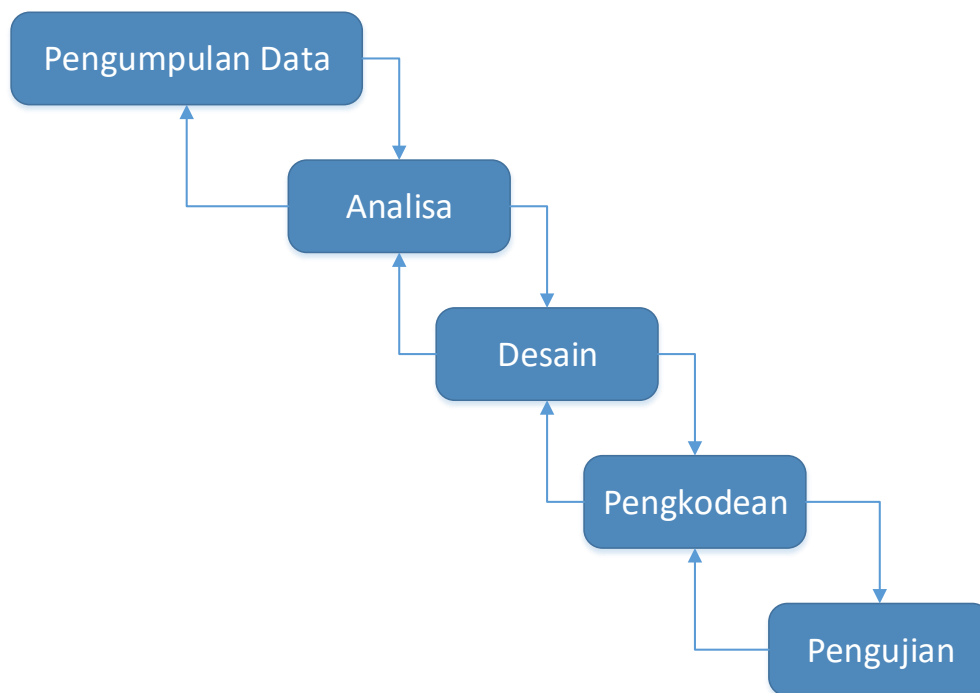


## BAB III CARA DAN METODOLOGI PENELITIAN

### 3.1 Metodologi Penelitian

Penelitian tentang Sistem Informasi Perusahaan dan Kepegawaian PT. BUHARUM berbasis *website* menggunakan metode *Software Development Life Cycle* (SDLC) model *waterfall*. Model *waterfall* digambarkan seperti pada Gambar 3.1.



**Gambar 3.1** SDLC Alur Penelitian

#### a. Pengumpulan Data

Pengumpulan data merupakan usaha yang dilakukan untuk memperoleh informasi dalam bentuk data yang dibutuhkan dalam penelitian. Metode pengumpulan data dapat dilakukan dengan metode wawancara (*interview*), metode pengamatan (*observasi*), dokumentasi (*documentation*).

b. Tahap Analisis

Tahap berikutnya setelah data terkumpul adalah tahap analisis kebutuhan sistem. Kegiatan ini diintensifkan dan difokuskan pada sistem, yaitu menganalisa kebutuhan dan persyaratan proses pada sistem yang akan dibangun.

c. Tahap Desain

Tahapan ini bertujuan untuk membuat rancangan dari hasil analisa yang telah dilakukan pada tahap sebelumnya. Perancangan yang akan dilakukan dalam pengembangan sistem ini contohnya, perancangan struktur data, perancangan proses dan perancangan antar muka.

d. Tahap Pengkodean

Tahapan implementasi merupakan tahapan mengubah rancangan yang telah dibuat menjadi kumpulan kode atau instruksi yang akan dijalankan oleh komputer.

e. Tahap Pengujian

Pengujian sistem dilakukan untuk mengukur kelayakan dan kesesuaian sistem yang dibangun. Semua fungsi harus diuji supaya bebas dari *error* dan dapat berjalan sebagaimana yang diharapkan.

### 3.2 Tempat dan Waktu Penelitian

Tempat yang digunakan penulis dalam melakukan penelitian ini yaitu PT. BUHARUM. Adapun waktu penelitian tentang Sistem Informasi Perusahaan dan Kepegawaian PT. BUHARUM ini dilaksanakan dari bulan April sampai Juni 2017.

### 3.3 Alat dan Bahan Penelitian

Penelitian membutuhkan alat-alat untuk mendukung berjalannya perancangan dan implementasi *website*, antara lain:

#### 3.3.1 Perangkat Keras

*Personal Computer (PC)/Laptop*

- a. *32/64 bit architecture processor*
- b. *4 GB Random Access Memmory (RAM)*

### 3.3.2 Perangkat Lunak

#### a. Sistem Operasi *Windows 10*

*Windows 10* merupakan sistem operasi komputer pribadi yang dikembangkan oleh Microsoft sebagai bagian dari keluarga sistem operasi *Windows NT*. *Windows 10* bertujuan untuk mengatasi kekurangan dalam antarmuka pengguna pertama kali diperkenalkan oleh *Windows 8* dengan menambahkan mekanik tambahan yang dirancang untuk meningkatkan pengalaman pengguna untuk perangkat yang tidak ada layar sentuh (seperti komputer meja dan laptop), termasuk kebangkitan menu mulai yang terlihat di *Windows 7*, sistem desktop maya, dan kemampuan untuk menjalankan aplikasi bursa *Windows* dalam jendela pada desktop daripada modus layar penuh.

#### b. *Netbeans*

Perangkat lunak *Netbeans* merupakan alat utama dalam melaksanakan pembuatan *web* perusahaan dan kepegawaian PT. BUHARUM.

#### c. *MySql*

Sebagai alat yang digunakan untuk mengelola dan membuat *database* beserta isinya.

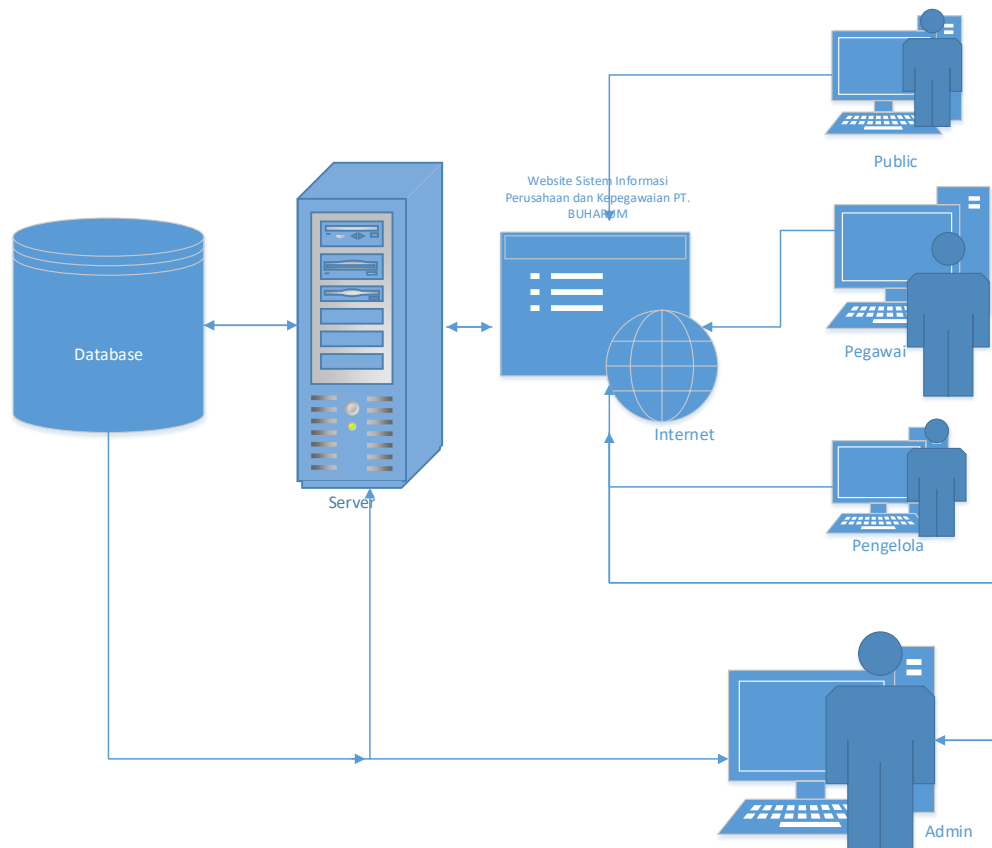
#### d. XAMPP

Sebagai alat yang digunakan untuk menjadi sebuah *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan Bahasa pemrograman PHP dan *Perl*.

#### e. *Bootstrap*

Sebagai alat yang digunakan untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain *website*.




### 3.4 Arsitektur



**Gambar 3.2** Arsitektur Sistem Informasi Perusahaan dan Kepegawaian

*Database server* yang digunakan pada aplikasi *MySQL* dan menggunakan *Xampp* sebagai *web server*. Komunikasi antar pengguna dan *web server* menggunakan internet dan *web browser*. Saat pengguna (*public*) mengakses aplikasi, *web server* memuat antarmuka dan melakukan pengambilan data yang diperlukan dari *database server*. Dan saat pengguna (*pegawai*) mengakses aplikasi, *web server* memuat antarmuka dan melakukan pengambilan data yang diperlukan dari *database server*. Melalui antarmuka yang dimuat *web server* pengguna (*pegawai*) bisa menyimpan data ke *database server*.

**Tabel 3.1** Keterangan Komponen arsitektur Sistem

	<p>Pengguna <i>Public</i>,</p> <p>Pengguna Pegawai (Staff Perdivisi) ,</p> <p>Pengelola</p> <p>Admin</p>
 <p>Database</p>	<p>Berisi data-data sebagai berikut:</p> <ol style="list-style-type: none"> <li>1. Data Pegawai</li> <li>2. Data <i>User</i></li> <li>3. Data Admin</li> <li>4. Data Pangkalan Gas</li> <li>5. Data <i>Setting</i></li> <li>6. Data Berita</li> <li>7. Data <i>Slide Show</i></li> </ol>
 <p>Server</p>	<p><i>Server</i> digunakan sebagai tempat untuk penyimpanan data-data tersebut.</p>

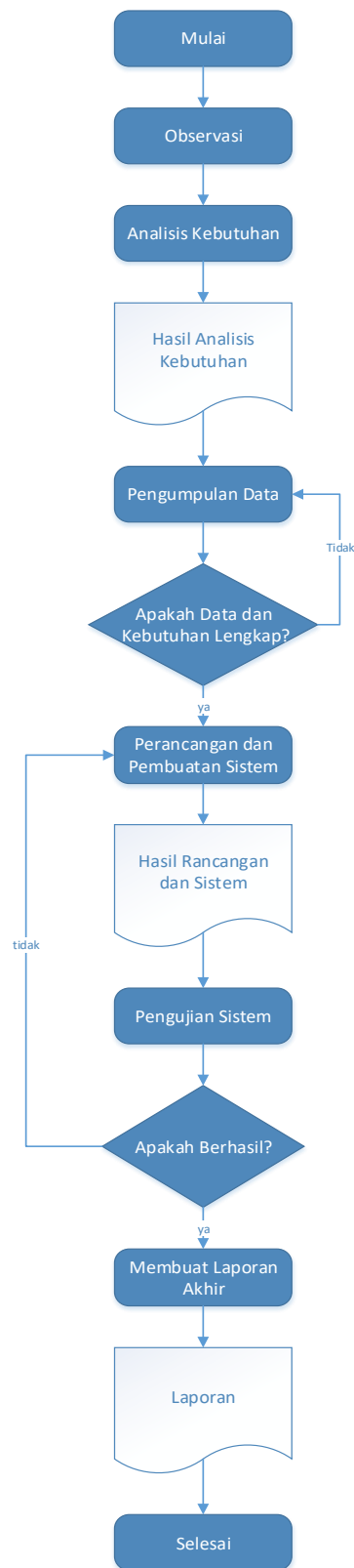
### 3.5 Teknik Pengumpulan Data

Tahap pada alur teknik pengumpulan data digambarkan dalam *flow chart* pada gambar 3.3. Penelitian sistem informasi perusahaan dan kepegawaian PT. BUHARUM berbasis *website* menggunakan metode SDLC, hal ini bertujuan apabila terdapat kesalahan sistem ataupun *error* maka dilakukan analisis kebutuhan

kembali dari awal memperbaiki sistem. Alur penelitian penulis dilakukan dalam beberapa tahap sebagai berikut:

1. Menganalisis Kebutuhan

Analisis kebutuhan pada dasarnya merupakan tahap merancang dan membangun sebuah sistem informasi. Analisis kebutuhan merupakan suatu proses untuk mendapatkan informasi, model dan spesifikasi tentang perangkat lunak yang dibutuhkan pada aplikasi. Sebelum membuat program aplikasi berbasis *web*, terlebih dahulu melakukan wawancara kepada *Manager* Divisi di PT. BUHARUM yang berguna untuk mendapatkan data maupun informasi terkait bagaimana fitur-fitur yang diharapkan oleh perusahaan supaya *web* yang dibuat sesuai dengan harapan perusahaan dan pengguna.



