

**SKRIPSI**  
**RANCANG BANGUN SISTEM MONITORING ANGGARAN**  
**KEUANGAN BERBASIS WEB PADA BIRO HUKUM DAN KERJA SAMA**  
**LUAR NEGERI**

(Studi Kasus: Sekretariat Jenderal Kementerian Agama Republik Indonesia)



Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Sistem Informasi

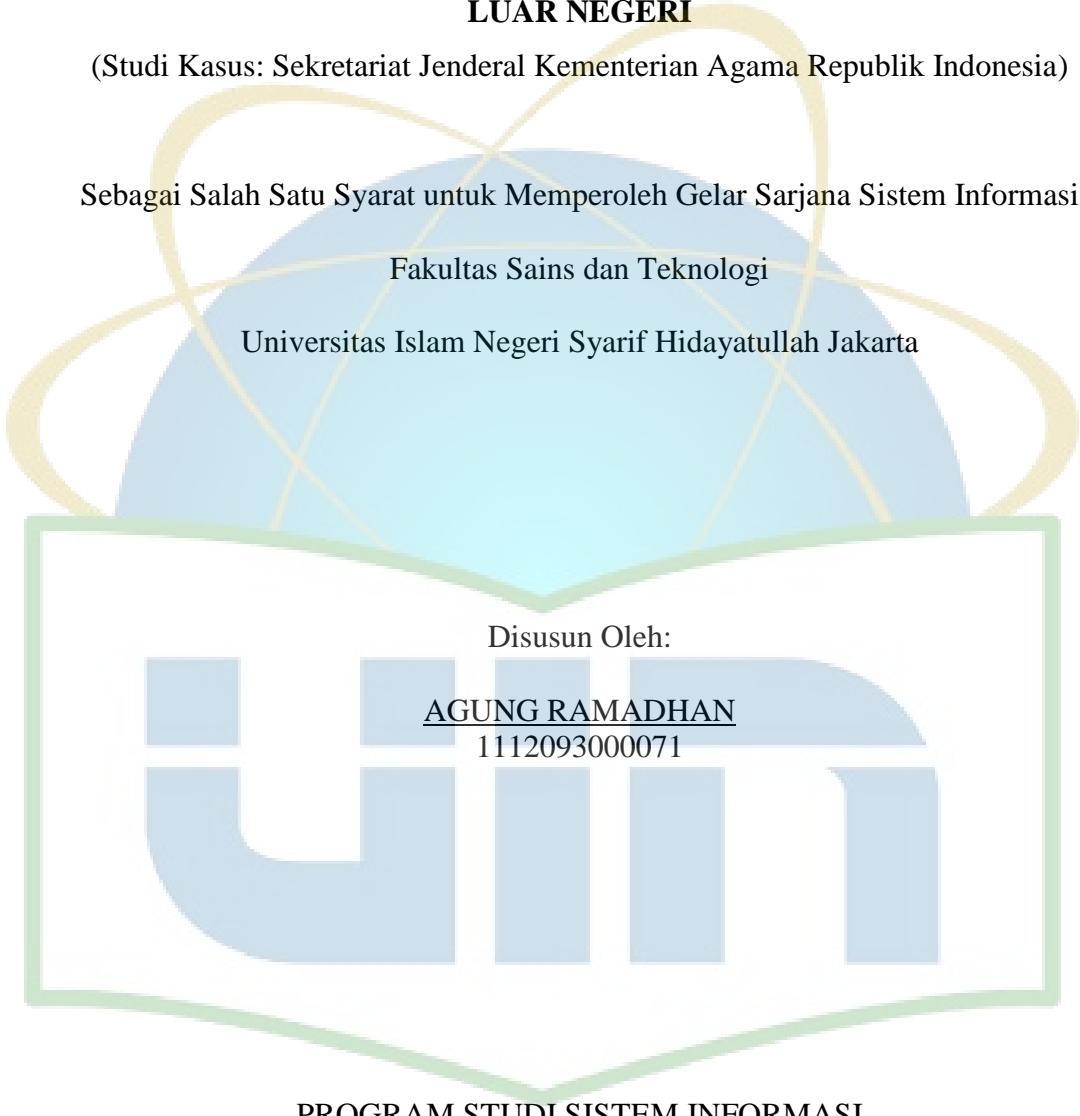


PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
2019 M / 1440 H

**SKRIPSI**

**RANCANG BANGUN SISTEM MONITORING ANGGARAN  
KEUANGAN BERBASIS WEB PADA BIRO HUKUM DAN KERJA SAMA  
LUAR NEGERI**

(Studi Kasus: Sekretariat Jenderal Kementerian Agama Republik Indonesia)



PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
2019 M / 1440 H

## LEMBAR PENGESAHAN

### RANCANG BANGUN SISTEM MONITORING ANGGARAN KEUANGAN BERBASIS WEB PADA BIRO HUKUM DAN KERJA SAMA LUAR NEGERI

(Studi Kasus: Sekretariat Jenderal Kementerian Agama Republik Indonesia)



Disusun Oleh:

AGUNG RAMADHAN

1112093000071

Menyetujui,

Pembimbing I.

Nur Aeni Hidayah, MMSI  
NIP. 19750818 200501 2008

Pembimbing II.

Yusuf Durachman, M.Sc, MIT  
NIP. 19710522 200604 1002

Mengetahui,

Ketua Program Studi Sistem Informasi Fakultas Sains dan Teknologi

UIN Syarif Hidayatullah Jakarta

A'ang Subiyakte, Ph.D  
NIP. 19760219 200710 1 002

## PENGESAHAN UJIAN

Skripsi yang berjudul Sistem Informasi Rancang Bangun Sistem *Monitoring Aggaran Keuangan Berbasi Web Pada Biro Hukum dan Kerja Sama Luar Negeri* (Studi Kasus : Sekretariat Jenderal Kementerian Agama Republik Indonesia) yang ditulis oleh Agung Ramadhan, NIM. 1112093000055 telah diuji dan dinyatakan lulus dalam sidang Munaqosah Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta pada hari Senin, 25 Juni 2019. Skripsi ini telah diterima sebagai salah satu syarat memperoleh gelar Sarjana Satu (S1) Program Studi Sistem Informasi.

Menyetujui,

Pengaji I

Nurwasin, M.Kom

NIP. 197607152 201101 1005

Pengaji II

Fitroh, M.Kom

NIP. 19790923 200912 2 006

Pembimbing I,

  
Nur Aeni Halayah, MMSI  
NIP. 19750818 200501 2008

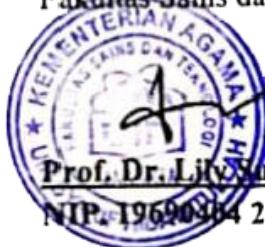
Pembimbing II,

  
Yusuf Durachman, M.Sc, MIT  
NIP. 19710522 200604 1002

Mengetahui,

Dekan

Fakultas Sains dan Teknologi



Prof. Dr. Lily Nurayya Eka Putri, M. Env. Stud  
NIP. 19690304 200501 2 005

Ketua

Program Studi Sistem Informasi

  
A'ang Subivakto, Ph.D  
NIP. 19760219 200710 1 002

## PERNYATAAN

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

Jakarta, 29 Mei 2019



Agung Ramadhan  
111209300071

## ABSTRAK

Agung Ramadhan (1112093000071), "Rancang Bangun Sistem *Monitoring* Anggaran Keuangan Pada Biro Hukum dan Kerja Sama Luar Negeri (Studi Kasus: Sekretariat Jenderal kementerian Agama Republik Indonesia)" di bawah bimbingan Nur Aeni Hidayah dan Yusuf Durachman.

*Monitoring* anggaran keuangan adalah pemantauan keluar masuknya berkas pada pengelola keuangan guna menghindari adanya kesalahan penginputan maupun ketidak tertiban dalam alur berkas keluar masuk, pengelolaan keuangan secara manual biasanya masih terdapat permasalahan yakni kesulitan dalam tertib administrasi keuangan. Hal tersebut disebabkan tidak adanya *database* yang terintegrasi, sehingga pihak pengelola keuangan tidak dapat memantau alur berkas secara langsung. Dan yang paling fatal adalah sering terjadinya tindakan saling tuduh pada berkas yang hilang, berkas yang masuk dan keluar tidak terjadwal dengan baik. Tujuan dari penelitian ini adalah merancang dan membangun sistem informasi yang menyajikan informasi aliran keluar masuknya berkas anggaran keuangan pada Biro Hukum dan Kerja Sama Luar Negeri. Metode pengumpulan data pada penelitian ini yaitu obervasi dan wawancara. Metode pengembangan sistem yang digunakan adalah *Rapid Application Development* (RAD) yang meliputi *planning, analysis, design* dan *implementation*. Alat perancangan yang digunakan adalah *Unified Modeling Language* (UML). Sistem ini berbasis *web* dibangun dengan bahasa pemrograman *PHP* dan *MySQL* sebagai *database*. Didalam *monitoring* anggaran keuangan yang mencakup proses pengelolaan anggaran yang mampu menyajikan data penyerapan anggaran keuangan berdasarkan kecocokan realisasi dengan akun belanja. Dapat mengetahui keberadaan berkas, meminimalisir hilang dan tidak tertib antrian yang terjadi pada sistem manual serta mengamankan berkas pertanggung jawaban di dalam sistem. Sehingga dengan adanya penelitian ini, diharapkan lembaga pemerintahan dapat melaksanakan dapat meminimalisir terjadinya sikap saling tuduh akibat berkas yang hilang dan tidak tertib antrian antara pengelola dan penyerapan anggaran menjadi menjadi lebih efisien dan tepat waktu.

**Kata Kunci :** *Monitoring, Sirkulasi Material, Proyek, UML, RAD*

V Bab + 304 Halaman + xxv Halaman + 124 Gambar + 84 Tabel + Pustaka + Lampiran  
Pustaka Acuan (49, 2007 – 2017)

## KATA PENGANTAR

*Assalamu'alaikum Wr.Wb.*

*Alhamdulillahirobbil'aalamin*, puji syukur yang sebesar-besarnya kehadiran Allah SWT, karena dengan limpahan rahmat, hidayah, dan taufiq-Nya, penulis diberikan nikmat sehat dan ilmu sehingga dapat melaksanakan, dan menyelesaikan penulisan skripsi dengan judul “Rancang Bangun Sistem Monitoring Anggaran Keuangan Berbasis Web Pada Biro Hukum dan Kerja Sama Luar Negeri (Studi Kasus : Sekretariat Jenderal Kementerian Agama Republik Indonesia)”. Shalawat beriringan salam semoga selalu tercurah kepada junjungan Nabi Muhammad SAW beserta keluarga, sahabat, kerabat, serta muslimin, dan muslimat, semoga kita semua mendapatkan syafa’at dari beliau di akhirat kelak. Amin.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada seluruh pihak yang telah banyak membantu baik moril dan materil dalam menyelesaikan laporan ini. Untuk itu perkenankanlah penulis menyampaikan ungkapan terima kasih kepada:

1. Bapak Dr. Agus Salim, M.Si, selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.
2. Ibu Nia Kumaladewi, MMSI, selaku Ketua Program Studi Sistem Informasi Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.
3. Ibu Meinarini Catur Utami, MT, selaku Sekretaris Program Studi Sistem Informasi Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.

4. Ibu Nur Aeni Hidayah, S.E., M.M.S.I selaku dosen pembimbing pertama, dan Bapak Yusuf Durachman, M.TI selaku dosen pembimbing kedua yang telah banyak membimbing penulis dalam menyusun skripsi ini.
5. Dosen-dosen program studi Sistem Informasi yang telah memberikan ilmunya selama penulis duduk dibangku kuliah.
6. Bapak Prof. H. Achmad Gunaryo selaku Kepala Biro Hukum dan Kerja Sama Luar Negeri yang telah mengizinkan penulis untuk melakukan penilitian pada perusahaannya.
7. Ibu Nilam Nur Azizah selaku Bendahara Pengeluaran Pembantu Biro Hukum dan Kerja Sama Luar Negeri yang telah banyak membantu, dan memberikan informasi yang saya butuhkan, sehingga penulis bisa menyelesaikan skripsi ini, serta memberikan masukan dan saran kepada penulis.

Penulis menyadari bahwa dalam penulisan skripsi ini masih jauh dari sempurna, oleh karena itu penulis mengharapkan kritik dan saran agar menjadi lebih baik lagi yang dapat disampaikan melalui *email* agung.ramadhan12@mhs.uinjkt.ac.id.

Akhirkata penulis mngucapka terimakasih. *Wassalamu 'alaikum Wr.Wb.*

Jakarta, 27 Mei 2019

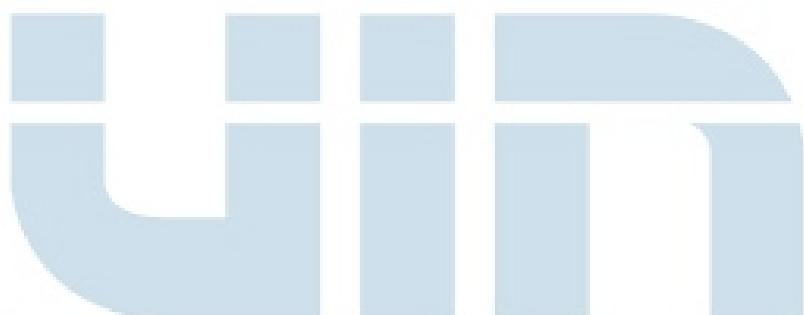
  
Agung Ramadhan  
1112093000071

## LEMBAR PERSEMBAHAN

Pada kesempatan kali ini, penulis ingin mempersembahkan skripsi ini kepada seluruh pihak yang telah membantu baik moril dan materil dalam menyelesaikan skripsi ini, terutama kepada:

1. Ansori, dan Maryati, orang tua saya, yang sangat saya cintai dan sayangi. Terima kasih banyak atas Doa, motivasi, arahan, serta kasih sayang yang tidak terhingga sepanjang masa. Semoga kelak agung mampu membahagiakan dan membuat bapak dan mamah bangga.
2. Kakak-kakakku tercinta Rizal Ahmad, Sukma Adiyatma dan istriku tersayang Habibatun Rachmah yang selalu memberi doa serta motivasi selama pembuatan skripsi.
3. Keluarga Markas CV. Internal Media, Win, Endi, Ladur Cobain, Crise, Ame, Thoriq, Haji Rapik, Murpi, Iky, Ipeng, Ega, Iway, Umin, Iqbal, Akrom yang sering membantu penulis dalam berbagai hal dari awal sampai akhir penulisan skripsi.
4. Penghuni Kost Es krim Duo Gadih Kamar Pojok Kiri Bawah, Yoga dan Yusnil yang sabar dalam mendengarkan musik-musik aneh yang diputar penulis saat penulis mengerjaan skripsi.
5. Teman-teman dari Indramayu, Kang Deni Syuhada, Kang Gobak, Kang Manuk, Kang Pawon, Kang Datuk, Kang Jaki, Om Botak, dan Bu Pety yang sering direpotkan atas pertanyaan-pertanyaan aneh dari penulis.

6. Anggota PESTA garis keras yaitu Gilang, Bagong, Lutpi, Meha, Cacing, Acus, Eko, Padang dan Baok. Semoga selamanya kalian tetap asyik dan seru. Salam Gocek!!!.
7. Sahabat-sahabat SI B 2012, dan SI A 2012. Terima kasih untuk kebersamaan, kerjasama, motivasi, dan selalu mendoakan dalam kebaikan. Waktu bersama sahabat-sahabat sangatlah berkesan.
8. Seluruh pihak yang telah banyak berjasa terhadap proses penyelesaian skripsi ini yang tidak bisa disebutkan namanya satu persatu namun tidak mengurangi sedikitpun rasa hormat dan rasa terima kasih.
9. Semua orang yang menanyakan kapan lulus, kapan wisuda dan kapan nikah kepada penulis. Terimakasih atas pertanyaannya, penulis jadikan pertanyaan tersebut cambuk untuk menyelesaikan skripsi ini.



## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	iii
<b>PENGESAHAN UJIAN.....</b>	iii
<b>PERNYATAAN.....</b>	iv
<b>ABSTRAK .....</b>	vi
<b>KATA PENGANTAR.....</b>	vii
<b>LEMBAR PERSEMBERAHAN .....</b>	ix
<b>DAFTAR ISI.....</b>	xi
<b>DAFTAR GAMBAR.....</b>	xv
<b>DAFTAR TABEL .....</b>	xviii
<b>DAFTAR SIMBOL .....</b>	xx
<b>BAB I PENDAHULUAN.....</b>	1
<b>1.1 Latar Belakang .....</b>	1
<b>1.2 Identifikasi Masalah .....</b>	7
<b>1.3 Rumusan Masalah .....</b>	7
<b>1.4 Batasan Masalah.....</b>	8
<b>1.5 Tujuan Penelitian .....</b>	9
<b>1.5.1 Tujuan Khusus .....</b>	9
<b>1.5.2 Tujuan Umum .....</b>	10

1.6 Metodelogi Penelitian .....	10
1.7 Sistematika Penulisan .....	12
<b>BAB II LANDASAN TEORI .....</b>	<b>14</b>
<b>2.1. Konsep Dasar Sistem Informasi.....</b>	<b>14</b>
2.1.1. Pengertian Rancang Bangun .....	14
2.1.2. Pengertian Sistem.....	15
2.1.3. Karakteristik Sistem .....	16
2.1.4. Pengertian Sistem Infomasi.....	18
<b>2.2. Konsep Dasar Aplikasi Berbasis Web .....</b>	<b>21</b>
2.2.1. Pengertian Aplikasi .....	21
2.2.2. Pengertian Website .....	21
2.2.3. Pengertian Aplikasi Berbasis Web.....	21
<b>2.3. Pengertian Monitoring .....</b>	<b>23</b>
2.3.1. Peranan Pengawasan .....	24
2.3.2. Tujuan Monitoring .....	25
2.3.3. Prinsip-prinsip Monitoring.....	26
2.3.4. Sistem Monitoring.....	27
<b>2.4. Pengertian Anggaran .....</b>	<b>28</b>
2.4.1. Jenis Anggaran .....	28
2.4.2. Fungsi Anggaran .....	29

2.4.3. Pelaksanaan Anggaran .....	30
2.5. Pengeluaran Keuangan Negara .....	31
2.6. Metode Pengumpulan Data .....	34
2.6.1. Studi Literatur .....	34
2.6.2. Wawancara.....	37
2.6.3. Observasi.....	37
2.7. Pengembangan Sistem.....	38
2.7.1. Pengembangan Sistem Berbasis Objek .....	38
2.7.2. Karakter Dasar Sistem Berorientasi Objek .....	39
2.7.3. <i>Rapid Application Development (RAD)</i> .....	41
2.7.4. <i>Unified Modeling Language (UML)</i> .....	49
2.7.5. <i>Rich Picture</i> .....	60
2.6.7. Konsep Basis Data .....	61
2.6.8. Bahasa Pemrograman .....	68
2.6.8. Testing.....	71
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>73</b>
3.1. Metode Pengumpulan Data .....	73
3.1.1. Observasi.....	73
3.1.2. Studi Literatur .....	74
3.1.3. Wawancara.....	75

3.2. Metode Pengembangan Sistem .....	75
3.3. Alasan Menggunakan RAD.....	79
3.4. Kerangka Penelitian .....	80
<b>BAB IV PEMBAHASAN.....</b>	<b>82</b>
<b>4.1. <i>Requirement Planning</i> .....</b>	<b>82</b>
4.1.1. Gambaran Umum Kementerian Agama Republik Indonesia.....	82
4.1.2. Analisis Sistem.....	90
<b>4.2. <i>Workshop Design</i> .....</b>	<b>95</b>
4.2.1. Perancangan Desain Sistem .....	95
4.2.2. Perancangan Database.....	165
4.2.3. <i>Spesifikasi Database</i> .....	167
4.2.4. Design Interface .....	173
<b>BAB V PENUTUP.....</b>	<b>190</b>
5.1 Kesimpulan.....	190
5.2 Saran.....	191
<b>DAFTAR PUSTAKA .....</b>	<b>192</b>
<b>LAMPIRAN-LAMPIRAN .....</b>	<b>1</b>

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Karakteristik Sistem.....	18
<b>Gambar 2 2</b> Komponen Sistem Informasi .....	19
<b>Gambar 2 3</b> Tahapan Sistem Monitoring .....	27
<b>Gambar 2 4</b> Siklus RAD.....	42
<b>Gambar 3 1.</b> Kerangka Penelitian.....	81
<b>Gambar 4. 1</b> Logo Kementerian Agama Republik Indonesia .....	87
<b>Gamabar 4. 2</b> Struktur Organisasi Biro Hukum dan Kerja sama Luar Negeri Kementerian Agama RI.....	89
<b>Gamabar 4. 3</b> <i>Rich Picture Sistem</i> Berjalan .....	91
<b>Gamabar 4. 4</b> <i>Rich Picture Sistem</i> Usulan .....	95
<b>Gamabar 4. 5</b> <i>Use Case Diagram</i> .....	102
<b>Gambar 4. 6</b> <i>Activity Diagram</i> Login .....	125
<b>Gambar 4 7</b> <i>Activity Diagram</i> Manage User .....	127
<b>Gambar 4. 8</b> <i>Activity Diagram</i> Manage Anggaran.....	129
<b>Gambar 4. 9</b> <i>Activity Diagram</i> Manage Detail Anggaran .....	131
<b>Gambar 4. 10</b> <i>Activity Diagram</i> Pengajuan Kegiatan.....	133
<b>Gambar 4. 11</b> <i>Activity Diagram</i> Monitoring Anggaran.....	135
<b>Gambar 4. 12</b> Status Berkas .....	136
<b>Gambar 4. 13</b> <i>Activity Diagram</i> Validasi pengajuan kegiatan .....	137
<b>Gambar 4. 14</b> Upload Berkas Tagihan Pencairan.....	139

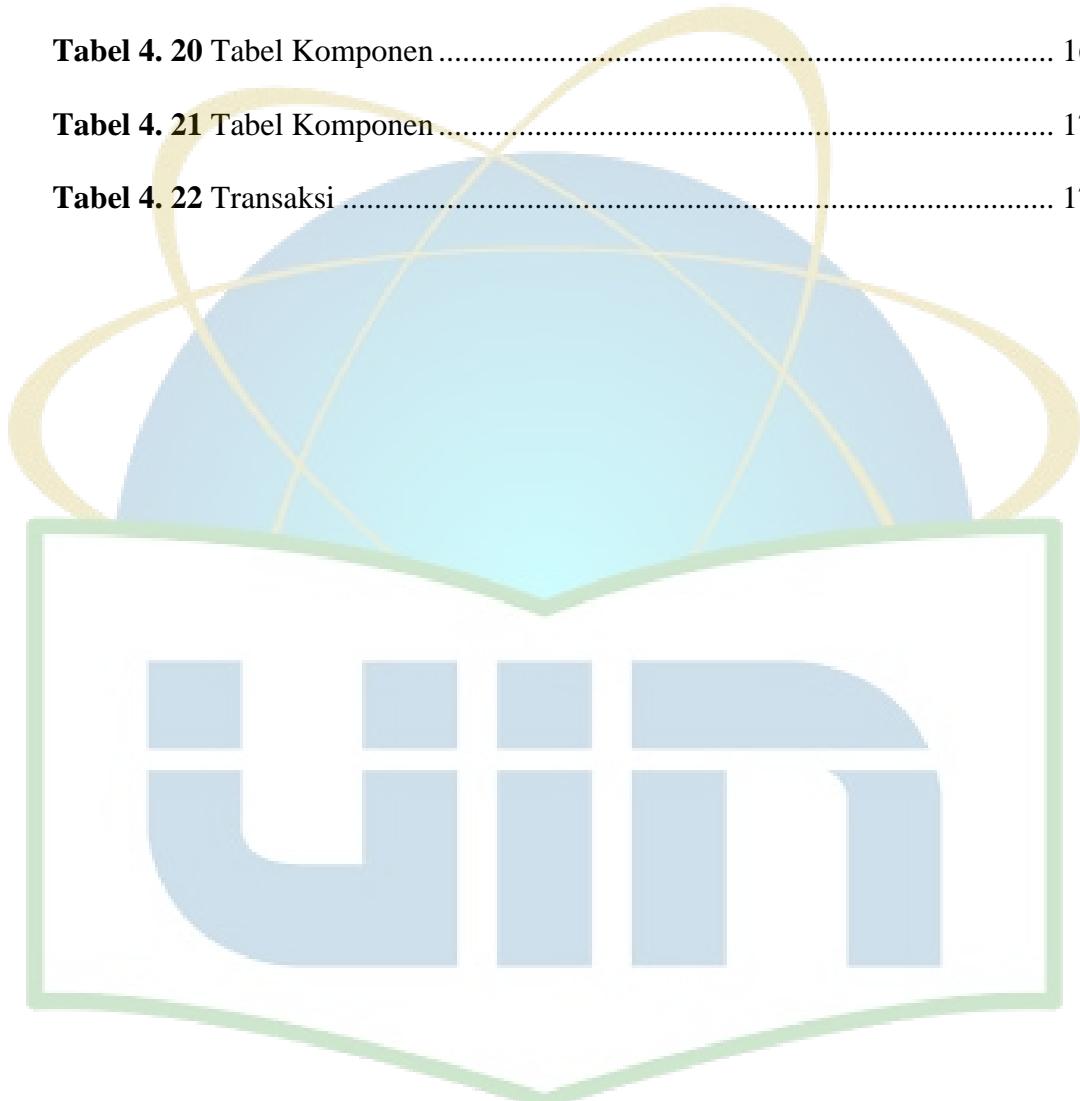
<b>Gambar 4. 15</b>	<i>Activity Diagram</i> Approve Tagihan Pencairan.....	140
<b>Gambar 4. 16</b>	<i>Activity Diagram</i> Input Perintah Pencairan .....	142
<b>Gambar 4. 17</b>	<i>Activity Diagram</i> Surat Perintah Pembayaran .....	143
<b>Gambar 4. 18</b>	<i>Activity Diagram</i> Logout .....	144
<b>Gambar 4. 19</b>	<i>squence Diagram</i> Login .....	145
<b>Gambar 4. 20</b>	<i>Sequence Diagram</i> Manage Anggaran .....	146
<b>Gambar 4. 21</b>	<i>Sequence Diagram</i> Manage User .....	148
<b>Gambar 4. 22</b>	<i>Squence Diagram</i> Manage Detail Anggaran .....	150
<b>Gambar 4. 23</b>	<i>Squence Diagram</i> Pengajuan Kegiatan .....	152
<b>Gambar 4. 24</b>	<i>Sequence Diagram</i> Monitoring Anggaran.....	154
<b>Gambar 4. 25</b>	<i>Sequence Diagram</i> Status Berkas.....	155
<b>Gambar 4. 26</b>	<i>Sequence Diagram</i> Validasi Pengajuan Kegiatan .....	156
<b>Gambar 4. 27</b>	<i>Sequence Diagram</i> Upload Berkas Pencairan .....	158
<b>Gambar 4. 28</b>	<i>Sequence Diagram</i> Approve Tagihan Penciaran.....	160
<b>Gambar 4. 29</b>	<i>Sequence Diagram</i> Perintah Pencairan.....	162
<b>Gambar 4. 30</b>	<i>Sequence Diagram</i> Surat Perintah Pembayaran .....	163
<b>Gambar 4. 31</b>	<i>Sequence Diagram</i> Logout .....	164
<b>Gambar 4. 32</b>	<i>Class Diagram</i> .....	165
<b>Gambar 4. 33</b>	<i>Mapping Class Diagram</i> .....	166
<b>Gambar 4. 34</b>	<i>interface login</i> .....	173
<b>Gambar 4. 35</b>	<i>Interface Admin</i> .....	174
<b>Gambar 4. 36</b>	<i>Interface Admin</i> .....	175
<b>Gambar 4. 37</b>	<i>Interface Admin</i> .....	176

<b>Gambar 4. 38</b>	<i>Interface Kabag</i>	177
<b>Gambar 4. 39</b>	<i>Interface Pelaksana Kegiatan</i>	178
<b>Gambar 4. 40</b>	<i>Interface Pelaksana Kegiatan</i>	179
<b>Gambar 4. 41</b>	<i>Interface Pelaksana Kegiatan</i>	180
<b>Gambar 4. 42</b>	<i>Interface Verifikator</i>	181
<b>Gambar 4. 43</b>	<i>Interface Verifikator</i>	182
<b>Gambar 4. 44</b>	<i>Interface Verifikator</i>	183
<b>Gambar 4. 45</b>	<i>Interface Bendahara Pengeluaran Pembantu (BPP)</i>	184
<b>Gambar 4. 46</b>	<i>Interface Bendahara Pengeluaran Pembantu (BPP)</i>	185
<b>Gambar 4. 47</b>	<i>Interface Bendahara Pengeluaran Pembantu (BPP)</i>	186
<b>Gambar 4. 48</b>	<i>Interface Operator Bendahara</i>	187
<b>Gambar 4. 49</b>	<i>Interface Operator Bendahara</i>	188
<b>Gambar 4. 50</b>	<i>Interface Operator Bendahara</i>	189

## DAFTAR TABEL

<b>Tabel 2. 1</b> Analysis Sistem .....	44
<b>Tabel 2. 2</b> Simbol-Simbol Use Case Diagram .....	50
<b>Tabel 2. 3</b> Simbol-Simbol Activity Diagram .....	52
<b>Tabel 2. 4</b> Simbol-Simbol Sequence Diagram .....	55
<b>Tabel 2. 5</b> Simbol-Simbol Class Diagram.....	59
<b>Tabel 2. 6</b> Contoh Spesifikasi Database .....	68
<b>Tabel 4. 1</b> Identifikasi Actor.....	96
<b>Tabel 4. 2</b> Use Case Login .....	103
<b>Tabel 4. 3</b> Use Case Manage User .....	104
<b>Tabel 4. 4</b> Use Case Manage Anggaran .....	106
<b>Tabel 4. 5</b> Use Case Manage Detail Kegiatan.....	108
<b>Tabel 4. 6</b> Use Case Pengajuan Kegiatan.....	113
<b>Tabel 4. 7</b> Narasi Use Case Monitoring Anggaran.....	115
<b>Tabel 4. 8</b> Narasi Use Case Status Berkas.....	116
<b>Tabel 4. 9</b> Narasi Use Case Validasi Pengajuan Kegiatan .....	117
<b>Tabel 4. 10</b> Narasi Use Case Upload Berkas Tagihan Pecairan .....	119
<b>Tabel 4. 11</b> Narasi Use Case Aprove Tagihan Pencairan.....	120
<b>Tabel 4. 12</b> Narasi Use Case Perintah pencairan.....	122
<b>Tabel 4. 13</b> Narasi Use Case Surat Perintah Pembayaran .....	123
<b>Tabel 4. 14.</b> Narasi Use Case Logout .....	124
<b>Tabel 4. 15</b> Tabel User .....	127

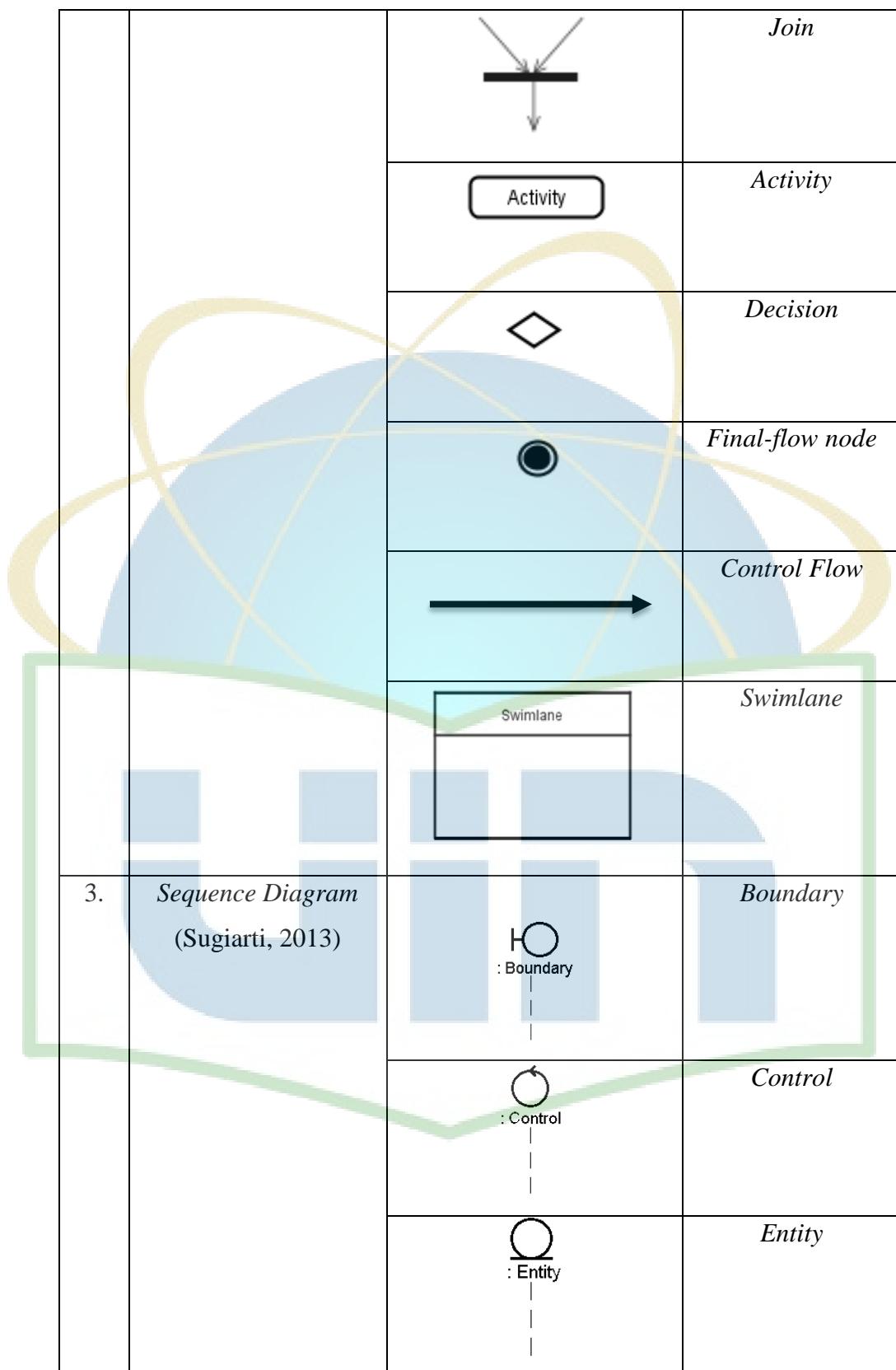
<b>Tabel 4. 16</b> Tabel Level.....	167
<b>Tabel 4. 17</b> Tabel Kegiatan .....	168
<b>Tabel 4. 18</b> Tabel Sub Kegiatan .....	168
<b>Tabel 4. 19</b> Tabel Program.....	169
<b>Tabel 4. 20</b> Tabel Komponen .....	169
<b>Tabel 4. 21</b> Tabel Komponen.....	170
<b>Tabel 4. 22</b> Transaksi .....	171



## DAFTAR SIMBOL

### DIAGRAM UML (*Unified Modelling Language*)

No.	Diagram	Simbol	Nama
1	<i>Usecase Diagram</i> (Dennis, Wixom, & Tegarden, 2012)		<i>Usecase</i> <i>Aktor</i> <i>Asosiasi</i> <i>Extend</i> <i>include</i> <i>Subject boundary</i>
2.	<i>Activity Diagram</i> (Dennis, Wixom, & Tegarden, 2012)		<i>Initial node</i> <i>Fork</i>



			Aktor
			stimulus
4.	<p><i>Class Diagram</i> (Dennis, Wixom, &amp; Tegarden, 2012)</p>		Class
			Association
			Generalization
			Dependency
			Aggregation

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Di era digital pada saat ini banyak perkembangan teknologi dan ilmu pengetahuan yang mempengaruhi banyaknya tuntutan keinginan dari pada sebuah informasi. Bukan hanya kebutuhan sekedar informasi saja namun tuntutan bahwa informasi itu datangnya cepat, cepat dan akurat. Semakin banyaknya tuntutan itu maka memicu persaingan bisnis dalam mengembangkan teknologi informasi. Karena teknologi informasi itu sangat penting bagi kelangsungan dalam proses bisnis dan penunjang pekerjaan seseorang. Di sebuah lembaga pemerintahan atau pun perusahaan sangat dibutuhkan adalah sebuah sistem. Sistem yang membantu mempermudah dan mempercepat kinerja kerja membuat pekerjaan menjadi praktis dan tidak akan memakan waktu.

Dan karena informasi itu sangatlah penting bagi setiap pelaku-pelaku usaha atau perusahaan dan lembaga pemerintahan dalam mengatur strategi untuk mencapai suatu tujuan. Dengan banyaknya informasi atau data data yang dibutuhkan di suatu perusahaan atau lembaga pemerintahan. Maka, dari itulah dibutuhkan suatu kebijakan mengawal atau memantau aktifitas-aktifitas yang ada dalam pengelolaan aplikasi tersebut. Karena sangat mempengaruhi ketidak sesuaian perencanaan dan implementasi. Aktifitas tersebut biasa di kenal dengan sebutan *Monitoring*.

Dalam pengertian para ahli Sistem adalah seperangkat elemen yang membentuk kumpulan atau prosedur-prosedur atau bagan-bagan pengolahan yang mencari suatu tujuan bagian atau tujuan bersama dengan mengoperasikan data atau barang pada waktu sebuah Sistem dalam kegiatan bisnisnya. Dalam pengertiannya Sistem adalah Murdick, R. G (1991:27), Sistem rujukan tertentu untuk menghasilkan informasi dan energi dan barang. Peran dari sistem itu ialah mengolah data menjadi informasi. Pada zaman sekarang informasi itu sangat penting bagi dunia bisnis maupun dunia pemerintahan. Di dalam perkembangannya banyak perusahaan dan khususnya lembaga pemerintahan mengembangkannya untuk meningkatkan efektifitas dan kinerja suatu instansi. Dari pengumpulan data yang dikelola sedemikian rupa untuk menjadi suatu informasi yang tepat dan bermanfaat.

Berdasarkan Peraturan Menteri Keuangan Republik Indonesia Nomor 190/PMK.05/2012 Tentang tata cara pembayaran dalam rangka pelaksanaan anggaran pendapatan belanja negara Bab 1 Ketentuan umum Pasal 1. Tentang tata cara pembayaran dalam rangka pelaksanaan anggaran pendapatan dan pelaksanaan anggaran pendapatan dan belanja Negara.

Kementerian Agama Republik Indonesia adalah salah satu lembaga pemerintahan yang bergerak di bidang keagaman Republik Indonesia. Di ruang lingkup kementerian Agama Khususnya di Sekretariat Jendral Kementerian Agama dibagi beberapa biro, Satu Biro dibagi beberapa Kabag, Kasubag dan Pelaksana Kegiatan. Dari satu Sekretariat terdapat Bendahara Pengeluaran (BP) yang mengatur pengajuan pencairan keuangan ke Kementerian Keuangan dan

di biro-biro ada Badan Pengeluaran Pembantu (BPP) yang bertugas mengatur pengajuan berkas ke Biro keuangan. Pekerjaan yang membutuhkan tingkat kehati-hatian yang lebih di karenakan menyangkut dengan keuangan negara.

Di Ketuai oleh Pejabat Pembuat Surat Perintah Membayar (PPSPM) Kepala Biro keuangan dan Pejabat Pembuat Komitmen (PPK) yang di masing-masing Biro yang memiliki Sertifikat menjadi PPK. dan oleh sebab itu di butuhkannya sistem *Pengolah Keuangan* agar dapat mengontrol peredaran uang tersebut yang nantinya dapat di pertanggung jawabkan oleh staf-stafnya.

Kelengkapan berkas untuk diajukan harus meliputi, nominal uang yang diajukan, kwitansi, surat tugas, Surat Keputusan (SK) dan Daftar nominative peserta kegiatan. Disetiap Biro Khususnya Biru Hokum dan Kerja sama Luar Negeri memiliki empat bagian yang di pimpin Kepala bagian. Disetiap bagian bagian memiliki pelaksana kegiatan lalu berkas itu lanjut ke verifikator, Bendahara Pengeluaran Pembantu, Operator bendahara, dan Pejabat Pembuat komitmen. dengan perincian sebagai berikut:

Biro Hukum dan Kerja sama Luar Negeri mempunyai anggaran berjumlah Rp. 13.3 miliar yang masing-masing dibagi ke empat bagian.

Dari anggaran yang telah diberikan tersebut, setiap bagian dibagi lagi menjadi tiga layanan, Layanan Perkantoran, Layanan Hukum, Layanan Luar Negeri. Contoh dari satu bagian yang berjumlah 2.6 miliar memiliki mata kegiatan, dari satu mata kegiatan meiliki sub mata kegiatan. Diantaranya:

1. Layanan perkantoran

a. Pelayanan perkantoran

b. Pemeliharaan Peralatan dan Mesin

2. Layanan Hukum

a. Kordinasi Penyusunan Produk Hukum

- Penyusunan Peraturan Menteri
- Penyusuna Keputusan Menteri
- Penyusunan Rancangan Peraturan Perundangan-Undangan dan Peraturan Pemerintah
- Penyusunan Rancangan Peraturan Presiden, Keputusan Presiden, Naskah Perjanjian

b. Bantuan Hukum

- Penyelesaian Kasus Aset Negara
- Penyelesaian kasus Tata Usaha Negara
- Penyelesaian Kasus Hukum Keagamaan
- Penyelesaian Kasus Hukum di Lingkungan Kementeriaan Agama

c. Pembinaan Di Bidang Hukum

- Pelakasana Gugus Tugas Harian
- Sosialisasi Edukasi Pencegahan dan Penanganan Pornografi
- Penyuluhan Hukum Bagi Aparatur Negara
- Rapat Kordinasi Biro Hukum

- Konsultasi dan Evaluasi Uji Publik Peraturan Perundangan Undangan
  - Konsultasi dan Evaluasi Peraturan Perundangan-Undangan
  - Pembinaan Teknis Kepengacaraan bagi aparatur Negara
  - Orientasi Penyusunan Keputusan Pada Kementerian agama

### 3. Layanan Kerjasama Internasional

- Administrasi dan Fasilitas Kerjasama Luar Negeri
- Pembinaan di Bidang Hukum dan Kerjasama Luar Negeri

Dari sekian miliar anggaran yang dimiliki oleh masing masing bagian, hanya dikelola oleh satu pelaksana bagian. Setiap pelaksana bagian memiliki tugas untuk mengelola keuangan yang nantinya diajukan ke verifikator untuk diuji berkas tersebut apakah lampirannya lengkap dan sesuai dengan mata anggaran. Dari setiap bagian-bagian dapat mengirimkan lebih dari satu berkas dalam satu hari, yang diambil dari satu mata anggaran dan sub mata anggaran. Dan peran seorang verifikator yang menguji kelayakan berkas antara lain, menguji belanja sewa, Honor Pegawai, Jasa Profesi, Perjalanan Dinas dalam dan Luar Kota, Rapat dalam Kantor, Transport, semua itu menyangkut apakah nominal uang sesuai standar *Standar Biaya Masukan (SBM)*, apakah ada daftar nominatif peserta kegiatan, benarkah nomer rekening pegawai dan peserta kegiatan, benarkah bahwa pengadaan konsumsi di rumah makan tersebut, dan masih banyak lagi yang diuji oleh verifikator.

Berkas yang masuk dan membutuhkan proses validasi itu menimbulkan antrian berkas yang masih berbentuk hardcopy yang beresiko hilang dan juga berkas yang diproses tidak sesuai dengan antrian. Terkadang verifikator memproses berkas yang mudah terlebih dahulu dan itu menimbulkan perotes dari pelaksana-pelaksana bagian. Dari kasus itu maka dibutuhkannya aplikasi dari pengajuan berkasnya berbentuk softcopy hasil scan untuk mengantisipasi berkas yang asli hilang dan juga pelaksana bagian bisa mengontrol alur berkas tersebut agar cair tepat waktu dan efisien.

dan Monalisa (2015) mengadakan penelitian tentang sistem informasi *monitoring* perkembangan proyek yang tujuannya untuk mempermudah dalam pemanatauan perkembangan proyek dan mempermudah dalam pengambilan keputusan. Dari penelitian-penelitian sebelumnya, menunjukkan akan pentingnya sistem *monitoring* dalam menunjang berbagai proses bisnis.

Maka dari itu Penulis memilih Kementerian Agama Republik Indonesia sebagai tempat Penelitian Skripsi. Dengan maksud dan tujuannya untuk meningkatkan efektifitas penyebaran informasi menjadi efektif dan efisien dalam pengolah keuangan anggaran Biro hukum Setjen Kementerian Agama Republik Indonesia.

Dalam merumuskan solusi bagi permasalahan tersebut diatas, maka penulis memilih judul “**RANCANG BANGUN SISTEM MONITORING ANGGARAN KEUANGAN BERBASIS WEB PADA BIRO HUKUM DAN KERJASAMA LUAR NEGERI**” (Studi Kasus: Sekretariat Jenderal Kementerian Agama Republik Indonesia).

## 1.2 Identifikasi Masalah

1. Sering terjadi penyerapan anggaran yang tidak sesuai akun belanja, tidak sesuai realisasi pengeluaran belanja dan kurangnya pendokumentasian terhadap berkas pertanggung jawaban.
2. Tidak adanya mekanisme kontrol terhadap berkas pengajuan keuangan.
3. Tidak tersedianya akses bagi setiap pengelola keuangan untuk memonitoring berkas yang diajukan.

## 1.3 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijabarkan di atas bahwa didapat perumusan masalah sebagai berikut:

1. Bagaimana cara membuat *Sistem* yang dapat mengelola penyerapan anggaran dan realisasi anggaran dan pengarsipan berkas di sistem yang akan dibuat ?
2. Bagaimana cara membuat sistem yang dapat mengontrol sirkulasi berkas ?
3. Bagaimana mengembangkan Sistem Informasi Monitoring Anggaran Berbasis Web dapat mempermudah aktor yang terlibat memonitoring anggaran di Biro Hukum dan Kerjasama Luar Negeri Setjen Kementerian Agama ?

#### 1.4 Batasan Masalah

Berdasarkan rumusan masalah diatas, maka batasan masalah yang akan dibahas sebagai berikut:

1. Pengumpulan data dilakukan pada Sekretariat Jenderal Kementerian Agama Republik Indonesia yang dikhkususkan dibidang anggaran, Wilayah penelitian hanya mencangkup wilaya Biro Hukum dan Kerja Sama Luar Negeri.
2. Sistem yang dibangun berbasis Web para pengelola keuangan dapat mengetahui posisi berkas pengajuan keuangan tersebut berada dimana.
3. Metodologi pengembangan sistem yang digunakan yaitu metodologi berorientasi objek dengan metode *Rapid Application Development* (RAD) yang meliputi Fase *Requirement Planning*, Fase *Workshop design*, dan Fase Implementasi. Pada fase implementasi hanya pada tahap pemrograman dan *testing* menggunakan *blackbox testing*.
4. penyajian data dengan menggunakan *Tool, Unified Modelling Language* (UML).

## 1.5 Tujuan Penelitian

### 1.5.1 Tujuan Khusus

Adapun tujuan khusus dari pembuatan sistem dalam Skripsi ini adalah sebagai berikut :

1. Karena sistem yang sudah ada hanya mengacu pada dua kementerian langsung antara Kementerian Agama dan Kementerian Keuangan tidak adanya pengawasan internal. Maka dari itu sistem ini bertujuan untuk pengawasan anggaran di Biro Hukum dan KLN.
2. Sistem yang suadah ada tidak ada mekanisme kontrol terhadap arus berkas, yang berakibat berkas itu tidak dapat di deteksi hilang dimna dan lambatnya perpindahan berkas adanya dmna..
3. Sistem yang sudah ada masih berkendala dalam hak akses diantaranya tidak semua staf hanya pengelola keuangannya saja. Maka dari itu system yang baru ini dapat memisahkan hak akses, dapat mengetahui kode akun yang menjadi tanggung jawab setiap user pemakainya. Bila terjadi kesalahan maka ketahuan siapa yang telah mengaksesnya dari akun masing –masing user.

### 1.5.2 Tujuan Umum

Adapun tujuan umum dalam pembuatan Skripsi, diantara sebagai berikut.

1. Tujuan umum dari penelitian ini adalah menghasilkan Sistem Informasi Monitoring Anggaran Biro Hukum dan Kerjasama Luar Negeri Setjen Kementerian Agama penyerapan anggarannya lebih terstruktur antara akun benja dan realisasi belanja dan semua yang terlibat sebagai pengelola keuangan dapat mudah mengakses sistem dan memonitoring berkas anggaran yang telah diajukan. Menjadikan keluar masuk berkas menjadi tertib administrasi, efisien dan tepat waktu.

### 1.6 Metodelogi Penelitian

Adapun metode penelitian yang dilakukan pada Skripsi ini adalah sebagai berikut :

#### 1. Teknik Pengumpulan Data

Metode-metode yang digunakan dalam pengumpulan data pada praktik kerja lapangan ini antara lain :

##### a. Metode Konsultasi

Melakukan konsultasi dengan pembimbing lapangan mengenai sistem yang akan dibangun.

b. Metode Wawancara

Mengumpulkan data-data dan informasi dengan cara interview atau wawancara yang dilakukan langsung dengan pihak yang bersangkutan.

c. Studi Pustaka

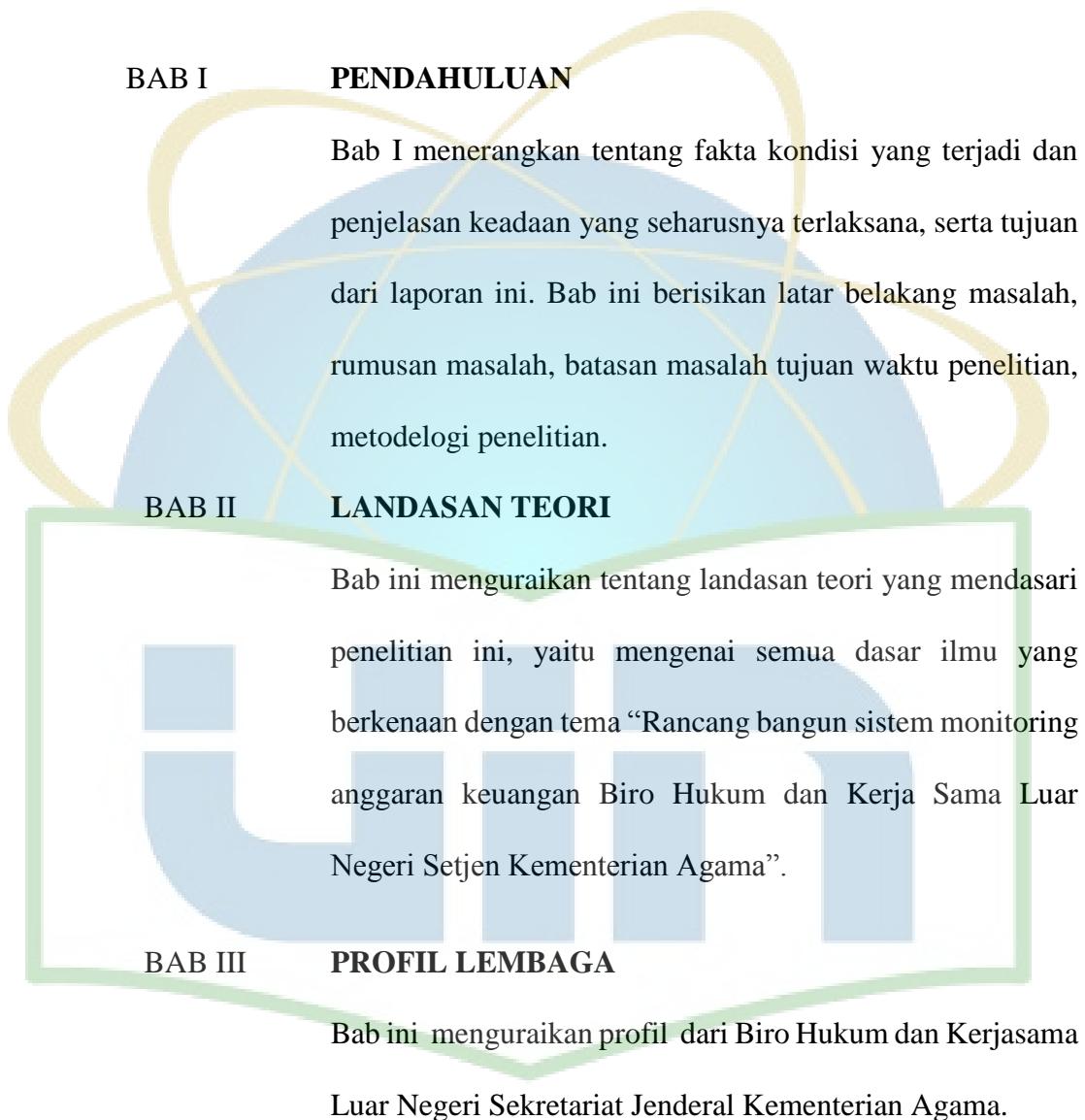
Mencari data-data dengan mempelajari buku, makalah, Terjun langsung atau bekerja di tempat tersebut, skripsi, *browsing* internet ataupun hasil penelitian yang berkaitan tentang pembangunan sebuah *Aplikasi Monitoring*

d. Studi Literatur Sejenis

Pengumpulan data dan informasi dengan melihat hasil Skripsi dari mahasiswa yang pernah melakukan penelitian dengan kasus yang serupa di lingkungan perusahaan dan lembaga pemerintahan atau dari hasil penulisan sejenis yang sudah ada, dimana teori yang sudah ada dapat digunakan sebagai bahan perbandingan dengan penulisan yang akan dibuat. Sehingga penulisan yang dibuat lebih baik dari penulisan sebelumnya.

## 1.7 Sistematika Penulisan

Penyusunan Skripsi penulis membagi dalam beberapa bab yang secara singkat dapat dijelaskan sebagai berikut:



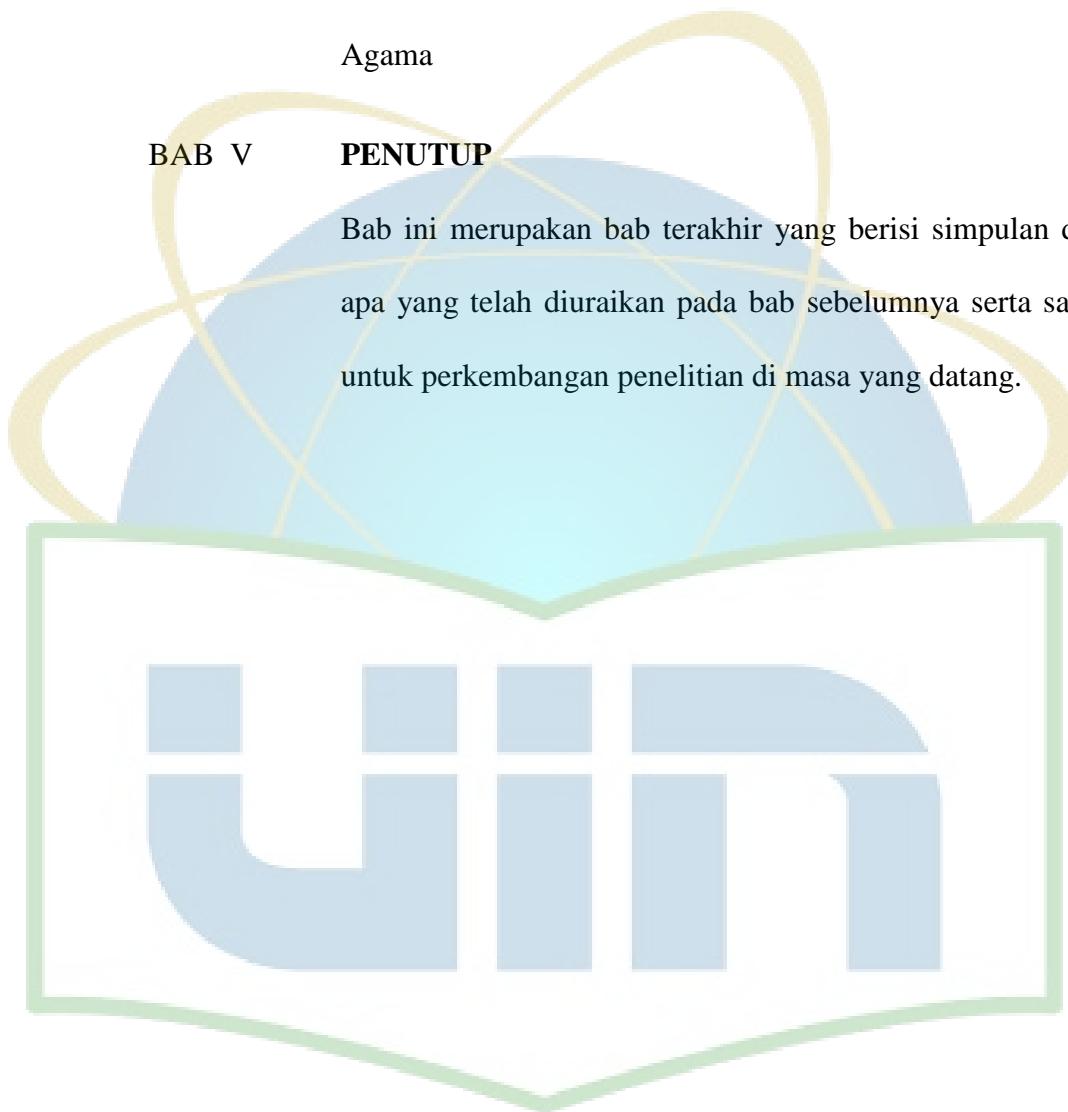
## BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai seluruh tahapan rancang bangun system aplikasi monitoring anggaran keuangan biro Hukum dan Kerja Sama Luar Negeri Setjen Kementerian Agama

## BAB V

### PENUTUP

Bab ini merupakan bab terakhir yang berisi simpulan dari apa yang telah diuraikan pada bab sebelumnya serta saran untuk perkembangan penelitian di masa yang datang.



