

**PEMBANGUNAN APLIKASI *MEETING ARRANGER*  
DI LINGKUNGAN BADAN PERENCANAAN  
PEMBANGUNAN DAERAH  
PROVINSI JAWA BARAT**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan  
Jenjang Strata Satu (S1)  
Pada Program Studi Teknik Informatika**

Oleh :  
**GUN GUN SOLIHIN  
361841003**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI  
2022**

**LEMBAR PENGESAHAN**

**PEMBANGUNAN APLIKASI *MEETING ARRANGER*  
DI LINGKUNGAN BADAN PERENCANAAN  
PEMBANGUNAN DAERAH  
PROVINSI JAWA BARAT**

Oleh :  
Gun Gun Solihin  
361841003

Tugas akhir ini telah diterima dan disahkan untuk  
memenuhi persyaratan mencapai gelar

**SARJANA TEKNIK INFORMATIKA**

Pada  
**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI**

Bandung, Juni 2022  
Disahkan oleh

Ketua Program Studi,



Chalifa Chazar, S.T. MT  
NIDN: 0421098704

Dosen Pembimbing,,



Chalifa Chazar, S.T. MT  
NIDN: 0421098704

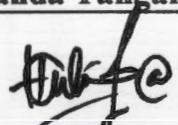
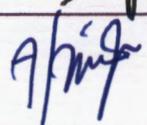
**LEMBAR PERSETUJUAN REVISI**

**PEMBANGUNAN APLIKASI *MEETING ARRANGER*  
DI LINGKUNGAN BADAN PERENCANAAN  
PEMBANGUNAN DAERAH  
PROVINSI JAWA BARAT**

Oleh :  
Gun Gun Solihin  
361841003

Telah melakukan sidang tugas akhir dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang tugas akhir.

Bandung, Juni 2022  
Menyetujui,

No	Nama Dosen	Keterangan	Tanda Tangan
1	Chalifa Chazar, S.T. MT	Pembimbing	
2	Moch. Ali Ramdhani, S.T., M.Kom.	Penguji 1	
3	Yudhi W. Arthana R., S.T., M.Kom.	Penguji 2	

Mengetahui,  
Ketua Program Studi Teknik Informatika,

  
  
Chalifa Chazar, S.T. MT  
NIDN: 042 1098704

## SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa :

- (1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- (2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah diluar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, Juni 2022

Yang membuat pernyataan,



Gun Gun Solihin

NIM. 361841003

## ABSTRAK

# PEMBANGUNAN APLIKASI *MEETING ARRANGER* DI LINGKUNGAN BADAN PERENCANAAN PEMBANGUNAN DAERAH PROVINSI JAWA BARAT

Oleh

Gun Gun Solihin  
NIM 361841003

Banyaknya agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat mengakibatkan kesulitan dalam membagikan disposisi atau tugas kepada pegawai dan terjadi tumpang tindihnya penugasan, serta kekeliruan dan kesalahan dalam pelaksanaan tugas baik internal maupun eksternal. Penelitian ini bertujuan untuk membangun suatu aplikasi berbasis *website* yang berfungsi dalam mengelola, mendistribusikan agenda rapat dan kegiatan untuk dapat diinformasikan pada layar disetiap ruangan maupun diakses pada *smartphone*, laptop maupun komputer melalui *browser* sehingga dapat dipantau secara langsung atau *realtime* oleh pegawai yang ditugaskan oleh pimpinan.

Hal inilah yang mendorong penulis untuk melakukan penulisan ilmiah dengan judul “Pembangunan Aplikasi *Meeting Arranger* di lingkungan Bappeda Provinsi Jawa Barat”. Metode penelitian yang digunakan oleh penulis dalam menyelesaikan skripsi ini adalah Metode pengembangan sistem *waterfall*, *tools* yang digunakan adalah bahasa pemrograman PHP. Pembangunan aplikasi *meeting arranger* diharapkan dapat membantu dalam pengelolaan, pendistribusian dan penyebaran informasi agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat.

**Kata Kunci** : Bappeda Provinsi Jawa Barat, *Meeting Arranger*, *Waterfall*, *Website*

## ***ABSTRACT***

# **PEMBANGUNAN APLIKASI *MEETING ARRANGER* DI LINGKUNGAN BADAN PERENCANAAN PEMBANGUNAN DAERAH PROVINSI JAWA BARAT**

Oleh

Gun Gun Solihin  
NIM 361841003

*The large number of meeting agendas and activities at the Bappeda of West Java Province caused difficulties in distributing dispositions or tasks to employees, as well as overlapping assignments, mistakes, and errors in carrying out tasks both internally and externally. This research aims to create a website-based application that manages and distributes meeting agendas and activities to be displayed on screens in every room or accessed via browsers on smartphones, laptops, or computers so that they can be monitored directly or in real time by employees assigned by the leadership.*

*This is what prompted the author to write scientifically under the title “Pembangunan Aplikasi Meeting Arranger di lingkungan Bappeda Provinsi Jawa Barat”. The author used the waterfall system development method to finish this research, and the PHP programming language was used as a tool. The meeting arranger application is expected to aid in the management, distribution, and dissemination of information on meeting agendas and activities in West Java Province's Bappeda.*

*Keywords: Bappeda of West Java Province, Meeting Arranger, Waterfall, Website*

## KATA PENGANTAR

Dengan mengucapkan Alhamdulillah sebagai wujud syukur kepada Allah SWT, yang senantiasa memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan penelitian tugas akhir ini dengan baik dan tepat waktu.

Tugas akhir ini, berjudul *PEMBANGUNAN APLIKASI MEETING ARRANGER* DI LINGKUNGAN BADAN PERENCANAAN PEMBANGUNAN DAERAH PROVINSI JAWA BARAT, disusun untuk melengkapi tahapan akhir studi yang dipelajari pada Perguruan Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Tugas akhir ini berisi mengenai pembuatan aplikasi yang dapat membantu pengelolaan, pendistribusian serta penyebaran informasi tentang agenda rapat di Badan Perencanaan Pembangunan Daerah Provinsi Jawa Barat.

Dengan segala keterbatasan tentunya diharapkan tugas akhir ini dapat bermanfaat bagi berbagai pihak, khususnya bagi penulis sendiri

Bandung, Juni 2022

Penulis,



Gun Gun Solihin

NIM. 361841003

## UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Allah SWT, karena atas limpahan rahmat dan karunia-Nya tugas akhir ini dapat terselesaikan dengan baik. Skripsi ini saya persembahkan untuk kedua orang tua yang senantiasa mendoakan dan memberikan bantuan moril maupun materil. Tak lupa penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. Chairrudin, Ir., M.M., M.T. selaku ketua STMIK Indonesia Mandiri;
2. Ibu Chalifa Chazar S.T., M.T. selaku Ketua Program Studi Teknik Informatika pada STMIK Indonesia Mandiri sekaligus dosen pembimbing yang telah banyak membantu dan memberikan bimbingan selama pembuatan tugas akhir ini;
3. Bapak Moch. Ali Ramdhani, S.T., M.Kom. dan Bapak Yudhi W. Arthana R., S.T., M.Kom. selaku Penguji pada Sidang Tugas Akhir ini;
4. Seluruh Dosen, Staff dan Karyawan STMIK Indonesia Mandiri yang telah mendidik dan membantu dalam memberikan informasi serta motivasi dalam proses studi maupun tugas akhir berlangsung;
5. Untuk Windy yang telah setia mendampingi dalam penyelesaian tugas akhir ini; dan
6. Terakhir kepada semua pihak yang tidak bisa penulis sebutkan satu persatu atas semua bantuannya penulis mengucapkan banyak terima kasih.

Semoga Allah SWT memberikan balasan yang berlipat ganda kepada semuanya. Demi perbaikan selanjutnya, saran dan kritik yang membangun akan penulis terima dengan senang hati. Akhirnya, hanya kepada Allah SWT penulis serahkan segalanya, semoga dapat bermanfaat khususnya bagi penulis umumnya bagi kita semua.

Bandung,  
Penulis,



Gun Gun Solihin  
NIM. 361841003

## DAFTAR ISI

<b>SURAT PERNYATAAN .....</b>	<b>i</b>
<b>ABSTRAK .....</b>	<b>ii</b>
<b>ABSTRACT .....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>UCAPAN TERIMA KASIH .....</b>	<b>v</b>
<b>DAFTAR ISI.....</b>	<b>vi</b>
<b>DAFTAR TABEL .....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>ix</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah .....	9
1.3 Tujuan Penulisan .....	9
1.4 Batasan Masalah .....	10
1.5 Metode Penelitian .....	10
1.6 Sistematika Penulisan .....	12
<b>BAB II LANDASAN TEORI .....</b>	<b>14</b>
2.1 Penelitian Terkait.....	14
2.2 Pengertian <i>Waterfall</i> .....	18
2.3 Pengertian Aplikasi .....	20
2.4 Pengertian Pengelolaan .....	22
2.5 Pengertian Pendistribusian .....	22
2.6 <i>Unified Modeling Language (UML)</i> .....	24
2.6.1 Diagram UML .....	24
2.6.2 <i>Use Case Diagram</i> .....	26
2.6.3 <i>Sequence Diagram</i> .....	30
2.7 <i>MySQL</i> .....	33
2.8 <i>Hypertext Preprocessor (PHP)</i> .....	34
2.9 Apache.....	34
2.10 <i>Open Source</i> .....	34
2.11 Pengujian Perangkat Lunak.....	35
2.11.1 Definisi .....	35
2.11.2 Tujuan Pengujian aplikasi .....	35
2.11.3 <i>Black box Testing</i> .....	36
<b>BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM .....</b>	<b>37</b>
3.1 Analisis (Observasi Lapangan).....	37
3.1.1 Analisis Sistem Eksisting .....	40
3.1.2 Analisis Permasalahan .....	40
3.1.3 Analisis Sistem Usulan.....	42
3.2 Desain .....	50
3.2.1 Perancangan Program.....	50

3.2.2	Definisi Aktor .....	50
3.2.3	Langkah Aplikasi .....	51
3.2.4	<i>Use Case Diagram Meeting Arranger</i> .....	54
3.2.5	Skenario <i>Use Case</i> .....	55
3.2.6	<i>Activity Diagram</i> .....	68
3.2.7	<i>Sequence Diagram</i> .....	76
3.2.8	<i>Class Diagram</i> .....	83
3.2.9	ERD ( <i>Entity Relationship Diagram</i> ) .....	84
3.2.10	Kebutuhan Perangkat Lunak dan Perangkat Keras .....	85
3.3	Perancangan Antarmuka .....	86
3.3.1	Perancangan Antarmuka Halaman Utama .....	87
3.3.2	Perancangan Antarmuka Halaman <i>Login</i> .....	87
3.3.3	Perancangan Antarmuka Halaman Admin .....	88
3.3.4	Perancangan Antarmuka Halaman Data Kegiatan .....	89
3.3.5	Perancangan Antarmuka Halaman Formulir Kegiatan .....	90
<b>BAB IV IMPLEMENTASI DAN UJI COBA .....</b>		<b>92</b>
4.1	Implementasi Aplikasi .....	92
4.1.1	<i>Physical Database</i> .....	92
4.1.2	Implementasi Perangkat Lunak .....	93
4.1.3	Implementasi Perangkat Keras .....	94
4.2	Implementasi Antarmuka .....	95
4.2.1	Implementasi Antarmuka Halaman Utama .....	95
4.2.2	Implementasi Antarmuka Halaman <i>Login</i> .....	96
4.2.3	Implementasi Antarmuka Halaman Admin .....	96
4.2.4	Implementasi Antarmuka Halaman Data Kegiatan .....	97
4.2.5	Implementasi Antarmuka Halaman Formulir Kegiatan .....	98
4.2.6	Implementasi Antarmuka Penginputan Agenda Rapat .....	99
4.2.7	Implementasi Antarmuka melihat Agenda Rapat yang sudah tampil di aplikasi melalui Web Browser .....	100
4.3	Uji Coba Aplikasi .....	100
4.3.1	<i>User Acceptance Test Use Case Login</i> .....	101
4.3.2	<i>User Acceptance Test Use Case Melihat Kegiatan</i> .....	102
4.3.3	<i>User Acceptance Test Use Case Tambah Kegiatan</i> .....	103
4.3.4	<i>User Acceptance Test Use Case Ubah Kegiatan</i> .....	107
4.3.5	<i>User Acceptance Test Use Case Hapus Kegiatan</i> .....	111
4.3.6	<i>User Acceptance Test Use Case Filter Kegiatan</i> .....	114
4.3.7	<i>User Acceptance Test Use Case Logout</i> .....	116
<b>BAB V PENUTUP .....</b>		<b>118</b>
5.1	Kesimpulan .....	118
5.2	Saran .....	118
<b>DAFTAR PUSTAKA .....</b>		<b>120</b>
<b>LAMPIRAN .....</b>		<b>122</b>

## DAFTAR TABEL

TABEL 2.1 Tabel Referensi .....	15
TABEL 2.2 Simbol pada <i>Use Case Diagram</i> .....	26
TABEL 2.3 Simbol pada Diagram <i>Sequence</i> .....	31
TABEL 3.1 Formulir Wawancara.....	38
TABEL 3.2 Deskripsi Aktor .....	50
TABEL 3.3 <i>Use Case Scenario</i> Melihat Kegiatan.....	55
TABEL 3.4 <i>Use Case Scenario</i> login .....	56
TABEL 3.5 <i>Use Case Scenario</i> Tambah Kegiatan.....	59
TABEL 3.6 <i>Use Case Scenario</i> Ubah Kegiatan .....	61
TABEL 3.7 <i>Use Case Scenario</i> Hapus Kegiatan.....	64
TABEL 3.8 <i>Use Case Scenario</i> Filter Kegiatan .....	66
TABEL 3.9 <i>Use Case Scenario</i> Cari Kegiatan .....	67
TABEL 3.10 <i>Use Case Scenario</i> Logout Kegiatan.....	68
TABEL 4.1 <i>User Acceptance Test Use Case Login</i> .....	101
TABEL 4.2 <i>User Acceptance Test Use Case</i> Melihat Kegiatan.....	103
TABEL 4.3 <i>User Acceptance Test Use Case</i> Tambah Kegiatan .....	104
TABEL 4.4 <i>User Acceptance Test Use Case</i> Ubah Kegiatan.....	108
TABEL 4.5 <i>User Acceptance Test Use Case</i> Hapus Kegiatan .....	112
TABEL 4.6 <i>User Acceptance Test Use Case</i> Filter Kegiatan.....	114
TABEL 4.7 <i>User Acceptance Test Use Case Logout</i> .....	116

## DAFTAR GAMBAR

GAMBAR 2.1 Ilustrasi Model <i>Waterfall</i> .....	19
GAMBAR 2.2 Diagram UML .....	25
GAMBAR 3.1 <i>Flowchart</i> SOP Surat Masuk di Bappeda Provinsi Jawa Barat....	40
GAMBAR 3.2 Sistem Usulan Halaman Utama.....	43
GAMBAR 3.3 Sistem Usulan Halaman <i>Login</i> .....	44
GAMBAR 3.4 Sistem Usulan Tambah Kegiatan .....	45
GAMBAR 3.5 Sistem Usulan Ubah Kegiatan.....	46
GAMBAR 3.6 Sistem Usulan Hapus Kegiatan .....	47
GAMBAR 3.7 Sistem Usulan Filter Data.....	48
GAMBAR 3.8 Sistem Usulan <i>logout</i> .....	49
GAMBAR 3.9 <i>Use Case Diagram</i> .....	54
GAMBAR 3.10 <i>Activity Diagram</i> Halaman Utama.....	69
GAMBAR 3.11 <i>Activity Diagram</i> Login .....	70
GAMBAR 3.12 <i>Activity Diagram</i> Tambah Kegiatan .....	71
GAMBAR 3.13 <i>Activity Diagram</i> Ubah Kegiatan.....	72
GAMBAR 3.14 <i>Activity Diagram</i> Hapus Kegiatan .....	73
GAMBAR 3.15 <i>Activity Diagram</i> Filter Kegiatan.....	74
GAMBAR 3.16 <i>Activity Diagram Logout</i> .....	75
GAMBAR 3.17 <i>Sequence Diagram</i> Melihat Kegiatan .....	77
GAMBAR 3.18 <i>Sequence Diagram Login</i> .....	78
GAMBAR 3.19 <i>Sequence Diagram</i> Tambah Kegiatan.....	79
GAMBAR 3.20 <i>Sequence Diagram</i> Ubah Kegiatan .....	80
GAMBAR 3.21 <i>Sequence Diagram</i> Hapus Kegiatan .....	81
GAMBAR 3.22 <i>Sequence Diagram</i> Filter Kegiatan .....	82
GAMBAR 3.23 <i>Sequence Diagram Logout</i> .....	83
GAMBAR 3.24 <i>Class Diagram</i> .....	84
GAMBAR 3.25 <i>Entity Relationship Diagram</i> .....	85
GAMBAR 3.26 Perancangan Antarmuka Halaman Utama.....	87
GAMBAR 3.27 Perancangan Antarmuka Halaman Login.....	88
GAMBAR 3.28 Perancangan Antarmuka Halaman Admin .....	89
GAMBAR 3.29 Perancangan Antarmuka Halaman Data Kegiatan .....	90
GAMBAR 3.30 Perancangan Antarmuka Halaman Formulir Kegiatan.....	91
GAMBAR 4.1 <i>Physical Database</i> .....	93
GAMBAR 4.2 Implementasi Halaman Utama .....	95
GAMBAR 4.3 Implementasi Antarmuka Halaman Login .....	96
GAMBAR 4.4 Implementasi Antarmuka Halaman Admin.....	97
GAMBAR 4.5 Implementasi Halaman Data Kegiatan .....	98
GAMBAR 4.6 Implementasi Antarmuka Halaman Formulir Kegiatan .....	99
GAMBAR 4.7 Implementasi Antarmuka Penginputan Agenda Rapat.....	99
GAMBAR 4.8 Implementasi Antarmuka melihat Agenda Rapat yang sudah tampil di aplikasi melalui <i>web browser</i> .....	100

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Badan Perencanaan Pembangunan Daerah Provinsi Jawa Barat yang kemudian disebut sebagai BAPPEDA Provinsi Jawa Barat adalah sebuah instansi pemerintahan dengan mempunyai tugas pokok melaksanakan fungsi penunjang urusan pemerintahan bidang perencanaan, meliputi perencanaan, pengendalian dan evaluasi pembangunan daerah, pemerintahan dan pembangunan manusia, perekonomian dan sumber daya alam serta infrastruktur dan kewilayahan yang menjadi kewenangan daerah provinsi, menyelenggarakan tugas dekonsentrasi dan melaksanakan tugas pembantuan sesuai bidang tugasnya berdasarkan ketentuan perundang-undangan.

Dalam menyelenggarakan tugas pokok di atas, Bappeda mempunyai fungsi, yaitu:

- a. Penyelenggaraan perumusan kebijakan teknis bidang perencanaan yang menjadi kewenangan Daerah Provinsi.
- b. Penyelenggaraan perencanaan yang menjadi kewenangan Daerah Provinsi.
- c. Penyelenggaraan pengendalian evaluasi perencanaan pembangunan serta pelaporan pelaksanaan tugas dukungan teknis Badan,
- d. Penyelenggaraan administrasi Badan,

- e. Penyelenggaraan evaluasi dan pelaporan Bidang, dan
- f. Penyelenggaraan fungsi lain sesuai tugas pokok dan fungsinya.

Berdasarkan perhitungan nilai variabel baik umum maupun teknis, Bappeda Provinsi Jawa Barat masuk dalam tipologi perangkat daerah A. Dalam melaksanakan tugas pokok dan fungsi serta sebagai upaya pencapaian visi dan misi Pemerintah Daerah Provinsi Jawa Barat, Bappeda berupaya untuk terus meningkatkan kinerjanya sebagai lembaga perencanaan yang andal dengan menjadikan organisasi pembelajaran (*learning organization*) dalam semua aspek termasuk penerapan *good governace* dan *clean government*.

Adapun tugas pokok dan fungsi pada setiap bidang, bagian sub bidang dan sub bagian pada Bappeda Provinsi Jawa Barat, yaitu sebagai berikut:

A. Sekretariat

Sekretariat mempunyai tugas pokok menyelenggarakan administrasi Badan, meliputi perencanaan dan pelaporan, keuangan dan aset serta kepegawaian dan umum, serta membantu Kepala Badan mengkoordinasikan bidang-bidang.

Sekretariat terbagi ke dalam 3 (tiga) sub bagian, yaitu:

1. Sub Bagian Perencanaan dan Pelaporan,
2. Sub Bagian Keuangan dan Aset, dan
3. Sub Bagian Kepegawaian dan Umum.

Dalam menyelenggarakan tugas pokok Sekretariat mempunyai fungsi:

- a. Penyelenggaraan koordinasi, menghimpun dan pengkajian bahan kebijakan teknis bidang perencanaan pembangunan yang dilaksanakan oleh bidang-bidang;

- b. Penyelenggaraan perencanaan dan pelaporan, pengadministrasian keuangan dan aset, serta kepegawaian dan umum;
- c. Penyelenggaraan evaluasi dan pelaporan Badan; dan
- d. Penyelenggaraan fungsi lain sesuai tugas pokok dan fungsinya.

B. Bidang Perencanaan, Pengendalian dan Evaluasi Pembangunan Daerah

Bidang Perencanaan, Pengendalian dan Evaluasi Pembangunan Daerah mempunyai tugas pokok menyelenggarakan fungsi penunjang pelaksanaan urusan pemerintahan bidang perencanaan, pengendalian dan evaluasi pembangunan daerah, meliputi Perencanaan dan Pendanaan, Data dan Informasi, serta Pengendalian, Evaluasi dan Pelaporan. Bidang ini terbagi ke dalam 3 (tiga) sub bidang, yaitu:

1. Sub Bidang Perencanaan dan Pendanaan,
2. Sub Bidang Data dan Informasi, dan
3. Sub Bidang Pengendalian, Evaluasi, dan Pelaporan.

Dalam menyelenggarakan tugas pokok, Bidang Perencanaan, Pengendalian dan Evaluasi Pembangunan Daerah mempunyai fungsi:

- a. Penyelenggaraan pengkajian bahan kebijakan teknis bidang perencanaan, pengendalian dan evaluasi pembangunan daerah;
- b. Penyelenggaraan perencanaan, pengendalian dan evaluasi pembangunan daerah;
- c. Penyelenggaraan evaluasi dan pelaporan Bidang Perencanaan, Pengendalian dan Evaluasi Pembangunan Daerah; dan

d. Penyelenggaraan fungsi lain sesuai tugas pokok dan fungsinya.

C. Bidang Pemerintahan dan Pembangunan Manusia

Bidang Pemerintahan dan Pembangunan Manusia mempunyai tugas pokok menyelenggarakan fungsi penunjang pelaksanaan urusan pemerintahan bidang pemerintahan dan pembangunan manusia, meliputi penyusunan bahan kebijakan teknis dan fasilitasi perencanaan pembangunan pemerintahan Daerah Provinsi aspek pemerintahan, pendidikan, agama dan kebudayaan, serta kesehatan, kependudukan dan ketenagakerjaan. Bidang ini terbagi ke dalam 3 (tiga) sub bidang, yaitu:

1. Sub Bidang Pemerintahan dan Pembangunan Manusia I;
2. Sub Bidang Pemerintahan dan Pembangunan Manusia II;
3. Sub Bidang Pemerintahan dan Pembangunan Manusia III;

Dalam menyelenggarakan tugas pokok, Bidang Pemerintahan dan Pembangunan Manusia mempunyai fungsi:

- a. Penyelenggaraan pengkajian bahan kebijakan teknis perencanaan pembangunan lingkup Bidang Pemerintahan dan Pembangunan Manusia;
- b. Penyelenggaraan pengelolaan perencanaan pembangunan Pemerintahan dan Pembangunan Manusia;
- c. Penyelenggaraan penyusunan bahan kebijakan teknis dan fasilitasi perencanaan pembangunan pemerintahan Daerah Kabupaten/Kota;

- d. Penyelenggaraan evaluasi dan pelaporan Bidang Pemerintahan dan Pembangunan Manusia; dan
- e. Penyelenggaraan fungsi lain sesuai tugas pokok dan fungsinya.