**Gambar 3.3** Alur Teknik Pengumpulan Data

## 2. Pengumpulan data dan menentukan kebutuhan

Pengumpulan data berasal dari *requirement* yang telah ditentukan berdasarkan penggabungan data primer dan sekunder. *Requirement* merupakan daftar kebutuhan dan persyaratan dari aplikasi. Dengan adanya *requirement*, pembuatan aplikasi akan dapat terarah dan terstruktur. *Requirement* juga dapat membantu dalam melakukan *testing* ketika aplikasi telah selesai dibuat.

## 3. Perancangan dan Pembuatan sistem

Pembuatan rancangan bertujuan supaya *website* yang dibuat dapat sesuai dengan yang diharapkan dan tidak akan ada fitur yang dihilangkan ataupun tertinggal.

## 4. Pengujian sistem

Sistem akan diuji sesuai dengan *requirement* yang telah ditentukan sebelumnya. Semua *requirement* harus terpenuhi dan tidak ada yang tertinggal ataupun tidak sesuai dengan *requirement*.

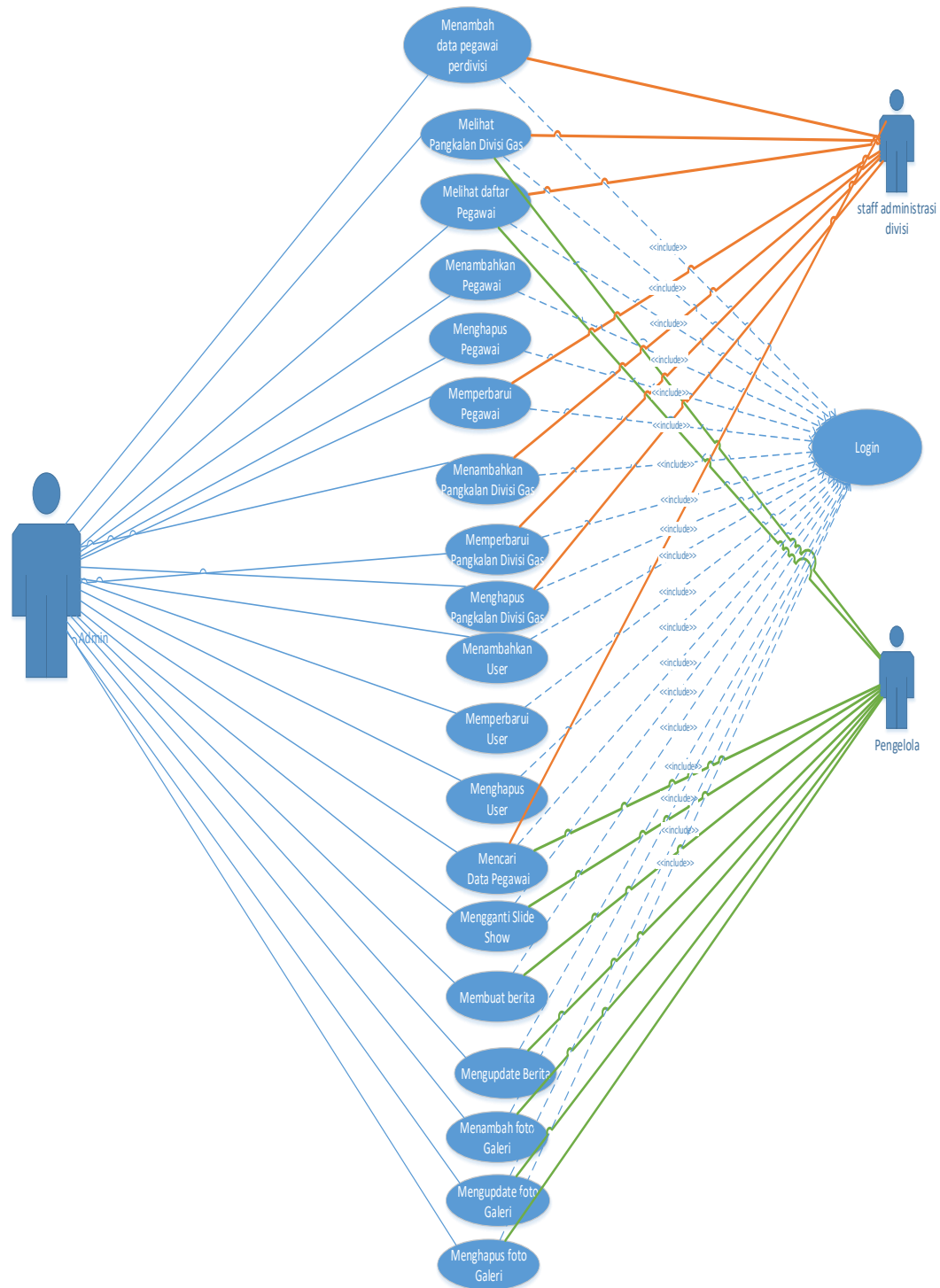
### 3.6 Rancangan

Dalam pembuatan aplikasi dilakukan perancangan *database* menggunakan bantuan Diagram ER. Metode perancangan lain yang digunakan dalam aplikasi adalah *Unified Markup Language* (UML) yang dipakai dalam pengembangan aplikasi yaitu model *Use case Diagram*, *Activity Diagram*, dan *Class Diagram*.

#### 3.6.1 Use Case Diagram

Gambaran *Use Case Diagram* yang digunakan dalam aplikasi dapat dilihat pada gambar 3.4.





**Gambar 3.4** Use Case Diagram aplikasi

Berdasarkan Use case Gambar 3.4 dapat dilihat actor administrator, berikut penjelasannya.

a. Administrator

Administrator merupakan *actor* yang dapat melakukan aktifitas apapun yang terdapat didalam *website*. Administrator dapat diartikan seperti *super user* yang kegiatannya tidak dibatasi.

1. *Login*
2. Melihat daftar pegawai
3. Menambahkan pegawai
4. Menghapus pegawai
5. Memperbarui pegawai
6. Mencari data pegawai
7. Mencetak data pegawai
8. Menambah data pegawai perdivisi
9. Melihat pangkalan divisi gas
10. Menambah pangkalan divisi gas
11. Memperbarui pangkalan divisi gas
12. Menghapus pangkalan divisi gas
13. Menambah *user login*
14. Memperbarui *user login*
15. Menghapus *user login*
16. Mengganti *slide show*
17. Membuat berita
18. Memperbarui berita
19. Menambah foto galeri
20. Memperbarui foto galeri
21. Menghapus foto galeri

Semua kegiatan yang ada di *website* admin dapat digunakan oleh administrator. Administrator dapat mengelola secara penuh halaman umum *website*. Dalam *use case* diagram pada gambar 3.4 terdapat penghubung antar *use case* berupa relasi asosiassi.

b. Pengelola

Aktor pengelola merupakan pengguna atau aktor yang mengatur *website* utama dengan kemampuan dibawah administrator. Pengelola hanya diberikan izin melakukan beberapa hal. Ada beberapa kegiatan yang tidak dapat dilakukan oleh pengelola namun dapat dilakukan oleh administrator. Berikut daftar kegiatan yang dapat dilakukan oleh pengelola:

1. *Login*
2. Melihat daftar pegawai
3. Mencari data pegawai
4. Melihat data divisi
5. Melihat pangkalan divisi gas
6. Mengganti *slide show*
7. Membuat berita
8. Memperbarui berita
9. Menambah foto galeri
10. Memperbarui foto galeri
11. Menghapus foto galeri

Dilihat dari daftar kegiatan yang dapat dilakukan oleh pengelola, dapat diartikan bahwa pengelola tidak dapat menghapus pegawai, menghapus pangkalan divisi gas, menambah *user staff* administrasi divisi, menghapus *user staff* administrasi divisi.

c. *Staff* Administrasi Divisi

Aktor *Staff* administrasi divisi merupakan pengguna atau aktor yang bertugas mengatur pegawai perdivisi dengan kemampuan dibawah pengelola dan admin. Berikut daftar kegiatan yang dapat dilakukan oleh *staff* administrasi divisi:

1. *Login*
2. Melihat data pegawai
3. Melihat pangkalan divisi gas
4. Menambahkan pangkalan divisi gas

5. Memperbarui pangkalan divisi gas
6. Menghapus pangkalan divisi gas
7. Menambah data pegawai perdivisi
8. Memperbarui data pegawai perdivisi
9. Mencari data pegawai

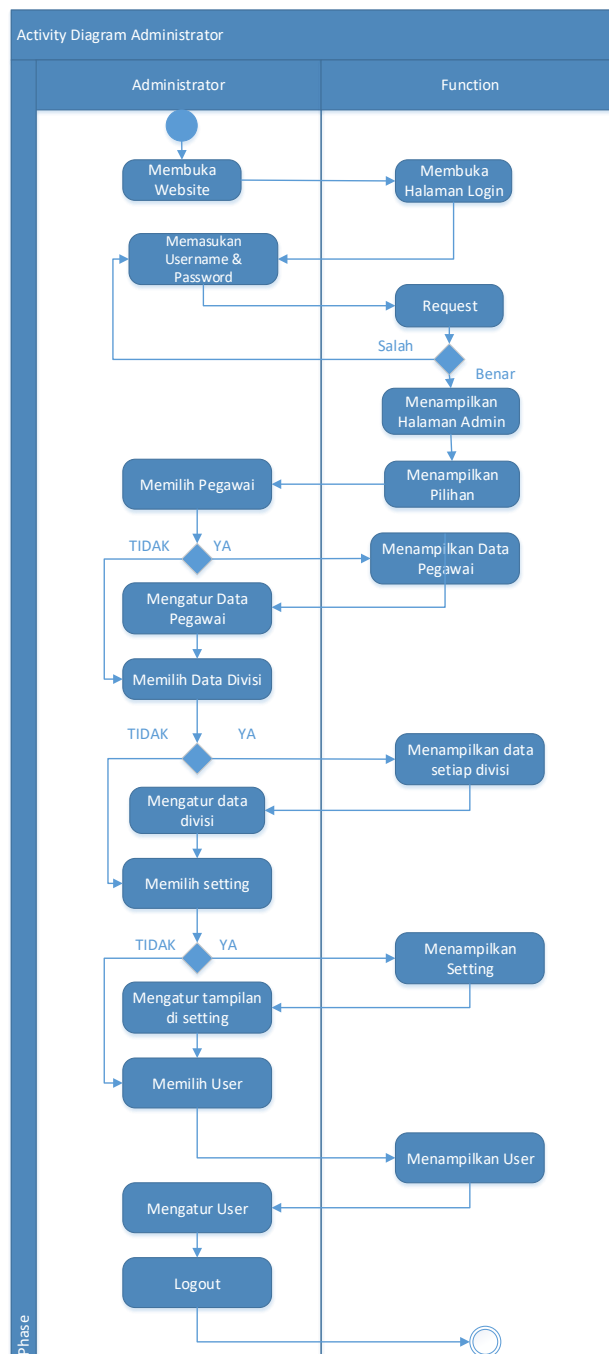
Aktor *staff* administrasi divisi hanya bertugas untuk menambahkan apabila ada pegawai baru di dalam divisinya atau memperbarui apabila ada data pegawai yang harus diubah atau diganti. Aktor *staff* administrasi divisi tidak dapat mengatur tampilan *website* seperti halnya administrator atau pengelola.

