

BAB III

METODELOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini mengambil tempat di Kota Yogyakarta dengan objek penelitian masyarakat pengguna majalah *online* berupa *website*. Adapun penelitian dilaksanakan mulai tanggal 30 Maret 2016 sampai dengan tanggal 10 Juni 2016.

3.2 Jenis Data dan Teknik Pengumpulan Data

Jenis data yang digunakan adalah data primer dan data sekunder, data primer merupakan data yang diperoleh atau dikumpulkan oleh peneliti secara langsung dari sumber datanya. Data tersebut didapat dari sumber data yang dikumpulkan secara khusus yang berhubungan langsung dengan permasalahan yang diteliti. Sedangkan data sekunder merupakan data yang sudah tersedia sehingga kita tinggal mencari dan mengumpulkan. Walaupun data sekunder secara fisik sudah tersedia, namun data yang diambil dan dikumpulkan harus disaring serta diambil intisari data yang akan digunakan untuk penelitian.

Teknik pengumpulan data dalam penelitian ini adalah dengan metode angket (kuesioner), yakni teknik pengumpulan data yang dilakukan dengan cara memberikan seperangkat pertanyaan atau pernyataan kepada orang lain yang dijadikan responden dan kemudian responden diminta menjawab sesuai dengan pendapat mereka.

3.3 Teknik Pengambilan Sampel

Pengambilan sampel dalam penelitian ini menggunakan teknik *convenience sampling* yang artinya mengambil sampel menurut kemudahan untuk mengakses sampel tersebut dan anggota populasi tersebut tidak mempunyai peluang yang sama untuk terpilih menjadi sampel. Dalam teknik ini, peneliti tidak mempunyai pertimbangan lain kecuali berdasarkan kemudahan saja.

3.4 Analisis Kebutuhan

3.4.1 Analisis Kebutuhan Sistem Fungsional

1. Administrator

- a. Sistem harus dapat melakukan *entry* berita dan informasi.
 - i. Pengguna bisa memasukkan berbagai jenis berita dan informasi dengan membedakan berdasarkan kategori pada rubrik *website*.
 - ii. Pengguna dapat memilah data dari kiriman *member website* untuk *diposting* pada *website*.
 - iii. Pengguna dapat menolak data dari kiriman *member website* jika tidak layak untuk *diposting* pada *website*.
 - iv. Pengguna dapat mengedit data dari kiriman *member website* jika ada beberapa konten yang harus diubah sebelum *diposting* pada *website*.
- b. Sistem harus dapat melakukan pendataan *user* (konsumen).
 - i. Pengguna dapat menampilkan konsumen yang terdaftar.
 - ii. Pengguna bisa menghapus data *member*.

2. Editor

Sistem harus dapat melakukan :

- i. Pengguna dapat melakukan filter terkait berita dari *member*.
- ii. Pengguna dapat melakukan filter terkait foto dari *member*.

3. Member

Sistem harus dapat melakukan :

- i. Pengguna dapat memberikan komentar pada berita dan informasi yang *diposting* oleh administrator.
- ii. Pengguna dapat memberikan penilaian suka (*like*) pada berita dan informasi yang *diposting* oleh administrator.
- iii. Pengguna dapat memberikan berita atau informasi untuk *diposting* oleh administrator.

4. Pengguna (*User*)

User hanya akan dapat melakukan *view* atau membaca berita dan informasi yang telah di *post* oleh administrator. Jika *user* ingin memberikan komentar pada berita dan informasi, maka *user* harus *login* dengan *plugin facebook* atau mendaftarkan diri sebagai *member*.

3.4.2 Analisis Kebutuhan Sistem Non Fungsional

1. Operasional

a. Perangkat Keras

Perangkat keras yang digunakan minimal memiliki spesifikasi :

- I. Menggunakan sistem operasi Windows 8 (bisa juga dibuka dengan menggunakan sistem operasi Windows 7, Machintos, Linux).
- II. Proccesor Intel Pentium IV atau lebih.
- III. RAM 256 – 512MB atau lebih.
- IV. Media koneksi internet (modem, LAN, Wi-Fi)

b. Perangkat Lunak

Pembuatan desain pada *website* bahasa pemrograman HTML, CSS dan PHP. Sedangkan untuk penyempurnaan desain grafis agar *website* ini terlihat menarik maka digunakanlah adobe photoshop dan corel draw sebagai pengembang multimedia pada *website*.

Database yang digunakan adalah MySQL yang telah terinstall pada sebuah *hosting*. Selain untuk menyimpan data desain *template* untuk *website* yang berupa gambar maupun file php dan css. *Database* ini juga menyimpan data input yang merupakan hasil masukan dari *software* yang terintegrasi dengan *website* ini yang kemudian nantinya akan ditampilkan sebagian data tersebut pada salah satu halaman *website*. Digunakannya *template* yang telah tersedia di internet yaitu pada situs <http://all-free-download.com/free-website-templates/> berupa CCS HTML5 bertujuan untuk mempermudah pengembangan aplikasi.

2. Keamanan

Sistem keamanan yang diterapkan adalah dengan mencantumkan *username* baik untuk administrator, editor dan *member*.

3. Informasi

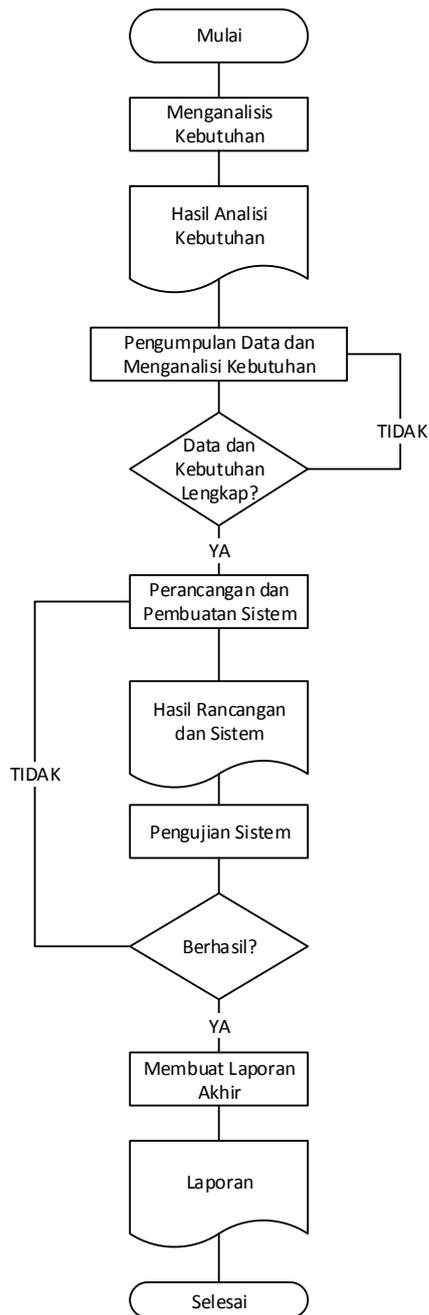
- I. Digunakan untuk menginformasikan kepada pengguna (administrator atau *member*) apabila *username* atau *password* yang dimasukkan salah.
- II. Digunakan untuk menyampaikan berita dan informasi di setiap rubrik pada *website* baik informasi tersebut berasal dari jurnalis *website* atau *member website*.
- III. Administrator mendapatkan informasi terkait *member* yang memberikan data berupa informasi dan berita.

4. Kinerja

- I. Penyampaian informasi dan berita yang lebih masif dari *member website* akan meminimalisir kinerja jurnalis.
- II. *Member* akan mendapatkan informasi berupa data penyampaian informasi dan berita dari *member* atau jurnalis.

3.5 Jalan Penelitian

Tahap pada alur penelitian di atas digambarkan dalam *flow chart* pada gambar 3.3



Gambar 3. 1 Alur penelitian

Penelitian Aplikasi portal berita dengan informasi dua arah serta penyalur opini masyarakat berbasis *website* menggunakan metode SDLC, hal ini bertujuan

apabila dalam perjalanan sistem tersebut terdapat kesalahan, kerusakan 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 mencakup kebutuhan *software* dan kebutuhan *hardware*, analisis kebutuhan isi dan interaksi menu pada aplikasi. Sebelum membuat program aplikasi berbasis *website*, terlebih dahulu melakukan wawancara terhadap beberapa orang terkait penggunaan majalah *online* berupa *website*. Wawancara yang dilakukan akan melengkapi data dari kuisioner yang sangat terbatas jumlah data yang dapat diterima. Wawancara akan memberikan data terkait bagaimana fitur-fitur yang diharapkan oleh pengguna penggunaan majalah *online* berupa *website* agar *website* yang dibuat sesuai dengan harapan pengguna.

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. Selain itu, *requirement* juga dapat membantu dalam melakukan *testing* ketika aplikasi telah selesai dibuat.

3. Perancangan dan pembuatan sistem

Sebelum sistem atau aplikasi dibuat, penulis membuat rancangan dari aplikasi *website*. Pembuatan rancangan tersebut bertujuan agar *website* yang dibuat dapat sesuai dengan yang diharapkan dan tidak akan ada fitur yang dihilangkan atau tertinggal.

4. Pengujian sistem

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

ataupun tidak sesuai dengan *requirement*. Pengujian akan dilakukan dengan menggunakan metode *balckbox*.