## BAB II

### LANDASAN TEORI

#### 2.1. Konsep Dasar Sistem Informasi

##### 2.1.1. Pengertian Rancang Bangun

###### 2.1.1.1. Rancang

Perancangan merupakan suatu hal yang penting dalam membangun sebuah aplikasi. Yang bertujuan untuk memberikan gambaran yang jelas kepada pemrograman dan ahli teknik yang terlibat di dalamnya. Perancangan tersebut harus rinci, singkat dan jelas sehingga berguna dan mudah di pahami sehingga mudah digunakan.

Perancangan adalah sebuah Proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses penggerjaanya.

Menurut (Pressman, 2009) perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan.

### 2.1.1.2. Bangun

Menurut (Pressman, 2009) pengertian pembangunan atau bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan.

Jadi dapat disimpulkan bahwa Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

### 2.1.2. Pengertian Sistem

Dalam bidang sistem informasi, sistem diartikan sebagai sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur. Apabila suatu komponen tidak memberikan kontribusi terhadap sistem untuk mencapai tujuan, tentu saja komponen tersebut bukan bagian dari sistem (A. Mulyanto, 2009).

Sistem adalah suatu komponen yang saling bekerjasama dalam satu hubungan yang berurutan dan memiliki kesaling-terkaitan (*interconnection*) antara satu dengan yang lain dalam sebuah prosedur kerja untuk mencapai suatu tujuan. (Syopiansyah Jaya Putra, A'ang Subiyakto)

mengemukakan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi. (HM, Jugiyanto, 2005)

### 2.1.3. Karakteristik Sistem

Menurut (Syopiansyah Jaya Putra, A'ang Subiyakto) suatu sistem mempunyai karakteristik atau ciri-ciri tertentu, yaitu sebagai berikut:

- a. Suatu sistem mempunyai komponen-komponen sistem (*components*) atau subsistem-subsystem. Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian bagian dari sistem.
- b. Suatu sistem mempunyai batas sistem (*boundary*). Yakni daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipasang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

- c. Suatu sistem mempunyai lingkungan luar (*environment*).

Yakni apapun di luar batas sistem yang mempengaruhi operasi sistem yang dapat bersifat menguntungkan atau merugikan sistem tersebut.

- d. Suatu sistem mempunyai penghubung (*interface*). Yakni media penghubung antara satu subsistem dengan

subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem yang lain.

- e. Masukan sistem. Yakni energi yang dimasukkan kedalam sistem. Masukan dapat berupa peralatan (*maintenance input*)

dan masukan sinyal (*signal input*). *Maintenance input* adalah energy yang diproses agar didapatkan keluaran.

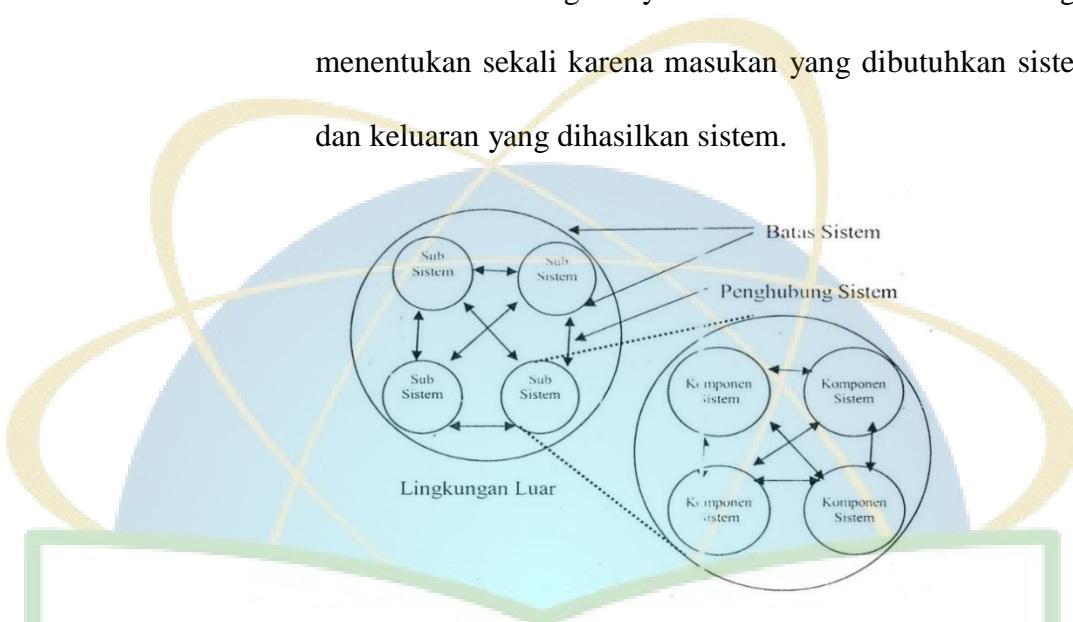
Sebagai contoh di dalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya sedangkan data adalah *signal input* untuk diolah menjadi informasi.

- f. Keluaran sistem. Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi

keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain yang dibutuhkan oleh manajemen.

- g. Suatu sistem mempunyai tujuan (*goal*). Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*).

Kalau sistem tidak mempunyai sasaran maka operasi sistem tidak aka nada gunanya. Sasaran dari sistem sangat menentukan sekali karena masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem.



*Gambar 2. 1 Karakteristik Sistem*

#### 2.1.4. Pengertian Sistem Infomasi

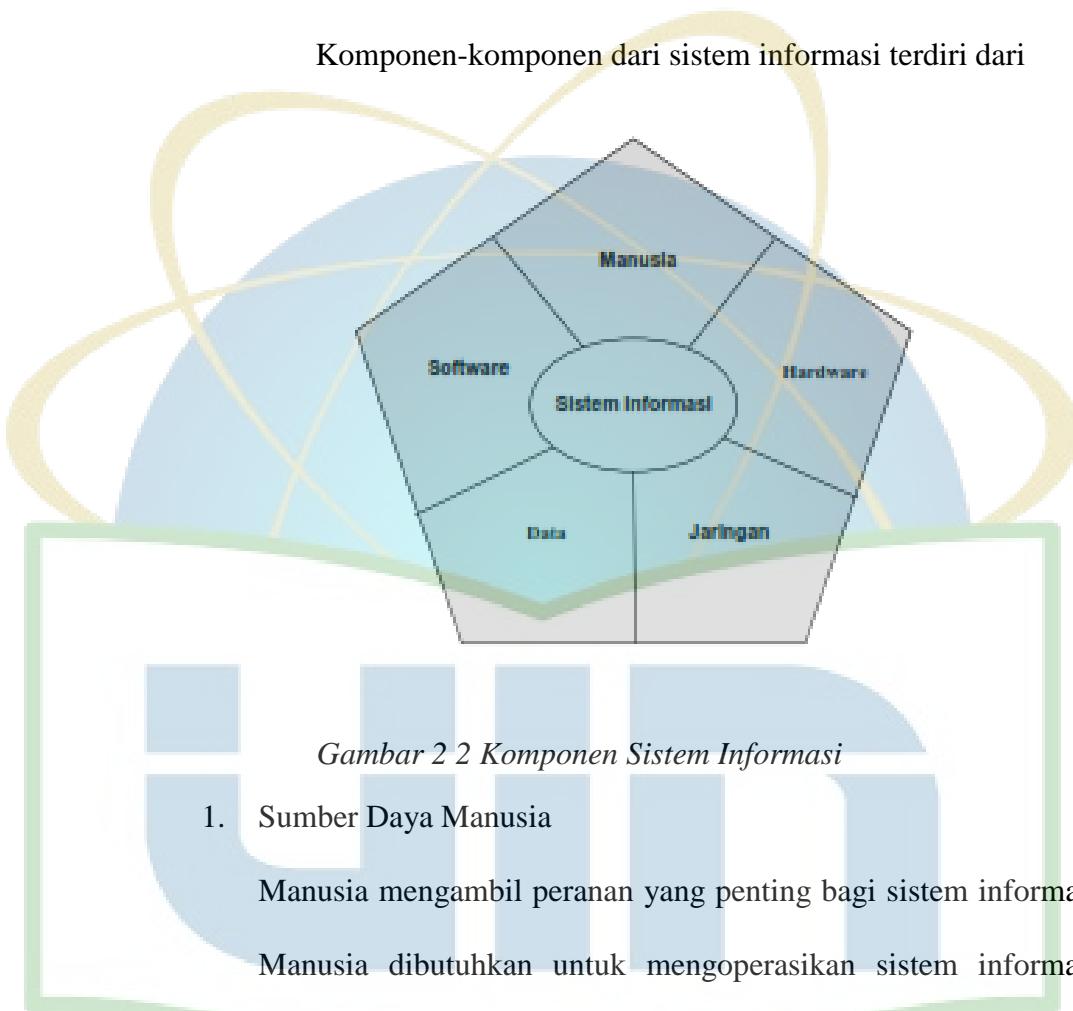
Sistem informasi merupakan suatu komponen yang terdiri dari manusia, teknologi informasi dan prosedur kerja yang memproses, menyimpan, menganalisis dan menyebarkan informasi untuk mencapai suatu tujuan (A. Mulyanto, 2009)

Sistem informasi merupakan gabungan yang terdiri dari manusia, data, proses, dan teknologi informasi yang berinteraksi untuk mengumpulkan, memproses, menyimpan, dan menyediakan sebagai output informasi yang diperlukan untuk mendukung sebuah organisasi (J. L. Whitten and L. D. Bentley, 2007)

Sistem informasi dibangun untuk beberapa tujuan yang berbeda sesuai dengan kebutuhan pengguna dan kebutuhan bisnis, jenis-jenis sistem itu sebagai berikut (K. E. Kendall and J. E. Kendall, 2011)

#### 2.1.4.1. Komponen Sistem Informasi

Komponen-komponen dari sistem informasi terdiri dari



##### 1. Sumber Daya Manusia

Manusia mengambil peranan yang penting bagi sistem informasi.

Manusia dibutuhkan untuk mengoperasikan sistem informasi.

Sumber daya manusia dapat dibedakan menjadi dua kelompok yaitu pengguna akhir dan pakar sistem informasi.

##### 2. Sumber Daya Hardware

Sumber daya *hardware* adalah semua peralatan yang digunakan dalam pemrosesan informasi. Sumber daya *hardware* tidak hanya

sebatas komputer saja, melainkan semua media data seperti lembaran kertas dan *disk* magnetik atau optikal.

### 3. Sumber Daya *Software*

Sumber daya *software* adalah semua rangkaian perintah (instruksi) yang digunakan untuk memproses informasi. Sumber daya *software* tidak hanya berupa program saja, tetapi juga berupa prosedur. Program merupakan sekumpulan instruksi untuk pemrosesan informasi. Sedangkan prosedur adalah sekumpulan aturan yang digunakan untuk mewujudkan pemrosesan informasi dan mengoperasikan perintah bagi orang-orang yang akan menggunakan informasi.

### 4. Sumber Daya Data

Sumber daya data bukan hanya sekedar bahan baku untuk masukan sebuah sistem informasi, melainkan sebagai dasar membentuk sumber daya organisasi.

### 5. Sumber Daya Jaringan

Sumber daya jaringan merupakan media komunikasi yang menghubungkan komputer, pemroses komunikasi, dan peralatan lainnya, serta dikendalikan melalui *software* komunikasi.

## 2.2. Konsep Dasar Aplikasi Berbasis Web

### 2.2.1. Pengertian Aplikasi

Aplikasi adalah merupakan satu *unit* perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, *game*, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia. (Permana, 2007)

### 2.2.2. Pengertian Website

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada website disebut dengan web page dan link dalam website memungkinkan pengguna bisa berpindah dari satu page ke page lain (*hyper text*), baik diantara page yang disimpan dalam server yang sama maupun server diseluruh dunia. Pages diakses dan dibaca melalui browser seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome dan aplikasi browser lainnya. (Hakim Lukmanul, 2004)

### 2.2.3. Pengertian Aplikasi Berbasis Web

Aplikasi Berbasis *web* adalah jenis aplikasi yang menggunakan arsitektur *client-server*. Pada jenis arsitektur ini, sebuah program *client* terhubung pada sebuah *server* untuk informasi yang dibutuhkan untuk melengkapi tugas-tugas yang telah diset oleh *user* (Jugiyanto, 2001).

Ada yang disebut aplikasi *thin client*, dan ada juga *thick client*.

*Thin client* adalah *client* yang hanya berisikan sedikit dari apa yang di perlukan untuk pengalaman *user*, kebanyakan hanya *interface*, semua logika bisnis, semua data, terkecuali yang disediakan *user*, berada dalam *server*, dan tidak banyak pengolahan diperlukan untuk tugas-tugas *user* yang spesifik.

#### 2.2.3.1. Web Server

Pada dasarnya, *server* menerima permintaan dari para *client web browser* dan kesediaan meresponnya. Beberapa permintaan yang datang dari *client* disertai nama dan alamat *item* yang *client* cari, sebagaimana beberapa data *user* yang disediakan. *Server* menerima permintaan tersebut, mempersesnya, dan kemudian merespons data yang dicari oleh *client* atau sebuah kode *error* yang mengindikasikan bahwa *item* tidak terdapat pada *server* atau jika terjadi beberapa *error* lain (Rahman, 2010).

#### 2.2.3.2. Web Client

Tugas browser adalah menyediakan sebuah *interface* dimana *browser* akan meminta *server* dan menampilkan respon dari *server*. Ketika *user* meminta *server* (sebagai contoh, mendapatkan dokumen, atau mungkin mengirim *submit* yang server dapat mengerti. Begitu *server* telah selesai memproses permintaan dan kemudian mengirim respon, *browser* mengambil data yang di perlukan dari *respon* yang

diberikan *server* dan kemudian *me-rendernya* untuk di tampilkan ke *user* (Rahman, 2010).

Aplikasi berbasis web memiliki beberapa kelebihan dibanding aplikasi *desktop client-server* dan Efektif Jika sistem dijalankan secara *multiuser, deployment* sistem.

### 2.3. Pengertian Monitoring

Monitoring dapat diartikan juga sebagai pengawasan, yaitu proses pengamatan pada pelaksanaan seluruh kegiatan organisasi untuk menjamin agar semua pekerjaan yang dilakukan berjalan normal dan baik sesuai dengan rencana yang telah ditentukan semula. Sedangkan menurut (Syarif, 2004) adalah kegiatan *manager* yang mengusahakan agar pekerjaan terlaksana sesuai rencana yang telah ditetapkan dan dapat mencapai hasil yang dikehendaki.

Monitoring didefinisikan sebagai sebuah fungsi berulang yang menggunakan pengumpulan data secara sistematis tentang indikator tertentu untuk memberikan pihak manajemen dan pemangku kepentingan utama dari sebuah proyek yang sedang berjalan dengan indikasi tingkat kemajuan dan pencapaian tujuan dan kemajuan dalam penggunaan dana yang dialokasikan. (PSC, 2016)

Monitoring mengacu pada pengukuran kemajuan apakah masukan menghasilkan keluaran yang sesuai dengan rencana. Dengan kata lain monitoring melihat apakah sebuah proyek tersebut sesuai dengan rancangan awal yang telah direncanakan. (EvaluationToolbox, 2016)

### 2.3.1. Peranan Pengawasan

Secara umum ada beberapa manfaat yang diperoleh dengan diberlakukannya pengawasan pada suatu organisasi, yaitu (I. Fahmi,2012)

- a. Pengawasan memiliki peran penting terutama dalam memastikan setiap pekerjaan terlaksana sesuai dengan direncanakan.
- b. Disamping itu pengawasan juga memiliki peran dalam membantu manajer dalam mengawal dan mewujudkan keinginan visi dan misi perusahaan ,dan tidak terkecuali telah menempatkan manajer sebagai pihak yang memiliki wewenang sentral di suatu organisasi.
- c. Pengawasan bernilai positif dalam membangun hubungan yang baik antara pimpinan dan karyawan. Karena manajer yang efektif menggambarkan pengawasan untuk membagi-bagi informasi, memuji pelaksanaan yang baik dan melihat mereka yang memerlukan bantuan serta menentukan bantuan jenis apa yang mereka perlukan.
- d. Pengawasan yang baik memiliki peran dalam menumbuhkembangkan keyakinan para *stakeholders* pada organisasi. *Stakeholders* adalah mereka yang memiliki kepedulian tinggi pada organisasi. Mereka yang dikategorikan investor, akuntan publik, akademisi, lembaga penilai, karyawan.

### 2.3.2. Tujuan Monitoring

Monitoring digunakan untuk berbagai tujuan. Berikut ini Adalah tujuan utama dari monitoring.( PSC, 2016)

#### a. Manajemen Pengambilan Keputusan

Monitoring meningkatkan proses manajerial dan memberikan bukti dalam pengambilan keputusan. Monitoring tidak bisa menggantikan praktek manajemen yang baik, melainkan menambah dan melengkapi manajemen. Beberapa contoh monitoring yang digunakan dalam konteks ini adalah keputusan tentang alokasi sumber daya, pilihan antara strategi bersaing untuk mencapai tujuan yang sama, keputusan kebijakan, dan keputusan pada desain dan pelaksanaan program.

Keakuratan informasi dan cara monitoring sangatlah penting untuk mendukung manajemen dalam proses pengambilan keputusan mereka.

#### b. Pembelajaran Organisasi

Monitoring dapat digunakan sebagai alat penelitian untuk mengeksplorasi rancangan suatu program ataupun solusi untuk memecahkan masalah apakah suatu rancangan program dan proses operasional akan menghasilkan suatu nilai yang terbaik. Monitoring menghasilkan informasi temuan yang bisa dijadikan untuk pembelajaran.

#### c. Akuntabilitas

Tujuan monitoring dalam hal ini dapat menjelaskan persepsi bahwa monitoring adalah "kebijakan". Monitoring dalam hal ini yaitu bagaimana seseorang yang telah diberi suatu kepercayaan menggunakan wewenang atas kepercayaan yang telah diterimanya. Monitoring memberikan informasi, dengan cara yang terstruktur dan formal, yang memungkinkan pengawasan kegiatan pelayanan publik di semua tingkat.

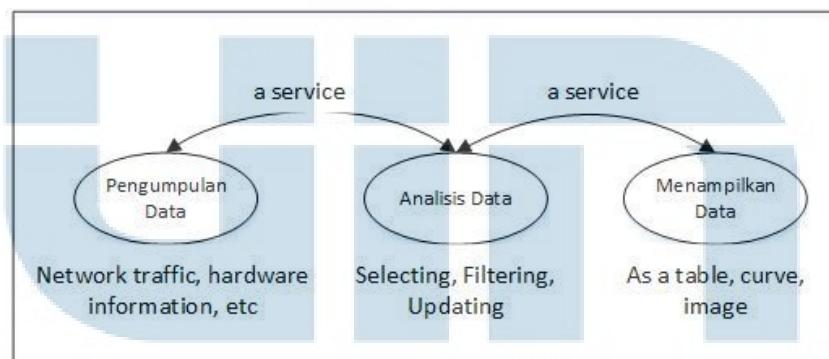
### 2.3.3. Prinsip-prinsip Monitoring

Seorang pimpinan atau manajer sangat memerlukan prinsip pengawasan dalam membandingkan rencana dengan pelaksanaan adalah sebagai berikut.( T. H. Handoko, 2008)

- a. Prinsip perencanaan, merupakan suatu standar atau alat pengukur dari pada suatu pekerjaan sesuai dengan rencana, yaitu apakah pelaksanaan yang direncanakan berhasil atau tidak.
- b. Prinsip wewenang, merupakan suatu kegiatan pemimpin dalam memberikan kepercayaan kepada bawahan dalam melakukan sistem pengawasan. Wewenang dan intruksi-intruksi yang jelas harus dapat diberikan kepada bawahan dalam melakukan sistem pengawasan.
- c. Prinsip tercapainya tujuan. Pengawasan yang ditujukan kearah tercapainya tujuan, yaitu dengan mengadakan perbaikan untuk menghindari penyimpangan dari rencana yang disusun sebelumnya.

- d. Prinsip efisiensi. Pengawasan dikatakan efisien apabila dapat menghindarkan penyimpangan dari rencana, sehingga tidak menimbulkan hal-hal lain yang diluar dugaan.
- e. Prinsip tanggung jawab. Pelaksanaan pengawasan yang efektif dan efisien menurut tanggung jawab penuh dari seorang pemimpin atau manajer terhadap pelaksanaan rencana organisasi.
- f. Prinsip masa depan. Kegiatan pengawasan yang efektif dan efisien harus ditunjukan ke arah pencegahan penyimpangan perencanaan yang akan terjadi baik pada waktu sekarang maupun pada masa yang akan datang.

#### 2.3.4. Sistem Monitoring



Gambar 2 3 Tahapan Sistem Monitoring

Sistem monitoring merupakan suatu proses untuk mengumpulkan data dari berbagai sumber daya. Biasanya data yang dikumpulkan merupakan data yang *realtime*. Secara garis besar tahapan dalam sebuah

sistem monitoring terbagi ke dalam tiga proses besar seperti yang terlihat pada gambar dibawah ini. (L. Rizqi, 2014)

## 2.4. Pengertian Anggaran

Anggaran Negara merupakan rencana keuangan pemerintah dalam suatu waktu tertentu , biasanya dalam satu tahun mendatang, yang satu pihak memuat jumlah pengeluaran setinggi-tingginya untuk membiayai tugas-tugas negara di segala bidang, dan di lain pihak memuat jumlah penerimaan negara yang diperkirakan dapat menutup pengeluaran tersebut dalam periode yang sama ( Sumarsono Sonny, 2010)

### 2.4.1. Jenis Anggaran

Menurut (Sumarsono Sonny, 2010) ada beberapa jenis anggaran yang diantaranya sebagai berikut:

#### 1. Anggaran Oprasional

Anggaran Oprasioanal digunakan untuk merencanakan kebutuhan sehari-hari dalam menjalankan pemerintah. Pengeluaran pemerintah yang dapat dikategorikan dalam anggaran oprasional adalah *belanja rutin.*

*Belanja Rutin* adalah pengeluaran yang manfaatnya hanya untuk satu tahun anggaran dan tidak dapat menambah aset atau kekayaan bagi pemerintahan. Disebut *rutin* karena sifat pengeluaran tersebut berulang-ulang dan ada disetiap tahun.

## 2. Anggaran Modal/Investasi

Anggaran modal menunjukkan rencana jangka panjang dan pembelanjaan atas aktiva tetap. Seperti gedung, peralatan, kendaraan, perabotan, dan sebagainya. Pengulangan yang besar biasanya adalah pengeluaran yang cenderung melebihi satu tahun anggaran dan dapat menambah aset, kekayaan pemerintah dan juga akan menambah anggaran rutin untuk biaya operasional dan pemeliharaan.

### 2.4.2. Fungsi Anggaran

Fungsi anggaran sebagai berikut (Sumarso Sonny, 2010)

1. Anggaran merupakan hasil akhir proses penyusunan rencana kerja.
2. Anggaran merupakan catak biru aktivitas yang akan dilaksanakan di masa mendatang.
3. Anggaran sebagai alat komunikasi intern yang menghubungkan berbagai unit kerja dan mekanisme kerja antar atasan dan bawahan
4. Anggaran sebagai alat pengendalian unit kerja.
5. Anggaran sebagai alat motivasi dan persuasi tindakan efektif.
6. Anggaran sebagai alat motivasi dan persuasi tindakan efektif dan efisien dalam pencapaian visi organisasi.
7. Anggaran merupakan instrumen politik.
8. Anggaran merupakan instrumen kebijakan fiskal.

### 2.4.3. Pelaksanaan Anggaran

Menurut (Suarsono Sonny, 2010) Pelaksanaan Anggaran Direktorat Jenderal Anggaran (DJA) dan Departemen / Lembaga membebas rincian tentang penyusunan DIK dan indeks satuan biaya yang disusun oleh tim independen. Hasil pembahasan tersebut didokumentasikan ke dalam dokumen-dokumen berikut :

#### 1. Dokumen Isian Kegiatan (DIK)

DIK merupakan dokumen anggaran yang berlaku sebagai otorisasi untuk pengeluaran rutin pada masing-masing unit organisasi pada Departemen/Lembaga yang dirinci ke dalam belanja pegawai dan non operasional.

#### 2. Daftar Isian Proyek (DIP)

DIP merupakan dokumen anggaran yang berlaku sebagai otorisasi untuk pengeluaran pembangunan untuk masing-masing proyek pada Departemen/Lembaga yang dirinci kedalam belanja modal dan penunjang.

#### 3. Surat Pengesahan Alokasi Anggaran Rutin (SPAAR)

SPAAR adalah dokumen yang menetapkan besaran alokasi anggaran rutin untuk setiap kantor/satuan kerja departemen teknis di daerah yang selanjutnya akan dibahas antara kantor wilayah DJA dan instansi vertikal Departemen/Lembaga Untuk kemudian dituangkan dalam DIK.

#### 4. Surat Pengesahan Alokasi Anggaran Pembangunan (SPAAP)

SPAAP adalah dokumen yang menetapkan besaran alokasi anggaran pembangunan untuk setiap proyek/bagian proyek yang selanjutnya akan dibahas antara kantor Wilayah DJA dengan instansi vertikal/dinas untuk kemudian dituangkan dalam DIP.

#### 5. Surat Keputusan Otorisasi (SKO)

SKO adalah dokumen otorisasi untuk penyediaan dana kepada Departemen, Lembaga, Pemerintahan Daerah dan pihak lain yang berhak baik untuk rutin maupun pembangunan yang tidak dapat ditampung dengan DIK atau DIP.

### 2.5. Pengertian Keuangan Negara

Keuangan Negara adalah semua hak dan kewajiban negara yang dapat dinilai dengan uang, serta segala sesuatu baik berupa uang maupun berupa barang yang dapat dijadikan milik negara berhubungan dengan pelaksanaan hak dan kewajiban (Suarsono Sonny, 2010).

#### 2.5.1. Ruang Lingkup Keuangan Negara

(Suarsono Sonny, 2010) Ruang Lingkup Keuangan Negara Meliputi:

1. Hak negara memungut pajak, mengeluarkan dan mengedarkan uang dan melakukan pinjaman.
2. kewajiban negara untuk menyelenggarakan tugas layanan umum pemerintah negara dan membayar tagihan pihak ketiga
3. Penerimaan negara.
4. Pengeluaran negara.

5. Pengeluaran daerah
6. Kekayaan negara/kekayaan daerah yang dikelola sendiri oleh pihak lain berupa uang, surat berharga, barang, serta hak-hak lain yang dapat dinilai dengan uang termasuk kekayaan yang dipisahkan pada perusahaan negara/perusahaan daerah.
7. Kekayaan lain yang dikuasai pemerintah dalam rangka penyelenggaraan tugas pemerintah dan kepentingan umum.

### 2.5.2. Dasar Hukum Keuangan Negara

Dasar hukum keuangan negara dapat dipahami atas tiga penafsiran terhadap pasal 23 UUD 45 yang merupakan landasan konstitusi keuangan negara, yang berbunyi:

“Pengertian keuangan negara diartikan secara sempit, dan untuk itu disebut sebagai keuangan negara dalam arti sempit, yang hanya meliputi keuangan negara yang bersumber pada APBN sebagai sub sistem keuangan negara dalam arti sempit.”

“Penafsiran kedua negara dalam arti luas yang meliputi keuangan negara yang berasal dari APBN, APBD, BUMN, BUMD, dan pada hakikatnya seluruh harta kekayaan negara sebagai suatu sistem keuangan negara.”

Penafsiran Ketiga dilakukan melalui “Pendekatan sistematik dan teleologis atas sosiologi terhadap keuangan negara yang dapat memberikan penafsiran yang relatif lebih akurat dengan tujuannya.”

### 2.5.3. Sistem Pengelolaan Keuangan Negara

#### 1. Sifat Lembaga Pemerintah

Sifat lembaga pemerintah berbedadengan perusahaan yang bertujuan mencari laba. Keinginan mengejar laba tidak inklusif dalam usaha dan kegiatannya. Tidak memiliki secara pribadi akan tetapi secara kolektif oleh seluruh warga negara dan pemilika ini tidak dibuktikan oleh adanya pemilikan saham yang dapat diperjual belikan atau di perdagangkan

#### 2. Sistem Pemerintah Suatu Negara

Sistem pemerintah sangat besar pengarunya terhadap bentuk akuntansi pemerintahan suatu negara. Sebagai gambaran dapat dibandingkan antara akuntansi pemerintahan di Indonesia dan di Amerika serikat, pemerintah Amerika yang menganut sistem pemerintahan federasi yang di tandai dengan adanya negara-negara yang

bersifat otonom

#### 3. Mekanisme Pengurusan Keuangan dan Sistem Anggaran

Akuntansi pemerintah pada dasarnya adalah akuntasi kokro yang bersifat mencatat dan melaporkan realisasi anggaran suatu negara. Berbagai undang-undang dan ketentuan yang mengatur mekanisme pengelolaan keuangan dan sistem anggaran suatu negara dengan sendirinya mengikat pula sifatnya bagi penyelenggara akuntansi pemerintah di negara itu.

## 2.6. Metode Pengumpulan Data

Pengumpulan data adalah suatu proses pengadaan data primer untuk keperluan penelitian. Pengumpulan data adalah prosedur yang sistematis dan standar untuk memperoleh data yang diperlukan. (M.Nazir,2009).

### 2.6.1. Studi Literatur

Pengumpulan data dengan teknik survei terhadap data yang telah ada dengan menggali teori-teori yang telah berkembang dalam bidang ilmu yang berkepentingan, mencari metode-metode serta teknik penelitian baik dalam mengumpulkan data atau dalam menganalisis data yang telah pernah digunakan oleh peneliti-peneliti terdahulu (Nazir, 2009). Dalam hal ini penulis mendapatkan sumber-sumber teori dari referensi perpustakaan.

Dengan mencari sumber data sekunder yang mendukung penelitian, serta mengetahui sampai mana ilmu yang berhubungan dengan penelitian yang telah berkembang, kesimpulan dan generalisasi yang pernah dibuat, sehingga situasi yang diperlukan dapat diperoleh.

<b>Tabel 2.1</b> Studi Literatur Jurnal	<b>Penulis</b>	<b>Kelebihan</b>	<b>Kekurangan</b>
Aplikasi Sistem Penganggaran dan Pengendalian Anggaran (studi kasus: UNMUH Ponorogo)	1. Bhaga Yanuardo Missa 2. Arifin Puji Widod 3. Tan Amelia	Dalam membuat anggaran menggunakan evaluasi 3 periode, sehingga anggaran yang dibuat bisa tepat guna (tidak terlalu besar dan tidak terlalu kecil).	1. perbandingan antara anggaran dengan realisasi tidak menggunakan satuan persentase. 2. Jenis pengendalian anggaran yang dibahas adalah pengajuan, bukan pengajuan anggaran dan realisasi.
Pengembangan Sistem Informasi Akuntansi Dan Pengelolaan Keuangan Negara (Siapkan) Modul Administrasi Menggunakan Teknologi Java Dan PostgreSQL	1. Wahyudiono, 2. Dr. Eng. Febriyan Samopa, S.Kom, M.Kom, dan 3. Alwi S.E.	1. Aplikasi terhubung dengan sistem lama, sehingga saling singkronisasi 2. Adanya modul penganggaran, pencatatan asset tetap, persediaan, pembayaran dan laporan.	1. Tidak adanya realisasi anggaran yang dapat memonitor anggaran.
Skenario Penentuan Subsidi Silang pada Perencanaan Anggaran Universitas PGRI Adi Buana Surabaya	1. Rizcha Rahmawati Hayuningtyas 2. Arifin Puji Widodo 3. Mochammad Arifin	1. Membahas tentang subsidi silang pada anggaran pada suatu sektor yang <i>minus</i> . 2. Sistem mampu menunjukkan urutan-	1. Tidak dibuat secara <i>online</i> masih secara desktop. 2. Dengan konsep seperti ini dapat merusak pagu yang sudah ditentukan

Pengembangan sistem informasi manajemen surat pada bagian program dan penyusunan anggaran di bagian Meteorology, Klimatologi, dan Geofisika pusat.	Prastuti Damariyanti	sistem yang dibuat sudah terkomputerisasi sehingga pengarsipan surat menjadi lebih baik, rapi, dan lengkap, serta mudah dalam pencarian surat.	Administrator sistem belum dapat mengetahui aktivitas user selama online. Selain itu juga belum dilengkapi dengan template surat atau pembuatan suratnya serta pengimplementasianya pada penelitian ini.
Analisis dan Perancangan Sistem Informasi Akuntasi Realisasi Anggaran pada Direktorat Jendral Pelayanan dan Rehabilitas Sosial.	Asep Kusnaldi	1. Sistem dapat menghasilkan laporan realisasi anggaran.	1. Tidak online sehingga pimpinan tidak bisa mengetahui data saat ini bila tidak dikantor. 2. Tidak adanya saldo awal pada buku besar
Analisis dan Perancangan Sistem Informasi Anggaran pada PT. Thomas Express	Elly Fiah	Aplikasi dibuat dengan perencanaan anggaran serta pengawasan. Adanya laporan laba rugi.	Sistem tidak terhubung dengan internet, sehingga pimpinan tidak bisa control selain hanya di kantor.
Pengembangan sistem Informasi Monitoring dan Evaluasi Terhadap Penggunaan Anggaran dana berbasis Web	Aziza Amelia	berbasis web. Terdapat menu Upload file, sehingga sistem dapat melakukan transfer data/pengiriman data tanpa menggunakan fasilitas email.	beberapa aktor yang dapat bisa mengakses sistem. 2. Monitor pada penggunaan anggaran dana tidak menyeluruh.

Berdasarkan studi literatur tersebut yang membedakan dengan sistem yang peneliti buat dengan judul “Rancang Bangun Sistem Monitoring Anggaran Keuangan Pada Biro Hukum dan KLN saat berjalan.

1. Sistem yang dibangun *real time* sehingga *monitoring* dan evaluasi bisa dilakukan setiap saat.
2. Sistem yang dibangun dapat memonitor anggaran keuangan dengan tiga periode yaitu per-triwulan sehingga penyerapan anggaran bisa tepat guna.
3. Sistem yang dibangun memiliki fungsi untuk melihat setiap relisasi yang berjalan beserta surat pertanggung jawabannya.

### 2.6.2. Wawancara

Yang dimaksud dengan wawancara adalah proses memperoleh keterangan untuk tujuan penelitian dengan cara tanya jawab, sambil bertatap muka antara si penanya atau pewawancara dengan si penjawab atau responden dengan menggunakan alat yang dinamakan *interview guide* (panduan wawancara). (M.Nazir,2009)

### 2.6.3. Observasi

Pengumpulan data melalui pengamatan dan pencatatan oleh pengumpul data terhadap gejala/peristiwa yang diselidiki pada objek penelitian. Pengumpulan data dengan observasi langsung atau dengan pengamatan langsung adalah cara pengambilan data dengan menggunakan mata tanpa ada pertolongan alat standar lain untuk keperluan tersebut. Jadi Observasi adalah

pengumpulan data melalui pengamatan langsung terhadap objek yang ada pada instansi terkait untuk mendapatkan data-data yang diperlukan. (M.Nazir, 2009)

## 2.7. Pengembangan Sistem

### 2.7.1. Pengembangan Sistem Berbasis Objek

Sistem berorientasi objek berfokus pada pengambilan struktur dan perilaku sistem informasi dalam modul kecil yang mencakup data dan proses. Modul kecil ini dikenal sebagai objek (A. Dennis). Pemrograman berorientasi objek berbeda dari pemrograman prosedural tradisional dengan memeriksa objek-objek yang merupakan bagian dari sistem. Setiap objek adalah representasi komputer dari beberapa hal atau peristiwa yang sebenarnya (K. E. Kendall and J. E. Kendall, 2011).

Pemodelan objek merupakan sebuah teknik yang mencoba untuk menyatukan data dan proses ke dalam konsepsi tunggal yang disebut objek. Model-model objek adalah diagram-diagram yang mendokumentasikan sebuah sistem dalam artian objek-objek dan interaksi-interaksinya. Pemodelan objek adalah basis atau dasar dari metodologi-metodologi analisis dan desain berorientasi objek. Teknik pemodelan objek menyajikan penggunaan metodologi dan notasi diagram yang sama sekali berbeda dengan teknik lainnya (J. L. Whitten and L. D. Bentley, 2007).

### 2.7.2. Karakter Dasar Sistem Berorientasi Objek

Berikut ini adalah karakter-karakter dasar dari sistem berorientasi objek (A. Dennis, B. H. Wixom and D. Tegarden, 2012). :

- a. *Class* adalah template umum yang kita gunakan untuk mendefinisikan dan membuat contoh, atau objek tertentu. Setiap objek dikaitkan dengan kelas. Misalnya, semua benda yang menangkap informasi tentang pasien dapat masuk ke dalam kelas yang disebut Pasien, karena ada atribut (misalnya nama, alamat, tanggal lahir, telepon, dan pembawa asuransi) dan metode (misalnya, membuat janji temu, hitung kunjungan terakhir, Mengubah status, dan memberikan riwayat kesehatan) yang dimiliki semua pasien.
- b. *Object* adalah sebuah perwujudan dari sebuah kelas. Dengan kata lain, objek adalah seseorang, tempat, atau benda yang ingin kita ambil informasi. Jika kami membangun sistem janji untuk kantor kedokteran, kelas bisa mencakup dokter, pasien, dan janji. Pasien-pasien tertentu, dianggap sebagai contoh, atau objek, dari kelas pasien. Setiap objek memiliki atribut yang menggambarkan informasi tentang objek, seperti nama pasien, tanggal lahir, alamat, dan nomor telepon. Atribut juga digunakan untuk merepresentasikan hubungan antar objek; misalnya, mungkin ada atribut departemen dalam objek karyawan dengan nilai objek departemen yang menangkap di mana departemen objek karyawan bekerja. Keadaan suatu objek ditentukan oleh nilai atribut dan hubungannya dengan

benda lain pada titik waktu tertentu. Misalnya, pasien mungkin memiliki keadaan baru atau saat ini atau mantan. Setiap objek juga memiliki perilaku. Perilaku menentukan apa yang bisa dilakukan objek. Misalnya, objek janji temu mungkin dapat menjadwalkan janji temu baru, menghapus janji temu, dan menemukan janji yang ada selanjutnya. Dalam pemrograman berorientasi objek, perilaku diimplementasikan sebagai metode.

c. *Methods* adalah cara untuk menerapkan perilaku pada suatu objek. Sebuah metode tidak lebih dari sebuah tindakan yang dapat dilakukan suatu objek. Dengan demikian, sebuah metode analog dengan fungsi atau prosedur dalam bahasa pemrograman tradisional seperti C, COBOL, atau Pascal. data yang mewakili karakteristik *interest* tentang sebuah objek.

d. *Message* adalah informasi yang dikirim ke objek untuk memicu metode. Pesan pada dasarnya adalah sebuah fungsi atau prosedur yang memanggil dari satu objek ke objek lain. Misalnya, jika pasien baru ke kantor dokter, resepsionis mengirimkan pesan buat aplikasi.

Kelas pasien menerima pesan pembuatan dan menjalankan metode `create()`, yang kemudian menciptakan objek baru: Pasien.

e. *Encapsulation* adalah pengemasan beberapa *item* ke satu unit

f. *Inheritance* adalah karakteristik pengembangan sistem informasi, diusulkan dalam pemodelan data pada akhir 1970an dan awal 1980an. Literatur pemodelan data menyarankan penggunaan pewarisan untuk

mengidentifikasi kelas objek yang lebih tinggi, atau lebih umum.

g. *Polymorphism* berarti bahwa pesan yang sama dapat ditafsirkan secara berbeda oleh kelas dari objek yang berbeda. Misalnya, memasukkan pasien berarti sesuatu yang berbeda daripada memasukkan janji temu. Oleh karena itu, potongan informasi yang berbeda perlu dikumpulkan dan disimpan. Kita hanya bisa mengirim pesan ke suatu objek, dan objek itu akan bertanggung jawab untuk menafsirkan pesan dengan tepat. Misalnya, jika seorang seniman mengirim pesan tersebut, gambarkan diri Anda ke objek persegi, objek lingkaran, dan objek segitiga, hasilnya akan sangat berbeda, meskipun pesannya sama.

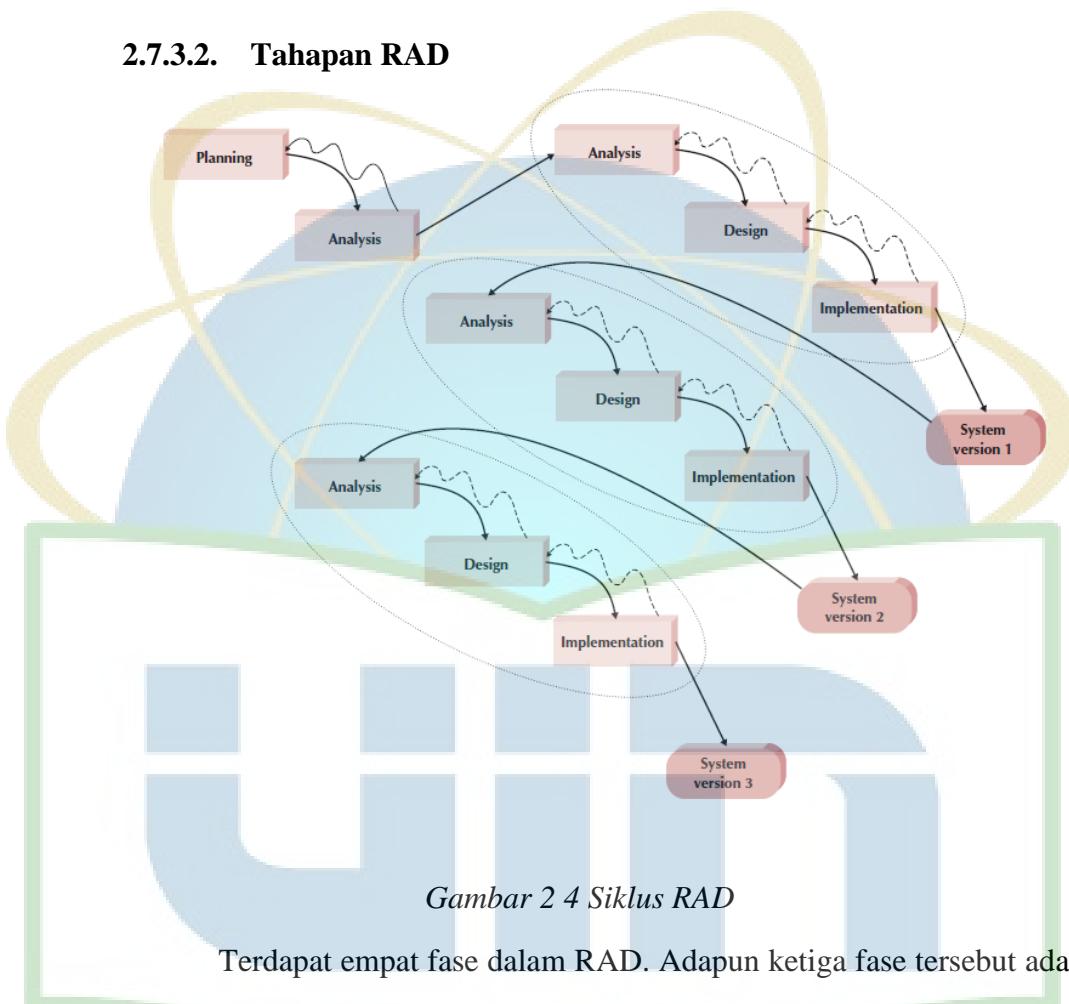
### 2.7.3. *Rapid Application Development (RAD)*

#### 2.7.3.1. Pengertian RAD

Metode pengembangan sistem yang penulis gunakan adalah metode RAD (*Rapid Application Development*). RAD adalah kelas baru dari metodologi pengembangan sistem yang muncul di tahun 1990an. Metodologi berbasis RAD berupaya mengatasi kelemahan kedua metodologi desain terstruktur dengan menyesuaikan fase SDLC agar beberapa bagian sistem berkembang dengan cepat dan masuk ke tangan pengguna. Dengan cara ini, pengguna bisa lebih memahami sistem dan menyarankan revisi yang membawa sistem lebih dekat dengan apa yang dibutuhka (A. Dennis, B. H. Wixom and D. Tegarden, 2012). RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode

pengembangan perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi (K. E. Kendall, 2011).

#### 2.7.3.2. Tahapan RAD



Terdapat empat fase dalam RAD. Adapun ketiga fase tersebut adalah

*Planning, Analysis, Design* dan *Implementation*. Berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi (A. Dennis, B. H. Wixom and D. Tegarden, 2012).

##### 1. ***Planning***

Merupakan tahap proses dasar untuk memahami mengapa sistem informasi ini harus dibangun dan menentukan bagaimana tim

proyek akan membangun proyek itu. Selama inisiasi proyek, nilai bisnis sistem terhadap organisasi diidentifikasi dalam bentuk permintaan sistem. Permintaan sistem menyajikan ringkasan singkat tentang kebutuhan bisnis, dan menjelaskan bagaimana sistem yang mendukung kebutuhan akan menciptakan nilai bisnis.

## 2. *Analysis*

Tahap analisis adalah tahap untuk menjawab pertanyaan-pertanyaan tentang siapa yang akan menggunakan sistem, apa yang akan dilakukan oleh sistem, di mana dan kapan akan digunakan sistem tersebut. Ada tiga tahapan dalam fase analisis ini antara lain:

- a. Membuat strategi analisis untuk diajukan sebagai acuan upaya tim proyek. Strategi seperti itu biasanya mencakup analisis sistem saat ini dan masalahnya, kemudian cara merancang sistem baru. peneliti juga melakukan analisis PIECES (*Performance, Information, Economic, Control, Efficiency* dan *Service*). Kerangka kerja analisis PIECES ialah sebagai berikut:
  - *Performance* (Kinerja) yaitu Kebutuhan untuk memperbaiki atau meningkatkan kinerja.
  - *Information* (Informasi) yaitu kebutuhan untuk memperbaiki atau meningkatkan informasi dan data.
  - *Economic* (Ekonomi) yaitu kebutuhan untuk memperbaiki atau meningkatkan ekonomi, biaya kontrol, atau meningkatkan keuntungan.

- Control (Kontrol) yaitu ebutuhan untuk memperbaiki atau meningkatkan kontrol atau keamanan.
- *Eficiency* (Efisien) yaitu ebutuhan untuk memperbaiki atau meningkatkan efisiensi orang dan proses.
- *Services* (Pelayanan) yaitu ebutuhan untuk memperbaiki atau meningkatkan layanan kepada pelanggan, pemasok, mitra, karyawan dan sebagainya.

#### Contoh Analisis PIECES

**Tabel 2. 1 Analysis Sistem**

Indikator	Keterangan
<i>Performance</i>	Dalam proses kegiatan monitoring anggaran keuangan ini kinerja apa yang perlu ditingkatkan dalam sistem usulan?
<i>Information</i>	Informasi yang seperti apa yang dapat dihasilkan oleh sistem usulan sehingga dapat mengoptimalkan kegiatan monitoring anggaran keuangan?
<i>Economy</i>	Pada bagian mana dari sistem usulan yang mampu meningkatkan laba atau mengurangi biaya pada kegiatan monitoring anggaran keuangan?
<i>Control</i>	Apa yang perlu ditingkatkan pada sistem usulan mengenai <i>control</i> dalam monitoring anggaran keuangan?
<i>Efficiency</i>	Apa yang perlu diperbaiki dan ditingkatkan pada sistem berjalan sehingga memberikan kemudahan bagi pemakai sistem usulan?
<i>Service</i>	Manakah dari beberapa aspek pelayanan sistem berjalan monitoring anggaran keuangan yang perlu ditingkatkan?

- b. Mengumpulkan persyaratan/kebutuhan untuk membuat konsep sistem. Persyaratan bisa bersifat fungsional atau nonfungsional.

Persyaratan fungsional berhubungan langsung dengan proses yang harus dilakukan atau informasi yang harus dikandungnya.

Misalnya, persyaratan yang menyatakan bahwa sistem harus memiliki kemampuan untuk mencari persediaan yang tersedia atau melaporkan biaya aktual dan yang dianggarkan adalah persyaratan fungsional. Persyaratan fungsional mengalir langsung ke dalam pembuatan model fungsional, struktural, dan perilaku yang mewakili fungsionalitas sistem yang berkembang.

Persyaratan nonfungsional mengacu pada sifat perilaku yang harus dimiliki sistem, seperti kinerja dan kegunaan.

Kemampuan untuk mengakses sistem menggunakan *web browser* dianggap sebagai persyaratan nonfungsional.

Persyaratan nonfungsional dapat memengaruhi sisa analisis (model fungsional, struktural, dan perilaku) namun seringkali hanya dilakukan secara tidak langsung. Persyaratan nonfungsional digunakan terutama dalam desain saat keputusan dibuat mengenai antarmuka pengguna, perangkat keras dan perangkat lunak, dan arsitektur fisik dasar sistem. Konsep sistem digunakan untuk dasar pembuatan analisis model bisnis. Kemudian kebutuhan tersebut akan dimodelkan atau digambarkan dengan teknik analisis dan alat bantu tertentu.

Sebagai contoh kebutuhan fungsional dapat dimodelkan dengan menggunakan *data flow diagram*, kamus data, dan spesifikasi proses jika menggunakan analisis tersruktur dan *use case* diagram dan skenario sistem jika menggunakan analisis berorientasi objek.

- c. Analisis, konsep sistem dan model bisnis dikombinasikan untuk membuat proposal sistem, proposal ini akan diajukan kepada pihak yang akan memutuskan apakah proyek dilanjutkan atau tidak dilanjutkan.

### 3. *Design*

Tahap desain memutuskan bagaimana sistem akan beroperasi, dalam hal perangkat keras, perangkat lunak, dan infrastruktur jaringan, antarmuka pengguna, bentuk dan laporan; dan program khusus, *database*, dan *file* yang akan dibutuhkan. Pemodelan desain menggabungkan persyaratan nonfungisional. Artinya, pemodelan desain berfokus pada bagaimana sistem akan beroperasi. Pertama, tim proyek memverifikasi dan memvalidasi model analisis (fungisional, struktural, dan perilaku). Desain kelas dan metode diilustrasikan dengan menggunakan *class diagram*. Selanjutnya, dipetakan untuk merancang *database* atau struktur *file* sebenarnya. Pada saat bersamaan, tim menghasilkan desain antarmuka pengguna. Desain arsitektur fisik dibuat dengan menggunakan *deployment diagram* dan

spesifikasi perangkat keras. Hasil dari semua ini mewakili spesifikasi sistem yang diserahkan ke tim pemrograman untuk implementasi.

#### 4. *Implementations*

Tahap yang biasanya mendapat perhatian yang besar, karena bagi kebanyakan sistem itu adalah tahapan terpanjang dan bagian paling mahal dari proses pembangunan sistem. Fase ini memiliki tiga langkah:

1. Konstruksi sistem. Sistem ini dibangun dan diuji untuk memastikan ia bekerja seperti yang telah dirancang. Karena biaya perbaikan bisa sangat besar, pengujian adalah salah satu langkah yang paling penting dalam implementasi. Banyak organisasi memberikan lebih banyak waktu dan perhatian pada pengujian daripada menulis program.
2. Instalasi Sistem. Instalasi adalah proses dimana sistem lama nonaktifkan dan yang baru aktifkan.
3. Tim analis menetapkan rencana *support* untuk sistem. Rencana ini biasanya mencakup kajian setelah implementasi formal atau informal serta cara sistematis untuk mengidentifikasi perubahan besar dan kecil diperlukan untuk sistem.

### 2.7.3.3. Keuntungan RAD

Beberapa kelebihan yang dimiliki metode RAD antara lain (R. S. Pressman and B. R. Maxim, 2015)

- a. Siklus dan fase pengembangan sistem yang relatif cepat.
- b. Aplikasi dapat terlihat lebih cepat di awal, karena digunakan *prototyping* dengan perancangannya menggunakan *graphic user interface* (GUI).
- c. Memiliki fleksibilitas tinggi karena dibangun atas dasar modularisasi dimana pengembang dapat mendesain ulang atau me-re-use modul dan bersifat *life cycles*.

### 2.7.3.4. Kelemahan RAD

Kekurangan penerapan metode RAD adalah sebagai berikut (K. E. Kendall and J. E. Kendall, 2011).

- a. Dengan metode RAD, penganalisis berusaha mempercepat proyek dengan terburu-buru.
- b. Kelemahan yang berkaitan dengan waktu dan perhatian terhadap detail. Aplikasi dapat diselesaikan secara lebih cepat, tetapi tidak mampu mengarahkan penekanan terhadap permasalahan-permasalahan perusahaan yang seharusnya diarahkan.
- c. RAD menyulitkan *programmer* yang tidak berpengalaman menggunakan perangkat ini di mana *programmer* dan *analyst* dituntut untuk menguasai kemampuan-kemampuan baru

sementara pada saat yang sama mereka harus bekerja mengembangkan sistem.

#### 2.7.4. *Unified Modeling Language (UML)*

UML adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (A. Nugroho, 2010).

UML menyediakan satu set standar alat untuk mendokumentasikan analisis dan desain sistem perangkat lunak. Seperangkat alat UML termasuk diagram yang memungkinkan orang untuk memvisualisasikan pembangunan sistem berorientasi objek, mirip dengan cara satu set *blueprint* yang memungkinkan orang untuk memvisualisasikan pembangunan gedung. Dokumentasi yang dibuat dengan UML menyediakan sarana komunikasi yang efektif antara tim pengembangan dan tim bisnis pada sebuah proyek (K. E. Kendall and J. E. Kendall, 2011).

##### 2.7.4.1. *Use Case Diagram*

*Use Case* adalah salah satu diagram yang ada dalam UML (*Unified Modeling Language*). *Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) aplikasi perangkat lunak yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan aplikasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi atau proses apa saja yang ada didalam sebuah aplikasi

dan siapa saja yang berhak menggunakan fungsi-fungsi itu (R. A. S. and M. Shalahuddin, 2011).