#### D. Bidang Perekonomian dan Sumber Daya Alam

Bidang Perekonomian dan Sumber Daya Alam mempunyai tugas pokok menyelenggarakan fungsi penunjang pelaksanaan urusan pemerintahan Bidang Perekonomian dan Sumber Daya Alam, meliputi penyusunan bahan kebijakan teknis dan fasilitasi perencanaan pembangunan pemerintahan Daerah Provinsi aspek pertanian dan ketahanan pangan, kelautan dan perikanan, koperasi dan usaha kecil, investasi, perindustrian, perdagangan, ketenagakerjaan, pariwisata dan kebudayaan serta, energi dan sumber daya mineral. Bidang ini terbagi ke dalam 3 (tiga) sub bidang, yaitu:

1. Sub Bidang Perekonomian dan Sumber Daya Alam I;
2. Sub Bidang Perekonomian dan Sumber Daya Alam II;
3. Sub Bidang Perekonomian dan Sumber Daya Alam III;

Dalam menyelenggarakan tugas pokok, Bidang Perekonomian dan Sumber Daya Alam mempunyai fungsi:

- a. Penyelenggaraan pengkajian bahan kebijakan teknis di bidang perencanaan Perekonomian dan Sumber Daya Alam;
- b. Penyelenggaraan pengelolaan perencanaan perekonomian dan sumber daya alam;

- c. Penyusunan bahan kebijakan teknis dan fasilitasi perencanaan pembangunan perekonomian dan sumber daya alam Daerah Kabupaten/Kota;
- d. Penyelenggaraan evaluasi dan pelaporan Bidang Perekonomian dan Sumber Daya Alam; dan
- e. Penyelenggaraan fungsi lain sesuai tugas pokok dan fungsinya.

#### E. Bidang Infrastruktur dan Kewilayahan

Bidang Infrastruktur dan Kewilayahan mempunyai tugas pokok menyelenggarakan fungsi penunjang pelaksanaan urusan pemerintahan di bidang perencanaan aspek infrastruktur dan kewilayahan serta penyusunan bahan kebijakan teknis dan fasilitasi perencanaan pembangunan Pemerintahan Kabupaten/Kota, meliputi infrastruktur wilayah, sarana prasarana perumahan dan permukiman, tata ruang, lingkungan hidup dan kehutanan. Bidang ini terbagi ke dalam 3 (tiga) sub bidang, yaitu:

1. Sub Bidang Infrastruktur dan Kewilayahan I;
2. Sub Bidang Infrastruktur dan Kewilayahan II;
3. Sub Bidang Infrastruktur dan Kewilayahan III;

Dalam menyelenggarakan tugas pokok, Bidang Infrastruktur dan Kewilayahan mempunyai fungsi:

- a. penyelenggaraan pengkajian bahan kebijakan teknis di bidang perencanaan Infrastruktur dan Kewilayahan;

- b. penyelenggaraan pengelolaan perencanaan infrastruktur dan kewilayahan;
- c. penyusunan bahan kebijakan teknis dan fasilitasi perencanaan pembangunan infrastruktur dan kewilayahan Daerah Kabupaten/Kota;
- d. penyelenggaraan evaluasi dan pelaporan Bidang Infrastruktur dan Kewilayahan; dan
- e. penyelenggaraan fungsi lain sesuai tugas pokok dan fungsinya.

#### F. Kelompok Jabatan Fungsional

Kelompok Jabatan Fungsional mempunyai tugas melaksanakan sebagian tugas Pemerintah Daerah Provinsi sesuai dengan keahlian dan kebutuhan. Kelompok Jabatan Fungsional terdiri dari sejumlah tenaga fungsional yang ditetapkan berdasarkan ketentuan peraturan perundang-undangan. Kelompok Jabatan Fungsional dipimpin oleh tenaga fungsional yang ditunjuk. Jenis dan jenjang Jabatan Fungsional ditetapkan berdasarkan ketentuan peraturan perundang-undangan. Jumlah Tenaga Jabatan Fungsional ditetapkan berdasarkan beban kerja. Rincian tugas Kelompok Jabatan Fungsional ditetapkan berdasarkan ketentuan peraturan perundang-undangan. Pembinaan administratif Kelompok Jabatan Fungsional, diselenggarakan oleh Sekretaris Badan, meliputi penilaian dan penetapan angka kredit, usulan kenaikan pangkat, gaji berkala, serta pendidikan dan pelatihan. Hasil pelaksanaan tugas Jabatan Fungsional disampaikan kepada Kepala Badan, Kepala Bidang terkait, dan Sekretaris Badan melalui Subbagian Perencanaan dan Pelaporan.

Secara garis besar Badan Perencanaan Pembangunan Daerah Provinsi Jawa Barat atau yang biasa disingkat Bappeda Provinsi Jawa Barat adalah instansi yang mendukung ketercapaian sasaran pembangunan Pemerintah Daerah Provinsi Jawa Barat untuk merumuskan program dan kegiatan yang menunjang pencapaian prioritas pembangunan daerah di Provinsi Jawa Barat. Oleh karena itu, Bappeda Provinsi Jawa Barat tidak melayani secara langsung masyarakat seperti instansi lainnya akan tetapi melayani dan memfasilitasi program dan kegiatan Pimpinan Daerah atau Gubernur, Dewan Perwakilan Rakyat Daerah (DPRD), Perangkat Daerah/Biro dilingkungan Pemerintah Daerah Provinsi Jawa Barat serta Kementerian dan Lembaga instansi lainnya.

Dengan demikian, agenda kegiatan rapat di Bappeda Provinsi Jawa Barat sangat banyak sehingga mengakibatkan kesulitan pada pimpinan untuk membagikan disposisi atau tugas kepada bidang dan terjadi tumpang tindihnya penugasan sehingga mengakibatkan kekeliruan dan kesalahan dalam pelaksanaan kegiatan rapat ataupun kegiatan baik internal maupun eksternal. Hal ini dapat menyebabkan kurangnya dalam pencapaian kinerja Bappeda Provinsi Jawa Barat dalam mendukung kinerja program dan kegiatan yang diselenggarakan di Provinsi Jawa Barat.

Oleh karena itu, penelitian ini bertujuan untuk membangun suatu aplikasi berbasis web yang berfungsi dalam mengelola agenda rapat dan kegiatan yang bisa ditampilkan di layar setiap ruangan sehingga dapat dilihat oleh seluruh staff di Bappeda Provinsi Jawa Barat agenda rapat dan kegiatan yang ditugaskan oleh pimpinan. Hal inilah yang mendorong penulis untuk melakukan penulisan ilmiah

dengan judul “Pembangunan Aplikasi *Meeting Arranger* di lingkungan Badan Perencanaan Pembangunan Daerah Provinsi Jawa Barat”. Aplikasi ini dibangun dengan menggunakan metode *Waterfall*. Menurut (Handrianto & Sanjaya, 2020), Model *Waterfall* adalah “model menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian”.

## **1.2 Identifikasi Masalah**

Berdasarkan dari latar belakang diatas maka dapat diidentifikasi masalah sebagai berikut :

1. Bagaimana membuat aplikasi yang dapat mempermudah pengelolaan agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat;
2. Bagaimana aplikasi ini dapat membantu proses pendistribusian tugas agenda rapat dan kegiatan kepada bidang-bidang di Bappeda Provinsi Jawa Barat; dan
3. Bagaimana aplikasi ini dapat mempermudah dalam penyebaran informasi agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat.

## **1.3 Tujuan Penulisan**

Sedangkan tujuan yang hendak dicapai oleh penulis dalam merancang dan membangun aplikasi tersebut adalah :

1. Dapat memudahkan dalam pengelolaan agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat;
2. Dapat membantu pimpinan dalam mendistribusikan tugas agenda rapat dan kegiatan kepada bidang-bidang di Bappeda Provinsi Jawa Barat; dan
3. Dapat mempermudah dalam penyebaran informasi agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat.

#### **1.4 Batasan Masalah**

Dari pembahasan diatas dapat diilustrasikan bahwa penulis membatasi permasalahannya, yaitu sebagai berikut:

1. Membuat aplikasi untuk memudahkan dalam pengelolaan agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat;
2. Membuat aplikasi untuk membantu pimpinan dalam mendistribusikan tugas agenda rapat dan kegiatan kepada bidang-bidang di Bappeda Provinsi Jawa Barat; dan
3. Membuat aplikasi untuk mempermudah dalam penyebaran informasi agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat.

#### **1.5 Metode Penelitian**

Metode penelitian yang digunakan oleh penulis dalam menyelesaikan skripsi ini adalah Metode pengembangan sistem *waterfall*. Menurut (Handrianto &

Sanjaya, 2020), Model *Waterfall* adalah “model menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian”.

Adapun tahapan dari metode ini adalah sebagai berikut dibawah ini:

a. Analisis

Melakukan analisis kebutuhan perangkat lunak, fungsi dan proses dari *web* yang dibuat, pengidentifikasian kendala dalam pembuatan *web*, menganalisis keandalan, kelemahan, dan teknologi yang dipakai.

b. Desain

Desain perangkat lunak adalah proses beberapa tahapan langkah pada rancangan pembuatan program perangkat lunak meliputi struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahapan analisis kebutuhan ke representasi rancangan agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Pada tahap ini, hasil dari desain perangkat lunak yang telah ada didokumentasikan.

c. Pengkodean

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai desain yang telah dibuat pada tahap desain. Atau tahapan penulis membuat program dengan bahasa program seperti PHP, HTML, SCC dan lain-lain.

d. Pengujian

Tahapan ini penulis melakukan pengujian terhadap program yang telah dibuat untuk mengetahui kekurangan dari program tersebut. Seperti validasi halaman *login*, apakah sesuai dengan harapan.

## **1.6 Sistematika Penulisan**

### **BAB I PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang masalah, maksud dan tujuan, rumusan masalah, batasan masalah, metodologi penelitian dan sistematika penulisan.

### **BAB II LANDASAN TEORI**

Bab ini mengenai landasan teori yang menjelaskan secara garis besar beberapa teori yang menjadi dasar penulisan skripsi, diantaranya mengenai teori-teori sistem informasi dan bahasa pemrograman yang digunakan.

### **BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM**

Bab ini menguraikan mengenai metode pengumpulan data dan model dari metode pengembangan sistem yang digunakan.

### **BAB IV IMPLEMENTASI DAN UJI COBA**

Bab ini berisikan tentang implementasi atas perancangan sistem baru yang meliputi batasan-batasan terhadap perangkat lunak maupun perangkat keras yang digunakan.

### **BAB V PENUTUP**

Bab ini berisikan beberapa kesimpulan berdasarkan uraian-uraian yang

diperoleh dari bab-bab sebelumnya dan beberapa saran untuk pengembangan sistem lebih lanjut.

#### **DAFTAR PUSTAKA**

#### **LAMPIRAN-LAMPIRAN**

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Penelitian Terkait**

Pada jurnal yang disusun oleh (Nangin et al., 2019), melakukan sebuah penelitian perancangan Aplikasi Penjadwalan Kerja di Usaha Perjalanan Wisata Berbasis Android. Aplikasi akan menguji penjadwalan kerja sebuah proses pembuatan urutan rencana kerja kedalam bentuk daftar catatan yang berisi kegiatan sehari-hari, saat ini penggunaan teknologi Android sudah berkembang pesat diberbagai bidang kehidupan, seperti halnya penerapan aplikasi pengingat yang dibuat untuk menjadwalkan agenda.

(Ramsari & Rifaldi, 2018) juga melakukan penelitian tentang perancangan Aplikasi Penjadwalan Kegiatan Akademik disertai Sistem Reminder Berbasis *Responsive Web Design*. Aplikasi akan menguji Penjadwalan disertai sistem *reminder* untuk mempermudah proses pencatatan dan pengingat kegiatan akademik FIKI Universitas Nurtanio Bandung.

Beberapa Referensi ada pada tabel berikut ini :

**TABEL 2.1 Tabel Referensi**

<b>No</b>	<b>Penulis</b>	<b>Judul</b>	<b>Pembahasan</b>
1	(Nangin et al., 2019)	Rancang Bangun Aplikasi Penjadwalan Kerja di Usaha Perjalanan Wisata Berbasis Android	Penjadwalan kerja merupakan sebuah proses pembuatan urutan rencana kerja kedalam bentuk daftar catatan yang berisi kegiatan sehari- hari, saat ini penggunaan teknologi Android sudah berkembang pesat diberbagai bidang kehidupan, seperti halnya penerapan aplikasi pengingat yang dibuat untuk menjadwalkan agenda

No	Penulis	Judul	Pembahasan
2	(Ramsari & Rifaldi, 2018)	Rancang Bangun Aplikasi Penjadwalan Kegiatan Akademik disertai Sistem <i>Reminder</i> Berbasis Responsive <i>Web Design</i>	Penjadwalan disertai sistem reminder untuk mempermudah proses pencatatan dan pengingat kegiatan akademik FIKI Universitas Nurtanio Bandung.
3.	(Maulana & Herdiana, 2019)	Pengembangan Aplikasi Peningat Jadwal Periksa Medis <i>Short Message Service (SMS) Gateway</i> Berbasis Web Kepada Penderita Gagal Ginjal	Aplikasi pengingat jadwal penderita gagal ginjal berbasis web dibuat dengan bahasa pemrograman PHP serta MySQL sebagai media penyimpanan data serta Gammu sebagai SMS gateway yang ditujukan untuk

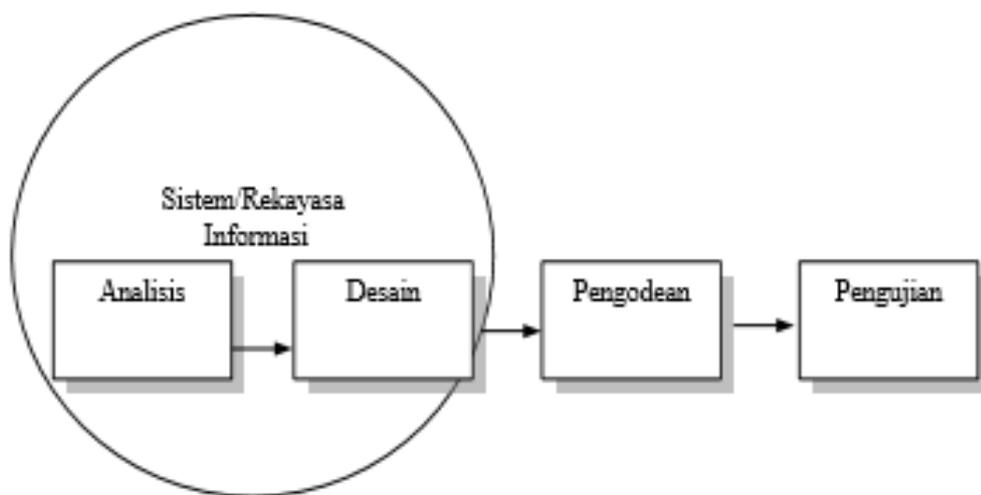
No	Penulis	Judul	Pembahasan
			<p>layanan rumah sakit daerah umum majalengka yaitu mengingatkan pasien penderita gagal ginjal untuk melakukan cuci darah.</p>
4.	(Widagdo, 2020)	<p>Pembuatan Aplikasi Mobile Peningat Minum Obat Untuk Orang Tua</p>	<p>Penjadwalan Peningat Minum Obat untuk Orang Tua melalui mobile.</p>
5.	(Vinandha et al., 2019)	<p>Pengembangan Aplikasi <i>Mobile</i> Peningat Jadwal Layanan Posyandu dengan Menggunakan Teknologi <i>Firebase Cloud</i></p>	<p>Sehingga dilakukanlah penelitian dengan mengembangkan suatu aplikasi yang dapat membantu untuk mengingat informasi serta layanan yang ada pada posyandu,</p>

No	Penulis	Judul	Pembahasan
			dengan melakukan pencatatan administrasi pelayanan posyandu pada sebuah aplikasi yang dapat membantu masyarakat untuk memperoleh informasi mengenai pelayanan posyandu serta mengingat jadwal pelayanan yang ada pada posyandu.

## 2.2 Pengertian *Waterfall*

Dalam Pembangunan Aplikasi *Meeting Arranger* di lingkungan Badan Perencanaan Pembangunan Daerah Jawa Barat di Bappeda Provinsi Jawa Barat, penulis menggunakan metode *Waterfall*. Sedangkan pemodelan yang digunakan untuk pengembangan sistem yang dilakukan adalah Model *Waterfall*. Menurut, (Handrianto & Sanjaya, 2020) Model *Waterfall* adalah “model menyediakan

pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian”. Berikut gambar dari model waterfall dalam SDLC.



**GAMBAR 2.1 Ilustrasi Model *Waterfall***

a. Analisis

Melakukan analisis kebutuhan perangkat lunak, fungsi dan proses dari *web* yang dibuat, pengidentifikasian kendala dalam pembuatan web, menganalisis keandalan, kelemahan, dan teknologi yang dipakai.

b. Desain

Desain perangkat lunak adalah proses beberapa tahapan langkah pada rancangan pembuatan program perangkat lunak meliputi struktur

data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahapan analisis kebutuhan ke representasi rancangan agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Pada tahap ini, hasil dari desain perangkat lunak yang telah ada didokumentasikan.

c. Pengkodean

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai desain yang telah dibuat pada tahap desain. Atau tahapan penulis membuat program dengan bahasa program seperti PHP, HTML, SCC dan lain-lain.

d. Pengujian

Tahapan ini penulis melakukan pengujian terhadap program yang telah dibuat untuk mengetahui kekurangan dari program tersebut. Seperti validasi halaman login, apakah sesuai dengan harapan.

### 2.3 Pengertian Aplikasi

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja *to apply* yang dalam bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu subkelas perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Contoh utama perangkat lunak aplikasi adalah program pengolah kata,

lembar kerja, dan pemutar media. Kumpulan aplikasi komputer yang digabung menjadi suatu paket biasanya disebut paket atau *suite* aplikasi (*application suite*).

Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*. Pengertian aplikasi menurut para ahli adalah sebagai berikut.

Menurut Rachmad Hakim S (2018) pada jurnal (Tri, 2020), Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur *Windows* dan permainan (*game*), dan sebagainya.

Sedangkan menurut Harip Santoso (2017), Aplikasi adalah suatu kelompok file (form, class, rePort) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi *payroll*, aplikasi *fixed asset*, dan lain-lain. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan.

Sedangkan Menurut Kamus Besar Bahasa Indonesia (KBBI) “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari *user* (pengguna).”

Dengan demikian dapat disimpulkan bahwa pengertian aplikasi adalah sebuah rancangan dari sebuah sistem yang dapat difungsikan secara khusus untuk dapat mengerjakan suatu tugas atau perintah tertentu yang telah diatur sesuai dengan yang membangun rancangan tersebut.

## **2.4 Pengertian Pengelolaan**

Pengelolaan berasal dari kata kelola, dalam Kamus Besar Bahasa Indonesia berarti memimpin, mengendalikan, mengatur, dan mengusahakan agar lebih baik, lebih maju, serta bertanggung jawab atas pekerjaan tertentu. Pengelolaan adalah proses yang membantu merumuskan kebijaksanaan dan tujuan memberikan pengawasan pada semua hal yang terlibat dalam pelaksanaan dan pencapaian tujuan.

Pengelolaan bisa diartikan sebagai manajemen, yaitu suatu proses kegiatan yang dimulai dari perencanaan, pengorganisasian, pengarahan dan pengawasan usaha-usaha para anggota organisasi dan penggunaan-penggunaan sumber daya organisasi lainnya agar mencapai tujuan organisasi yang telah ditentukan.

Menurut Terry, mengartikan fungsi pengelolaan sebagai usaha untuk mencapai tujuan yang telah ditentukan sebelumnya melalui usaha orang lain.

Pengelolaan tidak akan terlepas dari kegiatan sumber daya manusia yang ada dalam suatu kantor, instansi, maupun organisasi. Manajer yang baik selalu bekerja dengan langkah-langkah manajemen yang fungsional, yaitu merencanakan, mengorganisasikan, mengarahkan, dan mengontrol.

Dengan demikian, target yang dituju dengan mudah dapat dicapai dengan baik.

## **2.5 Pengertian Pendistribusian**

Keputusan mengenai saluran distribusi dalam pemasaran adalah merupakan

salah satu keputusan yang paling kritis yang dihadapi manajemen. Saluran yang dipilih akan mempengaruhi seluruh keputusan pemasaran yang lainnya. Dalam rangka untuk menyalurkan barang dan jasa dari produsen kepada konsumen maka perusahaan harus benar-benar memilih atau menyeleksi saluran distribusi yang akan digunakan, sebab kesalahan dalam pemilihan saluran distribusi ini dapat menghambat bahkan dapat memacetkan usaha menyalurkan barang atau jasa tersebut.

Menurut (Hall 2001) Pendistribusian atau distribusi adalah kegiatan untuk mengirimkan produk ke pelanggan setelah penjualan. Dapat ditarik kesimpulan bahwa prosedur pendistribusian adalah suatu tahapan atau rangkaian aktivitas yang dilakukan secara berulang yang berhubungan dengan pemasaran produk.

Mulai dari produk masih berada pada entitas yang memproduksi hingga produk tersebut dipasarkan. Dengan adanya prosedur dalam pendistribusian tersebut maka proses pemasaran akan berjalan dengan efektif dan tujuan dari perusahaan tercapai

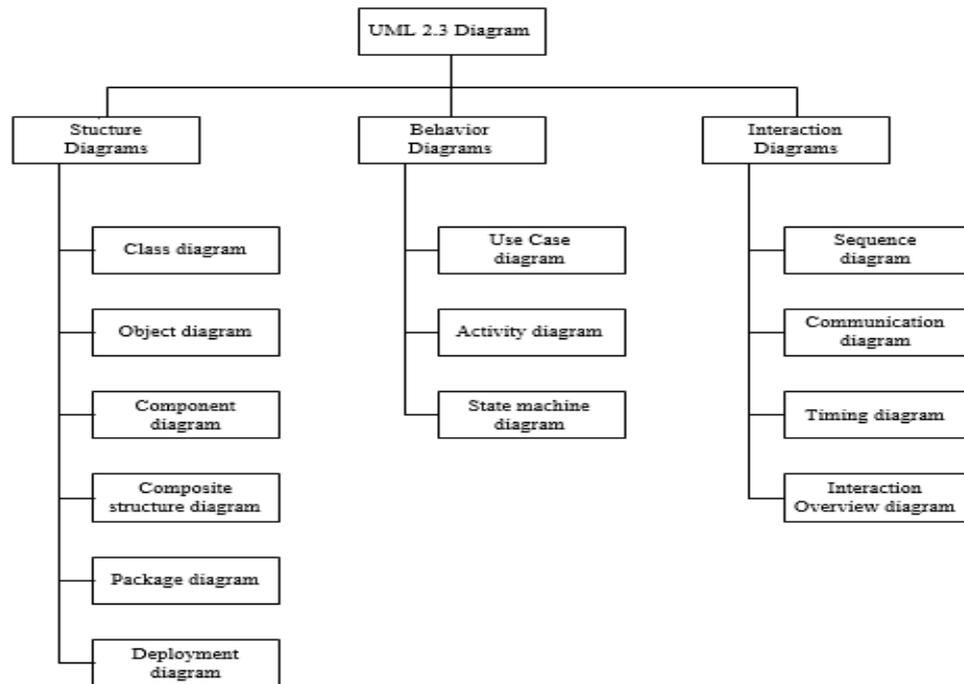
Menurut Subagyo, Nur dan Indra, (2018) Distribusi merupakan pergerakan atau perpindahan barang atau jasa dari sumber sampai ke konsumen akhir, konsumen atau pengguna, melalui saluran distribusi (*distribution channel*), dan gerakan pembayaran dalam arah yang berlawanan, sampai ke produsen asli atau pemasok. Menurut Arif, (2018) Distribusi dapat diartikan sebagai kegiatan pemasaran yang berusaha memperlancar dan mempermudah penyampaian barang dan jasa dari produsen kepada konsumen, sehingga penggunaannya sesuai dengan yang diperlukan.