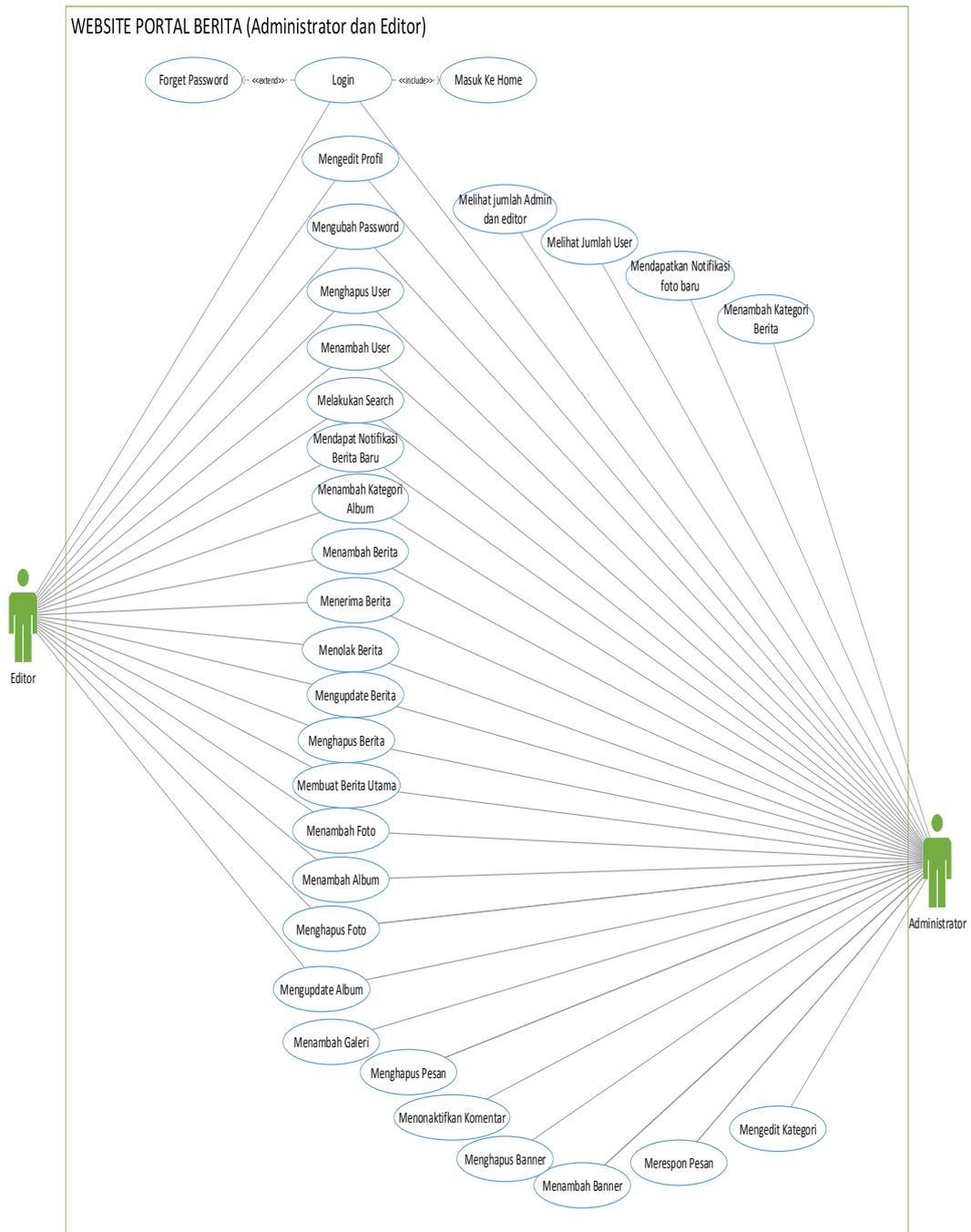
3.6 Perancangan Sistem

3.6.1 Use Case Diagram

Proses pembangunan sistem *website* portal berita alternatif dengan komunikasi informasi berita dua arah dimulai dengan proses merangkum *requirement analysis*. Proses *requirement analysis* merupakan proses untuk mendapatkan persyaratan-persyaratan aplikasi, jalannya aplikasi, fitur aplikasi sampai dengan larangan pada aplikasi. Dalam *requirement*, akan melibatkan beberapa aktor dalam jalannya aplikasi. Aktor yang terlibat dalam aplikasi ini antara lain :

1. Administrator
2. Editor
3. *User* Biasa
4. *Member*

Keempat aktor memiliki aktifitas masing-masing. Aktor yang satu dengan aktor yang lain memiliki persyaratan dan *role* yang dapat digunakan. Penggambarannya akan terlihat dengan *use case* diagram. *Use case* yang akan dibuat dalam sistem secara umum seperti dijelaskan pada gambar 3.2 dan gambar 3.3. Gambar 3.2 merupakan kegiatan-kegiatan dari aktor administrator dan aktor editor. Sedangkan gambar 3.3 merupakan kegiatan-kegiatan dari aktor *member* dan *user* biasa.



Gambar 3. 2 Use case actor administrator dan editor

Berdasarkan Use case Gambar 3.2 dapat dilihat actor administrator dan aktor editor memiliki role yang berbeda.

a. Administrator

Administrator merupakan aktor yang dapat melakukan aktifitas apapun yang terdapat didalam *website*. Administrator dapat diartikan seperti *super user* yang kegiatannya tidak dibatasi kecuali hal-hal yang bersifat rahasia akun. Dilihat dari *Use case* gambar 3.2 administrator dapat melakukan beberapa kegiatan sebagai berikut :

1. *Login*
2. Mengedit profil
3. Mengubah *password*
4. Menghapus *member* atau *user*
5. Menambah *member* atau *user*
6. Melakukan *search*
7. Mendapat *notifikasi* berita baru
8. Mendapat *notifikasi* foto baru
9. Menambah kategori album
10. Menambah kategori berita
11. Menambah berita
12. Menerima berita
13. *Mengupdate* berita
14. Membuat berita utama
15. Menambah foto
16. Menghapus foto
17. *Mengupdate* galeri
18. Menambah galeri
19. Menghapus pesan
20. Menonaktifkan komentar
21. Menghapus *banner*
22. Menambahkan *banner*
23. Merespon pesan
24. Megedit kategori

25. Melihat jumlah admin, editor dan *member*

26. Melihat jumlah *user*

Semua kegiatan yang ada di *website* admin dapat digunakan oleh administrator. Administrator dapat mengelola secara penuh halaman umum *website*. Dalam *usecase* diagram pada gambar 3.2 terdapat penghubung antar *usecase* berupa relasi asosiasi. Selain itu terdapat penghubung *include* yang merupakan penjabar bahwa setelah aktor melakukan kegiatan pada *usecase* terkait maka selanjutnya akan dipindahkan secara otomatis oleh *website* ke *usecase* lainnya sesuai dengan arah panah. Sedangkan penghubung *extend* merupakan penjabar bahwa aktor dapat melakukan kegiatan pada *usecase* terkait ataupun tidak (*optional*).

b. Editor

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

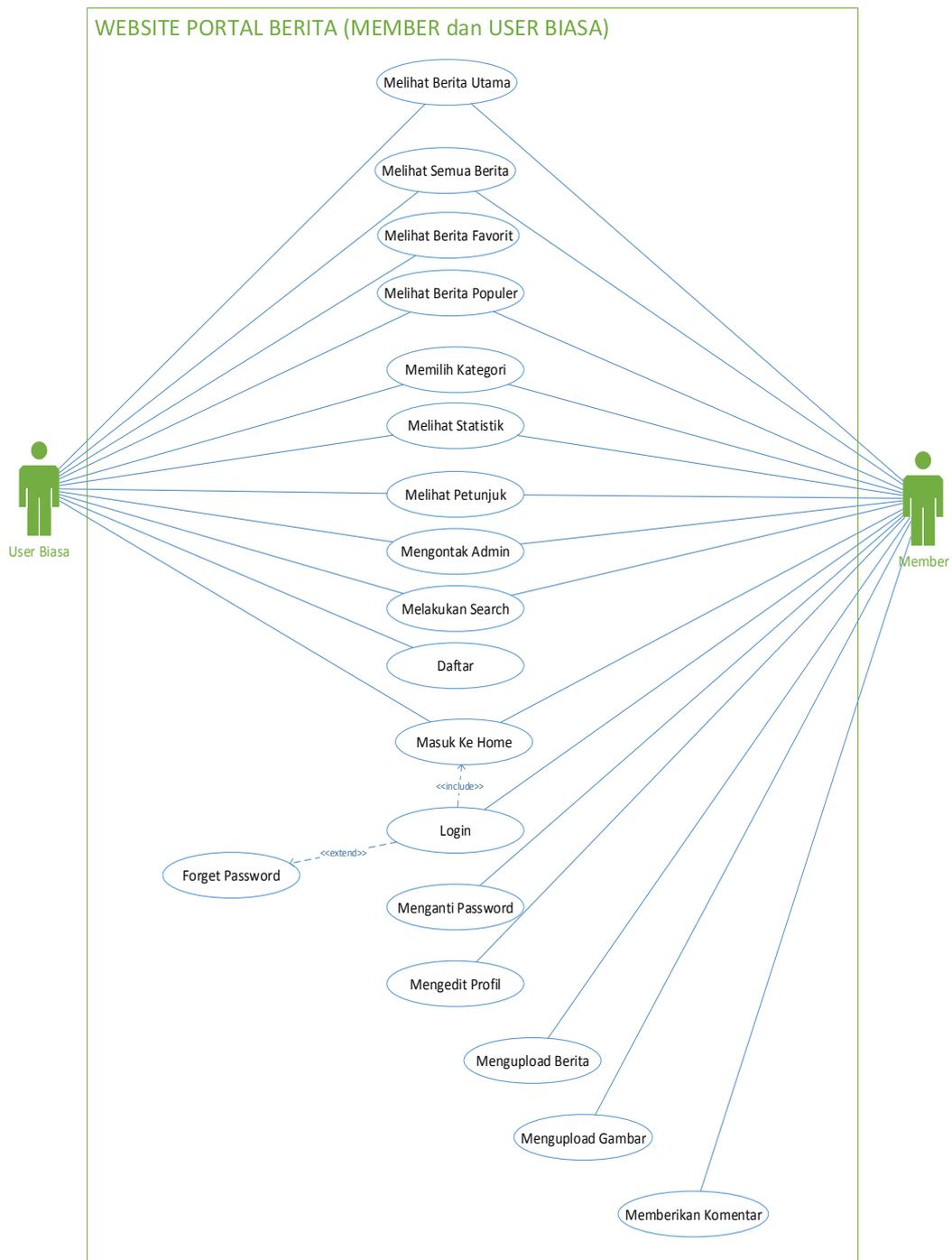
1. *Login*
2. Mengedit profil
3. Mengubah *password*
4. Menghapus *member* atau *user*
5. Menambah *member* atau *user*
6. Melakukan *search*
7. Mendapat notifikasi berita baru
8. Mendapat notifikasi foto baru
9. Menambah kategori album
10. Menambah kategori berita
11. Menambah berita
12. Menerima berita
13. *Mengupdate* berita
14. Membuat berita utama

15. Menambah foto

16. Menghapus foto

17. *Mengupdate gallery*

Dilihat dari daftar kegiatan yang dapat dilakukan oleh editor, dapat diartikan bahwa editor hanya diberikan izin untuk mengelola *website* terkait berita dan foto.



Gambar 3.3 Use case actor member dan user biasa

Berdasarkan Use case Gambar 3.3 terlihat aktor *member* dan *user biasa* memiliki role yang berbeda. Usecase diagram diatas merupakan usecase diagram terkait halaman web utama.

a. *Member*

Aktor *member* merupakan transformasi dari aktor *user* biasa yang telah mendaftarkan dirinya sebagai *member website*. Aktor *member* memiliki kegiatan-kegiatan sebagai berikut :

1. Melihat berita utama
2. Melihat semua berita
3. Melihat berita favorit
4. Melihat berita populer
5. Memilih kategori
6. Melihat statistik
7. Melihat petunjuk
8. Mengontak admin atau mengirim pesan
9. Melakukan *search*
10. Masuk ke *home*
11. *Login*
12. Mengganti *password*
13. Mengedit profil
14. *Mengupload* berita
15. *Mengupload* gambar
16. Memberikan komentar

Fitur-fitur yang ada didalam *website* semua dapat digunakan oleh *member*. Komunikasi dua arah yang dimaksudkan dalam *website* ini terjadi antara aktor *member* dengan aktor administrator atau editor. *Member* dapat memberikan informasi berupa berita ataupun gambar yang di *upload* kedalam sistem, selanjutnya menunggu editor atau administrator untuk mengedit dan mengkonfirmasi berita yang di *upload* oleh *member* tersebut.

b. *User Biasa*

Aktor *user* biasa memiliki hak yang lebih sedikit dibandingkan dengan aktor *member*. Aktor *user* biasa memiliki kegiatan-kegiatan sebagai berikut :

1. Melihat berita utama
2. Melihat semua berita
3. Melihat berita favorit
4. Melihat berita populer
5. Memilih kategori
6. Melihat statistik
7. Melihat petunjuk
8. Mengontak admin atau mengirim pesan ke admin
9. Melakukan *search*
10. Masuk ke *home*
11. Daftar

Ketika pengguna *website* hanya sebagai aktor *user* biasa, maka *website* hanya akan seperti *website* berita lainnya. Hanya kegiatan membaca berita yang dapat dilakukan seperti pada *website* berita lain.