Berikut adalah penjelasan dari simbol-simbol *use case diagram* (A. Dennis, B. H. Wixom and D. Tegarden, 2012). Berikut ini adalah contoh dari *use case diagram* :

**Tabel 2. 2 Simbol-Simbol Use Case Diagram**

Simbol	Deskripsi
 <i>Usecase</i>	Proses utama yang dilakukan sistem dan yang memberikan manfaat kepada aktor. <i>Usecase</i> yang diberi label dengan menggunakan frase kata kerja deskriptif-kata benda
 <i>Actor</i>	Seseorang yang memperoleh manfaat dari dan berada di luar subjek (sistem). Pada dasarnya, aktor mewakili elemen utama di lingkungan tempat sistem beroperasi. Aktor dapat memberikan masukan kepada sistem, menerima output dari sistem, atau keduanya
 <i>Asosiasi</i>	Penghubung interaksi antara seorang aktor dengan <i>use case</i>
	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

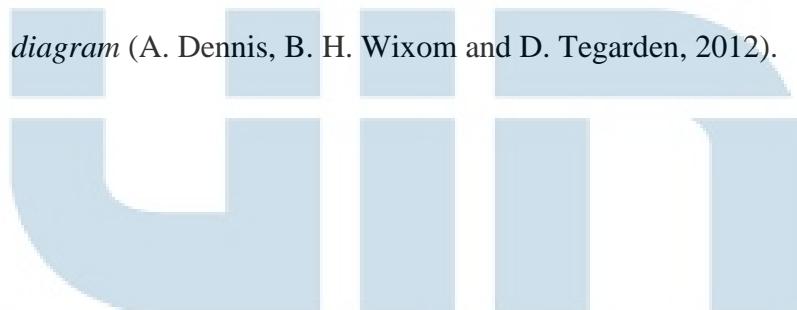
<i>Subject Boundary</i> 	Mewakili cakupan subjek, misalnya, sistem atau individu, proses bisnis. Kotak yang mendefinisikan ruang lingkup sistem dan secara jelas menggambarkan bagian diagram mana yang bersifat eksternal atau internal
--	---

#### 2.7.4.2. *Activity Diagram*

*Activity diagram* menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis (R. A. S. and M. Shalahuddin, 2012).

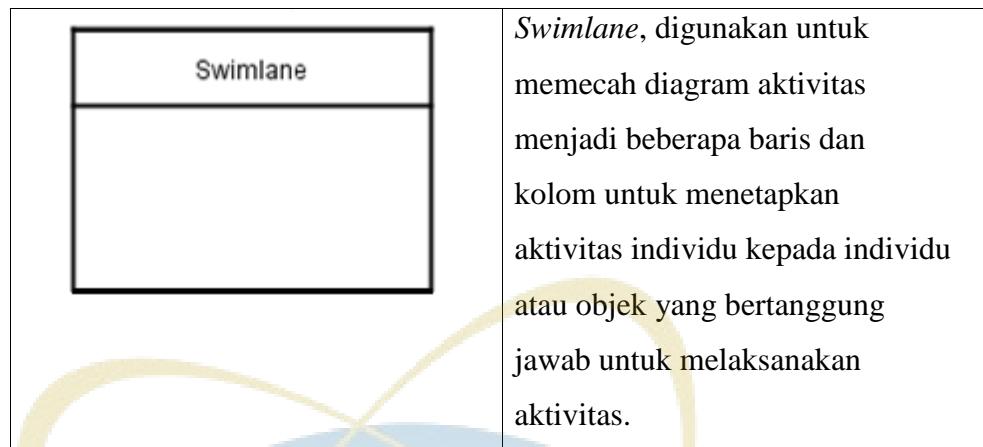
*Activity diagram* menunjukkan urutan kegiatan dalam proses, termasuk sekuensial dan paralel kegiatan, dan keputusan yang dibuat. Diagram aktivitas biasanya dibuat untuk satu *use case* dan dapat menunjukkan skenario yang mungkin berbeda (K. E. Kendall and J. E. Kendall, 2011).

Berikut adalah tabel penjelasan dari simbol-simbol yang ada pada *Activity diagram* (A. Dennis, B. H. Wixom and D. Tegarden, 2012).

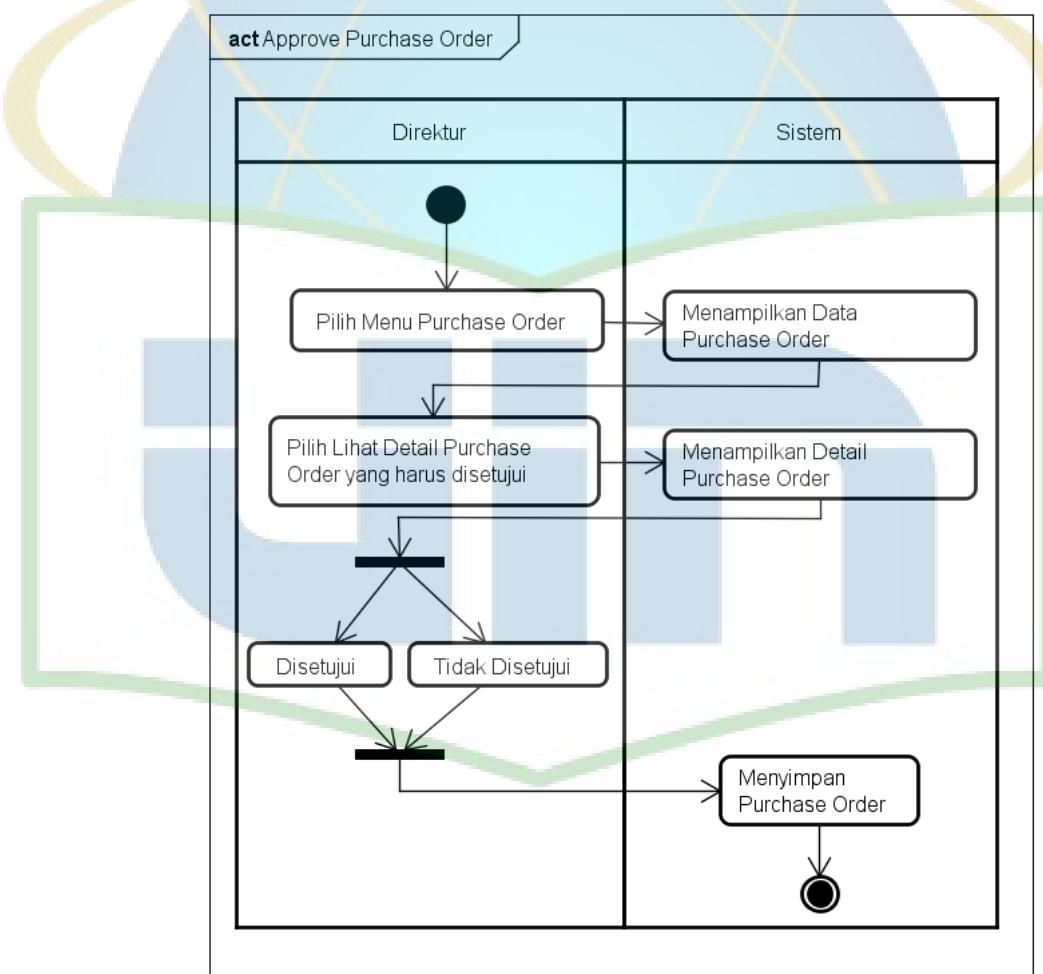


**Tabel 2. 3 Simbol-Simbol Activity Diagram**

Simbol	Keterangan
	<i>Initial node</i> menjelaskan awal serangkaian tindakan atau kegiatan pada <i>activity diagram</i>
	<i>Fork</i> , digunakan untuk membagi perilaku menjadi set arus aktivitas paralel atau bersamaan.
	<i>Join</i> digunakan untuk menyatukan kembali serangkaian aktivitas paralel atau bersamaan.
	<i>Activity</i> merepresentasikan aktivitas yang dilakukan pada sistem, aktivitas biasanya diawali dengan kata kerja
	<i>Decision</i> (keputusan) digunakan untuk mewakili kondisi pengujian untuk memastikan aliran kontrol atau arus objek hanya turun satu jalur.
	<i>Final-flow node</i> yaitu status akhir yang silakukan sistem, status akhir ketika mengakhiri diagram.
	<i>Control Flow</i> menunjukkan urutan eksekusi.



Berikut ini adalah contoh dari *activity diagram* :



Contoh gambar tersebut adalah *activity diagram approve purchase order* pada sistem informasi monitoring sirkulasi material proyek. Pada contoh tersebut dijelaskan kegiatan menyetujui atau tidak menyetujui *purchase order* yang ada di dalam sistem informasi monitoring sirkulasi material proyek. Setelah aktor selesai mengubah status *purchase order*, sistem akan menyimpan data *purchase order*.

#### 2.7.4.3. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antara kelas atau contoh objek dari waktu ke waktu. *Sequence diagram* sering digunakan untuk menggambarkan proses yang dijelaskan dalam skenario *use case*. Dalam prakteknya, diagram urutan yang berasal dari analisis *use case* dan digunakan dalam desain sistem untuk menurunkan interaksi, hubungan, dan metode dari objek dalam sistem. *Sequence diagram* digunakan untuk menunjukkan pola keseluruhan dari kegiatan atau interaksi dalam kasus penggunaan. Setiap skenario *use case* mungkin membuat satu *sequence diagram*, meskipun *sequence diagram* tidak selalu diciptakan untuk skenario kecil (K. E. Kendall and J. E. Kendall, 2011).

Berikut ini adalah tabel penjelasan dari simbol-simbol yang digunakan pada *sequence diagram* (T. Raharjo, 2014).

Tabel 2. 4 Simbol-Simbol Sequence Diagram

Simbol	Keterangan
 : Boundary	<i>Boundary</i> biasanya berupa tepi dari sistem, seperti <i>user interface</i> atau suatu alat yang berinteraksi dengan sistem lain
 : Control	<i>Control</i> merupakan elemen yang mengatur aliran dari informasi untuk sebuah <i>scenario</i> . Perilaku bisnis umumnya diatur oleh objek ini
 : Entity	<i>Entity</i> biasanya elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa model objek
 : Actor	Merepresentasikan entitas yang berada di luar sistem, mereka bisa berupa manusia, atau perangkat sistem lain.
 Message1	Relasi ini digunakan untuk memanggil operasi atau metode yang dimiliki oleh suatu objek. <i>Message</i> mengharuskan kita menyelesaikan proses baru kemudian memanggil proses berikutnya.
 Message2	Relasi ini menunjukkan bahwa suatu objek hendak memanggil dirinya sendiri.

#### 2.7.4.4. Object Identification

Untuk membuat diagram kelas, terlebih dahulu mengidentifikasi objek. *Object identification* merupakan salah satu kegiatan untuk menemukan objek-objek potensial yang digunakan untuk membuat *class diagram*. Pendekatan-pendekatan untuk mengidentifikasi objek adalah sebagai berikut (A. Dennis, B. H. Wixom and D. Tegarden, 2012).

##### 1. Analisis tekstual

Analisis tekstual adalah analisis teks dalam deskripsi *use case*. Analis memulai dengan meninjau deskripsi *use case* dan *use case diagram*. Teks dalam deskripsi diperiksa untuk mengidentifikasi objek, atribut, operasi, dan hubungan potensial. Kata benda dalam *use case* menyarankan kelas yang memungkinkan, dan kata kerja menyarankan kemungkinan operasi.

- Kata benda yang umum atau tidak benar menyiratkan kelas objek.
- Kata benda atau referensi langsung menyebutkan sebuah contoh dari sebuah kelas.
- Kata benda kolektif menyiratkan kelas objek yang terdiri dari kelompok contoh kelas lain.
- Kata sifat menyiratkan atribut objek.
- Kata kerja yang dilakukan menyiratkan sebuah operasi.
- Kata kerja perwujudan menyiratkan hubungan klasifikasi antara objek dan kelasnya.
- Kata kerja memiliki arti hubungan agregasi atau hubungan asosiasi.

- Kata kerja transitif menyiratkan sebuah operasi.
- Kata kerja intransitif menyiratkan pengecualian.
- Ungkapan-ungkapan predikat atau deskriptif menyiratkan sebuah operasi.
- Kata keterangan menyiratkan atribut hubungan atau operasi.

## 2. *Brainstorming*

*Brainstorming* adalah teknik penemuan yang telah berhasil digunakan dalam mengidentifikasi kelas kandidat. Intinya, dalam konteks ini, *brainstorming* adalah proses satu set individu yang mengatur dan menyarankan kelas potensial yang dapat berguna untuk masalah yang sedang dipertimbangkan. Biasanya, sesi *brainstorming* dimulai oleh seorang fasilitator yang meminta satu set individu untuk menjawab pertanyaan atau pernyataan tertentu yang membingkai sesi tersebut. Ini hanya meminta peserta untuk mengidentifikasi objek yang mereka hadapi. Misalnya, kumpulan objek potensial yang muncul dalam pikiran adalah dokter, perawat, resepsionis, pengangkatan, penyakit, perawatan, resep, kartu asuransi, dan catatan medis. Begitu jumlah objek kandidat telah diidentifikasi, peserta harus mendiskusikan dan memilih objek calon mana yang harus dipertimbangkan lebih lanjut. *Brainstorming* lebih lanjut dapat dilakukan untuk mengidentifikasi atribut, operasi, dan relasi potensial untuk masing-masing objek yang teridentifikasi.

### 3. Daftar Objek Umum

Sesuai namanya, daftar objek yang umum hanyalah daftar objek yang umum untuk domain bisnis sistem. Beberapa kategori objek telah ditemukan untuk membantu analis dalam membuat daftar, seperti hal-hal fisik atau nyata, insiden, peran, dan interaksi. Analis harus terlebih dahulu mencari hal-hal fisik atau halal dalam domain bisnis.

#### 2.7.4.5. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur aplikasi berorientasi objek dari sisi pendefinisian kelas-kelas yang akan dibuat untuk membangun aplikasi. Kelas memiliki apa yang disebut atribut dan metode atau operasi (R. A. S. and M. Shalahuddin, 2011).

*Class diagram* yaitu gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukkan kelas-kelas objek yang menyusun sebuah sistem dan juga hubungan antara kelas objek tersebut (J. L. Whitten and L. D. Bentley, 20017).

Berikut ini adalah tabel penjelasan dari simbol-simbol yang digunakan pada *class diagram* (T. Raharjo, 2014).

Tabel 2. 5 Simbol-Simbol Class Diagram

Simbol	Keterangan
	Class, merepresentasikan orang, tempat, atau hal tentang sistem yang dibutuhkan untuk menangkap dan menyimpan informasi.. Terdiri dari tiga elemen : <ul style="list-style-type: none"> <li>- <i>Class Name</i>, nama dari sebuah objek kelas</li> <li>- <i>Attribute</i> adalah Merupakan properti yang menggambarkan keadaan suatu objek..</li> <li>- <i>Operation</i>, aksi atau fungsi dilakukan oleh sebuah kelas</li> </ul>
	<i>Association</i> , relasi antar kelas atau dirinya sendiri, asosiasi juga biasanya juga disertai dengan dengan <i>multiplicity</i> yang mewakili berapa kali minimum atau maksimum <i>instance</i> kelas dapat dikaitkan dengan instance kelas terkait.
	<i>Generalization</i> , relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
	<i>Dependency</i> , relasi antar kelas dengan makna kebergantungan antar kelas
	<i>Aggregation</i> , relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

Hubungan antar objek dalam *class* terdapat multiplisitas, yang menunjukkan jumlah suatu objek yang bisa berhubungan dengan objek lain antara lain

- 1 (pasti 1),
- 0...1 (nol atau satu), 0...\* (nol atau lebih) atau 1...\* (satu atau lebih),
- Tidak ada batasan (bisa 0, 1, ..., n),
- Biasanya didefinisikan batas bawah dan atas, kecuali untuk yang pasti bernilai 1.

#### 2.7.5. *Rich Picture*

Menurut Valente dan Marchetti (N. Widyaningsih, 2014), *rich picture* merupakan pengetahuan tentang domain dan harus membimbing para pengembang sistem selama *fase* awal definisi dan konstruksi prototipe sistem.

Penciptaan *rich picture* adalah kegiatan awal desain, tahap dimana pengembang sistem harus membingkai masalah untuk menemukan solusi yang memadai, berfokus pada teknologi berorientasi objek.

Menurut Mathiassen (A. Khairunisa, 2013), *rich picture* adalah sistem atau situasi dengan menggunakan gambar-gambar. Gambaran keseluruhan dari orang, objek, proses, struktur dan masalah pada keseluruhan proses bisnis yang ada di perusahaan. *Rich picture* juga merupakan gambaran informal yang mempresentasikan pemahaman ilustrator terhadap situasi yang ada.

*Rich picture* memberikan deskripsi yang luas mengenai suatu situasi yang memungkinkan adanya interpretasi yang berbeda-beda. Tahap ini dilakukan untuk memperoleh pandangan menyeluruh terhadap situasi dan berbagai cara orang menginterpretasikannya.

Berikut ini adalah kegunaan *rich picture* (B. Fista, 2017)

- Alat yang ideal untuk berkomunikasi mengenai situasi yang rumit dan bermasalah
- Keterkaitan antar elemen serta hubungan yang terjalin, langsung maupun tidak lebih mudah untuk dilihat
- Memudahkan identifikasi pemilik masalah dan membantu dalam mengidentifikasi potensi masalah dan konflik
- Membantu dalam pembuatan batasan dan cakupan masalah

## 2.7.6. Konsep Basis Data

### 2.7.6.1. Pengertian Basis Data

Secara umum kita bisa mendefinisikan sistem basis data sebagai koleksi dari data-data yang terorganisasi sedemikian rupa sehingga data mudah disimpan dan dimanipulasi (diperbaharui, dicari, diolah dengan perhitungan-perhitungan tertentu, serta dihapus (A. Nugroho, 2011).

*Database* merupakan kumpulan *file* yang saling berhubungan. Tetapi, *database* tidak hanya kumpulan *file*. *Record* di dalam tiap *file* harus dapat dihubungkan dengan *record* di dalam *file* lain (J. L. Whitten and L. D. Bentley, 2007).

Dalam manajemen *database relational* terdapat komponen utama dalam konsep *database* (J. L. Whitten and L. D. Bentley, 2007), *Field* adalah unit terkecil data yang disimpan dalam *database*. Unit terkecil data yang disimpan dalam *database* :

- a. *Primary key*, yaitu *field* yang unik dan mengidentifikasi satu *record*.
- b. *Secondary key*, yaitu *field* yang mengidentifikasi sebuah *record* atau bagian dari beberapa *record* yang terkait.
- c. *Foreign key*, yaitu *field* yang menunjuk beberapa *record* pada *file* lain.
- d. *Descriptive field*, yaitu *non-key field*

1. *Record* adalah kumpulan *field* yang diatur dalam format yang *predetermined* (telah ditentukan).

a. *Fixed-length record structures*. Sebagian besar teknologi *database* memaksakan struktur *record fixed length*, dalam artian setiap *instance record* mempunyai *field* yang sama, jumlah *field* yang sama dan ukuran logika yang sama. Akan tetapi, beberapa sistem *database* akan mengompresi *field-field* dan nilai-nilai yang tidak digunakan untuk menghemat ruang penyimpanan *disk*.

b. *Variable-length record structures*. Memperbolehkan *record-record* pada *file* yang sama memiliki *length* yang berbeda.

## 2. *File* dan Tabel

*File* adalah kumpulan semua kejadian dari struktur *record* yang ditentukan. Tipe-tipe dari *file*, yaitu :

- a. *File* induk/master adalah *file* penting dalam sistem dan akan tetap ada selama siklus hidup sistem informasi berputar.
- b. *File* transaksi adalah *file* yang digunakan untuk merekam data dari suatu transaksi yang terjadi.

- c. *File* laporan adalah *file* yang berisi sistem informasi yang akan ditampilkan.
- d. *File* sejarah adalah *file* yang berisi data masa lalu yang sudah tidak aktif lagi.
- e. *File* pelindung adalah salinan dari *file-file* yang masih aktif di *database* pada saat tertentu yang digunakan bila *file database* rusak.
- f. *File* kerja adalah suatu proses program secara sementara karena memori komputer tidak mencukupi. Sedangkan tabel adalah suatu kesatuan unit dari *row / record* dengan atribut-atributnya (*column*).

#### 2.7.6.2. Bahasa Basis Data

Sistem basis data menyediakan bahasa untuk mendefinisikan basis data DDL (*Data Definition Language*) serta bahasa untuk memanipulasi basis data DML (*Data Manipulation Language*) untuk melakukan operasi-operasi tertentu pada basis data. Dalam praktiknya, kedua jenis Bahasa basis data itu tidak benar-benar dapat dipisahkan secara tegas. Saat ini keduanya merupakan bagian dari bahasa basis data tunggal yang disebut *Structured Query Language* (SQL) yang merupakan bahasa basis data standar untuk basis data bertipe relasional (A. Nugroho, 2011).

##### 1. *Data Definition Language* (DDL)

Skema basis data dengan sekumpulan definisi yang diekspresikan dengan bahasa khusus yang dinamakan *Data Definition Language*

(DDL). Kamus data (*data dictionary* atau *data directory*) adalah suatu himpunan dari metadata, suatu data yang menerangkan data yang lainnya. Skema tabel adalah suatu contoh dari metadata. Sistem basis data akan membaca kamus data sebelum membaca atau memodifikasi data yang sebenarnya. Data yang tersimpan pada basis data biasanya memiliki batasan-batasan (*constraint*) tertentu. Dalam hal ini DDL dapat digunakan untuk menentukan batasan-batasan basis data tersebut. Sistem basis data memeriksa batasan-batasan setiap kali basis data disisipkan dan diperbarui.

## 2. *Data Manipulation Language* (DML)

Secara umum, manipulasi data pada basis data meliputi hal-hal berikut:

- a. Pemanggilan informasi yang tersimpan di basis data (*query*)
- b. Penambahan informasi baru di basis data (*inserttion*)
- c. Penghapusan informasi yang tidak diperlukan lagi di basis data (*deletion*)
- d. Modifikasi informasi yang ada di basis data (*updation*)

*Data Manipulation Language* (DML) adalah bahasa yang memungkinkan pengguna untuk mengakses atau memanipulasi data dalam sistem basis data yang bertipe relational. Pada dasarnya, ada dua jenis DML, yaitu:

- a. DML Prosedural, yang menghendaki pengguna untuk menspesifikasikan data yang diperlukan dan cara untuk mendapatkan data itu. Hal ini dilakukan dengan bahasa-bahasa

pemograman yang mampu mengakses basis data (misalnya, Bahasa C/C++ atau Java)

- b. DML Deklaratif (DML Nonprosedural), yang menghendaki pengguna untuk menspesifikasikan data yang diperlukan tanpa harus menspesifikasikan cara untuk mendapatkannya. Contoh DML Nonprosedural adalah SQL.

*Query* adalah pernyataan yang meminta pemanggilan informasi tertentu dari basis data. Sebagian dari DML dinamakan bahas *Query (query language)*.

#### **2.7.6.3. Mapping Problem Domain Object to a RDBMS Format**

*Mapping Problem Domain Object to a RDBMS Format* bertujuan untuk mengoptimalkan *database* yang terbentuk sehingga tidak ada tabel yang mengalami redundansi pada saat pemanggilan data.

Cara memetakan *problem domain object* dalam *class diagram* ke format RDBMS ada 8 (delapan) tahapan, yaitu (A. Dennis, B. H. Wixom and D. Tegarden, 2012).

- a. Petakan semua *class konkret* dalam *problem domain* ke tabel RDBMS. Jika, *problem domain class abstrak* memiliki beberapa subclass langsung, petakan juga ke dalam tabel RDBMS.
- b. Petakan atribut bernilai tunggal ke dalam kolom tabel.
- c. Petakan metode untuk disimpan di dalam prosedur atau modul program.

- d. Petakan agregasi yang bernilai tunggal dan hubungan asosiasi dengan kolom yang dapat menyimpan *primary key* dari tabel terkait, yaitu, menambahkan *foreign key* ke dalam tabel. Lakukan hal tersebut untuk kedua sisi yang berhubungan.
- e. Petakan atribut *multi-valued* dan buat hubungan *one-to many* dari tabel yang asli ke tabel yang baru.
- f. Petakan *multi-valued* agregasi dan hubungan asosiasi ke tabel asosiatif baru yang menghubungkan dua tabel asli bersama-sama. Masukkan *primary key* dari kedua *table* tersebut ke dalam tabel asosiatif yang baru, yaitu menambahkan *foreign key* ke tabel.
- g. Untuk hubungan agregasi dan asosiasi dengan tipe campuran, *copy primary key* dari sisi *single-valued* (1..1 atau 0..1) ke kolom baru pada tabel yang memiliki hubungan *multi-valued* (1..\* atau 0..\*) yang dapat menyimpan *primary key* dari tabel terkait. Yaitu menambahkan *foreign key* ke tabel *multi-valued*.
- h. Untuk hubungan generalisasi, pastikan bahwa *primary key* dari *subclass* sama dengan *primary key* dari *superclass*. *Multiplicity* dari hubungan asosiasi yang baru dari *subclass* ke *superclass* harus 1..1. Jika *superclass* konkret, yaitu mereka dapat menginisiasi sendiri, maka *multiplicity* dari *superclass* ke *subclass* adalah 0..\*, jika tidak konkret hubungannya adalah 1..1. Selanjutnya, *Exclusive-or* (XOR) harus ditambahkan ke dalam asosiasi. Lakukan langkah ini untuk setiap *superclass*. Atau, sesuaikan *inheritance* dengan menyalin

atribut *superclass* ke semua *subclass* dan hapus *superclass* dari desain.

#### 2.7.6.4. Skema Database

Skema *database* yang dihasilkan merupakan RDBMS (*Relational Database Management System*) yang diperoleh dari proses *mapping class* yang telah optimal, disertai dengan pemetaan *primary key* yang menjadi *foreign key* pada tabel lain yang memiliki relasi.

RDBMS secara sederhana dapat diartikan sebagai suatu sistem dimana data dilihat oleh penggunanya hanya sebagai tabel dan operator yang digunakan oleh pengguna (Fathansyah, 2012).

#### 2.7.6.5. Spesifikasi Database

Spesifikasi basis data menggambarkan struktur data fisik pada suatu sistem atau aplikasi. Spesifikasi basis data menyajikan bagaimana penyimpanan data dilakukan di *software* basis data. Bentuk dari spesifikasi basis data sendiri secara umum berupa tabel yang menyajikan informasi *field* untuk seluruh tabel yang digunakan. Informasi *field* yang ditampilkan antara lain nama *field*, tipe *field*, panjang *field* dan *field* yang menjadi *field* kunci (*primary key*) (Fathansyah, 2012).

*Tabel 2. 6 Contoh Spesifikasi Database*

Nama Field	Tipe Data	Ukuran	Keterangan
user_id	<i>int</i>	4	<i>primary key user</i>
username	<i>varchar</i>	35	<i>username user</i>
password	<i>varchar</i>	35	<i>password user</i>
pegawai_id	<i>Int</i>	4	<i>foreign key</i>
level_id	<i>Int</i>	4	<i>foreign key</i>

Berdasarkan contoh tabel spesifikasi *database* di atas, dapat dijelaskan bahwa setiap variabel dari masing-masing *field* memiliki tipe data yang berbeda. Sebagai contoh, *field* dengan nama user\_id, memiliki tipe data *integer* dan panjang *field* = 4, artinya variabel tersebut hanya dapat menerima input bilangan bulat dan tidak bisa menerima *input* karakter selain angka dan memiliki panjang 4 *digit* yaitu dari angka 0 sampai dengan angka 9999. Sedangkan variabel yang mempunyai tipe data *varchar* dapat menerima *input* semua karakter.

## 2.6.7. Bahasa Pemrograman

### 2.6.7.1. PHP

PHP (akronim dari PHP: *Hypertext Preprocessor*) adalah bahasa pemrograman yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-parsing di dalam *web server* oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser* (B. Raharjo,2012)

#### 2.6.7.1.1. Codeigniter

*Codeigniter* adalah sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis *web* yang disusun dengan menggunakan bahasa PHP. *Codeigniter* bersifat *free opensource software* dimana setiap orang berhak menggunakannya tanpa harus dikenai biaya lisensi (Riyanto, 2011). *Codeigniter* merupakan framework yang menggunakan pola *model-view-controller (MVC)*. *MVC* adalah sebuah pola rancangan yang memungkinkan para pengembang memisahkan kode program mereka ke dalam tiga kelompok, yaitu *Model*, *View*, *Controller*. Berikut penjelasan mengenai *MVC*:

##### 1. *Model*

*Model* merupakan bagian yang didesain untuk bekerja dengan *database* yang biasanya berisi kumpulan *query* yang dialamatkan pada sebuah *database*. *Model* bisa juga diartikan sebagai bagian yang digunakan untuk membuat semacam *prototype* data.

##### 2. *View*

*View* adalahh bagian presentasi yang berfungsi sebagai antarmuka program aplikasi dengna pengguna.

### 3. Controller

*Controller* merupakan bagian yang digunakan untuk menempatkan fungsi-fungsi logika pengolah data yang telah diperoleh dari model atau sumber lainnya, kemudian ditampilkan melalui atau tanpa *view*.

#### 2.6.7.2. HTML

*HyperText Markup Language* (HTML) adalah sebuah bahasa yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah internet *browser*. HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*.

HTML5 adalah standar baru dari HTML. HTML5 didesain untuk memenuhi hampir semua kebutuhan *user* tanpa *plugin* tambahan. Kebutuhan-kebutuhan tersebut antara lain menampilkan animasi, menjalankan aplikasi, memutar music dan film. HTML5 juga *cross-platform*. Itu artinya dapat dijalankan di berbagai *platform* dan *device* seperti *smartphone*, *tablet*, *netbook*, *laptop* bahkan *smart TV* (P. Hidayatullah and K. J. Kawistara, 2013).

#### 2.6.7.3. MySQL

*MySQL* adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya. *MySQL* bersifat *open source* dan *free* (tidak perlu membayar untuk menggunakannya) pada berbagai platform, *MySQL* merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl) (M. R. Arief, 2011).

### 2.6.8. Testing

Tujuan pengujian adalah untuk menemukan kesalahan, dan pengujian yang baik adalah pengujian yang memiliki probabilitas tinggi untuk menemukan kesalahan. Pengujian itu sendiri memiliki serangkaian karakteristik yang mencapai tujuan untuk menemukan kesalahan paling banyak dengan sedikit usaha. Berikut ini adalah karakteristik dari pengujian (R. S. Pressman and B. R. Maxim, 2015).

- Pengujian yang bagus memiliki probabilitas tinggi untuk menemukan kesalahan. Untuk mencapai tujuan ini, penguji harus memahami perangkat lunak dan mencoba mengembangkan gambaran tentang bagaimana kemungkinan perangkat lunak tersebut gagal.
- Pengujian yang bagus tidak berlebihan. Waktu dan sumber daya yang terbatas. Tidak ada gunanya melakukan pengujian yang memiliki tujuan yang sama seperti pengujian lainnya. Setiap pengujian harus memiliki tujuan yang berbeda.
- Pengujian yang bagus seharusnya tidak terlalu sederhana atau terlalu rumit. Meskipun terkadang mungkin mengharuskan untuk menggabungkan serangkaian pengujian ke dalam satu kasus uji, kemungkinan efek samping yang terkait dengan pendekatan ini mungkin menutupi kesalahan. Secara umum, setiap pengujian harus dilakukan secara terpisah.

#### 2.6.8.1. *Black-Box Testing*

Adapun metode yang digunakan dalam testing ini yaitu dengan *black box testing*. Pengujian *black box* disebut juga pengujian *behavioral*, yang berfokus pada kebutuhan fungsional *software*. Artinya, teknik inti memungkinkan penguji memperoleh serangkaian kondisi masukan yang

akan sepenuhnya menjalankan semua persyaratan fungsional untuk sebuah program (R. S. Pressman and B. R. Maxim, 2015). *Black box* mencoba untuk menemukan kesalahan dalam kategori berikut (R. S. Pressman and B. R. Maxim, 2015).

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dan struktur data atau akses database eksternal
4. Kesalahan kinerja atau tingkah laku dan
5. Inisialisasi dan kesalahan terminasi.

*Black box testing* dengan sengaja mengabaikan struktur kontrol, perhatian difokuskan pada domain informasi. Pengujian ini dirancang untuk menjawab pertanyaan berikut (R. S. Pressman and B. R. Maxim, 2015).

- Bagaimana validitas fungsional diuji?
- Bagaimana perilaku dan kinerja sistem yang diuji?
- Masukan yang seperti apa yang akan membuat kasus uji yang bagus?
- Apakah sistem sangat sensitif terhadap nilai input tertentu?
- Bagaimana batas kelas data terisolasi?
- Berapa nilai data dan volume data yang dapat ditoleransi sistem?
- Apa pengaruh kombinasi data yang spesifik terhadap operasi system.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Metode Pengumpulan Data

Untuk melakukan analisis-analisis dalam pembuatan skripsi ini, penulis melakukan pengumpulan data terkait dengan rancang bangun sistem monitoring anggaran keuangan basis *Web*. Berikut adalah metodologi pengumpulan data yang penulis lakukan pada laporan ini.

##### 3.1.1. Observasi

Observasi (*observation*) adalah teknik atau pendekatan untuk yang bertujuan untuk mendapatkan data *primer* dengan cara mengamati langsung objek datanya. Pengamatan yang dilakukan penulis dalam skripsi ini adalah terhadap beberapa aspek yaitu navigasi umum pada Kementerian Agama Republik Indonesia, *detail end point/application user interface*, serta lingkungan (*environment*) lainnya.

Proses observasi ini dilakukan bersamaan dengan proses perancangan aplikasi. Kegiatan ini akan dilakukan ketika sebuah data yang diperlukan belum ada dan masih mungkin ditemukan atau diamati sendiri melalui observasi.

Dalam tahap observasi ini, dilakukan pengamatan secara langsung terhadap bagaimana para pegawai Kementerian Agama Republik Indonesia bekerja dan apa yang mereka kerjakan kesehariannya.

Pengamatan dilakukan mulai Juni 2017 sampai dengan selesai.

Pengamatan pada tanggal 10 Agustus – 25 Desember 2017 mulai jam 07.30 WIB – 16.00 WIB.

### 3.1.2. Studi Literatur

Studi literatur adalah studi yang dilakukan dengan menggunakan literatur sebagai objek kajiannya. Tujuan studi literatur adalah untuk mendapatkan peta tentang domain penelitian yang dilaksanakan.

Studi literatur yang penulis lakukan pada penelitian ini adalah dengan mengambil referensi dari buku-buku yang terkait dengan topik penelitian. Selain dari buku, penulis juga mengambil referensi dari jurnal ilmiah, skripsi, dan internet.

Dalam teknik ini penulis mengambil data-data dan informasi yang dijadikan landasan teori dan dasar dari pembuatan sistem dan pendokumentasian sistem berdasarkan buku-buku yang terdapat diperpustakaan yang berhubungan dengan perancangan sistem dan pembuatan sistem. Adapun buku-buku tersebut antara lain : Modul PHP dan MySQL, Rekayasa Sistem Berorientasi Objek, Sistem Informasi Konsep dan Aplikasi, Metode Desain dan Analisis Sistem, Analisis dan Desain Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis, Metodologi Penelitian Sistem Informasi dan lain sebagainya.

### 3.1.3. Wawancara

Wawancara (*interview*) adalah komunikasi dua arah untuk mengetahui proses bisnis yang ada. Wawancara ini dilakukan dengan melakukan diskusi langsung dengan Imam Syaukani selaku kepala bagian untuk mengetahui apa saja yang dibutuhkan dalam perancangan dan pembuatan sistem ini.

Berdasarkan wawancara yang penulis lakukan, didapatkan informasi antara lain:

1. Segala informasi tentang Kementerian Agama Republik Indonesia,
2. Proses yang ada pada pencarian data tentang anggaran keuangan yang ada di Kementerian Agama Republik Indonesia.
3. Proses cara perancangan aplikasi pada sistem yang berjalan.
4. Keluhan akan kelemahan-kelemahan terhadap sistem yang sedang berjalan saat ini.

### 3.2. Metode Pengembangan Sistem

Metodologi pengembangan sistem yang digunakan untuk mengembangkan sistem yang akan dirancang ini adalah metodologi berorientasi objek dengan pendekatan *Rapid Application Development* (RAD) dan *tools UML* untuk pemodelannya. RAD merupakan sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang ekstensif dalam konstruktif, cepat, berulang dan

bertambah serangkaian *prototype* bekerja sebuah sistem yang pada akhirnya berkembang kedalam sistem *final*.

Dalam pembuatan RAD terdapat beberapa tahap pengembangan yaitu :

### 1. Fase Perencanaan Kebutuhan (*Requirement Planning*)

Dalam fase ini penulis dengan Imam Syaukani yang menangani penelitian bertemu membicarakan tentang penelitian, setelah beberapa lama, penulis diajak untuk mengidentifikasi tujuan aplikasi dan sistem serta mengidentifikasi informasi kebutuhan yang ditambahkan dari tujuan tersebut. Dalam pertemuan tersebut diperoleh perencanaan yang akan dibuat dalam sistem yaitu:

- a. Tujuan Pengembangan Sistem : Tujuan pengembangan sistem merupakan inti masalah mengapa sistem monitoring anggaran keuangan berbasis *web* ini harus dirancang.
- b. Analisis Kebutuhan Sistem : Analisis kebutuhan sistem merupakan bagian dimana peneliti menjabarkan kebutuhan dari sistem ini.
- c. Definisi Masalah : Merupakan hasil akhir dan kesimpulan masalah yang ada dari hasil analisa sistem berjalan dalam pembuatan sistem ini.

### 2. Fase *Workshop Design*

Fase ini merupakan fase untuk merancang yang dapat digambarkan sebagai *workshop*. Penulis merancang system informasi ini dengan menggunakan *tools Unified Modelling Language* (UML). Dengan tahapan sebagai berikut:

a. Membuat *Use Case Diagram*

*Use case* merupakan *diagram* yang menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan interaksi dengan sistem itu. Dalam pembuatan sistem ini terdapat 2 (dua) aktor yang merupakan pengguna dari sistem informasi denah ruangan untuk kios informasi yaitu admin sebagai yang mengelola sistem dan pegawai yang menggunakan sistem untuk proses pencarian letak ruangan.

b. Membuat Narasi *Use Case Diagram*

Narasi *use case* merupakan sebuah cara untuk menjelaskan rincian yang ada pada *use case diagram*. Dengan membuat narasi *use case* penulis berharap bisa menuangkan dan menjelaskan *use case diagram* dengan jelas. Adapun skenario *use case* ini terdiri dari tabel-tabel yang menjelaskan nama *use case*, siapa actor yang menggunakan, deskripsi dari *use case*, bagaimana prekondisinya, bagaimana kegiatan para pelaku dan sistem yang meresponnya, serta alternatif kondisi dan kesimpulan dari suatu *use case* tersebut.

c. Membuat *Activity Diagram*

*Activity diagram* merupakan sebuah *diagram* yang menggambarkan rangkaian aliran aktivitas baik proses bisnis atau *use case*. Penulis disini menuangkan *activity diagram* untuk menggambarkan rangkaian aliran aktivitas dari *use case diagram* yang mana dalam

*activity* ini juga menjadi patokan bagi penulis untuk memodelkan *action* yang dilakukan saat sebuah operasi dieksekusi.

d. Membuat *Sequence Diagram*

*Sequence diagram* disini merupakan cara penulis untuk menggambarkan bagaimana objek berinteraksi satu sama lainnya melalui pesan pada saat sebuah *use case* dieksekusi. Dengan membuat *sequence diagram* ini penulis bisa melihat tahap demi tahap apa yang harus terjadi untuk menghasilkan sesuatu di dalam *use case*.

e. Membuat *Class Diagram*

Dalam merancang *class diagram* ini penulis melihat hubungan yang terjadi antar *class object* dan menuangkannya dalam *class diagram*.

Dalam *class diagram* ini diharapkan dapat membantu dalam memvisualisasikan struktur kelas–kelas dalam model desain (*logical view*) dari suatu sistem. Dengan *class diagram* ini terlihat pula struktur dari suatu kelas yang membentuk arsitektur sistem ini.

f. Membuat Perancangan *Database*

Perancangan *database* merupakan langkah penulis untuk menuangkan dan menggambarkan tabel–tabel data yang diperlukan dalam pembuatan arsitektur sistem informasi ini. Dalam perancangan database ini, dapat diketahui *field* ataupun isi dari database beserta jenis data yang dipakai dalam sebuah database dan keterangannya.

#### g. Membuat Desain *User Interface*

Desain *user interface* merupakan langkah terakhir penulis menuangkan perencanaan terhadap arsitektur sistem informasi ini.

Dalam membuat desain *user interface*, penulis merancang tampilan *interface* yang akan digunakan oleh *user* dan membuat tampilan dari arsitektur sistem ini sebaik mungkin untuk memudahkan *user* dalam menggunakannya nanti. Dalam membuat desain *user interface*, penulis juga berkonsultasi dengan para pengguna sistem tentang bagaimana tampilan yang mereka inginkan dari arsitektur sistem ini.

### 3. Fase *Implementation*

Pada tahapan ini merupakan tahap pengujian atau *testing* terhadap aplikasi dan sistem serta melakukan sosialisasi terhadap aplikasi dan sistem yang telah dibuat kepada pihak terkait. Namun karena penilitian ini terbatas sampai menganalisa dan merancang suatu sistem baru, sistem ini belum sampai ke tahap implementasi yang sempurna, maka penulis hanya melampirkan kodingan, hasil *interface* akhir, spesifikasi *hardware* dan *software* yang dibutuhkan untuk penerapan arsitektur sistem ini nantinya.

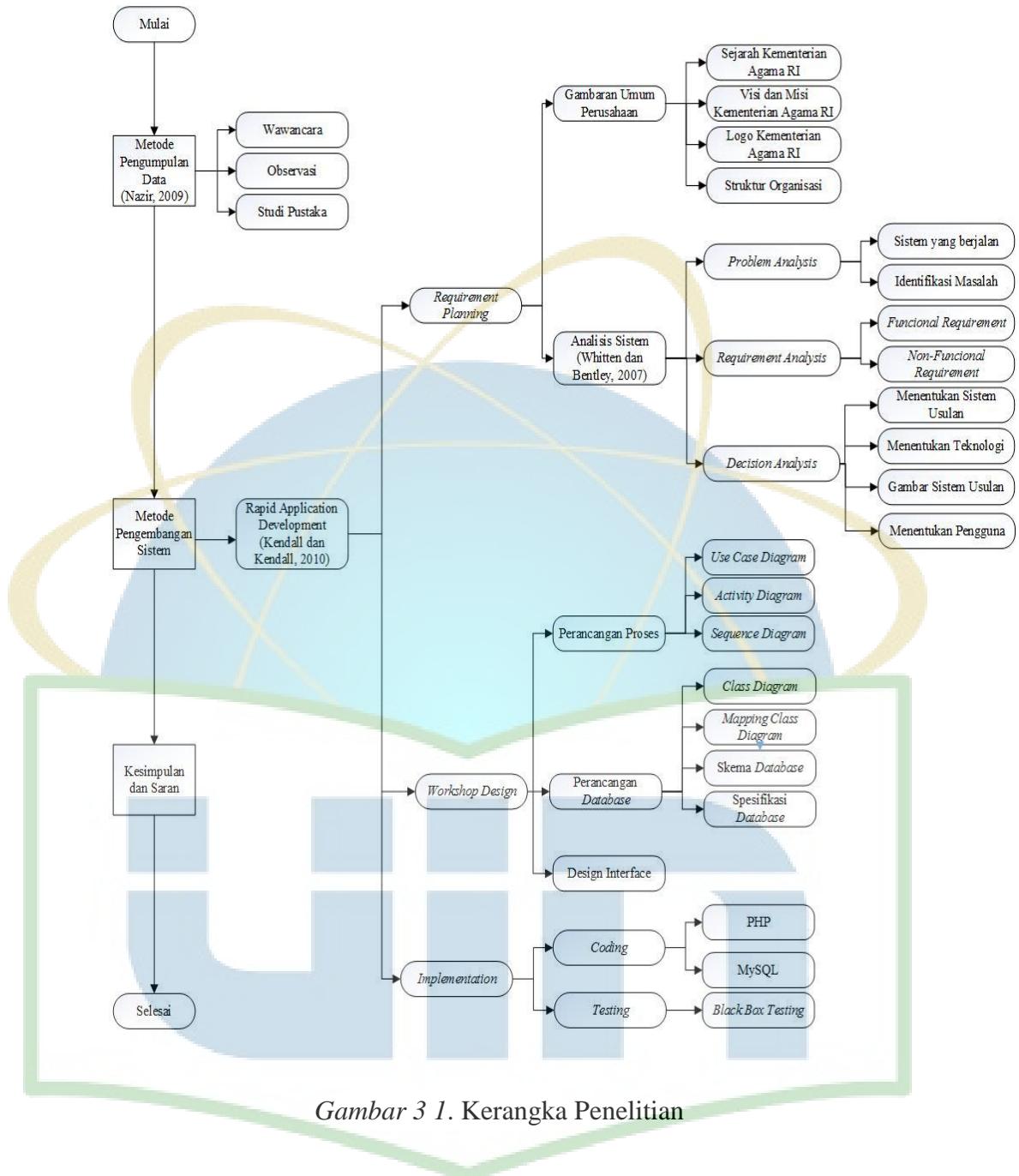
#### 3.3. Alasan Menggunakan RAD

Berikut akan dijelaskan beberapa alasan penulis menggunakan metodologi pengembangan sistem RAD dalam merancang sistem ini :

1. Karena melibatkan langsung para pengguna sistem dalam aktivitas analisis, design, dan kontruksi dapat berperan memberikan masukan dan arahan dalam pembuatan sistem.
2. Mengakselerasi fase-fase analisis dan desain persyaratan melalui pendekatan konstruksi berulang. Karena RAD dapat digunakan dalam pengembangan berkelanjutan.
3. Proses RAD memungkinkan mempersingkat waktu yang sesuai dengan penelitian ini karena kebutuhan yang cepat dipahami.
4. RAD memungkinkan pendokumentasian untuk keperluan penulisan hasil penelitian.

#### **3.4. Kerangka Penelitian**

Dari hasil penjabaran metodologi pengumpulan data dan metodologi pengembangan sistem di atas, maka kerangka penelitian yang dilakukan akan terlihat pada gambar 3.1. di bawah ini.



## BAB IV

### PEMBAHASAN

#### 4.1. *Requirement Planning*

Sebagaimana telah diuraikan di bab sebelumnya bahwa fase ini adalah fase dimana peneliti bertemu dengan *end-user* untuk sama-sama mengidentifikasi tujuan, syarat-syarat dari kebutuhan sistem yang ditimbulkan atas tujuan sistem yang dirumuskan, serta mengidentifikasi masalah yang menjadi latar belakang dalam perancangan sistem.

##### 4.1.1. Gambaran Umum Kementerian Agama Republik Indonesia

###### 4.1.1.1. Sejarah Kementerian Agama Republik Indonesia

Semangat keagamaan di Negara Indoneisa, tercermin dalam kehidupan bernegara yang dapat dijumpai dalam dokumen-dokumen kenegaraan tentang falsafah negara Pancasila, UUD 1945, GBHN, dan buku Repelita serta memberi jiwa dan warna pada pidato-pidato kenegaraan. Dalam pelaksanaan pembangunan nasional semangat keagamaan tersebut menjadi lebih kuat dengan ditetapkannya asas keimanan dan ketaqwaan terhadap Tuhan yang Maha Esa sebagai salah satu asas pembangunan.

Secara historis benang merah nafas keagamaan tersebut dapat ditelusuri sejak abad V Masehi, dengan berdirinya kerajaan Kutai yang bercorak Hindu di Kalimantan melekat pada kerajaan-kerajaan di pulau Jawa, antara lain kerajaan Tarumanegara di Jawa Barat, dan

kerajaan Purnawarman di Jawa Tengah. Pada abad VIII corak agama Budha menjadi salah satu ciri kerajaan Sriwijaya yang pengaruhnya cukup luas sampai ke Sri Lanka, Thailand dan India. Menurut salah satu sumber, Islam mulai memasuki Indonesia sejak abad VII melalui para pedagang Arab yang telah lama berhubungan dagang dengan kepulauan Indonesia tidak lama setelah Islam berkembang di jazirah Arab. Agama Islam tersiar secara hampir merata di seluruh kepulauan nusantara seiring dengan berdirinya kerajaan-kerajaan Islam seperti Perlak dan Samudera Pasai di Aceh, kerajaan Demak, Pajang dan Mataram di Jawa Tengah, kerajaan Cirebon dan Banten di Jawa Barat dan lain-lain. Pola pemerintahan kerajaan-kerajaan tersebut diatas pada umumnya selalu memiliki dan melaksanakan fungsi sebagai berikut:

1. Fungsi pemerintahan umum, hal ini tercermin pada gelar “Sampean Dalem Hingkang Sinuhun” sebagai pelaksana fungsi pemerintahan umum.
2. Fungsi pemimpin keagamaan tercermin pada gelar “Sayidin Panatagama Kalifatulah”.
3. Fungsi keamanan dan pertahanan, tercermin dalam gelar raja “Senopati Hing Ngalogo”.

Pokok-pokok kebijaksanaan pemerintah Hindia Belanda di bidang agama adalah sebagai berikut:

1. Bagi golongan Nasrani dijamin hak hidup dan kedaulatan organisasi agama dan gereja, tetapi harus ada izin bagi guru agama, pendeta dan petugas misi/zending dalam melakukan pekerjaan di suatu daerah tertentu.
2. Bagi penduduk pribumi yang tidak memeluk agama Nasrani, semua urusan agama diserahkan pelaksanaan dan perigawasannya kepada para raja, bupati dan kepala bumiputera lainnya.

Berdasarkan kebijaksanaan tersebut, pelaksanaannya secara teknis dikoordinasikan oleh beberapa instansi di pusat yaitu:

1. Soal peribadatan umum, terutama bagi golongan Nasrani menjadi wewenang *Departement van Onderwijs en Eeredienst* (Departemen Pengajaran dan Ibadah)
2. Soal pengangkatan pejabat agama penduduk pribumi, soal perkawinan, kemasjidan, haji, dan lainlain, menjadi urusan *Departement van Binnenlandsch Bestuur* (Departemen Dalam Negeri).
3. Soal Mahkamah Islam Tinggi atau *Hofd voor Islamietische Zaken* menjadi wewenang *Departement van Justitie* (Departemen Kehakiman). Pada masa penjajahan Jepang kondisi tersebut pada dasarnya tidak berubah. Pemerintah Jepang membentuk *Shumubu*, yaitu kantor agama pusat yang berfungsi sama dengan *Kantoor voor Islamietische Zaken* dan

mendirikan *Shumuka*, kantor agama karesidenan, dengan menempatkan tokoh pergerakan Islam sebagai pemimpin kantor.

Secara filosofis, sosio politis dan historis agama bagi bangsa Indonesia sudah berurat dan berakar dalam kehidupan bangsa. Itulah sebabnya para tokoh dan pemuka agama selalu tampil sebagai pelopor pergerakan dan perjuangan kemerdekaan baik melalui partai politik maupun sarana lainnya. Pada masa kemerdekaan kedudukan agama menjadi lebih kokoh dengan ditetapkannya Pancasila sebagai ideologi dan falsafah negara dan UUD 1945. Sila Ketuhanan Yang Maha Esa yang diakui sebagai sumber dari sila-sila lainnya mencerminkan karakter bangsa Indonesia yang sangat religius. Berdirinya Departemen Agama pada 3 Januari 1946, sebagai realisasi dan penjabaran ideologi Pancasila dan UUD 1945. Ketentuan yuridis tentang agama tertuang dalam UUD 1945 BAB E pasal 29 tentang Agama ayat 1, dan 2:

1. Negara berdasarkan atas Ketuhanan Yang Maha Esa;
2. Negara menjamin kemerdekaan tiap-tiap penduduk untuk memeluk agamanya masing-masing dan beribadah menurut agamanya dan kepercayaannya itu. Dengan demikian agama telah menjadi bagian dari sistem kenegaraan sebagai hasil konsensus nasional dan konvensi dalam\_praktek kenegaraan Republik Indonesia yang berdasarkan Pancasila dan UUD 1945.

#### 4.1.1.2. Visi Dan Misi Kementerian Agama Republik Indonesia

##### a. Visi

“Terwujudnya masyarakat Indonesia yang taat beragama, rukun, cerdas, mandiri dan sejahtera lahir batin.” (Keputusan Menteri Agama Nomor 2 Tahun 2010).

##### b. Misi

1. Meningkatkan kualitas kehidupan beragama.
2. Meningkatkan kualitas kerukunan umat beragama.
3. Meningkatkan kualitas raudhatul athfal, madrasah, perguruan tinggi agama, pendidikan agama, dan pendidikan keagamaan.
4. Meningkatkan kualitas penyelenggaraan ibadah haji.
5. Mewujudkan tata kelola kepemerintahan yang bersih dan berwibawa.

#### 4.1.1.3. Logo Kementerian Agama Republik Indonesia



Gambar 4. 1 Logo Kementerian Agama Republik Indonesia

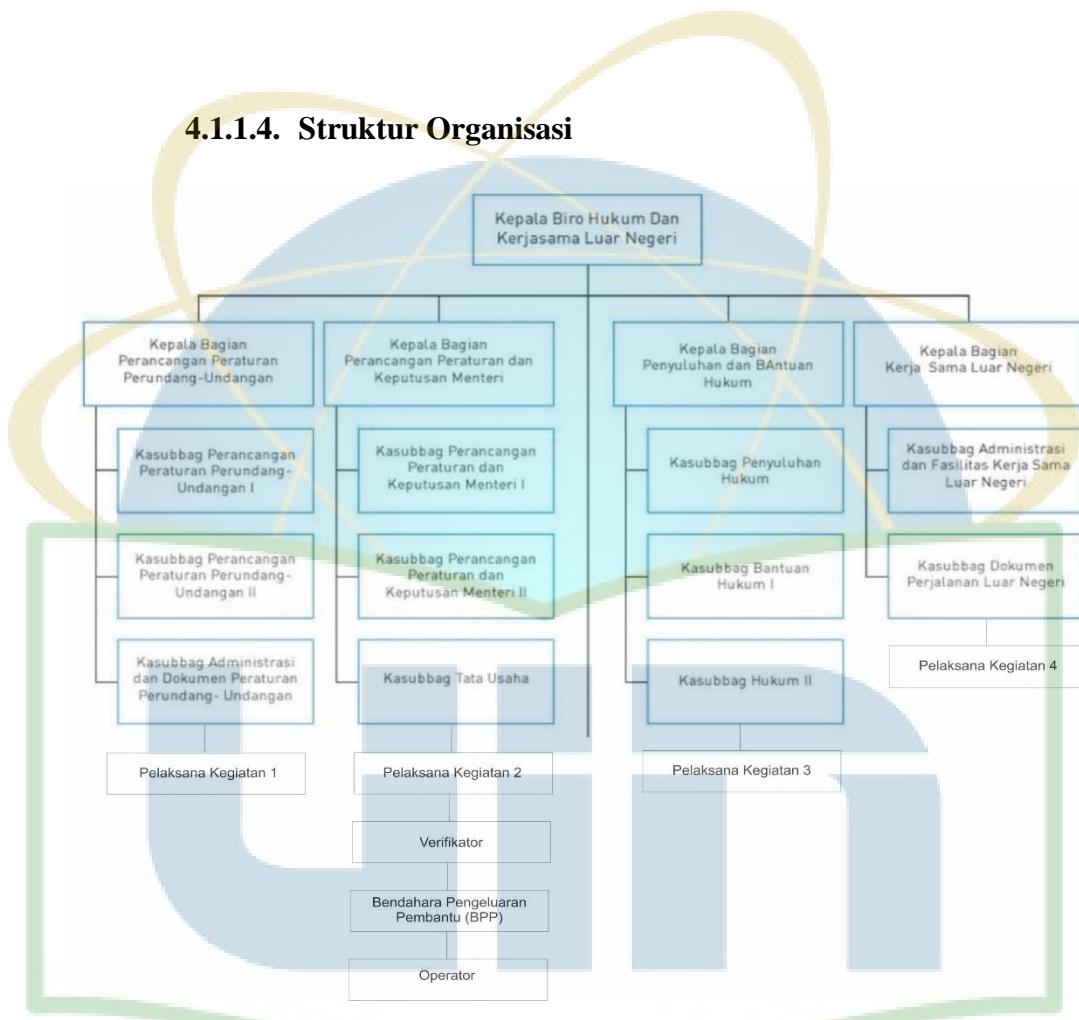
Berikut penjelasan dari makna yang terkandung dalam logo Kementerian Agama Republik Indonesia, antara lain:

1. Bintang bersudut lima yang melambangkan sila Ketuhanan Yang Maha Esa dalam Pancasila, bermakna bahwa karyawan Departemen Agama selalu menaati dan menjunjung tinggi norma-norma agama dalam melaksanakan tugas Pemerintahan dalam Negara Republik Indonesia yang berdasarkan Pancasila.
2. 17 kuntum bunga kapas, 8 baris tulisan dalam Kitab Suci dan 45 butir padi bermakna Proklamasi Kemerdekaan republik Indonesia pada tanggal 17 Agustus 1945, menunjukkan kebulatan tekad para Karyawan Departemen Agama untuk

membela Kemerdekaan Negara Kesatuan republik Indonesia yang diproklamirkan pada tanggal 17 Agustus 1945.