## **2.6 Unified Modeling Language (UML)**

Menurut (James Rumbaugh et al., 2013) dalam bukunya yang berjudul “*The Unified Modeling Language Reference Manual*” menjelaskan bahwa *Unified Modeling Language (UML)* adalah bahasa pemodelan visual universal yang digunakan untuk mendefinisikan, memvisualisasikan, membuat, dan merekam artefak produk. Sistem perangkat lunak. Ini menangkap keputusan dan pemahaman tentang sistem mana yang harus digunakan dibangun. Ini digunakan untuk memahami, merancang, mencari, mengkonfigurasi, memelihara dan kontrol informasi tentang sistem. Ini dirancang untuk digunakan dengan semua metode pengembangan, tahapan siklus hidup, domain aplikasi dan media. Pemodelan bahasa yang dirancang untuk menyatukan pengalaman masa lalu dalam teknologi pemodelan, dan integrasikan praktik terbaik perangkat lunak saat ini ke dalam metode standar. UML mencakup konsep, simbol, dan pedoman semantik. Ini memiliki bagian statis, dinamis, lingkungan dan organisasi. Didesain untuk didukung secara interaktif alat pemodelan visual dengan *encoder* dan penulis laporan.

### **2.6.1 Diagram UML**

Menurut (Ariani Sukamto & Shalahuddin, 2016) pada UML 2.3 berisi 13 (tiga belas) grafik yang terbagi dalam 3 (tiga) kategori. Kategori dan jenis grafik adalah sebagai berikut.



**GAMBAR 2.2 Diagram UML**

Berikut penjelasan dari pembagian kategori tersebut.

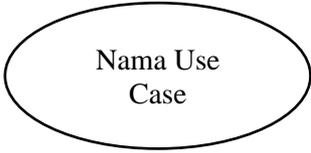
1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan struktur statis dari sistem yang akan dimodelkan.
2. *Behavior diagrams* yaitu sekumpulan diagram yang digunakan untuk menggambarkan perilaku sistem atau rangkaian perubahan sistem.
3. *Interaction diagram* yaitu Sekumpulan diagram yang digunakan untuk menggambarkan interaksi antara sistem dan sistem lain serta interaksi antar subsistem dalam sistem.

### 2.6.2 Use Case Diagram

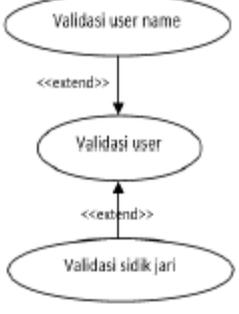
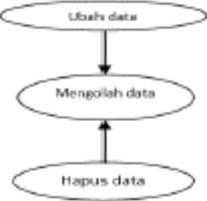
Menurut (Ariani Sukanto & Shalahuddin, 2016) *Use case* menggambarkan interaksi antara satu atau lebih partisipan dan sistem informasi yang akan dibuat. Secara garis besar *use case* digunakan untuk menentukan fungsi-fungsi dalam suatu sistem informasi dan siapa yang berhak menggunakan fungsi tersebut.

Berikut adalah simbol dalam *use case* diagram:

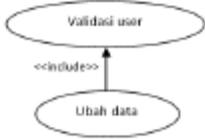
**TABEL 2.2 Simbol pada Use Case Diagram**

Simbol	Deskripsi
Use Case  	Fungsi yang disediakan oleh sistem sebagai satu kesatuan dapat bertukar pesan antar unit atau partisipan; biasanya dengan Gunakan kata kerja di awal atau awal frasa dalam nama kasus penggunaan
Aktor/ <i>Actor</i>  	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi akan dibuat di luar sistem informasi itu sendiri, oleh karena itu, meskipun simbol aktor adalah gambaran dari

Simbol	Deskripsi
<p data-bbox="520 584 683 613">Nama Aktor</p>	<p data-bbox="847 365 1353 613">suatu karakter, aktor tersebut belum tentu karakter. Biasanya menggunakan kata ini Objeknya ada di awal frasa nama aktor.</p>
<p data-bbox="384 663 659 692"><i>Asosiasi/ association</i></p> 	<p data-bbox="847 663 1353 911">Komunikasi antara aktor yang berpartisipasi dalam use case atau use case dan use case yang dimiliki Interaksi dengan aktor</p>
<p data-bbox="384 960 595 990"><i>Ekstensi/ extend</i></p>	<p data-bbox="847 960 1353 1794">Komunikasi dengan kasus penggunaan tambahan dari kasus penggunaan, di mana meskipun tidak ada kasus penggunaan tambahan, kasus penggunaan tambahan dapat ada secara independen; mirip dengan prinsip pewarisan dalam pemrograman berorientasi objek; biasanya, kasus penggunaan lain memiliki nama gudang perangkat lunak yang sama dengan kasus penggunaan tambahan, misalnya:</p>

Simbol	Deskripsi
	 <p>Panah menunjuk ke kasus penggunaan tambahan</p>
Generalisasi/ <i>generalization</i>	<p>Misalnya, hubungan antara generalisasi dan spesialisasi (umum-spesifik) antara dua kasus penggunaan, di mana satu fungsi lebih umum daripada yang lain.</p>  <p>Arah panah menunjuk ke use case, dan use case menjadi generalisasinya (umum).</p>
	<p>Hubungan dengan use case lain yang telah ditambahkan ke use case yang membutuhkan use case untuk menjalankan fungsinya atau sebagai</p>

Simbol	Deskripsi
	syarat untuk menjalankan use case ini.
<p data-bbox="379 439 754 472">Menggunakan/ <i>include/ uses</i></p> <div data-bbox="501 533 707 719" style="text-align: center;"> <pre> graph TD     subgraph include         direction LR         I1[&lt;&lt;include&gt;&gt;] --- A1[-----&gt;]     end     subgraph uses         direction LR         U1[&lt;&lt;uses&gt;&gt;] --- A2[—————&gt;]     end </pre> </div>	<p data-bbox="845 439 1348 763">Hubungan dengan use case lain yang telah ditambahkan ke use case yang membutuhkan use case untuk menjalankan fungsinya atau sebagai syarat untuk menjalankan use case ini.</p> <p data-bbox="845 801 1348 913">Terkait kasus penggunaan, ada dua tampilan yang cukup besar:</p> <ol data-bbox="898 952 1348 1279" style="list-style-type: none"> <li>1. Include berarti bahwa use case yang ditambahkan akan selalu dipanggil ketika use case tambahan dijalankan, misalnya dalam kasus berikut ini:</li> </ol> <div data-bbox="1042 1328 1262 1485" style="text-align: center;"> <pre> graph TD     Login([login]) -- &lt;&lt;include&gt;&gt; --&gt; Validasi([Validasi user name]) </pre> </div> <ol data-bbox="898 1547 1348 1944" style="list-style-type: none"> <li>2. Include artinya use case tambahan akan selalu mengecek apakah use case tambahan sudah dieksekusi sebelum use case tambahan dieksekusi, misalnya pada case</li> </ol>

Simbol	Deskripsi
	<p data-bbox="943 362 1043 394">berikut:</p>  <pre data-bbox="1043 434 1248 573"> graph TD     A([Validasi user])     B([Ubah data])     B -- &lt;&lt;include&gt;&gt; --&gt; A </pre> <p data-bbox="847 622 1353 875">Berdasarkan pertimbangan dan penjelasan yang diperlukan, kedua penjelasan di atas dapat menjadi salah satu dari dua</p>

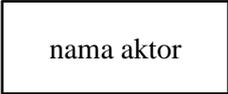
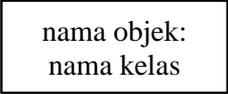
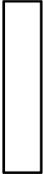
### 2.6.3 Sequence Diagram

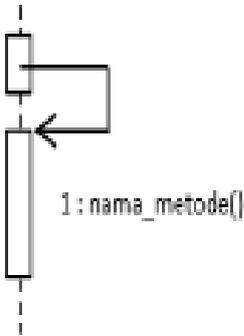
Menggambarkan perilaku objek dalam *use case* dengan menjelaskan siklus hidup objek dan pesan yang dikirim dan diterima di antara objek. Oleh karena itu, untuk menggambar diagram urutan, Anda harus mengetahui objek yang terlibat dalam kasus penggunaan dan metode yang termasuk dalam kelas yang digunakan oleh objek tersebut.

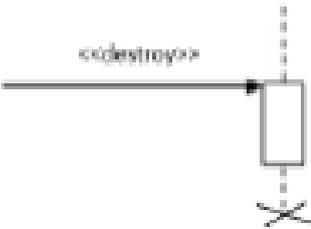
Jumlah *sequence* diagram yang harus digambar adalah sebanyak definisi *use case* dan alurnya masing-masing. Yang terpenting adalah semua *use case* telah diidentifikasi dan interaksi jalur pesan dimasukkan ke dalam *sequence* diagram. Oleh karena itu, semakin banyak *use case* teridentifikasi, semakin banyak pula operasi yang harus dilakukan.

Berikut ini simbol pada *sequence* diagram.

**TABEL 2.3 Simbol pada Diagram Sequence**

Simbol	Deskripsi
<p>Aktor</p>  <p>Atau</p>  <p>tanpa waktu aktif</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi akan dibuat di luar sistem informasi itu sendiri. Oleh karena itu, meskipun lambang seorang aktor merupakan gambaran dari seorang tokoh, seorang aktor belum tentu merupakan seorang tokoh. Biasanya, kata benda digunakan di awal frasa nama aktor</p>
<p>Garis Hidup/ <i>lifeline</i></p> 	<p>Merepresentasikan kehidupan objek tersebut.</p>
<p>Objek</p>  <p>Waktu aktif</p> 	<p>Merepresentasikan objek yang berinteraksi dengan pesan Deklarasikan objek aktif dan tukar pesan</p>

<p>Pesan tipe <i>create</i></p>  <p>A horizontal arrow pointing to the right. Above the arrow is the text '&lt;&lt;create&gt;&gt;'.</p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>Call</i></p>  <p>A horizontal arrow pointing to the right. Above the arrow is the text '1: nama_metode()'.</p>	<p>Menyatakan objek Operasi / metode panggilan Di objek lain atau dirinya sendiri</p>  <p>A sequence diagram showing two vertical lifelines. The top lifeline has a message arrow that loops back to itself, labeled '1: nama_metode()'. The bottom lifeline also has a message arrow pointing to it from the top lifeline, also labeled '1: nama_metode()'.</p> <p>Arah / panah menunjuk ke objek melalui operasi / metode, karena menurut diagram kelas objek interaktif, operasi / metode dipanggil, dan operasi / metode yang dipanggil harus ada pada diagram kelas</p>
<p>Pesan tipe <i>send</i>,</p>  <p>A horizontal arrow pointing to the right. Above the arrow is the text '1: masukan'.</p>	<p>Menyatakan bahwa satu objek mengirimkan data / input / informasi ke objek lain, arah</p>

	Panah menunjuk ke objek yang akan dikirim.
<p>Pesan tipe <i>return</i></p> 	Ketika sebuah objek yang telah melakukan operasi atau metode mengembalikan objek tertentu, panah menunjuk ke objek yang menerima nilai pengembalian.
<p>Pesan tipe <i>destroy</i></p> 	Nyatakan bahwa suatu objek mengakhiri umur objek lain, dan panah menunjuk ke objek yang diakhiri. Sebaiknya ada objek yang dibuat, lalu ada objek yang hancur.

## 2.7 MySQL

Menurut Budi Raharjo (2015) MySQL adalah RDBMS (*database server*), dapat mengatur *database* dengan sangat cepat, Bisa menampung banyak data dan bisa diakses banyak orang pengguna.

Menurut definisi Heni A. Puspitosari (2011) MySQL Merupakan salah satu perangkat lunak yang paling banyak digunakan untuk *server database*, MySQL adalah *open source* dan menggunakan SQL.

## 2.8 *Hypertext Preprocessor (PHP)*

PHP adalah bahasa skrip *slide server*, bahasa pemrograman yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi web. PHP adalah singkatan dari *Hypertext Preprocessor*, sebelumnya dikenal sebagai "beranda pribadi".

## 2.9 Apache

Apache adalah *server web*, sangat populer dan paling banyak digunakan di dunia. Dikelola oleh *Apache Software Foundation*, Apache diluncurkan pada tahun 1995 dan tetap populer selama setahun setelah itu hingga hari ini. Sebagai fungsi *web browser*, Apache bertindak sebagai penghubung antara pengguna (*browser*) dan server. Apache awalnya dikembangkan sebagai server web *open source* untuk sistem operasi seperti UNIX. Apache telah banyak digunakan oleh banyak perusahaan besar seperti LinkedIn, Adobe, General Electric, IBM, dan penyedia layanan panel kontrol juga menggunakan Apache sebagai *browser web* mereka.

## 2.10 *Open Source*

Menurut (kominfo, 2013) *Open source* adalah sistem pengembangan yang tidak dikoordinasikan oleh individu / instansi pusat, tetapi oleh peserta yang bekerja sama melalui distribusi dan penggunaan kode sumber secara gratis (biasanya menggunakan alat komunikasi internet). Model pengembangan ini menggunakan

model *bazaar style*, sehingga model *open source* ini memiliki keunikan tersendiri yaitu dorongan dari budaya donasi, artinya ketika komunitas menggunakan program *open source* dan mendapatkan keuntungan, mereka akan termotivasi untuk melamar satu pertanyaan. Pengguna dapat memberikan kembali kepada orang banyak.

Model *open source* lahir karena memiliki kebebasan untuk bekerja, tanpa memikirkan intervensi dan menggunakan pengetahuan dan produk yang sesuai untuk mengekspresikan apa yang diinginkan. Ketika kebebasan diumumkan kepada publik, kebebasan menjadi pertimbangan utama. Komunitas lain memiliki kebebasan untuk belajar, mengutak-atik, memodifikasi, membenarkan bahkan menyalahkan, tetapi kebebasan ini juga hadir dengan tanggung jawab, bukan kebebasan tanpa tanggung jawab.

## **2.11 Pengujian Perangkat Lunak**

### **2.11.1 Definisi**

Definisi pengujian perangkat lunak menurut (perry, W. E. 1990) pada jurnal (MZ, 2016) Pengujian perangkat lunak adalah proses menjalankan program atau sistem yang bertujuan untuk menemukan atau melibatkan setiap aktivitas yang bertujuan untuk mengevaluasi sifat atau fungsi program atau sistem dan menentukan hasil yang memenuhi kebutuhan perusahaan.

### **2.11.2 Tujuan Pengujian aplikasi**

1. Menemukan beberapa kemungkinan kesalahan dalam perangkat

lunak yang dirancang; dan

2. Menguji fungsi dari aplikasi tersebut.

### **2.11.3 *Black box Testing***

Fokus pada persyaratan fungsional perangkat lunak. Oleh karena itu, pengujian kotak hitam memungkinkan pengembang perangkat lunak untuk membuat sekumpulan kondisi input untuk mengimplementasikan semua persyaratan fungsional program. Pengujian *black box* bukanlah alternatif pengujian *white box*, tetapi metode pelengkap untuk menemukan kesalahan selain metode *white box*.

## **BAB III**

### **ANALISA MASALAH DAN PERANCANGAN PROGRAM**

Bab ini membahas mengenai pembangunan aplikasi *meeting arranger* di Bappeda Provinsi Jawa Barat. Langkah awal dari perancangan aplikasi *meeting arranger* adalah analisis yang dimulai dengan observasi, observasi disini mengambil teknik wawancara terhadap beberapa pimpinan dan pegawai di Bappeda Provinsi Jawa Barat agar dapat menentukan analisis masalah dan penentuan kebutuhan komponen aplikasi untuk perancangan program.

#### **3.1 Analisis (Observasi Lapangan)**

Pada tahapan ini penulis melakukan observasi lapangan untuk memperoleh informasi keterangan yang berhubungan dengan pokok permasalahan pada penelitian untuk mengumpulkan data dengan langkah wawancara terhadap pimpinan dan beberapa pegawai diantaranya pegawai yang mengelola surat masuk merasa kesulitan dengan menumpuknya atapun hilang surat disposisi karena dalam proses pengelolaan dan pendistribusian masih manual menggunakan kertas yang diantar oleh pegawai penatausahaan.

Adapun beberapa pedoman pertanyaan dalam wawancara ini adalah sebagai berikut:

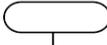
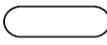
**TABEL 3.1 Formulir Wawancara**

No	Pertanyaan Wawancara	Jawaban Responden	Responden
1.	Apakah anda pernah kesulitan dalam pengelolaan rapat dan kegiatan pimpinan dengan menggunakan cara manual?	Ya, karena saya menyusun kembali disposisi surat yang telah diisi oleh pimpinan untuk dikerjakan oleh pegawai	Staff Tata Usaha Kepala Bappeda Provinsi Jawa Barat
2.	Apakah anda pernah kesulitan dalam pendistribusian rapat dan kegiatan yang ditugaskan pimpinan kepada pegawai dengan menggunakan cara manual?	Ya, karena saya harus bergerak dan keliling ke setiap ruangan untuk mengantarkan disposisi surat yang telah diisi oleh pimpinan untuk dikerjakan oleh pegawai	Staff Tata Usaha Kepala Bappeda Provinsi Jawa Barat
3.	Apakah anda pernah kesulitan untuk menginformasikan rapat dan kegiatan kepada pegawai yang lain?	Ya, meskipun disposisi surat sudah saya antarkan ke pegawai yang bersangkutan, tetap saja ada yang terlewat dan lupa untuk dihadiri oleh pegawai	Staff Tata Usaha Kepala Bappeda Provinsi Jawa Barat

No	Pertanyaan Wawancara	Jawaban Responden	Responden
		tersebut	
4.	Apakah anda paham dengan fitur tambah, edit, dan hapus yang ada pada aplikasi ?	Saya paham dengan fitur tambah, edit, dan hapus yang ada di aplikasi. Fitur tersebut cukup mudah untuk dijalankan.	Staff Tata Usaha Kepala Bappeda Provinsi Jawa Barat
5.	Bagaimana menurut anda dengan adanya aplikasi ini?	<p>Dengan adanya aplikasi ini memudahkan saya dalam pengelolaan, pendistribusian dan menginformasikan agenda rapat dan kegiatan baik internal maupun eksternal.</p> <p>Dengan adanya aplikasi ini, saya hanya perlu membuka handphone atau laptop untuk mengecek tugas yang diberikan kepada saya melalui browser.</p>	<p>Staff Tata Usaha Kepala Bappeda Provinsi Jawa Barat</p> <p>Kepala Sub Bagian Perencanaan dan Pelaporan</p>

### 3.1.1 Analisis Sistem Eksisting

Tahapan analisis kondisi eksisting merupakan kegiatan peninjauan ke sumber data dengan tujuan untuk memperoleh data yang akurat dan relevan. Sumber data tersebut dapat dilihat pada gambar *flowchart* dibawah ini:

NO	Uraian Kegiatan	Pelaksana			
		PengADM Pegaw ai 3	KASUBAG UMUM 4	Sekretaris 5	KEPALA BAPPEDA 6
1	Menerima Surat Masuk Dari Pengirim (Pos / Caraka/ Kurir / Perorangan) Serta Meneliti Kebenaran Alamat Naskah Dinas Apabila Benar Kemudian Diagendakan Tapi Apabila Salah Alamat Dikembalikan Pada Pengirim				
2	Meneliti Surat/ Naskah Dinas Yang Masuk Dan Mencantumkan Kode Klasifikasi Pada Surat, Apabila Ada Kesalahan Dikembalikan Pada Pengelola Kearsipan				
3	Menuliskan Catatan Pada Kartu Disposisi				
4	Menuliskan Instruksi / Arahan Lebih Lanjut Untuk Penanganan Surat tersebut				
5	Menuliskan Pada Kartu Kendali Kemudian Memberikan Surat tersebut Kepada Unit Pengolahan (Unit Kerja Dilingkungan Dinas / Kaban) Sesuai Arahan / Catatan / Instruksi Sekertaris Dan/ Atau Kadis/ Kaban				

**GAMBAR 3.1** *Flowchart* SOP Surat Masuk di Bappeda Provinsi Jawa Barat

### 3.1.2 Analisis Permasalahan

Hal yang mendasari penulis membangun aplikasi *meeting arranger* di Bappeda Provinsi Jawa Barat yaitu banyaknya surat undangan yang masuk serta kegiatan yang diselenggarakan, sehingga menyebabkan agenda

kegiatan rapat di Bappeda Provinsi Jawa Barat sangat banyak sehingga mengakibatkan kesulitan pada pimpinan untuk membagikan tugas kepada bidang dan terjadi tumpang tindihnya penugasan sehingga mengakibatkan kekeliruan dan kesalahan dalam pelaksanaan kegiatan rapat ataupun kegiatan, baik internal maupun eksternal sebagaimana sudah dijelaskan pada Bab I diatas. Hal ini dapat menyebabkan kurangnya ketercapaian dalam pencapaian kinerja Bappeda Provinsi Jawa Barat dalam mendukung kinerja program dan kegiatan yang diselenggarakan di Provinsi Jawa Barat. Setelah penulis menelaah penyebab utama dari masalah ini, yaitu pertama banyak kartu disposisi dari pimpinan yang tidak diketahui oleh bawahannya disaat bawahan sedang mengikuti rapat atau melaksanakan kegiatan yang lain karena berkas fisik atau *print out* yang tertumpuk dan bahkan hilang. Kemudian yang kedua, selain melalui kartu disposisi yang tertumpuk terkadang pimpinan memberikan tugas atau kegiatan itu melalui pesan pribadi ataupun grup *WhatsApp*, meskipun sudah menggunakan aplikasi tersebut, tetap saja masih ada rapat atau kegiatan yang tertumpuk dengan banyaknya agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat.

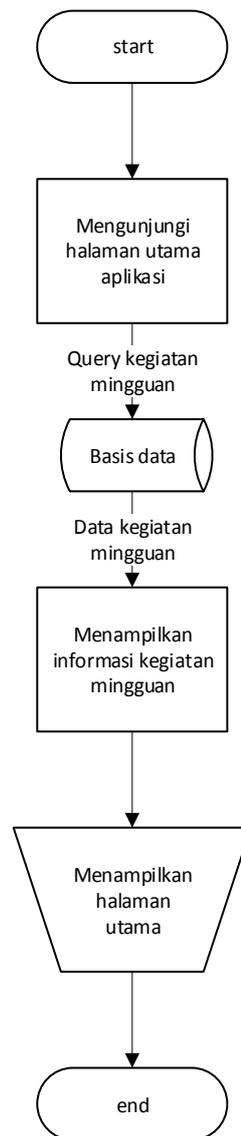
Aplikasi yang akan dibangun pada tugas akhir ini adalah Pembangunan Aplikasi *Meeting Arranger* di Bappeda Provinsi Jawa Barat dengan menggunakan metode *Waterfall*. Aplikasi ini bertujuan untuk mempermudah dalam pengelolaan agenda rapat dan kegiatan, pendistribusian tugas agenda rapat dan kegiatan kepada pegawai dan penyebaran informasi kepada para pegawai mengetahui rapat dan kegiatan yang dilaksanakan oleh

Bappeda Provinsi Jawa Barat setiap hari secara *realtime* maupun yang akan datang, aplikasi ini akan diinput oleh administrator aplikasi *meeting arranger*. Sehingga seluruh pegawai di Bappeda Provinsi Jawa Barat dapat mengakses ketika pegawai sedang berada dimanapun berada karena aplikasi tersebut berbasis *website*.

### **3.1.3 Analisis Sistem Usulan**

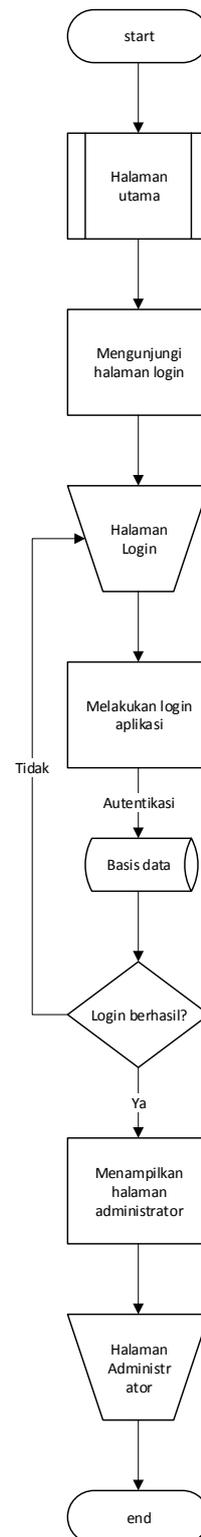
Analisis sistem usulan menggambarkan rancangan sistem yang diusulkan sebagai dasar pembuatan aplikasi yang dipresentasikan dengan menggunakan diagram *flowchart*. Pembahasan mengenai analisis sistem usulan dibagi berdasarkan beberapa fungsi utama dari aplikasi yang akan dibangun.