### 3.6.2 Activity Diagram

*Activity* Diagram akan menjelaskan bagaimana jalannya aplikasi dan hubungan antara pengguna, *website* dan *database*. *Activity* diagram pada rancangan *website* yang akan dibuat dibagi menjadi beberapa *Activity* diagram yaitu :

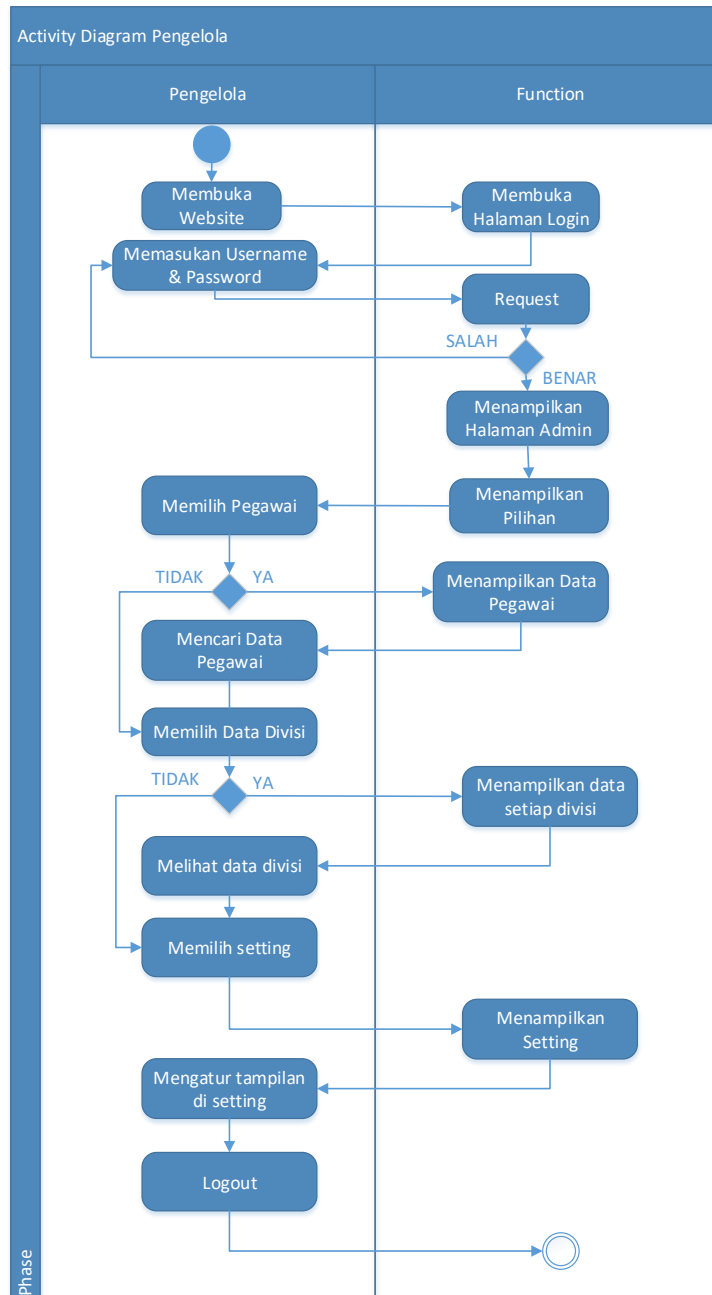
#### a. Activity Diagram Administrator

Gambar 3.5 menunjukkan *Activity* Diagram Administrator pada *web* Administrator menggambarkan bagaimana kegiatan yang dilakukan oleh *user* atau aktor administrator dengan *website*. *Website* yang dimaksud adalah *website* pengolah *website* utama. Administrator setelah membuka *website* harus mengisi *form login*. Jika *username* dan *password* yang dimasukkan oleh administrator adalah benar, maka administrator akan masuk kedalam menu *home* atau halaman utama yang banyak terdapat pilihan-pilihan kegiatan yang dapat dilakukan oleh administrator. Namun jika *username* dan *password* yang dimasukkan oleh administrator adalah salah, maka *website* akan memberikan notifikasi bahwa *username* dan *password* yang dimasukkan oleh administrator salah dan tetap berada di halaman *login*. Jika sudah masuk, administrator akan diberikan beberapa pilihan yang dapat dipilih. Pilihan yang dipilih oleh administrator secara langsung akan direspon oleh *website* untuk menampilkan halaman terkait dengan data *terupdate*.



Gambar 3.5 Activity Diagram Administrator

b. Activity Diagram Pengelola

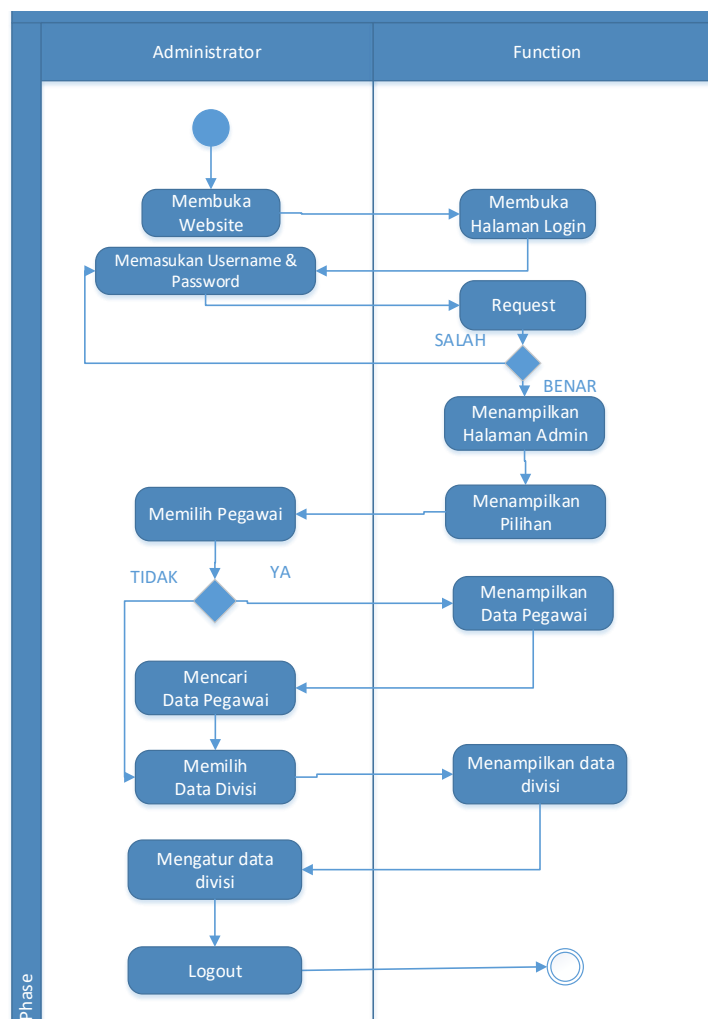


**Gambar 3.6** Activity Diagram Pengelola

Gambar 3.6 menunjukkan Activity Diagram pengelola pada *website*. Sama seperti gambar 3.5 Activity Diagram Administrator, Activity Diagram Pengelola juga harus memasukkan terlebih dahulu *username* dan *password* supaya *website* dapat merespon dengan tampilan menu *home* yang berisi banyak pilihan. Jika

dibandingkan antara *Activity Diagram* administrator dengan *Activity Diagram* pengelola terlihat jelas bahwa kegiatan dan interaksi yang dapat dilakukan oleh pengelola lebih sedikit dibandingkan dengan administrator. Hal tersebut terlihat dari jumlah pilihan yang dapat dipilih oleh aktor pengelola.

c. *Activity Diagram* Staff Administrasi Divisi



**Gambar 3.7** *Activity Diagram* Staff Administrasi Divisi

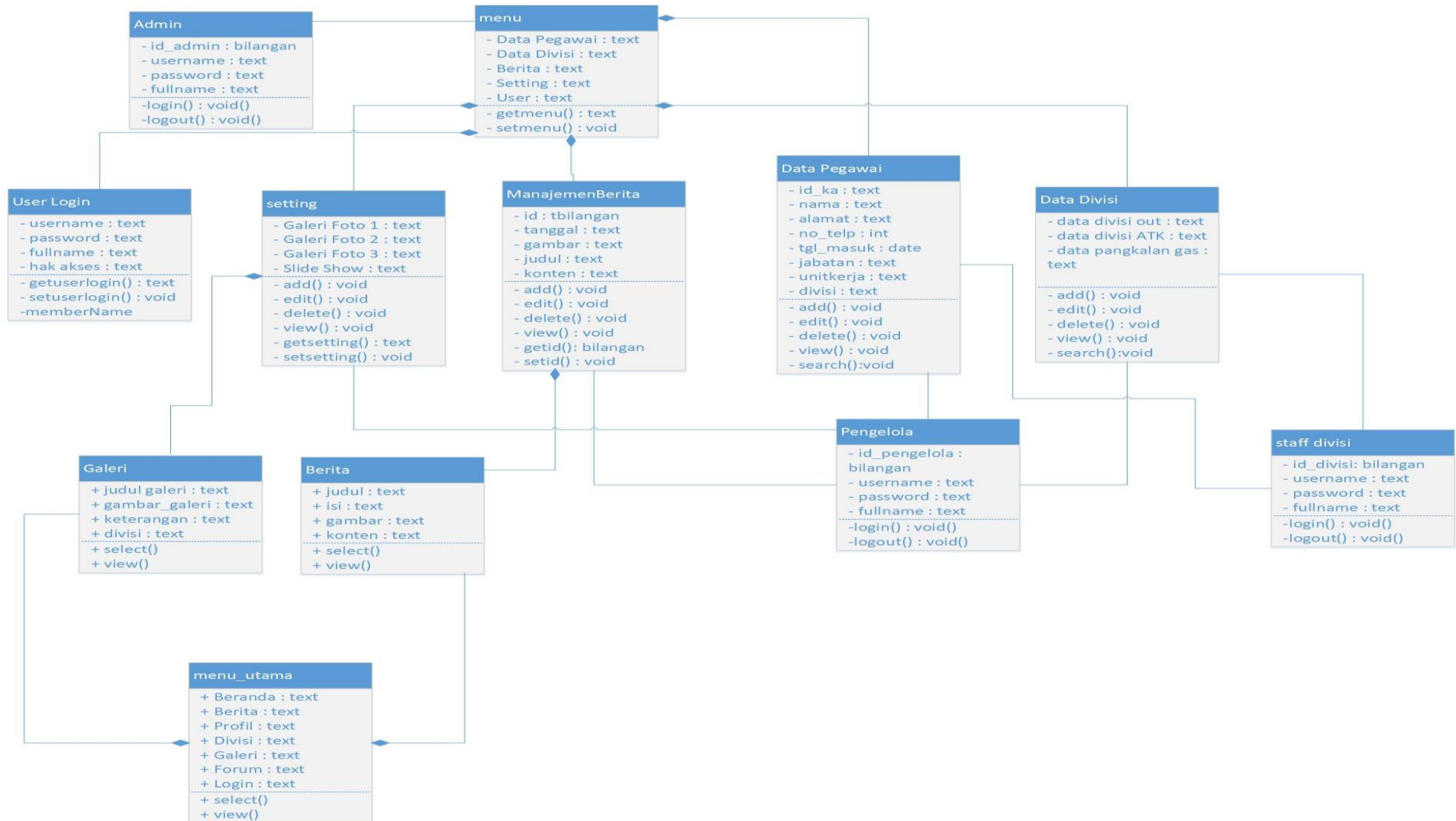
Pada gambar 3.7 merupakan *Activity diagram* Staff Adminstrasi Divisi. Gambar 3.7 menggambarkan bagaimana kegiatan yang dilakukan oleh *user* atau aktor staff administrasi divisi dengan *website*. Staff administrasi Divisi juga sama seperti aktor lainnya yang harus melakukan *login* terlebih dahulu dengan mengisi

*username* dan *password*. Jika dibandingkan antara *Activity Diagram* administrator dan pengelola terlihat jelas bahwa kegiatan dan interaksi yang dapat dilakukan oleh staff administrasi divisi ini lebih sedikit. Hal tersebut karena staff administrasi divisi hanya mengatur bagian divisi masing-masing.