3. Butiran Padi dan Kapas yang melingkar berbentuk bulatan bermakna bahwa Karyawan Departemen mengemban tugas untuk mewujudkan masyarakat yang sejahtera, adil, makmur dan merata.
4. Kitab Suci bermakna sebagai pedoman hidup dan kehidupan yang serasi antara kebahagiaan duniawi dan ukhrawi, materil dan spirituial dengan ridha Allah SWT Tuhan Yang Maha Esa.
5. Alas Kitab Suci bermakna bahwa pedoman hidup dan kehidupan harus ditempatkan pada proporsi yang sebenarnya sesuai dengan potensi dinamis dari Kitab Suci.
6. Kalimat Ikhlas Beramal bermakna bahwa Karyawan Departemen Agama dalam mengabdi kepada masyarakat dan Negara berlandaskan niat beribadah dengan tulus dan ikhlas.
7. Perisai yang berbentuk segi lima sama sisi dimaksudkan bahwa kerukunan hidup antar umat beragama RI yang berdasarkan Pancasila dilindungi sepenuhnya sesuai dengan Undang-Undang Dasar 1945.
8. Kelengkapan makna lambang Departemen Agama melukiskan motto : Dengan Iman yang teguh dan hati yang

suci serta menghayati dan mengamalkan Pancasila yang merupakan tuntutan dan pegangan hidup dalam kehidupan bermasyarakat dan bernegara, karyawan Departemen Agama bertekad bahwa mengabdi kepada Negara adalah ibadah.



*Gambar 4. 2 Struktur Organisasi Biro Hukum dan Kerja sama Luar Negeri Kementerian Agama RI*

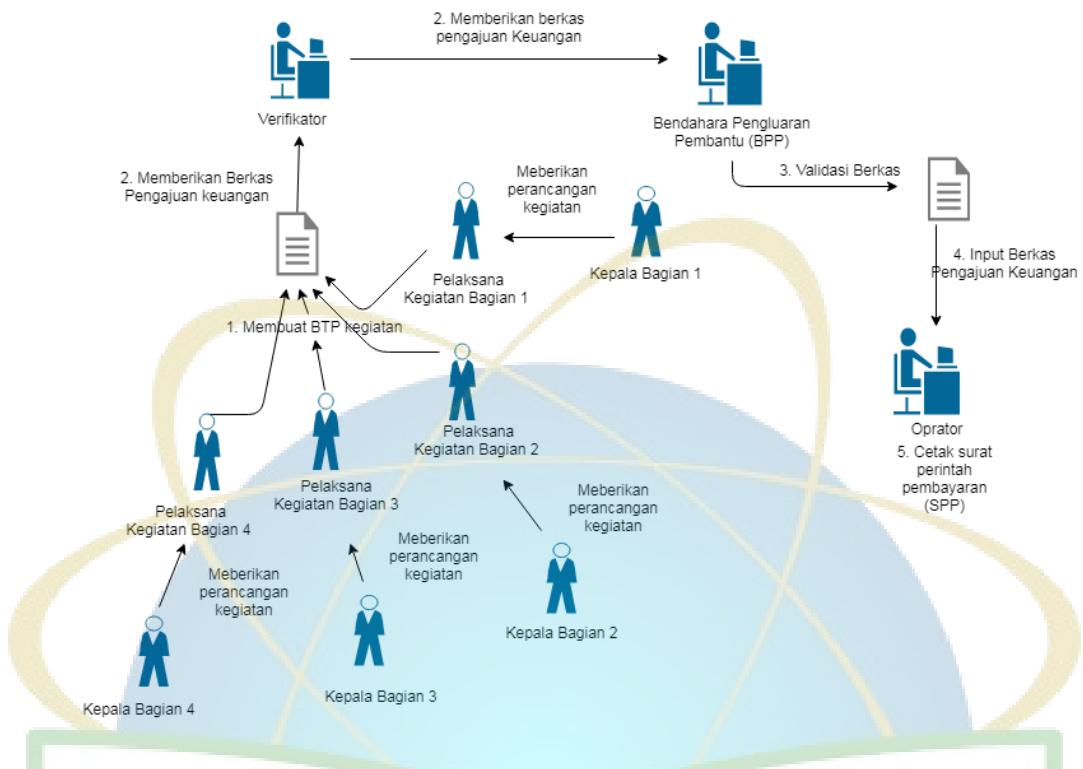
#### 4.1.2. Analisis Sistem

##### 4.1.2.1. Analisis Sistem Berjalan

Berdasarkan hasil dari pengumpulan data dengan observasi dan wawancara yang dilakukan penulis kepada *user* yang terlibat maka di jelaskan sistem yang sedang berjalan. Sistem yang berjalan di Biro Hukum dan KLN saat ini ialah dengan menggunakan exel/form serah terima berkas yang bertujuan untuk dalam rangka pengawasan berkas. Yang memudahkan Pelaksana Kegiatan melaporkan letak posisi dan pengoreksian berkas kepada Kepala Bagian masing-masing.

Masing-masing Kepala Bagian memberikan perintah *Rancangan Kegiatan* kepada pelaksana-pelaksanannya dan masing-masing untuk pembuatan BTP (Berkas Tagihan Pencairan) dari rancangan kegiatan-kegiatan yang akan dilaksanakan. Lalu diberikan kepada verifikator untuk dikoreksi kelengkapan berkas dan kebenaran akun PAGU yang digunakan. Kemudian diberikan ke BPP (Bendahara Pengluaran Pembantu) untuk di validasi berkas tersebut dan di berikannya kepada operator untuk penginputan SPP (Surat Perintah Pembayaran).

*Gamabar 4 3 Rich Picture Sistem Berjalan*



#### **4.1.2.1.1. Identifikasi Masalah**

Di dalam mengidentifikasi masalah, peneliti menganalisisi sisitem yang berjalan di dalam perusahaan. System yang di gunakan untuk melakukan dalam memonitoring keuangan yang pada dasarnya untuk memenuhi prosedur kebutuhan. Namun system tersebut masih terdapat kekurangan, diantaranya sebagai berikut:

1. Sering terjadinya berkas pengajuan anggaran yang dilakukan secara manual menimbulkan sikap saling tuduh yang dikarenakan berkas itu hilang, tidak tertib antrian.

2. Tidak adanya wadah untuk menyimpan soft copy berkas pertanggung jawaban atau SPJ yang telah rampung.
3. Tidak tersedianya akses bagi setiap pengelola keuangan untuk memonitoring berkas yang diajukan.

#### **3.1.1.1. *Requirement Analysis***

Tahapan ini mendefinisikan dan menganalisis masalah-masalah persyaratan system yang mendukung dalam memonitoring keuangan. Tahapan ini yang bertujuan untuk menentukan apa yang bisa dilakukan dalam memonitoring keuangan untuk mengelola keuangan menjadi efektif dan efisien.

##### ➤ *Fungsional Requirement*

System yang dikembangkan harus mempunyai *fungsional requirement* sebagai berikut:

1. System yang dapat mengelola, menyimpan, dan mengintegrasikan data dalam proses manage data keuangan.
2. Terdapat fungsi tambah, edit, danhapus dalam pengolahan data-data di dalam sistem.

➤ *Nonfungsional Requirement*

Nonfungsional requirement dari system informasi memonitoring keungan yang di kembangkan sebagai berikut:

1. System yang dapat mengefisienkan waktu
2. Tampilan system yang bersifat *user friendly* agar dapat dimudahkan dalam penggunaannya.
3. System dapat menyediakan data informasi yang actual dan akurat

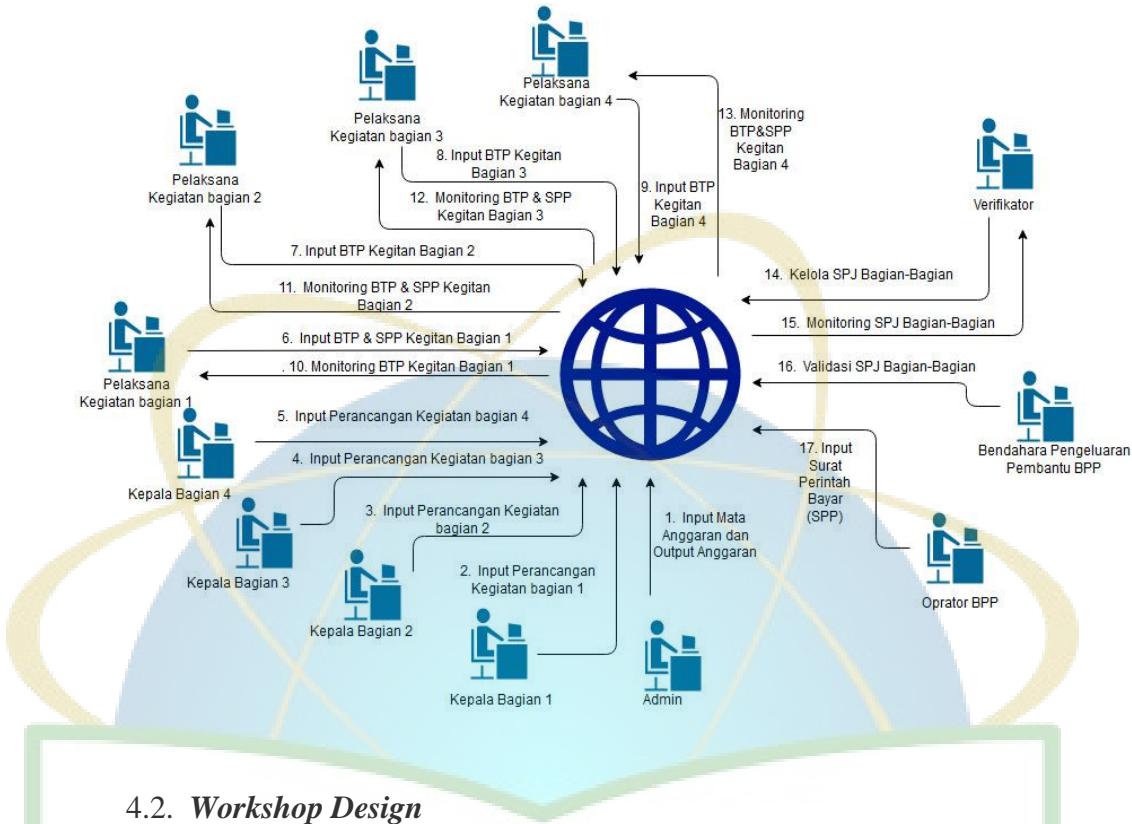
#### **4.1.2.2. *Decision Analysis (Analisis Kebutuhan)***

Dari tahapan analisis sebelumnya telah diketahui permasalahan dari system yang sedang berjalan, persyaratan dan kebutuhan yang diinginkan, maka tahapan selanjutnya adalah analisis kebutuhan yaitu menetukan komponen-komponen dari system usulan yang akan di rancang, di bangun dan di implementasikan. System usulan ini di harapkan mampu memenuhi kebutuhan dalam proses memonitoring keuangan dan memberi solusi keuangan dalam system yang berjalan.

#### 4.1.2.2. Menentukan Sistem Usulan

Berdasarkan persyaratan yang telah didapat sebelumnya, maka dapat disimpulkan bahwa monitoring anggaran Biro Hukum dan KLN dimulai dari aktor admin. Admin disini mempunyai tugas untuk menginput rincian anggaran kegiatan beserta dasar anggaran PAGU. Yang dimaksud rincian kegiatan ini ialah, data jenis kegiatan, jenis belanja, dan detail jenis belanja. Kemudian kepala bagian masing-masing menginput data rancangan kegiatan. Pelaksana kegiatan menginput Berkas Tagihan Pencairan (BTP) dan semua berkas yang dia ajukan dapat di monitoring oleh pelaksana. lalu akan di verifikasi oleh Verifikator dari segi kelengkapan berkas dan penggunaan PAGU output apakah sudah sesuai bila tidak sesuai akan di UP kembali disistem dengan status perbaikan berkas. Bila sudah sesuai lalu BPP memvalidasi data tersebut. Dan oprator membuat Surat Perintah Pembayaran (SPP) dari data yang telah di validasi oleh BPP.

**Gamabar 4. 4 Rich Picture Sistem Usulan**



## 4.2. Workshop Design

### 4.2.1. Perancangan Desain Sistem

#### 4.2.1.1. Perancangan Use Case Diagram

*Use case diagram* di bawah ini yaitu menggambarkan mengenai interaksi antara sistem, eksternal sistem dan *user*. Adapun langkah-langkah dalam membuat *use case diagram* yaitu sebagai berikut:

Perancangan *Use case* di bawah ini yaitu menggambarkan mengenai interaksi antara sistem, eksternal sistem dan *user*. Adapun langkah-langkah dalam membuat perancangan *Use case* yaitu sebagai berikut:

a. Identifikasi *actor*.

b. Identifikasi *use case*.

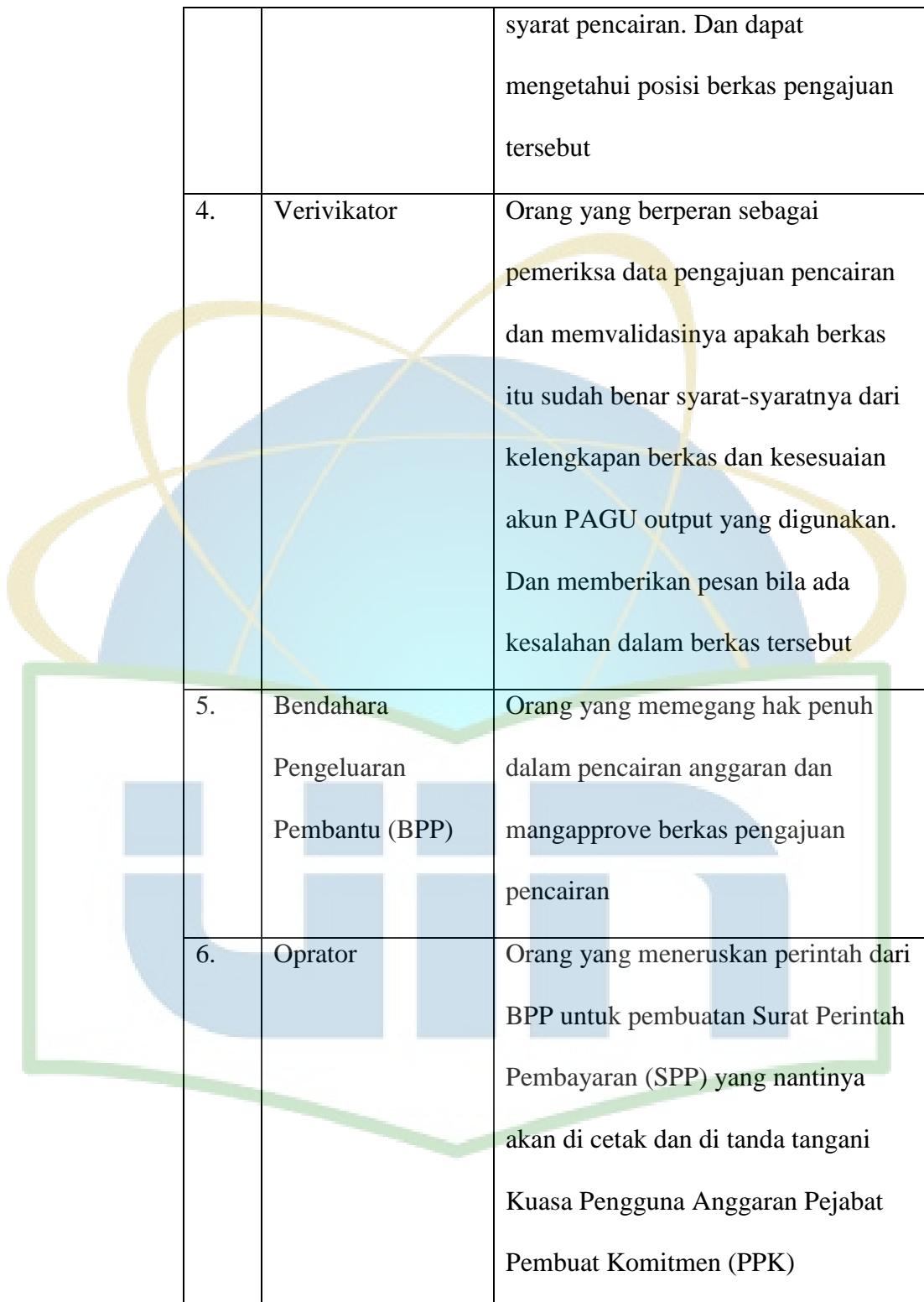
c. *Use case diagram*.

d. Narasi *use case*.

#### a) Identifikasi Actor

Tabel 4. 1 Identifikasi Actor

No.	Actor	Description
1.	Admin	mempunyai hak akses untuk mengendalikan sistem ini. Hak akses yang dimiliki oleh admin adalah me-manage data kegiatan, jenis kegiatan, detail jenis kegiatan, jenis belanja serta me-manage kegiatan unit.
2.	Kepala Bagian	Pegawai yang menjabat sebagai perancangan kegiatan dan pemegang hak kuasa anggaran bagian
3.	Pelaksana kegiatan	Orang yang meneruskan perintas kepala bagian dalam menginput berkas pengajuan bertakas tagihan pencairan dan melengkapi sayarat-

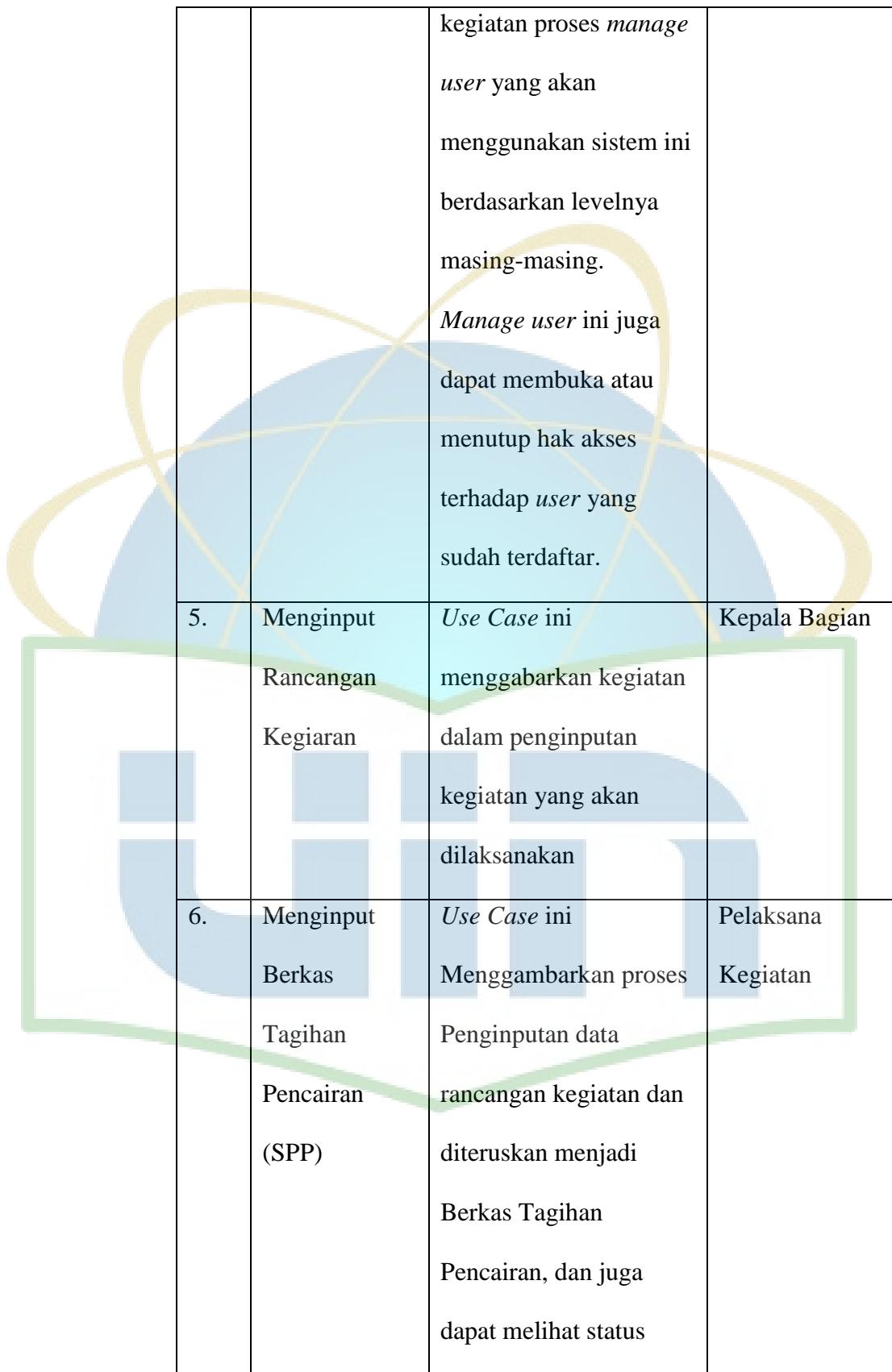


		syarat pencairan. Dan dapat mengetahui posisi berkas pengajuan tersebut
4.	Verifikator	Orang yang berperan sebagai pemeriksa data pengajuan pencairan dan memvalidasinya apakah berkas itu sudah benar syarat-syaratnya dari kelengkapan berkas dan kesesuaian akun PAGU output yang digunakan. Dan memberikan pesan bila ada kesalahan dalam berkas tersebut
5.	Bendahara Pengeluaran Pembantu (BPP)	Orang yang memegang hak penuh dalam pencairan anggaran dan mangapprove berkas pengajuan pencairan
6.	Oprator	Orang yang meneruskan perintah dari BPP untuk pembuatan Surat Perintah Pembayaran (SPP) yang nantinya akan di cetak dan di tanda tangani Kuasa Pengguna Anggaran Pejabat Pembuat Komitmen (PPK)

### b) Identifikasi *Use Case*

Tabel 0.2 Identifikasi Actor

No.	<i>Usecase name</i>	<i>Description</i>	Actor
1.	<i>Login</i>	<i>Use case yang menggambarkan proses masuk ke dalam aplikasi Username dan Password.</i>	Admin, kepala bagian, Pelaksana Bagian, Verifikator, BPP, Oprator
2.	<i>Logout</i>	<i>Use case ini menggambarkan kegiatan keluar dari sistem.</i>	Pelaksana Bagian, Bendahara Biro, Verivikator, Oprator
3.	<i>Ganti password</i>	<i>Use case ini menggambarkan kegiatan ubah password</i>	Admin
4.	<i>Manage User</i>	<i>Use case ini menggambarkan</i>	Admin

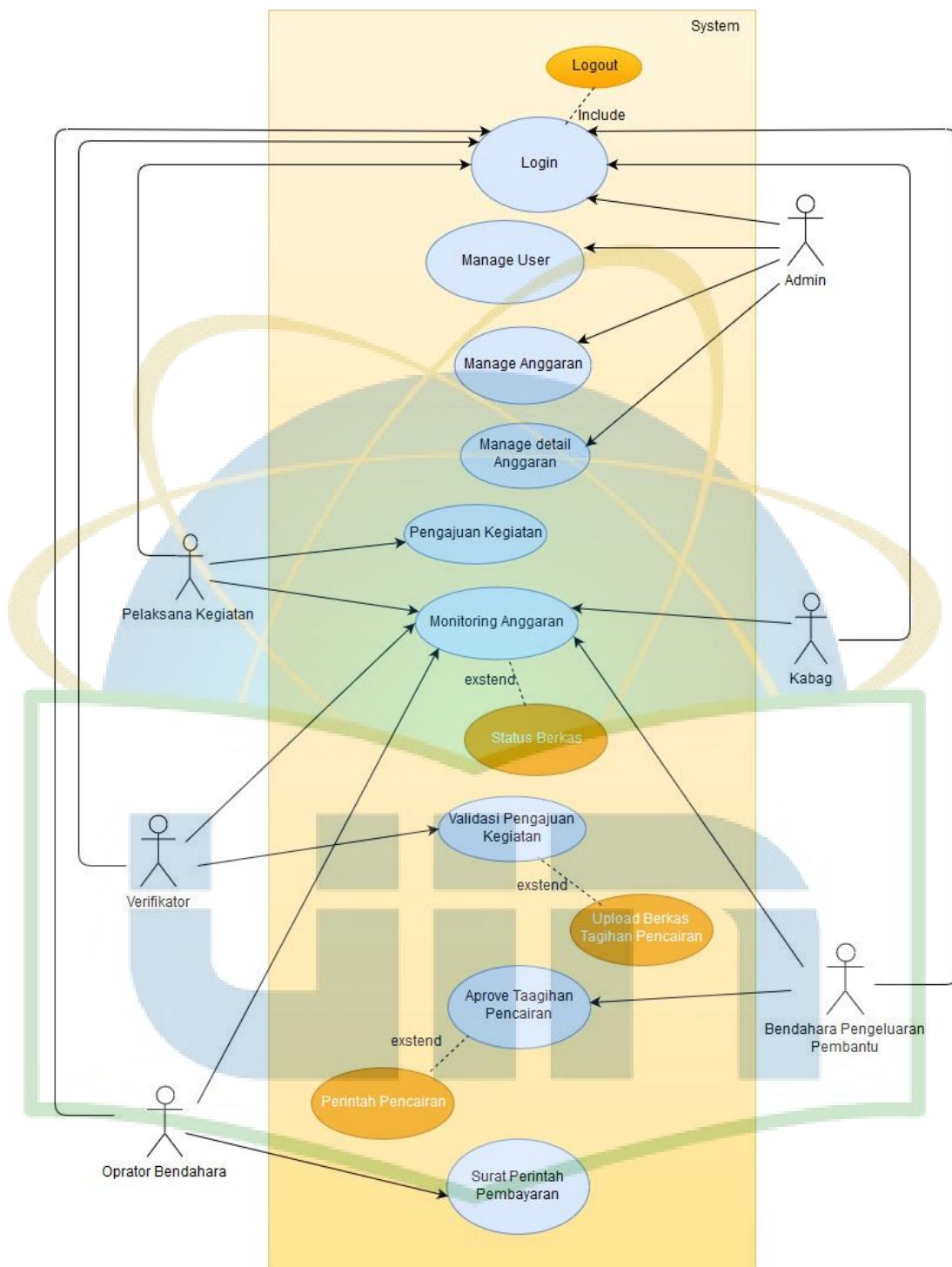


		kegiatan proses <i>manage user</i> yang akan menggunakan sistem ini berdasarkan levelnya masing-masing. <i>Manage user</i> ini juga dapat membuka atau menutup hak akses terhadap <i>user</i> yang sudah terdaftar.	
5.	Menginput Rancangan Kegiatan	<i>Use Case</i> ini menggambarkan kegiatan dalam penginputan kegiatan yang akan dilaksanakan	Kepala Bagian
6.	Menginput Berkas Tagihan Pencairan (SPP)	<i>Use Case</i> ini Menggambarkan proses Penginputan data rancangan kegiatan dan diteruskan menjadi Berkas Tagihan Pencairan, dan juga dapat melihat status	Pelaksana Kegiatan

		letak berkas pengjuannya	
7.	Kelola data Berkas Tagihan Pencairan	<i>Use Case</i> ini menggambarkan proses penerimaan berkas yang akan dikoreksi kelengkapan berkas, apakah memenuhi persyaratan dan memberikan pesan bila berkas ditolak	verifikator
8.	Validasi Berkas Tagihan Pencairan	<i>Use Case</i> Ini menggambarkan proses pemvalidasian data untuk dijadikannya Surat Perintah Pembayaran (SPP)	Bendahara Pengeluaran Pembantu (BPP)
9.	Inputan Surat Perintah pembayaran	<i>Use Case</i> ini menggambarkan proses pembuatan data Surat Perintah Pembayaran (SPP) dan nantinya SPP itu Akan dicetak dan di	Oprator

		tandatangani oleh Pejabat Pembuat Komitmen (PPK)	
--	--	--	--





Gamabar 4. 5 Use Case Diagram

### c) Narasi Use Case Diagram

Narasi *use case* digunakan untuk mendeskripsikan interaksi dan langkah-langkah yang terjadi antara sistem dengan penggunanya secara textual. Berikut adalah narasi *use case* yang akan menjelaskan *use case* yang telah dibuat sebelumnya.

#### 1. Narasi *Use Case Login*

Tabel 4. 2 *Use Case Login*

<i>Use case name</i>	<i>Login</i>									
<i>Use case Id</i>	1									
<i>Actor</i>	Admin, Kabag, pelaksana bagian, verifikator, BPP, Oprator Bendahara									
<i>Description</i>	<i>Use Case</i> ini menggambarkan kegiatan masuk kedalam sistem dengan data yang telah diregistrasi oleh Admin.									
<i>Pre condition</i>	Pengguna sistem harus telah terdaftar di dalam sistem.									
<i>Trigger</i>	<i>Use case</i> ini dilakukan agar pengguna sistem dapat masuk kedalam sistem dan keluar dari sistem serta menggunakan sistem sesuai dengan level pengguna.									
<i>Typical course of events</i>	<table border="1"> <thead> <tr> <th><i>Actor Action</i></th> <th><i>System Response</i></th> </tr> </thead> <tbody> <tr> <td>1. Membuka sistem</td> <td>2. Menampilkan halaman <i>login</i></td> </tr> <tr> <td>3. Masukkan “<i>username</i>”</td> <td></td> </tr> <tr> <td>4. Masukkan “<i>password</i>”</td> <td></td> </tr> </tbody> </table>	<i>Actor Action</i>	<i>System Response</i>	1. Membuka sistem	2. Menampilkan halaman <i>login</i>	3. Masukkan “ <i>username</i> ”		4. Masukkan “ <i>password</i> ”		
<i>Actor Action</i>	<i>System Response</i>									
1. Membuka sistem	2. Menampilkan halaman <i>login</i>									
3. Masukkan “ <i>username</i> ”										
4. Masukkan “ <i>password</i> ”										

	5. Pilih <i>login</i>	4. Validasi <i>username</i> dan <i>password</i>
		5. Menampilkan halaman utama pengguna
<i>Alternate courses</i>	Alternatif no. 3 dan 4 jika <i>username</i> dan <i>password</i> tidak sesuai maka akan menampilkan pesan data salah dan diminta mengisi kembali no. 3 dan 4.	
<i>Conclusion</i>	<i>Use case</i> ini dibuat untuk masuk ke dalam sistem.	
<i>Post condition</i>	Setelah validasi <i>Username</i> dan <i>password</i> benar sistem akan membuka halaman utama sesuai dengan level pengguna.	

## 2. Narasi *Use Case Manage User*

Tabel 4. 3 *Use Case Manage User*

<i>Use case name</i>	<i>Manage User</i>
<i>Use case Id</i>	2
<i>Actor</i>	Admin
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola data Mata Anggaran Kegiatan. Dalam <i>use case</i> ini ada beberapa proses untuk mengelolanya yaitu, <i>input</i> , <i>update</i> dan <i>delete</i> .
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .

<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan sehala jenis kegiatan yang nantinya dapat dikelola.	
<i>Typical course of events</i>	<i>Actor Action</i>	<i>System Response</i>
	1. pilih menu <i>Manage User</i>	2. Menampilkan halaman <i>Manage User</i>
	3. Pilih menu <i>Tambah data User</i>	4. Menampilkan form input data User
	5. Mengisi data	
	6. Pilih <i>simpan</i>	7. Sistem akan menyimpan dalam database
		8. menampilkan Data User
	9. pilih “ <i>Edit</i> ”	10. menghubungkan database
		11. menampilkan halaman edit
	12. <i>Edit</i> data Data User	
	13. Klik “ <i>Simpan</i> ”	14. menghubungkan ke database
		15. Menampilkan Data User
	16. Klik “ <i>Hapus</i> ”	17. <i>Menghapus</i> database
		18. <i>Menghapus</i> Data User
		19. Menampilkan Data User

<i>Alternate courses</i>	<ol style="list-style-type: none"> <li>1. Jika pengguna tidak menginginkan perubahan data bisa kembali ke No. 2</li> <li>2. Jika pengguna tidak menginginkan mengedit data bisa kembali ke No. 8</li> <li>3. Jika pengguna tidak menginginkan mehapus data bisa kembali ke NO. 15</li> </ol>
<i>Conclusion</i>	<i>Data Berhasil di manage</i>
<i>Post condition</i>	Data informasi tersimpan dalam <i>database</i>

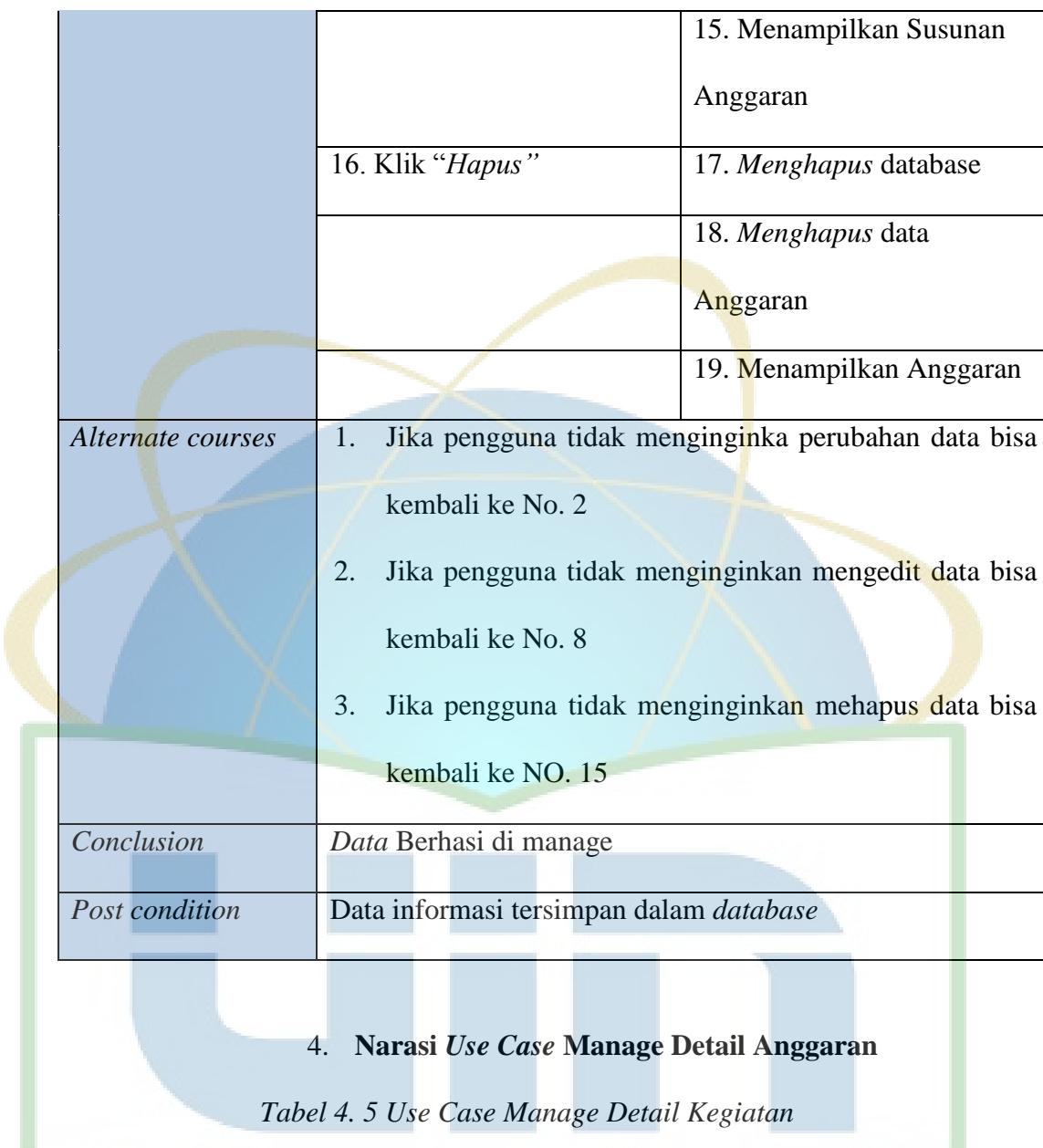
<i>Alternate courses</i>	<ol style="list-style-type: none"> <li>1. Jika pengguna tidak menginginkan perubahan data bisa kembali ke No. 2</li> <li>2. Jika pengguna tidak menginginkan mengedit data bisa kembali ke No. 8</li> <li>3. Jika pengguna tidak menginginkan mehapus data bisa kembali ke NO. 15</li> </ol>
<i>Conclusion</i>	<i>Data Berhasil di manage</i>
<i>Post condition</i>	Data informasi tersimpan dalam <i>database</i>

### 3. Narasi **Use Case Manage Anggaran**

Tabel 4. 4 Use Case Manage Anggaran

<i>Use case name</i>	<i>Manage Anggaran</i>
<i>Use case Id</i>	3
<i>Actor</i>	admin

<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola data jenis kegiatan. Dalam <i>use case</i> ini ada beberapa proses untuk <i>me-manage</i> diantaranya, <i>input</i> , <i>update</i> dan <i>delete</i> .		
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .		
<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan sehala jenis kegiatan yang nantinya dapat dikelola.		
<i>Typical course of events</i>	<i>Admin</i>	<i>System Response</i>	
	1. pilih menu <i>Manage Manage Anggaran</i>	2. Menampilkan halaman <i>Manage Anggaran</i>	
	3. Pilih menu <i>Tambah Jumlah pagu</i>	4. Menampilkan <i>form input Pagu</i>	
	5. Mengisi data <i>Pagu</i>		
	6. Pilih <i>simpan</i>	7. Sistem akan menyimpan dalam database	
		8. menampilkan Anggaran	
	9. pilih “ <i>Edit</i> ”	10. menghubungkan database	
		11. menampilkan halaman edit	
	12. <i>Edit</i> data Anggaran		
	13. Klik “ <i>Simpan</i> ”	14. menghubungkan ke database	



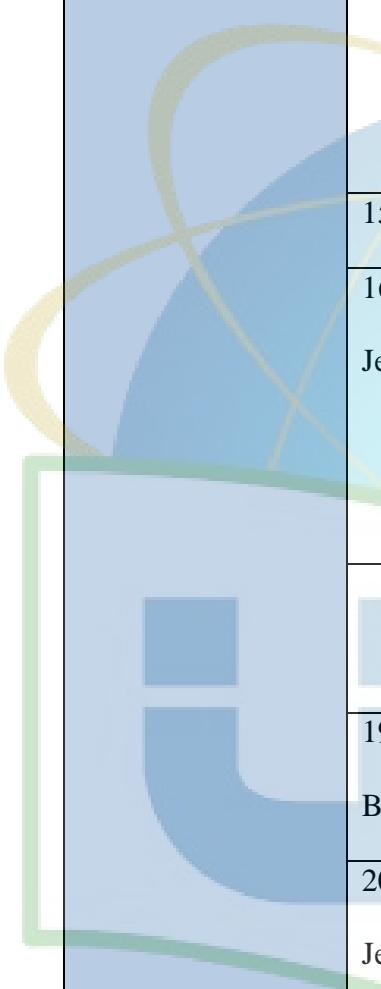
	15. Menampilkan Susunan Anggaran
	16. Klik "Hapus"
	17. Menghapus database
	18. Menghapus data Anggaran
	19. Menampilkan Anggaran
<i>Alternate courses</i>	<p>1. Jika pengguna tidak menginginkan perubahan data bisa kembali ke No. 2</p> <p>2. Jika pengguna tidak menginginkan mengedit data bisa kembali ke No. 8</p> <p>3. Jika pengguna tidak menginginkan menghapus data bisa kembali ke NO. 15</p>
<i>Conclusion</i>	Data Berhasil di manage
<i>Post condition</i>	Data informasi tersimpan dalam <i>database</i>

#### 4. Narasi Use Case Manage Detail Anggaran

Tabel 4. 5 Use Case Manage Detail Kegiatan

<i>Use case name</i>	Manage Detail Anggaran
<i>Use case Id</i>	4
<i>Actor</i>	Admin
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola data Kode kegiatan, nama kegiatan, pagu kegiatan, <i>input</i> , <i>update</i> dan <i>delete</i> .

<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .	
<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan sebala jenis kegiatan yang nantinya dapat dikelola.	
<i>Typical course of events</i>	<i>Actor Action</i>	<i>System Response</i>
	1. pilih menu <i>Manage Detail Kegiatan</i>	2. Sistem Akan Menampilkan halaman <i>Manage Detail Kegiatan</i>
		3. Sistem Menampilkan <i>Tambah data Akun Kegiatan</i>
	4. Pilih Tambah Akun Kegiatan	5. Sistem akan Menampilkan form <i>Tambah Akun Kegiatan</i>
		6. Sistem Menampilkan <i>Tambah Akun Kegiatan</i>
	7. Input Akun Kegiatan	
	8. Pilih Nama Kegiatan	9. Sistem akan Menampilkan form <i>Tambah nama Kegiatan</i>
		10. Sistem Menampilkan <i>Tambah Nama Kegiatan</i>
	11. Input Nama Kegiatan	



12. Pilih Tambah Jumlah Pagu Kegiatan	13. Sistem akan Menampilkan <i>form</i> Tambah Jumlah Pagu Kegiatan
	14. Sistem Menampilkan Tambah Jumlah Pagu Kegiatan Kegiatan
15. Input Jumlah Kegiatan	
16. Pilih Tambah Akun Jenis Belanja	17. Sistem akan Menampilkan <i>form</i> Tambah Akun Jenis Belanja
	18. Sistem Menampilkan Tambah Akun jenis belanja
19. Input Akun Jenis Belanja	
20. Pilih Tambah Nama Jenis Belanja	21. Sistem akan Menampilkan <i>form</i> Tambah Jenis Belanja
	22. Sistem Menampilkan Tambah Nama Jenis Belanja
23. Input Nama Jenis Belanja	

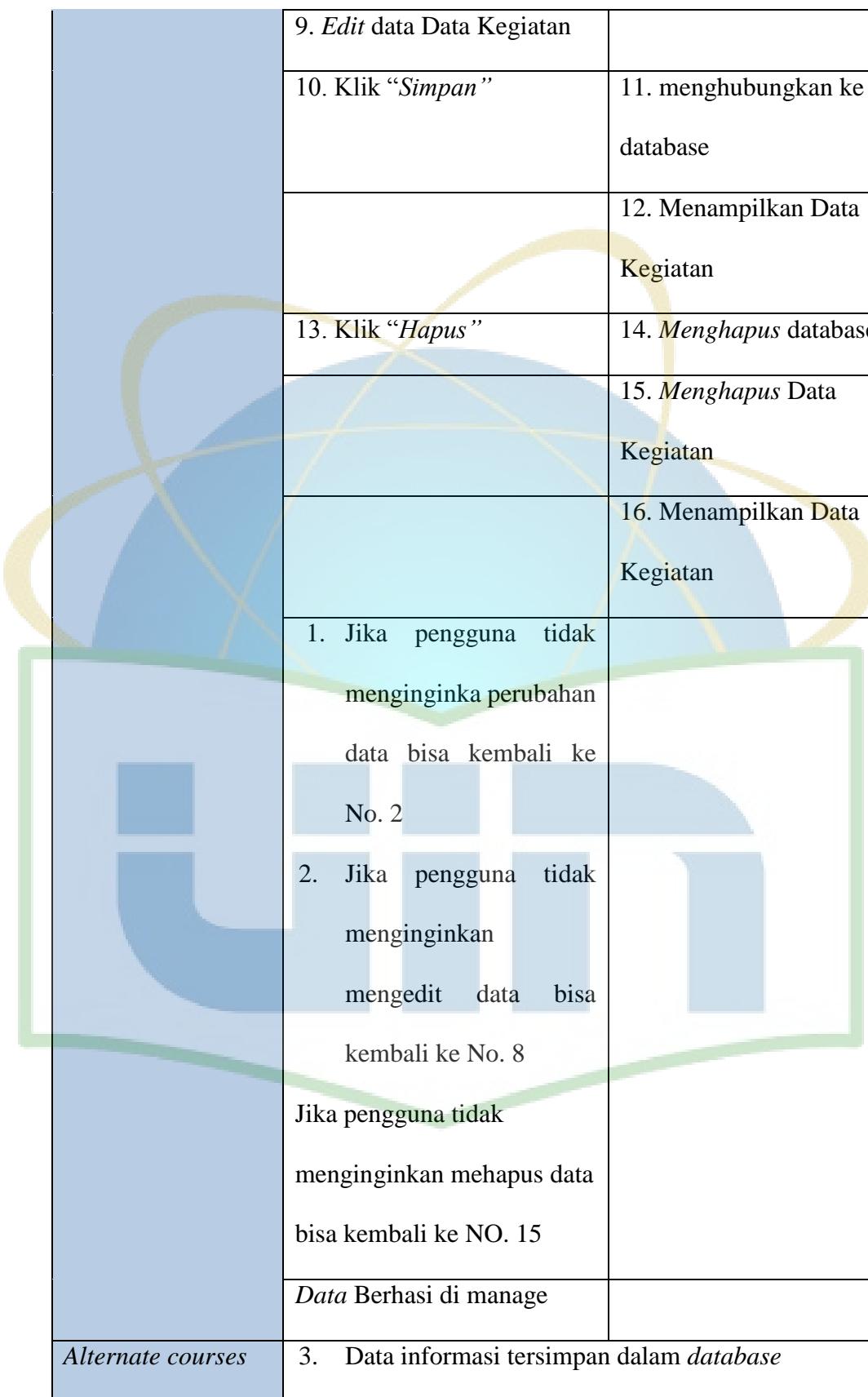
	24. Pilih Tambah Pagu Jenis Belanja	25. Sistem akan Menampilkan <i>form</i> Tambah Pagu Jenis Belanja
		26. Sistem Menampilkan Tambah Pagu Jenis Belanja
	27. Input Pagu Jenis Belanja	
	28. Pilih Tambah Jenis Surat Perintah Pembayaran	29. Sistem akan Menampilkan <i>form</i> Tambah Jenis Surat Perintah Pembayaran
		30. Sistem Menampilkan Jenis Surat Perintah Pembayaran
	31. Input Jenis Surat Perintah Pembayaran	
	32. Pilih <i>simpan</i>	33. Sistem akan menyimpan data Manage Detail Anggaran
		34. Data Tersimpan
	35. pilih “Edit”	36. Sistem akan menampilkan form edit

	37. mengubungkan ke data base
	38. Sistem menampilkan meu edit detail anggaran
39. <i>Edit</i> data manage detai anggaran	
40. Klik “Simpan”	41. menghubungkan ke database
	42. Menampilkan data terupdate
43. Klik “Hapus”	44. <i>Menghapus</i> database
	45. <i>Menghapus</i> data manage deatail anggaran
	46. Menampilkan Jenis Kegiatan
<i>Alternate courses</i>	<p>1. Jika pengguna tidak menginginkan perubahan data bisa kembali ke No. 1</p> <p>2. Jika pengguna tidak menginginkan mengedit data bisa kembali ke No. 32</p> <p>3. Jika pengguna tidak menginginkan mehapus data bisa kembali ke NO. 32</p>
<i>Conclusion</i>	<i>Data Berhasi di manage</i>
<i>Post condition</i>	Data informasi tersimpan dalam <i>database</i>

## 5. Narasi Use Case Pengajuan Kegiatan

Tabel 4.6 Use Case Pengajuan Kegiatan

<i>Use case name</i>	<i>Pengajuan Kegiatan</i>		
<i>Use case Id</i>	6		
<i>Actor</i>	Pelaksana Kegiatan		
<i>Description</i>	<p><i>Use case</i> ini menggambarkan pelaksana kegiatan proses pengajuan kegiatan. Dalam <i>use case</i> ini ada beberapa proses untuk mengelolanya yaitu, <i>input</i>, <i>update</i> dan <i>delete</i>.</p>		
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .		
<i>Trigger</i>	<p><i>Use case</i> ini bertujuan untuk menyimpan sebala jenis kegiatan yang nantinya dapat dikelola.</p>		
<i>Typical course of events</i>	<i>Actor Action</i>	<i>System Response</i>	
	1. pilih menu <i>Pengajuan Kegiatan</i>	2. Menampilkan halaman <i>Upload Pengajuan Kegiatan</i>	
	3. Pilih <i>simpan</i>	4. Sistem akan menyimpan dalam database	
		5. menampilkan Data Kegiatan	
	6. pilih “ <i>Edit</i> ”	7. menghubungkan database	
		8. menampilkan halaman edit	



	9. <i>Edit</i> data Data Kegiatan	
	10. Klik “Simpan”	11. menghubungkan ke database
		12. Menampilkan Data Kegiatan
	13. Klik “Hapus”	14. <i>Menghapus</i> database
		15. <i>Menghapus</i> Data Kegiatan
		16. Menampilkan Data Kegiatan
	1. Jika pengguna tidak menginginkan perubahan data bisa kembali ke No. 2  2. Jika pengguna tidak menginginkan mengedit data bisa kembali ke No. 8  Jika pengguna tidak menginginkan mehapus data bisa kembali ke NO. 15	
	<i>Data Berhasil di manage</i>	
<i>Alternate courses</i>	3. Data informasi tersimpan dalam <i>database</i>	

<i>Conclusion</i>	
<i>Post condition</i>	

## 6. Narasi Use Case Monitoring Anggaran

Tabel 4. 7 Narasi Use Case Monitoring Anggaran

<i>Use case name</i>	<i>Monitoring Anggaran</i>							
<i>Use case Id</i>	6							
<i>Actor</i>	Kabag, Pelaksana Kegiatan, Verifikator, Bendahara Pengeluaran Pembantu, Operator Bendahara							
<i>Description</i>	<p><i>Use case</i> ini menggambarkan kegiatan untuk mengelola data jenis kegiatan. Dalam <i>use case</i> ini ada beberapa proses untuk me-manage diantaranya, <i>input</i>, <i>update</i> dan <i>delete</i>.</p>							
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .							
<i>Trigger</i>	<i>Use case</i> ini bertujuan untuk menyimpan sebala jenis kegiatan yang nantinya dapat dikelola.							
<i>Typical course of events</i>	<table border="1"> <thead> <tr> <th><i>Actor Action</i></th> <th><i>System Response</i></th> </tr> </thead> <tbody> <tr> <td>1. pilih menu <i>Monitoring Anggaran</i></td> <td>2. Menampilkan halaman <i>Monitoring Anggaran</i></td> </tr> <tr> <td>3. Pilih menu <i>View Monitoring Anggaran</i></td> <td>4. Menampilkan <i>form</i> data Monitoring Anggaran</td> </tr> </tbody> </table>	<i>Actor Action</i>	<i>System Response</i>	1. pilih menu <i>Monitoring Anggaran</i>	2. Menampilkan halaman <i>Monitoring Anggaran</i>	3. Pilih menu <i>View Monitoring Anggaran</i>	4. Menampilkan <i>form</i> data Monitoring Anggaran	
<i>Actor Action</i>	<i>System Response</i>							
1. pilih menu <i>Monitoring Anggaran</i>	2. Menampilkan halaman <i>Monitoring Anggaran</i>							
3. Pilih menu <i>View Monitoring Anggaran</i>	4. Menampilkan <i>form</i> data Monitoring Anggaran							
<i>Conclusion</i>	<i>Data Berhasil Dimunculkan</i>							
<i>Post condition</i>	Data informasi tersimpan dalam <i>database</i>							

## 7. Narasi Use Case Status Berkas

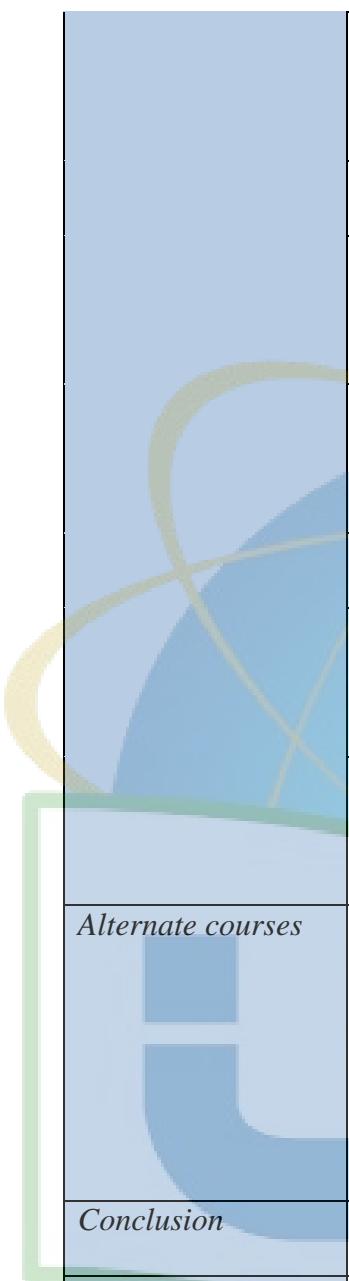
Tabel 4. 8 Narasi Use Case Status Berkas

<i>Use case name</i>	<i>Status Berkas</i>				
<i>Use case Id</i>	7				
<i>Actor</i>	Pelaksana Kegiatan, Verifikator, Bendahara Pengeluaran Pembantu, Operator Bendahara				
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola data jenis kegiatan. Dalam <i>use case</i> ini ada beberapa proses untuk me-manage diantaranya, <i>input</i> , <i>update</i> dan <i>delete</i> .				
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .				
<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan sehala jenis kegiatan yang nantinya dapat dikelola.				
<i>Typical course of events</i>	<i>Actor Action</i>	<i>System Response</i>			
	1. pilih menu <i>Status Berkas</i>	2. Menampilkan halaman <i>Status Berkas</i>			
	3. Pilih Update	4. Maka sistem akan Menyimpan di Database			
<i>Conclusion</i>	<i>Data Berhasil Dimunculkan</i>				
<i>Post condition</i>	<i>Data informasi tersimpan dalam database</i>				

## 8. Narasi Use Case Validasi Pengajuan Kegiatan

Tabel 4. 9 Narasi Use Case Validasi Pengajuan Kegiatan

<i>Use case name</i>	<i>Validasi Pengajuan Kegiatan</i>															
<i>Use case Id</i>	8															
<i>Actor</i>	Verifikator															
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola data jenis kegiatan. Dalam <i>use case</i> ini ada beberapa proses untuk <i>me-manage</i> diantaranya, <i>input</i> , <i>update</i> dan <i>delete</i> .															
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .															
<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan sebala jenis kegiatan yang nantinya dapat dikelola.															
<i>Typical course of events</i>	<table border="1"> <thead> <tr> <th><i>Actor Action</i></th> <th><i>System Response</i></th> </tr> </thead> <tbody> <tr> <td>1. pilih menu <i>Validasi Pengajuan Kegiatan</i></td> <td>2. Menampilkan halaman <i>Validasi Pengajuan Kegiatan</i></td> </tr> <tr> <td>3. Pilih menu <i>Tambah pesan</i></td> <td>4. Menampilkan form <i>Tulis Pesan</i></td> </tr> <tr> <td>5. Mengisi data</td> <td></td> </tr> <tr> <td>6. Pilih “<i>Kirim</i>”</td> <td>7. Sistem akan menyimpan di database</td> </tr> <tr> <td></td> <td>8. menampilkan</td> </tr> <tr> <td>9. pilih “<i>Validasi</i>”</td> <td>10. menghubungkan database</td> </tr> </tbody> </table>	<i>Actor Action</i>	<i>System Response</i>	1. pilih menu <i>Validasi Pengajuan Kegiatan</i>	2. Menampilkan halaman <i>Validasi Pengajuan Kegiatan</i>	3. Pilih menu <i>Tambah pesan</i>	4. Menampilkan form <i>Tulis Pesan</i>	5. Mengisi data		6. Pilih “ <i>Kirim</i> ”	7. Sistem akan menyimpan di database		8. menampilkan	9. pilih “ <i>Validasi</i> ”	10. menghubungkan database	
<i>Actor Action</i>	<i>System Response</i>															
1. pilih menu <i>Validasi Pengajuan Kegiatan</i>	2. Menampilkan halaman <i>Validasi Pengajuan Kegiatan</i>															
3. Pilih menu <i>Tambah pesan</i>	4. Menampilkan form <i>Tulis Pesan</i>															
5. Mengisi data																
6. Pilih “ <i>Kirim</i> ”	7. Sistem akan menyimpan di database															
	8. menampilkan															
9. pilih “ <i>Validasi</i> ”	10. menghubungkan database															