Sistem usulan untuk halaman utama menggambarkan alur proses yang ditampilkan ketika pengguna aplikasi mengakses halaman utama aplikasi. Pada halaman ini aplikasi akan menampilkan data kegiatan mingguan. Adapun sistem usulan halaman utama dapat dilihat pada gambar berikut:



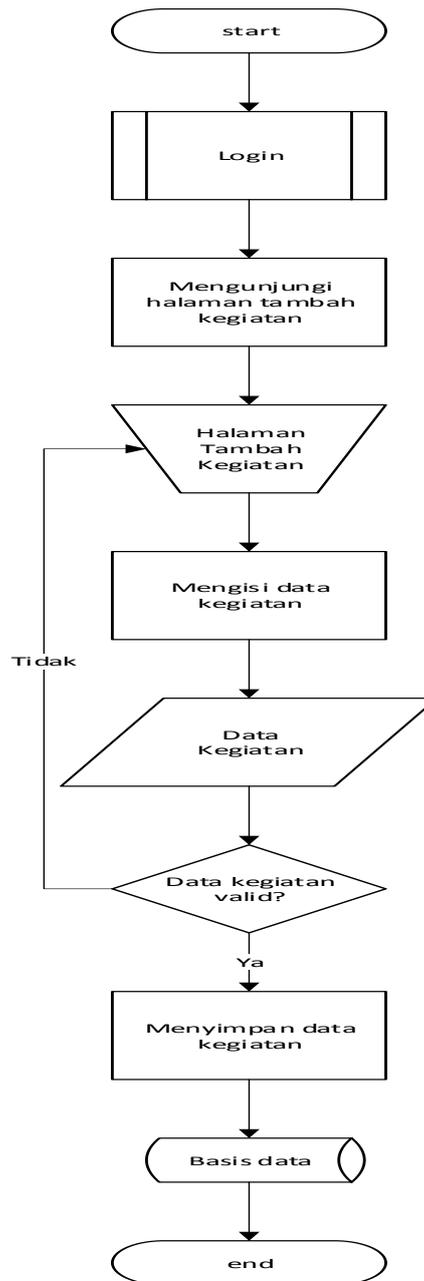
**GAMBAR 3.2 Sistem Usulan Halaman Utama**

Sistem usulan untuk halaman *login* menggambarkan alur proses yang ditampilkan ketika pengguna aplikasi ingin masuk kedalam aplikasi untuk mengelola kegiatan. Pada halaman ini aplikasi akan menampilkan formulir *login*. Adapun sistem usulan halaman *login* dapat dilihat pada gambar berikut:



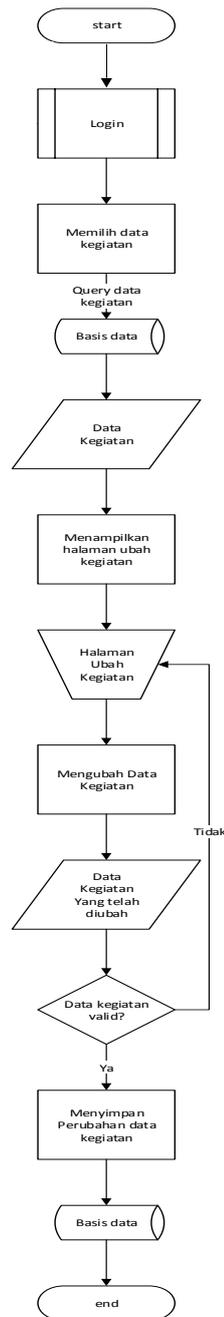
**GAMBAR 3.3** Sistem Usulan Halaman *Login*

Sistem usulan tambah kegiatan menggambarkan alur proses yang ditampilkan ketika pengguna aplikasi ingin menambahkan kegiatan baru pada aplikasi. Adapun sistem usulan tambah kegiatan dapat dilihat pada gambar berikut:



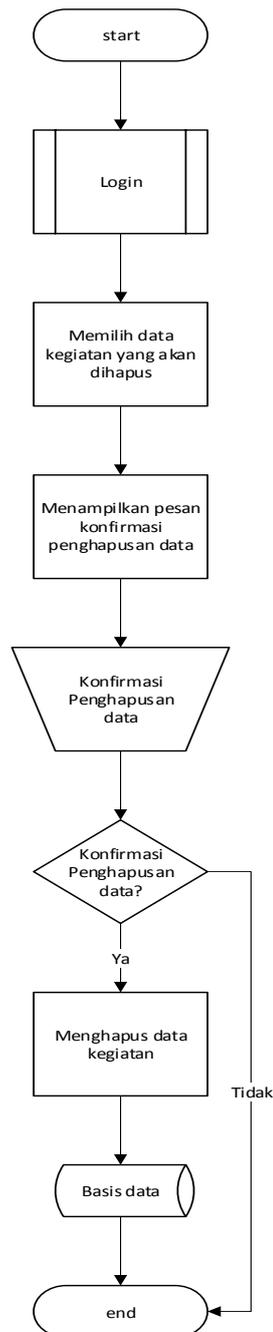
**GAMBAR 3.4 Sistem Usulan Tambah Kegiatan**

Sistem usulan ubah kegiatan menggambarkan alur proses yang ditampilkan ketika pengguna aplikasi ingin mengubah kegiatan yang telah terdaftar pada aplikasi. Adapun sistem usulan ubah kegiatan dapat dilihat pada gambar berikut:



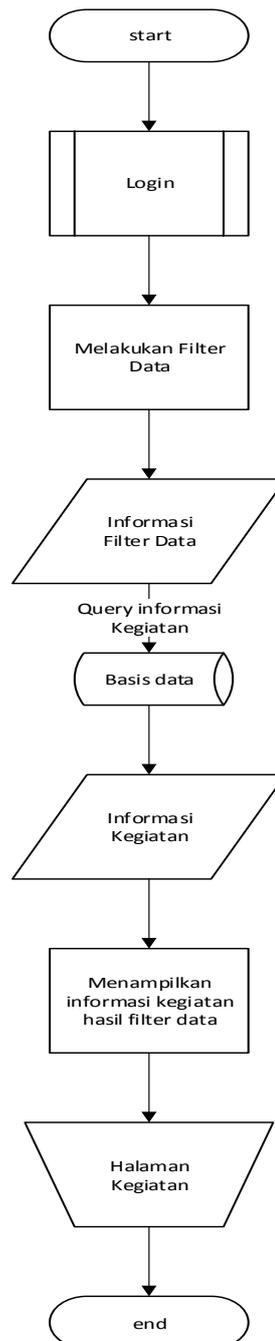
**GAMBAR 3.5 Sistem Usulan Ubah Kegiatan**

Sistem usulan hapus kegiatan menggambarkan alur proses yang ditampilkan ketika pengguna aplikasi ingin menghapus kegiatan yang telah terdaftar pada aplikasi. Adapun sistem usulan hapus kegiatan dapat dilihat pada gambar berikut:



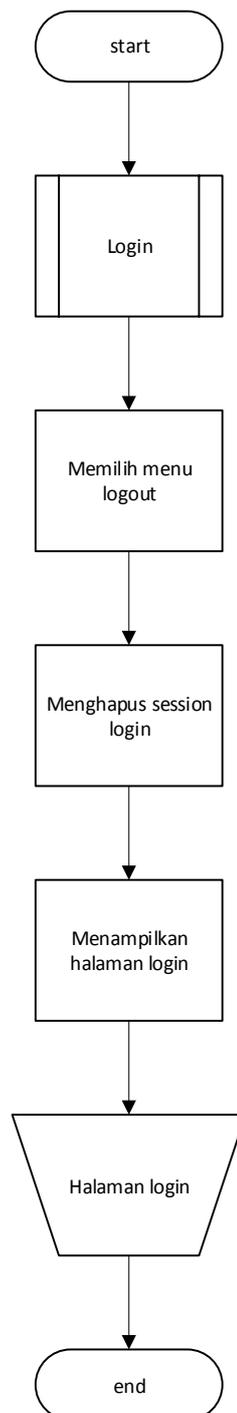
**GAMBAR 3.6 Sistem Usulan Hapus Kegiatan**

Sistem usulan filter data menggambarkan alur proses yang ditampilkan ketika pengguna aplikasi ingin melakukan penyaringan data kegiatan yang telah terdaftar pada aplikasi. Adapun sistem usulan filter data dapat dilihat pada gambar berikut:



**GAMBAR 3.7 Sistem Usulan Filter Data**

Sistem usulan *logout* menggambarkan alur proses yang ditampilkan ketika pengguna aplikasi ingin keluar dari aplikasi. Adapun sistem usulan *logout* dapat dilihat pada gambar berikut:



**GAMBAR 3.8 Sistem Usulan *logout***

## 3.2 Desain

### 3.2.1 Perancangan Program

Pada perancangan program dalam proses pembangunan aplikasi ini, penulis menggunakan diagram *Use case* dan *Sequence* diagram, kebutuhan perangkat lunak dan perangkat keras, kemudian yang terakhir adalah Perancangan Antarmuka (*Interface*).

### 3.2.2 Definisi Aktor

**TABEL 3. 2 Deskripsi Aktor**

Aktor	Deskripsi
Admin	Aktor yang memiliki otoritas untuk mengelola data kegiatan pada aplikasi seperti menambah, mengubah, menghapus dan mencari data kegiatan. Aktor dengan otoritas ini akan diberikan kepada TU Pimpinan serta asisten pimpinan di kantor BAPPEDA Provinsi Jawa Barat
User	Aktor yang dapat mengakses aplikasi menggunakan perangkat yang dimiliki namun hanya dapat melihat data kegiatan mingguan pada halaman utama

### 3.2.3 Langkah Aplikasi

Langkah aplikasi merupakan penjelasan dalam setiap melakukan aktivitas pada aplikasi *meeting arranger*, langkah dimulai dari tahapan pertama sampai tahapan terakhir dengan penjelasan sebagai berikut:

#### a) *Use Case* Melihat Kegiatan

Aktor : User dan Admin

Tujuan : Aplikasi dapat menampilkan daftar kegiatan mingguan pada perangkat yang digunakan oleh aktor

Deskripsi : Aktor mengakses alamat aplikasi pada browser web kemudian aplikasi akan menampilkan daftar kegiatan mingguan

#### b) *Use Case* Login

Aktor : Admin

Tujuan : Melakukan verifikasi data akses aktor berupa username dan password sebelum masuk kedalam aplikasi sebagai fitur keamanan aplikasi

Deskripsi : Aktor memasukan username dan password yang valid untuk masuk kedalam halaman admin aplikasi

#### c) *Use Case* Tambah Kegiatan

Aktor : Admin

Tujuan : Menambahkan kegiatan kedalam database aplikasi

Deskripsi : Aktor memilih menu “kegiatan” kemudian menekan tombol “Tambah Kegiatan” dan mengisi formulir kegiatan yang didalamnya terdapat informasi yang harus diisi

**d) Use Case Ubah Kegiatan**

Aktor : Admin

Tujuan : Mengubah informasi kegiatan yang telah tercatat pada database aplikasi

Deskripsi : Aktor memilih menekan tombol “ubah” kemudian mengubah informasi kegiatan sesuai dengan yang diharapkan

**e) Use Case Hapus Kegiatan**

Aktor : Admin

Tujuan : Menghapus kegiatan dari database aplikasi

Deskripsi : Aktor memilih menekan tombol “ubah” kemudian mengubah informasi kegiatan sesuai dengan yang diharapkan

**f) Use Case Filter Kegiatan**

Aktor : Admin

Tujuan : Menyaring data kegiatan sesuai dengan informasi

yang diinginkan

Deskripsi : Aktor menekan tombol pada kolom tabel yang berisi informasi kegiatan kemudian memasukan informasi filter yang akan digunakan sehingga data kegiatan pada tabel akan tampil sesuai dengan informasi yang diinginkan

**g) Use Case Cari Kegiatan**

Aktor : Admin

Tujuan : Mencari kegiatan berdasarkan informasi yang diinginkan

Deskripsi : Aktor memasukan informasi pada kolom pencarian sehingga data kegiatan akan tampil sesuai dengan keinginan aktor

**h) Use Case Logout**

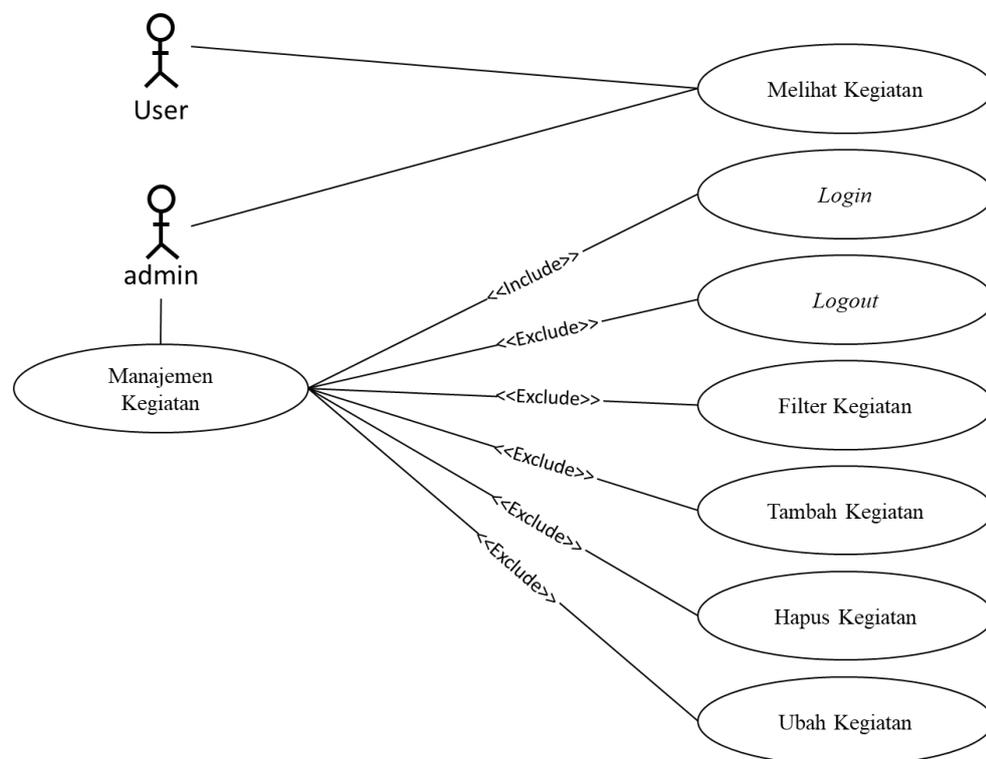
Aktor : Admin

Tujuan : Keluar dari aplikasi dan menghapus session sebagai fitur keamanan aplikasi

Deskripsi : Aktor menekan tombol *logout* kemudian aplikasi akan menghapus session dan menampilkan halaman *login*

### 3.2.4 Use Case Diagram Meeting Arranger

Use case diagram merupakan suatu diagram yang menggambarkan fungsionalitas yang dimiliki oleh suatu sistem beserta aktor-aktor yang terlibat di dalamnya. Adapun fungsionalitas dan aktor yang terlibat dalam aplikasi ini dapat dilihat pada GAMBAR 3.9 Use Case Diagram



**GAMBAR 3.9 Use Case Diagram**

### 3.2.5 Skenario *Use Case*

#### a) *Use Case Scenario* Melihat Kegiatan

- Aktor : *User* dan *Admin*
- Tujuan : Aplikasi dapat menampilkan daftar kegiatan mingguan pada perangkat yang digunakan oleh aktor
- Deskripsi : Aktor mengakses alamat aplikasi pada browser web kemudian aplikasi akan menampilkan daftar kegiatan mingguan
- Asumsi : Aktor memiliki aplikasi *browser* web

**TABEL 3.3** *Use Case Scenario* Melihat Kegiatan

Aktor	Sistem
1. Aktor ( <i>user/admin</i> ) mengakese aplikasi dengan cara memasukan alamat aplikasi pada <i>browser</i> web	2. Aplikasi melakukan <i>routing</i> halaman utama
	3. Aplikasi melakukan <i>query</i> data kegiatan dalam satu minggu
	4. Aplikasi menampilkan hasil <i>query</i> kegiatan mingguan pada halaman utama

**b) Use Case Scenario Login**

- Aktor : *Admin*
- Tujuan : Melakukan verifikasi data akses aktor berupa *username* dan *password* sebelum masuk kedalam aplikasi sebagai fitur keamanan aplikasi
- Deskripsi : Aktor memasukan *username* dan *password* yang valid untuk masuk kedalam halaman admin aplikasi
- Asumsi : Aktor belum melakukan *login* kedalam aplikasi

**TABEL 3.4 Use Case Scenario login**

Aktor	Sistem
<b>Skenario Berhasil Login</b>	
1. Aktor mengakses halaman login dengan menggunakan <i>browser</i> web	
2. Aktor mengisi dan mengirimkan <i>username</i> serta <i>password</i> pada formulir <i>login</i>	3. Aplikasi melakukan validasi <i>username</i> dan <i>password</i> pada basis data aplikasi
	4. Aplikasi berhasil melakukan autentikasi <i>username</i> dan <i>password</i> yang dikirimkan oleh aktor

Aktor	Sistem
	5. Aplikasi menampilkan halaman administrator
<b>Skenario Gagal Login Karena Username dan Password Bernilai Kosong</b>	
1. Aktor mengakses halaman login dengan menggunakan <i>browser</i> web	
2. Aktor tidak memasukan <i>username</i> serta <i>password</i> pada formulir <i>login</i> kemudian menekan tombol <i>login</i>	
	3. Aplikasi melakukan validasi nilai masukan <i>username</i> dan <i>password</i>
	4. Aplikasi mendeteksi <i>username</i> dan <i>password</i> yang dimasukan bernilai kosong
5. Aplikasi menampilkan halaman <i>login</i> dengan pesan yang menginformasikan bahwa <i>username</i> dan <i>password</i> harus diisi	
<b>Skenario Gagal Login Karena Username Tidak Terdaftar di basis data</b>	
1. Aktor mengakses halaman login dengan menggunakan <i>browser</i> web	
2. Aktor memasukan <i>username</i> serta <i>password</i> pada formulir	

Aktor	Sistem
<i>login</i> kemudian menekan tombol <i>login</i>	
	3. Aplikasi melakukan validasi nilai masukan <i>username</i> dan <i>password</i>
	4. Aplikasi gagal melakukan autentikasi <i>username</i> dan <i>password</i> pada basis data aplikasi
	5. Aplikasi menampilkan halaman <i>login</i> dengan pesan yang menginformasikan bahwa <i>username</i> atau <i>password</i> yang dikirimkan tidak terdaftar pada basis data aplikasi

c) **Use Case Scenario Tambah Kegiatan**

Aktor : Admin

Tujuan : Menambahkan kegiatan kedalam database aplikasi

Deskripsi : Aktor memilih menu “kegiatan” kemudian menekan tombol “Tambah Kegiatan” dan mengisi formulir kegiatan yang didalamnya terdapat informasi yang harus diisi

Asumsi : Aktor telah melakukan *login* aplikasi

**TABEL 3.5 Use Case Scenario Tambah Kegiatan**

Aktor	Sistem
<b>Skenario Berhasil Menambahkan Kegiatan</b>	
1. Aktor mengakses halaman tambah kegiatan	
2. Aktor mengisi formulir kegiatan berupa asal surat, nama kegiatan, lokasi, tanggal kegiatan, waktu, disposisi, keterangan dihadiri kepala badan, keterangan peliputan dan keterangan ditampilkan pada layar kemudian mengirimkan informasi kegiatan untuk disimpan	
	3. Aplikasi melakukan validasi nilai masukan informasi kegiatan
	4. Nilai masukan informasi kegiatan berhasil divalidasi
	5. Aplikasi menyimpan data informasi kegiatan kedalam basis data
	6. Aplikasi menampilkan halaman tambah kegiatan dengan pesan yang menyampaikan bahwa informasi kegiatan

Aktor	Sistem
	berhasil disimpan kedalam basis data
<b>Skenario Gagal Menambah Kegiatan Karena Informasi Kegiatan Tidak Sesuai</b>	
1. Aktor mengakses halaman tambah kegiatan	
2. Aktor mengisi formulir kegiatan berupa asal surat, nama kegiatan, lokasi, tanggal kegiatan, waktu, disposisi, keterangan dihadiri kepala badan, keterangan peliputan dan keterangan ditampilkan pada layar dengan nilai yang tidak sesuai kemudian mengirimkan informasi kegiatan untuk disimpan	
	3. Aplikasi melakukan validasi nilai masukan informasi kegiatan
	4. Nilai masukan informasi kegiatan gagal divalidasi
	5. Aplikasi menampilkan halaman tambah kegiatan dengan pesan validasi gagal

**d) Use Case Scenario Ubah Kegiatan**

- Aktor : Admin
- Tujuan : Mengubah informasi kegiatan yang telah tercatat pada database aplikasi
- Deskripsi : Aktor memilih menekan tombol “ubah” kemudian mengubah informasi kegiatan sesuai dengan yang diharapkan
- Asumsi : 1. Aktor telah melakukan *login* aplikasi  
2. Terdapat data informasi kegiatan yang telah tersimpan di dalam basis data

**TABEL 3. 6 Use Case Scenario Ubah Kegiatan**

Aktor	Sistem
<b>Skenario Berhasil Mengubah Kegiatan</b>	
1. Aktor memilih informasi kegiatan yang akan diubah dengan cara menekan tombol ubah pada informasi kegiatan yang ditampilkan pada halaman kegiatan	
	2. Aplikasi menampilkan halaman ubah informasi kegiatan

Aktor	Sistem
<p>3. Aktor mengubah data kegiatan dengan informasi yang sesuai pada halaman ubah kegiatan kemudian mengirimkan data kegiatan yang telah diubah</p>	
	<p>4. Aplikasi melakukan validasi nilai masukan informasi kegiatan yang telah diubah</p>
	<p>5. Nilai masukan informasi kegiatan yang telah diubah berhasil divalidasi</p>
	<p>6. Aplikasi menyimpan perubahan informasi kegiatan kedalam basis data</p>
	<p>7. Aplikasi menampilkan halaman ubah kegiatan dengan informasi yang menyampaikan bahwa data kegiatan berhasil diubah</p>
<b>Skenario Berhasil Mengubah Kegiatan</b>	
<p>1. Aktor memilih informasi kegiatan yang akan diubah dengan cara menekan tombol ubah pada informasi kegiatan yang ditampilkan pada halaman kegiatan</p>	

Aktor	Sistem
	2. Aplikasi menampilkan halaman ubah informasi kegiatan
3. Aktor mengubah data kegiatan dengan informasi yang tidak sesuai pada halaman ubah kegiatan kemudian mengirimkan data kegiatan yang telah diubah	
	4. Aplikasi melakukan validasi nilai masukan informasi kegiatan yang telah diubah
	5. Nilai masukan informasi kegiatan yang telah diubah gagal divalidasi
	6. Aplikasi menampilkan halaman ubah kegiatan dengan pesan kesalahan validasi <i>input</i> nilai

e) **Use Case Hapus Kegiatan**

Aktor : Admin

Tujuan : Menghapus kegiatan dari database aplikasi

Deskripsi : Aktor memilih menekan tombol “ubah” kemudian mengubah informasi kegiatan sesuai dengan yang diharapkan

Asumsi : 1. Aktor telah melakukan *login* aplikasi

2. Terdapat data informasi kegiatan yang telah tersimpan di dalam basis data

**TABEL 3. 7 Use Case Scenario Hapus Kegiatan**

Aktor	Sistem
<b>Skenario Berhasil Menghapus Kegiatan</b>	
1. Aktor memilih informasi kegiatan yang akan dihapus dengan cara menekan tombol hapus pada informasi kegiatan yang ditampilkan pada halaman kegiatan	2. Aplikasi menampilkan pesan konfirmasi penghapusan data
3. Aktor memberikan konfirmasi penghapusan data dengan menekan tombol Yes	4. Aplikasi menghapus data informasi kegiatan dari basis data 5. Aplikasi menampilkan ulang halaman kegiatan
<b>Skenario Berhasil Menghapus Kegiatan</b>	
1. Aktor memilih informasi kegiatan yang akan dihapus dengan cara menekan tombol	