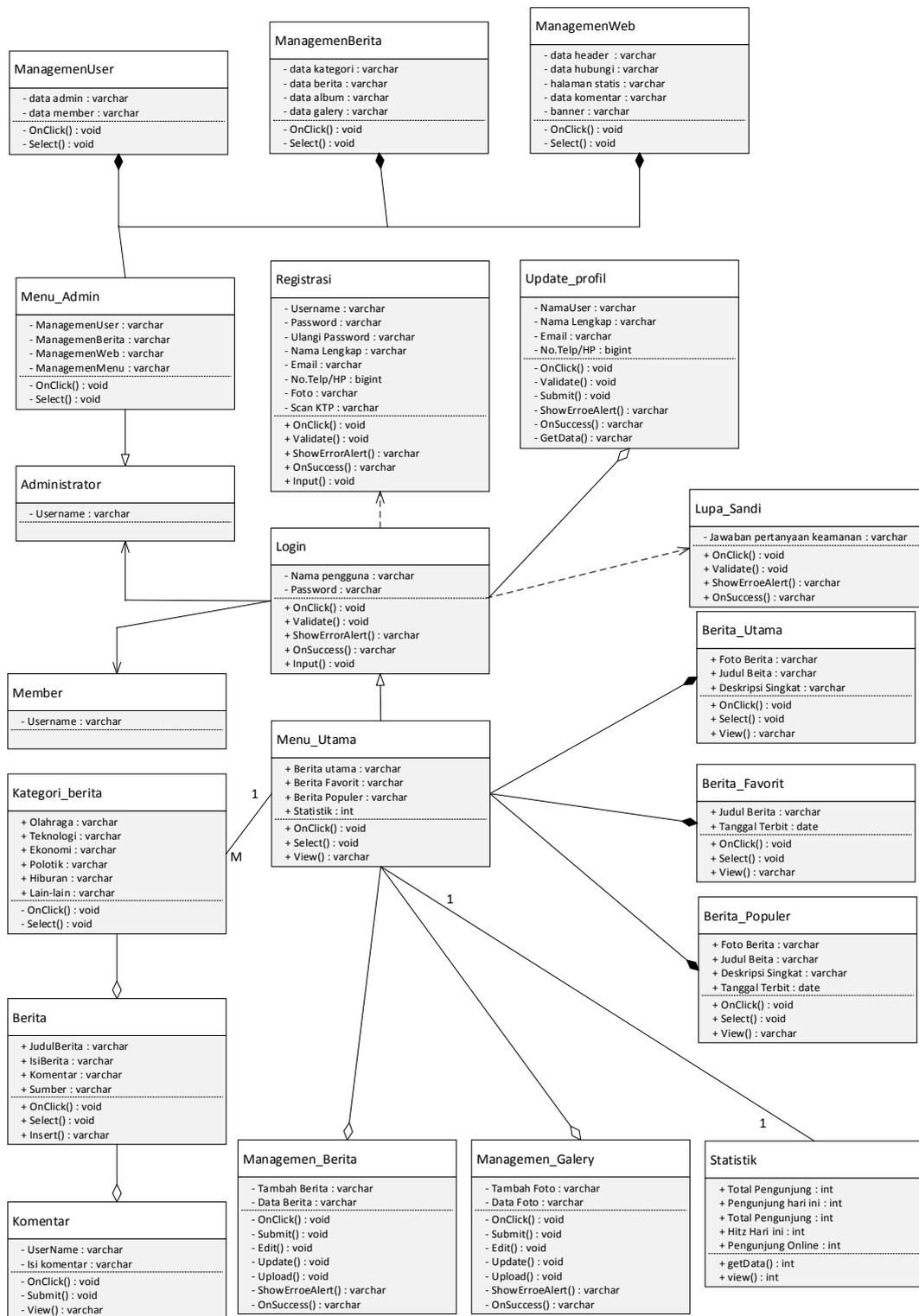
3.6.2 Class Diagram

Class diagram dibuat untuk menjelaskan hubungan antara kelas yang satu dengan kelas yang lainnya. Didalam 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.

Pada gambar 3.4 terlihat beberapa simbol dan bagian dari *class diagram*, simbol-simbol tersebut antara lain :

Tabel 3.1 Simbol dalam *class diagram*

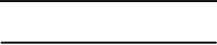
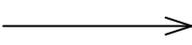
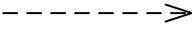
Gambar Simbol	Keterangan
+	<i>Public</i> , berarti atribut ataupun <i>operation</i> tersebut dapat diakses secara publik.
-	<i>Private</i> , berarti atribut ataupun <i>operation</i> tersebut tidak dapat diakses secara publik dan membutuhkan persyaratan lain agar atribut dan <i>operation</i> tersebut dapat digunakan.



Gambar 3. 4 Class Diagram Aplikasi

Pada gambar 3.4 dihubungkan dengan garis relasi yang memiliki makna tertentu, makna-makna tersebut disesuaikan dengan hubungan dan fungsi antar tabel yang berhubungan, relasi-relasi yang dipakai antara lain dapat dilihat pada tabel 3.2

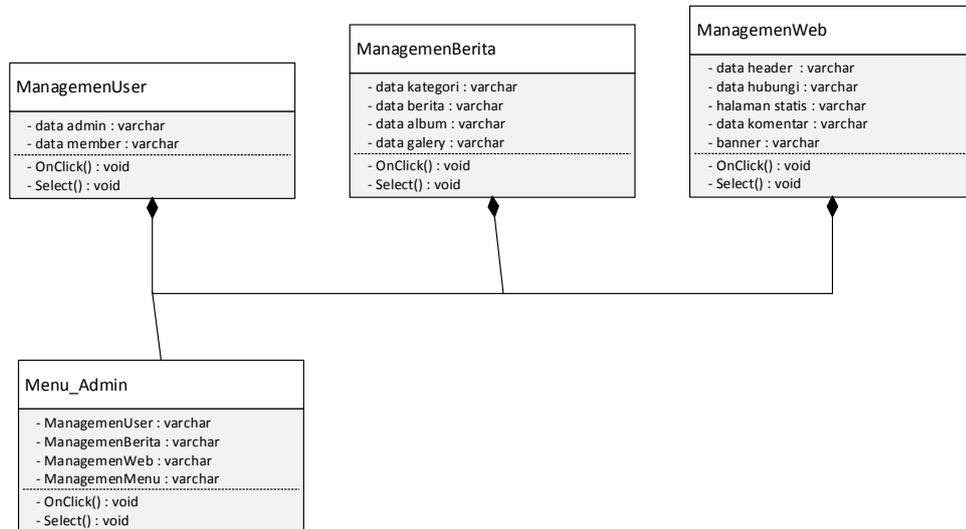
Tabel 3.2 Relasi yang digunakan dalam *class diagram*

Nama Relasi	Gambar	Keterangan
<i>Association</i>		Relasi antar kelas dengan makna umum. Biasanya disertai dengan <i>multiplicity</i> (keterangan banyak).
<i>Directed Association</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
<i>Agregation</i>		Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).
<i>Dependency</i>		Relasi antar kelas dengan makna kebergantungan antar kelas.
<i>Generalisation</i>		Relasi antar kelas dengan makna kebergantungan antar kelas.
<i>Compotition</i>		Relasi antar kelas dengan makna ada beberapa kelas yang merupakan bagian dari kelas utama.

Pada gambar 3.4 menjelaskan bahwa aplikasi berbasis *website* yang akan dibuat dibagi atas beberapa *class* yang saling berhubungan. Kelas-kelas yang saling berhubungan memiliki arti dan makna yang berbeda-beda antara hubungan yang satu dengan yang lain.

Kelas-kelas yang berelasi secara detail antara lain :

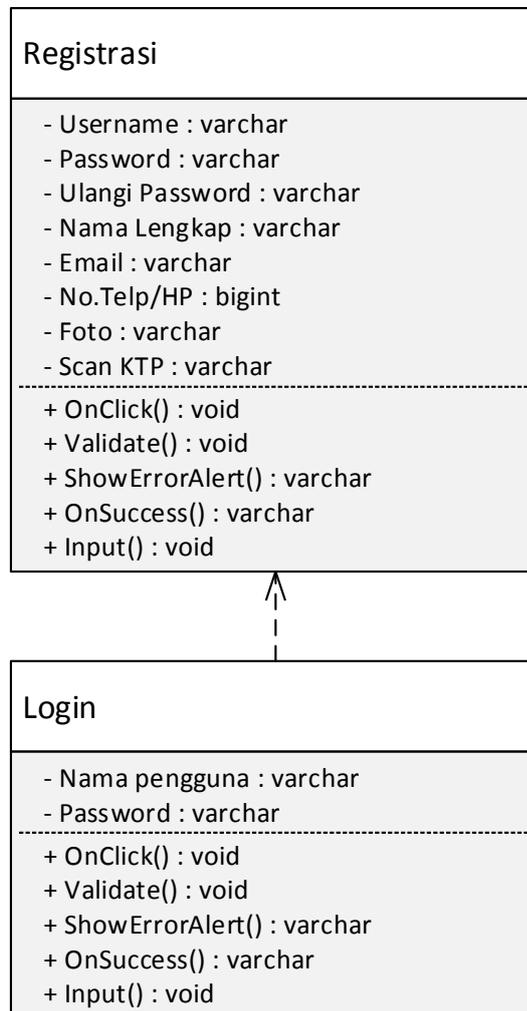
- a. Relasi antara *class* Menu_admin dengan *class* ManajemenUser, ManajemenBerita dan ManajemenWeb



Gambar 3. 5 Relasi antara class Menu_admin dengan class ManajemenUser, ManajemenBerita dan ManajemenWeb

Relasi antara *class* Menu_admin dengan *class* ManajemenUser, ManajemenBerita dan ManajemenWeb dihubungkan dengan relasi komposisi. Relasi tersebut berarti *class* ManajemenUser, ManajemenBerita dan ManajemenWeb merupakan komponen penyusun dari *class* Menu_Admin. Dengan relasi komposisi, tergambar sangat jelas bahwa *class* ManajemenUser, *class* ManajemenBerita ataupun *class* ManajemenWeb tidak dapat berdiri sendiri untuk dapat digunakan melainkan harus berasal dari *class* menu_admin terlebih dahulu.