	11. menampilkan halaman Proses
12. <i>Upload</i> status Berkas	
13. Klik “Simpan”	14. menghubungkan ke database
	15. Menampilkan List data Tagihan Pencairan
16. Klik “Hapus”	17. <i>Menghapus</i> database
	18. <i>Menghapus</i> data Tagihan Pencairan
	19. Menampilkan data Tagihan Pencairan
<i>Alternate courses</i>	<p>1. Jika pengguna menganggap ada kesalahan data pada berkas pengajuan bisa lakukan perintah No. 2</p> <p>2. Jika pengguna menganggap tidak ada kesalahan pada berkas bisa lanjut ke No. 9</p>
<i>Conclusion</i>	<i>Data Berhasil di proses</i>
<i>Post condition</i>	Data informasi tersimpan dalam <i>database</i>

## 9. Narasi Use Case Upload Berkas Tagihan Pencairan

Tabel 4. 10 Narasi Use Case Upload Berkas Tagihan Pecairan

<i>Use case name</i>	<i>Upload Berkas Tagihan Pencairan</i>							
<i>Use case Id</i>	9							
<i>Actor</i>	Verifikator							
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengupload Berkas.							
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .							
<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan segala jenis kegiatan yang nantinya dapat dikelola.							
<i>Typical course of events</i>	<table border="1"> <thead> <tr> <th><i>Actor Action</i></th> <th><i>System Response</i></th> </tr> </thead> <tbody> <tr> <td>1. pilih menu <i>Upload Berkas</i></td> <td>2. Menampilkan halaman Upload Tagihan Pencairan</td> </tr> <tr> <td>3. Pilih Update</td> <td>4. Maka sistem akan Menyimpan di Database</td> </tr> </tbody> </table>	<i>Actor Action</i>	<i>System Response</i>	1. pilih menu <i>Upload Berkas</i>	2. Menampilkan halaman Upload Tagihan Pencairan	3. Pilih Update	4. Maka sistem akan Menyimpan di Database	
<i>Actor Action</i>	<i>System Response</i>							
1. pilih menu <i>Upload Berkas</i>	2. Menampilkan halaman Upload Tagihan Pencairan							
3. Pilih Update	4. Maka sistem akan Menyimpan di Database							
<i>Conclusion</i>	<i>Data Berhasil Dimunculkan</i>							
<i>Post condition</i>	Data informasi tersimpan dalam <i>database</i>							

## 10. Narasi Use Case Approve Tagihan Pencairan

Tabel 4. 11 Narasi Use Case Aprove Tagihan Pencairan

<i>Use case name</i>	<i>Approve Tagihan Pencairan</i>															
<i>Use case Id</i>	10															
<i>Actor</i>	Bendahara Pengeluaran Pembantu (BPP)															
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola data jenis kegiatan. Dalam <i>use case</i> ini ada beberapa proses untuk <i>me-manage</i> diantaranya, <i>input</i> , <i>update</i> dan <i>delete</i> .															
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .															
<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan sebala jenis kegiatan yang nantinya dapat dikelola.															
<i>Typical course of events</i>	<table border="1"> <thead> <tr> <th><i>Actor Action</i></th> <th><i>System Response</i></th> </tr> </thead> <tbody> <tr> <td>1. pilih menu <i>Approve Tagihan Pencairan</i></td> <td>2. Menampilkan halaman <i>Approve Tagihan Pencairan</i></td> </tr> <tr> <td>3. Pilih menu <i>Download Berkas</i></td> <td>4. Menampilkan <i>Form Approve atau Tolak</i></td> </tr> <tr> <td>5. Approve tagihan pencairan</td> <td></td> </tr> <tr> <td>6. Pilih “<i>Kirim</i>”</td> <td>7. Sistem akan menyimpan di database</td> </tr> <tr> <td></td> <td>8. menampilkan</td> </tr> <tr> <td>9. pilih “<i>Tolak</i>”</td> <td>10. menghubungkan database</td> </tr> </tbody> </table>	<i>Actor Action</i>	<i>System Response</i>	1. pilih menu <i>Approve Tagihan Pencairan</i>	2. Menampilkan halaman <i>Approve Tagihan Pencairan</i>	3. Pilih menu <i>Download Berkas</i>	4. Menampilkan <i>Form Approve atau Tolak</i>	5. Approve tagihan pencairan		6. Pilih “ <i>Kirim</i> ”	7. Sistem akan menyimpan di database		8. menampilkan	9. pilih “ <i>Tolak</i> ”	10. menghubungkan database	
<i>Actor Action</i>	<i>System Response</i>															
1. pilih menu <i>Approve Tagihan Pencairan</i>	2. Menampilkan halaman <i>Approve Tagihan Pencairan</i>															
3. Pilih menu <i>Download Berkas</i>	4. Menampilkan <i>Form Approve atau Tolak</i>															
5. Approve tagihan pencairan																
6. Pilih “ <i>Kirim</i> ”	7. Sistem akan menyimpan di database															
	8. menampilkan															
9. pilih “ <i>Tolak</i> ”	10. menghubungkan database															

	11. menampilkan menu tolak
	12. status Berkas ditolak
	13. Klik “Upload status”
	14. menghubungkan ke database
	15. Menampilkan status berkas
	16. Klik “Simpan”
	17. Menyimpan database
	18. Menyimpan validasi data tagihan pencairan
	19. Menampilkan data validasi tagihan pencairan
Alternate courses	<p>3. Jika pengguna menganggap ada kesalahan data pada berkas pengajuan bisa lakukan perintah No. 9</p> <p>4. Jika pengguna menganggap tidak ada kesalahan pada berkas bisa lanjut ke No. 13</p>
Conclusion	Data Berhasil di proses
Post condition	Data informasi tersimpan dalam <i>database</i>

## 11. Narasi Use Case Perintah Pencairan

Tabel 4. 12 Narasi Use Case Perintah pencairan

<i>Use case name</i>	<i>Perintah Pencairan</i>							
<i>Use case Id</i>	11							
<i>Actor</i>	Bendahara Pengeluaran Pembantu							
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengupload Berkas.							
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .							
<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan segala jenis kegiatan yang nantinya dapat dikelola.							
<i>Typical course of events</i>	<table border="1"> <thead> <tr> <th><i>Actor Action</i></th> <th><i>System Response</i></th> </tr> </thead> <tbody> <tr> <td>1. pilih menu <i>Perintah Pencairan</i></td> <td>2. Menampilkan halaman Upload Perintah Pencairan</td> </tr> <tr> <td>3. Pilih Update</td> <td>4. Maka sistem akan Menyimpan di Database</td> </tr> </tbody> </table>	<i>Actor Action</i>	<i>System Response</i>	1. pilih menu <i>Perintah Pencairan</i>	2. Menampilkan halaman Upload Perintah Pencairan	3. Pilih Update	4. Maka sistem akan Menyimpan di Database	
<i>Actor Action</i>	<i>System Response</i>							
1. pilih menu <i>Perintah Pencairan</i>	2. Menampilkan halaman Upload Perintah Pencairan							
3. Pilih Update	4. Maka sistem akan Menyimpan di Database							
<i>Conclusion</i>	<i>Data Berhasil Dimunculkan</i>							
<i>Post condition</i>	Data informasi tersimpan dalam <i>database</i>							

## 12. Narasi Use Case Surat Perintah Pembayaran (SPP)

Tabel 4. 13 Narasi Use Case Surat Perintah Pembayaran

<i>Use case name</i>	<i>Surat Perintah Pembayaran (SPP)</i>							
<i>Use case Id</i>	12							
<i>Actor</i>	Operator Bendahara							
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengupload Berkas.							
<i>Pre condition</i>	Pengguna sistem harus <i>login</i> .							
<i>Trigger</i>	<i>Use case</i> Ini bertujuan untuk menyimpan segala jenis kegiatan yang nantinya dapat dikelola.							
<i>Typical course of events</i>	<table border="1"> <thead> <tr> <th><i>Actor Action</i></th> <th><i>System Response</i></th> </tr> </thead> <tbody> <tr> <td>1. pilih menu <i>Upload Berkas</i></td> <td>2. Menampilkan halaman Upload Surat Perintah Pembayaran (SPP)</td> </tr> <tr> <td>3. Pilih Update</td> <td>4. Maka sistem akan Menyimpan di Database</td> </tr> </tbody> </table>	<i>Actor Action</i>	<i>System Response</i>	1. pilih menu <i>Upload Berkas</i>	2. Menampilkan halaman Upload Surat Perintah Pembayaran (SPP)	3. Pilih Update	4. Maka sistem akan Menyimpan di Database	
<i>Actor Action</i>	<i>System Response</i>							
1. pilih menu <i>Upload Berkas</i>	2. Menampilkan halaman Upload Surat Perintah Pembayaran (SPP)							
3. Pilih Update	4. Maka sistem akan Menyimpan di Database							
<i>Conclusion</i>	<i>Data Berhasil Dimunculkan</i>							
<i>Post condition</i>	<i>Data informasi tersimpan dalam database</i>							

### 13. Narasi Use Case Logout

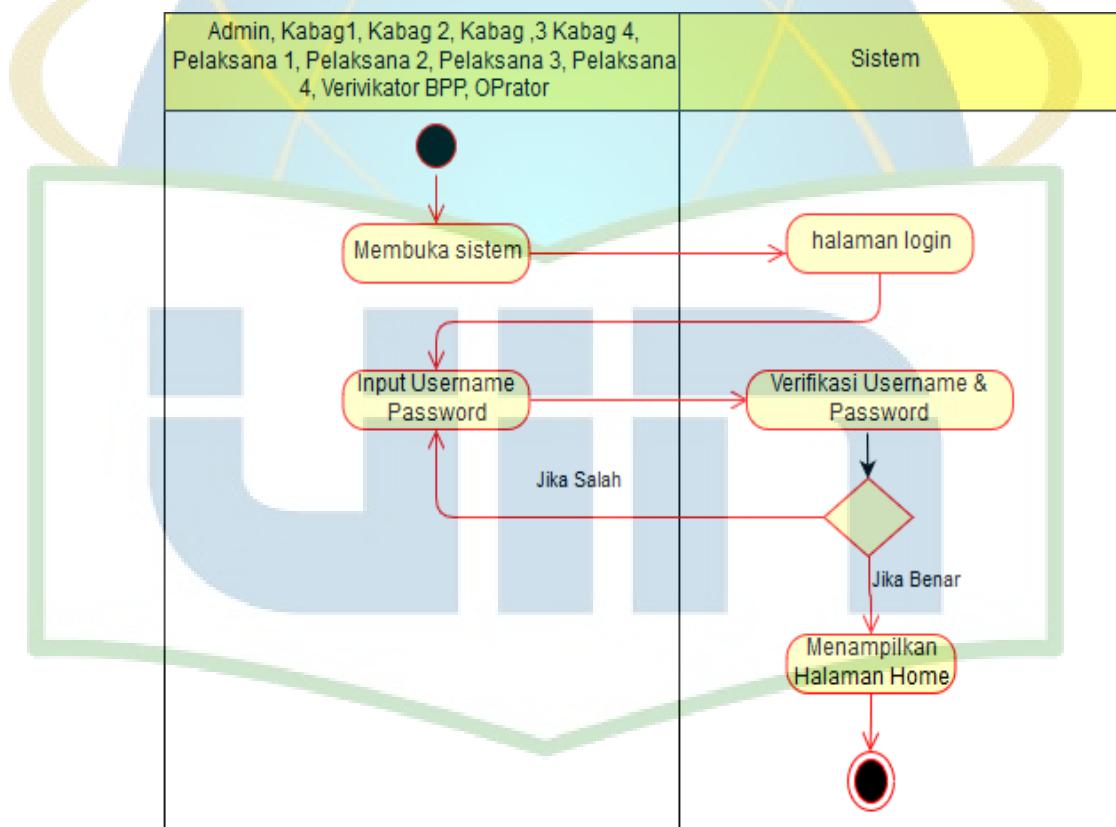
Tabel 4. 14. Narasi Use Case Logout

<i>Use case name</i>	<i>Logout</i>		
<i>Use case Id</i>	1		
<i>Actor</i>	Admin, Kabaga, Pelaksana Kegiatan pelaksana bagian 4, verifikator, BPP, Oprator, PPK		
<i>Description</i>	<i>Use Case</i> ini menggambarkan kegiatan masuk kedalam sistem dengan data yang telah diregistrasi oleh Admin.		
<i>Pre condition</i>	Pengguna sistem harus telah terdaftar di dalam sistem.		
<i>Trigger</i>	<i>Use case</i> ini dilakukan agar pengguna sistem dapat masuk kedalam sistem dan keluar dari sistem serta menggunakan sistem sesuai dengan level pengguna.		
<i>Typical course of events</i>	<i>Actor Action</i>	<i>System Response</i>	
	1. Menu Logut	2. Menampilkan halaman <i>logout</i>	
	3 Pilih Pilih Logut		
		4. Menampilkan halaman utama pengguna	

#### 4.2.1.2 Perancangan Activity Diagram

*Activity Diagram* menggambarkan aktifitas-aktifitas yang terjadi dalam sistem dan *user*. Berdasarkan dari narasi *use case* maka dapat digambarkan aktivitas-aktivitas yang terjadi atau alur kerja dalam *use case*. Aliran kerja tersebut digambarkan secara grafis dengan activity diagram. Berikut ini adalah *activity diagram* dari masing-masing *use case*:

##### 1. *Activity Diagram Login*

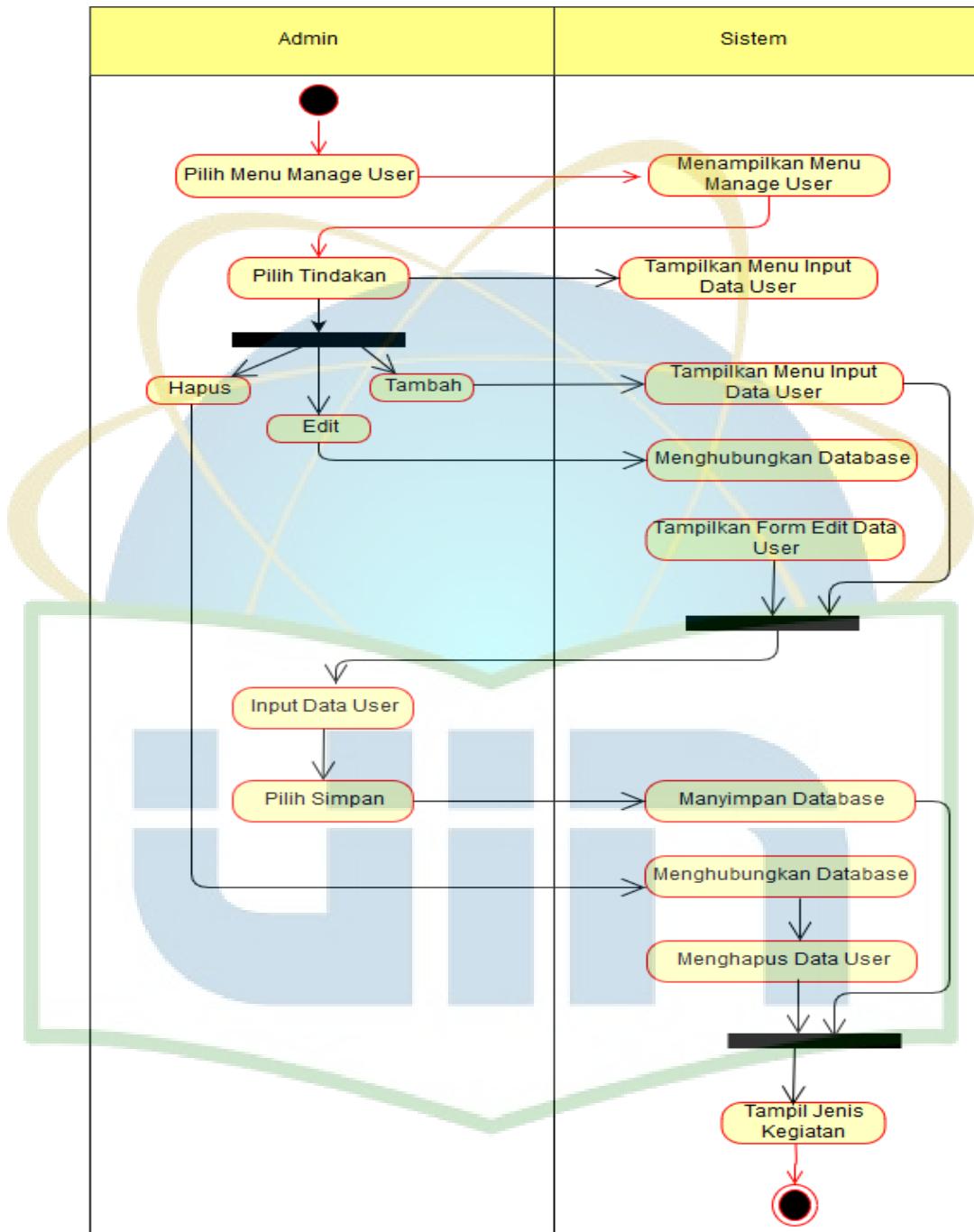


Gambar 4. 6 Activity Diagram Login

Aktivitas yang terjadi pada gambar 4.6 adalah aktivitas aktor yang melakukan *login* untuk dapat mengakses sistem. Pertama aktor memilih *login* pada sistem, kemudian sistem akan memunculkan *form login*, pada *form* tersebut aktor memasukan *user id* dan *password* yang telah dimiliki. Setelah memasukan data *user id* dan *password*, aktor dapat klik *login* agar sistem dapat melakukan proses pengecekan terhadap data yang telah di *input*. Jika *username* dan *password* yang di masukan tidak terdaftar maka sistem akan menampilkan pesan gagal serta mengharuskan aktor untuk dapat melakukan *input* ulang *username* dan *password* dan jika *username* dan *password* yang di *input* sudah benar maka sistem akan menampilkan pesan berhasil dan langsung menampilkan halaman utama.



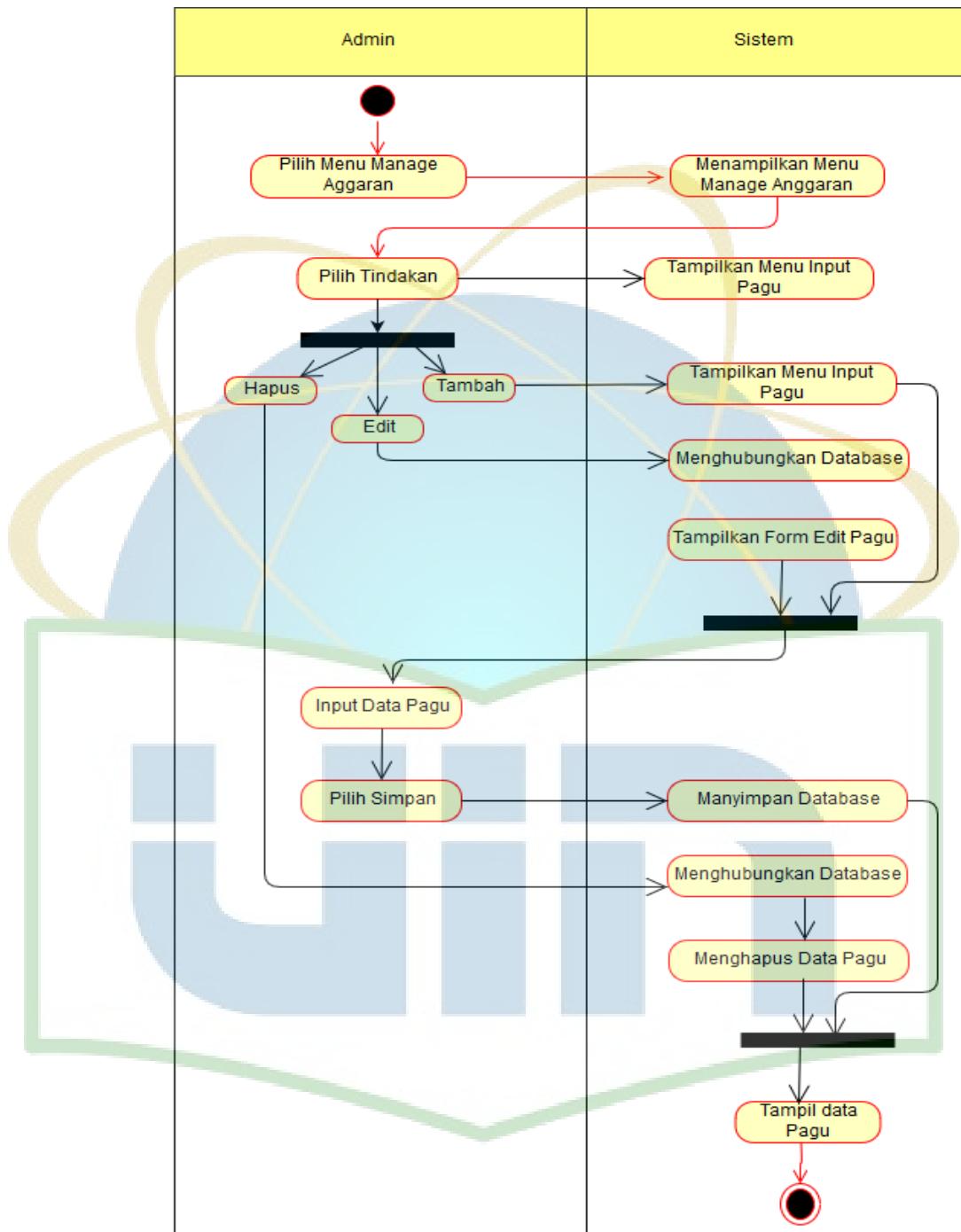
## 2. Activity Diagram Manage User



Gambar 4 7 Activity Diagram *Manage User*

*Manage user* adalah aktifitas dimana aktor dapat mengelola *user* yang akan menggunakan sistem ini. Pengelolaan user diantaranya menambahkan user baru, *edit user*, dan hapus *user*. Aktor yang dapat melakukan aktifitas ini adalah Wadek II, dimana Wadek II membuka sistem informasi monitoring dan evaluasi, kemudian memilih menu *manage user*, kemudian sistem akan menampilkan halaman *manage user* yang berisi *list user* yang sudah ada beserta tindakan pada pengelolaan, yaitu tambah, *edit* dan hapus. Jika aktor ingin menambahkan *user*, aktor memilih tambah, kemudian sistem akan menampilkan *form input user* baru. Setelah aktor mengisi *form input user* kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data *user*, aktor klik *edit* pada data *user* yang ada kemudian sistem akan menampilkan *form edit user*, dan aktor pilih simpan maka data *user* yang telah diperbaharui akan tersimpan di *database*. Kemudian jika aktor ingin menghapus data *user*, aktor tinggal memilih data *user* yang akan dihapus kemudian klik *hapus*, maka sistem akan menghubungkan ke *database* dan data *user* akan terhapus. Kemudian sistem akan menampilkan *list* data *Manage user*.

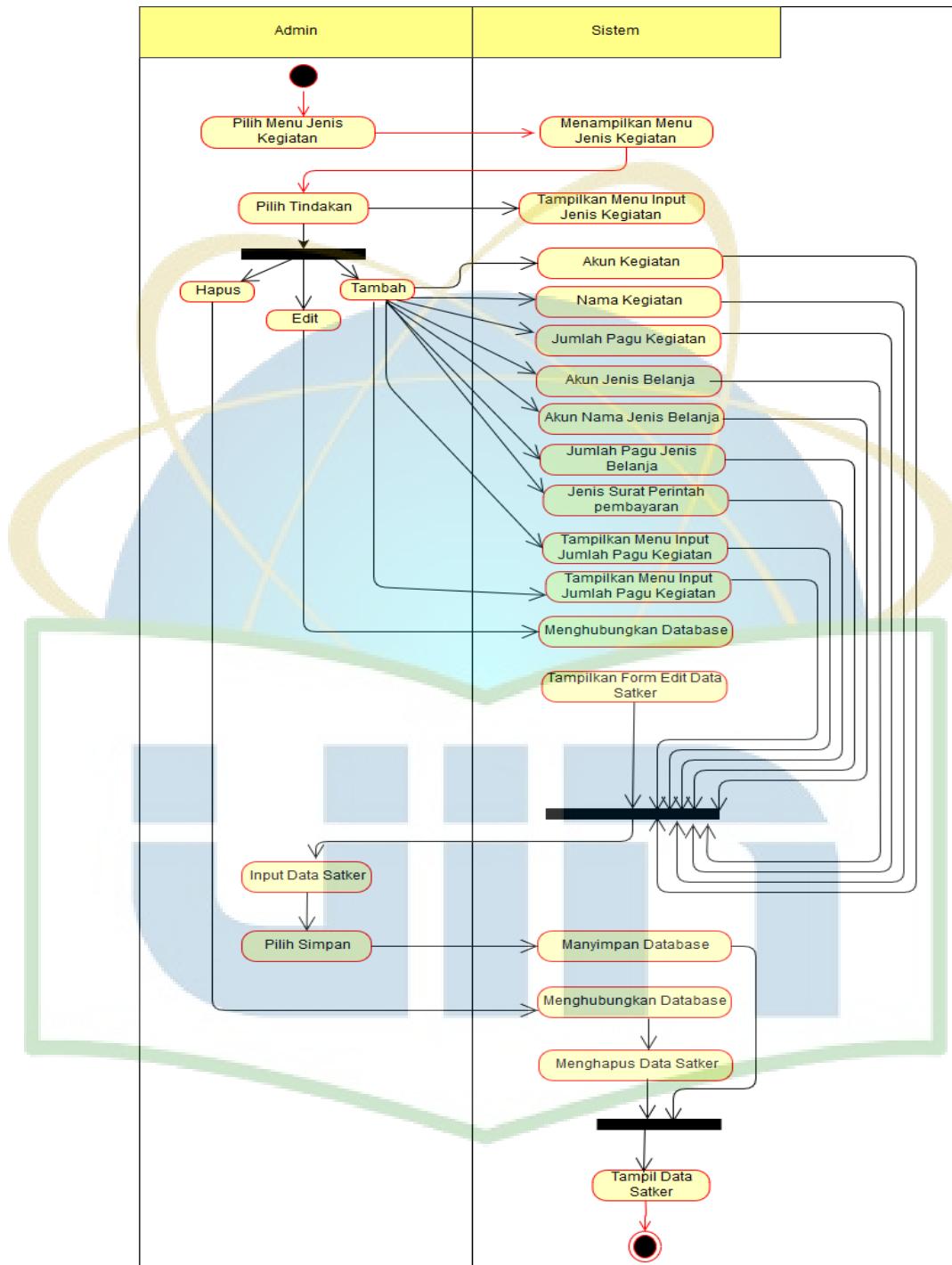
### 3. Activity Diagram Manage Anggaran



Gambar 4. 8 Activity Diagram Manage Anggaran

*Manage Anggaran* adalah aktifitas dimana aktor dapat mengelola *anggaran* yang akan menggunakan sistem ini. Pengelolaan anggaran diantaranya menambahkan Jumlah Anggaran baru, *edit Anggaran*, dan hapus *Anggaran*.. Aktor yang dapat melakukan aktifitas ini adalah Admin, dimana admin membuka sistem informasi monitoring keuangan, kemudian memilih menu *manage anggaran*, kemudian sistem akan menampilkan halaman *manage anggaran* yang berisi *list anggaran* yang sudah ada beserta tindakan pada pengelolaan, yaitu tambah, *edit* dan hapus. Jika aktor ingin menambahkan *anggaran*, aktor memilih tambah, kemudian sistem akan menampilkan *form input anggaran* baru. Setelah aktor mengisi *form input anggaran* kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data anggaran, aktor klik *edit* pada data *user* yang ada kemudian sistem akan menampilkan *form edit user*, dan aktor pilih simpan maka data *user* yang telah diperbarui akan tersimpan di *database*. Kemudian jika aktor ingin menghapus data *anggaran*, aktor tinggal memilih data anggaran yang akan dihapus kemudian klik hapus, maka sistem akan menghubungkan ke *database* dan data anggaran akan terhapus. Kemudian sistem akan menampilkan *list data Anggaran* kembali *Activity Diagram Input Data Keuangan*

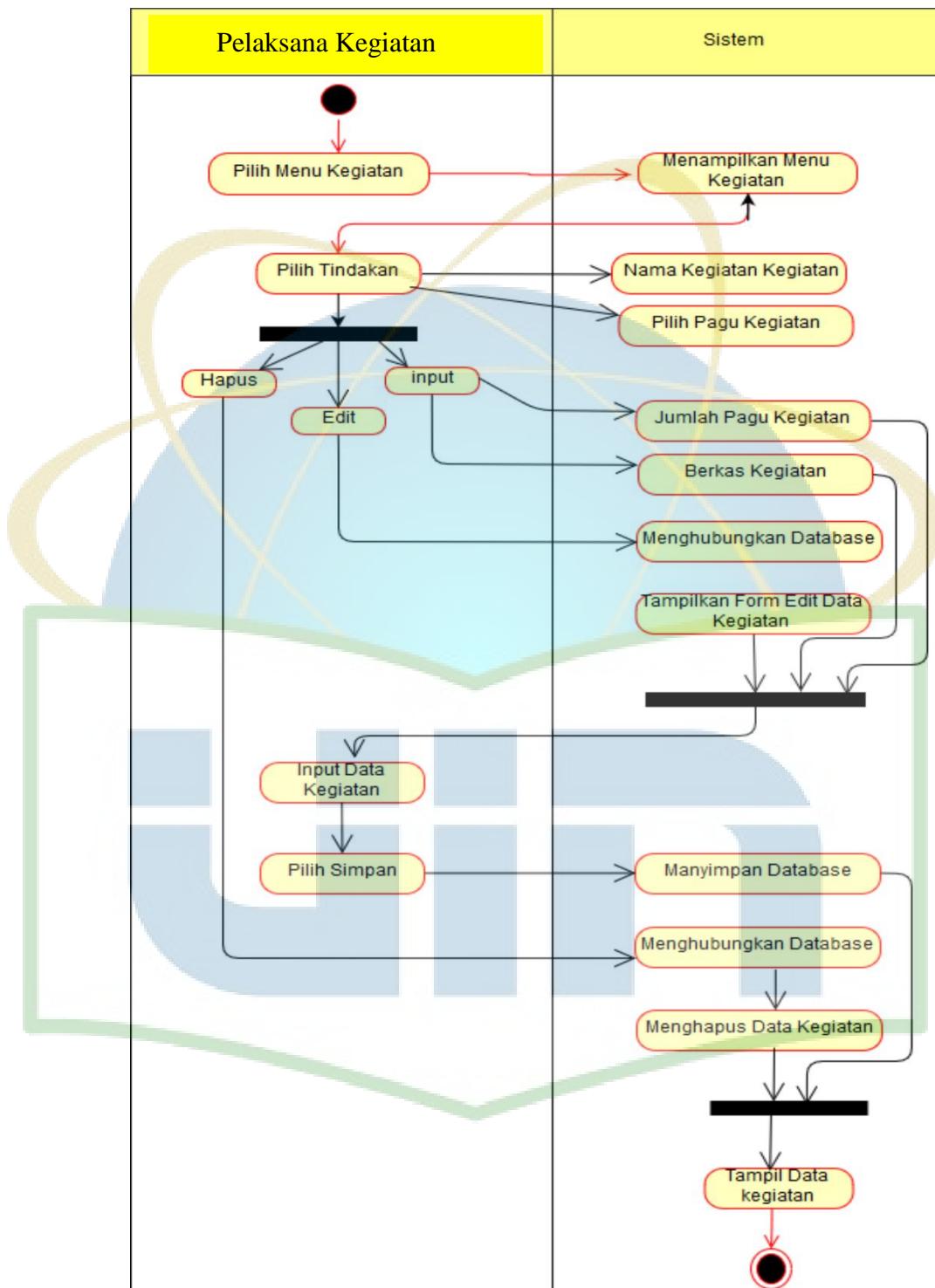
#### 4. Activity Diagram Manage Detail Anggaran



Gambar 4. 9 Activity Diagram Manage Detail Anggaran

*Jenis Kegiatan* adalah aktifitas dimana aktor dapat mengelola *jenis kegiatan* yang akan menggunakan sistem ini. Pengelolaan jenis kegiatan diantaranya menambahkan jenis kegiatan baru, *edit jenis kegiatan*, dan hapus *jenis kegiatan*. Aktor yang dapat melakukan aktifitas ini adalah Admin, dimana Admin membuka sistem informasi monitoring anggaran keuangan, kemudian memilih menu *jenis kegiatan*, kemudian sistem akan menampilkan halaman *Jenis Kegiatan* yang berisi *list jenis kegiatan* yang sudah ada beserta tindakan pada pengelolaan, yaitu tambah, *edit* dan hapus. Jika aktor ingin menambahkan *jenis kegiatan*, aktor memilih tambah, kemudian sistem akan menampilkan *form input nama kegiatan*, *menu input jenis belanja*, dan *menu input jumlah pagu*. Setelah aktor mengisi *form input jenis kegiatan* kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data *jenis kegiatan*, aktor klik *edit* pada data *jenis kegiatan* yang ada kemudian sistem akan menampilkan *form edit jenis kegiatan*, dan aktor pilih simpan maka data *user* yang telah diperbaharui akan tersimpan di *database*. Kemudian jika aktor ingin menghapus data *jenis kegiatan*, aktor tinggal memilih data *user* yang akan dihapus kemudian klik *hapus*, maka sistem akan menghubungkan ke *database* dan data *jenis kegiatan* akan terhapus. Kemudian sistem akan menampilkan *list data jenis kegiatan* kembali *Activity Diagram Manage Anggaran*.

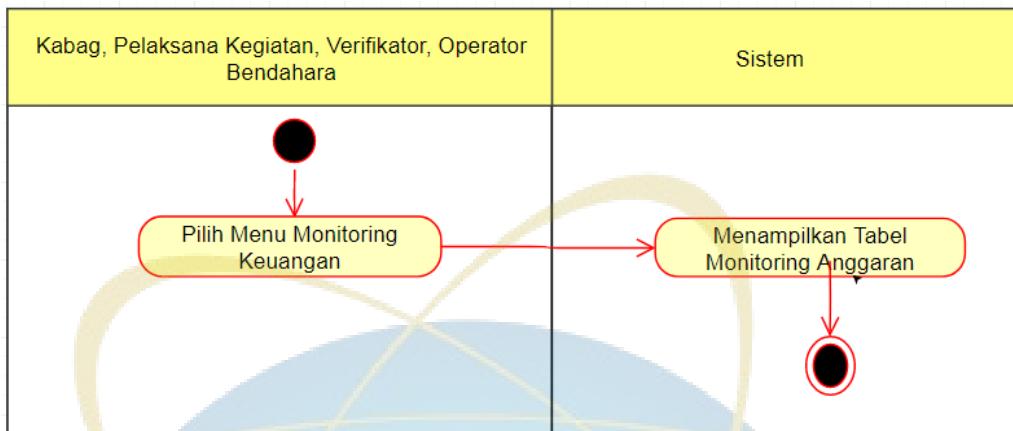
## 5. Activity Diagram Pengajuan Kegiatan



Gambar 4. 10 Activity Pengajuan Kegiatan

*Input Kegiatan* adalah aktifitas dimana aktor dapat mengelola *Kegiatan* yang akan menggunakan sistem ini. Pengelolaan Satker diantaranya menambahkan Satker baru, *edit Kegiatan*, dan *hapus kegiatan*. Aktor yang dapat melakukan aktifitas ini adalah Pelaksana Kegiatan, dimana Para aktor tersebut membuka sistem informasi monitoring anggaran keuangan. kemudian memilih menu *Input Kegiatan*, kemudian sistem akan menampilkan halaman Input Kegiatan yang berisi *list Input Kegiatan* yang sudah ada beserta tindakan pada pengelolaan, yaitu tambah, *edit* dan *hapus*. Jika aktor ingin menambahkan *Kegiatan*, aktor memilih tambah, kemudian sistem akan menampilkan *form input Kegiatan* baru, menampilkan *form input nama kegiatan*, dan menampilkan *form input jumlah pagu kegiatan*. Setelah aktor mengisi *form* tersebut kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data *kegiatan*, aktor klik *edit* pada data *kegiatan* yang ada kemudian sistem akan menampilkan *form edit kegiatan*, dan aktor pilih simpan maka data *kegiatan* yang telah diperbarui akan tersimpan di *database*. Kemudian jika aktor ingin menghapus data *kegiatan*, aktor tinggal memilih data user yang akan dihapus kemudian klik *hapus*, maka sistem akan menghubungkan ke *database* dan data *kegiatan* akan terhapus. Kemudian sistem akan menampilkan *list* data Input Kegiatan.

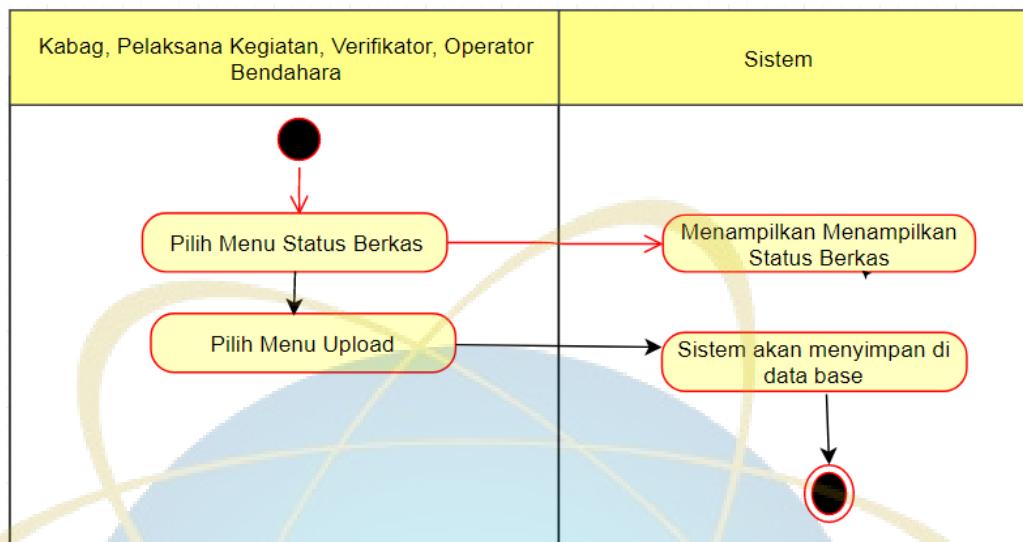
## 6. Activity Diagram Monitoring Anggaran



*Gambar 4. 11 Activity Diagram Monitoring Anggaran*

*Monitoring Keuangan* ini adalah aktifitas memonitor Keuangan unit yang dilakukan oleh masing-masing bagian di Biro Hukum dan KLN. Halaman yang ditampilkan pun juga hanya halaman *monitoring* Keuangan Bagian, tidak keseluruhan. Dan hanya dapat di lihat oleh bagian masing-masing dan sistem akan menampilkan trakc berkas anggaran sesuai dengan bagiannya.

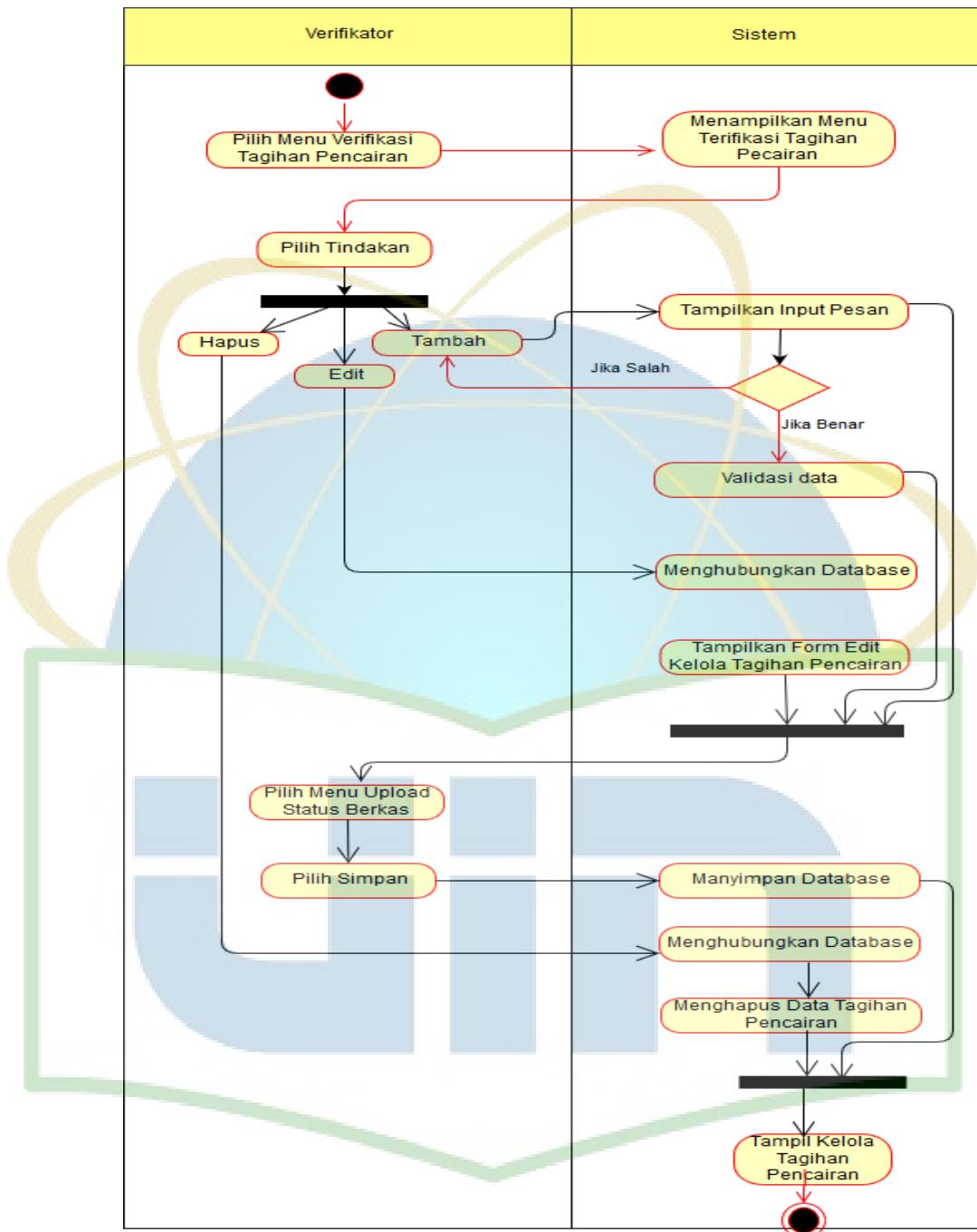
## 7. Activity Diagram Status Berkas



Gambar 4 12 Status Berkas

*Status Berkas* ini adalah aktifitas dimana aktor-aktor yang terdapat di user dapat mengupload berkas dan merubah setatus berkas unit yang dilakukan oleh masing-masing bagian di Biro Hukum dan KLN. Halaman yang ditampilkan pun juga hanya halaman *Status Berkas*, setiap user ada. Dan dapat dilihat untuk aktor-aktor yang terdaftar diuser.

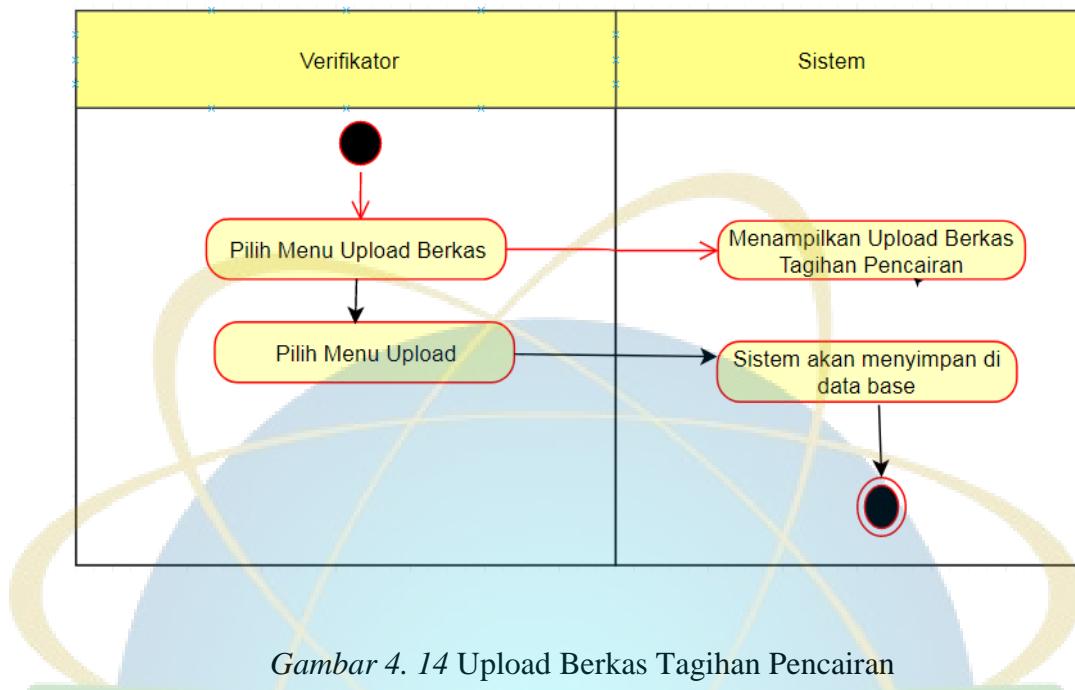
### 8. Activity Diagram Validasi Pengajuan Kegiatan



Gambar 4. 13 Activity Diagram Validasi pengajuan kegiatan

*Validasi pengajuan kegiatan* adalah aktifitas dimana aktor dapat mengelola kelola *Tagihan Pencairan* yang akan menggunakan sistem ini. Pengelolaan Tagihan diantaranya menambahkan Verifikasi data, dan hapus *Tagihan*. Aktor yang dapat melakukan aktifitas ini adalah Verifikator, dimana aktor tersebut membuka sistem informasi monitoring anggaran keuangan. kemudian memilih menu *Validasi Tagihan Pencairan*, kemudian sistem akan menampilkan halaman Input Kegiatan yang berisi *list validasi Tagihan Pencairan* yang sudah ada beserta tindakan pada pengelolaan, yaitu tambah, *edit* dan hapus. Jika aktor ingin membuat pesan *Tagihan Pencairan*, aktor memilih tambah pesan, kemudian sistem akan menampilkan *form tambah pesan*, lia ada kesalahan berkas maka aktor akan mengirim pesan, dan bila berkas tidak ada kesalahan maka aktor memvalidasi data. Setelah aktor mengisi *form* tersebut kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data *kelola Tagihan Pencairan*, aktor klik *edit* pada data *Validasi Tagihan Pencairan* yang ada kemudian sistem akan menampilkan *form Kelola Tagihan Pencairan*, dan aktor pilih simpan maka data *kegiatan* yang telah diperbarui akan tersimpan di *database*. Kemudian jika aktor ingin menghapus data *Validasi Tagihan pencairan*, aktor tinggal memilih data user yang akan dihapus kemudian klik hapus, maka sistem akan menghubungkan ke *database* dan data *kegiatan* akan terhapus. Kemudian sistem akan menampilkan *list* data tagihan pencairan.

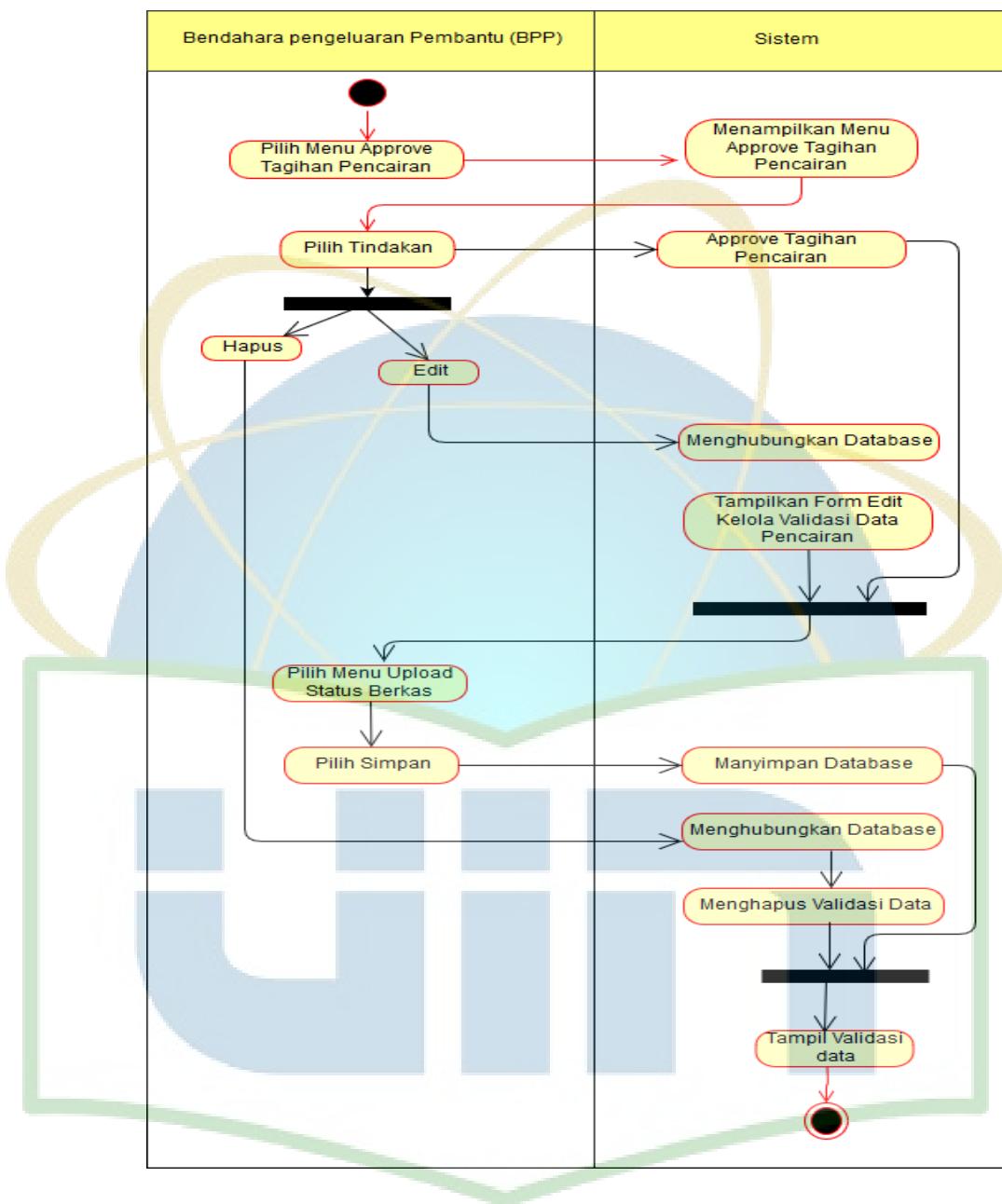
### 9. Activity Diagram Approve Upload Berkas Tagihan Pencairan



Gambar 4. 14 Upload Berkas Tagihan Pencairan

*Upload Berkas Tagihan Pencairan* ini adalah aktifitas dimana aktor yang terdapat di user dapat mengupload berkas dan merubah setatus berkas unit yang dilakukan oleh masing-masing bagian di Biro Hukum dan KLN. Halaman yang ditampilkan pun juga hanya halaman *Upload Berkas Tagihan Pencairan* setiap user ada. Dan tidak dapat digunakan oleh semua user hanya Verifikator saja yang dapat mengakses menu *Upload Berkas Tagihan Pencairan*.

### 10. Activity Diagram Approve Tagihan Pencairan

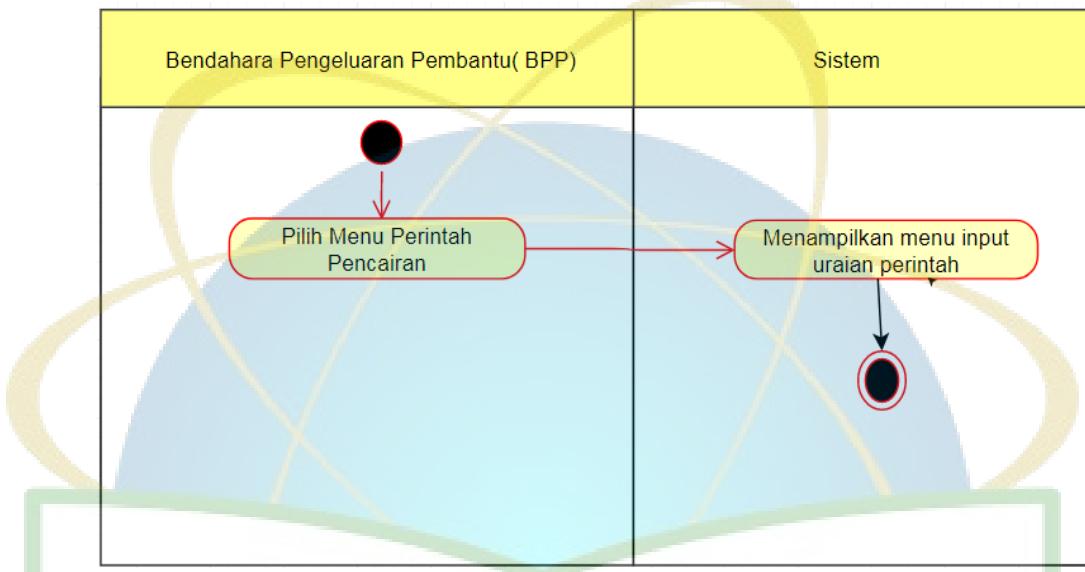


Gambar 4. 15 Activity Diagram Approve Tagihan Pencairan

*Approve Tagihan Pencairan* adalah aktifitas dimana aktor dapat mengelola Mengaprove *Tagihan Pencairan* yang akan menggunakan sistem ini. Pengelolaan Tagihan diantaranya menambahkan Verifikasi data, dan hapus *validasi data*. Aktor yang dapat melakukan aktifitas ini adalah Bendahara Pengeluaran Pembantu (BPP), dimana aktor tersebut membuka sistem informasi monitoring anggaran keuangan. kemudian memilih menu *Approve Tagihan Pencairan*, kemudian sistem akan menampilkan halaman Input Kegiatan yang berisi *list Approve Tagihan Pencairan* yang sudah ada beserta tindakan pada pengelolaan, yaitu tambah, *edit* dan hapus. Jika aktor ingin membuat pesan *Approve Tagihan Pencairan*, aktor memilih tambah pesan, kemudian sistem akan menampilkan *form tambah pesan*, lia ada kesalahan berkas maka aktor akan mengirim pesan, dan bila berkas tidak ada kesalahan maka aktor memvalidasi data. Setelah aktor mengisi *form* tersebut kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data *Validasi Tagihan Pencairan*, aktor klik *edit* pada data *Approve Tagihan Pencairan* yang ada kemudian sistem akan menampilkan *form Approve Tagihan Pencairan*, dan aktor pilih simpan maka data *kegiatan* yang telah diperbarui akan tersimpan di *database*. Kemudian jika aktor ingin menghapus data *Approve Tagihan Pencairan*, aktor tinggal memilih data user yang akan dihapus kemudian klik hapus, maka sistem akan menghubungkan ke *database* dan data *pencairan* akan

terhapus. Kemudian sistem akan menampilkan *list* data Validasi data pencairan.