### **3.6.3 Class Diagram**

*Class diagram* dibuat untuk menjelaskan hubungan antara kelas yang satu dengan kelas yang lainnya. Di dalam beberapa kelas, terdapat atribut yang menjelaskan komponen apa saja yang terdapat didalam kelas tersebut dan terdapat *operation* yang berfungsi sebagai penjelas kegiatan apa saja yang dapat dilakukan oleh suatu aktor ketika berada didalam kelas tersebut. Gambaran *Class Diagram* yang digunakan dalam aplikasi dapat dilihat pada gambar Gambar 3.8.





Gambar 3.8 Class Diagram

Berikut adalah penjelasan class diagram pada gambar 3.8.

- a. Pada *class setting*, memiliki fungsi untuk mengatur tampilan halaman depan aplikasi, di dalam *class setting* admin bisa melakukan penambahan *slide show* dan galeri, menghapus *slide show* dan galeri, *edit slide show* dan galeri.
- b. Pada *class* berita, memiliki fungsi untuk mengatur berita pada halaman depan *website*, di dalam *class* berita admin dapat melakukan menambah, menghapus, *edit* berita.
- c. Pada *class user login*, memiliki fungsi untuk mengatur siapa saja petugas yang bisa melakukan akses, di dalam *class user login* admin bisa menambah, memilih dan menghapus *user*.
- d. Pada *class* data divisi, memiliki fungsi untuk mengatur data perdivisi, di dalam *class* data divisi admin bisa menambah, mencari, menghapus dan edit data perdivisi.
- e. Pada *class* data pegawai, memiliki fungsi untuk mengatur data semua pegawai, di dalam *class* data pegawai admin bisa menambah, mencari, menghapus dan edit data pegawai.
- f. Pada *class* menu, memiliki fungsi untuk menampilkan menu yang dapat dipilih oleh *user*.
- g. Pada *class admin*, memiliki fungsi untuk masuk kedalam menu admin.
- h. Pada *class* pengelola,, memiliki fungsi untuk masuk sebagai pengelola.
- i. Pada *class staff* divisi, memiliki fungsi untuk masuk sebagai staff administrasi divisi.
- j. Pada *class* menu\_utama, memiliki fungsi untuk menampilkan *setting* yang telah diatur oleh admin dan pengelola, menampilkan divisi, profil perusahaan dan forum.
- k. Pada *class* berita, memiliki fungsi untuk menampilkan berita yang telah ditambahkan.
- l. Pada *class* galeri, memiliki fungsi untuk menampilkan galeri setiap divisi yang telah ditambahkan.

*Class* admin memiliki *association* dengan *class* menu, artinya *class* admin berhak mengakses apa saja yang tersedia di dalam *class* menu melalui *method* *getmenu()*.

*Class* menu memiliki *composition* dengan *class* *user login*, data pegawai, *setting*, manajemen berita dan data divisi. Relasi tersebut berarti *class* data pegawai, data divisi, manajemen berita, *setting* dan *user login* merupakan komponen penyusun dari *class* menu. Dengan relasi *composition*, tergambar jelas bahwa *class* data pegawai, data divisi, manajemen berita, *setting* dan *user login* tidak dapat berdiri sendiri untuk dapat digunakan melainkan harus berasal dari *class* menu terlebih dahulu.

*Class* pengelola memiliki *association* dengan *class* *setting*, artinya *class* pengelola boleh melihat data *setting* melalui *method* *view()*, dapat menambah data *setting* melalui *method* *add()*, dapat menghapus data *setting* melalui *method* *delete()*, dan dapat mengubah data *setting* melalui *method* *edit()*.

*Class* pengelola memiliki *association* dengan *class* manajemen berita, artinya *class* pengelola boleh melihat data manajemen berita melalui *method* *view()*, dapat menambah data manajemen berita melalui *method* *add()*, dapat menghapus data manajemen berita melalui *method* *delete()*, dan dapat mengubah data manajemen berita melalui *method* *edit()*.

*Class* pengelola memiliki *association* dengan *class* data pegawai, artinya *class* pengelola boleh melihat data pegawai melalui *method* *view()* dan mencari data pegawai melalui *method* *search()*.

*Class* pengelola memiliki *association* dengan *class* data divisi, artinya *class* pengelola boleh melihat data divisi melalui *method* *view()* dan mencari data divisi melalui *method* *search()*.

*Class* staff divisi memiliki *association* dengan *class* data divisi, artinya *class* staff divisi boleh melihat data divisi melalui *method* *view()*, dapat menambah data divisi melalui *method* *add()*, dapat menghapus data divisi melalui *method* *delete()*, dan dapat mengubah data divisi melalui *method* *edit()*.

*Class* staff divisi memiliki *association* dengan *class* data pegawai, artinya *class* divisi boleh melihat data pegawai melalui *method* *view()* dan mencari data pegawai melalui *method* *search()*.

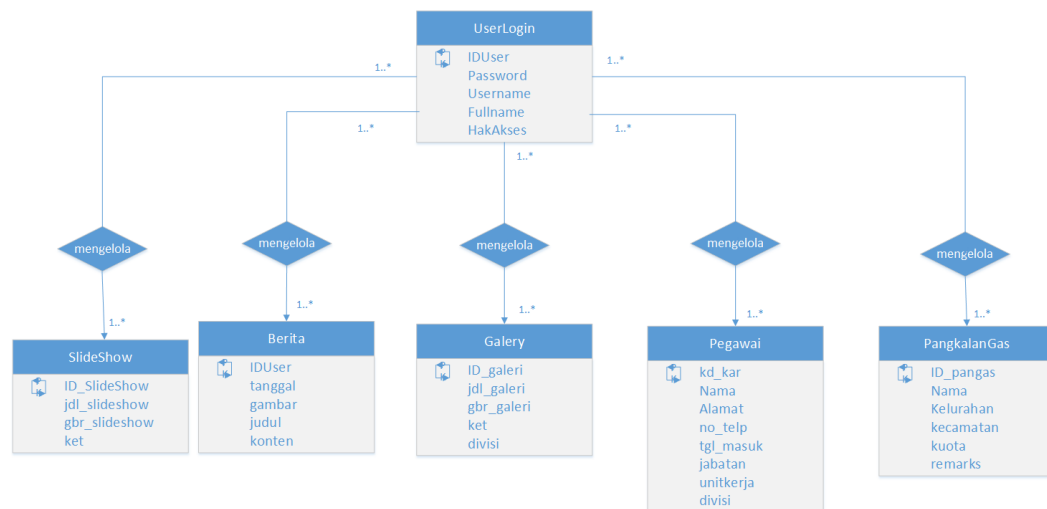
*Class* menu\_utama dengan *class* berita, galeri dihubungkan dengan *composition*. Artinya ada dua *class* penyusun *class* menu\_utama. *Class* penyusun tersebut juga tidak dapat digunakan jika tidak difungsikannya *class* menu\_utama, begitu juga sebaliknya.

*Class* galeri memiliki *composition* dengan *class* setting artinya *class* galeri merupakan bagian dari *class* setting. *Class* galeri tidak dapat berdiri sendiri apabila *class* setting tidak ada. *Class* galeri dapat mengambil data *setting* melalui *method* *getsetting()* dan menampilkan data galeri melalui *method* *setsetting()*.

*Class* berita memiliki *composition* dengan *class* manajemen berita artinya *class* berita merupakan bagian dari *class* manajemen berita. *Class* berita tidak dapat berdiri sendiri apabila *class* manajemen berita tidak ada. *Class* berita dapat mengambil data manajemen berita melalui *method* *getid()* dan menampilkan data berita melalui *method* *setid()*.

#### **3.6.4 ER Diagram**

Gambaran ER Diagram yang digunakan dalam aplikasi dapat dilihat pada gambar Gambar 3.9.



**Gambar 3.9** ER Diagram

Pada gambar 3.9 dapat dilihat *database* yang dirancang memiliki 6 buah entitas yaitu:

- a. *User Login*
- b. Pegawai
- c. *Slide Show*
- d. Berita
- e. Galeri
- f. Pangkalan Gas

Relasi *many-to-many* antara entitas *user login* dengan entitas pegawai, entitas *slide show*, berita, galeri dan pangkalan gas mempunyai arti bahwa *user login* dapat mempunyai banyak data pada pegawai, *slide show*, berita, galeri dan pangkalan gas.

**Tabel 3.2** Struktur tabel *UserLogin*

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id	PK	NN	Int	
2	Username		NN	Varchar	10
3	Password		NN	Varchar	10
4	Fullname		NN	Varchar	30
5	Hak_akses		NN	Int	11
6	Akses		NN	Varchar	10

**Tabel 3.3** Struktur tabel Data Pegawai

<b>No</b>	<b>Field Name</b>	<b>Key Type</b>	<b>Null</b>	<b>Data Type</b>	<b>Max. Length</b>
1	Kd_kar	PK	NN	Int	
2	Nama		NN	Varchar	30
3	Alamat		NN	Varchar	65
4	No_telp		NN	Varchar	12
5	Tgl_masuk		NN	Date	
6	Jabatan		NN	Varchar	20
7	Unitkerja		NN	Varchar	35
8	Divisi		NN	Varchar	22

**Tabel 3.4** Struktur tabel Berita

<b>No</b>	<b>Field Name</b>	<b>Key Type</b>	<b>Null</b>	<b>Data Type</b>	<b>Max. Length</b>
1	Id	PK	NN	Int	
2	Tanggal		NN	Date	
3	Gambar		NN	Varchar	40
4	Judul		NN	Varchar	50
5	Konten		NN	Text	500

**Tabel 3.5** Struktur tabel Galeri

<b>No</b>	<b>Field Name</b>	<b>Key Type</b>	<b>Null</b>	<b>Data Type</b>	<b>Max. Length</b>
1	Id_galeri	PK	NN	Int	
2	Jdl_galeri		NN	Varchar	30
3	Gbr_galeri		NN	Varchar	40
4	Ket		NN	Text	50
5	Divisi		NN	Varchar	7

**Tabel 3.6** Struktur tabel Pangkalan Gas

<b>No</b>	<b>Field Name</b>	<b>Key Type</b>	<b>Null</b>	<b>Data Type</b>	<b>Max. Length</b>
1	Id_pangas	PK	NN	Int	
2	Nama		NN	Varchar	40
3	Kelurahan		NN	Varchar	15
4	Kecamatan		NN	Varchar	15
5	Kuota		NN	Varchar	3
6	Remarks		NN	Varchar	4

