Data Gambar**

● Gambar Rencana

GSP  GSB  
GSP Samping  GSB Samping  
KDB  KLB  
Jumlah Lantai  Lantai  
Lebar Bentangan

● Struktur Konstruksi

Pondasi  Dinding   
Rangka Kap  Atap   
Struktur Kolom & Balok  Lantai

● Kesimpulan

Kesimpulan

Dengan alasan

## 5. Penetapan

Screenshot of the 'Penetapan' (Approval) screen for building permit applications.

The top navigation bar includes: Start, Aktivitas, Daftar Perizinan, Pembayaran, Penomoran, Penyerahan, Laporan, Setup, Tools, NSR.

The main title is 'Aktivitas 3 Izin Mendirikan Bangunan / IMB'.

Sub-tabs: Permohonan, Pemeriksaan Permohonan, **Penetapan Permohonan**, Penolakan Permohonan.

**Penetapan Retribusi** (Approval of Fines):

|                   |   |
|-------------------|---|
| No. Pendafataran  | <input type="text"/>                          |
| Nomor SKRD        | <input type="text"/>                          |
| Tanggal Penetapan | 08/05/2011 <input type="button" value="..."/> |
| Alamat Pemohon    | <input type="text"/>                          |
| Pemohon Lokasi    | <input type="text"/>                          |
| NPWDRD            | <input type="text"/>                          |

**Dasar Pengenaan** (Basis of Assessment):

Hitung (Calculate):

| No. | Uraian Jenis | Guna Bangunan | Luas | Koefisien | Jml Unit | Satuan | Tarif | Keletapan |
|-----|--------------|---------------|------|-----------|----------|--------|-------|-----------|
|     |              |               |      |           |          |        |       |           |

**Perhitungan Biaya** (Cost Calculation):

|           |                      |
|-----------|----------------------|
| Retribusi | 0                    |
| Terbilang | <input type="text"/> |

Simpan  Reset

Log message: Logged as Administrator on Sun, May 08, 2011 20:19:58 from IP ::1  
Last Logon on Sun, May 08, 2011 19:53:28 until Thu, January 01, 1970 07:00:00 from IP ::1

## 6. Pembayaran

Screenshot of the 'Pembayaran' (Payment) screen for building permit applications.

The top navigation bar includes: Start, Aktivitas, Daftar Perizinan, **Pembayaran**, Penomoran, Penyerahan, Laporan, Setup, Tools, NSR.

The main title is 'Pembayaran Izin'.

Sub-tabs: Entri Pembayaran, Sudah Dibayar.

**Informasi Perizinan** (Permit Information):

|                    |                      |
|--------------------|----------------------|
| Nomor Pendafataran | <input type="text"/> |
| Nama Pemohon       | <input type="text"/> |
| Jenis Izin/ Akta   | <input type="text"/> |
| Alamat Pemohon     | <input type="text"/> |
| Tanggal Permohonan | <input type="text"/> |
| Tanggal Penetapan  | <input type="text"/> |

**Informasi Pembayaran** (Payment Information):

|                        |   |
|------------------------|---|
| Tanggal Pembayaran     | 08/05/2011 <input type="button" value="..."/> |
| Denda (Bulan)          | <input type="text"/>                          |
| Jumlah Pembayaran      | <input type="text"/>                          |
| Terbilang              | <input type="text"/>                          |
| Jumlah Setelah Denda   | <input type="text"/>                          |
| Terbilang              | <input type="text"/>                          |
| Nomor Bukti Pembayaran | <input type="text"/>                          |

Simpan  Reset

Log message: Logged as Administrator on Sun, May 08, 2011 20:19:58 from IP ::1  
Last Logon on Sun, May 08, 2011 19:53:28 until Thu, January 01, 1970 07:00:00 from IP ::1

## 7. Penomoran

Logged as Administrator on Sun, May 08, 2011 20:19:58 from IP : -1  
Last Logon on Sun, May 08, 2011 19:53:28 until Thu, January 01, 1970 07:00:00 from IP : -1

## 8. Penyerahan

Logged as Administrator on Sun, May 08, 2011 20:19:58 from IP : -1  
Last Logon on Sun, May 08, 2011 19:53:28 until Thu, January 01, 1970 07:00:00 from IP : -1

## 9. Form Edit (Pendaftaran, Pemeriksaan, Penetapan)

### Pendaftaran

| No.  | Tanggal            | Nomor Pendaftaran              | Nama Pemohon            | Nama Perusahaan        | Lokasi                    |
|--|--------------------|--------------------------------|-------------------------|------------------------|---------------------------|
| 1  | 17/6/2010          | 03 999.10                      | NOEK BRISSINA SOEHARDJO | -                      | Kirai I Kavling Blok K/8  |
| 2  | 17/6/2010          | 03 998.10                      | CHAIRULSAN SIREGAR      | -                      | Griya Pamulang II B 02/   |
| 3  | 17/6/2010          | 03 997.10                      | IR. FATIMAH ULFAH       | -                      | Jl. Ki Hajar Dewantara    |
| 4  | 17/6/2010          | 03 996.10                      | ALAMSYAH HASIBUAN       | -                      | Jl. Ki Hajar Dewantara    |
| 5  | 17/6/2010          | 03 995.10                      | H. KARSONO              | -                      | Jl. Flamboyan Manis No    |
| 6  | 17/6/2010          | 03 994.10                      | H. ABDUL MALIK GOMULYA  | PT. WIRA MAKMUR SEJATI | Jl. Meruya II RT. 01/RW   |
| 7  | 17/6/2010          | 03 993.10                      | FENDI CHRISTANTO, SE    | -                      | Perum Alam Sutera, Jl.    |
| 8  | 17/6/2010          | 03 992.10                      | FENNY INDRAWATI         | -                      | Bintaro Jaya Sekt. VIII E |
| 9  | 17/6/2010          | 03 991.10                      | NINIK SOEMARNI          | -                      | BSD Blok C3 No. 6 Sek     |
| 10   | 17/6/2010          | 03 990.10                      | LINAH                   | -                      | Sektor III.3 Blok H.6 No. |
| 11   | 24/1/2011          | 03 99.11                       | IR. H. ASSAARI          | -                      | Jl. Legoso Raya           |
| Aplikasi ini dibuat oleh Dinas Perizinan dan Pengawas Lingkungan |                    |                                |                         |                        |                           |
| 15   | Halaman 1 dari 247 | Data 1 a.d 15 total 3691 items |                         |                        |                           |