Aktor	Sistem
hapus pada informasi kegiatan yang ditampilkan pada halaman kegiatan	
	2. Aplikasi menampilkan pesan konfirmasi penghapusan data
3. Aktor membatalkan penghapusan data dengan menekan tombol No	
	4. Aplikasi menampilkan ulang halaman kegiatan

**f) Use Case Scenario Filter Kegiatan**

Aktor : Admin

Tujuan : Menyaring data kegiatan sesuai dengan informasi yang diinginkan

Deskripsi : Aktor menekan tombol pada kolom tabel yang berisi informasi kegiatan kemudian memasukan informasi filter yang akan digunakan sehingga data kegiatan pada tabel akan tampil sesuai dengan informasi yang diinginkan

Asumsi : 3. Aktor telah melakukan *login* aplikasi  
4. Terdapat data informasi kegiatan yang telah tersimpan di dalam basis data

**TABEL 3. 8 Use Case Scenario Filter Kegiatan**

Aktor	Sistem
1. Aktor menekan tombol ikon <i>filter</i> pada kolom tabel informasi kegiatan pada halaman kegiatan	
2. Aktor memasukan nilai informasi yang diinginkan pada kolom filter yang muncul kemudian menekan tombol apply	
	3. Aplikasi melakukan query informasi kegiatan sesuai dengan nilai masukan pada kolom filter data
	4. Aplikasi menampilkan informasi hasil penyaringan data pada halaman kegiatan

**g) Use Case Scenario Cari Kegiatan**

Aktor : Admin

Tujuan : Mencari kegiatan berdasarkan informasi yang diinginkan

Deskripsi : Aktor memasukan informasi pada kolom pencarian sehingga data kegiatan akan tampil sesuai dengan keinginan aktor

Asumsi : 5. Aktor telah melakukan *login* aplikasi

6. Terdapat data informasi kegiatan yang telah tersimpan di dalam basis data

**TABEL 3. 9 Use Case Scenario Cari Kegiatan**

Aktor	Sistem
1. Aktor memasukan nilai informasi kegiatan yang akan dicari pada kolom pencarian di halaman kegiatan	
	2. Aplikasi melakukan query informasi kegiatan sesuai dengan nilai masukan informasi kegiatan
	3. Aplikasi menampilkan data kegiatan sesuai dengan informasi yang diinginkan pada halaman kegiatan

**h) Use Case Scenario Logout**

Aktor : Admin

Tujuan : Keluar dari aplikasi dan menghapus session sebagai fitur keamanan aplikasi

Deskripsi : Aktor menekan tombol *logout* kemudian aplikasi akan menghapus session dan menampilkan halaman

*login*

Asumsi : Aktor telah melakukan *login* aplikasi

**TABEL 3. 10 Use Case Scenario Logout Kegiatan**

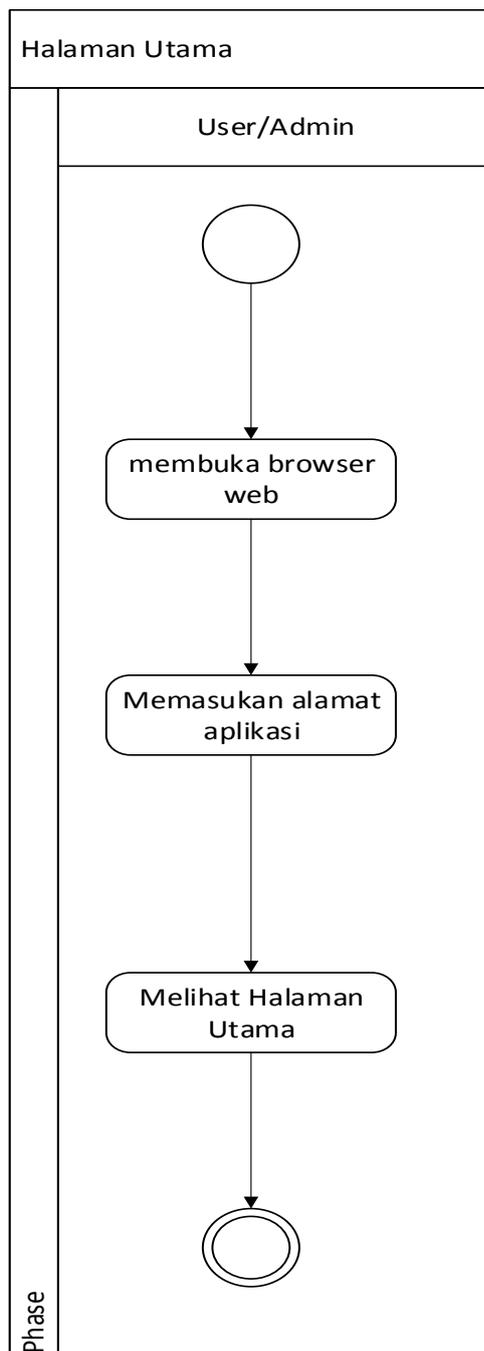
Aktor	Sistem
1. Aktor menekan tombol <i>logout</i> yang tersedia pada menu utama halaman administrator	
	2. Aplikasi melakukan penghapusan <i>session login</i>
	3. Aplikasi menampilkan halaman <i>login</i>

### 3.2.6 Activity Diagram

Activity diagram menggambarkan aktivitas interaksi yang dilakukan oleh aktor dengan sistem. Pembahasan aktivitas diagram dibagi berdasarkan fitur-fitur utama dari aplikasi ini yang telah di gambarkan pada diagram *use case* sebelumnya.

#### a) Activity Diagram Halaman Utama

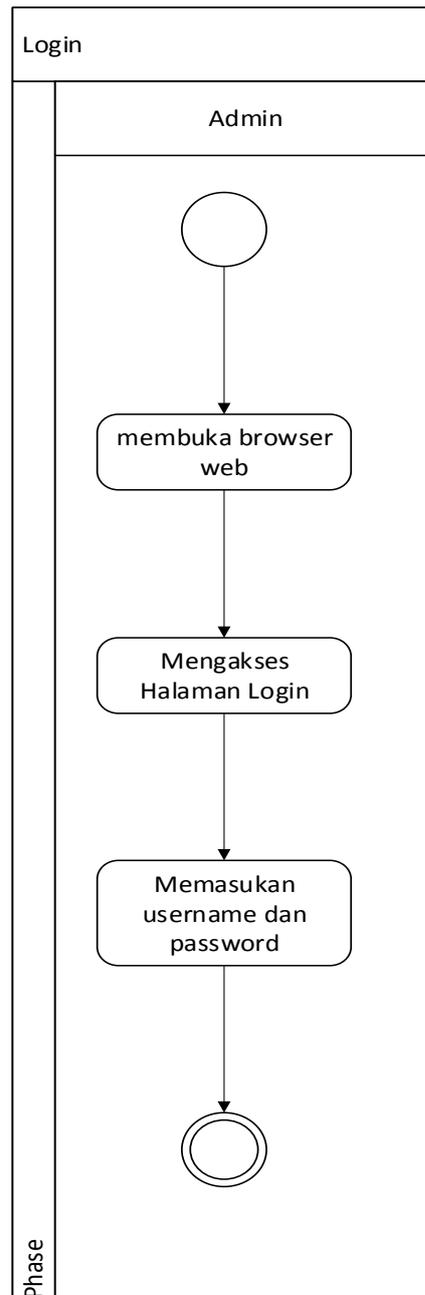
Diagram aktivitas halaman utama menggambarkan aktivitas yang dilakukan oleh pengguna aplikasi ketika mengakses halaman utama aplikasi. Adapun aktivitas pada halaman ini dapat dilihat pada gambar berikut:



**GAMBAR 3.10** *Activity Diagram Halaman Utama*

**b) Activity Diagram Login**

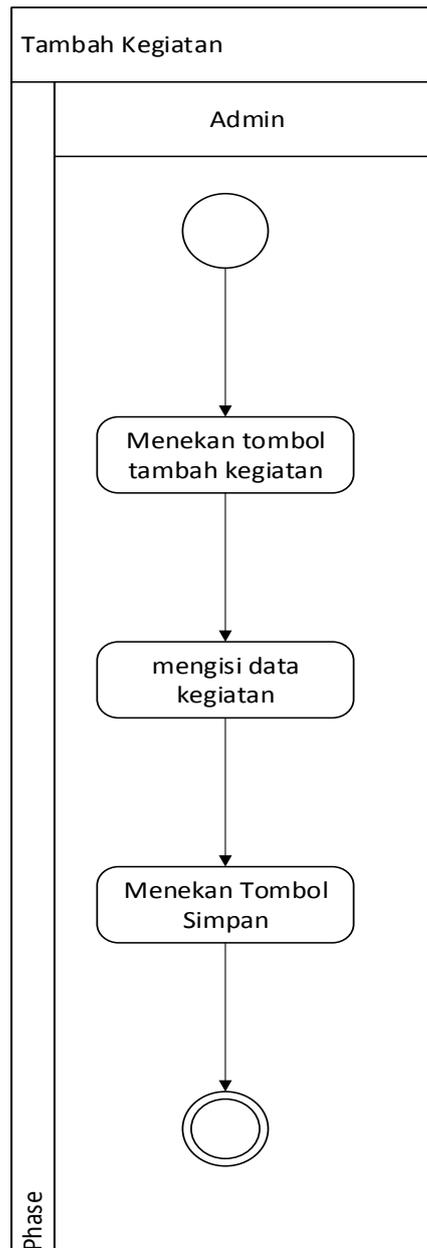
Diagram aktivitas *login* menggambarkan aktivitas yang dilakukan oleh pengguna aplikasi ketika ingin mengakses halaman administrator. Adapun aktivitas ini dapat dilihat pada gambar berikut:



**GAMBAR 3.11 Activity Diagram Login**

c) **Activity Diagram Tambah Kegiatan**

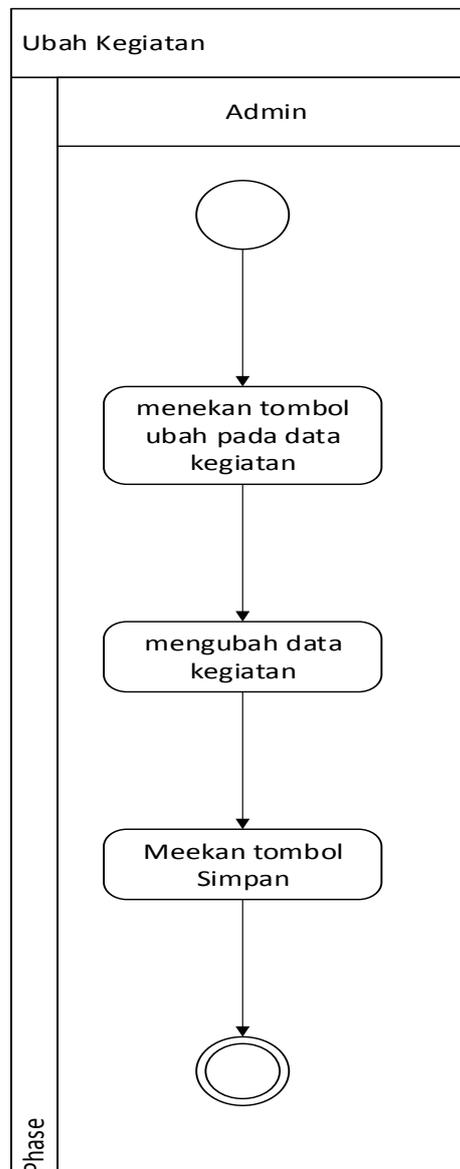
Diagram aktivitas tambah kegiatan menggambarkan aktivitas yang dilakukan oleh pengguna aplikasi ketika ingin menambah kegiatan kedalam aplikasi. Adapun aktivitas ini dapat dilihat pada gambar berikut:



**GAMBAR 3.12 Activity Diagram Tambah Kegiatan**

**d) Activity Diagram Ubah Kegiatan**

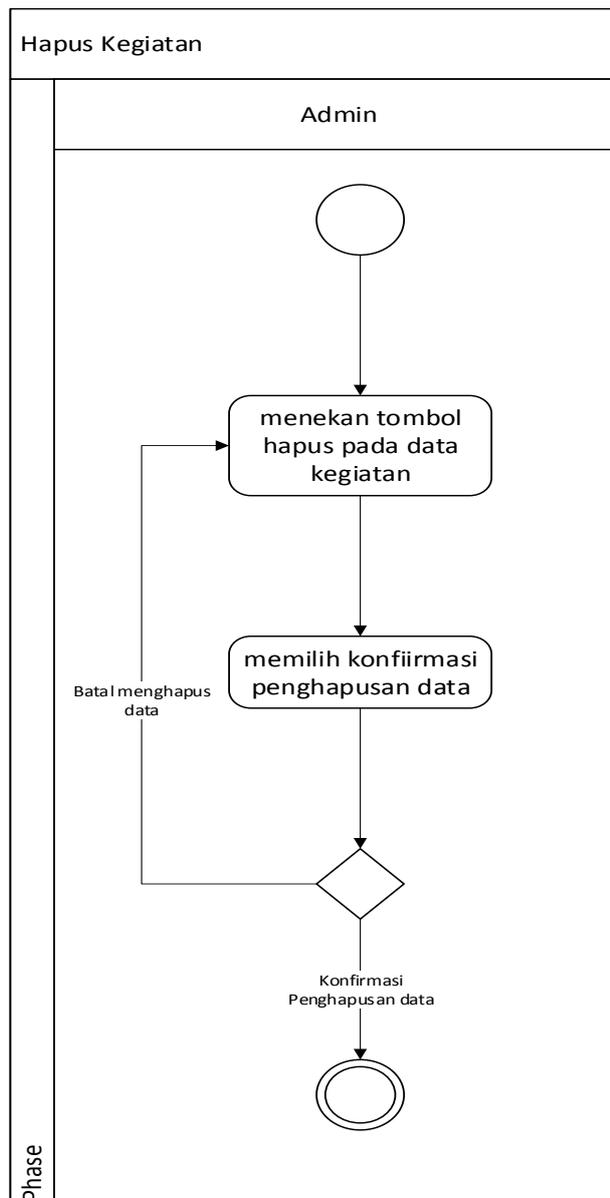
Diagram aktivitas ubah kegiatan menggambarkan aktivitas yang dilakukan oleh pengguna aplikasi ketika ingin mengubah kegiatan yang telah terdaftar di aplikasi. Adapun aktivitas ini dapat dilihat pada gambar berikut:



**GAMBAR 3.13 Activity Diagram Ubah Kegiatan**

e) **Activity Diagram Hapus Kegiatan**

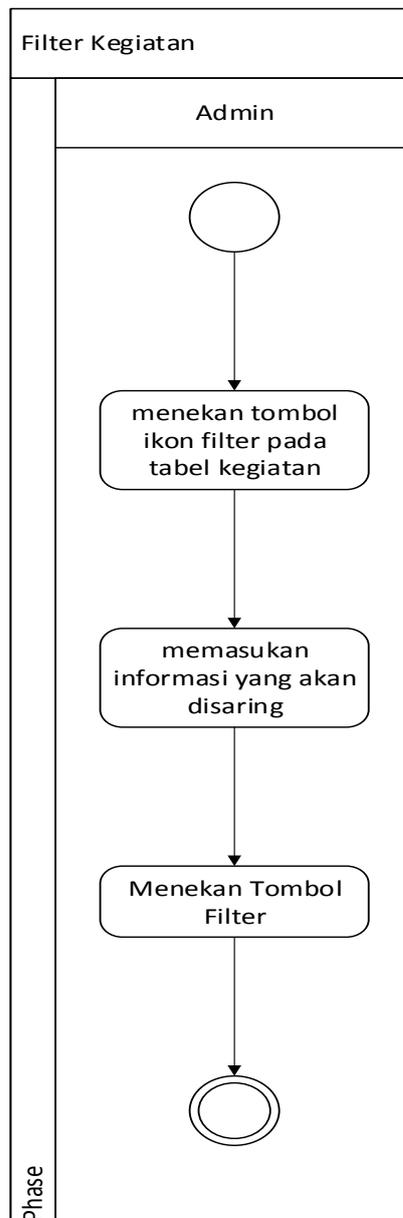
Diagram aktivitas hapus kegiatan menggambarkan aktivitas yang dilakukan oleh pengguna aplikasi ketika ingin menghapus kegiatan yang telah terdaftar di aplikasi. Adapun aktivitas ini dapat dilihat pada gambar berikut:



**GAMBAR 3.14 Activity Diagram Hapus Kegiatan**

f) **Activity Diagram Filter Kegiatan**

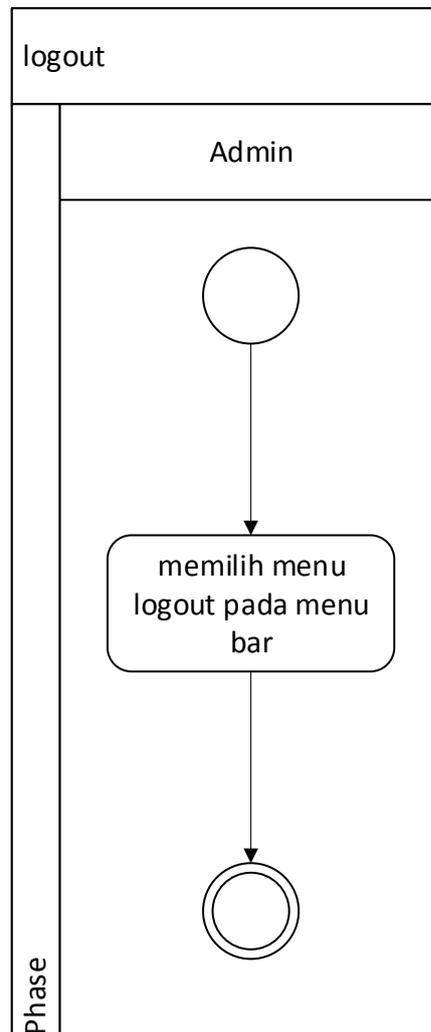
Diagram aktivitas filter kegiatan menggambarkan aktivitas yang dilakukan oleh pengguna aplikasi ketika ingin menyaring informasi kegiatan yang telah terdaftar di aplikasi. Adapun aktivitas ini dapat dilihat pada gambar berikut:



**GAMBAR 3.15 Activity Diagram Filter Kegiatan**

g) **Activity Diagram Logout**

Diagram aktivitas *logout* menggambarkan aktivitas yang dilakukan oleh pengguna aplikasi ketika ingin keluar dari aplikasi. Adapun aktivitas ini dapat dilihat pada gambar berikut:



**GAMBAR 3.16 Activity Diagram Logout**

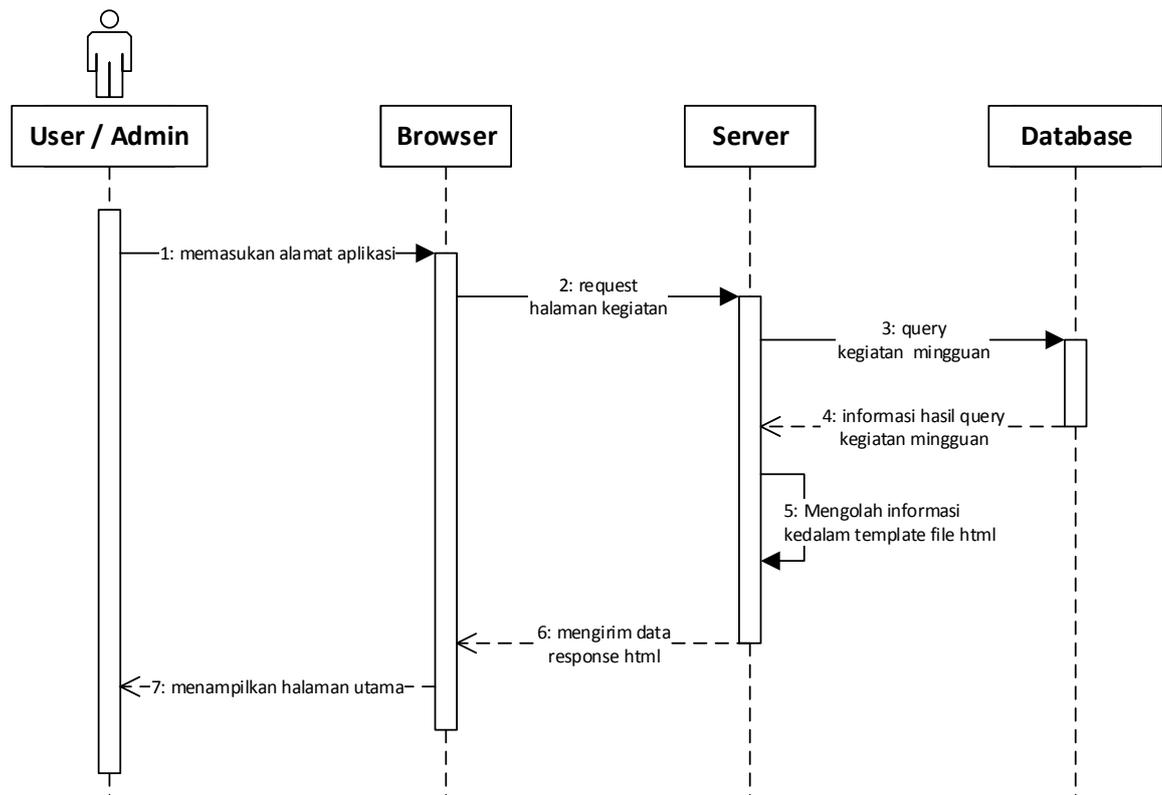
### 3.2.7 Sequence Diagram

*Sequence diagram* menggambarkan perilaku objek dalam *use case* dengan menjelaskan siklus hidup objek dan pesan yang dikirim dan diterima di antara objek. Oleh karena itu, untuk menggambar diagram urutan, Anda harus mengetahui objek yang terlibat dalam kasus penggunaan dan metode yang termasuk dalam kelas yang digunakan oleh objek tersebut.