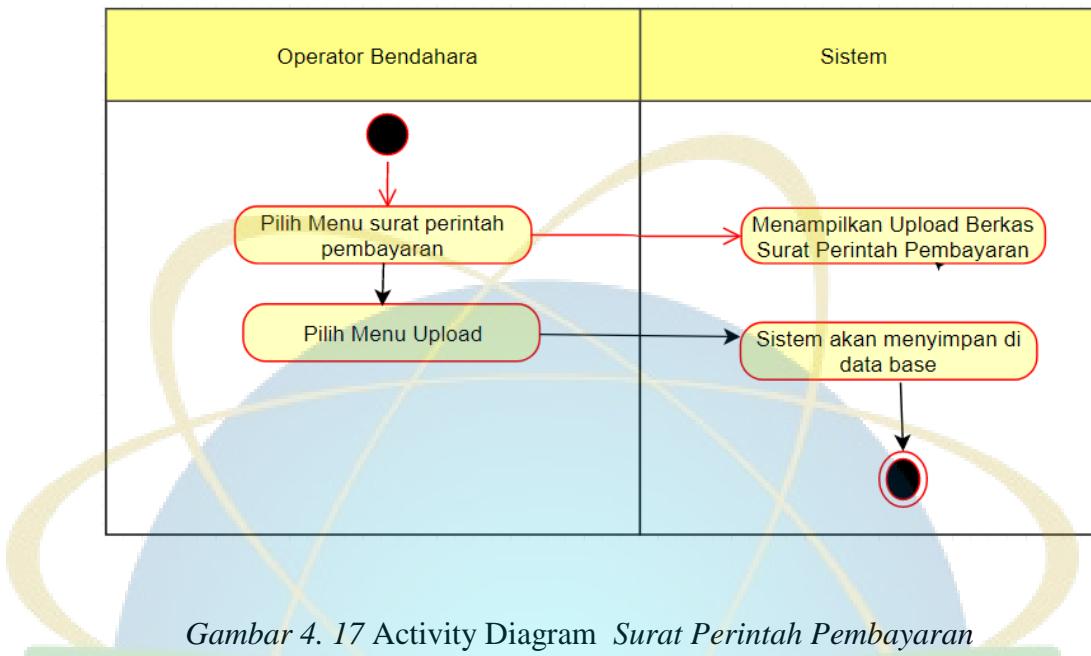
### 11. Activity Diagram Surat Perintah Pencairan



Gambar 4. 16 Activity Diagram *Input Perintah Pencairan*

*Input Perintah Pencairan* ini adalah aktifitas dimana aktor yang terdapat di user dapat melakukan perintah dengan menginput uraian perintah. Halaman yang ditampilkan pun juga hanya halaman *Input Perintah Pencairan* yang berada dihalaman Bendahara Pengeluaran Pembantu (BPP).

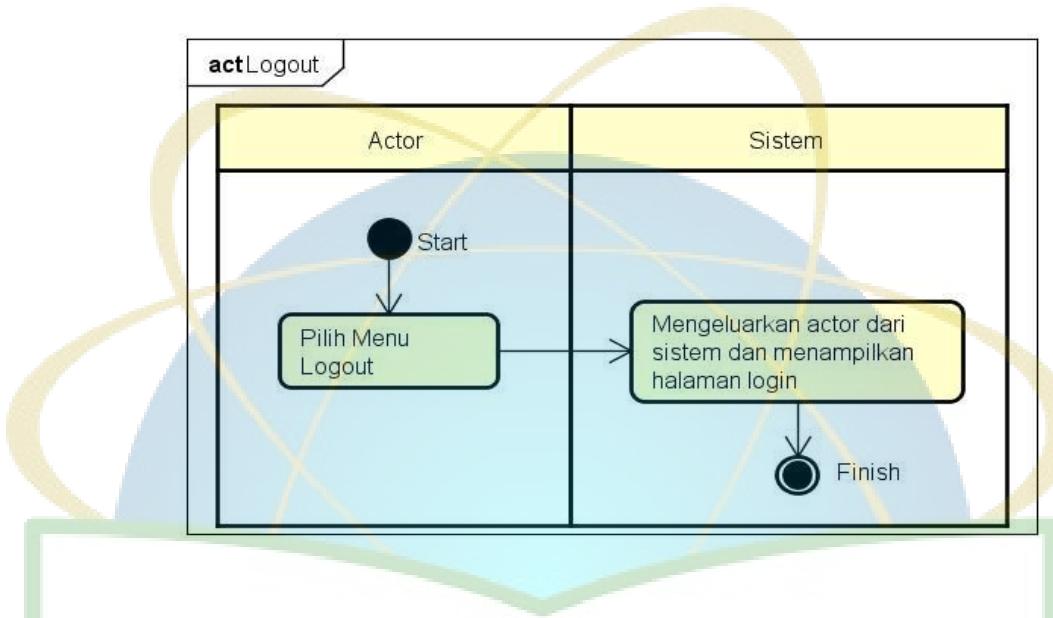
## 12. Activity Diagram Surat Perintah Pembayaran



Gambar 4. 17 Activity Diagram *Surat Perintah Pembayaran*

*Surat Perintah Pembayaran* ini adalah aktivitas dimana aktor yang terdapat di user dapat mengupload berkas dan merubah setatus berkas unit yang dilakukan oleh masing-masing bagian di Biro Hukum dan KLN. Halaman yang ditampilkan pun juga hanya halaman *Surat Perintah Pencairan* setiap user ada. Dan tidak dapat digunakan oleh semua user hanya Operator Bendahara saja yang dapat mengakses menu Upload Berkas Tagihan Pencarian.

### 13. Activity Diagram logout

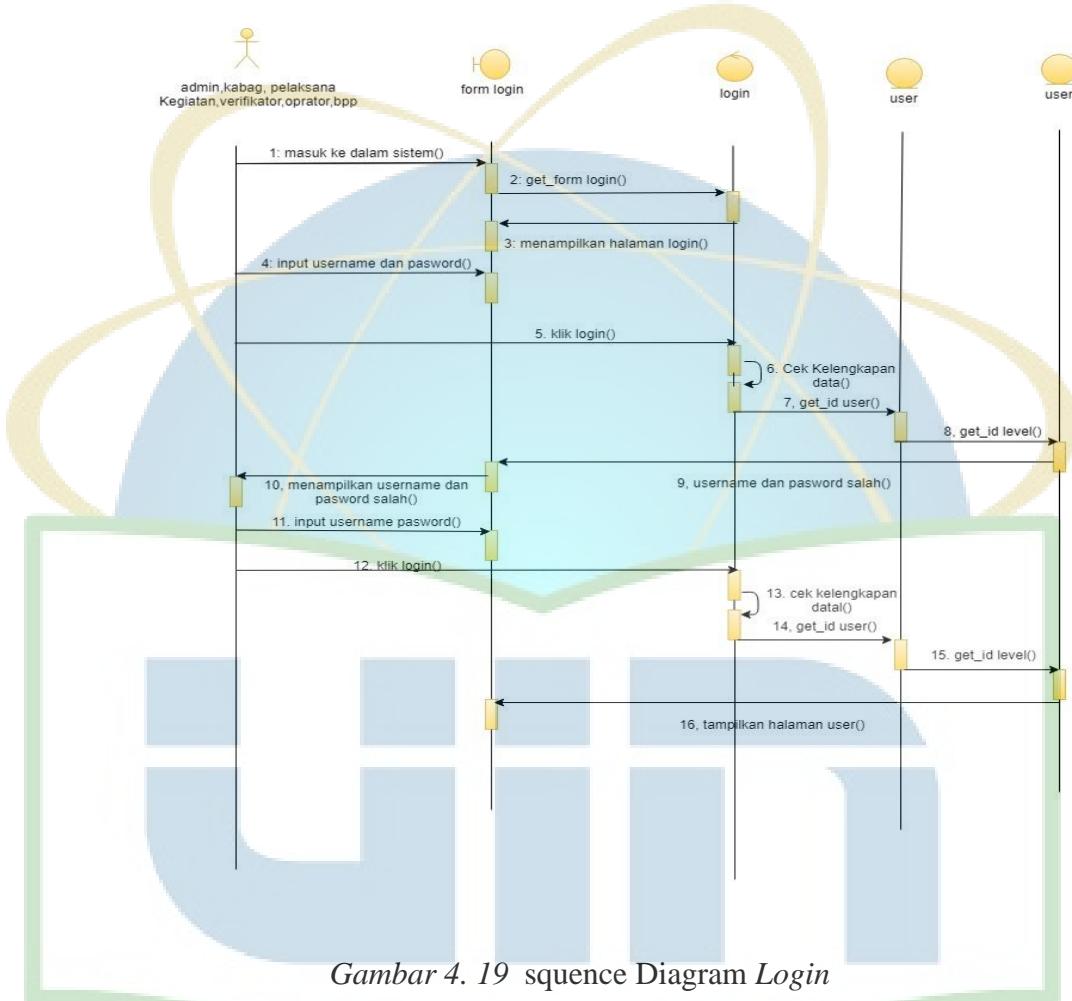


Gambar 4. 18 Activity Diagram Logout

Aktivitas yang terjadi pada gambar 4.19 adalah aktivitas yang dilakukan oleh semua aktor untuk dapat keluar dari sistem. Para actor memilih menu logout, kemudian sistem akan mengeluarkan masing-masing aktor dari sistem dan menampilkan kembali ke halaman *login*.

#### 4.1.2.1. Perancangan Sequence diagram

##### 1. Sequence Diagram Login

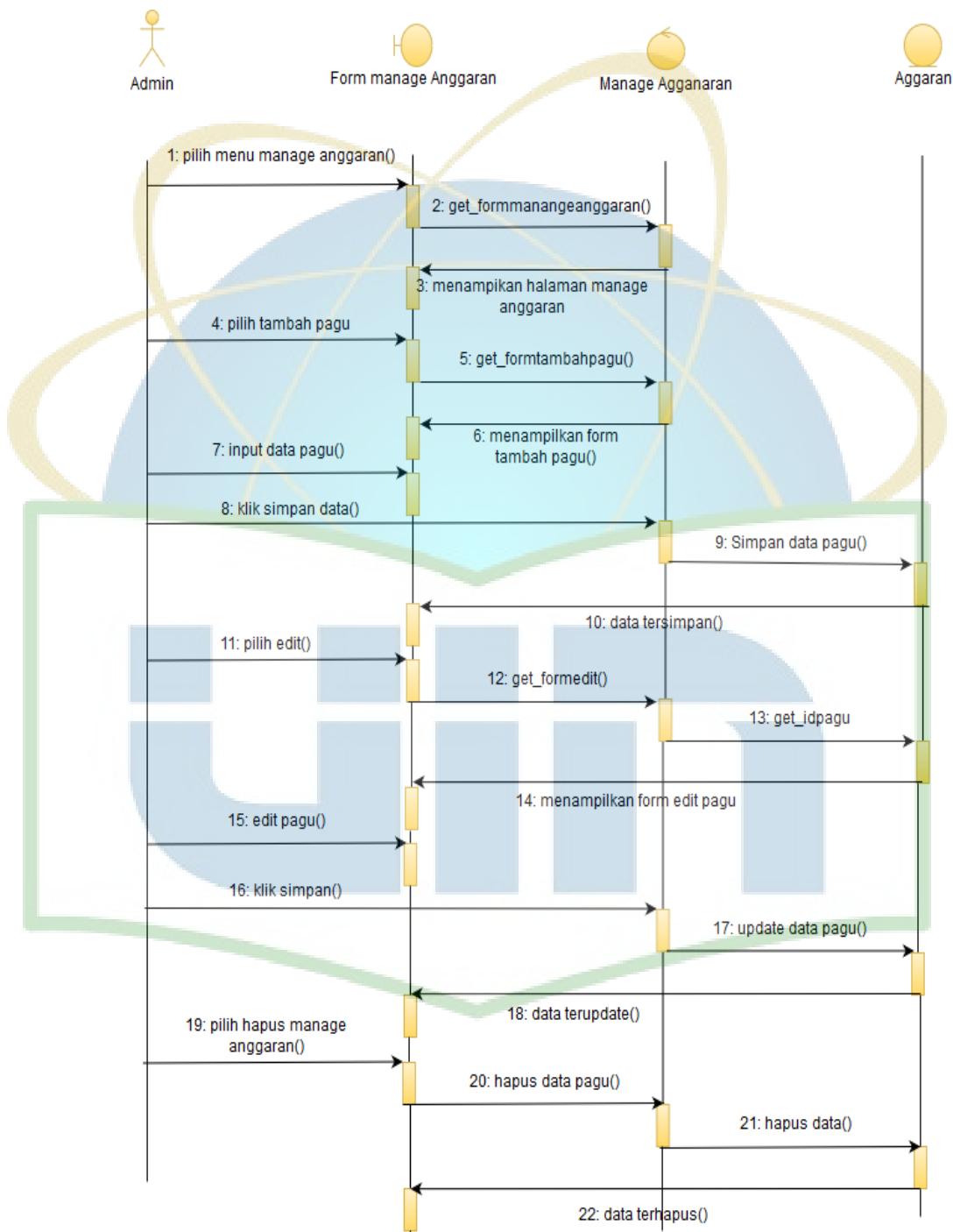


Gambar 4. 19 squence Diagram Login

Sequence diagram login bisa dilakukan oleh semua aktor yang ingin masuk ke dalam sistem. Awalnya aktor membuka sistem dan system akan menampilkan halaman login ke pada aktor, kemudian aktor memasukkan username & password dan klik login maka sistem akan melakukan verifikasi kebenaran data yang dimasukkan, kemudian mengecek apakah id level yang ada pada tabel user sesuai tidak dengan username & password

yang dimasukkan, jika benar maka aktor akan masuk ke halaman sistem informasi monitoring dan evaluasi sesuai dengan levelnya masing-masing.

## 2. Sequence Diagram Manage Anggaran

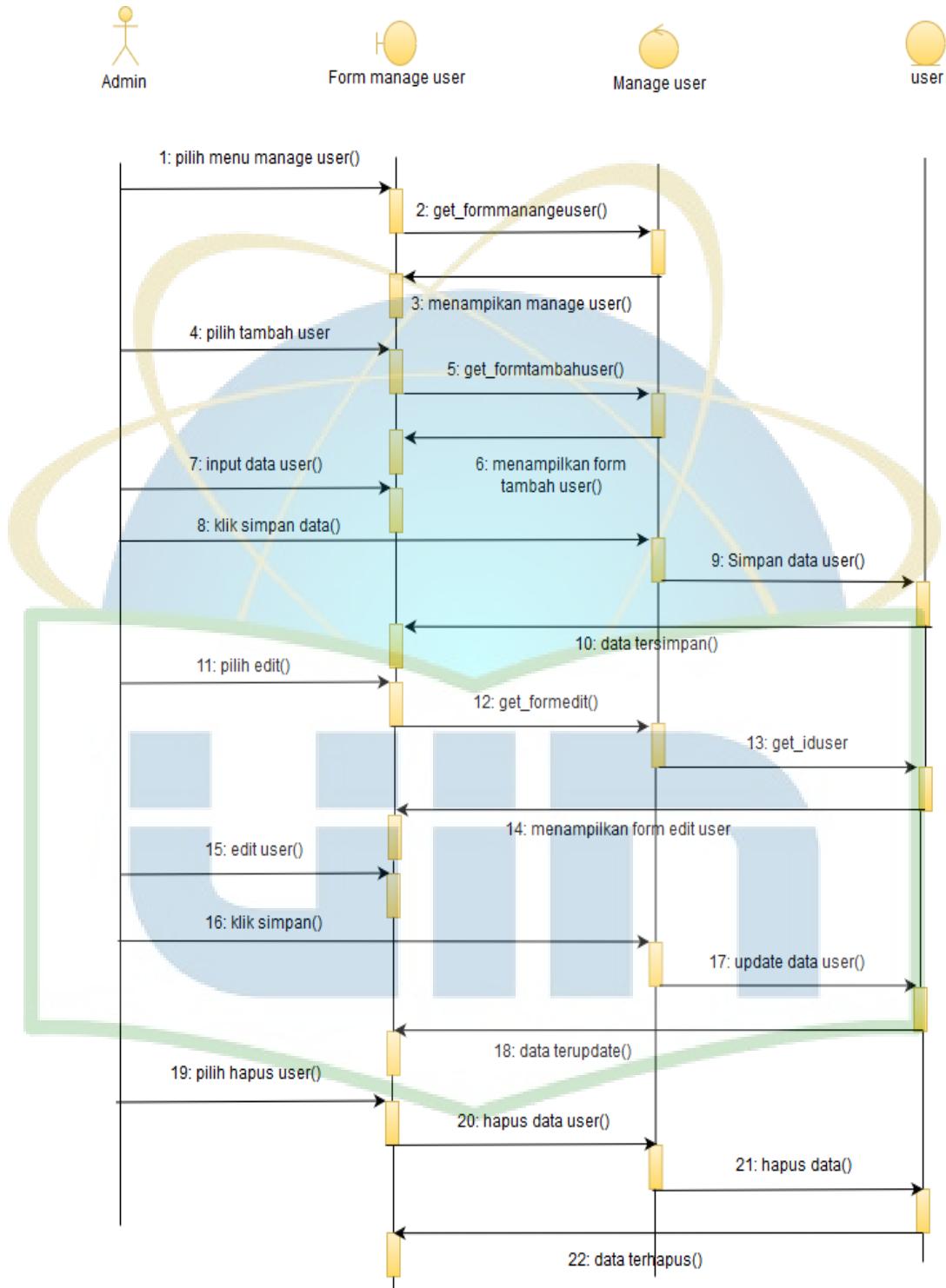


Gambar 4. 20 Sequence Diagram Manage Anggaran

Sequence manage anggaran dilakukan oleh Admin, dimana admin memilih menu manage anggaran kemudian sistem menampilkan halaman manage anggaran. Disini admin dapat melakukan tambah tambah, edit, dan hapus anggaran. Jika aktor melakukan tambah pagu baru, maka sistem akan menampilkan halaman tambah pagu kemudian aktor melakukan input dan klik simpan maka sistem menyimpannya dalam database. Kemudia jika melakukan edit, sistem akan menghubungkan database kemudian menampilkan halaman edit pagu. Dan setalah aktor edit pagu dan klik simpan maka data yang telah dirubah tersimpan di database. Begitu pula jika aktor menghapus data pagu, maka sistem akan menghapusnya dari database



### 3. Sequence Diagram Manage User

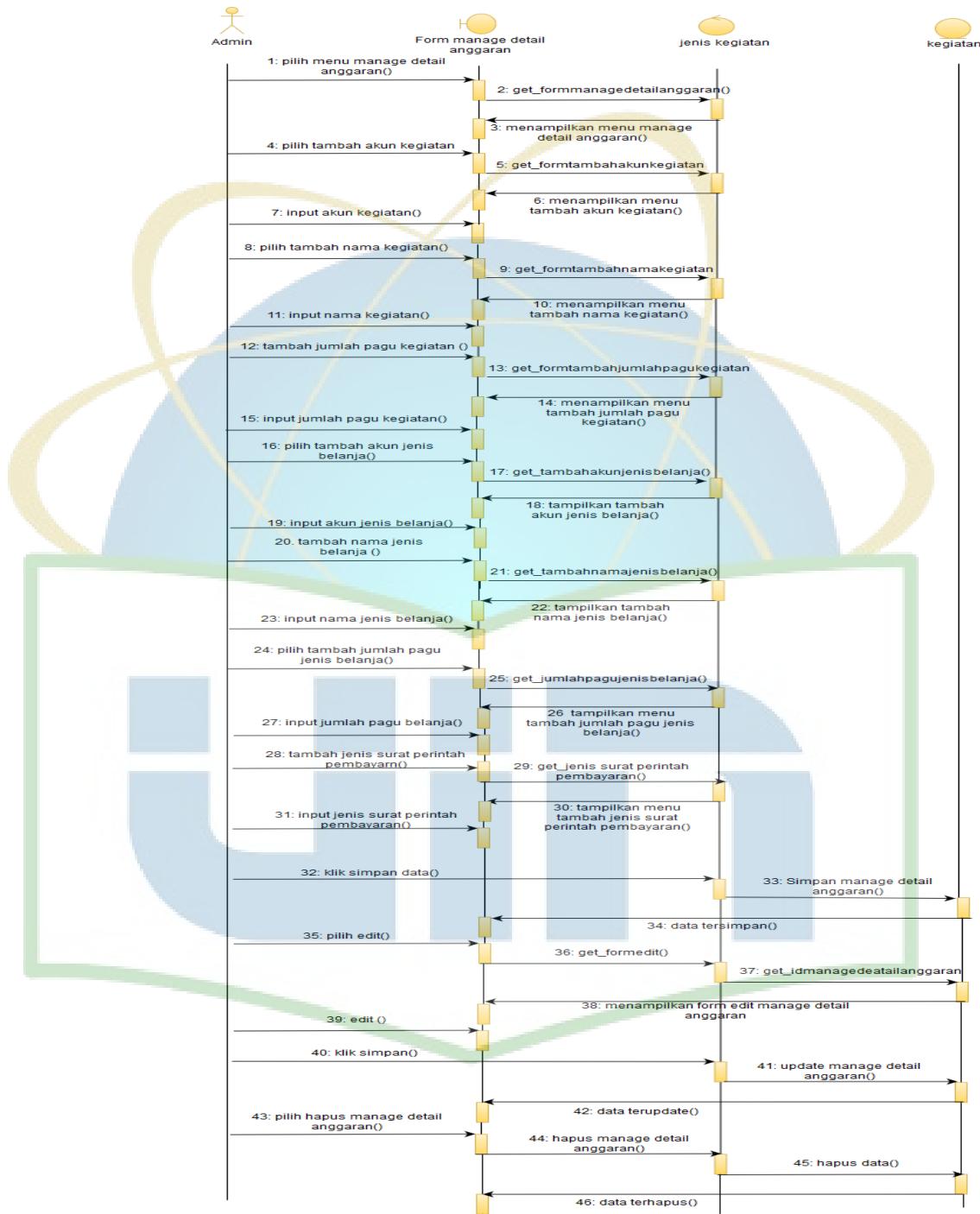


Gambar 4. 21 Sequence Diagram Manage User

Sequence jenis kegiatan dilakukan oleh Admin, dimana admin memilih menu manage user kemudian sistem menampilkan halaman manage user. Disini admin dapat melakukan tambah tambah, edit, dan hapus user. Jika admin melakukan tambah nama user, maka sistem akan menampilkan halaman tambah user kemudian admin melakukan tambah admin melakukan input dan klik simpan maka sistem menyimpannya dalam database. Kemudia jika melakukan edit, sistem akan menghubungkan database kemudian menampilkan halaman user kegiatan. klik simpan maka data yang telah dirubah tersimpan di database. Begitu pula jika admin menghapus jenis kegiatan, maka sistem akan mengahpusnya dari database



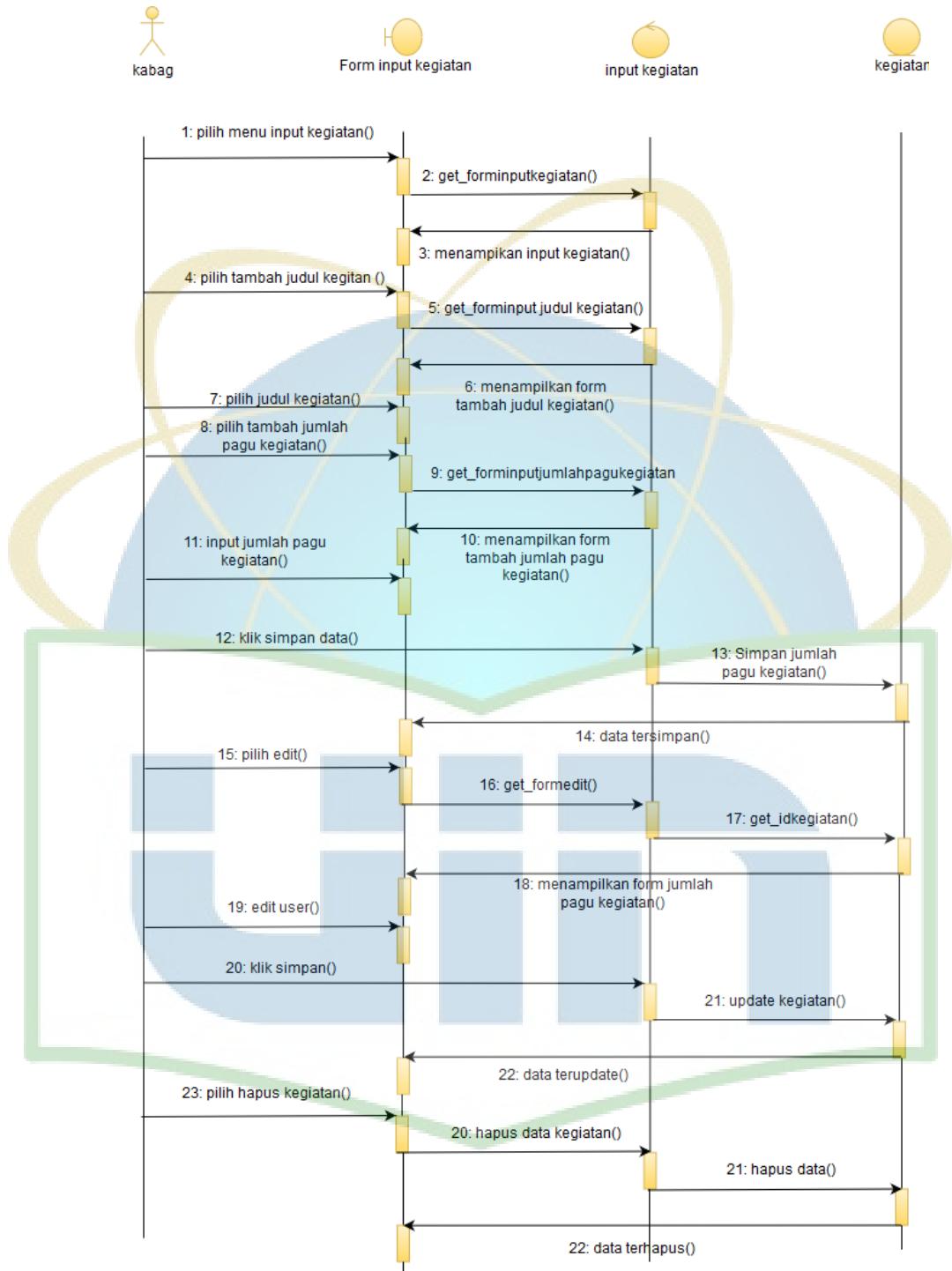
#### 4. Sequence Diagram Manage Detail Anggaran



Gamabar 4. 22 Squence Diagram Manage Detail Anggaran

Sequence Manage Detail Anggaran dilakukan oleh Admin, dimana admin memilih menu manage detail anggaran kemudian sistem menampilkan halaman manage deatai anggaran. Disini admin dapat melakukan tambah tambah akun kegiatan, nama kegiatan, pagu kegiatan, akun belanja, nama belanja , pagu belanja. Admin dapat edit, dan hapus data manage deatai kegiatan. Jika admin melakukan tambah manage detail kegiatan, maka sistem akan menampilkan halaman tambah manage detail kegiatan kemudian admin melakukan tambah manage detai kegiatan maka sistem menampilkan halaman tambah manage detai anggaran admin melakukan input dan klik simpan maka sistem menyimpannya dalam database. Kemudia jika melakukan edit, sistem akan menghubungkan database kemudian menampilkan halaman manage detail anggaran. Dan setalah aktor manage detai jenis kegiatan dan klik simpan maka data yang telah dirubah tersimpan di database. Begitu pula jika aktor menghapus jenis kegiatan, maka sistem akan mengahpusnya dari database.

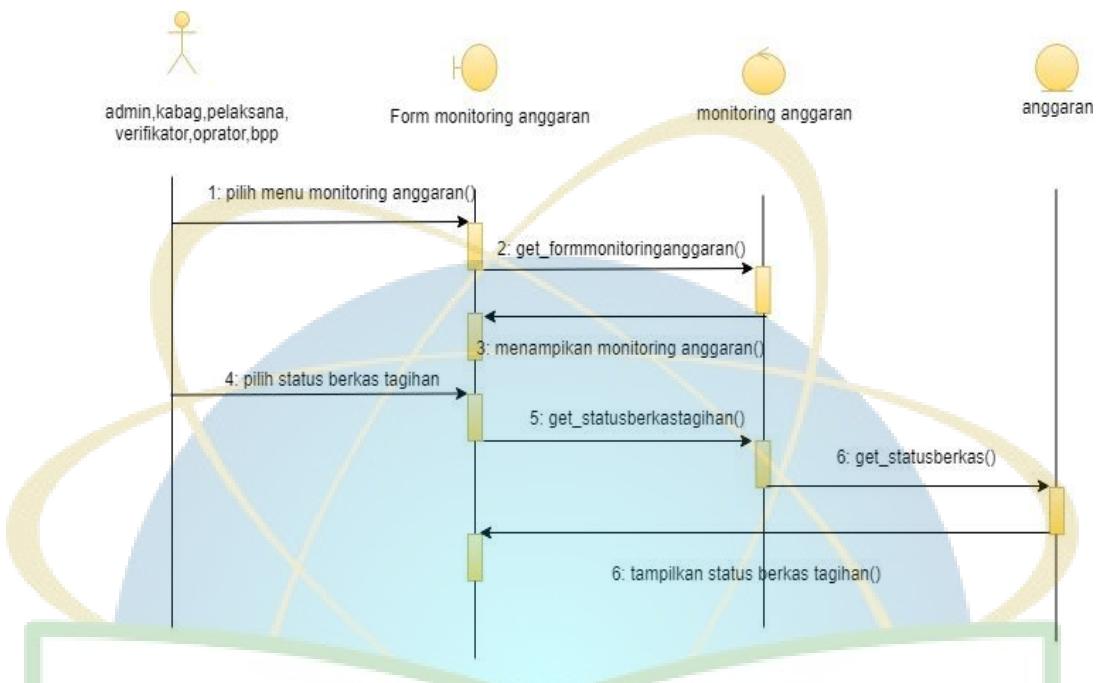
## 5. Sequence Diagram Pengajuan Kegiatan



Gamabar 4 23 Squence Diagram Pengajuan Kegiatan

Sequence kegiatan dilakukan oleh kabag., dimana kabag-kabag memilih menu input kegiatan kemudian sistem menampilkan halaman input kegiatan. Kabag-kabag memilih menu tambah judul kegiatan kemudian sistem menampilkan halaman input judul kegiatan. Kabag-kabag memilih menu tambah pagu kegiatan kemudian system menampilkan halaman tambah pagu kegiatan Disini kabag dapat melakukan tambah judul kegiatan dan jumlah pagu kegiatan, edit, dan hapus judul kegiatan dan pagu kegiatan. Jika kabag melakukan tambah baru, maka sistem akan menampilkan halaman tambah judul kegiatan dan pagu kegiatan kemudian kabag melakukan input dan klik simpan maka sistem menyimpannya dalam database. Kemudia jika melakukan edit, sistem akan menghubungkan database kemudian menampilkan halaman input kegiatan. Dan setalah admin edit judul kegiatan dan pagu kegiatan dan klik simpan maka data yang telah dirubah tersimpan di database. Begitu pula jika kabag menghapus data kegiatan, maka sistem akan mengahapusnya dari database

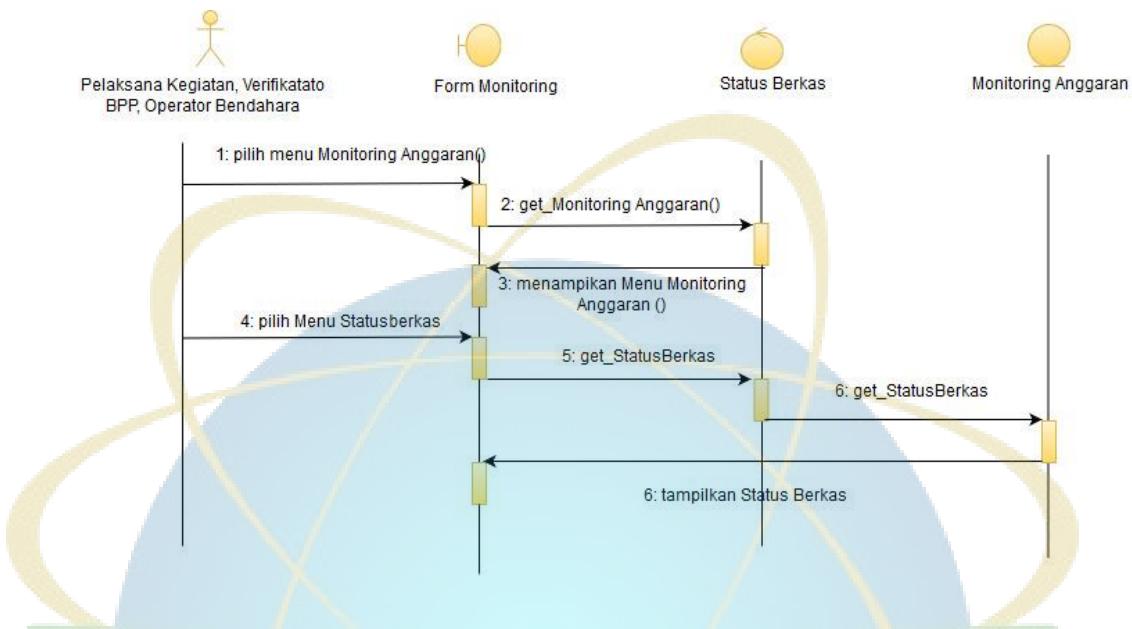
## 6. Sequence Diagram Monitoring Anggaran



Gambar 4. 24 Sequence Diagram Monitoring Anggaran

Sequence Tagihan Pencairan dilakukan oleh Admin, Kabag, pelaksana kegiatan, verifikator, operator, dan BPP dimana para aktor memilih menu monitoring anggaran sistem menampilkan halaman monitoring anggaran. Lalu Sistem Akan menampilkan table realisasi pagu anggaran.

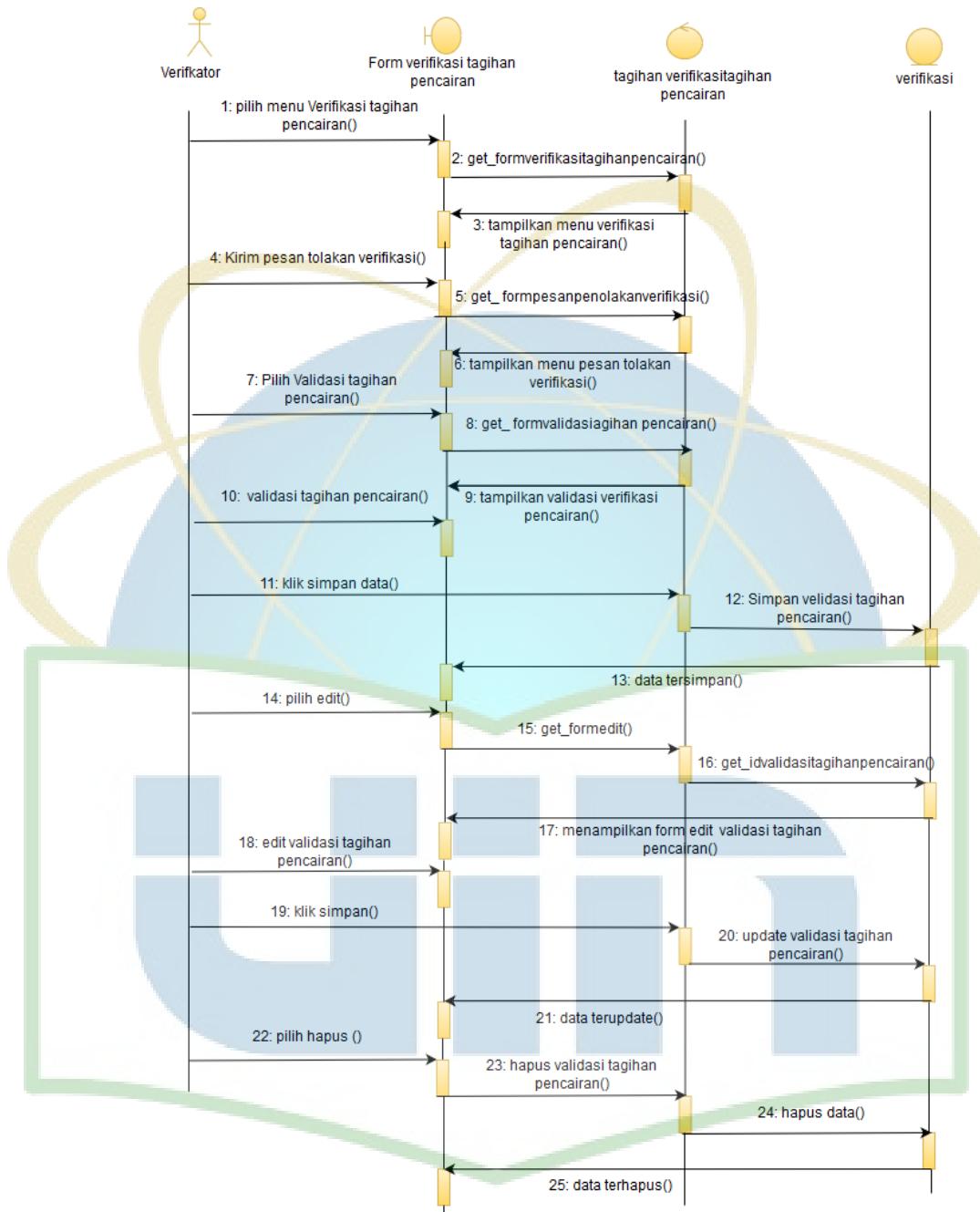
## 7. Sequence Diangaram Status Berkas



Gambar 4. 25 Sequence Diagram Status Berkas

Sequence Status Berkas dilakukan oleh, Kabag, pelaksana kegiatan, verifikatoro, operator, dan BPP dimana para aktor memilih menu monitoring anggaran sistem menampilkan halaman monitoring anggaran. Para actor memilih menu status berkas tanggahan kemudian sistem menampilkan halaman status berkass tagihan.

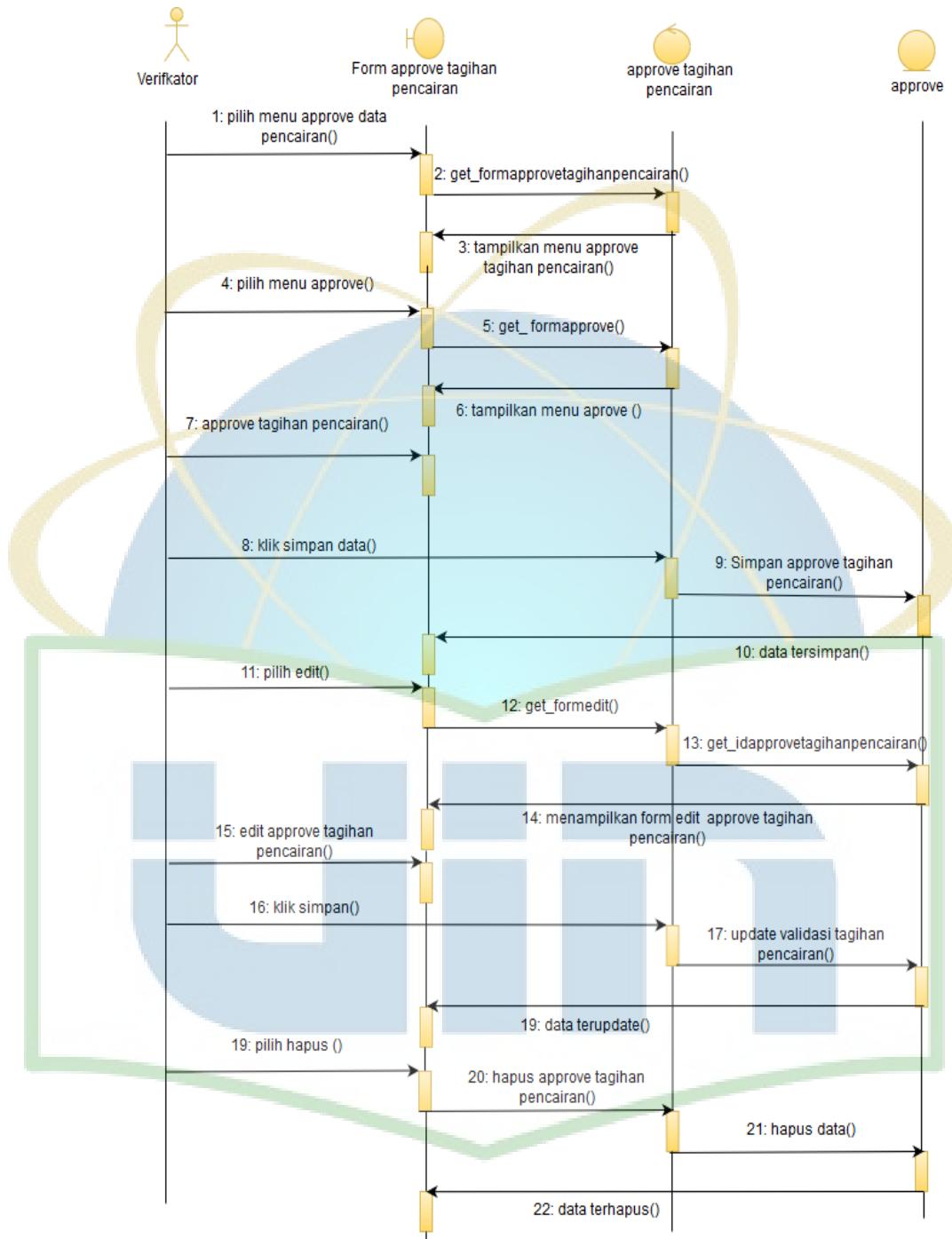
## 8. Sequence Diagram Validasi Pengajuan Kegiatan



Gambar 4. 26 Sequence Diagram Validasi Pengajuan Kegiatan

*Validasi Pengajuan Kegiatan* adalah aktifitas dimana aktor dapat mengelola kelola *Validasi Pengajuan Kegiatan* yang akan menggunakan sistem ini. Aktor menambahkan Verifikasi data, dan hapus *Tagihan*. Aktor yang dapat melakukan aktifitas ini adalah Verifikator, dimana aktor tersebut membuka sistem informasi monitoring anggaran keuangan. kemudian memilih menu *Validasi Pengajuan Kegiatan*, kemudian sistem akan menampilkan halaman Input Kegiatan yang berisi *list validasi Pengajuan Kegiatan* yang sudah ada beserta tindakan pada pengelolaan, yaitu tambah, *edit* dan hapus. Jika aktor ingin membuat pesan, aktor memilih tambah pesan, kemudian sistem akan menampilkan *form tambah pesan*, lia ada kesalahan berkas maka aktor akan mengirimpan pesan, dan bila berkas tidak ada kesalahan maka aktor memvalidasi data. Setelah aktor mengisi *form* tersebut kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data *kelola Tagihan Pencairan*, aktor klik *edit* pada data *Validasi Pengajuan Kegiatan* yang ada kemudian sistem akan menampilkan *form Validasi Pengajuan Kegiatan*, dan aktor pilih simpan maka data *kegiatan* yang telah diperbarui akan tersimpan di *database*. Kemudian jika aktor ingin menghapus data *Validasi Pengajuan Kegiatan* aktor tinggal memilih data user yang akan dihapus kemudian klik hapus, maka sistem akan menghubungkan ke *database* dan data *kegiatan* akan terhapus. Kemudian sistem akan menampilkan *list* data tagihan pencairan.

### 9. Sequence Diagram Upload Berkas Pencairan

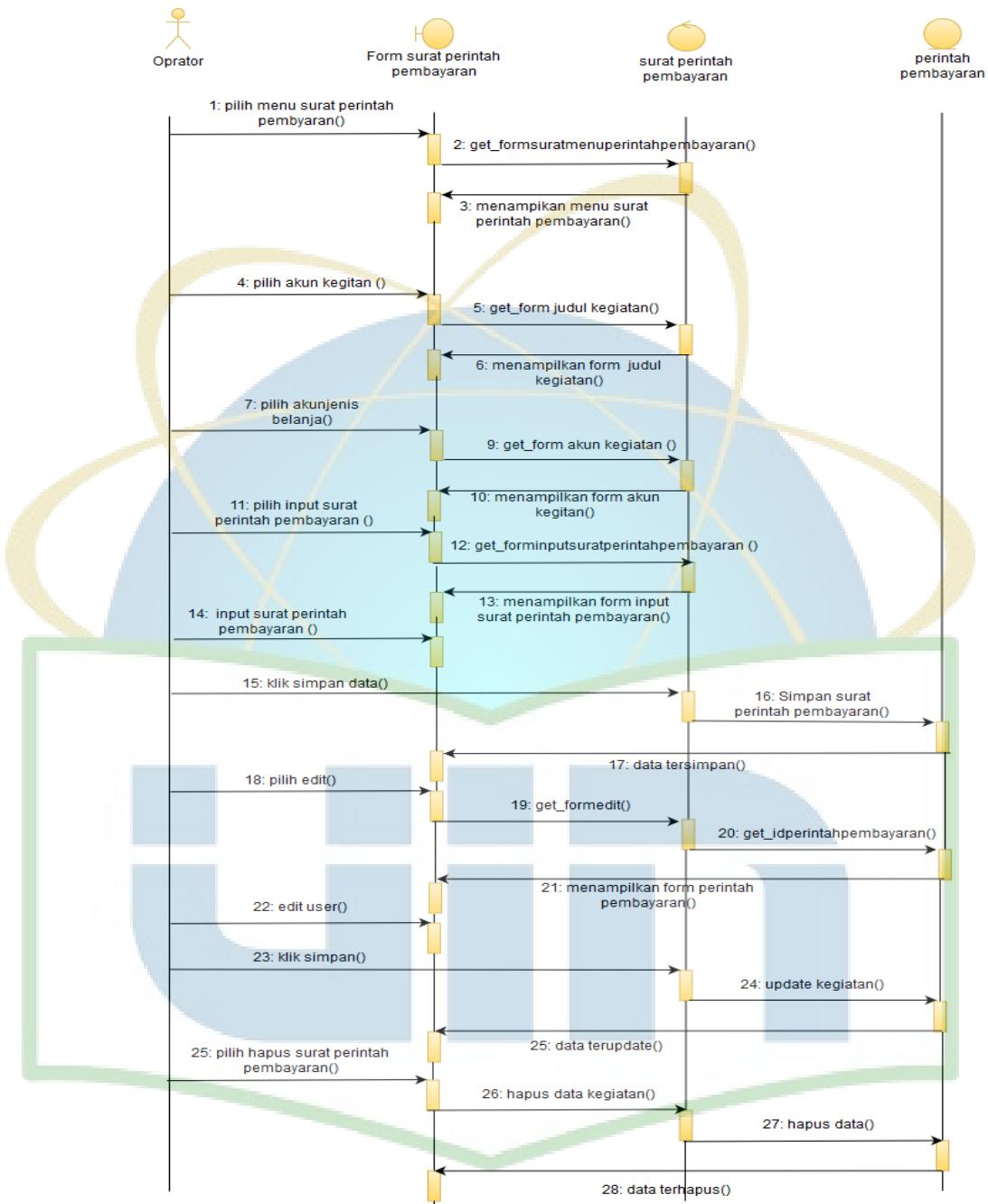


Gambar 4. 27 Sequence Diagram Upload Berkas Pencairan

*Upload Berkas Tagihan Pencairan* adalah aktifitas dimana aktor dapat mengelola Upload Berkas *Upload Tagihan Pencairan* yang akan menggunakan sistem ini. Pengelolaan Tagihan diantaranya menambahkan Verifikasi data, dan hapus *validasi data*. Aktor yang dapat melakukan aktifitas ini adalah Verifikator, dimana aktor tersebut membuka sistem Tagihan Pencairan. kemudian memilih menu Upload Tagihan Pencairan , kemudian sistem akan menampilkan halaman Upload Tagihan Pencairan, yaitu tambah, *edit* dan hapus. Jika aktor ingin membuat pesan , aktor memilih tambah pesan, kemudian sistem akan menampilkan *form tambah pesan*, lia ada kesalahan berkas maka aktor akan mengirim pesan, dan bila berkas tidak ada kesalahan maka aktor memvalidasi data. Setelah aktor mengisi *form* tersebut kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data Tagihan Pencairan aktor klik *edit* pada data *Validasi Pengajuan Kegiatan* yang ada kemudian sistem akan menampilkan *form Validasi Pengajuan Kegiatan* dan aktor pilih simpan maka data *kegiatan* yang telah diperbarui akan tersimpan di *database*.

Kemudian jika aktor ingin menghapus data *Validasi Tagihan Pencairan*, aktor tinggal memilih data user yang akan dihapus kemudian klik hapus, maka sistem akan menghubungkan ke *database* dan data *pencairan* akan terhapus. Kemudian sistem akan menampilkan *list* data *Validasi data pencairan*

## 10. Sequence Diangaram Approve tagihan Pencairan

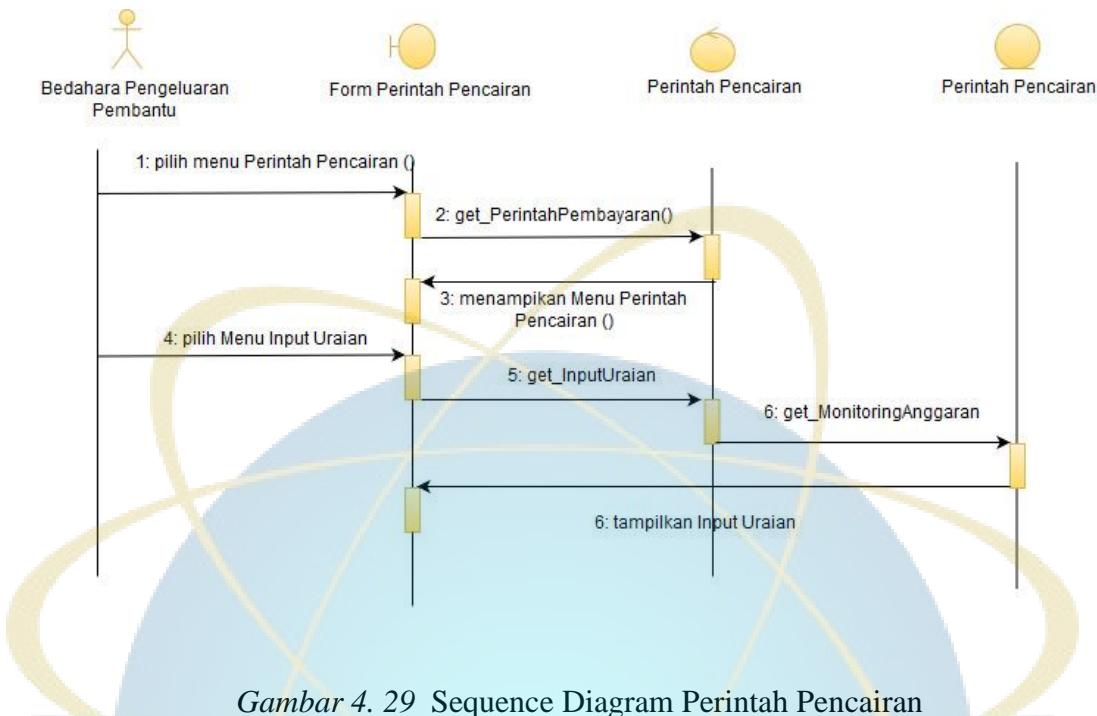


Gambar 4. 28 Sequence Diagram Approve Tagihan Pencairan

*Aprove Tagihan Pencairan* adalah aktifitas dimana aktor dapat mengelola sistem ini. *Approve Tagihan Pencairan* diantaranya menambahkan *Aprove Tagihan Pencairan*, dan *Tolak validasi data*. Aktor yang dapat melakukan aktifitas ini adalah Bendahara Pengeluaran pembantu, dimana aktor tersebut membuka sistem *Approve Tagihan Pencairan*. kemudian memilih menu *Approve Tagihan Pencairan* , kemudian sistem akan menampilkan halaman *Upload Tagihan Pencairan*, yaitu tambah, *edit* dan *hapus*. Jika aktor ingin membuat pesan , aktor memilih tambah pesan, kemudian sistem akan menampilkan *form tambah pesan*, lia ada kesalahan berkas maka aktor akan mengirimkan pesan, dan bila berkas tidak ada kesalahan maka aktor memvalidasi data. Setelah aktor mengisi *form* tersebut kemudian pilih simpan, dan data akan tersimpan pada *database*. Kemudian jika aktor ingin mengubah atau *edit* data *Tagihan Pencairan* aktor klik *edit* pada data *Approve Tagihan Pencairan* yang ada kemudian sistem akan menampilkan *form Aprove Tagihan Pncairan* dan aktor pilih simpan maka data *kegiatan* yang telah diperbarui akan tersimpan di *database*.

Kemudian jika aktor ingin menghapus data *Aprove Tagihan Pencairan*, aktor tinggal memilih data user yang akan dihapus kemudian klik *hapus*, maka sistem akan menghubungkan ke *database* dan data *pencairan* akan terhapus. Kemudian sistem akan menampilkan *list* data Validasi data pencairan

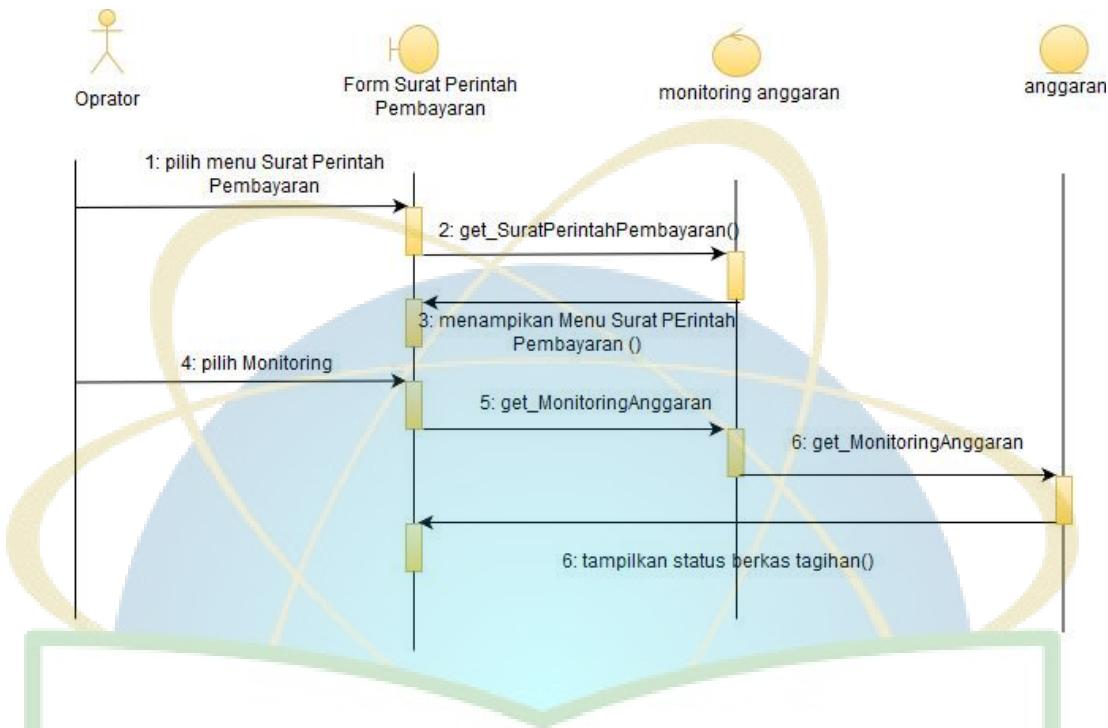
## 11. Sequence Diagram Perintah Pencairan



Gambar 4. 29 Sequence Diagram Perintah Pencairan

Aktivitas untuk Perintah Pencairan yang di lakukan oleh Bendahara Pengeluaran Pembantu. Pertama aktor memilih menu Perintah Pencairan sistem akan menampilkan halaman Perintah Pencairan, kemudian aktor memilih Input Uraian, setelah itu sistem akan menampilkan Input Uraian. Pilih Kirim Uraian maka akan tersimpan di database.