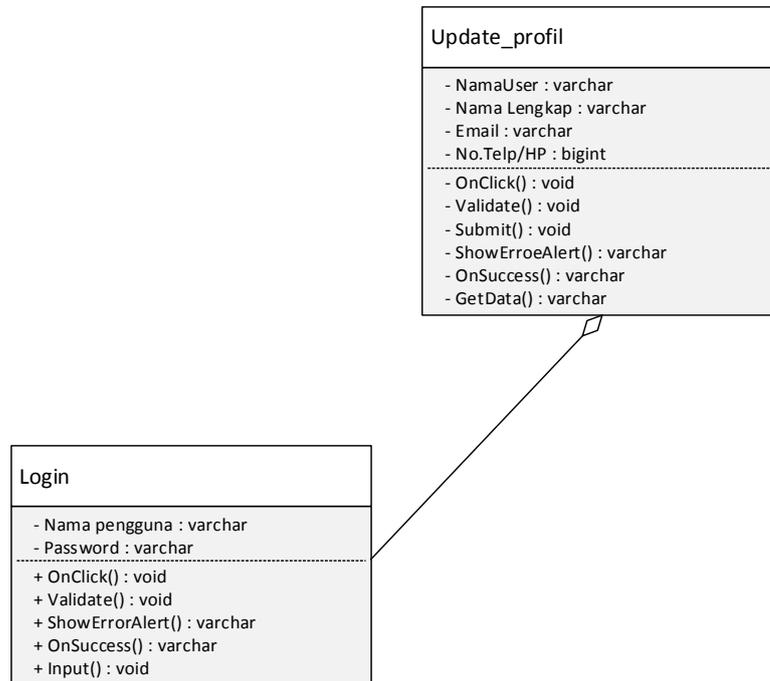
b. Relasi antara *class* Login dengan *class* registrasi



Gambar 3. 6 Relasi antara *class* Login dengan *class* registrasi

Relasi antara *class* Login dengan *class* registrasi dihubungkan dengan relasi *dependency* atau dalam bahasa indonesianya relasi dependensi. Relasi dependensi merupakan Relasi antar kelas dengan makna ketergantungan antar kelas. Jadi ketika pengguna akan melakukan *login*, maka pengguna tersebut harus terdaftar terlebih dahulu yaitu dengan mendaftarkan dirinya pada *class* registrasi untuk mendapatkan *username* dan *password* untuk *login*.

c. Relasi antara *class Login* dengan *class Update_profil*

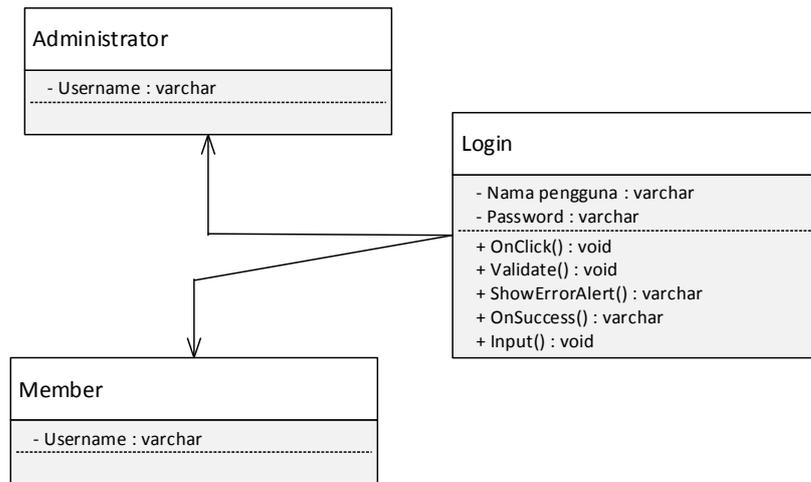


Gambar 3. 7 Relasi antara *class Login* dengan *class Update_profil*

Relasi antara *class Login* dengan *class Update_profil* dihubungkan dengan relasi *agregation* yaitu relasi antar kelas dengan makna semua bagian (*whole-part*). Ketika *member* ingin masuk kedalam *class update_profil*, maka terlebih dahulu harus melakukan *login*. Tanpa melakukan *login*, *member* tidak dapat melakukan *update_profil*.

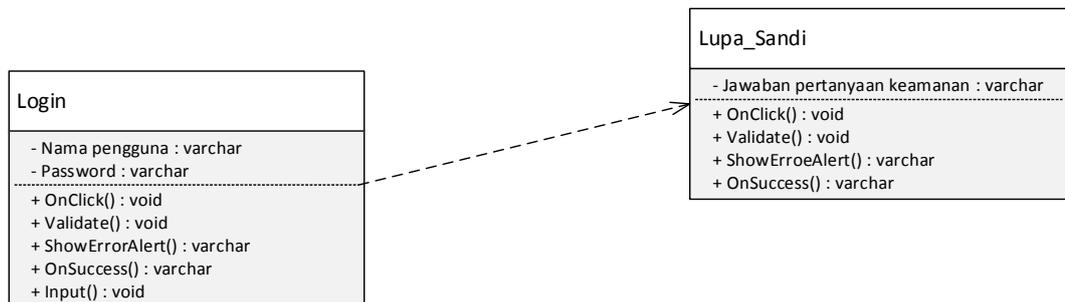
d. Relasi antara *class Login* dengan *class administrator* dan *member*

Relasi antara *class Login* dengan *class administrator* dan *member* dihubungkan dengan *Directed Association* yaitu relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. *Class Login* digunakan oleh pengguna *member* dan *administrator*.



Gambar 3. 8 Relasi antara *class Login* dengan *class administrator* dan *member*

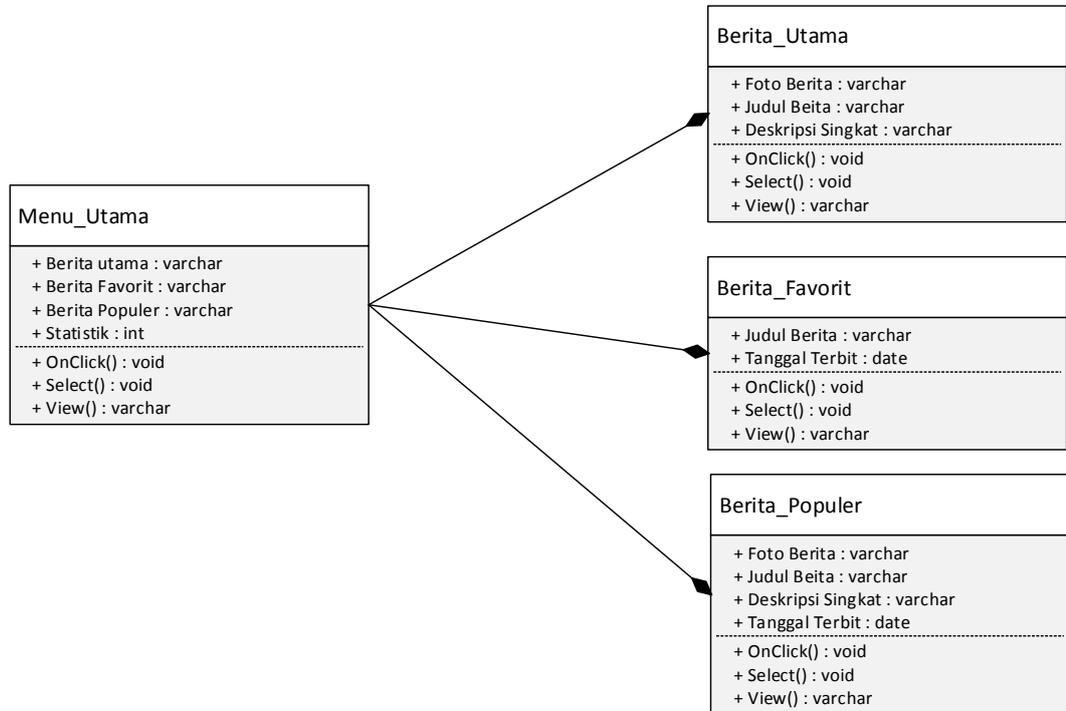
e. Relasi antara *class Login* dengan *class lupa_sandi*



Gambar 3. 9 Relasi antara *class Login* dengan *class lupa_sandi*

Relasi antara *class Login* dengan *class lupa_sandi* dihubungkan dengan relasi *dependensi*. *Class lupa_sandi* hanya akan dapat dilakukan dan digunakan setelah pengguna masuk kedalam *class login*.

- f. Relasi antara *class* menu_utama dengan *class* berita_utama, berita favorit, dan berita populer

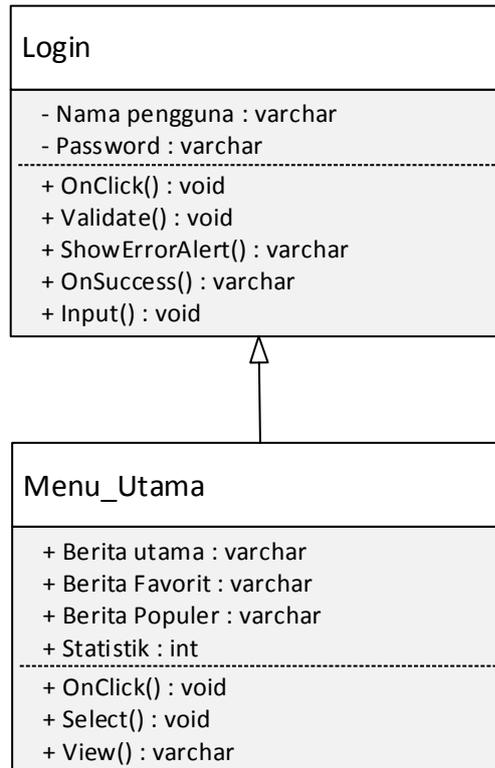


Gambar 3. 10 Relasi antara class menu_utama dengan class berita_utama, berita favorit, dan berita populer

Relasi antara *class* menu_utama dengan *class* berita_utama, berita favorit, dan berita populer dihubungkan dengan relasi komposisi. Artinya ada tiga *class* penyusun *class* menu_utama. *Class* penyusun tersebut juga tidak dapat digunakan jika tidak difungsikannya *class* menu_utama, begitu juga sebaliknya.

- g. Relasi antara *class* menu_utama dengan *class* login

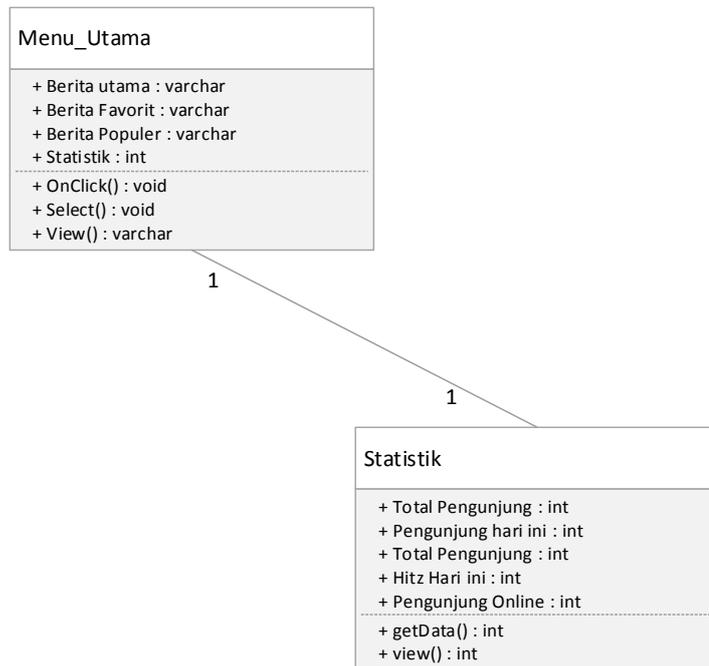
Relasi antara *class* menu_utama dengan *class* login terhubung dengan relasi generalisasi. Menu utama akan terbuka ketika *member* telah melakukan *login* dan sukses jika posisinya adalah pengguna ingin masuk langsung sebagai *member* (bukan *user* biasa).