Jumlah *Sequence* diagram yang harus digambar adalah sebanyak definisi *use case* dan alurnya masing-masing. Yang terpenting adalah semua *use case* telah diidentifikasi dan interaksi jalur pesan dimasukkan ke dalam *Sequence* diagram. Oleh karena itu, semakin banyak *use case* teridentifikasi, semakin banyak pula operasi yang harus dilakukan. *Sequence diagram* pada sistem ini sebagai berikut:

#### a) *Sequence Diagram* Melihat Kegiatan

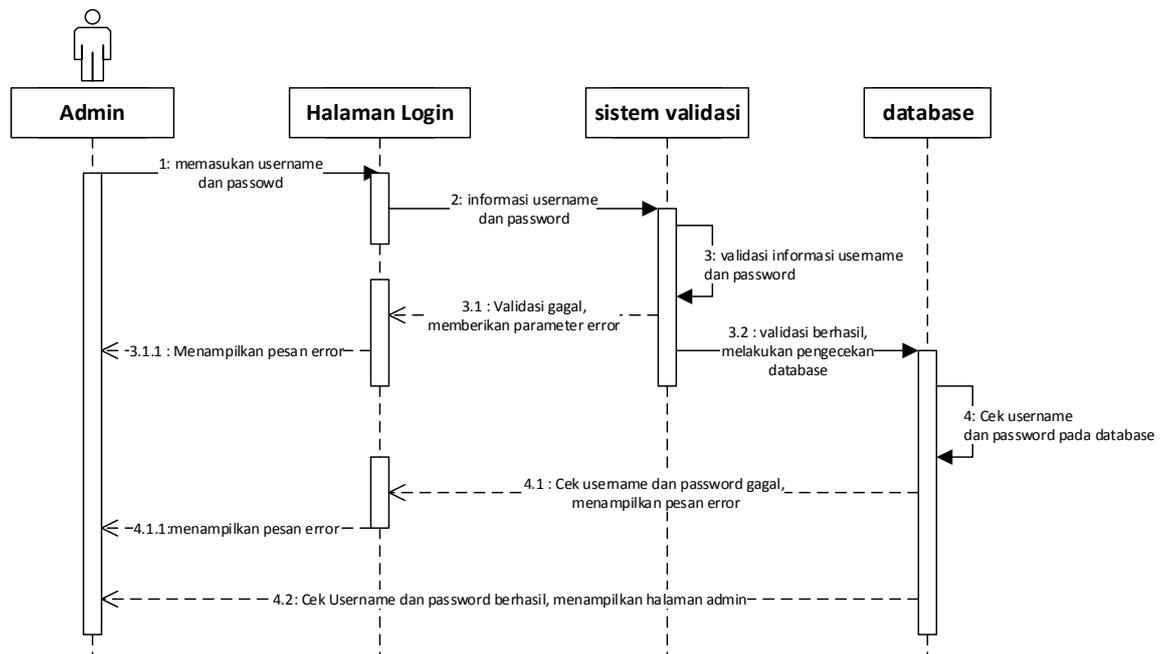
*Sequence Diagram* Melihat Kegiatan menggambarkan proses pengaksesan halaman muka aplikasi yang dapat dilakukan oleh User maupun Admin. Adapun tahapan yang dilalui dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.17** *Sequence Diagram* Melihat Kegiatan

**b) *Sequence Diagram Login***

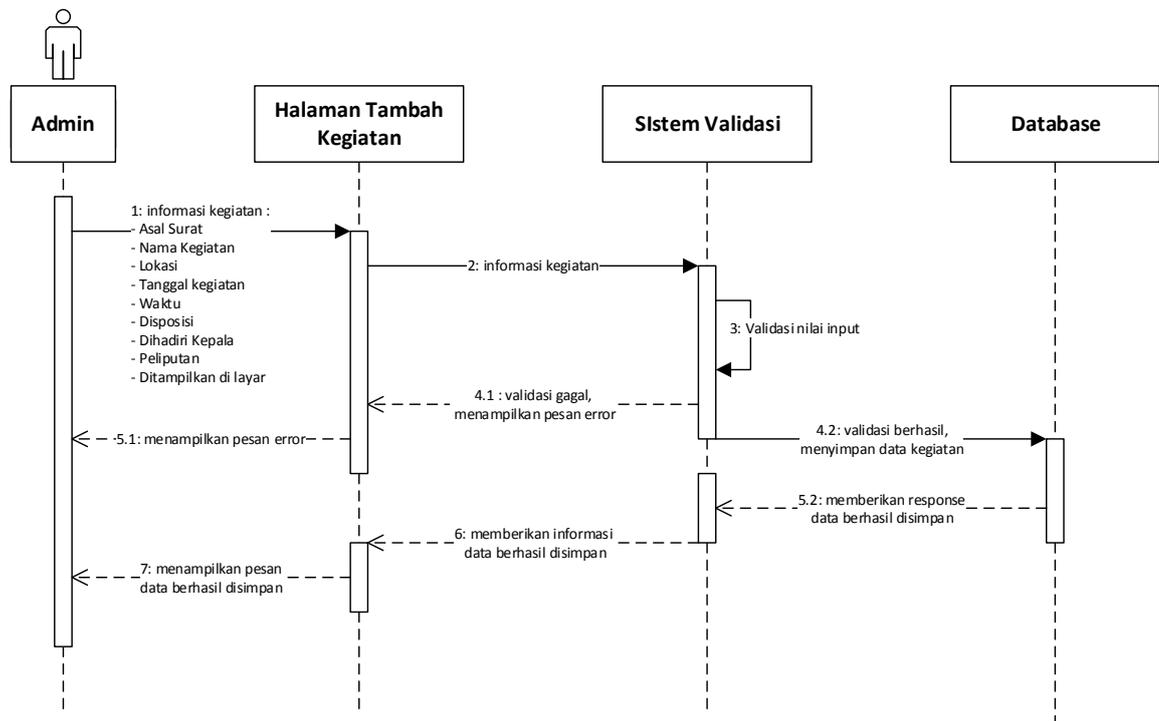
*Sequence Diagram Login* menggambarkan proses validasi *username* dan *password* yang dilakukan oleh aplikasi ketika pengguna aplikasi ingin mengakses halaman admin. Adapun tahapan yang dilalui dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.18** *Sequence Diagram Login*

**c) Sequence Diagram Tambah Kegiatan**

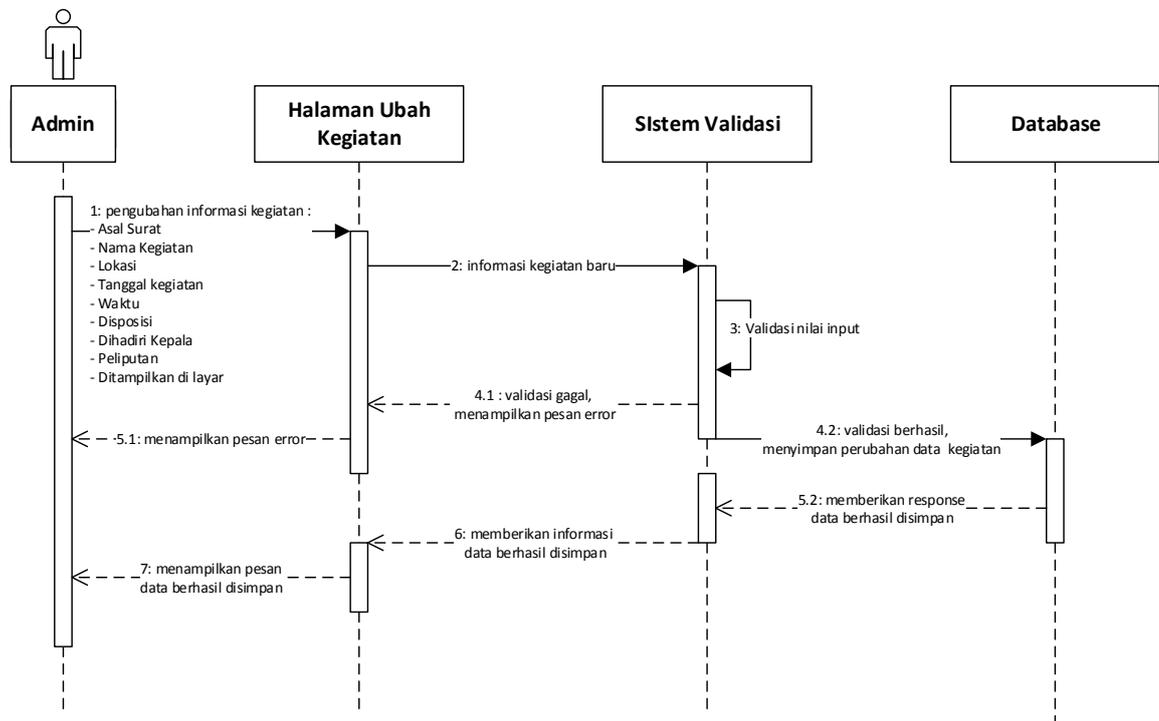
*Sequence Diagram Tambah kegiatan* menggambarkan proses penambahan data kegiatan ke dalam database dengan asumsi bahwa aktor telah berhasil *login* ke dalam halaman admin dan memilih menu tambah data kegiatan. Adapun tahapan yang dilalui dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.19** *Sequence Diagram* Tambah Kegiatan

**d) *Sequence Diagram* Ubah Kegiatan**

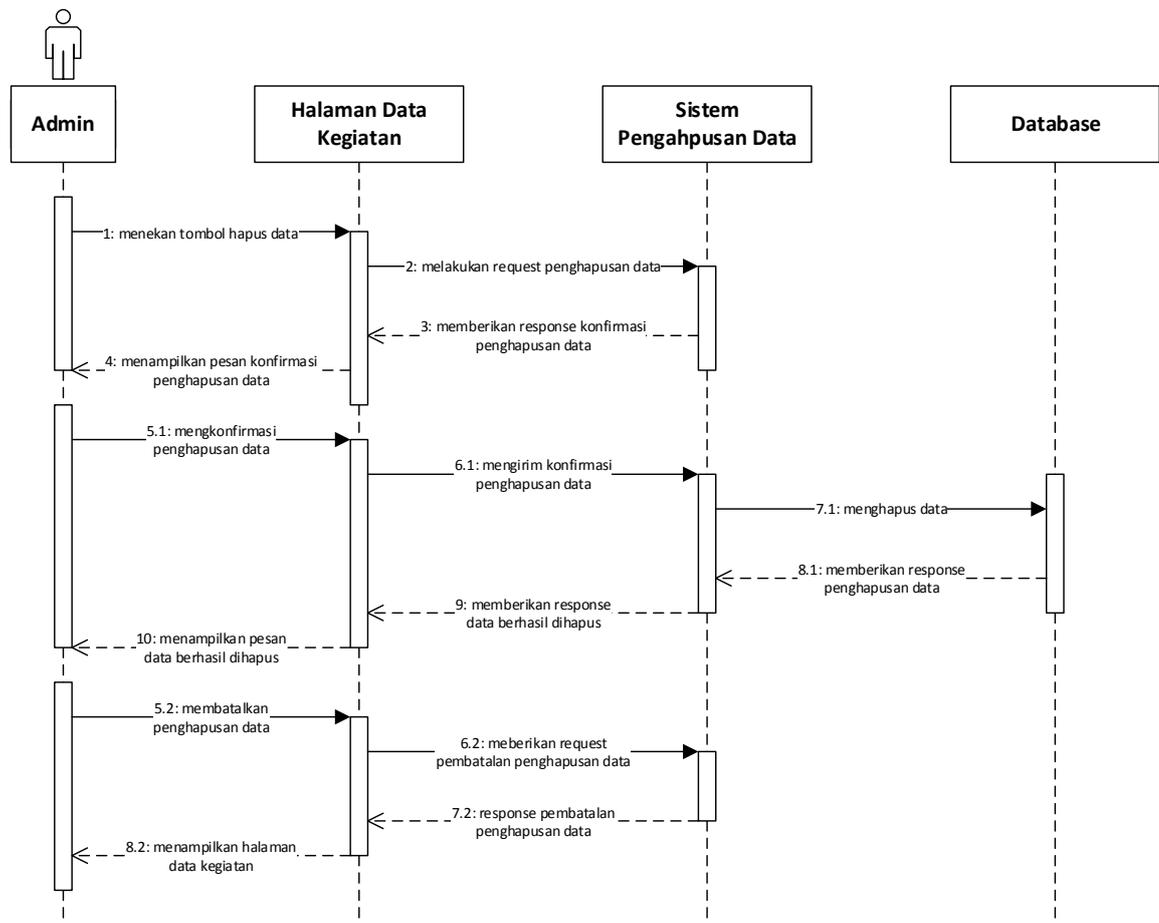
*Sequence Diagram* Ubah kegiatan menggambarkan proses pengubahan data kegiatan yang telah ada kemudian menyimpannya kedalam database dengan asumsi bahwa aktor telah berhasil *login* kedalam halaman admin dan memilih data kegiatan yang akan diubah. Adapun tahapan yang dilalui dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.20** *Sequence Diagram* Ubah Kegiatan

e) **Sequence Diagram** Hapus Kegiatan

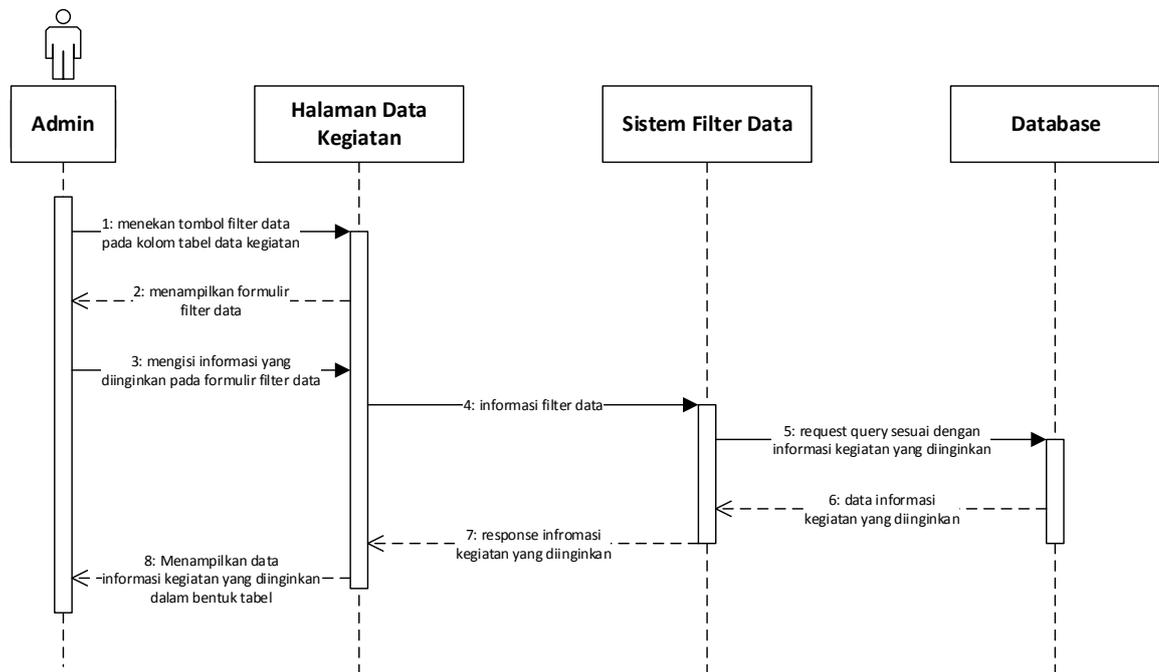
*Sequence Diagram* Hapus Kegiatan menggambarkan proses penghapusan data kegiatan yang telah ada dari database dengan asumsi bahwa aktor telah berhasil *login* kedalam halaman admin dan memilih data kegiatan yang akan dihapus. Adapun tahapan yang dilalui dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.21** *Sequence Diagram Hapus Kegiatan*

f) *Sequence Diagram Filter Kegiatan*

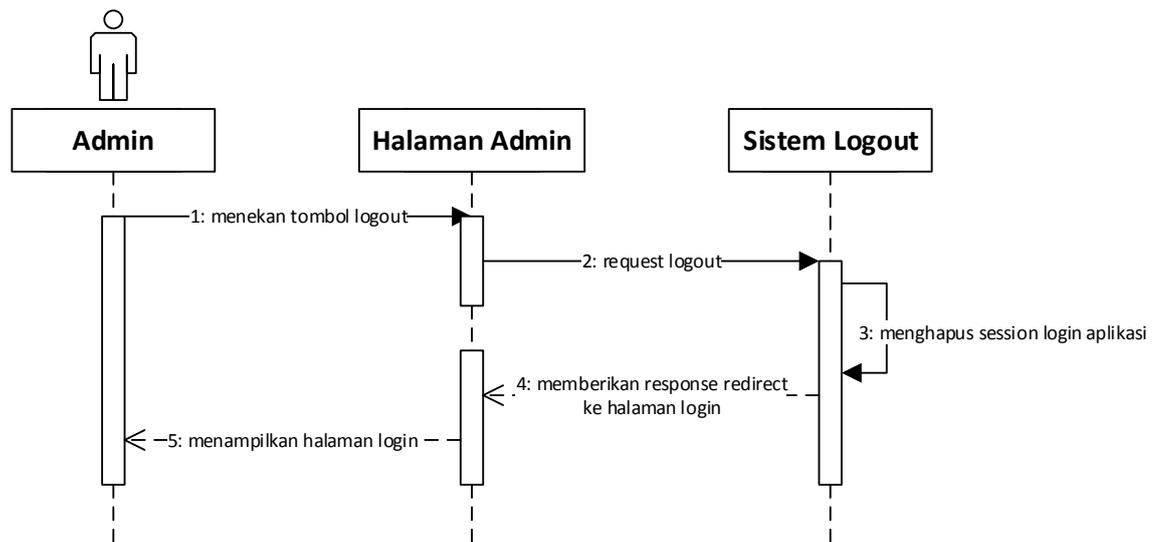
*Sequence Diagram Hapus Kegiatan* menggambarkan proses penghapusan data kegiatan yang telah ada dari database dengan asumsi bahwa aktor telah berhasil *login* kedalam halaman admin dan memilih data kegiatan yang akan dihapus. Adapun tahapan yang dilalui dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.22** *Sequence Diagram Filter Kegiatan*

**g)** *Sequence Diagram Logout*

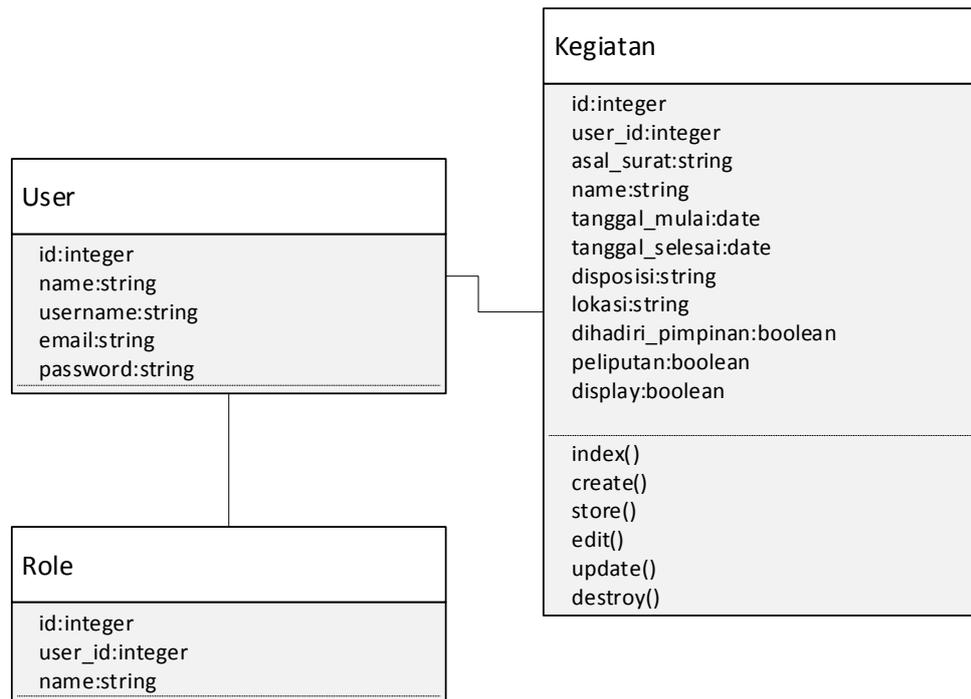
*Sequence Diagram Logout* menggambarkan proses penghapusan *session login* yang dimiliki oleh aktor dari *browser session* dengan asumsi bahwa aktor telah berhasil *login* ke dalam halaman admin dan menekan tombol *logout*. Adapun tahapan yang dilalui dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.23** *Sequence Diagram Logout*

### 3.2.8 Class Diagram

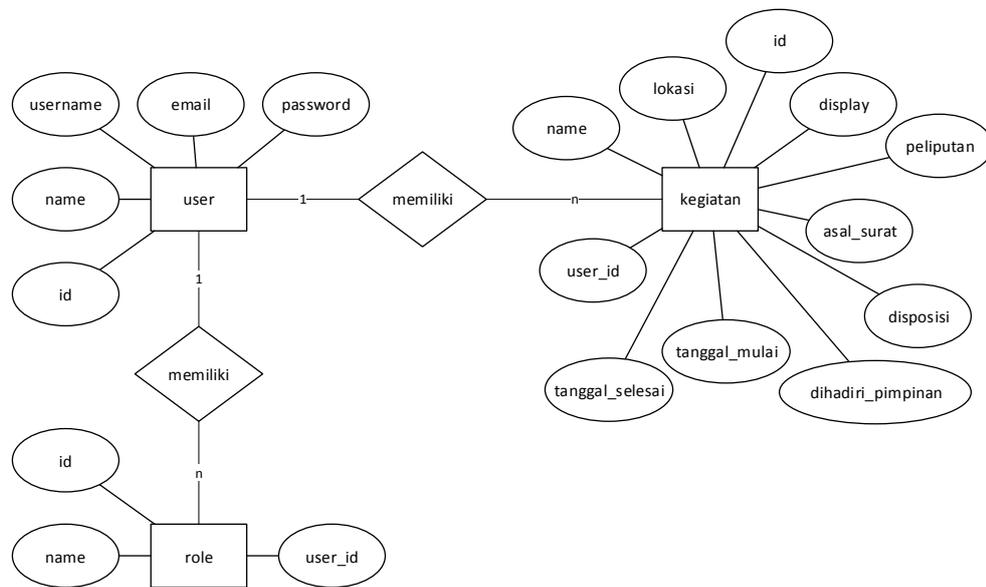
Aplikasi ini dibangun dengan konsep *Object Oriented* sehingga memiliki beberapa *class* utama sebagai dasar pengelolaan kegiatan. Struktur *class* yang akan diimplementasikan pada aplikasi ini dapat dilihat pada gambar berikut:



**GAMBAR 3.24** *Class Diagram*

### 3.2.9 ERD (*Entity Relationship Diagram*)

Hubungan antar entitas yang digunakan pada aplikasi ini dapat dilihat pada gambar berikut:



**GAMBAR 3.25 Entity Relationship Diagram**

### 3.2.10 Kebutuhan Perangkat Lunak dan Perangkat Keras

Kebutuhan perangkat lunak merupakan spesifikasi perangkat lunak yang dibutuhkan agar aplikasi dapat berjalan dengan lancar. Adapun daftar perangkat lunak pada sisi server yang dibutuhkan oleh aplikasi ini adalah sebagai berikut:

1. *Operating system* dari keluarga Unix/Linux;
2. *Web Server* untuk Bahasa pemrograman PHP; dan
3. *SQL Database Server*.

Sementara kebutuhan perangkat lunak pada sisi *client* untuk dapat mengakses aplikasi dengan baik adalah sebagai berikut:

1. *Operating system* Windows, Mac OS, Android atau *Operating system* lainnya yang dapat menjalankan *browser*; dan
2. Rekomendasi menggunakan *Browser* Mozilla Firefox, Google Chrome, Opera dengan versi minimal *update* pada tahun 2017.

Kebutuhan perangkat keras merupakan spesifikasi perangkat keras yang dibutuhkan agar aplikasi dapat berjalan dengan lancar. Adapun daftar perangkat keras pada sisi server yang dibutuhkan oleh aplikasi ini adalah sebagai berikut:

1. CPU Server dengan kecepatan 3092 MHz;
2. RAM Minimal 2 GB; dan
3. *Hard Disk* Minimal 100 GB.

Sementara kebutuhan perangkat keras pada sisi *client* untuk dapat mengakses aplikasi dengan baik adalah sebagai berikut:

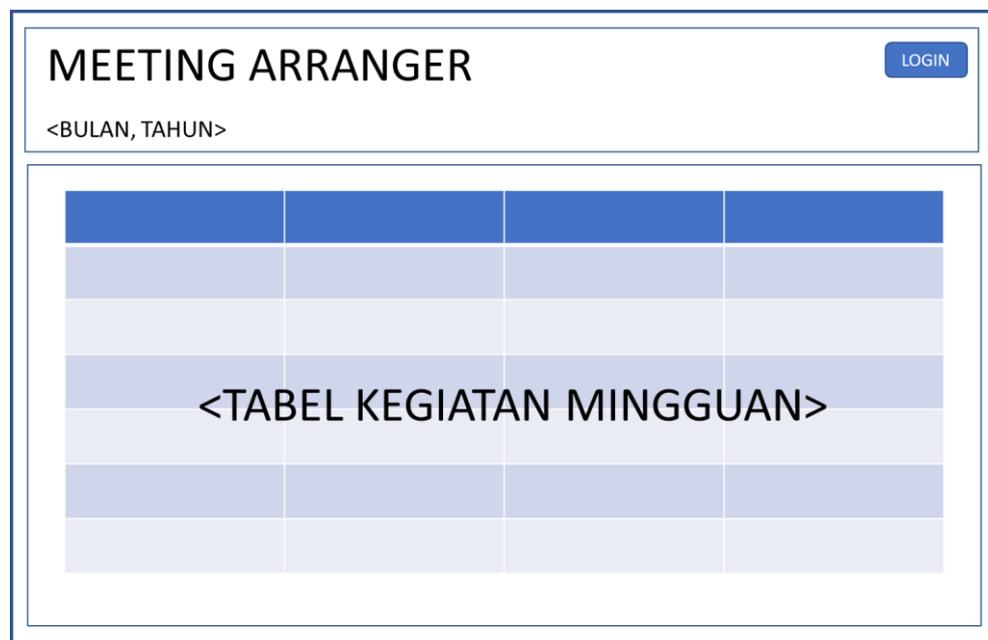
1. Minimal CPU dengan kecepatan 2.2 GHz;
2. Minimal RAM 4 GB; dan
3. Minimal *Hard Disk* 100GB.