**Tabel 3.7** Struktur tabel *Slide Show*

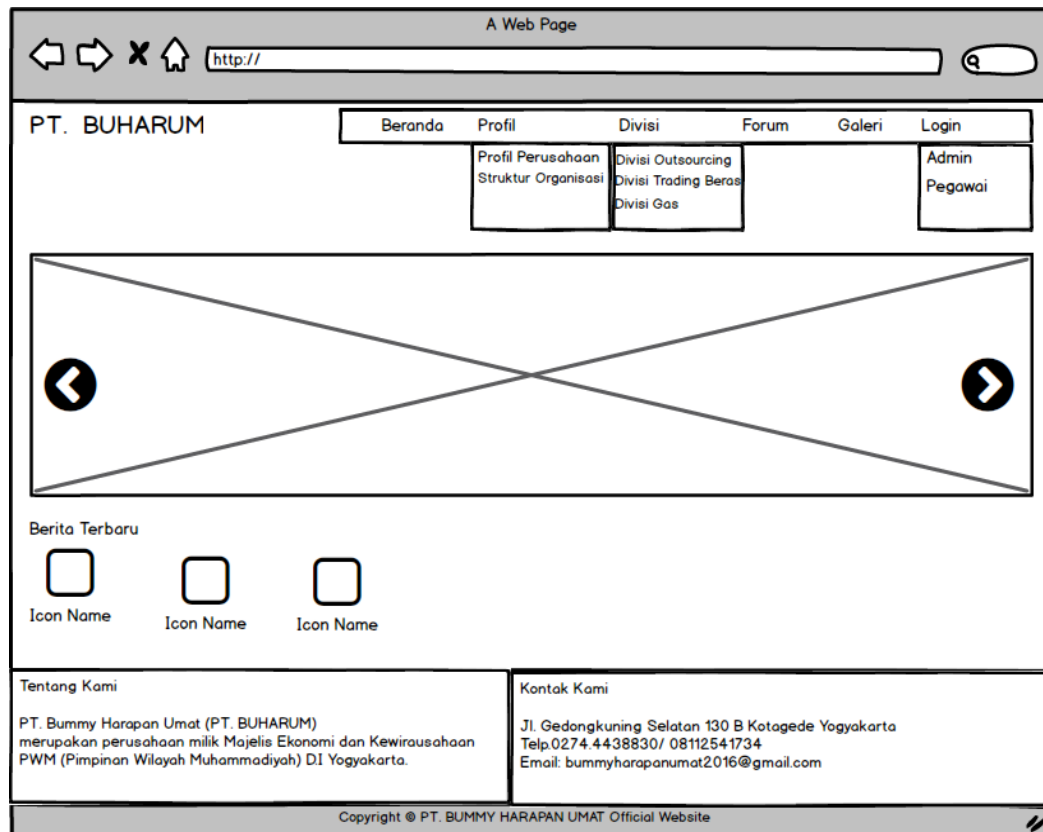
<b>No</b>	<b>Field Name</b>	<b>Key Type</b>	<b>Null</b>	<b>Data Type</b>	<b>Max. Length</b>
1	Id_slideshow	PK	NN	Int	
2	Jdl_slideshow		NN	Varchar	30
3	Gbr_slideshow		NN	Varchar	30
4	Ket		NN	Text	50

### 3.6.5 Rancangan Antarmuka (*User Interface*)

Rancangan tampilan antarmuka (*interface*) adalah gambaram secara umum mengenai tampilan *website*. *User Interface* sangatlah penting dalam suatu aplikasi karena merupakan bagian dari perangkat lunak yang menjadi sarana komunikasi antara pengguna dengan sistem.

### 3.6.5.1 Rancangan Antarmuka Halaman Utama *Public*

Rancangan antarmuka halaman utama *public* merupakan tampilan utama dari *website* yang dapat dilihat oleh *public*. Gambaran rancangan antarmuka halaman utama *website* dapat dilihat pada Gambar 3.10.

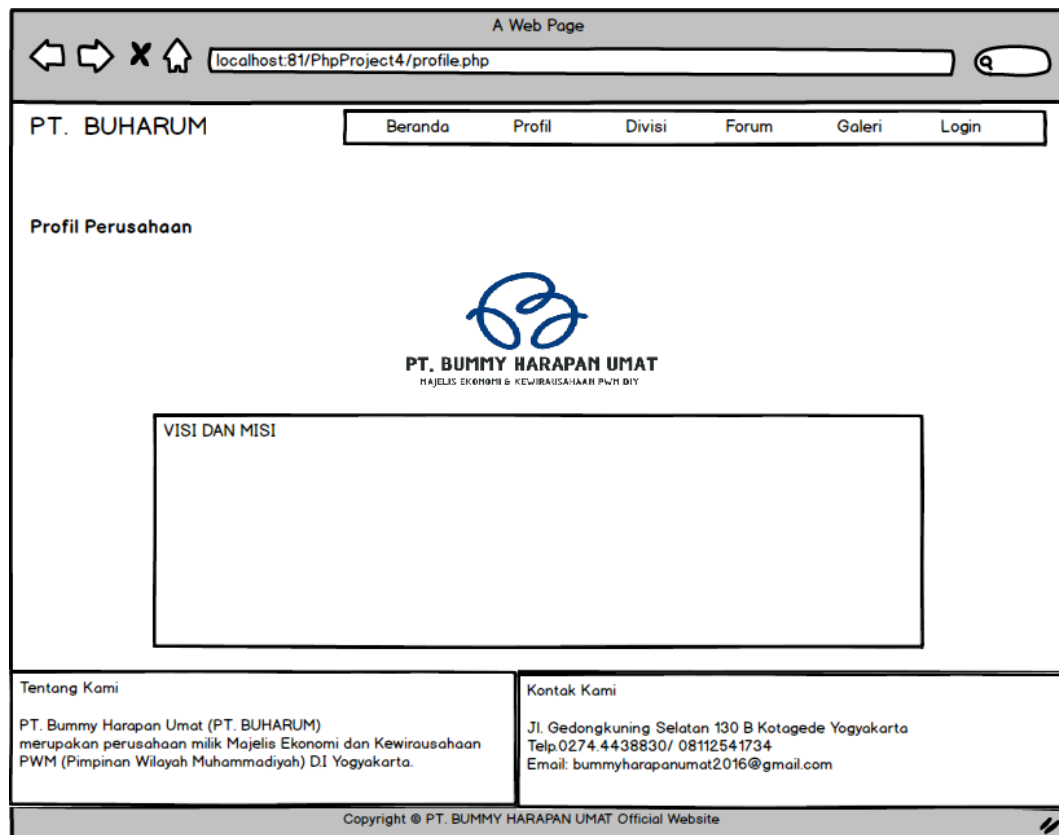


**Gambar 3.10** Rancangan Halaman Utama *Public*

### 3.6.5.2 Rancangan antarmuka halaman profil

Rancangan antarmuka halaman profil menampilkan profil perusahaan PT. BUHARUM yang berisi logo dan visi misi dari perusahaan tersebut. Gambaran rancangan antarmuka halaman profil dapat dilihat pada Gambar 3.11.

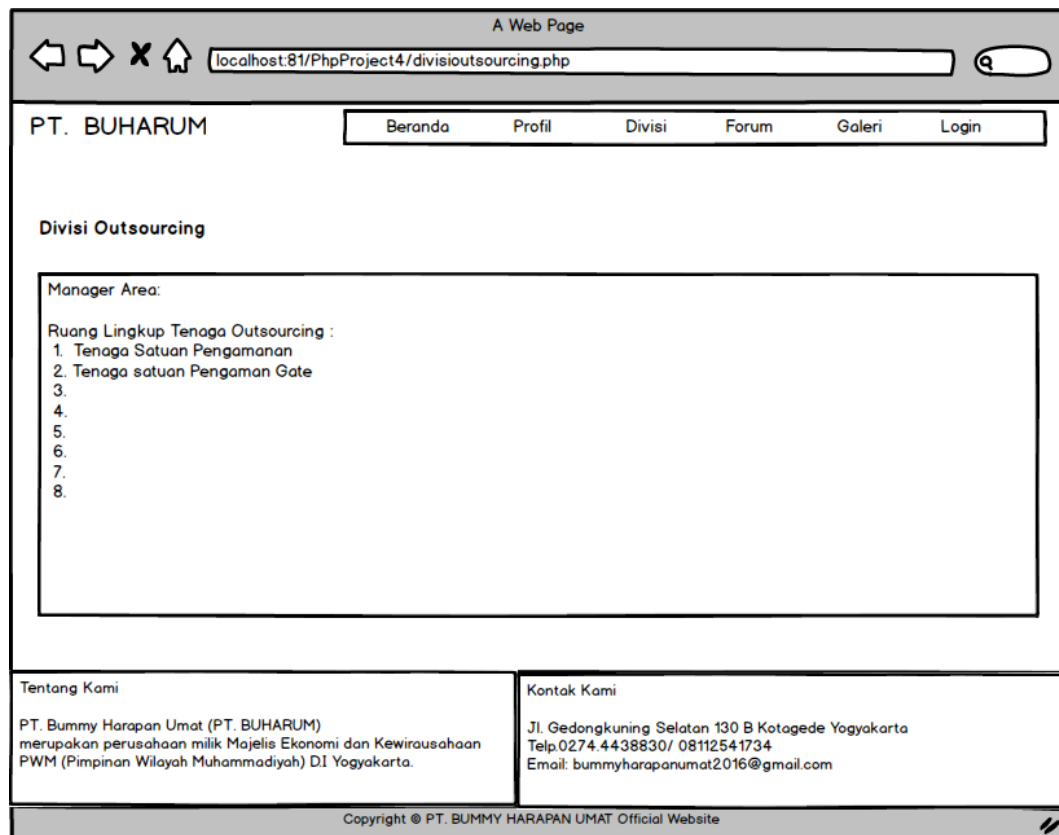




**Gambar 3.11** Rancangan Halaman Profil

### 3.6.5.3 Rancangan antarmuka halaman divisi

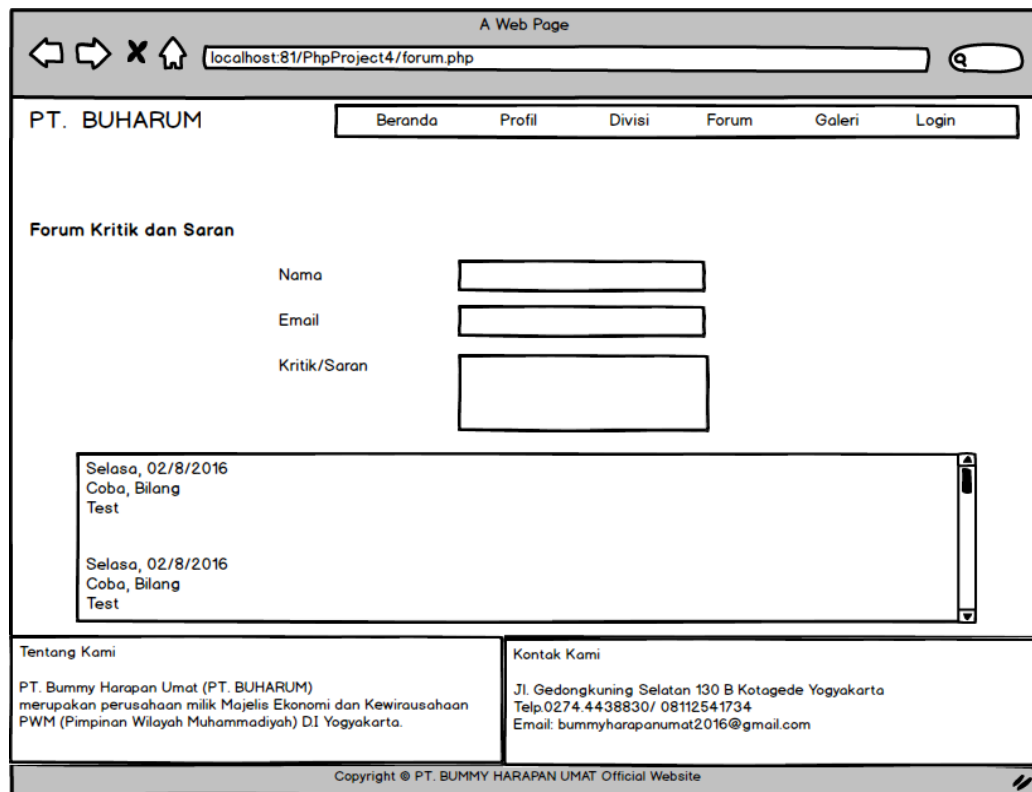
Rancangan antarmuka halaman divisi berfungsi untuk menampilkan divisi-divisi yang ada di perusahaan PT. BUHARUM. Gambaran rancangan antarmuka halaman divisi dapat dilihat pada Gambar 3.12.



**Gambar 3.12** Rancangan Halaman Divisi

### 3.6.5.4 Rancangan antarmuka halaman forum

Rancangan antarmuka halaman forum merupakan halaman yang digunakan untuk memasukkan saran, kritik atau pertanyaan terhadap perusahaan. Gambaran rancangan antarmuka halaman forum dapat dilihat pada Gambar 3.13.