Gambar 3. 11 Relasi antara *class* menu_utama dengan *class* login

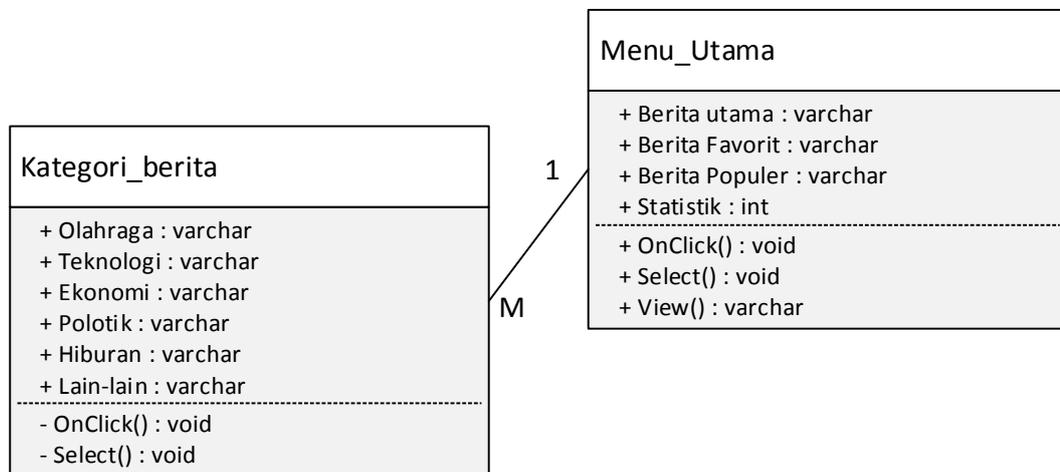
h. Relasi antara *class* menu_utama dengan *class* statistik

Relasi antara *class* menu_utama dengan *class* statistik dihubungkan dengan relasi asosiasi *one to one*. Relasi tersebut bermakna satu menu utama memiliki satu statistik dan satu statistik tersebut diperuntukkan untuk satu menu utama.



Gambar 3. 12 Relasi antara *class* menu_utama dengan *class* statistik

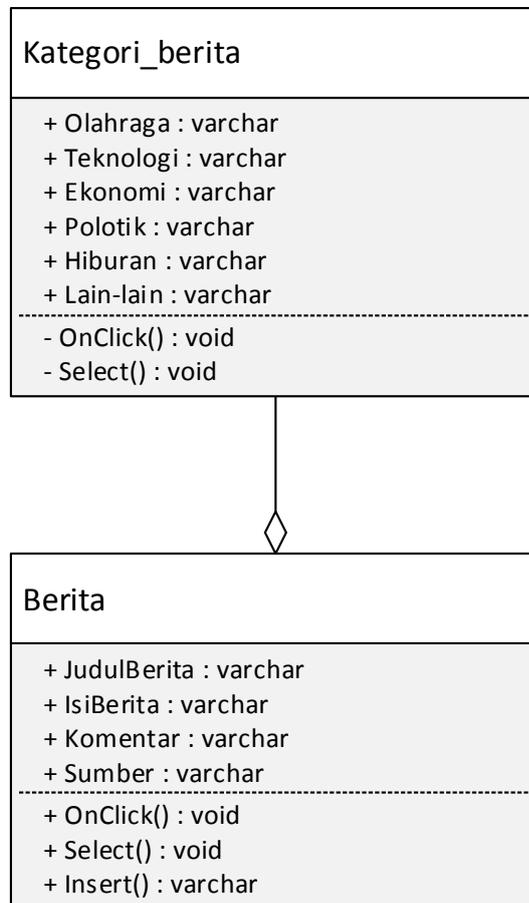
i. Relasi antara *class* menu_utama dengan *class* kategori_berita



Gambar 3. 13 Relasi antara *class* menu_utama dengan *class* kategori_berita

Relasi antara *class* menu_utama dengan *class* kategori_berita dihubungkan dengan relasi asosiasi *one to many* yang berarti satu menu utama memiliki banyak kategori_berita ataupun banyak kategori_berita diperuntukkan untuk satu menu_utama.

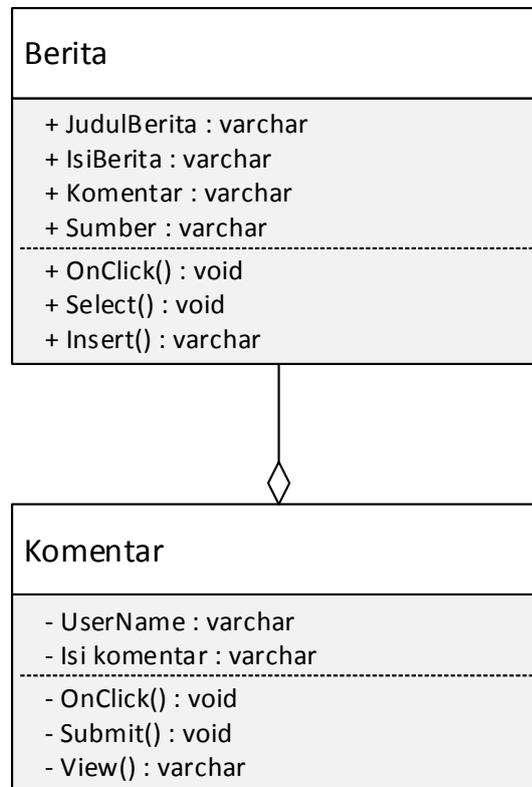
j. Relasi antara *class* kategori_berita dengan *class* berita



Gambar 3. 14 Relasi antara *class* kategori_berita dengan *class* berita

Relasi antara *class* kategori_berita dengan *class* berita dihubungkan dengan relasi agregasi. *Class* berita merupakan bagian dari *class* kategori_berita, hal itu berarti semua berita yang ada dimiliki oleh kategori_berita terkait.

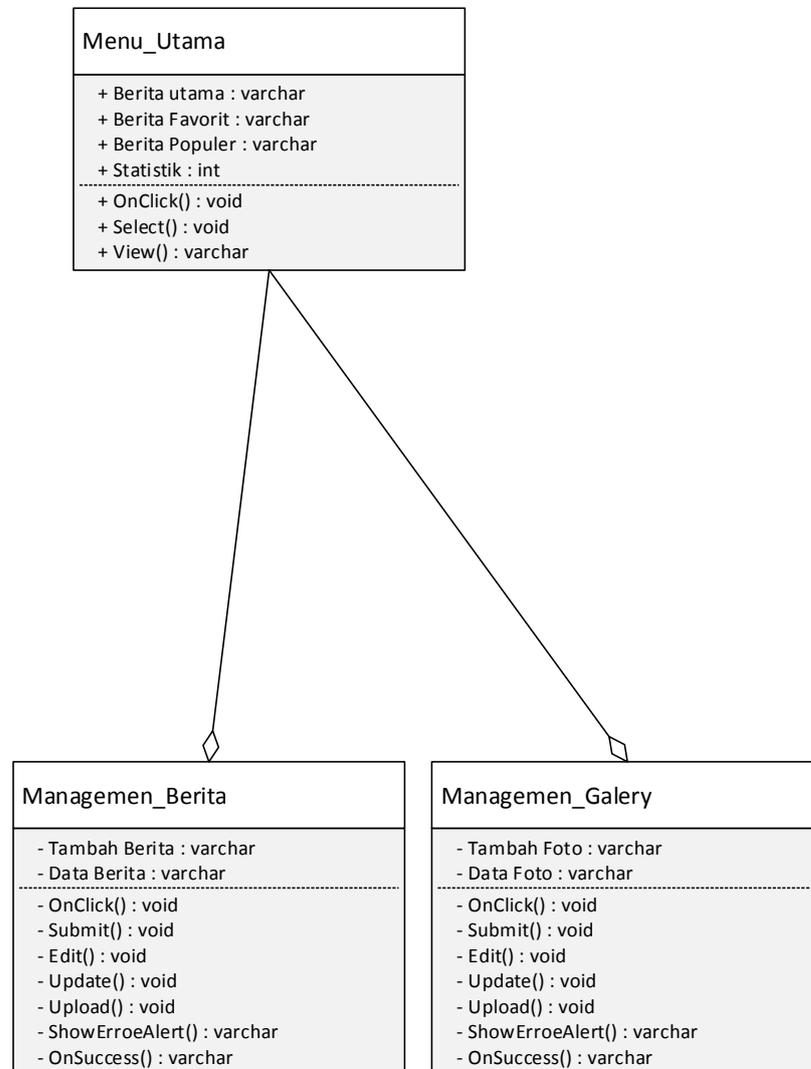
k. Relasi antara *class* berita dengan *class* komentar



Gambar 3. 15 Relasi antara *class* berita dengan *class* komentar

Relasi antara *class* berita dengan *class* komentar terhubung dengan relasi agregasi. Dengan relasi agregasi, dimaksudkan bahwa *class* komentar merupakan bagian dari *class* berita. Setiap berita memiliki bagian komentar masing-masing yang berbeda di setiap berita.

1. Relasi antara *class* menu_utama dengan *class* managemen_berita dan managemen_galery



Gambar 3. 16 Relasi antara class menu_utama dengan *class* managemen_berita dan managemen_galery

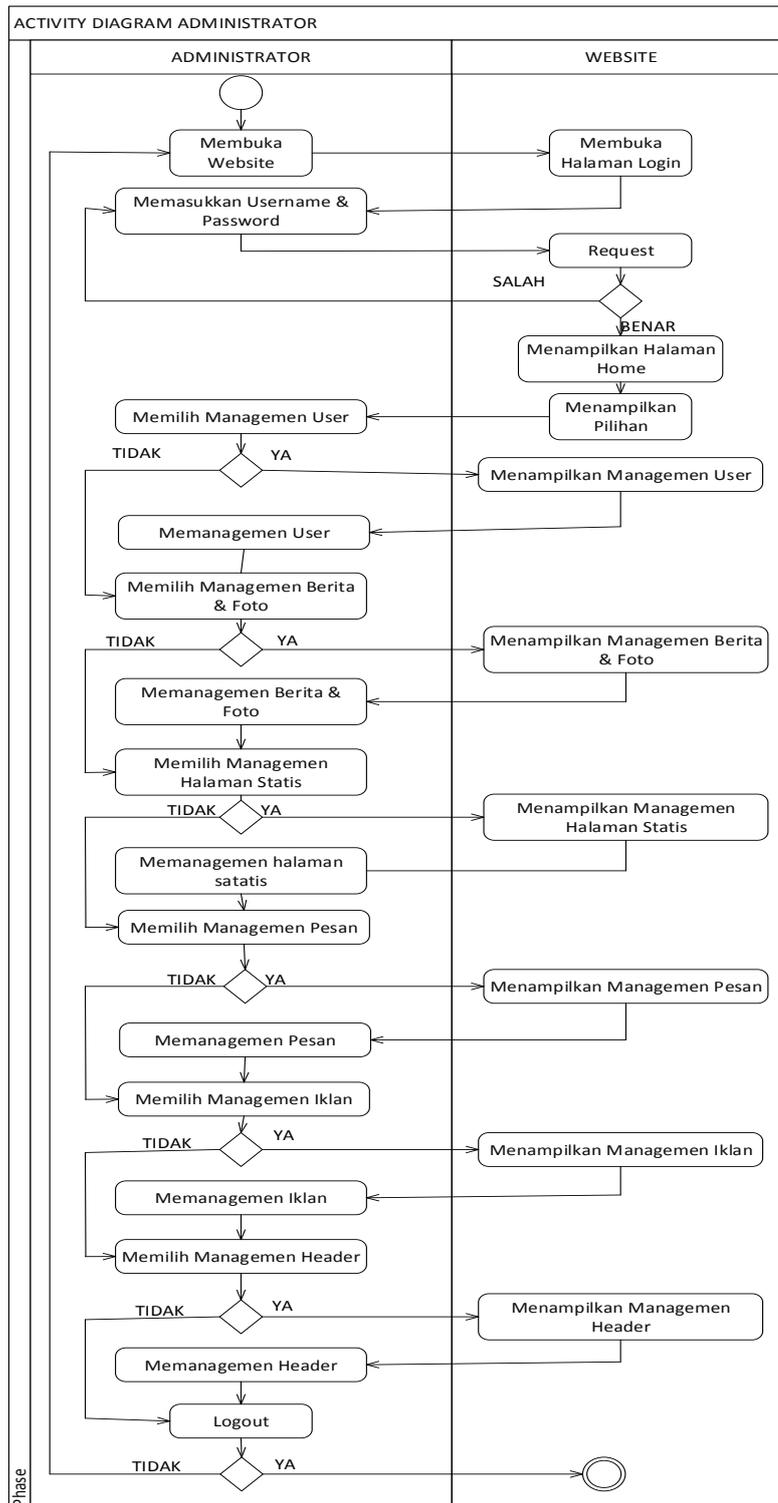
Relasi antara *class* menu_utama dengan *class* managemen_berita dan managemen_galery dihubungkan dengan relasi agregasi. *Class* menu_utama merupakan *main class* dari *sub class* bernama *class* managemen_berita dan managemen_galery.