### **3.3 Perancangan Antarmuka**

Perancangan antarmuka aplikasi menggambarkan rancangan desain tampilan aplikasi yang dapat digunakan oleh para pengguna atau aktor aplikasi dalam mengoperasikan serta mengakses segala fungsi yang telah disediakan oleh aplikasi.

### 3.3.1 Perancangan Antarmuka Halaman Utama

Perancangan antarmuka halaman utama merupakan tampilan yang berisikan informasi daftar kegiatan mingguan BAPPEDA Provinsi Jawa Barat. Halaman ini dapat diakses oleh pengguna dengan otoritas sebagai *user* maupun *admin*. Adapun tampilan rancangan antarmuka halaman utama dapat dilihat pada **Error! Reference source not found.** berikut:

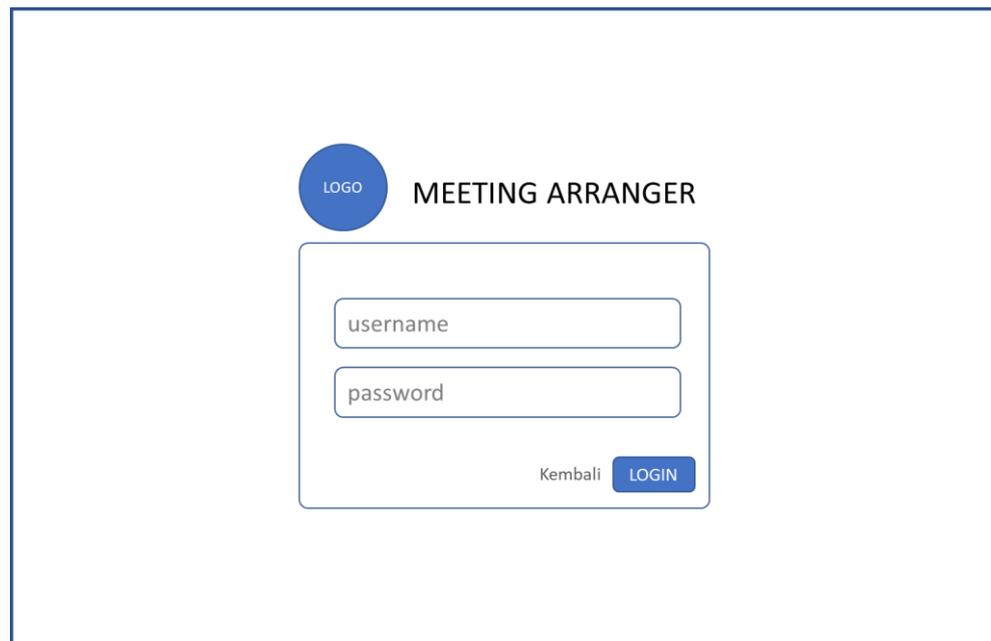


**GAMBAR 3.26** Perancangan Antarmuka Halaman Utama

### 3.3.2 Perancangan Antarmuka Halaman *Login*

Perancangan antarmuka halaman *login* menampilkan halaman formulir *login* yang dapat digunakan oleh pengguna aplikasi untuk masuk

kedalam halaman administrator. Adapun tampilan rancangan antarmuka halaman *login* dapat dilihat pada **Error! Reference source not found.** berikut:

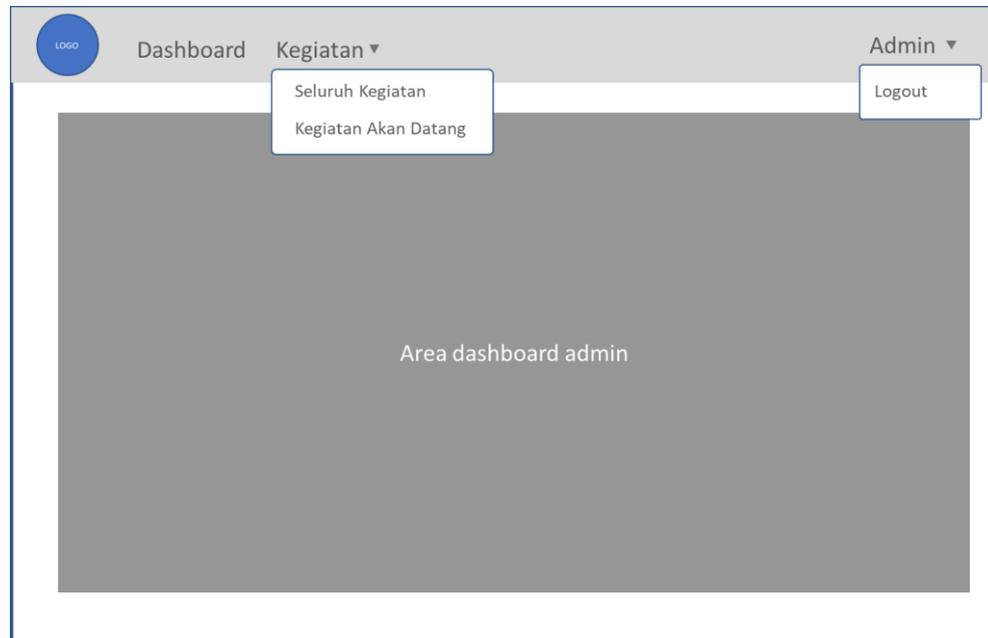


The image shows a wireframe for a login page titled "MEETING ARRANGER". At the top left, there is a blue circular logo with the word "LOGO" inside. To the right of the logo is the text "MEETING ARRANGER". Below this, there is a rounded rectangular container for the login form. Inside the container, there are two input fields: the first is labeled "username" and the second is labeled "password". At the bottom right of the container, there is a blue button labeled "LOGIN" and a link labeled "Kembali" (Back) to its left.

**GAMBAR 3.27** Perancangan Antarmuka Halaman Login

### 3.3.3 Perancangan Antarmuka Halaman Admin

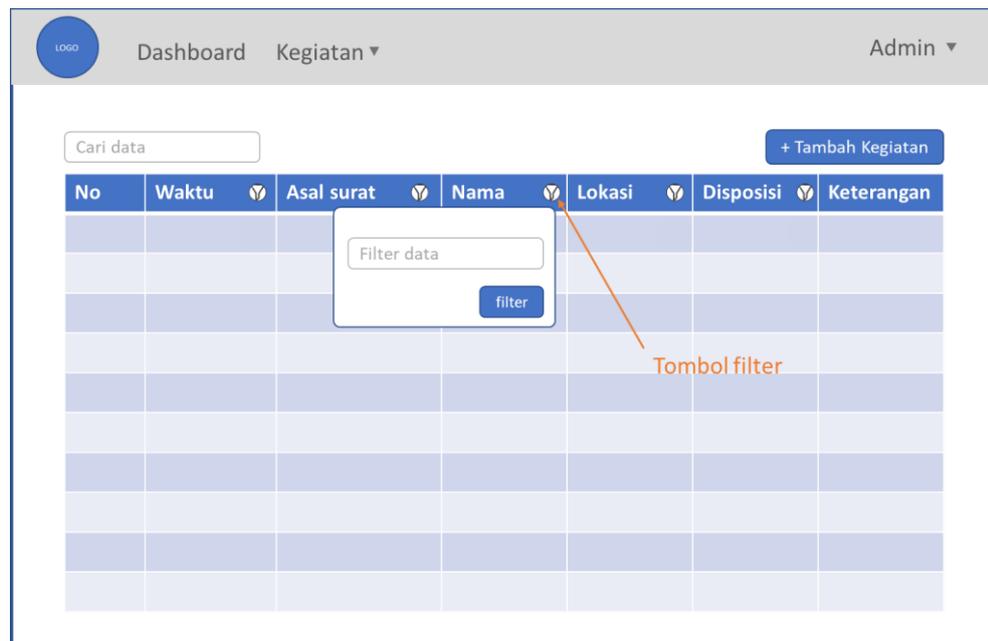
Perancangan antarmuka halaman *Admin* menampilkan halaman utama untuk aktor dengan otoritas sebagai *admin*. Halaman ini hanya dapat ditampilkan jika pengguna berhasil melakukan *login* kedalam aplikasi. Pada halaman ini pengguna ditampilkan menu-menu untuk mengelola kegiatan. Adapun tampilan rancangan antarmuka halaman *admin* dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.28 Perancangan Antarmuka Halaman Admin**

### **3.3.4 Perancangan Antarmuka Halaman Data Kegiatan**

Perancangan antarmuka halaman data kegiatan menampilkan informasi mengenai seluruh kegiatan yang terdaftar pada aplikasi. Pada halaman ini admin dapat melakukan filter dan pencarian data untuk mendapatkan informasi kegiatan sesuai dengan yang diinginkan. Adapun tampilan rancangan antarmuka halaman *admin* dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 3.29** Perancangan Antarmuka Halaman Data Kegiatan

### 3.3.5 Perancangan Antarmuka Halaman Formulir Kegiatan

Perancangan antarmuka halaman formulir kegiatan menampilkan halaman formulir yang dapat digunakan untuk menambah atau mengubah kegiatan yang akan atau telah terdaftar pada aplikasi. Adapun tampilan rancangan antarmuka halaman formulir kegiatan dapat dilihat pada **Error! Reference source not found.** berikut:

The image shows a web interface for activity management. At the top, there is a navigation bar with a logo, the text "Dashboard Kegiatan", and a user profile "Admin". The main content area contains a form with the following fields:

- Asal Surat (Text input)
- Nama Kegiatan (Text input)
- Lokasi (Text input)
- Tanggal Kegiatan (Text input)
- Waktu (Text input)
- Disposisi (Text input)
- Dihadiri Kepala (Dropdown menu)
- Peliputan (Dropdown menu)
- Ditampilkan di Layar (Dropdown menu)

At the bottom right of the form, there are two buttons: "Kembali" and "Simpan".

**GAMBAR 3.30** Perancangan Antarmuka Halaman Formulir Kegiatan

## **BAB IV**

### **IMPLEMENTASI DAN UJI COBA**

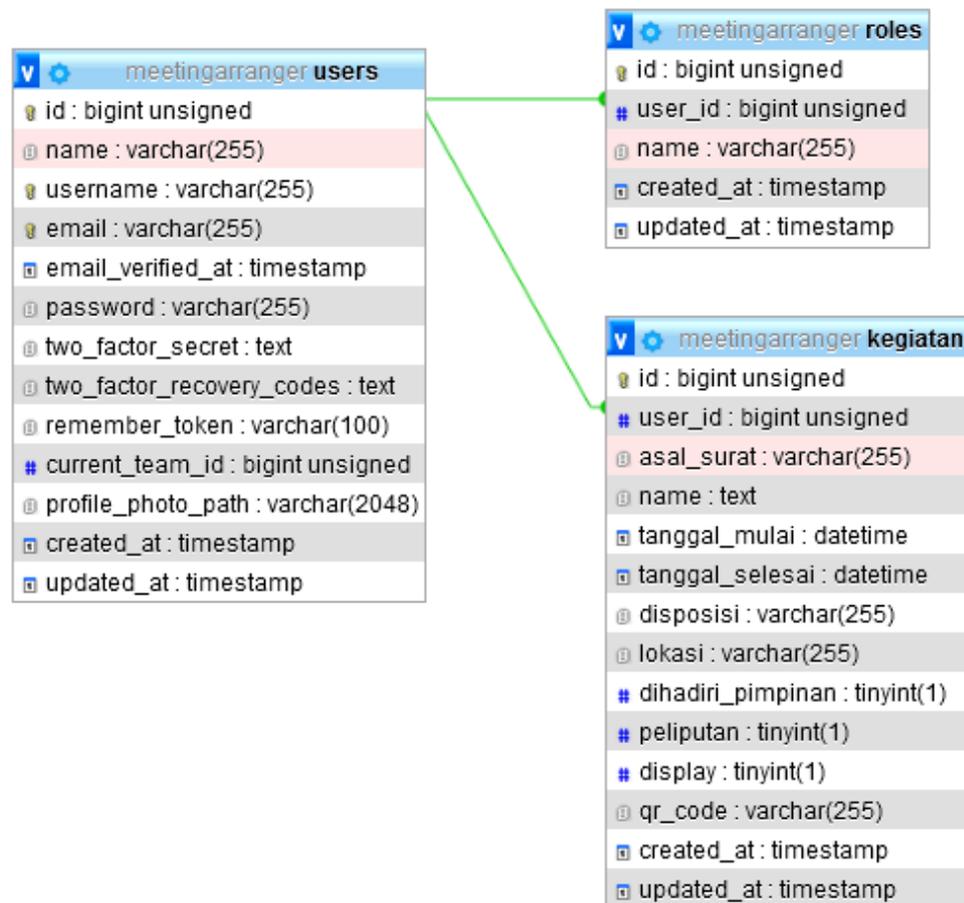
Bab ini membahas mengenai pengkodean dari hasil analisa dan perancangan yang telah dibahas pada BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM.

#### **4.1 Implementasi Aplikasi**

Pembahasan mengenai implementasi aplikasi *meeting arranger* di lingkungan kantor BAPPEDA Provinsi Jawa Barat akan dibagi kedalam 4 (empat) sub pembahasan, yaitu *physical database*, implementasi perangkat lunak, implementasi perangkat keras dan implementasi antarmuka.

##### **4.1.1 *Physical Database***

Implementasi *physical database* untuk pengelolaan kegiatan pada aplikasi *meeting arranger* di BAPPEDA Provinsi Jawa Barat dapat dilihat pada gambar berikut :



**GAMBAR 4.1** *Physical Database*

#### 4.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak diambil berdasarkan hasil Analisa kebutuhan perangkat lunak yang dapat membuat aplikasi ini berjalan. Adapun perangkat lunak yang diimplementasikan pada sisi server adalah sebagai berikut :

1. *Operating system* menggunakan CentOS versi 7 dengan kernel 3.10 x86\_64
2. *Web server Apache* versi 2.4.6 (CentOs)
3. *SQL Database Server* Menggunakan MariaDB versi 10.4.22

Sedangkan implementasi pada sisi *client* telah diuji dengan menggunakan:

1. *Operating System* menggunakan Windows 10 Pro; dan
2. *Browser* menggunakan *firefox* versi 101.0b9 (64-bit)

#### **4.1.3 Implementasi Perangkat Keras**

Implementasi perangkat keras diambil berdasarkan analisa kebutuhan perangkat keras agar aplikasi dapat berjalan dengan baik. Adapun perangkat keras yang diimplementasikan pada sisi server adalah sebagai berikut:

1. CPU Server dengan kecepatan 3092MHz
2. RAM 2 GB
3. *Hard Disk* 150 GB

Sedangkan implementasi perangkat keras pada sisi *client* telah diuji dengan menggunakan :

1. Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz 2.50 GHz
2. RAM 8 GB
3. *Hard Disk* 500 GB

## 4.2 Implementasi Antarmuka

Hasil dari penerapan rancangan antarmuka yang dapat diimplementasikan dalam aplikasi ini ditampilkan kedalam beberapa segmen tangkapan layar antarmuka aplikasi. Penggunaan warna pada antarmuka aplikasi sebisa mungkin dibuat agar nyaman untuk dipandang oleh mata.

### 4.2.1 Implementasi Antarmuka Halaman Utama

Halaman utama aplikasi ditampilkan ketika aplikasi pertama diakses pada *browser computer*. Aplikasi ini menampilkan informasi judul aplikasi, kegiatan mingguan serta tombol *login* yang dapat digunakan untuk pindah ke halaman *login*. Adapun desain antarmuka halaman utama dapat dilihat pada **Error! Reference source not found.** berikut:

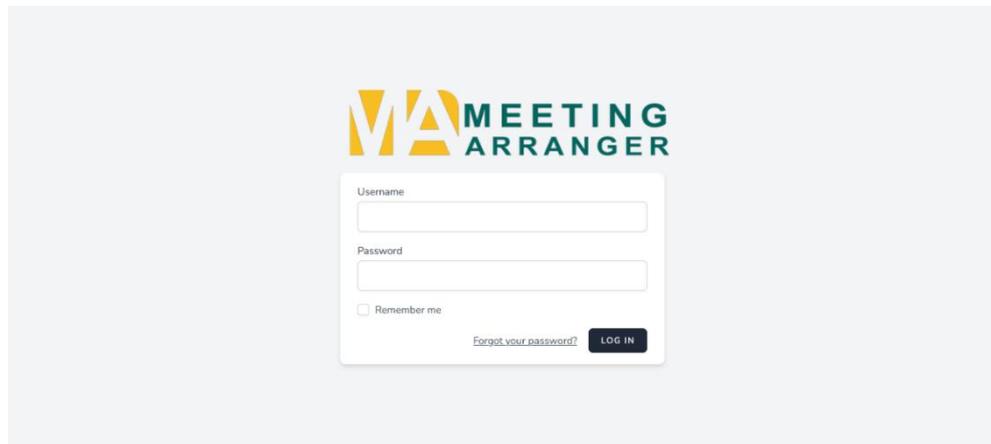


No	Hari dan Tanggal	Waktu	Asal Surat	Nama Kegiatan	Lokasi	Disposisi
1	Senin, 23 Mei 2022	08:30 - 12:00	BAPPELITBANGDA Kota Banjar	Koordinasi dan Konsultasi Tidak Lanjut Monev SDGs	Bappeda Jabar	Bidang PPM
2	Senin, 23 Mei 2022	08:30 - 15:30	Inspektorat Jabar	Undangan Bimtek Penilaian Maturitas SPIP Terintegrasi	Aula Inspektorat Prov.Jabar	Sekretaris dan Bidang PPEPD
3	Senin, 23 Mei 2022	09:30 - 16:00	Dinas SDA Provinsi Jabar	Undangan Tinjauan Lapangan Kawasan Wisata Desa Ciburuy	Kawasan Wisata Desa Ciburuy	Bidang Infracwil
4	Senin, 23 Mei	09:30 -	DPRD Provinsi	Undangan Rapat Kerja Pansus VI Membahas Pasal Per Pasal Rancangan Perda Provinsi Jabar Tentang	Mason Pine Hotel Kabupaten Bandung	Bidang

**GAMBAR 4.2 Implementasi Halaman Utama**

#### 4.2.2 Implementasi Antarmuka Halaman *Login*

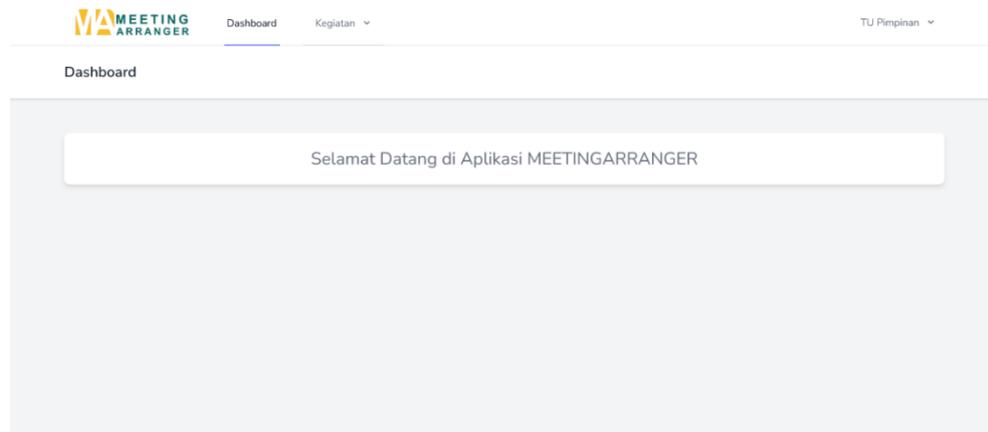
Halaman *login* aplikasi ditampilkan ketika pengguna aplikasi menekan tombol *login* pada halaman utama, pada halaman ini pengguna dapat memasukkan informasi *username* dan *password* untuk masuk kedalam halaman administrator. Adapun desain antarmuka halaman utama dapat dilihat pada **Error! Reference source not found.** berikut:



**GAMBAR 4.3** Implentasi Antarmuka Halaman Login

#### 4.2.3 Implementasi Antarmuka Halaman Admin

Halaman admin ditampilkan ketika pengguna aplikasi berhasil masuk kedalam aplikasi melalui halaman *login*. Pada halaman ini pengguna dapat mengelola informasi kegiatan yang dapat dilihat pada halaman utama oleh seluruh pengguna. Adapun desain antarmuka halaman utama dapat dilihat pada **Error! Reference source not found.** berikut :



**GAMBAR 4.4 Implementasi Antarmuka Halaman Admin**

#### **4.2.4 Implementasi Antarmuka Halaman Data Kegiatan**

Halaman Data Kegiatan ditampilkan ketika pengguna memilih menu Seluruh Kegiatan pada *navigation bar* aplikasi. Pada halaman ini pengguna dapat mengakses formulir kegiatan dengan menekan tombol tambah kegiatan, mengubah kegiatan dengan cara menekan tombol ubah, menghapus kegiatan dengan menekan tombol hapus, mencari kegiatan dengan mengisi informasi pada kolom pencarian serta melakukan penyaringan data dengan menekan ikon *filter* yang tersedia di setiap kolom tabel kegiatan. Adapun desain antarmuka halaman data Kegiatan dapat dilihat pada **Error! Reference source not found.** berikut :

No	Waktu Kegiatan	Asal Surat Kegiatan	Nama Kegiatan	Lokasi	Disposisi	Keterangan
1	Sabtu, 02-07-2022 Pukul 08:00 s/d 16:00 WIB	Bimtek Nasional Kearsipan	Undangan Bimtek Kearsipan	Swissbell Hotel Balikpapan	Sekretaris	Ditampilkan Pada Layer Tidak Hadir Peminan Tidak Untuk Diliput
2	Jumat, 01-07-2022 Pukul 08:00 s/d 16:00 WIB	Bimtek Nasional Kearsipan	Undangan Bimtek Kearsipan	Swissbell Hotel Balikpapan	Sekretaris	Ditampilkan Pada Layer Tidak Hadir Peminan Tidak Untuk Diliput

**GAMBAR 4.5 Implementasi Halaman Data Kegiatan**

#### 4.2.5 Implementasi Antarmuka Halaman Formulir Kegiatan

Halaman formulir kegiatan ditampilkan ketika pengguna aplikasi menekan tombol tambah kegiatan untuk menambah kegiatan baru, atau menekan tombol ubah untuk mengubah data kegiatan yang sudah ada. Adapun desain antarmuka halaman utama dapat dilihat pada **Error! Reference source not found.** berikut:

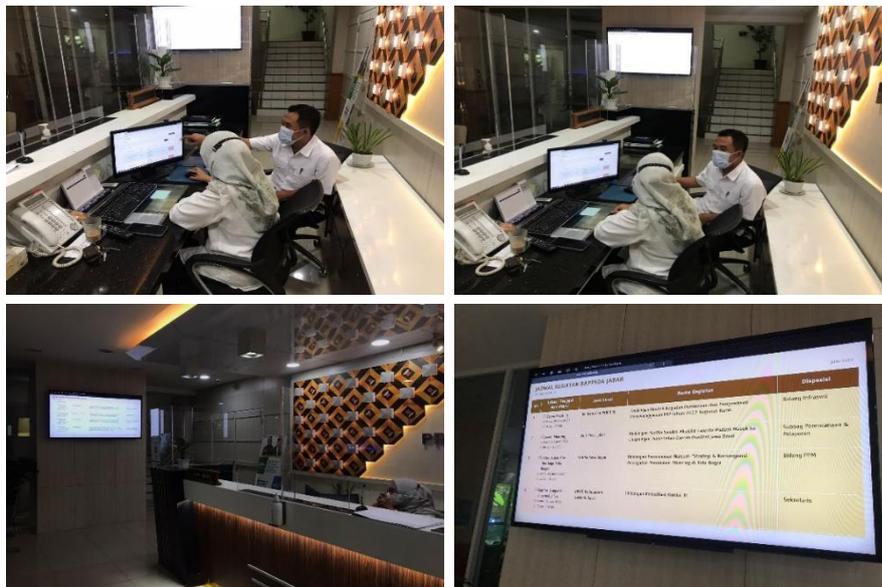
The screenshot shows a web-based form titled "Tambah Kegiatan" (Add Activity) within the "MEETING ARRANGER" application. The form is organized into two columns of input fields:

- Left Column:**
  - Asal Surat (Text input)
  - Nama Kegiatan (Text input)
  - Lokasi (Text input)
  - Tanggal Kegiatan (Text input, with a note "Dapat dipilin lebih dari satu")
- Right Column:**
  - Waktu (Text input, containing "sd")
  - Disposisi (Text input)
  - Dihadiri Kepala Badan (Dropdown menu, selected "Tidak")
  - Petiputan (Dropdown menu, selected "Tidak")
  - Ditampilkan (Dropdown menu, selected "Ya")

At the bottom right of the form, there are two buttons: "Kembali" (Back) and "Simpan" (Save).