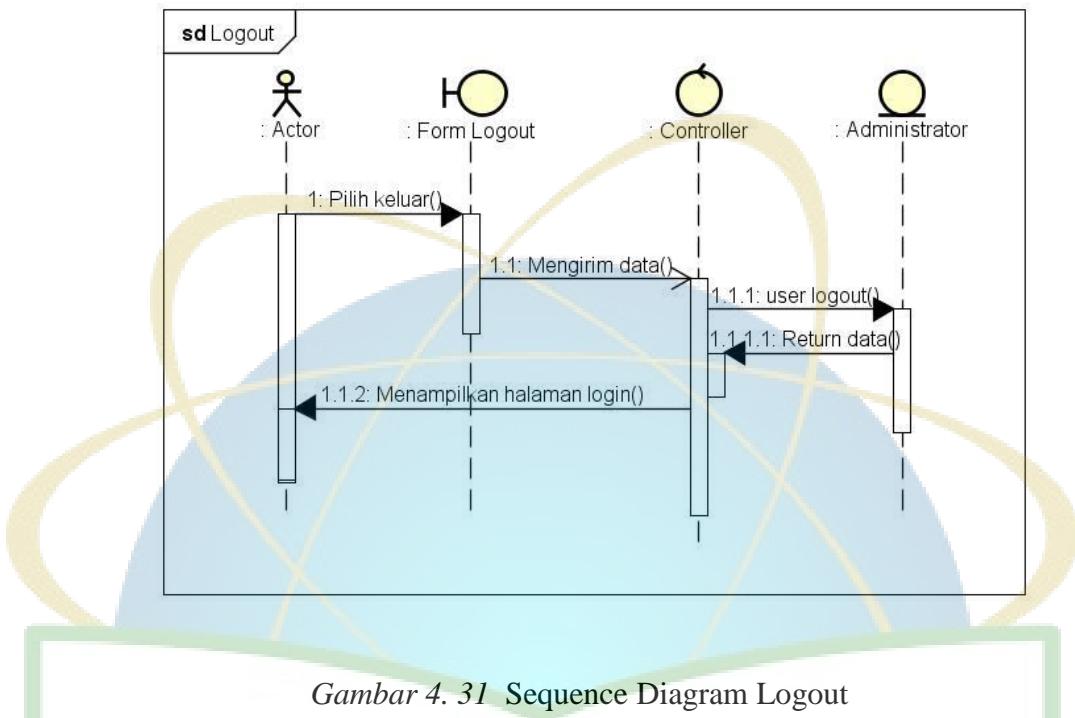
## 12. Sequence Diagram Surat Perintah Pembayaran



Gambar 4. 30 Sequence Diagram Surat Perintah Pembayaran

Aktivitas untuk meninput surat perintah pembayaran yang dilakukan oleh Operator. Pertama aktor memilih menu Surat Perintah Pembayaran sistem akan menampilkan halaman menu surat perintah pembayaran, kemudian aktor memilih Monitoring Anggaran, setelah itu sistem akan menampilkan Status Berkas.

### 13. Sequence Diagram Logout



Gambar 4. 31 Sequence Diagram Logout

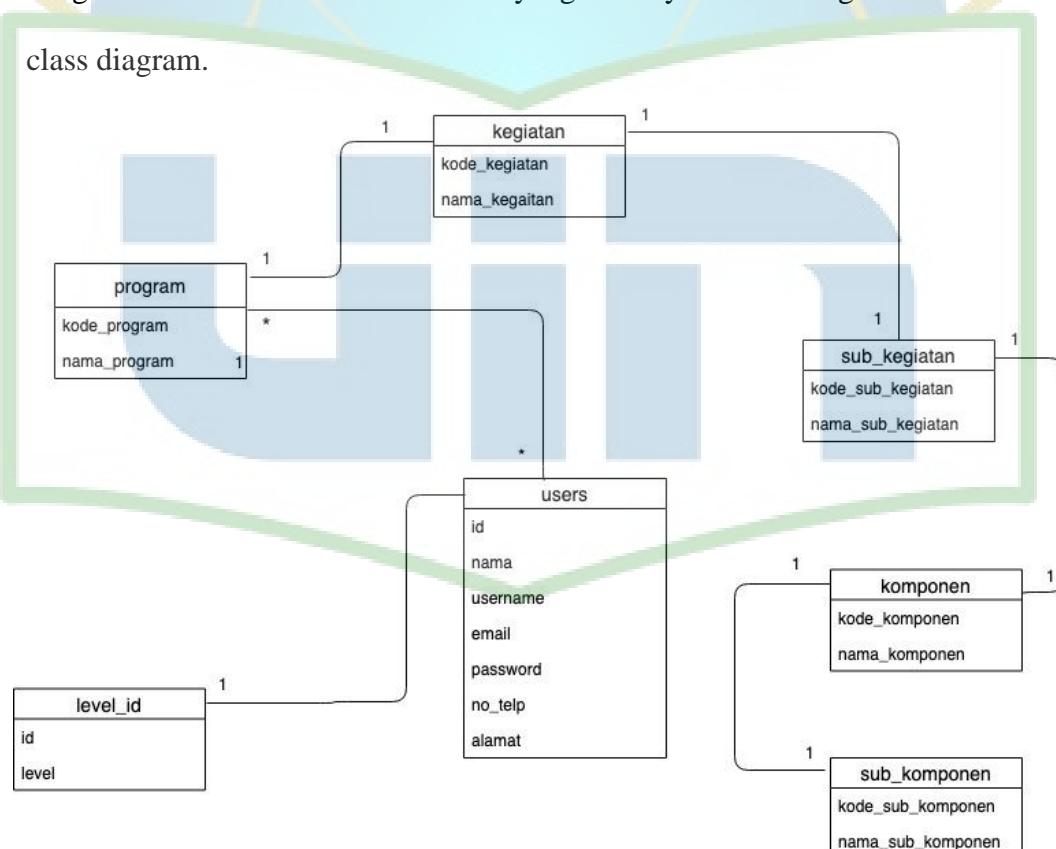
Sequence Diagram yang digambarkan ini adalah proses aktor yang dilakukan oleh admin untuk dapat keluar dari sistem. Actor memilih menu logout, kemudian sistem akan mengeluarkan aktor dari sistem dan menampilkan kembali khalaman *login*.

## 4.2.2. Perancangan Database

### 4.2.2.1. Class Diagram

Untuk membuat sebuah database sistem maka perlu ada rancangan database terlebih dahulu. Pada riset ini penulis menggunakan class diagram sebagai acuan dalam pembuatan *database* sistem.

Class diagram menggambarkan struktur dari sebuah sistem dengan menunjukkan kelas-kelas sistem, atributnya dan relasi antar kelas. Berikut adalah daftar objek potensial dari hasil identifikasi setiap kata benda yang berhubungan dengan keseluruhan bisnis atau event yang nantinya akan ditungkan dalam sebuah class diagram.

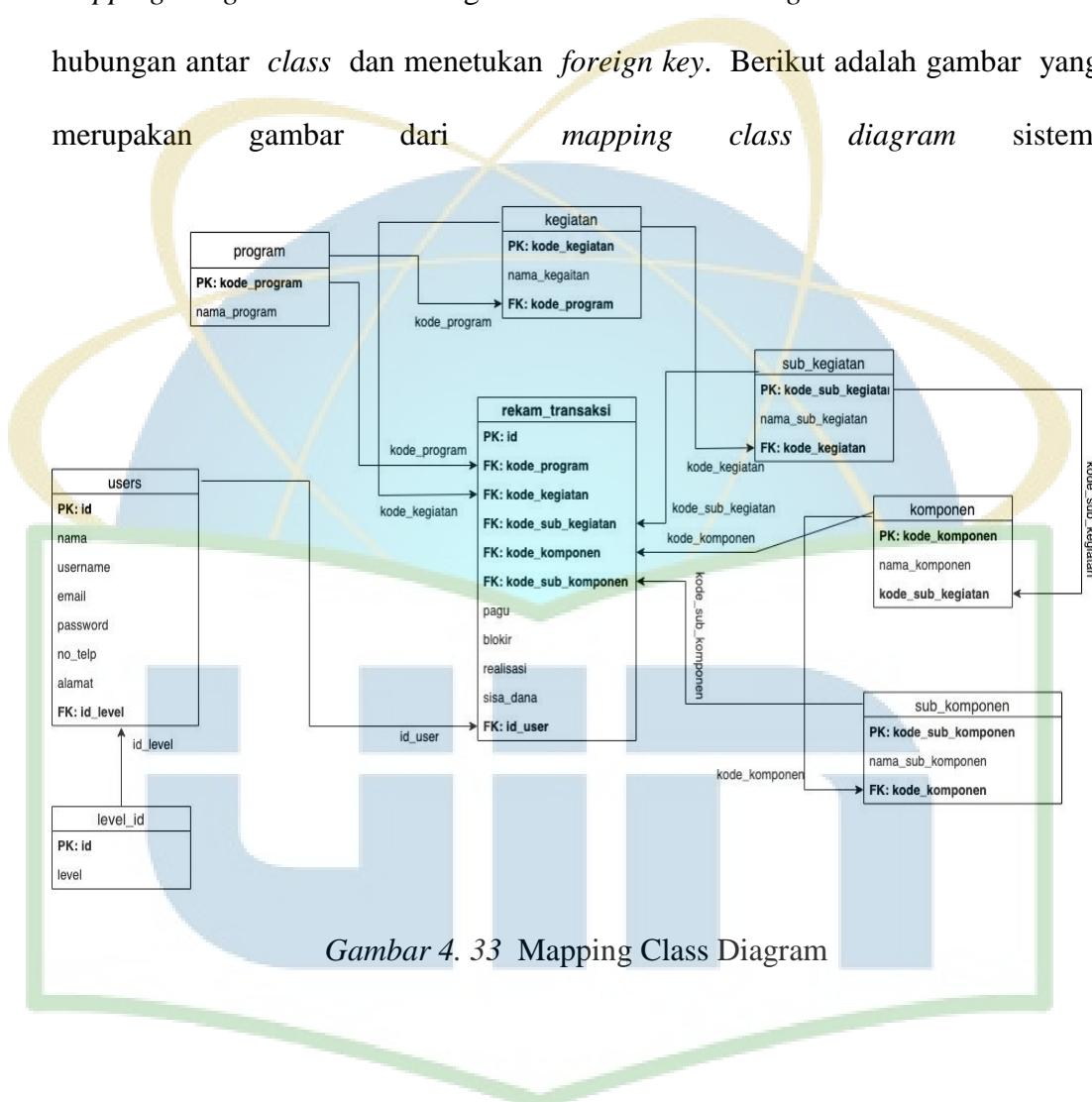


Gambar 4. 32 Class Diagram

#### 4.2.2.2. Mapping Class Diagram

Mapping Class Diagram digunakan untuk menjembatani antara class diagram ke format Relational Database Management System (RDBMS) tabel.

*Mapping Diagram* ini dirancang berdasarkan *class diagram* untuk menentukan hubungan antar *class* dan menetukan *foreign key*. Berikut adalah gambar yang merupakan gambar dari mapping class diagram sistem.



#### 4.2.3. Spesifikasi Database

##### 1. Tabel User

Nama Tabel : *User*  
 Tipe Tabel : Master  
*Primary Key* : *id\_user*  
*Foreign Key* : *id\_level*

*Tabel 4. 15 Tabel User*

Nama Field	Tipe Data	Ukuran	Keterangan
<i>id</i>	Int	11	Id dari user
<i>nama</i>	Varchar	30	Nama dari user
<i>username</i>	Varchar	30	<i>Username</i> dari user
<i>password</i>	Varchar	30	<i>Password</i> dari user
<i>remember_token</i>	Varchar	20	Token dari <i>user</i>
<i>no_telp</i>	Varchar	15	No telepon <i>user</i>
<i>email</i>	Varchar	30	Alamat email <i>user</i>
<i>level_id</i>	Int	11	Id dari level

##### 2. Tabel Level

Nama Tabel : *Level*  
 Tipe Tabel : Master  
*Primary Key* : *id\_level*

*Foreign*  
*Key*  
 : -

*Tabel 4. 16 Tabel Level*

Nama Field	Tipe Data	Ukuran	Keterangan
<i>id</i>	Int	11	Id dari level
<i>level</i>	Varchar	30	Tingkatan level

### 3. Tabel Kegiatan

Nama Tabel : Kegiatan  
 Tipe Tabel : Master  
*Primary Key* : kode\_kegiatan  
*Foreign Key* :

*Tabel 4. 17 Tabel Kegiatan*

Nama Field	Tipe Data	Ukuran	Keterangan
kode_kegiatan	Varchar	12	kode dari kegiatan
nama_kegiatan	Varchar	50	Nama kegiatan
kode_program	Varchar	12	Kode program dari kegiatan

### 4. Tabel Sub Kegiatan

Nama Tabel : Sub Kegiatan  
 Tipe Tabel : Master  
*Primary Key* : kode\_sub\_kegiatan  
*Foreign Key* : kode\_kegiatan

*Tabel 4. 18 Tabel Sub Kegiatan*

Nama Field	Tipe Data	Ukuran	Keterangan
kode_sub_kegiatan	Varchar	12	Kode dari sub kegiatan
nama_sub_kegiatan	Varchar	50	Nama sub kegiatan
kode_kegiatan	Varchar	12	Kode dari sub kegiatan

### 5. Tabel Program

Nama Tabel : Program  
 Tipe Tabel : Master  
*Primary Key* : kode\_program

*Tabel 4. 19 Tabel Program*

Nama Field	Tipe Data	Ukuran	Keterangan
kode_program	Varchar	12	Kode dari program
nama_program	Varchar	30	Nama dari program

### 6. Tabel Komponen

Nama Tabel : Komponen  
 Tipe Tabel : Transaksi  
*Primary Key* : kode\_komponen  
*Foreign Key* : kode\_sub\_kegiatan

*Tabel 4. 20 Tabel Komponen*

Nama Field	Tipe Data	Ukuran	Keterangan
kode_komponen	Varchar	12	Kode dari komponen
nama_komponen	Varchar	50	Nama komponen
kode_sub_kegiatan	Varchar	12	Kode sub kegiatan dari komponen

### 7. Tabel Sub Komponen

Nama Tabel : Sub Komponen  
 Tipe Tabel : Transaksi  
*Primary Key* : id\_sub\_komponen  
*Foreign Key* : kode\_komponen

*Tabel 4. 21 Tabel Komponen*

Nama Field	Tipe Data	Ukuran	Keterangan
id	Int	11	Id sub komponen
kode_sub_komponen	Varchar	12	Kode dari sub komponen
nama_sub_komponen	Varchar	50	Nama sub komponen
kode_komponen	Varchar	12	Kode dari sub komponen

### 8. Tabel Transaksi

Nama Tabel : Transaksi  
 Tipe Tabel : Transaksi  
*Primary Key* : id\_transaksi  
*Foreign Key* : kode\_program, kode\_kegiatan,  
 kode\_sub\_kegiatan, kode\_komponen,  
 kode\_sub\_komponen, id\_user

Tabel 4. 22 Transaksi

Nama Field	Tipe Data	Ukuran	Keterangan
id	Int	11	id dari data tahunan
kode_program	Varchar	30	Nama data
kode_kegiatan	Varchar	12	Tahun data
kode_sub_kegiatan	Varchar	12	Jumlah data
kode_komponen	Varchar	12	Id dari <i>user</i>
kode_sub_komponen	Varchar	12	Id dari pesantren
pagu	Double	-	Pagu dari transaksi
blokir	Double	-	Transaksi yang di blokir
realisasi	Double	-	Transaksi yang terealisasi
file_realisasi	Text	-	File transaksi yang terealisasi
sisa_dana	Double	-	Sisa dana dari transaksi
status	Enum	'rancangan_belum_diupload', 'belum_terverifikasi', 'terverifikasi', 'realisasi_terupload', 'pencairan_terupload', 'pencairan_disetujui', 'spp_terupload'	Status dari transaksi

id_user	Int	11	Id dari user
berkas_rancangan	Text	-	Berkas rancangan dari transaksi
berkas_tagihan_pencairan	Text	-	Berkas tagihan pencairan dari transaksi
surat_perintah_pembayaran	Text	-	Surat perintah pembayaran dari transaksi



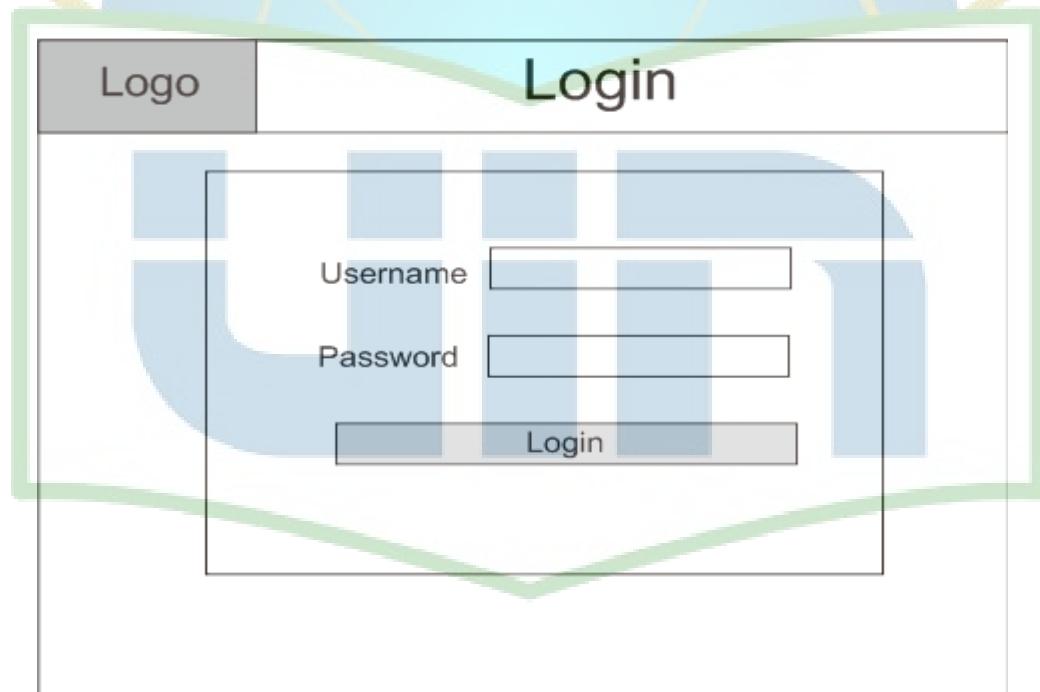
#### 4.2.4. Design Interface

Design interface adalah design untuk interaksi antara pengguna dengan aplikasi. Yang bertujuan untuk membuat interaksi pengguna sederhana dan efisien sebagai mana user bertinteraksi dengan komputer. Design interface ini sangatlah penting untuk mengetahui seberapa tingkat kesulitan aplikasi itu digunakan oleh user dan dirubah lagi sedemikian rupa supaya aplikasi tersebut menjadi sederhana dan mudah dioprasikan.

##### 1. Rancangan *Interface Login*

Nama *Usecase* : *Login*

Aktor : Admin, pelaksana kegiatan, kabag, verifikator, bendahara pengeluaran pembantu, operator bendahara



Gambar 4. 34 interface login

Sebelum masuk ke dalam sistem, aktor diharuskan melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password*. Aktor yang melakukan ini adalah Admin, Dekan, Wadek II, Kasubag keuangan, BPP, dan Unit fakultas. Setelah aktor melakukan *login*, maka aktor akan masuk ke halaman utama sistem sesuai dengan *level*-nya masing-masing.

## 2. Rancangan Interface Admin

Nama Usecase : *Manage User*

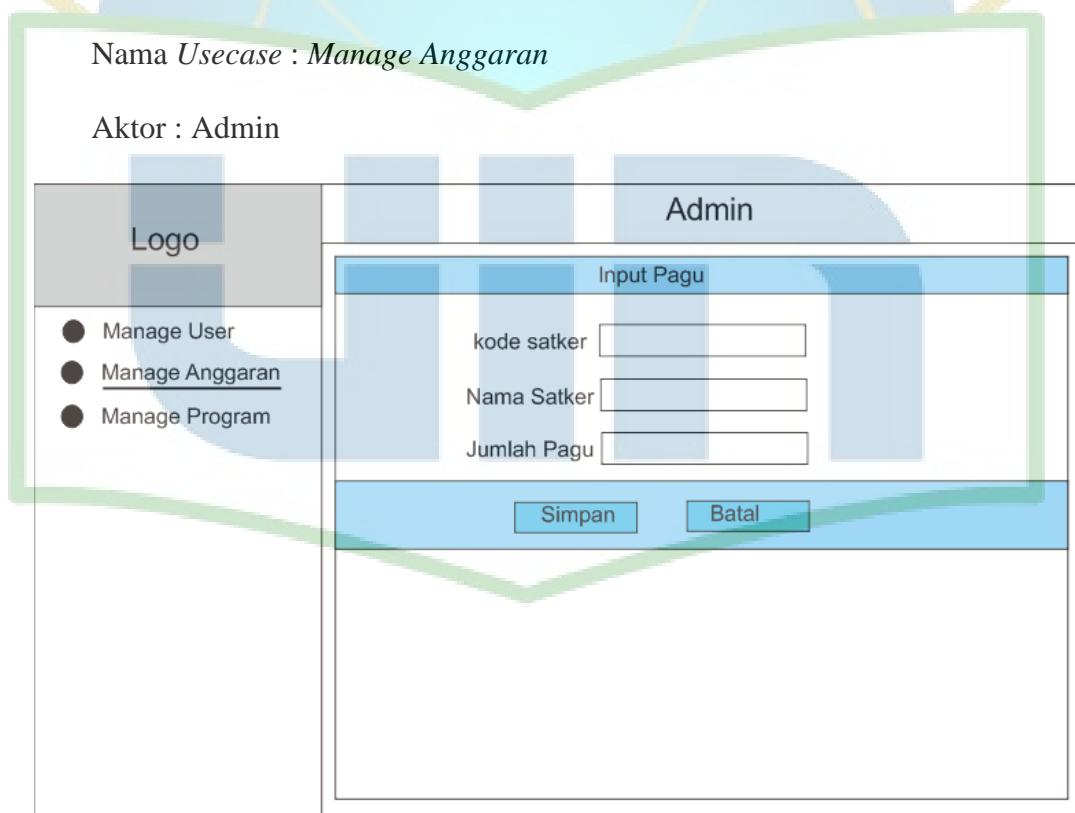
Aktor : Admin

The screenshot shows the 'Admin' interface. At the top, there is a 'Logo' section containing three circular icons with the text 'Manage User', 'Manage Anggaran', and 'Manage Program'. Below this is a navigation bar with the word 'Admin'. The main content area is titled 'Input User' and contains fields for 'Nama' (Name), 'Jabatan' (Position), and 'Level' (Level), each with an associated input field. At the bottom of the form are two buttons: 'Simpan' (Save) and 'Batal' (Cancel). The entire interface is framed by a green border.

Gambar 4. 35 Interface Admin

Perancangan *interface user* jenis kegiatan ini dilakukan oleh admin. Admin tidak bisa melakukan kegiatan *input* jenis kegiatan ini jika belum melakukan *login*. Mulanya admin memilih jenis anggaran yang ada pada sistem, 215 kemudian memasukkan nam, Jabatan, dan level. Setelah itu pilih simpan dan sistem akan menyimpan data yang baru dimasukkan ke dalam *database*. Setelah tersimpan maka admin dapat melihat dihalaman *manage user* yang berisikan data-data jenis kegiatan yang sudah ada atau yang sudah dimasukkan ke dalam *database*. Perancangan *input* desain *interface input* jenis kegiatan ini sama dengan desain *interface edit* jenis kegiatan, hanya saja jika *edit* setiap kolom *input* sudah tersedia data yang siap untuk di *edit*.

### 3. Rancangan *Interface Admin*



Gambar 4 36 Interface Admin

Perancangan *interface Manage Anggaran* jenis kegiatan ini dilakukan oleh admin. Mulanya admin memilih manage anggaran yang ada pada sistem, kemudian memasukkan kode satker, nama satker, dan jumlah pagu. Setelah itu pilih simpan dan sistem akan menyimpan data yang baru dimasukkan ke dalam *database*. Setelah tersimpan maka admin dapat melihat dihalaman *manage user* yang berisikan data-data jenis kegiatan yang sudah ada atau yang sudah dimasukkan ke dalam *database*. Perancangan *input* desain *interface input* jenis kegiatan ini sama dengan desain *interface edit* jenis kegiatan, hanya saja jika *edit* setiap kolom *input* sudah tersedia data yang siap untuk di *edit*.

#### 4. Rancangan *Interface Admin*

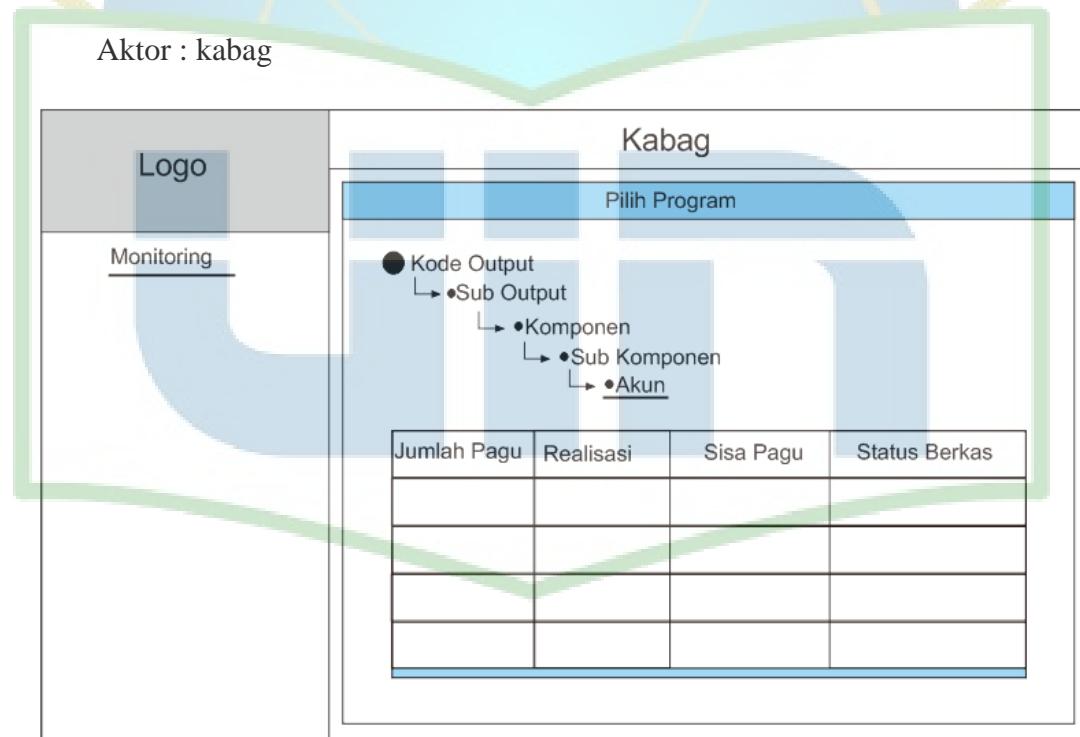
Nama Usecase : Manage Program	
Aktor : Admin	
 <ul style="list-style-type: none"> <li>● Manage User</li> <li>● Manage Anggaran</li> <li>● <u>Manage Program</u></li> </ul>	<p><b>Admin</b></p> <p><b>Input User</b></p> <p>Kode Kegiatan <input type="text"/></p> <p>Nama Kegiatan <input type="text"/></p> <p>Jumlah Pagu Kegiatan <input type="text"/></p> <p><b>Simpan</b> <b>Batal</b></p>

Gambar 4. 37 Interface Admin

Perancangan *interface Manage program* jenis kegiatan ini dilakukan oleh admin. Mulanya admin memilih manage program yang ada pada sistem, kemudian memasukkan kode kegiatan, nama kegiatan, dan jumlah pagu kegiatan. Setelah itu pilih simpan dan sistem akan menyimpan data yang baru dimasukkan ke dalam *database*. Setelah tersimpan maka admin dapat melihat dihalaman *manage user* yang berisikan data-data jenis kegiatan yang sudah ada atau yang sudah dimasukkan ke dalam *database*. Perancangan *input* desain *interface input* jenis kegiatan ini sama dengan desain *interface edit* jenis kegiatan, hanya saja jika *edit* setiap kolom *input* sudah tersedia data yang siap untuk di *edit*.

## 5. Rancangan *Interface Kabag*

*Nama Usecase : Monitoring*



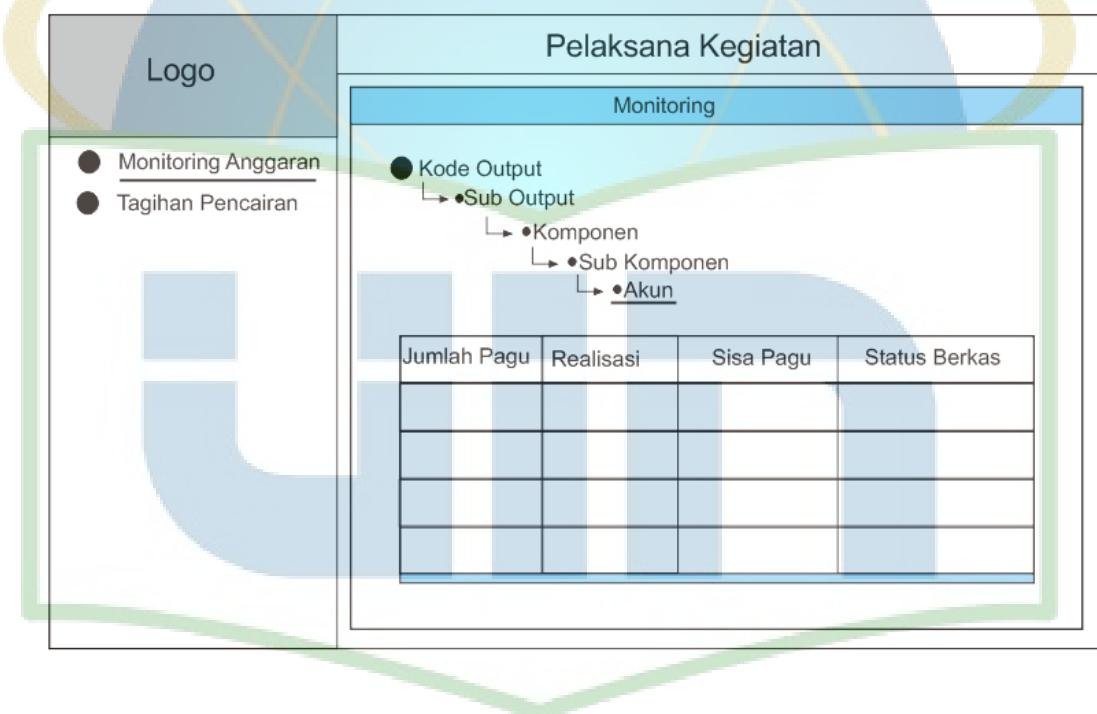
Gambar 4. 38 Interface Kabag

Perancangan *interface Monitoring* jenis kegiatan ini dilakukan oleh kabag. Kabag tidak bisa melakukan kegiatan ini jika belum melakukan *login*. Mulanya kabag memilih monitoring yang ada pada sistem, kemudian pilih kode output, pilih sub output, pilih komponen, pilih sub komponen, dan pilih akun. Setelah itu sistem akan menampilkan tampilan tabel jumlah pagu, realisasi, sisa pagu, dan status berkas

#### 6. Rancangan *Interface Pelaksana Kegiatan*

Nama *Usecase* : *Monitoring*

Aktor : Pelaksana Kegiatan



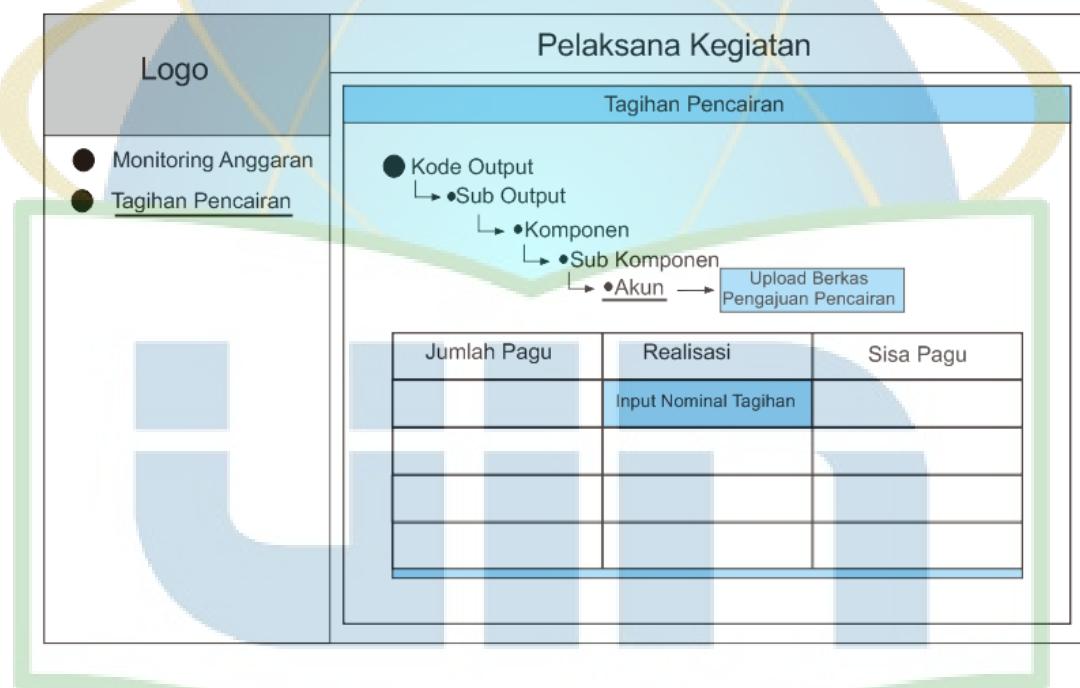
Gambar 4. 39 Interface Pelaksana Kegiatan

Perancangan *interface Monitoring* jenis kegiatan ini dilakukan oleh kabag. Mulanya kabag memilih monitoring yang ada pada sistem, kemudian pilih kode output, pilih sub output, pilih komponen, pilih sub komponen, dan pilih akun. Setelah itu sistem akan menampilkan tampilan tabel jumlah pagu, realisasi, sisa pagu, dan status berkas.

## 7. Rancangan *Interface Pelaksana Kegiatan*

Nama *Usecase* : Tagihan Pencairan

Aktor : Pelaksana Kegiatan



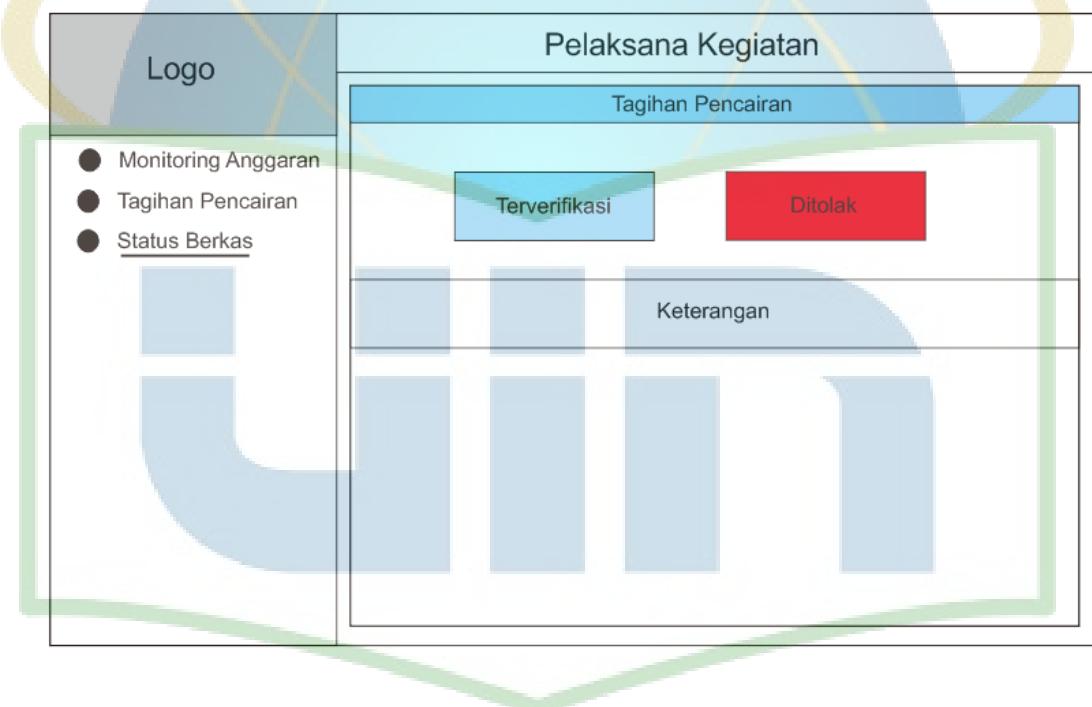
Gambar 4. 40 Interface Pelaksana Kegiatan

Perancangan *interface Tagihan Pencairan* jenis kegiatan ini dilakukan oleh Pelaksana Kegiatan. Mulanya kabag memilih monitoring yang ada pada sistem, kemudian pilih kode output, pilih sub output, pilih komponen, pilih sub komponen, dan pilih akun. Setelah itu sistem akan menampilkan tampilan tabel jumlah pagu, realisasi, sisa pagu, dan status berkas.

#### 8. Rancangan *Interface Pelaksana Kegiatan*

Nama Usecase : *Status Berkas*

Aktor : Pelaksana Kegiatan



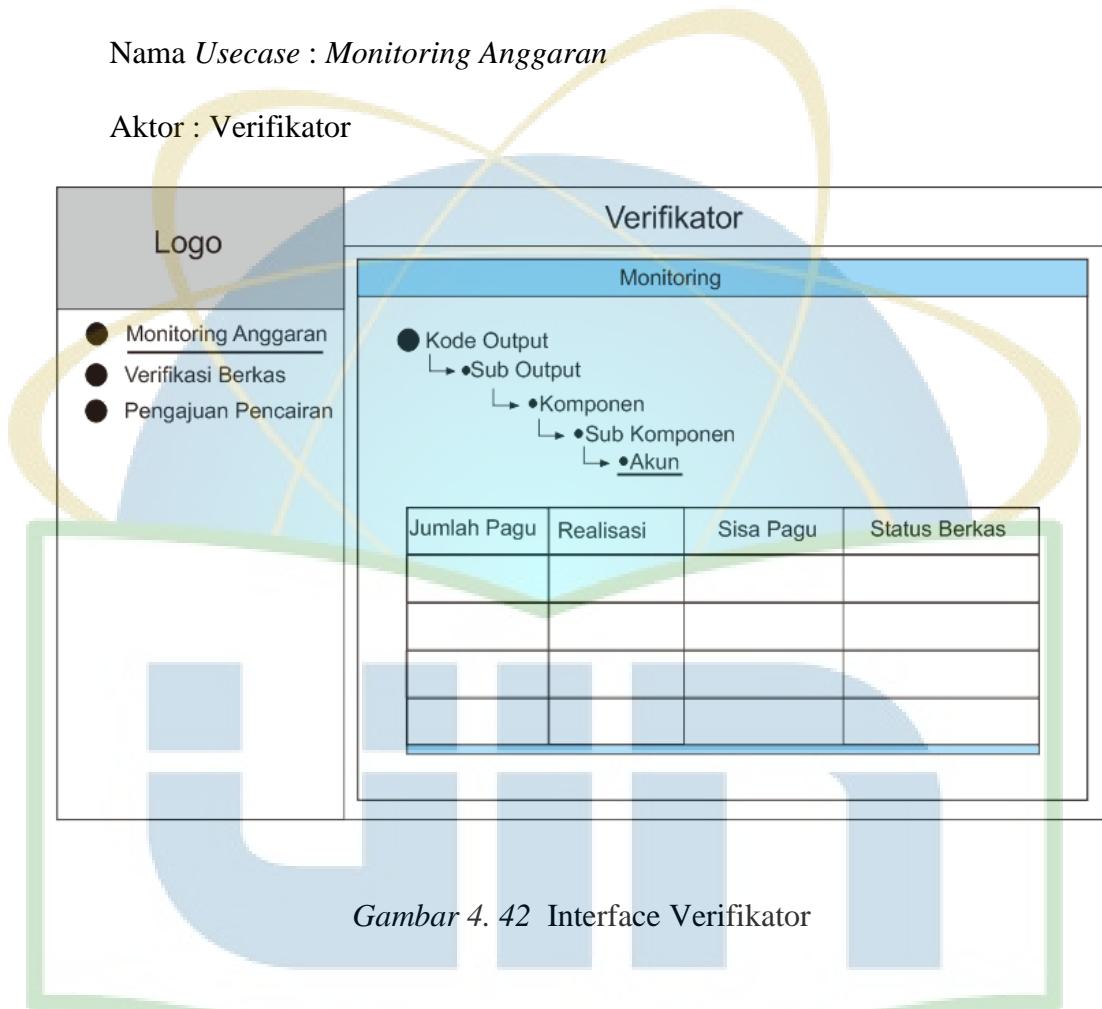
*Gambar 4. 41 Interface Pelaksana Kegiatan*

Perancangan *interface Status Berkas* jenis kegiatan ini dilakukan oleh Pelaksana Kegiatan. Mulanya kabag memilih Status Berkas yang ada pada sistem, kemudian sistem akan menampilkan status berkas tagihan pencairan.

#### 9. Rancangan *Interface Verifikator*

Nama *Usecase* : *Monitoring Anggaran*

Aktor : Verifikator



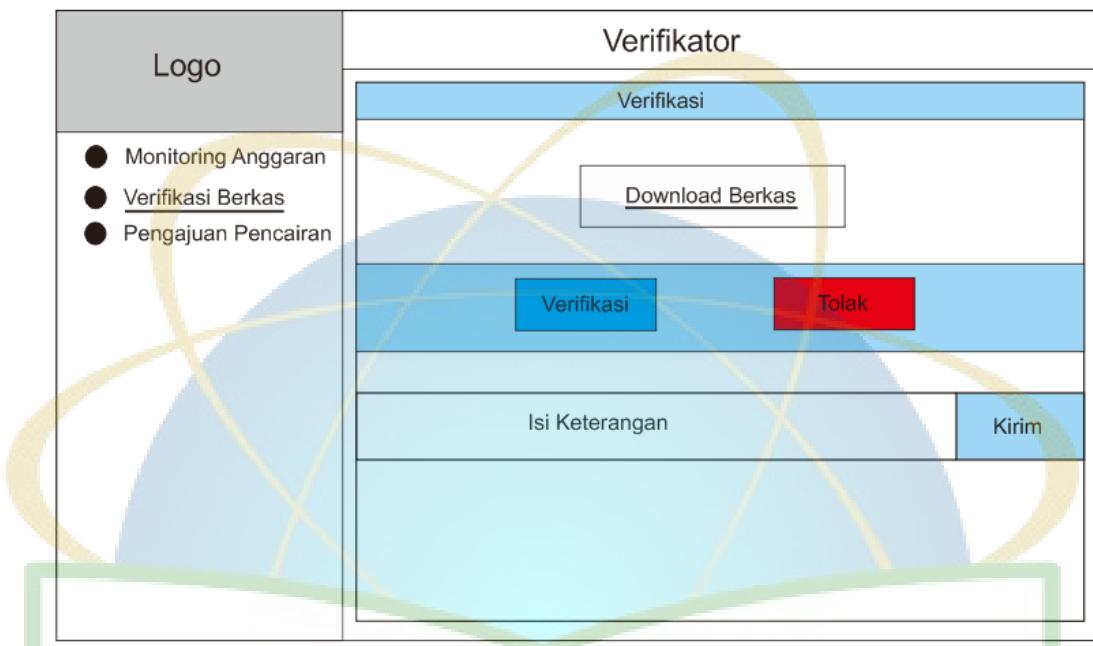
Gambar 4. 42 Interface Verifikator

Perancangan *interface Monitorin* jenis kegiatan ini dilakukan oleh verifikator untuk mengetahui jumlah pagu di dalam anggaran kegiatan yang sudah terpakai agar pagu tidak minus.. Mulanya verifikator memilih monitoring yang ada pada sistem, kemudian pilih kode output, pilih sub output, pilih komponen, pilih sub komponen, dan pilih akun. Setelah itu sistem akan menampilkan tampilan tabel jumlah pagu, realisasi, sisa pagu, dan status berkas.

## 10. Rancangan Interface Verifikator

Nama UseCase : *Verifikator Berkas*

Aktor : Verifikator



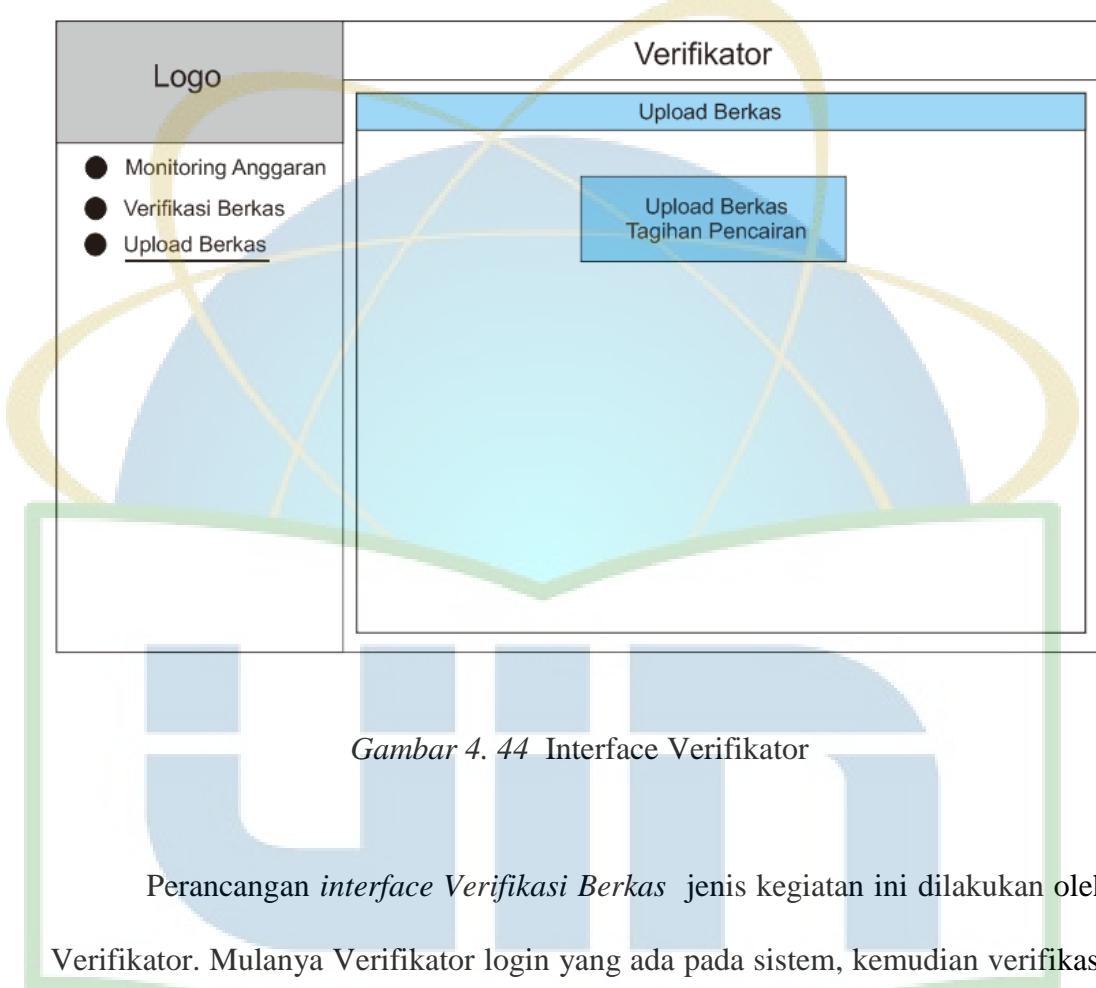
Gambar 4.43 Interface Verifikator

Perancangan *interface Verifikasi Berkas* jenis kegiatan ini dilakukan oleh Verifikator. Mulanya Verifikator login yang ada pada sistem, kemudian verifikasi berkas yang telah dikirimkan oleh Pelaksana Kegiatan, download berkas yang telah diupload Pelaksana Kegiatan untuk diverifikasi atau pilih tolak untuk menolak berkas. Setelah itu isi keterangan berkas telah di proses oleh verifikator.

## 11. Rancangan *Interface Verifikator*

Nama *Usecase* : *Upload Berkas*

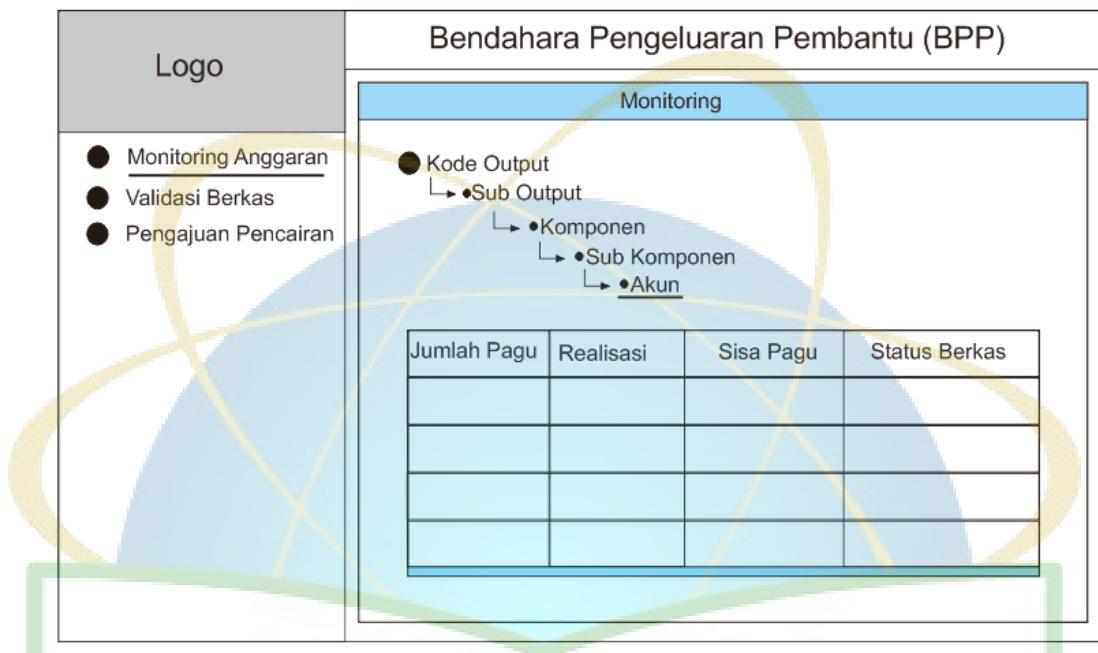
Aktor : Verifikator



## 12. Rancangan Interface Bendahara Pengeluaran Pembantu (BPP)

Nama Usecase : Monitoring Anggaran

Aktor : Bendahara Pengeluaran Pembantu (BPP)



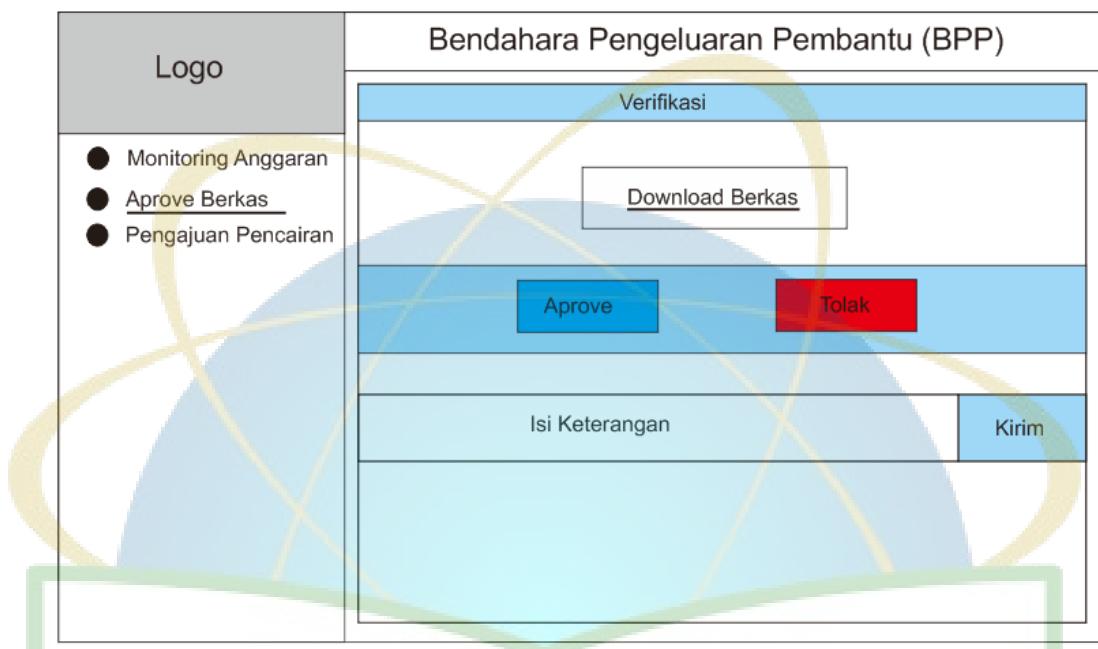
Gambar 4. 45 Interface Bendahara Pengeluaran Pembantu (BPP)

Perancangan interface Monitoring jenis kegiatan ini dilakukan oleh Bendahara Pengeluaran Pembantu (BPP) untuk mengetahui jumlah pagu di dalam anggaran kegiatan yang sudah terpakai agar pagu tidak minus.. Mulanya BPP memilih monitoring yang ada pada sistem, kemudian pilih kode output, pilih sub output, pilih komponen, pilih sub komponen, dan pilih akun. Setelah itu sistem akan menampilkan tampilan tabel jumlah pagu, realisasi, sisa pagu, dan status berkas.

### 13. Rancangan Interface Bendahara Pengeluaran Pembantu (BPP)

Nama Usecase : Aprove Berkas

Aktor : Bendahara Pengeluaran Pembantu (BPP)



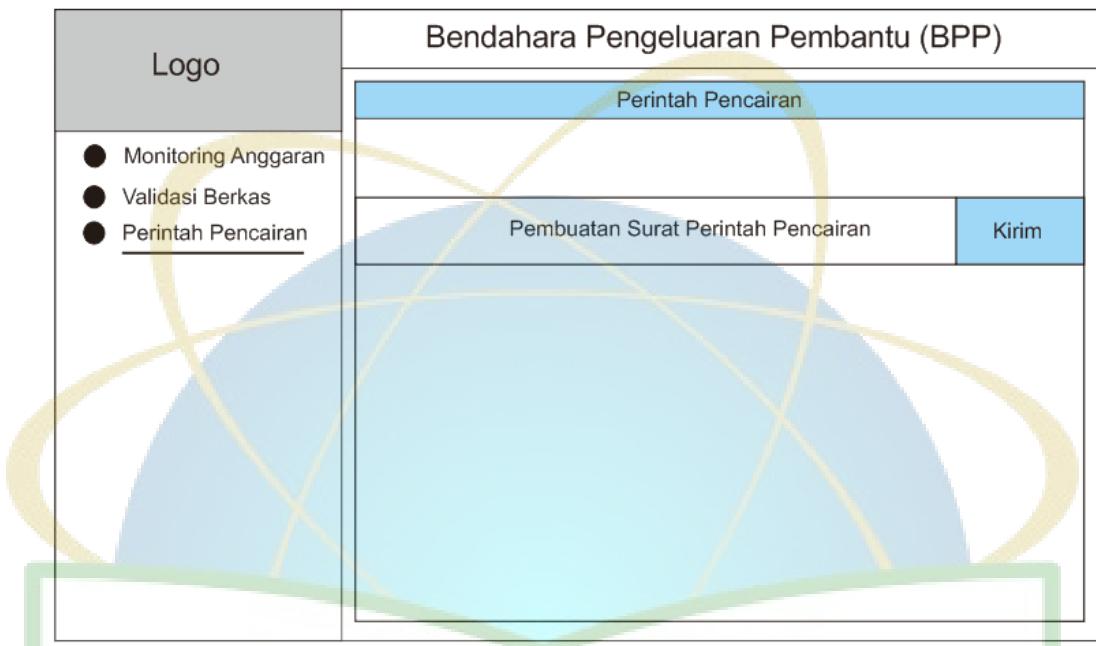
Gambar 4. 46 Interface Bendahara Pengeluaran Pembantu (BPP)

Perancangan *interface Aprove Berkas* jenis kegiatan ini dilakukan oleh Bendahara Pengeluaran Pembantu (BPP). Mulanya BPP login yang ada pada sistem, kemudian Aprove berkas pencairan yang telah dikirimkan oleh Pelaksana Verifikator, download berkas yang telah diupload Pelaksana Kegiatan untuk Aprove atau pilih tolak untuk menolak berkas. Setelah itu isi keterangan berkas telah di proses oleh Bendahara Pengeluaran Pembantu (BPP).