3.6.3 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 :

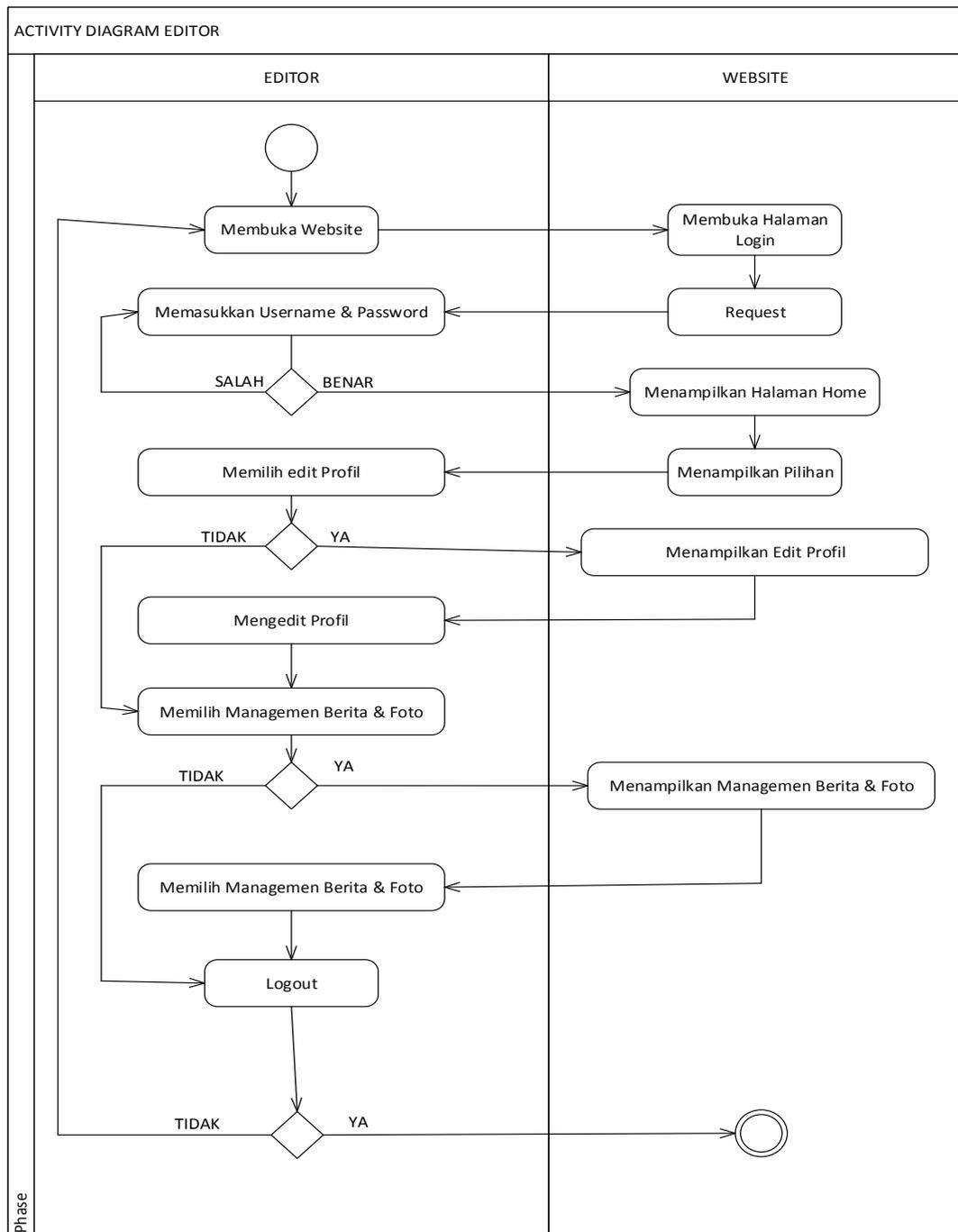
a. *Activity Diagram* Administrator

Activity Diagram 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* yang telah ditampilkan secara otomatis oleh *website*. Jika *username* dan *password* yang dimasukkan oleh administrator adalah benar, maka administrator akan masuk kedalam menu *home* atau halaman utama yang terdapat banyak sekali 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 langsung kembali ke halaman *login*. Administrator akan diberikan beberapa pilihan yang dapat dipilih ataupun tidak. Pilihan yang dipilih oleh administrator secara langsung akan direspon oleh *website* untuk menampilkan halaman terkait dengan data *terupdate*.



Gambar 3. 17 Activity Diagram Administrator

b. *Activity Diagram Editor*

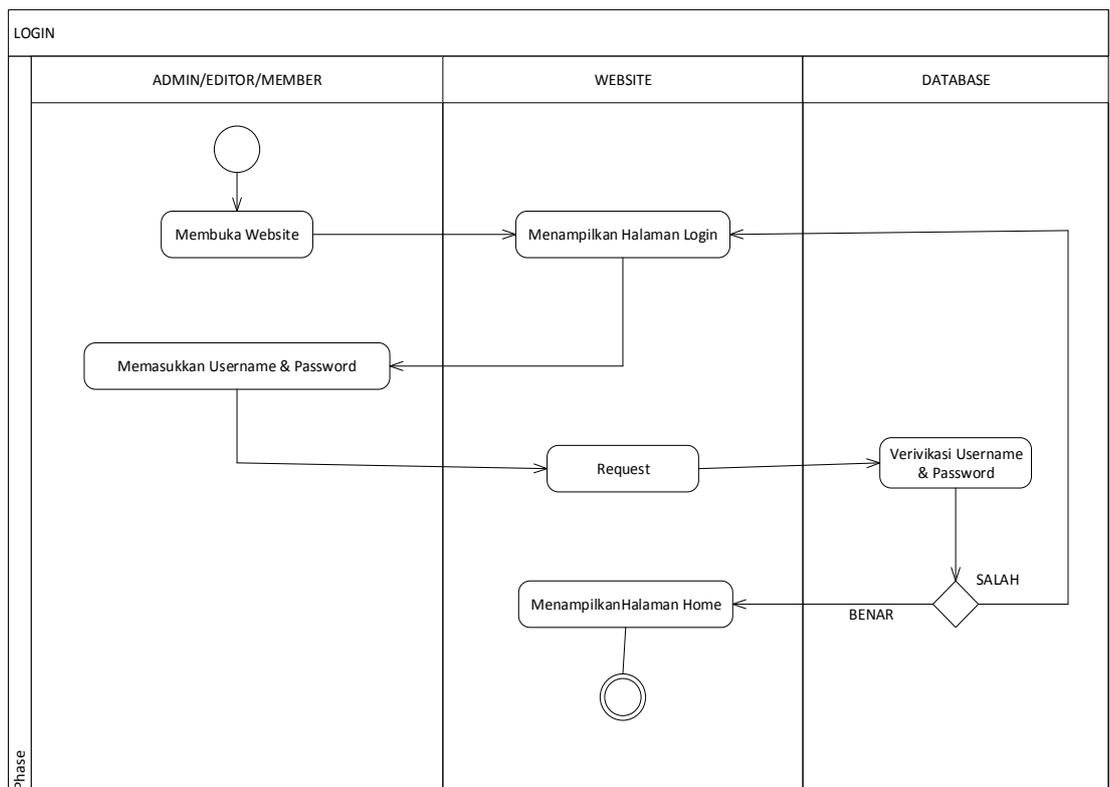


Gambar 3. 18 *Activity Diagram Editor*

Sama seperti gambar 3.17 *Activity Diagram Administrator*, *Activity Diagram Editor* juga harus memasukkan terlebih dahulu *username* dan *password* agar

website dapat merespon dengan tampilan menu *home* yang berisi banyak pilihan. Jika dibandingkan antara *Activity Diagram* administrator dengan *Activity Diagram* editor terlihat jelas bahwa kegiatan dan interaksi yang dapat dilakukan oleh editor lebih sedikit dibandingkan dengan administrator. Hal tersebut terlihat dari jumlah pilihan yang dapat dipilih oleh aktor editor.