A Web Page

localhost:81/PhpProject4/forum.php

PT. BUHARUM

Beranda Profil Divisi Forum Galeri Login

Forum Kritik dan Saran

Nama

Email

Kritik/Saran

Selasa, 02/8/2016  
Coba, Bilang  
Test

Selasa, 02/8/2016  
Coba, Bilang  
Test

Tentang Kami

PT. Bummy Harapan Umat (PT. BUHARUM)  
merupakan perusahaan milik Majelis Ekonomi dan Kewirausahaan  
PWM (Pimpinan Wilayah Muhammadiyah) DI Yogyakarta.

Kontak Kami

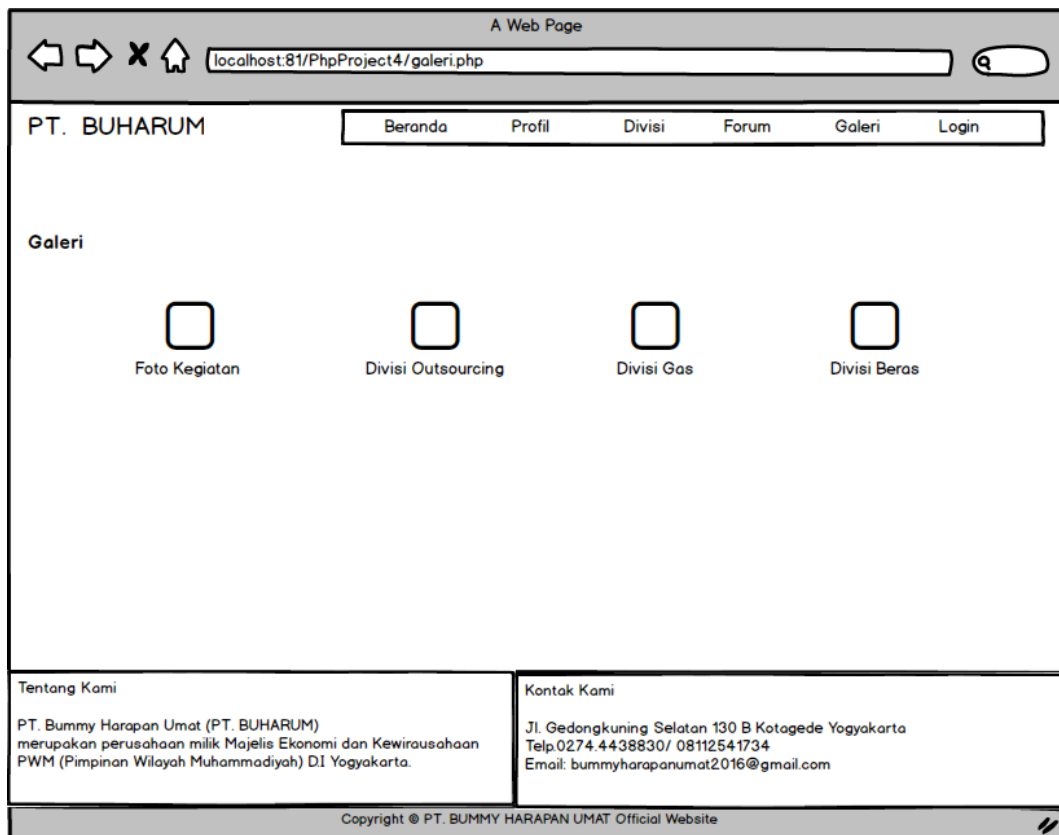
Jl. Gedongkuning Selatan 130 B Kotagede Yogyakarta  
Telp.0274.4438830/ 08112541734  
Email: bummyharapanumat2016@gmail.com

Copyright © PT. BUMMY HARAPAN UMAT Official Website

**Gambar 3.13** Rancangan Halaman Forum

### 3.6.5.5 Rancangan antarmuka halaman galeri

Rancangan antarmuka halaman galeri merupakan halaman yang digunakan untuk menampilkan foto-foto kegiatan perusahaan sehingga masyarakat bisa mengetahui dan melihat kegiatan-kegiatan pegawai di perusahaan tersebut. Gambaran rancangan antarmuka halaman galeri dapat dilihat pada Gambar 3.14.



**Gambar 3.14** Rancangan Halaman Galeri

### 3.6.5.6 Rancangan antarmuka halaman *login*

Rancangan antarmuka halaman *login* merupakan halaman yang berfungsi untuk dapat masuk ke halaman utama untuk admin, pengelola dan staff administrasi divisi. Gambaran rancangan antarmuka menu *login* dapat dilihat pada Gambar 3.15.

A Web Page

localhost:81/PhpProject4/login.php

PT. BUHARUM

Beranda Profil Divisi Forum Galeri Login

Login

Username

Password

Sign in

Tentang Kami

PT. Bummy Harapan Umat (PT. BUHARUM)  
merupakan perusahaan milik Majelis Ekonomi dan Kewirausahaan  
PWM (Pimpinan Wilayah Muhammadiyah) DI Yogyakarta.

Kontak Kami

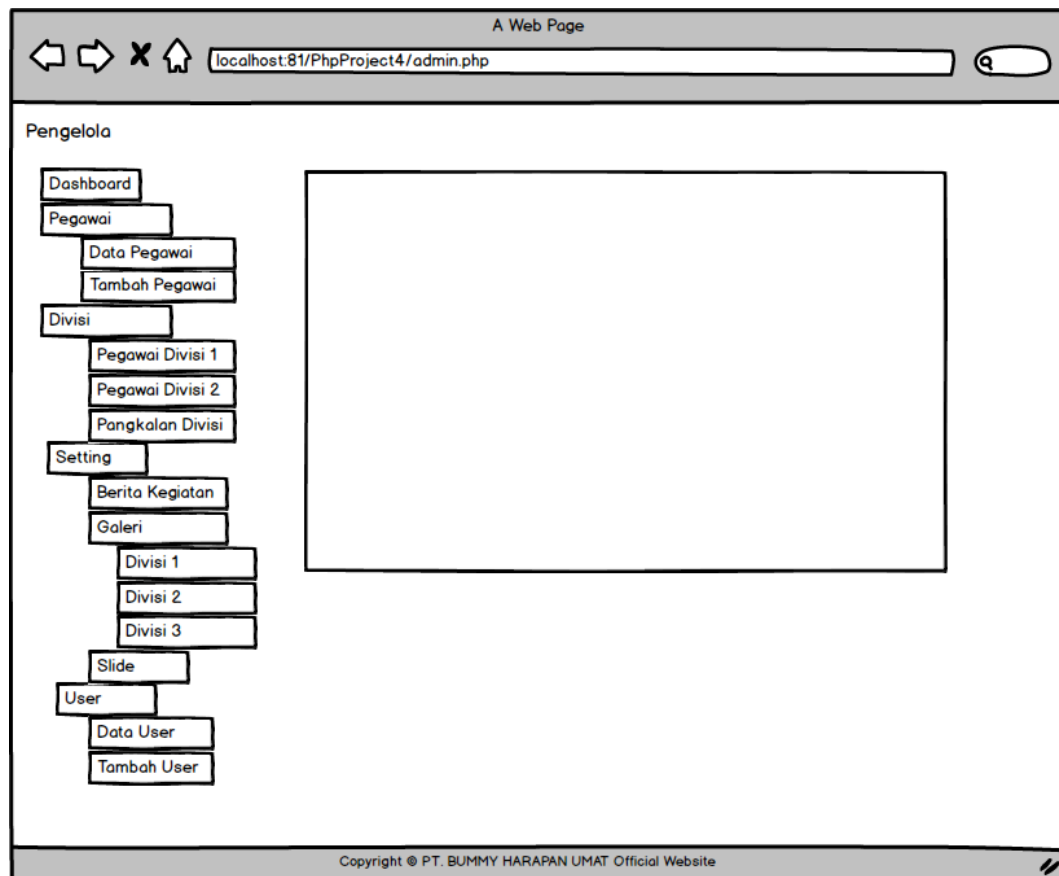
Jl. Gedongkuning Selatan 130 B Kotogede Yogyakarta  
Telp.0274.4438830/ 08112541734  
Email: bummyharapanumat2016@gmail.com

Copyright © PT. BUMMY HARAPAN UMAT Official Website

**Gambar 3.15** Rancangan Halaman *Login*

### 3.6.5.7 Rancangan antarmuka halaman utama admin

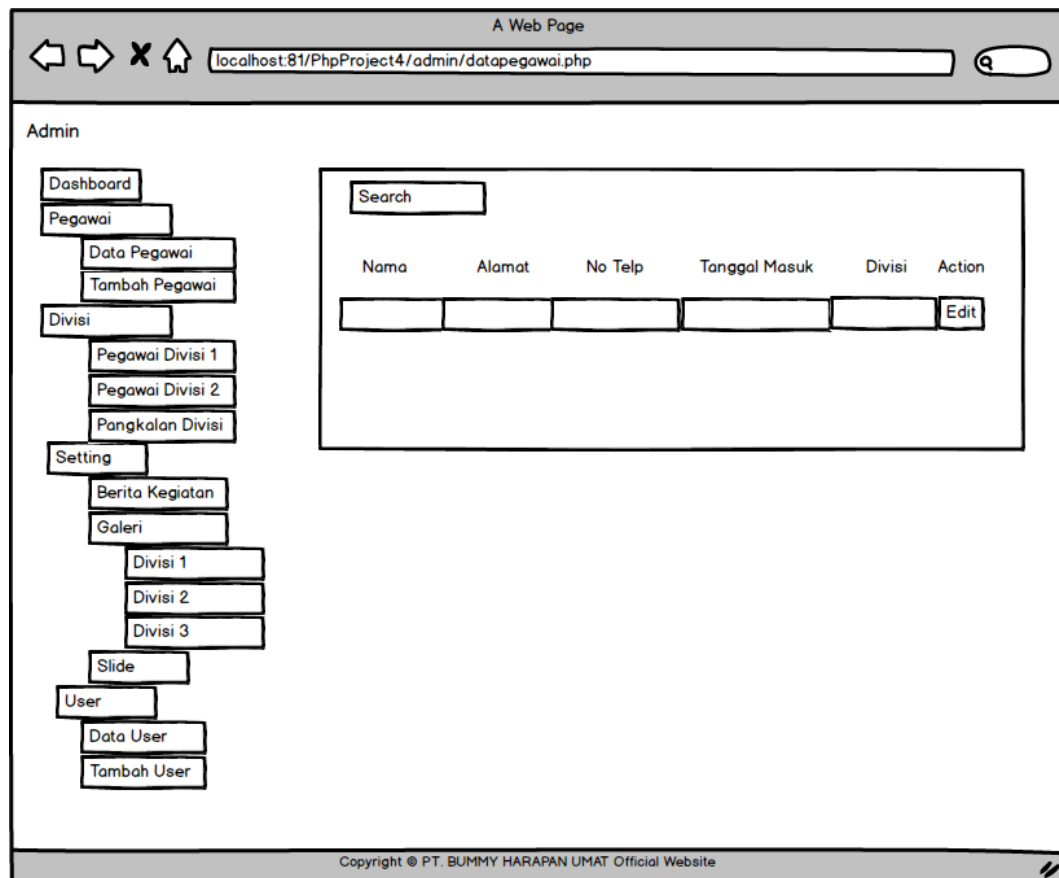
Rancangan antarmuka halaman utama admin merupakan tampilan utama untuk admin yang di dalamnya terdapat beberapa pilihan menu utama yang dapat diakses oleh admin. Gambaran rancangan halaman utama admin dapat dilihat pada Gambar 3.16.