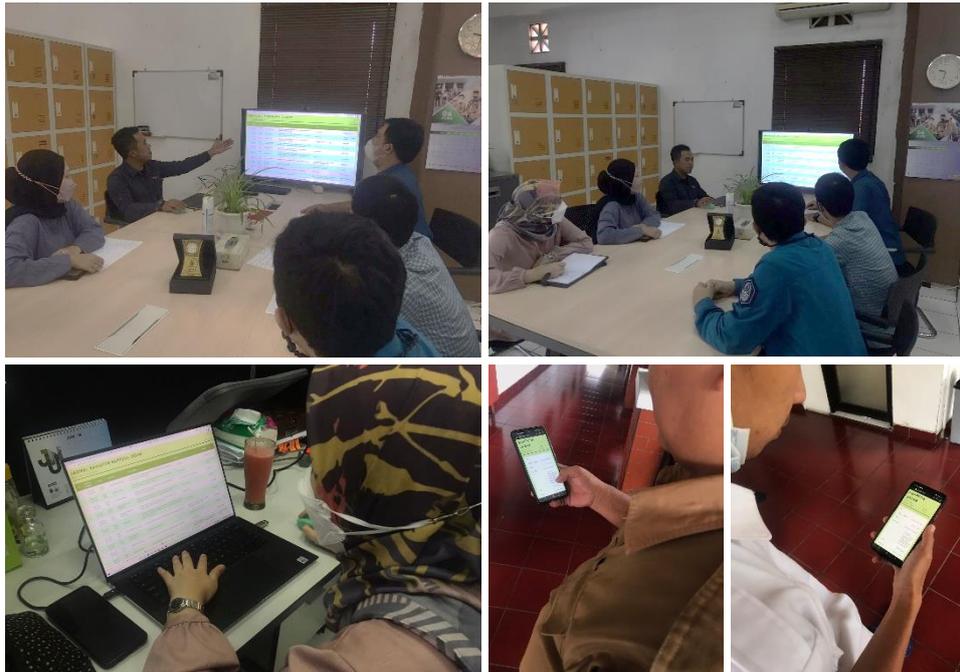
**GAMBAR 4.6 Implementasi Antarmuka Halaman Formulir Kegiatan**

#### 4.2.6 Implementasi Antarmuka Penginputan Agenda Rapat



**GAMBAR 4.7 Implementasi Antarmuka Penginputan Agenda Rapat**

#### 4.2.7 Implementasi Antarmuka melihat Agenda Rapat yang sudah tampil di aplikasi melalui Web Browser



**GAMBAR 4.8** Implementasi Antarmuka melihat Agenda Rapat yang sudah tampil di aplikasi melalui *web browser*

#### 4.3 Uji Coba Aplikasi

Uji coba aplikasi menggunakan metode *black box* dengan menggunakan tabel *User Acceptance Test* (UAT) sebagai media pengujian. Pengujian aplikasi dilakukan terhadap setiap *use case* yang telah disampaikan sebelumnya pada BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM.

#### 4.3.1 User Acceptance Test Use Case Login

Pengujian *user acceptance test* ini difokuskan terhadap *use case login*.

*Use case* ini berisi system autentikasi dan validasi pengguna untuk masuk kedalam halaman administrator. Adapun pengujian *user acceptance test* untuk *use case login* dapat dilihat pada *TABEL 4. 1 User Acceptance Test Use Case Login* berikut:

**TABEL 4. 1 User Acceptance Test Use Case Login**

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
1	Pengujian fungsi <i>login</i> dengan input parameter benar  Aksi : Mengunjungi Halaman <i>Login</i> kemudian memberikan nilai masukan pada kolom : <i>Username:</i> tu_pimpinan <i>Password:</i> bappedajabar	<i>Login</i> berhasil kemudian menampilkan halaman administrator	<i>Login</i> berhasil kemudian menampilkan halaman administrator	Berhasil
2	Pengujian fungsi	<i>Login</i> gagal	<i>Login</i> gagal	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	<p><i>login</i> dengan input parameter salah</p> <p>Aksi: Mengunjungi Halaman <i>Login</i> kemudian memberikan nilai masukan pada kolom :</p> <p><i>Username:</i> admin_aplikasi</p> <p><i>Password:</i> salah</p>	<p>kemudian menampilkan pesan yang menunjukkan <i>username</i> tidak terdaftar pada database aplikasi</p>	<p>kemudian menampilkan pesan yang menunjukkan <i>username</i> tidak terdaftar pada database aplikasi</p>	
3	<p>Pengujian fungsi <i>login</i> dengan input parameter kosong</p> <p>Aksi: Mengunjungi Halaman <i>Login</i> kemudian melakukan <i>login</i> tanpa mengisi <i>Username</i> dan <i>Password</i></p>	<p><i>Login</i> gagal kemudian menampilkan pesan validasi <i>username</i> dan <i>password</i> harus diisi</p>	<p><i>Login</i> gagal kemudian menampilkan pesan validasi <i>username</i> dan <i>password</i> harus diisi</p>	Berhasil

#### 4.3.2 User Acceptance Test Use Case Melihat Kegiatan

Pengujian *user acceptance test* ini difokuskan terhadap *use case* Melihat Kegiatan. *Use case* ini berisi halaman utama aplikasi yang

menampilkan informasi kegiatan mingguan di lingkungan kantor BAPPEDA Jabar. *Use case* ini dapat diakses tanpa perlu proses autentikasi. Adapun pengujian *user acceptance test* untuk *use case login* dapat dilihat pada **TABEL 4.2 User Acceptance Test Use Case Melihat Kegiatan** berikut:

**TABEL 4.2 User Acceptance Test Use Case Melihat Kegiatan**

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
1	Pengujian melihat halaman utama yang menampilkan informasi kegiatan mingguan di BAPPEDA Jabar  Aksi: Mengunjungi halaman utama dengan menggunakan <i>browser</i>	Menampilkan halaman utama dengan informasi kegiatan mingguan di lingkungan kantor BAPPEDA Jabar	Menampilkan halaman utama dengan informasi kegiatan mingguan di lingkungan kantor BAPPEDA Jabar	Berhasil

#### 4.3.3 User Acceptance Test Use Case Tambah Kegiatan

Pengujian *user acceptance test* ini difokuskan terhadap *use case* Tambah Kegiatan. *Use case* ini menampilkan halaman formulir untuk menambahkan data kegiatan kedalam aplikasi. Hanya aktor dengan otoritas

sebagai admin dan berhasil melalui proses *login* yang dapat mengakses *use case* ini . Adapun pengujian *user acceptance test* untuk *use case* Tambah Kegiatan dapat dilihat pada *TABEL 4.3 User Acceptance Test Use Case Tambah Kegiatan* berikut:

**TABEL 4.3 User Acceptance Test Use Case Tambah Kegiatan**

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
1	Pengujian menambahkan kegiatan baru kedalam <i>database</i> aplikasi dengan nilai isian benar  Aksi: Mengunjungi halaman tambah kegiatan dan mengisikan nilai pada kolom: Asal surat: LKPM Nama kegiatan: Undangan Peserta Bimtek	Kegiatan berhasil tersimpan kedalam <i>database</i> aplikasi	Kegiatan berhasil tersimpan kedalam <i>database</i> aplikasi	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	& Ujian Sertifikasi LV I Pengadaan Barang & Jasa Pemerintah Lokasi: Hotel EL-ROYALE BANDUNG Tanggal kegiatan: 23-6- 2022 Waktu: 08:00 dan 16:00 Disposisi: Subbag Kepegumhum Dihadiri kepala badan: Tidak Peliputan: Tidak Ditampilkan: Ya			
2	Pengujian menambahkan kegiatan baru kedalam <i>database</i> aplikasi dengan nilai isian salah atau kosong	Kegiatan gagal ditambahkan kedalam <i>database</i> aplikasi dan menampilkan pesan error	Kegiatan gagal ditambahkan kedalam <i>database</i> aplikasi dan menampilkan pesan error validasi	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	<p>Aksi:  Mengunjungi halaman tambah kegiatan dan mengisikan nilai pada kolom:  Asal surat:  [dikosongkan]  Nama kegiatan:  Undangan Peserta Bimtek &amp; Ujian Sertifikasi LV I Pengadaan Barang &amp; Jasa Pemerintah  Lokasi: Hotel EL-ROYALE BANDUNG  Tanggal kegiatan: dd-mm-yyyy  Waktu: aa:bb dan cc:dd  Disposisi:  Subbag Kepegumhum Dihadiri kepala</p>	validasi		

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	badan: Tidak Peliputan: Tidak Ditampilkan: Ya			
3	Pengujian pengaksesan langsung halaman tambah kegiatan tanpa melalui proses <i>login</i>  Aksi: Mengunjungi halaman tambah kegiatan tanpa melakukan proses <i>login</i> terlebih dahulu	Menampilkan halaman <i>login</i>	Menampilkan halaman <i>login</i>	Berhasil

#### 4.3.4 User Acceptance Test Use Case Ubah Kegiatan

Pengujian *user acceptance test* ini difokuskan terhadap *use case* Ubah Kegiatan. *Use case* ini menampilkan halaman formulir untuk mengubah data kegiatan yang telah terdaftar di aplikasi. Hanya aktor dengan otoritas sebagai admin dan berhasil melalui proses *login* yang dapat mengakses *use case* ini . Adapun pengujian *user acceptance test* untuk *use case* Ubah Kegiatan dapat

dilihat pada TABEL 4.4 *User Acceptance Test Use Case Ubah Kegiatan* berikut:

**TABEL 4.4** *User Acceptance Test Use Case Ubah Kegiatan*

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
1	Pengujian mengubah kegiatan yang telah ada pada <i>database</i> aplikasi dengan nilai isian benar  Aksi: Mengunjungi halaman tambah kegiatan dan mengisikan nilai pada kolom: Asal surat: LKPM edited Nama kegiatan: Undangan Peserta Bimtek	Kegiatan berhasil diubah dan tersimpan kedalam <i>database</i> aplikasi	Kegiatan berhasil diubah dan tersimpan kedalam <i>database</i> aplikasi	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	& Ujian Sertifikasi LV I Pengadaan Barang & Jasa Pemerintah edited Lokasi: Kantor BAPPEDA Jabar Tanggal kegiatan: 24-6- 2022 Waktu: 09:00 dan 14:00 Disposisi: Subbag Kepegumhum edited Dihadiri kepala badan: Ya Peliputan: Ya Ditampilkan: Tidak			
2	Pengujian mengubah kegiatan yang telah ada pada <i>database</i>	Kegiatan gagal diubah pada <i>database</i> aplikasi dan menampilkan	Kegiatan gagal diubah pada <i>database</i> aplikasi dan menampilkan	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	<p>aplikasi dengan nilai isian salah atau kosong</p> <p>Aksi: Mengunjungi halaman tambah kegiatan dan mengisikan nilai pada kolom: Asal surat: [dikosongkan] Nama kegiatan: [dikosongkan] Lokasi: Hotel EL-ROYALE BANDUNG edited Tanggal kegiatan: dd-mm-yyyy Waktu: aa:bb dan cc:dd Disposisi: Subbag Kepegumhum Dihadiri kepala</p>	<p>pesan error validasi</p>	<p>pesan error validasi</p>	

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	badan: Tidak Peliputan: Tidak Ditampilkan: Ya			
3	Pengujian pengaksesan langsung halaman ubah kegiatan tanpa melalui proses <i>login</i>  Aksi: Mengunjungi halaman ubah kegiatan tanpa melakukan proses <i>login</i> terlebih dahulu	Menampilkan halaman <i>login</i>	Menampilkan halaman <i>login</i>	Berhasil

#### 4.3.5 User Acceptance Test Use Case Hapus Kegiatan

Pengujian *user acceptance test* ini difokuskan terhadap *use case* Hapus Kegiatan. *Use case* ini memiliki fungsi untuk menghapus data kegiatan yang telah terdaftar di aplikasi. Hanya aktor dengan otoritas sebagai admin dan berhasil melalui proses *login* yang dapat mengakses *use case* ini . Adapun

pengujian *user acceptance test* untuk *use case* Hapus Kegiatan dapat dilihat pada TABEL 4.5 *User Acceptance Test Use Case Hapus Kegiatan* berikut:

**TABEL 4.5 *User Acceptance Test Use Case Hapus Kegiatan***

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
1	<p>Pengujian penghapusan kegiatan dengan menekan tombol Ya pada pesan konfirmasi penghapusan data.</p> <p>Aksi: Menekan tombol hapus pada kegiatan yang akan dihapus kemudian menekan tombol Ya pada pesan konfirmasi penghapusan data</p>	<p>Kegiatan berhasil terhapus pada <i>database</i> aplikasi</p>	<p>Kegiatan berhasil terhapus pada <i>database</i> aplikasi</p>	Berhasil
2	Pengujian	Kegiatan	Kegiatan gagal	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	<p>penghapusan kegiatan dengan menekan tombol Tidak pada pesan konfirmasi penghapusan data.</p> <p>Aksi: Menekan tombol hapus pada kegiatan yang akan dihapus kemudian menekan tombol Tidak pada pesan konfirmasi penghapusan data</p>	<p>gagal dihapus pada <i>database</i></p>	<p>dihapus pada <i>database</i></p>	
3	<p>Pengujian pengaksesan langsung request hapus kegiatan melalui proses <i>login</i></p> <p>Aksi: Melakukan <i>request</i></p>	<p>Menampilkan halaman <i>login</i></p>	<p>Menampilkan halaman <i>login</i></p>	<p>Berhasil</p>

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	penghapusan data kegiatan tanpa melakukan <i>login</i> terlebih dahulu			

#### 4.3.6 User Acceptance Test Use Case Filter Kegiatan

Pengujian *user acceptance test* ini difokuskan terhadap *use case Filter* Kegiatan. *Use case* ini memiliki fungsi untuk menyaring informasi kegiatan yang akan ditampilkan pada halaman seluruh kegiatan. Penyaringan data kegiatan dapat dilakukan dengan cara menekan tombol *filter* pada kolom tabel atau dengan cara mengisikan informasi yang diinginkan pada kolom pencarian data kegiatan. Hanya aktor dengan otoritas sebagai admin dan berhasil melalui proses *login* yang dapat mengakses *use case* ini . Adapun pengujian *user acceptance test* untuk *use case Filter* Kegiatan dapat dilihat pada TABEL 4.6 User Acceptance Test Use Case Filter Kegiatan berikut:

**TABEL 4.6 User Acceptance Test Use Case Filter Kegiatan**

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
1	Pengujian	Menampilkan	Menampilkan	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	<p>penyaringan informasi dengan menggunakan fitur <i>filter data</i> pada salah satu kolom yang tertampil di tabel kegiatan</p> <p>Aksi: Menekan tombol <i>filter</i> pada kolom Asal Surat Kegiatan kemudian masukan nilai “LKPM” untuk nilai dari informasi yang dicari</p>	<p>kegiatan yang memiliki kata kunci “LKPM” pada kolom Asal Surat Kegiatan</p>	<p>kegiatan yang memiliki kata kunci “LKPM” pada kolom Asal Surat Kegiatan</p>	
2	<p>Pengujian penyaringan informasi dengan menggunakan fitur kolom pencarian</p> <p>Aksi: Mengisikan</p>	<p>Menampilkan kegiatan yang memiliki kata “kepegum” pada semua kolom di tabel kegiatan</p>	<p>Menampilkan kegiatan yang memiliki kata “kepegum” pada semua kolom di tabel kegiatan</p>	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
	informasi “kepegum” pada kolom pencarian data			

#### 4.3.7 User Acceptance Test Use Case Logout

Pengujian *user acceptance test* ini difokuskan terhadap *use case Logout*. *Use case* ini memiliki fungsi untuk menghapus *login session* pada aplikasi. Hanya aktor dengan otoritas sebagai admin dan berhasil melalui proses *login* yang dapat mengakses *use case* ini . Adapun pengujian *user acceptance test* untuk *use case* Hapus Kegiatan dapat dilihat pada TABEL 4.7 *User Acceptance Test Use Case Logout* berikut:

**TABEL 4.7** *User Acceptance Test Use Case Logout*

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
1	Pengujian <i>logout</i> pada aplikasi  Aksi: Menekan tombol <i>logout</i> pada menu aplikasi	<i>Login session</i> aplikasi terhapus dan menampilkan halaman <i>login</i>	<i>Login session</i> aplikasi terhapus dan menampilkan halaman <i>login</i>	Berhasil

No	Kriteria Pengujian	Respon yang Diharapkan	Respon yang Dihasilkan	Kesimpulan Pengujian
2	Pengujian pengaksesan halaman administrator setelah <i>logout</i> dari aplikasi  Aksi: Menekan tombol <i>logout</i> pada menu aplikasi kemudian mengunjungi halaman administrator	Gagal menampilkan halaman administrator kemudian menampilkan halaman <i>login</i>	Gagal menampilkan halaman administrator kemudian menampilkan halaman <i>login</i>	Berhasil

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dalam penelitian ini, penulis telah menguraikan pembahasan mengenai “Pembangunan Aplikasi *Meeting Arranger* di lingkungan Badan Perencanaan Pembangunan Daerah Jawa Barat”. Beberapa kesimpulan yang diambil terhadap penelitian ini antara lain sebagai berikut.

1. Pembuatan aplikasi *Meeting Arranger* dapat mempermudah pengelolaan agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat;
2. Pembuatan aplikasi *Meeting Arranger* dapat membantu proses pendistribusian tugas agenda rapat dan kegiatan kepada bidang-bidang di Bappeda Provinsi Jawa Barat; dan
3. Pembuatan aplikasi *Meeting Arranger* dapat mempermudah dalam penyebaran informasi agenda rapat dan kegiatan di Bappeda Provinsi Jawa Barat.

#### **5.2 Saran**

Adapun saran yang diberikan penulis untuk pengembangan dari aplikasi yang telah dibangun, antara lain sebagai berikut:

1. Tampilan dari aplikasi supaya dapat dipahami dan dimengerti oleh semua pihak baik pihak internal maupun eksternal;
2. Agenda rapat atau jadwal pekerjaan yang ditampilkan harus *realtime*, jangan sampai ada agenda rapat yang sudah dilaksanakan masih muncul sehingga tidak mengakibatkan kekeliruan dalam pelaksanaan rapat; dan
3. Kedepannya aplikasi "*Meeting Arranger*" supaya dapat dikembangkan menjadi aplikasi yang dapat diakses oleh *smartphone*.

## DAFTAR PUSTAKA

- Ariani Sukamto, R., & Shalahuddin, M. (2016). Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung : Informatika. In *Jurnal Pilar Nusa Mandiri* (p. 28).
- Handrianto, Y., & Sanjaya, B. (2020). Model Waterfall Dalam Rancang Bangun Sistem Informasi Pemesanan Produk Dan Outlet Berbasis Web. *Jurnal Inovasi Informatika*, 5(2), 153–160. <https://doi.org/10.51170/jii.v5i2.66>
- James Rumbaugh, Jacobson, I., & Booch, G. (2013). The Unified Modeling Language Reference Manual. In *Journal of Chemical Information and Modeling* (4th ed., Vol. 53, Issue 9). ADDISON-WESLEY.
- Maulana, teguh dikri, & Herdiana, R. (2019). *Pengembangan Aplikasi Pengingat Jadwal Periksa Medis Short Message Service (SMS) Gateway Berbasis Web Kepada Penderita Gagal Ginjal*. 211–216.
- MZ, M. K. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions pada Aplikasi Sistem Informasi di Sekolah. *Jurnal Mikrotik*, 06(3), 02–16. <https://ojs.ummetro.ac.id/index.php/mikrotik/article/download/303/243>
- Nangin, W., Naj Joan, X. B. N., Sengkey, R., Informatika, T., Teknik, F., Sam, U., & Manado, R. (2019). *Usaha Perjalanan Wisata Berbasis Android*.
- Ramsari, N., & Rifaldi, A. (2018). *Kegiatan Akademik Disertai Sistem Reminder*. IX(1).
- Tri, M. B. (2020). Perancangan Sistem Informasi Management Siswa Berprestasi Berbasis Android Pada Smk Pgri Rawalumbu. *Jurnal Sains & Teknologi*

*Fakultas Teknik, X(2), 30–39.*

- Vinandha, V., Priyambadha, B., & Heru Nurwarsito. (2019). Pengembangan Aplikasi Mobile Peningat Jadwal Layanan Posyandu dengan Menggunakan Teknologi Firebase Cloud. ... *Teknologi Informasi Dan ...*, 3(5), 4133–4141.  
<http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5140>  
<http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/5140/2424>
- Widagdo, D. I. (2020). Pembuatan Aplikasi Mobile Peningat Minum Obat Untuk Orang Tua. *Jurnal Teknik Informatika*.

## LAMPIRAN

### Kode Program

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;
```

```
class AppToken extends Model  
{  
    use HasFactory;  
  
    protected $table = 'app_tokens';  
  
    protected $fillable = [  
        'name',  
        'role',  
        'token'  
    ];  
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;
```

```
class Attendance extends Model  
{  
    use HasFactory;  
  
    protected $table = 'attendance';  
  
    protected $fillable = [  
        'nip',  
        'name'  
    ];  
  
    public function kegiatan()
```

```

    {
        return $this->belongsTo(Kegiatan::class);
    }
}

```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Kegiatan extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table = 'kegiatan';
```

```
    protected $fillable = [
```

```
        'user_id',
```

```
        'asal_surat',
```

```
        'name',
```

```
        'tanggal_mulai',
```

```
        'tanggal_selesai',
```

```
        'disposisi',
```

```
        'lokasi',
```

```
        'dihadiri_pimpinan',
```

```
        'qr_code',
```

```
        'peliputan',
```

```
        'display',
```

```
    ];
```

```
    protected $appends = [
```

```
        'tanggal_kegiatan',
```

```
        'tanggal_kegiatan_name',
```

```
        'waktu_mulai',
```

```
        'waktu_selesai',
```

```
    ];
```

```
    public function users()
```

```
    {
```

```
        return $this->belongsTo(User::class);
```

```
    }
```

```
    public function attendance()
```

```

    {
        return $this->hasMany(Attendance::class);
    }

    public function getWaktuMulaiAttribute()
    {
        return date('H:i', strtotime($this->tanggal_mulai));
    }

    public function getWaktuSelesaiAttribute()
    {
        return date('H:i', strtotime($this->tanggal_selesai));
    }

    public function getTanggalKegiatanAttribute()
    {
        return date('d-m-Y', strtotime($this->tanggal_mulai));
    }

    public function getTanggalKegiatanNameAttribute()
    {
        setlocale(LC_ALL, 'id_ID');
        return strftime('%A, %d %B %Y', strtotime($this->tanggal_mulai));
    }
}

```

```
<?php
```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Role extends Model
{
    use HasFactory;

    protected $table = 'roles';

    protected $fillable = [
        'user_id',
        'name',
    ];

    public function user(){

```

```

        return $this->belongsTo(User::class);
    }
}

```

```
<?php
```

```
namespace App\Models;
```

```

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Fortify\TwoFactorAuthenticatable;
use Laravel\Jetstream\HasProfilePhoto;
use Laravel\Sanctum\HasApiTokens;

```

```
class User extends Authenticatable
```

```

{
    use HasApiTokens;
    use HasFactory;
    use HasProfilePhoto;
    use Notifiable;
    use TwoFactorAuthenticatable;

    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $fillable = [
        'name',
        'username',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array
     */
    protected $hidden = [
        'password',
        'remember_token',
        'two_factor_recovery_codes',
    ];
}

```

```

        'two_factor_secret',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    /**
     * The accessors to append to the model's array form.
     *
     * @var array
     */
    protected $appends = [
        'profile_photo_url',
    ];

    public function kegiatan()
    {
        return $this->hasMany(Kegiatan::class);
    }

    public function role()
    {
        return $this->hasOne(Role::class);
    }

    public function nip()
    {
        return $this->hasMany(Nip::class);
    }
}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class ViewKegiatan extends Model

```

```
{  
  use HasFactory;  
  
  protected $table = 'view_kegiatan';  
}
```