c. *Activity Diagram Login*



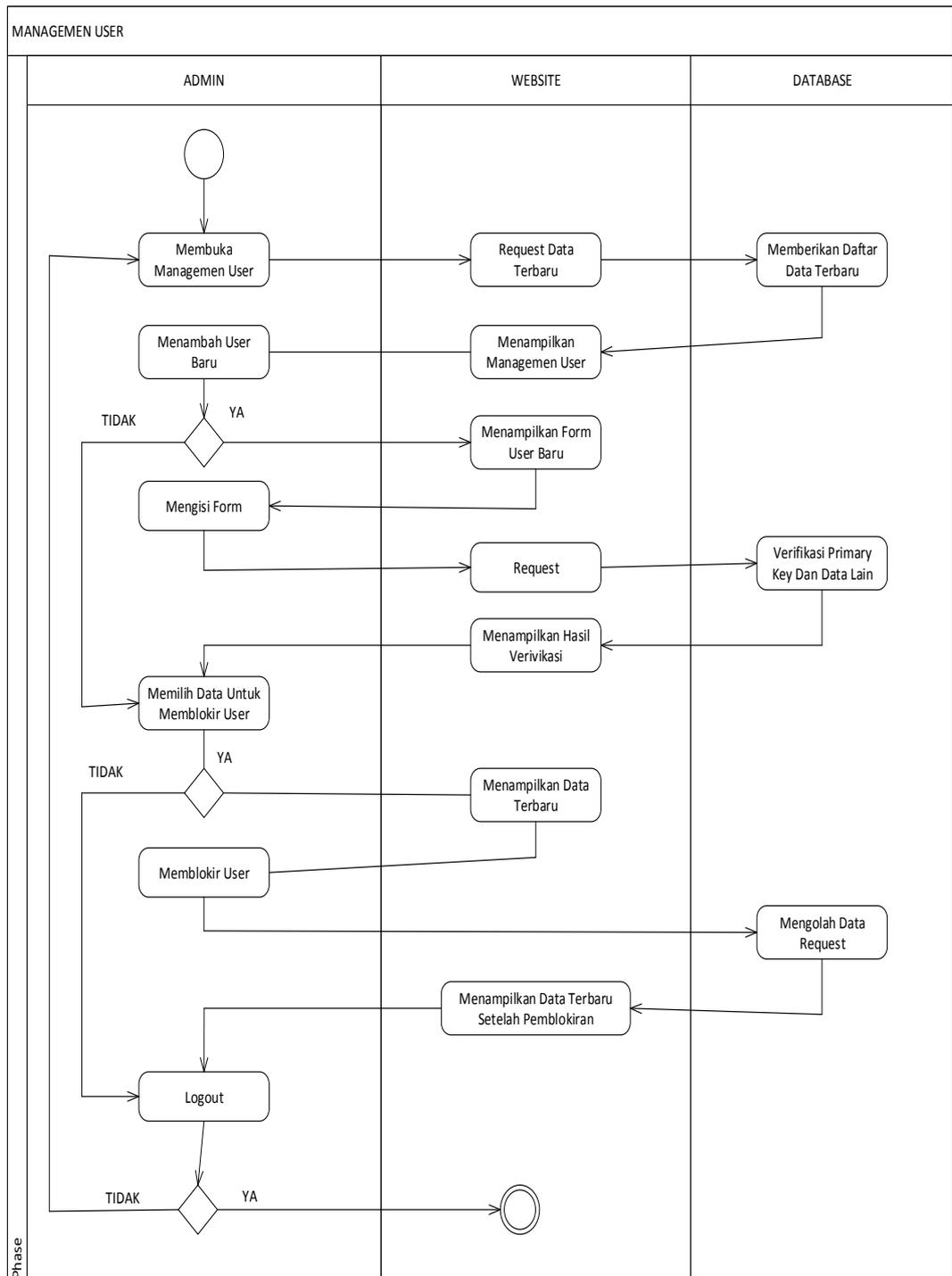
Gambar 3. 19 *Activity Diagram Login*

Agar lebih mengerucut dan detail serta lebih mudah dipahami, *Activity Diagram* dipecah-pecah kembali menjadi *Activity Diagram* lain yang sebenarnya juga sudah tercantum pada *Activity Diagram* administrator dan *Activity Diagram* editor. Dalam *Activity Diagram login*, ada tambahan *swimlane database* yang berfungsi untuk lebih memperjelas interaksi antara *swimlane* pengguna (administrator, editor dan *member*) dan *swimlane website*. Dalam *Activity Diagram* terlihat jelas fungsi dari *swimlane database* yang

berfungsi untuk memverifikasi *username* dan *password* yang dimasukkan oleh pengguna. Sedangkan fungsi dari *website* hanya menampilkan *result* dari verifikasi *database*.

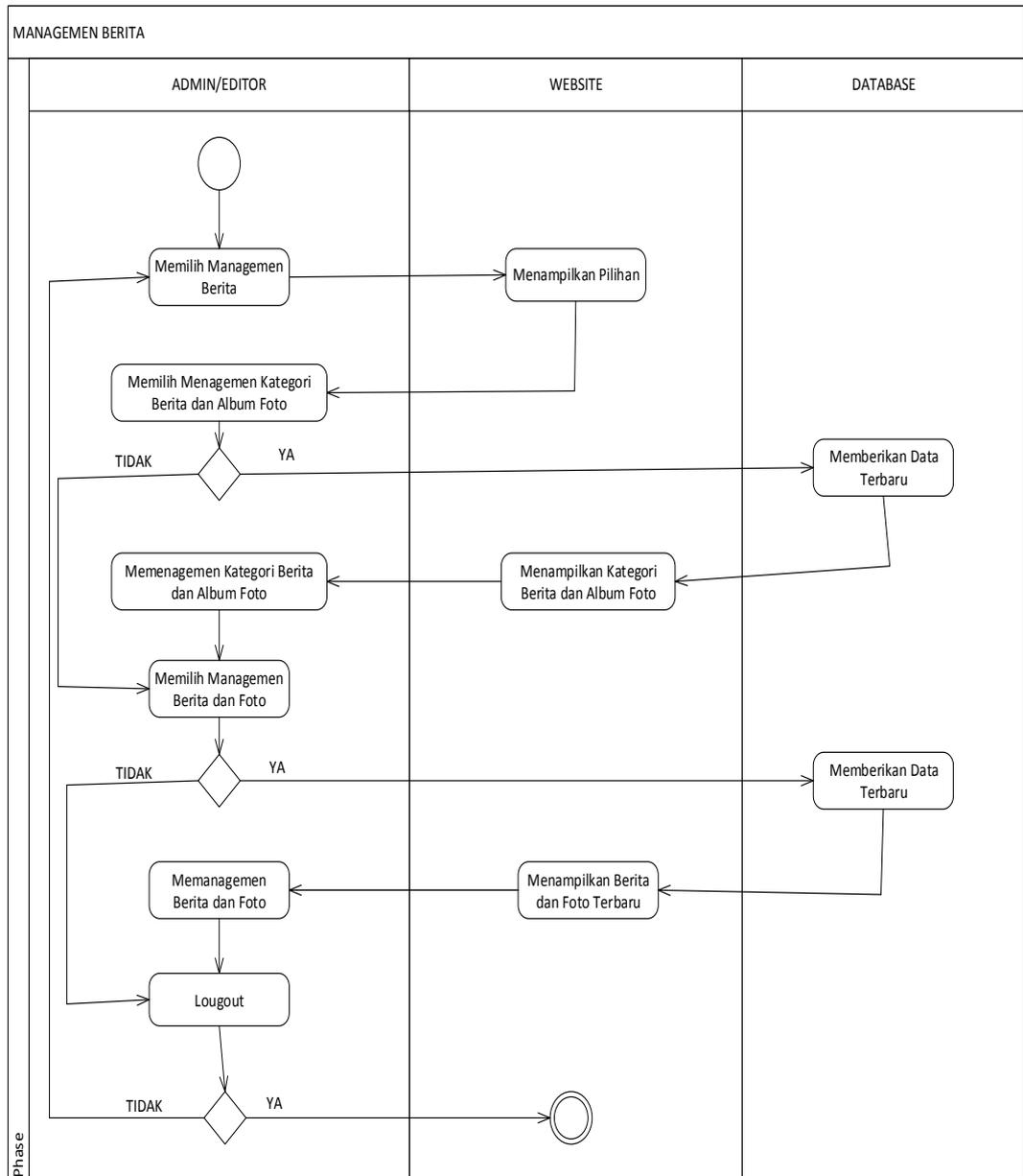
d. *Activity Diagram Manajemen User*

Activity Diagram Manajemen User merupakan *Activity Diagram* penjelas ketika aktor administrator memilih manajemen *user*. Didalam manajemen *user* terdapat beberapa aktifitas lagi yang tergambar pada gambar 3.20. *Activity Diagram Manajemen User* lebih menekankan pada detail isi dan kegiatan yang dapat dilakukan oleh administrator dalam pilihan manajemen *user* dan menjelaskan alur kegiatan beserta hubungan antara pengguna, *website* dan *database*.



Gambar 3. 20 Activity Diagram Managemen User

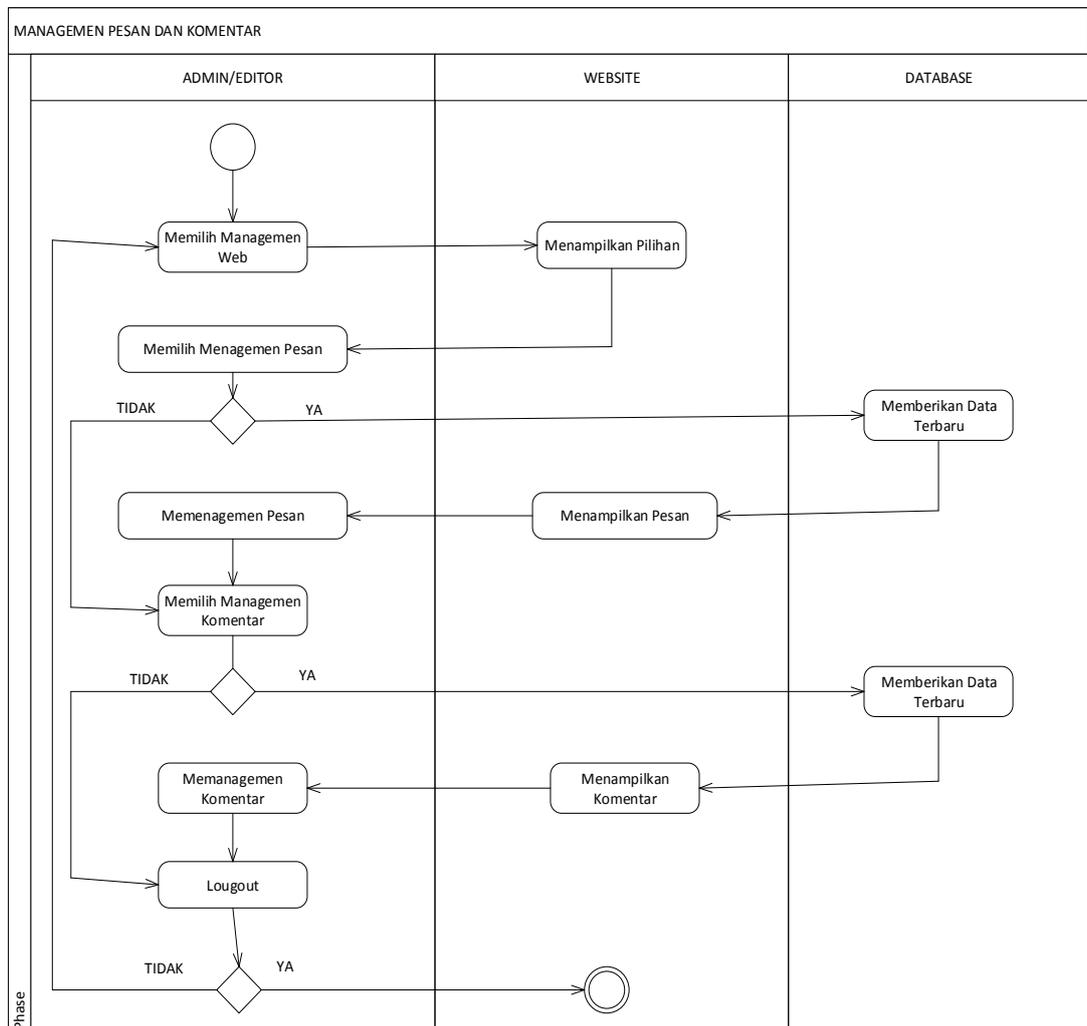
e. *Activity Diagram Managemen Berita*



Gambar 3. 21 *Activity Diagram Managemen Berita*

Sama seperti *Activity Diagram* Manajemen *User*, *Activity Diagram* Manajemen Berita juga menjelaskan lebih rinci terkait manajemen berita. Hubungan antara pengguna, *website* dan *database* akan dapat terlihat secara terperinci. Pilihan menu-menu dan kegiatan yang dilakukan oleh administrator juga beraneka ragam sesuai dengan keinginan dan kebutuhan. Terdapat beberapa *decision* pada *Activity Diagram* Manajemen Berita yang berfungsi sebagai penegasan keputusan untuk memilih pilihan terkait atau tidak. Jika iya, maka *database* akan mengambil data terbaru dan ditampilkan oleh *website* untuk dapat dikelola oleh pengguna. Jika tidak, pengguna dapat memilih pilihan yang lain atau sama sekali tidak memilih (dalam konteks *Activity Diagram* Manajemen Berita ini pilihan terakhir dari pengguna yang ditampilkan oleh sistem adalah *logout*).