**Gambar 3.16** Rancangan Halaman Utama Admin

### 3.6.5.8 Rancangan antarmuka halaman Data Pegawai

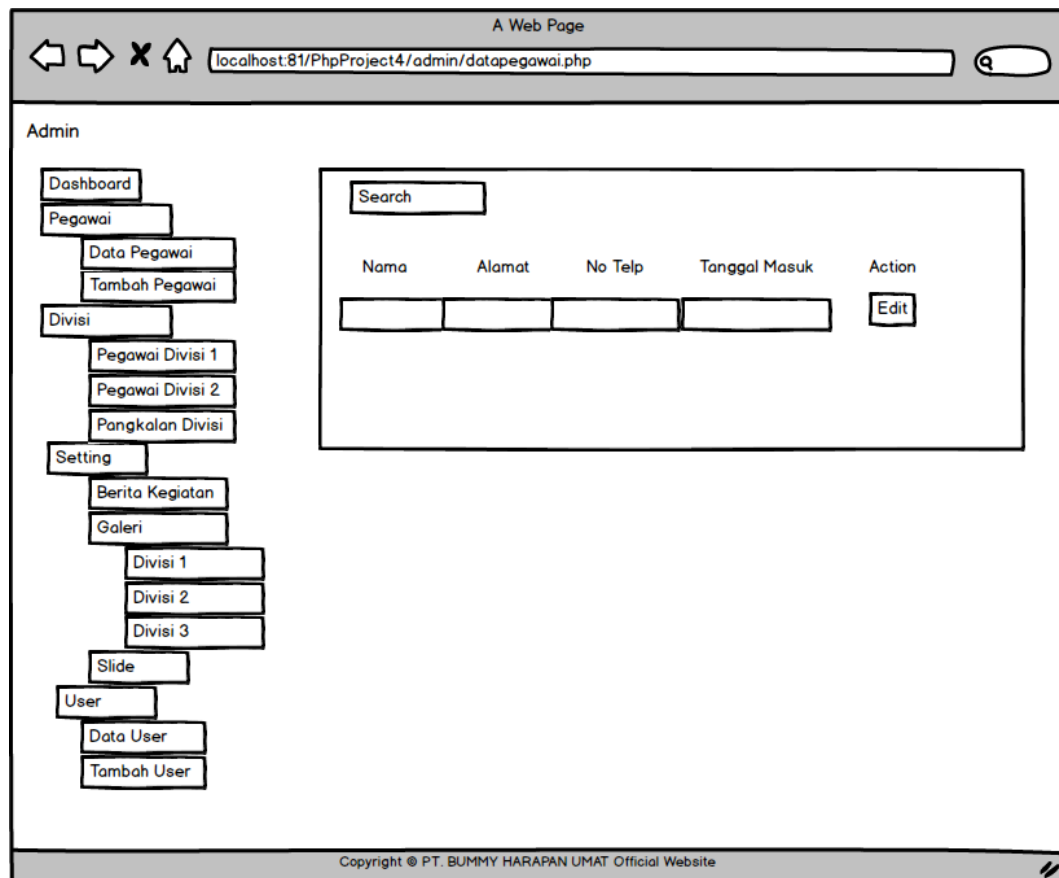
Rancangan antarmuka halaman data pegawai berfungsi untuk menampilkan semua data pegawai di PT. BUHARUM. Pada halaman ini juga admin dapat menambah, mengedit, mencari ataupun menghapus data pegawai tersebut. Gambaran rancangan antarmuka halaman data pegawai dapat dilihat pada Gambar 3.17.



**Gambar 3.17** Rancangan Halaman Utama Data Pegawai

### 3.6.5.9 Rancangan antarmuka halaman divisi

Rancangan antarmuka halaman divisi berfungsi untuk menampilkan langsung data perdivisi. Diantaranya data pangkalan gas dari divisi gas. Gambaran rancangan antarmuka halaman divisi dapat dilihat pada Gambar 3.18.

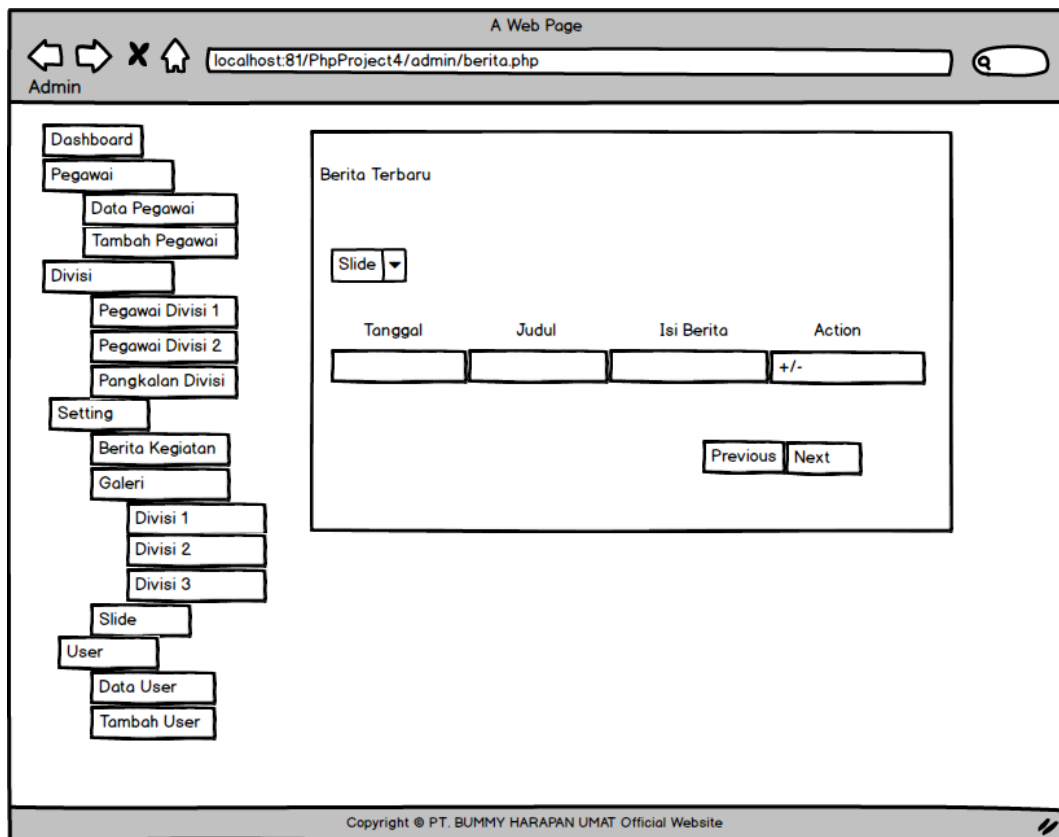


**Gambar 3.18** Rancangan Halaman Utama Divisi

### 3.6.5.10 Rancangan antarmuka halaman berita

Rancangan antarmuka halaman berita berfungsi untuk menampilkan berita di halaman utama *public*. Di halaman ini juga admin dan pengelola bisa menambahkan berita. Gambaran rancangan antarmuka halaman berita dapat dilihat pada Gambar 3.19.

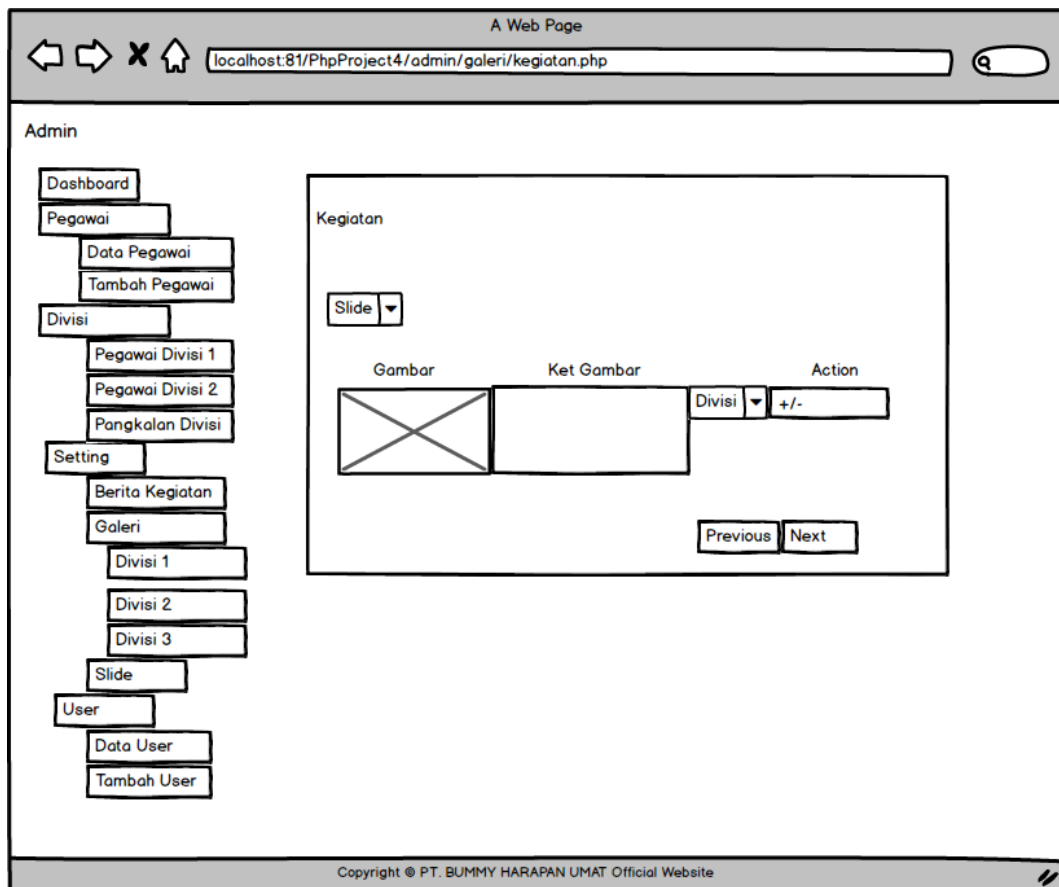




**Gambar 3.19** Rancangan Halaman Berita

### 3.6.5.11 Rancangan antarmuka halaman galeri

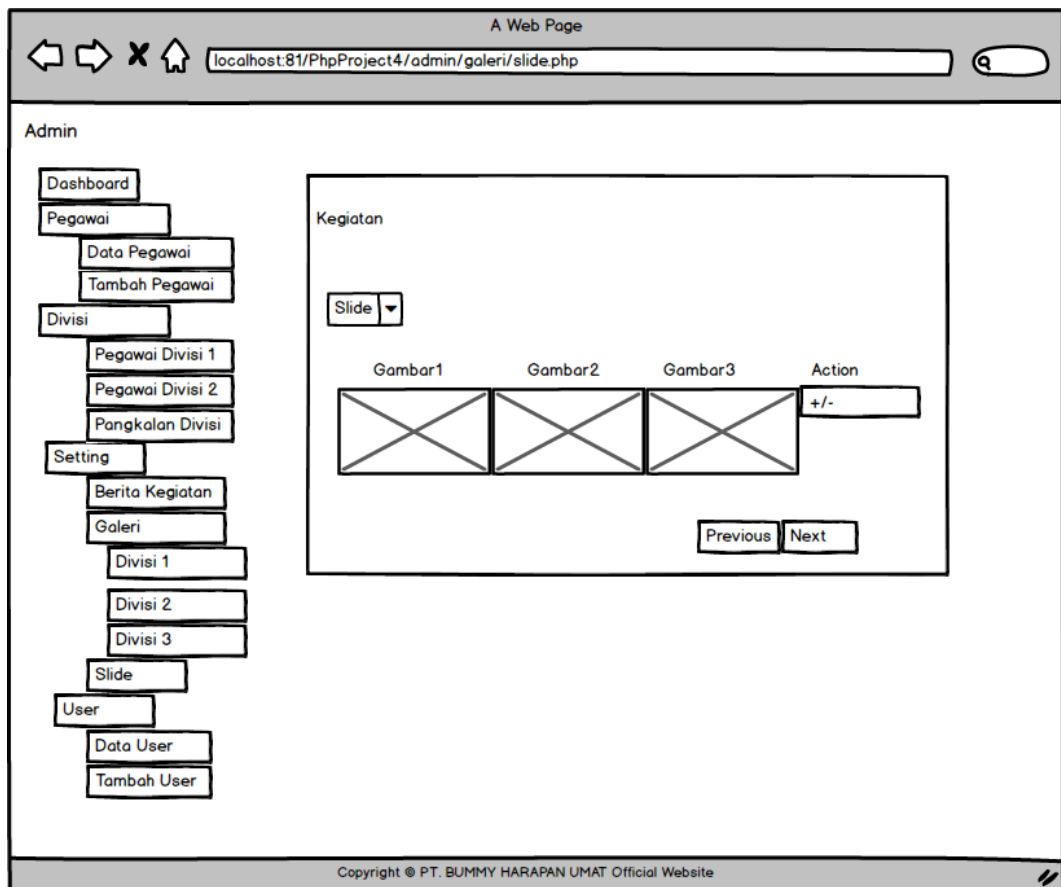
Rancangan antarmuka halaman galeri berfungsi untuk menampilkan dan menambahkan galeri foto perdivisi di halaman galeri pada halaman depan *public*. Gambaran rancangan antarmuka halaman galeri dapat dilihat pada Gambar 3.20.