#### 14. Rancangan Interface Bendahara Pengeluaran Pembantu (BPP)

Nama Usecase : Perintah Pencairan

Aktor : Bendahara Pengeluaran Pembantu (BPP)



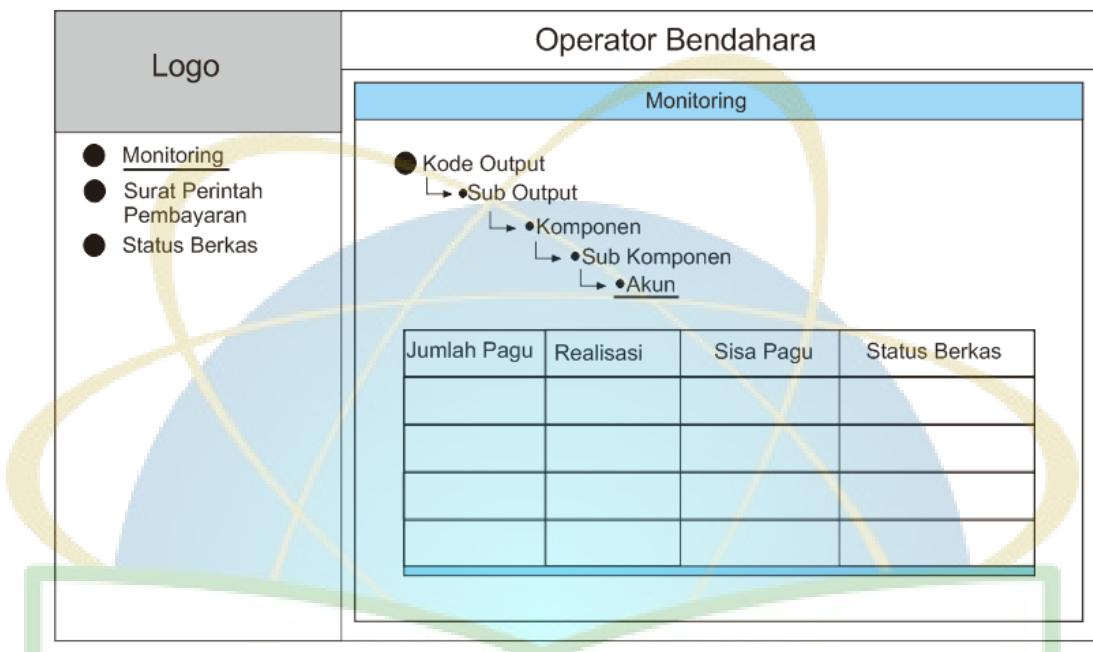
Gambar 4. 47 Interface Bendahara Pengeluaran Pembantu (BPP)

Perancangan *interface Pembuatan Surat Perintah Pencairan* jenis kegiatan ini dilakukan oleh Bendahara Pengeluaran Pembantu (BPP). Mulanya BPP login yang ada pada sistem, kemudian Perintah Pencairan penacairan yang Akan diproses oleh Oprator Bendahra.

## 15. Rancangan Interface Operator Bendahara

Nama Usecase : Monitoring Anggaran

Aktor : Operator Bendahara



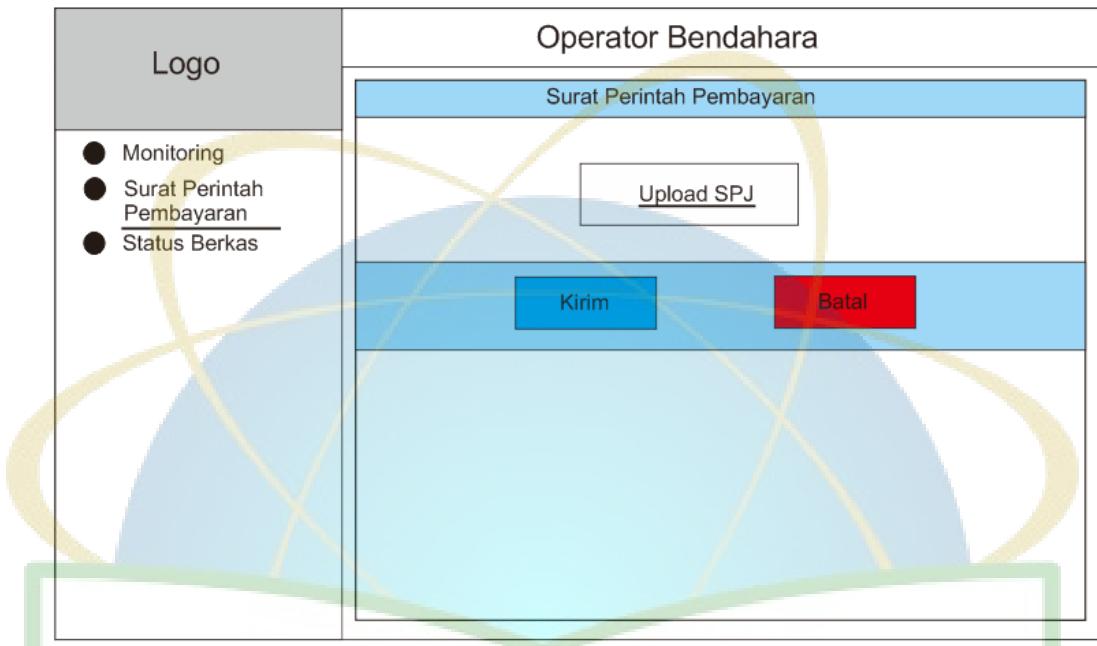
Gambar 4. 48 Interface Operator Bendahara

Perancangan *interface Monitorin* Anggaran kegiatan ini dilakukan oleh Operator bendaha untuk mengetahui jumlah pagu di dalam anggaran kegiatan yang sudah terpakai agar pagu tidak minus.. Mulanya Oprator Bendahara memilih monitoring yang ada pada sistem, kemudian pilih kode output, pilih sub output, pilih komponen, pilih sub komponen, dan pilih akun. Setelah itu sistem akan menampilkan tampilan tabel jumlah pagu, realisasi, sisa pagu, dan status berkas.

## 16. Rancangan Interface Operator Bendahara

Nama Usecase : Surat Perintah Pembayaran

Aktor : Operator Bendahara



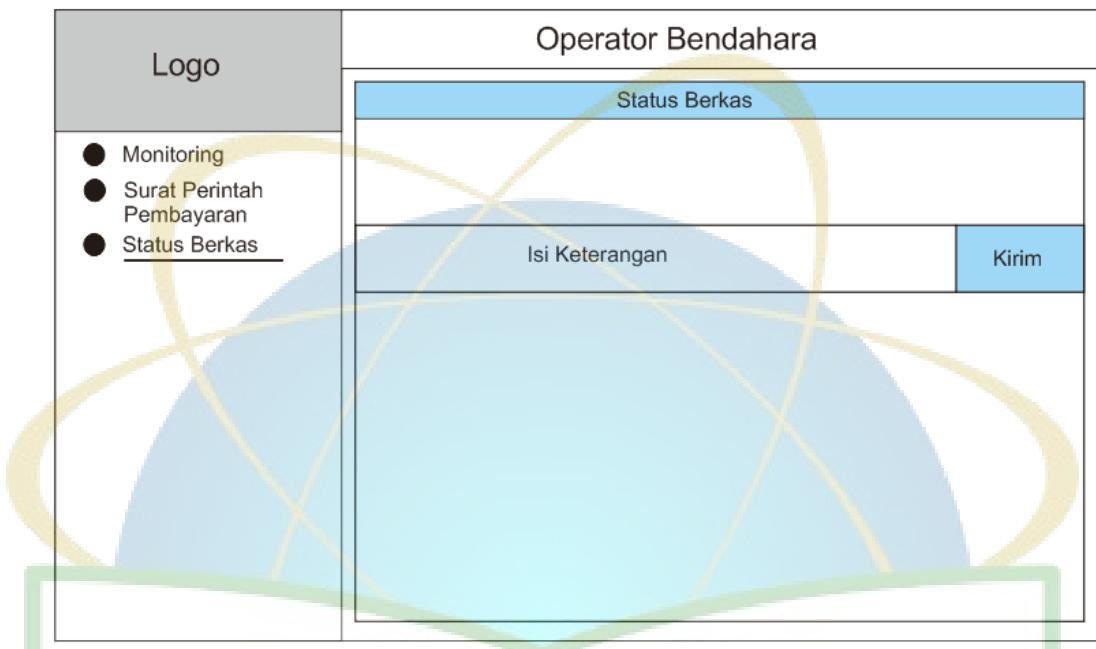
Gambar 4.49 Interface Operator Bendahara

Perancangan *interface Aprove Berkas* jenis kegiatan ini dilakukan oleh Bendahara Pengeluaran Pembantu (BPP). Mulanya BPP login yang ada pada sistem, kemudian Aprove berkas penacairan yang telah dikirimkan oleh Pelaksana Verifikator, download berkas yang telah diupload Pelaksana Kegiatan untuk Aprove atau pilih tolak untuk menolak berkas. Setelah itu isi keterangan berkas telah di proses oleh Bendahara Pengeluaran Pembantu (BPP).

## 17. Rancangan Interface Operator Bendahara

Nama Usecase : Status Berkas

Aktor : Operator Bendahara



Gambar 4. 50 Interface Operator Bendahara

Perancangan interface status berkas jenis kegiatan ini dilakukan oleh Operator Bendahara. Mulanya Operator Bendahara login yang ada pada sistem, kemudian pilih status berkas. Didalam sistem diminta untuk memberi keterangan status berkas yang akan muncul di setiap monitoring anggaran.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan pembahasan yang sudah diuraikan pada bab-bab sebelumnya, maka dapat diambil kesimpulan dari penelitian mengenai Rancang Bangun Sistem Monitoring Anggaran Keuangan Biro Umum adalah sebagai berikut:

1. Sistem memberikan kemudahan dalam pengelolaan penyerapan anggaran dan pencocokan realisasi dengan akun belanja .
2. Sistem memberikan kemudahan khususnya bagi Staf Biro Hukum agar dalam sistem yang baru ini dapat memisahkan hak akses dan dapat mengetahui kode akun yang menjadi tanggung jawab setiap user pengguna sistem monitoring keuangan Biro Hukum.
3. Menghasilkan sebuah rancangan aplikasi sistem monitoring anggaran keuangan berbasis web yang digunakan untuk memudahkan staf Biro Hukum mengetahui perincian anggaran keuangan setiap bulan ataupun setiap tahun pada Biro Hukum

Kementerian Agama RI.

## 5.2 Saran

Pada Rancang Bangun Sistem Monitoring Anggaran Keuangan Pada Biro Umum Kementerian Agama RI, masih terdapat keterbatasan dan kekurangan serta memerlukan perbaikan untuk meningkatkan manfaat dari sistem ini yang dapat diwujudkan pada penelitian selanjutnya.

1. Peneliti selanjutnya dapat melakukan pengembangan pada Rancang Bangun Sistem Monitoring Anggaran Keuangan Biro Hukum dan Kerja Sama Luar Negeri Kementerian Agama RI berupa penambahan fitur ataupun melakukan integrasi dengan bagian lain terkait anggaran keuangan.
2. Peneliti selanjutnya dapat menggunakan metode dan model analisa yang lainnya serta pengembangan aplikasi dengan menggunakan *framework* lain agar dalam pembuatan aplikasi lebih terstruktur dan dapat dengan mudah dikembangkan.
3. Perlu adanya perbaikan sistem dalam segi keamanan (*security*).

## DAFTAR PUSTAKA

- Anonim, 2005. PP no. 23/2005 tentang Pengelolaan Keuangan Badan Layanan Umum Pasal 1 ayat (1), Pasal 2.
- Anonim, 2012. Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia no. 58 Tahun 2012 tentang Bantuan Operasional Perguruan Tinggi yang diselenggarakan oleh Pemerintah.
- Biro Hukum Setjen Kementerian Agama. 2018. Struktur Organisasi. [Online] <https://hkln.kemenag.go.id/profil-9-struktur-organisasi.html>. Diakses Tanggal 16 Oktober 2018.
- Dadan, Umar, Daihani. 2001. Komputerisasi Pengambilan Keputusan. Jakarta. Elex Media Computindo.
- Departemen Pertanian. 2004. Sistem *Monitoring, Evaluasi, dan Pelaporan*. Departemen Pertanian.
- Hariyato, B. 2004. Sistem Manajemen Basis Data, Cetakan Pertama. Bandung. Informatika.
- Henderi, 2007. *Analysis and Design System with Unfied Modelling Language (UML)*. Tangerang, STMIK Raharja.
- Jogiyanto, H.M. 2005. Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur. Yogyakarta. Andi Offset.
- Jogiyanto. 2008. Metodologi Penelitian Sistem Informasi. Yogyakarta. Andi Offset.
- Kadir, A. 2003. Pengenalan Sistem Infromasi. Yogyakarta. Andi Offset.

- Kendall KE, Kendall JE. 2008. *System Analysis and Design Fifth Edition*. Hafidh TA, penerjemah, Jakarta: PT. Indeks Kelompok Gramedia, terjemahandari: Pearson Education.
- Kusek & Rist. 2004. Sepeluh Langkah untuk Berbasis Hasil Monitoring Evaluasi dan Sistem Buku Pegangan, untuk Praktisi Pembangunan. Washington. Publikasi Bank Dunia.
- Ladjamudin, al-bahra bin. 2005. Analisis dan Desain Sistem Informasi. Yogyakarta. Graha Ilmu.
- Mathiassen, Lars. et all. 2000. *Object-Oriented Analysis and Design*. Denmark. Marko Publishing.
- McLeod, Raymond & George Schell. 2004. Sistem Informasi Manajemen (Edisi Kedelapan). Edisi Bahasa Indonesia. Jakarta. INDEKS.
- Munawar. 2005. Pemodelan Visual dengan UML. Yogyakarta. Graha Ilmu.
- Nafarin, M. 2004. Penganggaran Perusahaan. Jakarta. Salemba Empat.
- Pressman, Roger S. 2002. Rekayasa Perangkat Lunak Pendekatan Praktisi. Yogyakarta. Andi.
- Sholiq. 2006. Pemodelan Sistem Informasi Berorientasi Objek dengan UML. Yogyakarta. Graha Ilmu.
- Sidik, Betha. 2005. Mysql Untuk Pengguna, Administrator, dan Pengembangan Web. Bandung, Informatika.
- Simarmata J. 2010. Rekayasa Perangkat Lunak. Yogyakarta. Andi.
- Siregar, Baldric dan Bonni Siregar, 2001. Akuntasi Pemerintahan dengan Sistem Dana. Yogyakarta. STIE YKPN

- Soetanto H. 2002. Sistem Informasi. Univ. Jakarta. Budi Luhur.
- Suhendar A, Gunadi H. 2002. Visual Modelling Menggunakan UML dan Rational Rose. Bandung. Informatika.
- Suprianto, 2008. Buku Pintar Pemrograman PHP. Bandung. OASE Media.
- Sutabri, Tata. 2012. Konsep Sistem Informasi. Yogyakarta: Penerbit ANDI
- Sutanta, Edhy. 2011. Basis Data dalam Tinjauan Konseptual.Yogyakarta. Andi Offset.
- Susanto. 2002. Sistem Informasi Manajemen, Edisi 2. Bandung. Lingga Jaya.
- Whitten, Bently, Ditman. 2004. Metode Desain & Analisis Sistem Edisi 6. Yogyakarta. Andi

## LAMPIRAN-LAMPIRAN

### LAMPIRAN 1 HASIL WAWANCARA

**Hari/tanggal : Kamis, 3 Juli 2018**

**Narasumber : Prof. H. Achmad Gunaryo (Kepala Biro Hukum dan Kerja**

**Sama Luar Negeri Setjen Kementerian Agama)**

**Ibu Nilam Nur Azizah (Bendahara Pengeluaran Pembantu)**

**Penanya : Agung Ramadhan**

**Hasil : Mengetahui proses bisnis yang berjalan dan mengetahui masalah yang dihadapi Biro Hukum dan Kerja Sama Luar Negeri Setjen Kementerian Agama**

1. Biro Hukum dan Kerjasama Luar Negeri itu satker yang bergerak dibidang apa?

Jawab : Biro Hukum dan Kerjasama Luar Negeri itu satker yang bergerak di bidang pelayanan hukum dan hubungan bilateral. Yang biasa kami kerjakan di biro hukum ini adalah membuat regulasi yang tentang peraturan Menteri Agama dan Keputusan menteri Agama, sebagai kuasa mutlak Menteri Agama di dalam naungan Sekretariat Jenderal Kementerian Agama Republik Indonesia dan juga pencancangan peraturan perintah, perancangan MOU, seperti percanangan undang-undang dalam lingkup agama. Hubungan bilateral luar Negeri (Prof. H. Achamnd Gunaryo)

2. Lantas bagaimana alur mulainya berkas untuk diajukan pencairan untuk kegiatan ?

Jawab : Biasanya ada nota dinas kebagian-bagian yang ada di dalam satker biro hukum dan kerja sama luar negeri dari kepala biro. Lalu dirapatkan menetukan tujuan kawil atau satker daerah raugn lingkup kementerian agama setelah disepakati dilanjutkan oleh kabag masing-

terlambat bisa ketahuan dari tanda terimanya masih manual. Tapi kita agak sulit juga, jadi terkadang kita agak sulit juga menemukan dan merapihkan data-data tersebut sampai jadi informasi yang berguna. (Nilam Nur Azizah)).

3. Apakah sering ada kendala terkait dengan arus sirkulasi berkas pencairan tersebut?

Jawab : Sering, kita sering terlambat dalam pencairan dikarenakan keluar masuk berkas tidak tertib. Sebenarnya hal tersebut sering kali terjadi karena semua berkas yang kami kelola masih manual, kami kesulitan untuk membuat rekapan segala alur berkas pencairan akibatnya kadang terjadi ketidaksesuaian data dan realita yang ada dilapangan. Bahkan ada sikap saling tuduh diantara bagian kepada pengelola keuangan dan terjadilah keributan. (Nilam Nur Azizah).

Ya dari pihak pengelola keuangan pun terkadang kesulitan dalam membuat rekapan berkas beserta riwayat sirkulasi yang ada di arsip karena data kami tercerer.

4. Apakah dengan sistem yang sudah berjalan selama ini sudah ada upaya perbaikan untuk mengatasi masalah alur berkas pencairan tersebut?

Jawab : Sudah kita upayakan perbaikan-perbaikan mendasar seperti menegaskan ke pelaksanaan kegiatan agar memberikan berkas harus mencatat di buku tanda terima. Tapi, ya itu tadi, karena ketidak tertiban dan pencatatannya masih manual para pengelola keuangan ini jadi tidak tercover dan jadilah berntakan. (Prof. H. Achmad Gunaryo).

5. Terkait dengan masalah tersebut apa yang dibutuhkan untuk meminimalisir hal tersebut?

Jawab : Berhubung kita mengelola data masih secara manual, ada baiknya jika ada aplikasi yang mengelola berkas pencairan, selain itu aplikasi yang kami harapkan mungkin bisa memberitahu dan arus berkas pencairan yang melebihi

batas kebutuhan, kecuali ada hal atau alasan tertentu. Sehingga memudahkan saya melakukan tugas saya dalam pencarian,

karena selama ini apabila ada pengajuan pencairan saya harus mengecek dulu dokumen-dokumen sebelumnya untuk memperhitungkan apakah berkas ini yang harus di dahulukan atau di terbelakangkan (Nilam Nur Azizah).

Saya juga butuh adanya aplikasi yang dapat mengontrol dan me-*monitoring* arus pencairan anggaran secara *real-time*. Selama ini dalam me-*monitoring* pencairan anggaran dirasa kurang maksimal, terlebih lagi apabila saya meminta tanda terima berkas, itu membutuhkan waktu yang tidak pasti, kadang bisa cepat, kadang bisa juga sampai besoknya karena harus disusun terlebih dahulu surat pertanggung jawabannya. Hal demikian tentunya sangat menghambat saya dalam mengambil keputusan (Nilam Nur Azizah).

6. Sebenarnya adakah harapan yang ingin dicapai dengan adanya aplikasi yang akan dibangun ini?

Jawab : Saya berharap dengan adanya aplikasi baru ini maka, peran saya selaku bendahara pengeluaran pembantu dalam setiap proses pencairan yang berjalan dapat dilaksanakan secara cepat dengan didukung data yang akurat, saya juga berharap agar saya dapat mengetahui dan mengontrol secara menyeluruh setiap kelengkapan berkas pencairan suatu kegiatan, sehingga dapat mendeteksi dini bahkan sampai mencegah terjadinya keterlambatan pencairan, dan kehilangan berkas di setiap proses yang berjalan. (Nilam Nur Azizah).

Kalau saya sih, cuma ingin agar bisa cepat dan mudah mencairkan anggaran suatu kegiatan, terutama mengenai alur berkas pencairan dan berkas tersebut sudah dalam bentuk surat pertanggung jawababan. (Nilam Nur Aziza

## LAMPIRAN 2

### TAMPILAN APLIKASI

#### 1. Halaman *Login*

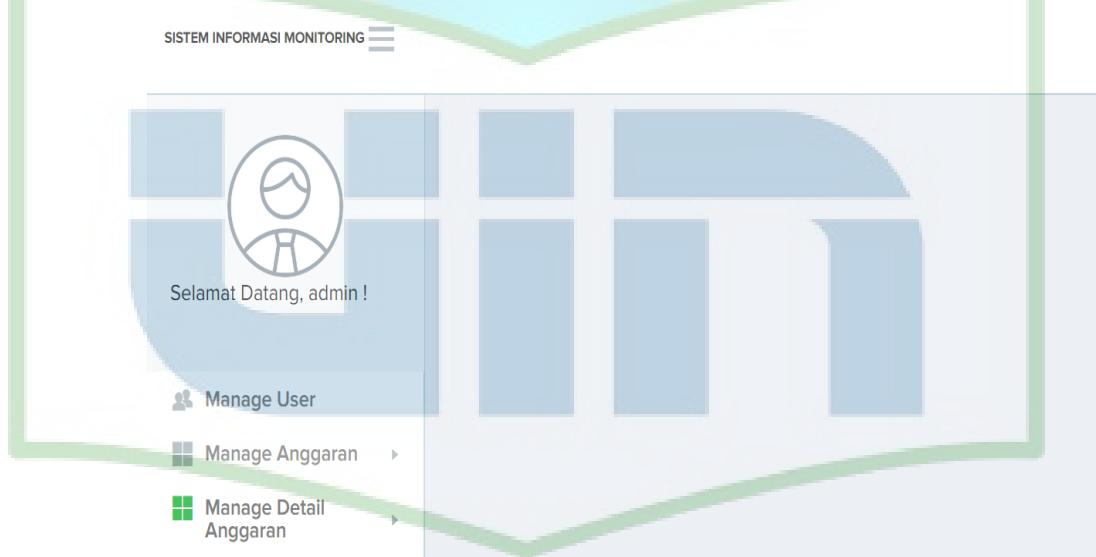
LOGIN

Username  
admin

Password  
.....

Masuk

#### 2. Halaman Home Admin



### 3. Halaman Kabag Monitoring Anggaran Keuangan

List Data					
		Pembinaan Administrasi Hukum dan KLN	Pagu: Rp 12.041.208.000,00	Realisasi	Sisa Dana: Rp 0,00
Kode	Urain	Pagu	Realisasi	Sisa Dana	
0957	Layanan Hukum	Rp 5.661.407.000,00	Rp 700.000,00	Rp 5.660.707.000,00	
001	Tangg Sub Output	Rp 5.631.407.000,00	Rp 700.000,00	Rp 5.630.707.000,00	
051	Koordinasi Pengusutan Produk Hukum	Rp 1.414.168.000,00	Rp 700.000,00	Rp 1.413.468.000,00	
➤ A	Penyusunan Peraturan Menteri	Rp 414.168.000,00	Rp 700.000,00	Rp 413.468.000,00	
➤ B	Penyusunan Keputusan Menteri	Rp 0,00	Rp 0,00	Rp 0,00	
➤ C	Penyusunan Rancangan Peraturan Undang-Undangan dan	Rp 400.000.000,00	Rp 0,00	Rp 0,00	
➤ D	PENYUSUNAN RANCANGAN PERATURAN PRESIDEN, KEPUTUSAN	Rp 0,00	Rp 0,00	Rp 0,00	
052	Bantuan Hukum	Rp 442.934.000,00	Rp 0,00	Rp 0,00	
053	Pembinaan di Bidang Hukum	Rp 3.774.305.000,00	Rp 0,00	Rp 0,00	

### 4. Halaman Admin Manage User

List Data						
ID	Name		Username	Email	No Telepon	Aksi
1	kabag 1		kabag	farian.tasa@gmail.co	8162126	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	pelaksana kegiatan		pelaksana	agung@gmail.crot	2147483647	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

## 5. Halaman Admin Manage Anggaran

SISTEM INFORMASI MONITORING

Selamat Datang, admin !

Manage User

Manage Anggaran

- Program
- Kegiatan
- Sub Kegiatan
- Komponen
- Sub Komponen
- Akun

Tambah

List Data

ID	Name	Username	Email	No Telepon	Aksi
1	kabag 1	kabag	farian.tasa@gmail.co	8162126	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	pelaksana kegiatan	pelaksana	agung@gmail.crot	2147483647	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

## 6. Halaman Admin Manage Detail Anggaran

SISTEM INFORMASI MONITORING

Selamat Datang, admin !

Manage User

Manage Anggaran

Manage Detail Anggaran

- Program
- Kegiatan
- Sub Kegiatan
- Komponen
- Sub Komponen

List Data

No	Kode Program	Program	Aksi
1	2098	Pembinaan Administrasi Hukum dan KLN	<button>Edit</button>

## 7. Halaman Pelaksana Kegiatan Monitoring Anggaran

SISTEM INFORMASI MONITORING

The screenshot shows a user interface for monitoring financial activities. On the left, there's a sidebar with a profile icon and two main menu items: "Monitoring Anggaran" and "Upload Pengajuan Kegiatan". The main content area has a title "List Data" and a table with the following data:

Kode	Uraian	Pagu	Realisasi	Sisa Dana
✓ 957	Layanan Hukum	Rp 5.661.407.000,00	Rp 700.000,00	Rp 5.660.707.000,00
✓ 001	Tanpa Sub Output	Rp 5.631.407.000,00	Rp 700.000,00	Rp 5.630.707.000,00
➢ 051	Koordinasi Penyusunan Produk Hukum	Rp 1.414.168.000,00	Rp 700.000,00	Rp 1.413.468.000,00
➢ 052	Bantuan Hukum	Rp 442.934.000,00	Rp 0,00	Rp 0,00
053	Pembinaan di Bidang Hukum	Rp 3.774.305.000,00	Rp 0,00	Rp 0,00

A yellow oval highlights the "Monitoring Anggaran" button in the sidebar.

## 8. Halaman Pelaksana Upload Pengajuan Kegiatan

SISTEM INFORMASI MONITORING

This screenshot shows a form for uploading a proposal. The left sidebar includes the same "Monitoring Anggaran" and "Upload Pengajuan Kegiatan" options as the previous page. The main form fields are:

- Layanan Hukum
- Nama Sub Kegiatan
- Nama Komponen
- Koordinasi Penyusunan Produk Hukum
- Nama Sub Komponen
- Nama Akun
- Nominal Pagu Akun
- Nama Kegiatan
- Nominal
- Berkas Kegiatan
- Pilih File (button)
- Tidak ada file yang dipilih (text)
- Simpan (green button at the bottom)

A green box highlights the "Upload Pengajuan Kegiatan" button in the sidebar.

## 9. Halaman Verifikator Validasi Berkas Pengajuan Kegiatan

SISTEM INFORMASI MONITORING

X

Kegiatan Realisasi

No	Nama Kegiatan	Nominal	Berkas	Status	Aksi	
1	Belanja ATK	700000	<a href="#">Download Berkas</a>	Menunggu verifikasi Verifikator	<a href="#">Verifikasi</a>	
▼ 957	Layanan Hukum			Rp 5.661.407.000,00	Rp 700.000,00	Rp 5.660.707.000,00
▼ 001	Tanpa Sub Output			Rp 5.631.407.000,00	Rp 700.000,00	Rp 5.630.707.000,00
▼ 051	Koordinasi Penyusunan Produk Hukum			Rp 1.414.168.000,00	Rp 700.000,00	Rp 1.413.468.000,00
▼ A	Penyusunan Peraturan Menteri			Rp 414.168.000,00	Rp 700.000,00	Rp 413.468.000,00
521211	belanja bahan			Rp 167.000.000,00	Rp 700.000,00	Rp 166.300.000,00
521219	Belanja Barang Non Operasional Lainnya			Rp 98.000.000,00	Rp 0,00	Rp 0,00
522151	Beban Jasa Profesi			Rp 149.168.000,00	Rp 0,00	Rp 0,00
► B	Penyusunan Keputusan Menteri			Rp 0,00	Rp 0,00	Rp 0,00
► C	Penyusunan Rancangan Peraturan Undang-Undangan dan			Rp 400.000.000,00	Rp 0,00	Rp 0,00

## 10. Halaman Pelaksana Monitoring keuangan

SISTEM INFORMASI MONITORING

LIST DATA

Pembinaan Administrasi Hukum dan KLN	Pagu Rp 12.041.208.000,00	Sisa Dana Rp 0,00		Pagu	Realisasi	Sisa Dana
Pembinaan Administrasi Hukum dan KLN	Pagu Rp 12.041.208.000,00	Sisa Dana Rp 0,00				
▼ 957	Layanan Hukum			Rp 5.661.407.000,00	Rp 700.000,00	Rp 5.660.707.000,00
▼ 001	Tanpa Sub Output			Rp 5.631.407.000,00	Rp 700.000,00	Rp 5.630.707.000,00
▼ 051	Koordinasi Penyusunan Produk Hukum			Rp 1.414.168.000,00	Rp 700.000,00	Rp 1.413.468.000,00
▼ A	Penyusunan Peraturan Menteri			Rp 414.168.000,00	Rp 700.000,00	Rp 413.468.000,00
521211	belanja bahan			Rp 167.000.000,00	Rp 700.000,00	Rp 166.300.000,00
521219	Belanja Barang Non Operasional Lainnya			Rp 98.000.000,00	Rp 0,00	Rp 0,00
522151	Beban Jasa Profesi			Rp 149.168.000,00	Rp 0,00	Rp 0,00
► B	Penyusunan Keputusan Menteri			Rp 0,00	Rp 0,00	Rp 0,00
► C	Penyusunan Rancangan Peraturan Undang-Undangan dan			Rp 400.000.000,00	Rp 0,00	Rp 0,00
► D	PENYUSUNAN RANCANGAN PERATURAN PRESIDEN, KEPUTUSAN			Rp 0,00	Rp 0,00	Rp 0,00
052	Bantuan Hukum			Rp 442.834.000,00	Rp 0,00	Rp 0,00
053	Pembinaan di Bidang Hukum			Rp 3.774.305.000,00	Rp 0,00	Rp 0,00

## 11. Halaman Bendahara Monitoring Anggaran

SISTEM INFORMASI MONITORING

The screenshot shows a dashboard for monitoring financial transactions. On the left, there's a logo and a welcome message "Selamat Datang, bendahara!". Below it is a section titled "Monitoring Anggaran". The main area displays a table titled "List Data" with the following information:

Pembinaan Administrasi Hukum dan KLN Pagu Rp 12.041.208.000,00 Sisa Dana Rp 0,00

Kode	Uraian	Pagu	Realisasi	Sisa Dana
▼ 957	Layanan Hukum	Rp 5.661.407.000,00	Rp 700.000,00	Rp 5.660.707.000,00
▼ 001	Taripa Sub Output	Rp 5.631.407.000,00	Rp 700.000,00	Rp 5.630.707.000,00
► 051	Koordinasi Penyusutan Produk Hukum	Rp 1.414.168.000,00	Rp 700.000,00	Rp 1.413.468.000,00
► 052	Bantuan Hukum	Rp 442.934.000,00	Rp 0,00	Rp 0,00
053	Pembinaan di Bidang Hukum	Rp 3.774.305.000,00	Rp 0,00	Rp 0,00
► 994	Pelayanan Perkantoran	Rp 858.397.000,00	Rp 0,00	Rp 0,00
► 051	Koordinasi Penyusutan Produk Hukum	Rp 1.414.168.000,00	Rp 700.000,00	Rp 1.413.468.000,00
► 052	Bantuan Hukum	Rp 442.934.000,00	Rp 0,00	Rp 0,00
053	Pembinaan di Bidang Hukum	Rp 3.774.305.000,00	Rp 0,00	Rp 0,00

## 12. Halaman Bendahara Validasi Data

SISTEM INFORMASI MONITORING

The screenshot shows a dashboard for validating financial data. It features a header with a logo and a welcome message "Selamat Datang, bendahara!". Below it is a section titled "Monitoring Anggaran". The main area has two main sections: "Kegiatan Realisasi" and "List Data".

**Kegiatan Realisasi**

No	Nama Kegiatan	Nominal	Berkas	Status	Aksi
1	Belanja ATK	700000	Download Berkas	Mengunggah verifikasi Verifikator	Approve Berkas

**List Data**

Pembinaan Administrasi Hukum dan KLN Pagu Rp 12.041.208.000,00 Sisa Dana Rp 0,00

Kode	Uraian	Pagu	Realisasi	Sisa Dana
▼ 957	Layanan Hukum	Rp 5.661.407.000,00	Rp 700.000,00	Rp 5.660.707.000,00
▼ 001	Taripa Sub Output	Rp 5.631.407.000,00	Rp 700.000,00	Rp 5.630.707.000,00
► 051	Koordinasi Penyusutan Produk Hukum	Rp 1.414.168.000,00	Rp 700.000,00	Rp 1.413.468.000,00
► A	Penyusutan Peraturan Menteri	Rp 414.168.000,00	Rp 700.000,00	Rp 413.468.000,00
521211	belanja bahan	Rp 167.000.000,00	Rp 700.000,00	Rp 166.300.000,00
521219	Belanja Barang Non Operasional Lainnya	Rp 98.000.000,00	Rp 0,00	Rp 0,00
522151	Behan Jasa Profesi	Rp 149.168.000,00	Rp 0,00	Rp 0,00
► B	Penyusutan Keputusan Menteri	Rp 0,00	Rp 0,00	Rp 0,00
► C	Penyusutan Rancangan Peraturan Undang-Undangan dan	Rp 400.000.000,00	Rp 0,00	Rp 0,00
► D	PENYUSUNAN RANCANGAN PERATURAN PRESIDEN, KEPUTUSAN	Rp 0,00	Rp 0,00	Rp 0,00

### 13. Halaman Operator Upload SPJ



## LAMPIRAN 4

### CONTOH SOURCE CODE

#### **Template.php**

DOCTYPE html>

<html>

<head lang="en">

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">

<meta http-equiv="x-ua-compatible"

content="ie=edge">

<title>{{ env('APP\_NAME') }} Admin

Dashboard</title>

<link href="img/favicon.144x144.png" rel="apple-

touch-icon" type="image/png" sizes="144x144">

<link href="img/favicon.114x114.png" rel="apple-

touch-icon" type="image/png" sizes="114x114">

<link href="img/favicon.72x72.png" rel="apple-

touch-icon" type="image/png" sizes="72x72">

<link href="img/favicon.57x57.png" rel="apple-

touch-icon" type="image/png">

<link href="img/favicon.png" rel="icon"

type="image/png">

<link href="img/favicon.ico" rel="shortcut icon">

<!-- HTML5 shim and Respond.js for IE8

support of HTML5 elements and media

queries -->

<!--[if lt IE 9]>

<script

src="https://oss.maxcdn.com/html5shiv/3.7.2/

html5shiv.min.js"></script>

<script

src="https://oss.maxcdn.com/respond/1.4.2/res-

pond.min.js"></script>

<![endif]-->

<link rel="stylesheet"

href="{{ asset\_back('css/lib/font-"

awesome/font-awesome.min.css') }}">

<link rel="stylesheet"

href="{{ asset\_back('css/lib/bootstrap/bootstra-"

p.min.css') }}">

<link rel="stylesheet"

href="{{ asset\_back('css/lib/bootstrap-"

sweetalert/sweetalert.css') }}">

<link rel="stylesheet"

href="{{ asset\_back('css/separate/vendor/select-"

2.min.css') }}">

<link rel="stylesheet"

href="{{ asset\_back('css/main.css') }}">

<link

href="{{ asset\_back('css/treetable/jquery.treeta-"

ble.css') }}" rel="stylesheet" />

</head>

<body class="with-side-menu control-panel"

control-panel-compact">

```

        ]
        'pegawai_id' => $pegawai_id,
        'level_id'    => $level_id
            );
            if($status){
                $pgw = $this-
>m_global->get_field_by_id('m_pegawai',
'status', 'pegawai_id', $pegawai_id);
                if($pgw){
                    $this-
>session->set_userdata($data);
                    $this-
>m_login->set_lastlogin();
                    get_redirecting('home');
                }
            } else{
                $this->session-
>set_flashdata('message', 'Pegawai tidak aktif');
                get_redirecting('login');
            }
        } else{
            $this->session-
>set_flashdata('message', 'User tidak aktif');
            get_redirecting('login');
        }
    } else {
        $this->session-
>set_flashdata('message', 'Username atau password salah.');
        get_redirecting('login');
    }
}

function process_logout() {
    $this->session-
>sess_destroy();
    get_redirecting('login');
}

```

**login.blade.php**

```

@extends('templatelogin')

<link rel="stylesheet"
 href="{!! asset_back('css/lib/lobipanel/lobi
panel.min.css') !!}">

```

## Index.php

```

@php
$list = [
    'v1' => 'Stop Point Public Transport',
    'v2' => 'Traffic Light',
    'v3' => 'U-Turn'
]

```

```
<link rel="stylesheet"
      href="{{ asset_back('css/separate/vendor/lobipanel.min.css') }}>
<link rel="stylesheet"
      href="{{ asset_back('css/lib/jqueryui/jquery-ui.min.css') }}>
<link rel="stylesheet"
      href="{{ asset_back('css/separate/pages/widget.s.min.css') }}>
<link rel="stylesheet"
      href="{{ asset_back('css/lib/fontawesome/fontawesome/font-awesome.min.css') }}>
<link rel="stylesheet"
      href="{{ asset_back('css/lib/bootstrap/bootstrap.min.css') }}>
<link rel="stylesheet"
      href="{{ asset_back('css/lib/bootstrap-sweetalert/sweetalert.css') }}>
<div class="page-content">
    <div class="container-fluid">
        <label>
            <h1>LOGIN</h1>
        </label>
        <div class="row" >
            <div class="col-md-6">
                <form role="form"
                      method="POST">
                    {{ csrf_field() }}
                    <div class="form-group">

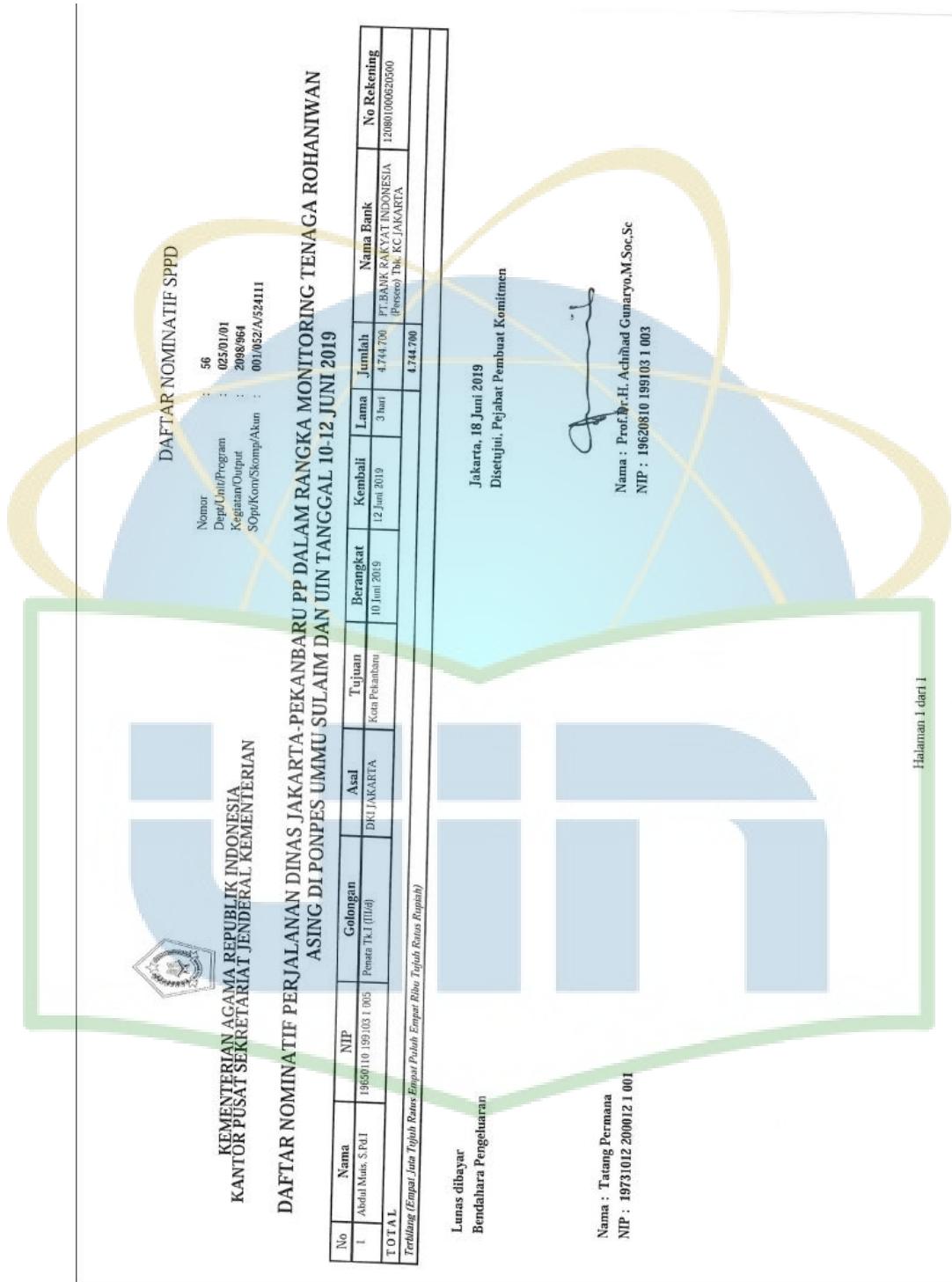
```

The diagram illustrates the flow of data from the left code block to the right code block. The flow is represented by three overlapping colored regions: yellow, blue, and green. The yellow region covers the first two lines of the left code. The blue region covers the next four lines. The green region covers the last two lines of the left code and continues through the right code block, ending with a large green box around the final closing tag. Arrows point from each line of the left code to its corresponding position in the right code, indicating the mapping of variables and structures.

```
                        <label for="exampleInputEmail1">Username</label>
                        <input type="text"
                              class="form-control" name="username"
                              id="exampleInputEmail" placeholder="Enter email">
                    </div>
                    <div class="form-group">
                        <label for="exampleInputPassword1">Password</label>
                        <input type="password"
                              class="form-control" name="password"
                              id="exampleInputPassword" placeholder="Password">
                    </div>
                    <button type="submit" class="btn btn-default">Masuk</button>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
```

## **LAMPIRAN 5**

### ***FORM-FORM PENDUKUNG PENELITIAN***



**SURAT PERNYATAAN TANGGUNG JAWAB BELANJA**  
NOMOR : 188/416289/05/HKLN/2019

1. Kode Satuan Kerja : 416289  
 2. Nama Satuan Kerja : KANTOR PUSAT SEKRETARIAT JENDERAL KEMENTERIAN AGAMA JAKARTA  
 3. Tanggal / No. DIPA : 05 Desember 2017 / SP DIPA- 025.01.1.416289/2018  
 4. Klasifikasi Anggaran : 01 / 03 / 01 / 52

Yang bertandatangan di bawah ini atas nama Kuasa Pengguna Anggaran Satuan Kerja KANTOR PUSAT SEKRETARIAT JENDERAL KEMENTERIAN AGAMA JAKARTA menyatakan bahwa saya bertanggungjawab secara formal dan material dan kebenaran perhitungan pemungutan pajak atas segala pembayaran tagihan yang telah kami perintahkan dalam SPM ini dengan perincian sebagai berikut:

No	Kode Akun	Penerima	Uraian	Jumlah	Pajak yang dipungut	
					PPN	PPH
1	524111	Abdul Muis, S.Pd.I	perjadin dari KOTA JAKARTA PUSAT menuju KOTA PEKANBARU, selama 3 hari, pada tanggal 10 s.d 12 Juni 2019. 2098.964 001.052.A	4.744.700	0	0
			Jumlah	4.744.700	0	0

Bukti-bukti pengeluaran anggaran dan asli setoran pajak (SSP/BPN) tersebut di atas disimpan oleh Pengguna Anggaran/Kuasa Pengguna Anggaran untuk kelengkapan administrasi dan pemeriksaan aparat pengawasan fungsional.

Demikian Surat Pernyataan ini dibuat dengan sebenarnya.

Jakarta, 18 Juni 2019  
Pejabat Pembuat Komitmen

Prof.Dr.H. Achmad Gunaryo,M.Soc.Sc  
NIP. 19620810 199103 1 003

Pembayaran Langsung (LS) Pihak Ketiga

## SURAT PERMINTAAN PEMBAYARAN

Tanggal : 19-06-2019 Nomor 01785/HKLN/TERLAMPIR/188  
 Sifat Pembayaran : (4) Pembayaran Langsung (LS) &  
 Jenis Pembayaran : (1) Pengeluaran Anggaran

1 Departemen/Lembaga	KEMENTERIAN AGAMA (025)	7 Kegiatan	Pembinaan Administrasi Hukum dan KLN
2 Unit Organisasi	SEKRETARIAT JENDERAL (01)	8 Kode Kegiatan	2098
3 Kantor/Satker	KANTOR PUSAT SEKRETARIAT JENDERAL KEMENTERIAN AGAMA JAKARTA (416289)	9 Kode Fungsi,	09.90.01
4 Lokasi	DKI JAKARTA (01)	S Fungsi, Program	
5 Tempat	KOTA JAKARTA PUSAT (51)	10 Kewenangan Pelaksanaan	(KP) Kantor Pusat
6 Alamat	JALAN LAPANGAN BANTENG BARAT 3-4 JAKARTA PUSAT		

Kepada

Yth. Pejabat Penanda Tangan Surat Permintaan Membayar  
 KANTOR PUSAT SEKRETARIAT JENDERAL KEMENTERIAN AGAMA JAKARTA  
 di KOTA JAKARTA PUSAT

Berdasarkan DIPA Nomor : DIPA-025.01.1.416289/2019, 05-12-2018, bersama ini kami ajukan permintaan pembayaran sebagai berikut :

1. Jumlah pembayaran yang dimintakan : Rp. 4.744.700

( empat juta tujuh ratus empat puluh empat ribu tujuh ratus rupiah )

2. Untuk keperluan

: Pembayaran Belanja Barang Sesuai Surat Tugas Nomor 96.144/B.V.4/KP.01.2/2019 Tanggal 21 Mei 2019 dan SPP Nomor 01785/HKLN/TERLAMPIR/188 Tanggal 19 Juni 2019

3. Jenis Belanja

: BELANJA BARANG

4. Atas Nama

: KANTOR PUSAT SEKRETARIAT JENDERAL KEMENTERIAN AGAMA

5. Alamat

: Jl. Lapangan Banteng Barat No. 3 - 4 Jakarta

6. Mempunyai rekening

: BANK RAKYAT INDONESIA SYARIAH

nomor rekening : TERLAMPIR

7. Nomor dan Tanggal SPK Kontrak

: , -

8. Nilai SPK/Kontrak

: Rp. 0

9. Dengan penjelasan

:

No	I KEGIATAN/OUTPUT/MAK (AKUN 6 DIGIT) BERSANGKUTAN II SEMUA KODE KEGIATAN DALAM DIPA	PAGU DALAM DIPA/SKPA (Rp.)	SPP/SPM S.D. YANG LALU (Rp.)	SPP INI (Rp.)	JUMLAH S.D. SPP INI (Rp.)	SISA DANA (Rp.)
	2098 . 964 . 524111	249.365.000	49.651.679	4.744.700	54.406.379	194.958.621
	Jumlah I	249.365.000	49.651.679	4.744.700	54.406.379	194.958.621
II	SEMUA KEGIATAN					
	2098	12.041.208.000	1.050.187.912	4.744.700	1.054.932.612	10.986.275.388
	2099	17.873.693.000	0	0	0	17.873.693.000
	2100	90.902.986.000	0	0	0	90.902.986.000
	2101	13.281.507.000	0	0	0	13.281.507.000
	2102	19.747.088.000	0	0	0	19.747.088.000
	2103	75.346.871.000	0	0	0	75.346.871.000
	2106	20.657.126.000	0	0	0	20.657.126.000
	2130	3.230.000.000	0	0	0	3.230.000.000
	5620	17.227.589.000	0	0	0	17.227.589.000
	5621	6.675.000.000	0	0	0	6.675.000.000
	Jumlah II	276.983.068.000	1.050.187.912	4.744.700	1.054.932.612	275.928.135.388
	UANG PERSEDIAAN					
Lampiran	0 Lembar Pendukung ..... Lembar	0	Surat Buku Pengeluaran ..... Lembar	0	STS..... Lembar	

Diterima oleh pengisi SPP / Penerbit SPM  
 KANTOR PUSAT SEKRETARIAT JENDERAL KEMENTERIAN AGAMA  
 pada tanggal

KOTA JAKARTA PUSAT, Tanggal seperti di atas  
 Pejabat Pembuat Komitmen  
 KANTOR PUSAT SEKRETARIAT JENDERAL KEMENTERIAN AGAMA

Mohamad Ali Irfan  
 NIP. 196611131994031001

Prof. Dr. H. Achmad Gunaryo, M.Soc.Sc  
 NIP. 195208101991031003

**LAMPIRAN 6**  
**SURAT-SURAT PENDUKUNG PENELITIAN**





**KEMENTERIAN AGAMA  
UNIVERSITAS ISLAM NEGERI (UIN)  
SYARIF HIDAYATULLAH JAKARTA  
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Ir. H. Juanda No. 96 Ciputat 15412 Indonesia  
Telp.: (62-21) 7493606, 7493547, Fax.: (62-21) 7493315

Email : fst@uinjkt.ac.id  
Website : fst.uinjkt.ac.id

Nomor : B- 3545 / F9 / KM. 01 / 05 / 2019  
Lampiran : -  
Perihal : Permohonan Data/Riset

Jakarta, 29 Mei 2019

Kepada Yth.

**Kepala Biro Hukum dan Kerja Sama Luar Negeri  
Setjen Kementerian Agama  
SEKRETARIAT JENDERAL KEMENTERIAN AGAMA  
REPUBLIK INDONESIA  
Jalan Lapangan Banteng Barat No 3-4**

*Assalamualaikum Wr. Wb*

Dengan hormat kami sampaikan bahwa :

Nama	: Agung Ramadhan
NIM	: 1112093000071
Jurusan/Semester	: Sistem Informasi/XIV (Empat Belas)
Tahun Akademik	: Semester Ganjil 2018/2019
Program	: S-1
Alamat	: Jl Srengseng Bambu 2 Kembangan Jakarta Barat
Telp.	: 089674280060

Adalah benar mahasiswa Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta dan bermaksud melakukan penelitian/riset data di instansi yang Bapak/Ibu pimpin, yang sedang dalam penyelesaian skripsi dengan judul skripsi:

**"RANCANG BANGUNS SISTEM MONITORING ANGGARAN KEUANGAN BERBASIS WEB PADA BIRO HUKUM DAN KERJA SAMA LUAR NEGERI (Study Kasus: Sekretariat Jenderal Kementerian Agama Republik Indonesia)"**

Untuk itu, kami mohon kesediaannya untuk memberikan kesempatan kepada mahasiswa tersebut dalam melaksanakan penelitian/riset data selama 3 bulan terhitung: 9 September s/d 8 Desember 2019 di instansi/perusahaan yang Bapak/Ibu pimpin.

Demikian, atas perhatian dan kerjasamanya kami ucapan terima kasih.

*Wassalamualaikum Wr. Wb.*



Nashrul Hakiem, S. Si., M.T., Ph.D  
NIP. 19710608 200501 1 005

Tembusan :  
Dekan (sebagai laporan)



# KEMENTERIAN AGAMA REPUBLIK INDONESIA

## SEKRETARIAT JENDERAL

Jalan Lapangan Banteng Barat No. 3 – 4 Jakarta  
Telpo : 3811244, 3811642, 3811654, 3811658, 3811679, 381779, 3812216  
(Hunting) 34833004 – 34833005

Nomor : B.V/1/Kp.01.2/5/2019  
Lampiran : -  
Hal : Memberikan Izin Pengambilan Data/Riset

Jakarta, 30 Mei 2019

Kepada Yth.  
Dekan Fakultas Sains dan Teknologi  
UIN Syarif Hidayatullah Jakarta  
Di tempat

*Assalamualaikum. Wr.Wb.*

Sehubung dengan adanya surat dari Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, nomor B- 3545 / F9/ KM.01 / 05 / 2019, tanggal 29 Mei 2019, perihal permohonan Data/Riset untuk:

Nama : Agung Ramadhan  
NIM : 1112093000071  
Fak/Jur : Fakultas Sains dan Teknologi/Sistem Insformasi  
Judul : Rancang Bangun Sisem Monitoring Anggaran Keuangan Berbasis Web Pada Biro Hukum dan Kerja Sama Luar Negeri. (Studi kasus: Sekretariat Jenderal Kementerian Agama Republik Indonesia)

Pada dasarnya kami tidak keberatan untuk memberikan izin pengambilan data yang berhubungan dengan riset yang di lakukan oleh mahasiswa yang bersangkutan.

Demikian izin ini diberikan untuk digunakan sebagai mestinya, atas perhatian dan kerjasamanya kami ucapkan terimakasih.

*Wassalamualaikum Wr.Wb.*



Dr. H. Achmad Gunaryo, M.Soc.Sc  
NIP. 1956208101991031003



**KEMENTERIAN AGAMA  
UNIVERSITAS ISLAM NEGERI (UIN)  
SYARIF HIDAYATULLAH JAKARTA  
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Ir. H. Juanda No. 95 Ciputat 15412 Indonesia  
Telp.: (62-21) 7493606, 7493547, Fax.: (62-21) 7493315

Email : fst@uinjkt.ac.id  
Website : fst.uinjkt.ac.id

Nomor : B- 534 / F9 / KM.01 / 02 / 2019  
Lampiran : -  
Perihal : Bimbingan Skripsi

Jakarta, 04 Februari 2019

**Kepada Yth.**

1. Nur Aeni Hidayah, MMSI
2. Yusuf Durachman, M.Sc, MIT

*Assalamu'alaikum Wr. Wb.*

Dengan ini diharapkan kesediaan Saudara untuk menjadi pembimbing I/II/ (Materi/Teknis)\* penulisan skripsi mahasiswa:

Nama : Agung Ramadhan  
NIM : 111209300071  
Program Studi : Sistem Informasi  
Judul Skripsi : **"RANCANG BANGUN SISTEM MONITORING ANGGARAN KEUANGAN BIRO HUKUM DAN KERJASAMA LUAR NEGERI (Studi Kasus Setjen Kementerian Agama Republik Indonesia)"**

Judul tersebut telah disetujui oleh Program Studi bersangkutan pada tanggal 26 September 2016 dengan outline, abstraksi dan daftar pustaka terlampir. Bimbingan skripsi ini diharapkan selesai dalam waktu 6 (enam) bulan setelah ditandatanganinya surat penunjukan pembimbing skripsi.

Apabila terjadi perubahan terkait dengan skripsi tersebut selama proses pembimbingan, harap segera melaporkan kepada Program Studi bersangkutan.

Demikian atas kesediaan Saudara, kami ucapan terima kasih.

*Wassalamu'alaikum Wr.Wb.*



Tembusan:  
Dekan (sebagai laporan)