## Pemeriksaan

Daftar Perizinan 3 • Izin Mendirikan Bangunan / IMB

Permohonan Pemeriksaan Permohonan Penetapan Permohonan Pendakian Permohonan

| No. | Tanggal   | Nomor Pemeriksaan       | Nama Perusahaan | Nomor Pendaftaran       |
|-----|-----------|-------------------------|-----------------|-------------------------|
| 1   | 28/6/2010 | 999/IMB-Bid.Pel/VI/2010 | 03.999.10       | NOEK BRISSINA SOEHARDJO |
| 2   | 22/6/2010 | 998/IMB-Bid.Pel/VI/2010 | 03.998.10       | CHAIRULSAN SIREGAR      |
| 3   | 5/7/2010  | 997/IMB-Bid.Pel/VI/2010 | 03.997.10       | Ir. FATIMAH ULFAH       |
| 4   | 5/7/2010  | 996/IMB-Bid.Pel/VI/2010 | 03.996.10       | ALAMSYAH HASIBUAN       |
| 5   | 21/7/2010 | 995/IMB-Bid.Pel/VI/2010 | 03.995.10       | H. KARSONO              |
| 6   | 9/7/2010  | 994/IMB-Bid.Pel/VI/2010 | 03.994.10       | H. ABDUL MALIK GOMULYA  |
| 7   | 29/6/2010 | 993/IMB-Bid.Pel/VI/2010 | 03.993.10       | FENDI CHRISTANTO, SE    |
| 8   | 8/7/2010  | 992/IMB-Bid.Pel/VI/2010 | 03.992.10       | FENNY INDRAWATI         |
| 9   | 8/7/2010  | 991/IMB-Bid.Pel/VI/2010 | 03.991.10       | NINIK SOEMARNI          |
| 10  | 29/6/2010 | 990/IMB-Bid.Pel/VI/2010 | 03.990.10       | LINAH                   |
| 11  | 21/2/2011 | 99/IMB-Bid.Pell/2011    | 03.99.11        | IR. H. ASSAARI          |

Halaman 1 dari 225 Data 1 s.d 15 total 3361 items

## Penetapan

Daftar Perizinan 3 • Izin Mendirikan Bangunan / IMB

Permohonan Pemeriksaan Permohonan Penetapan Permohonan Pendakian Permohonan

| No. | Tanggal   | Nomor Penetapan | Nomor Pendaftaran | Nama Pemohon            | Nama Pemohon |
|-----|-----------|-----------------|-------------------|-------------------------|--------------|
| 1   | 22/7/2010 | 1011            | 03.999.10         | NOEK BRISSINA SOEHARDJO |              |
| 2   | 2/7/2010  | 868             | 03.998.10         | CHAIRULSAN SIREGAR      |              |
| 3   | 13/6/2010 | 1206            | 03.995.10         | H. KARSONO              |              |
| 4   | 16/7/2010 | 945             | 03.994.10         | H. ABDUL MALIK GOMULYA  |              |
| 5   | 8/7/2010  | 898             | 03.993.10         | FENDI CHRISTANTO, SE    |              |
| 6   | 8/10/2010 | 1442            | 03.992.10         | FENNY INDRAWATI         |              |
| 7   | 26/7/2010 | 1059            | 03.991.10         | NINIK SOEMARNI          |              |
| 8   | 8/7/2010  | 902             | 03.990.10         | LINAH                   |              |
| 9   | 10/3/2011 | 381             | 03.99.11          | IR. H. ASSAARI          |              |
| 10  | 11/3/2010 | 299             | 03.99.10          | A.R RINI WINARTI        |              |
| 11  | 9/7/2010  | 914             | 03.989.10         | Ir. DEDDY IRawan        |              |

Halaman 1 dari 190 Data 1 s.d 15 total 2849 items

## 10. Cetak Surat Izin

Start Aktivitas Daftar Perizinan Pembayaran Penomoran Penyerahan Laporan Setup Tools NSR

Penomoran Izin

Entri Penomoran Sudah Diproses

Penomoran Izin Berhasil dengan Nomor Surat: 864.3/1024-BP2T/2009

Pejabat Penandatangan: H. EUTIK SUARTA (Walikota Tangerang)

Preview Surat Izin 1

Preview Surat Izin 1

Logged as Administrator on from IP 127.0.0.1  
Last Login on until from IP 127.0.0.1