**Gambar 3.20** Rancangan Halaman Galeri

### 3.6.5.12 Rancangan antarmuka halaman *slide*

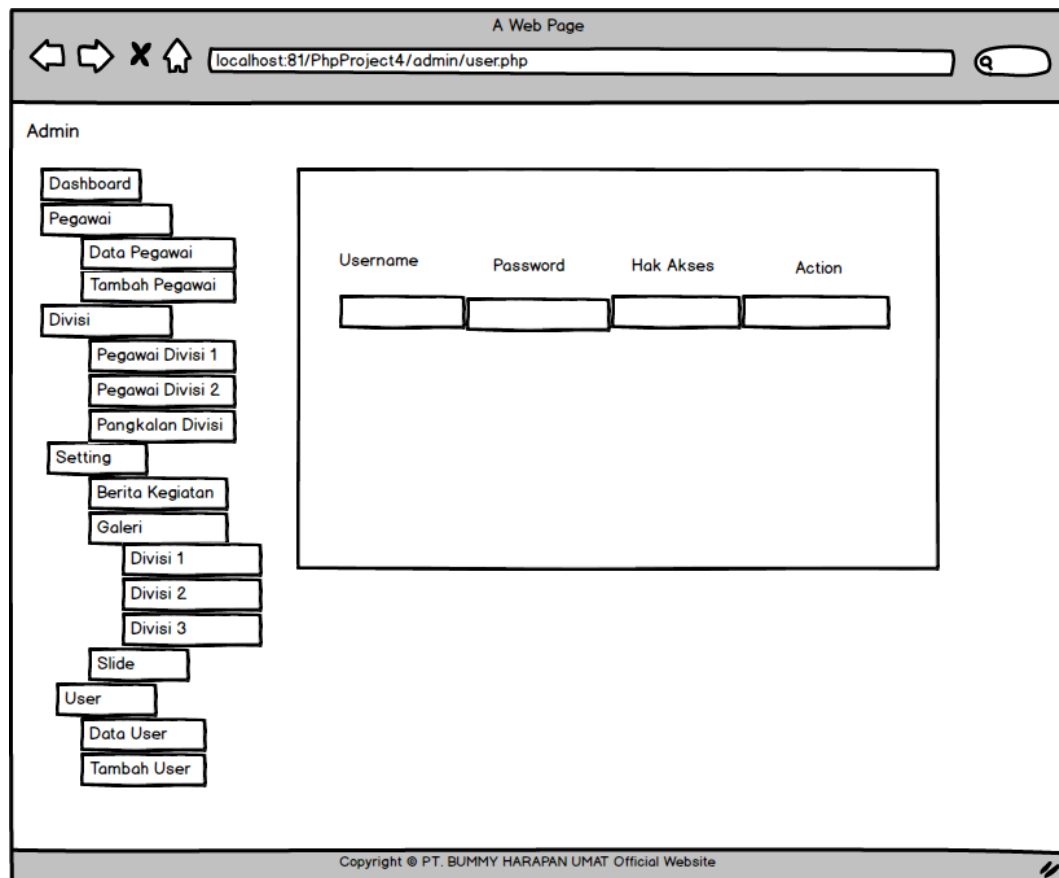
Rancangan antarmuka halaman *slide* berfungsi untuk mengatur gambar yang akan ditampilkan di *slide show* halaman utama *website*. Gambaran rancangan antarmuka halaman *slide* dapat dilihat pada Gambar 3.21.



**Gambar 3.21** Rancangan Halaman *Slide*

### 3.6.5.13 Rancangan antarmuka halaman *user*

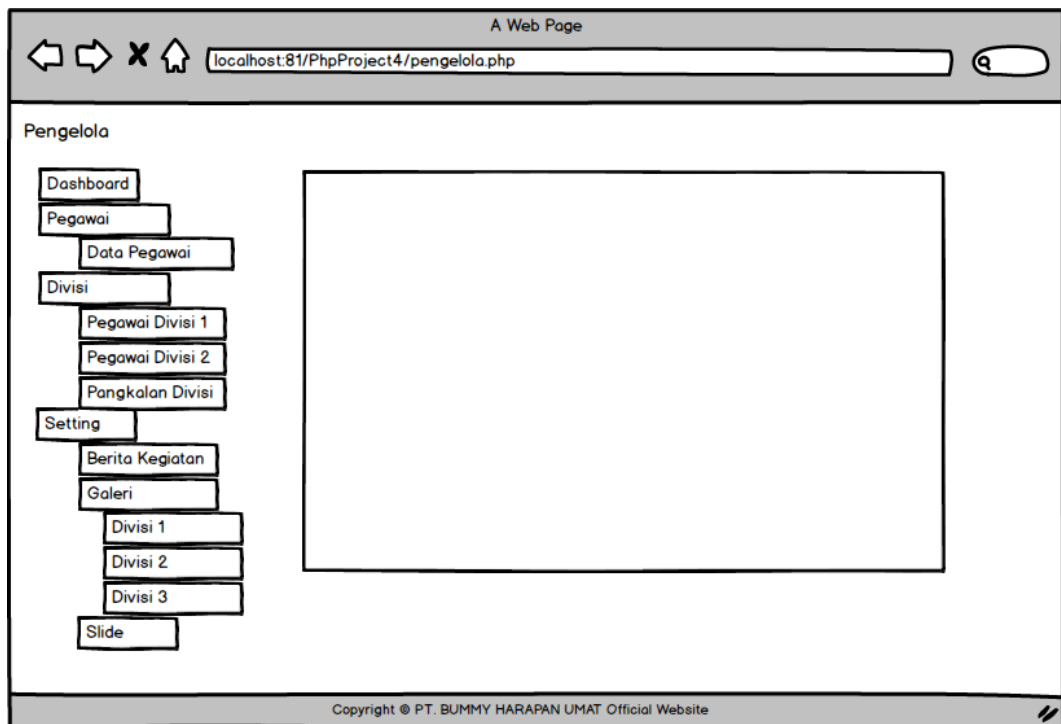
Rancangan antarmuka halaman *user* berfungsi untuk mengatur *user login*. Sehingga disini hanya admin yang dapat melihat *password user* dan mengatur *user* tersebut. Gambaran rancangan antarmuka halaman *user* dapat dilihat pada Gambar 3.22.



**Gambar 3.22** Rancangan Halaman *User*

#### 3.6.5.14 Rancangan antarmuka halaman utama pengelola

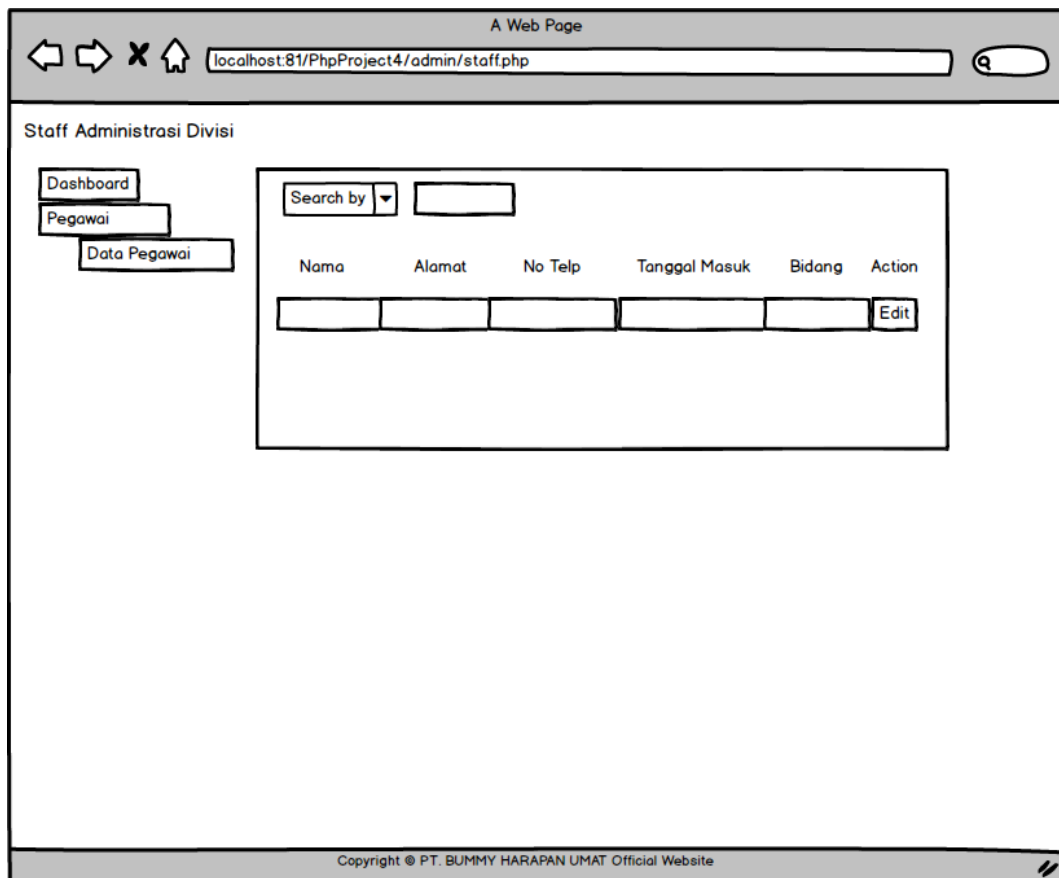
Rancangan antarmuka halaman utama pengelola menampilkan menu-menu pilihan yang dapat diatur oleh pengelola tersebut. Bedanya dengan admin adalah pengelola tidak dapat menambah, menghapus, mengedit pegawai dan data divisi. Pengelola hanya dapat melihat dan mencari data pegawai dan data divisi. Pengelola juga tidak bisa melihat dan mengatur *user login* yang seperti ada di halaman admin. Gambaran rancangan antarmuka halaman pengelola dapat dilihat pada Gambar 3.23.



**Gambar 3.23** Rancangan Halaman Utama Pengelola

#### **3.6.5.15 Rancangan antarmuka halaman utama staff administrasi divisi**

Rancangan antarmuka halaman utama staff administrasi divisi berfungsi untuk staff administrasi setiap divisi. Staff masing masing divisi hanya dapat mengatur data divisinya masing-masing. Gambaran rancangan antarmuka halaman utama staff administrasi dapat dilihat pada Gambar 3.24.



**Gambar 3.24** Rancangan Halaman Utama Staff Administrasi Divisi

### 3.7 Pengujian

Pengujian perangkat lunak merupakan suatu kegiatan yang bertujuan untuk mendapatkan informasi dan mengevaluasi kualitas dari layanan atau produk yang sedang diuji.

Metode pengujian yang dipakai dalam pengembangan aplikasi adalah *black box testing*. *Black box testing* atau tes fungsional adalah pengujian yang dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak yang dikembangkan.

Hal-hal yang menjadi perhatian dalam pengujian adalah sebagai berikut:

- a. Aplikasi dapat menyimpan, mengubah dan menghapus data pegawai.
- b. Aplikasi dapat mencari data pegawai berdasarkan nama, divisi dan tahun masuk.

- c. Aplikasi dapat menampilkan galeri foto dari halaman pengelola dan admin berdasarkan divisi.
- d. Aplikasi dapat menampilkan berita pada halaman utama *public*