f. *Activity Diagram Manajemen Web*

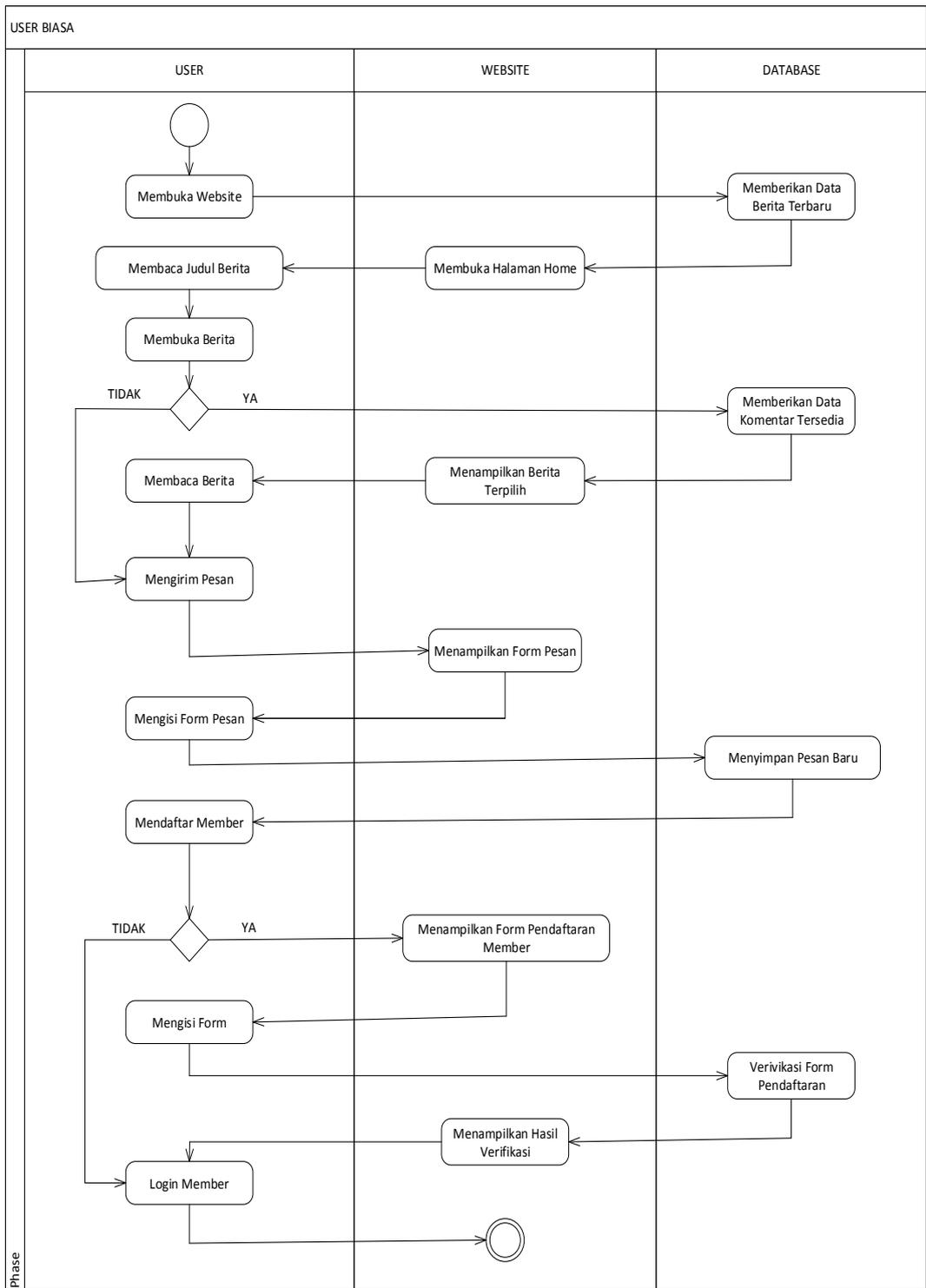


Gambar 3. 22 *Activity Diagram Manajemen Web*

Activity Diagram Manajemen Web menggambarkan kegiatan terkait administrator dalam membalas pesan dan memanagermen komentar-komentar yang tidak mengandung unsur-unsur yang tidak baik menurut administrator.

g. *Activity Diagram User*

Pada gambar 3.23 tentang *Activity Diagram User*, *Activity Diagram User* yang dimaksud terjadi pada *website* utama. Dalam *website* tersebut terlihat pengguna yang berperan sebagai *user* biasa dapat melakukan beberapa hal yang tersedia didalam *website* tersebut.



Gambar 3. 23 Activity Diagram User

Dalam *website* utama, pengguna yang mendapatkan fitur paling lengkap adalah *member*. *Member* merupakan kondisi dimana pengguna yang sebelumnya berperan sebagai *user* biasa melakukan *login* pada halaman *website* jika telah mendaftar sebelumnya. *Activity Diagram Member* diatas hanya menampilkan pilihan kegiatan tambahan yang tidak dapat dilakukan *user* biasa namun dapat dilakukan oleh *member*.

3.6.4 Perancangan Basis Data

Perancangan *database* pada aplikasi portal berita dengan komunikasi dua arah berbasis *website* ditentukan berdasarkan kebutuhan untuk mendukung jalannya aplikasi. Perancangan *database* dilakukan untuk mempermudah dalam pembuatan *database* dan nantinya *database* yang dibuat dapat efisien dari sisi penyimpanan serta mempermudah pengguna untuk melakukan pengeditan dan pengambilan data.

Dalam perancangan basis data diperlukan pembuatan desain tabel yang disertai dengan strukturnya. Maka dari itu untuk merancang aplikasi portal berita dengan komunikasi dua arah berbasis *website*, dibutuhkan tabel – tabel yang diperlukan untuk merancang *database*.

Tabel-tabel tersebut antara lain :

1. Tabel album

Tabel 3. 3 Struktur tabel album

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_album	PK	NN	Int	5
2	Jdl_album		NN	Varchar	100
3	Album_seo		NN	Varchar	100
4	Gbr_album		NN	Varchar	100
5	Aktif		NN	Enum	

Tabel album berfungsi sebagai pembuat album untuk *gallery* atau foto agar tertata dengan rapi. *Id_album* merupakan *primary key* yang diatur *auto increment* yaitu penambahan secara otomatis ketika album ditambahkan. Untuk kolom bernama aktif pada tabel album memiliki *type* data enum dengan isi *string* ‘Y’ untuk diaktifkan dan ‘N’ untuk tidak diaktifkan. Untuk *default* data yang disediakan adalah ‘Y’, berarti data *default* dari kolom aktif diaktifkan. Enum merupakan tipe data yang khusus untuk kolom yang nilai datanya sudah ditentukan sebelumnya.

2. Tabel banner

Tabel 3. 4 Struktur tabel banner

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_banner	PK	NN	Int	5
2	Jdl_banner		NN	Varchar	100
3	url		NN	Varchar	100
4	Gambar		NN	Varchar	100
5	Tgl_posting		NN	Date	

Tabel banner merupakan tabel yang berisi iklan dari *website* portal berita. *Primary key* pada tabel *banner* di *set* untuk *auto increment*. Pada kolom *url* berisi *link website* yang dapat dikunjungi ketika banner tersebut di *klik*.

3. Tabel berita

Primary key dibuat *auto increment* dan terpasang secara *default* ketika berita ditambahkan. Terdapat dua kolom *foreign key* yaitu kolom *id_kategori* dari tabel kategori dan *username* dari tabel *user*. Pada kolom *headline* terdapat *type data enum* dengan isi *string* ‘Y’ untuk diaktifkan dan ‘N’ untuk tidak diaktifkan. Jika diaktifkan, maka berita tersebut menjadi *headline*.

Tabel 3. 5 Struktur tabel berita

No	Field Name	Key	Null	Data Type	Max. Length
1	Id_berita	PK	NN	Int	5
2	Id_katagori	FK	NN	Int	5
3	Username	FK	NN	Varchar	100
4	Judul		NN	Varchar	100
5	Judul_seo		NN	Varchar	10
6	Headline		NN	Enum	
7	Isi_berita		NN	Text	
8	Hari		NN	Varchar	20
9	Tanggal		NN	Date	
10	Jam		NN	Time	
11	Gambar		NN	Varchar	100
12	Status		NN	Varchar	15
13	Dibaca		NN	Int	5
14	Nama_file		NN	Varchar	100

4. Tabel *gallery*

Tabel 3. 6 Struktur tabel *gallery*

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_gallery	PK	NN	Int	5
2	Id_album	FK	NN	Int	5
3	Jdl_gallery		NN	Varchar	100
4	Gallery_seo		NN	Varchar	100
5	Keterangan		NN	Text	
6	Status		NN	Varchar	15
7	Gbr_gallery		NN	Varchar	100
8	username	FK	NN	Varchar	100

Primary key pada tabel *gallery* adalah *id_gallery* dan terdapat dua kolom *foreign key* bernama *id_album* yang berasal dari tabel *album* dan *username* yang berasal dari tabel *users*. Pada tabel 3.6 terdapat kolom keterangan. Kolom keterangan memiliki tipe data *text*, yang berarti data yang dimasukkan tidak dibatasi jumlahnya dan berbentuk teks.

5. Tabel halamanstatis

Tabel 3. 7 Struktur tabel halamanstatis

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_halaman	PK	NN	Int	5
2	Judul		NN	Varchar	100
3	Isi_halaman		NN	Text	
4	Tgl_posting		NN	Date	
5	Gambar		NN	Varchar	100

Tabel halaman statis berfungsi untuk mengisi data halaman statis yang sewaktu-waktu dapat diubah. Kolom isi halaman diatur dengan tipe data *text* agar isi dan jumlah karakter pada halaman statis tidak dibatasi.

6. Tabel *header*

Tabel 3. 8 Struktur tabel *header*

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_header	PK	NN	Int	5
2	Judul		NN	Varchar	100
3	Gambar		NN	Varchar	100
4	Tgl_posting		NN	Date	

Tabel *header* merupakan tabel tempat mengatur gambar *header* pada halaman *website*.

7. Tabel hubungi

Tabel 3. 9 Struktur tabel hubungi

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_hubungi	PK	NN	Int	5
2	Nama		NN	Varchar	50
3	Email		NN	Varchar	100
4	Subjek		NN	Varchar	100
5	Pesan		NN	Text	
6	Tanggal		NN	Date	

Tabel hubungi merupakan wadah dari *user* biasa dan *member* yang ingin mengajukan pertanyaan, menyampaikan pendapat dan memberikan masukan terhadap pengelola *website*. Terdapat kolom *email* pada salah satu kolom di tabel hubungi, kolom *email* tersebut berfungsi untuk alamat surat ketika administrator akan menjawab surat atau pertanyaan yang masuk.

8. Tabel katajelek

Tabel 3. 10 Struktur tabel katajelek

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_jelek	PK	NN	Int	5
2	Kata		NN	Varchar	60
3	Ganti		NN	Varchar	60

Tabel katajelek merupakan tabel yang berfungsi sebagai *back up* dari tugas administrator dalam memfilter komentar yang dibuat oleh *member*.

9. Tabel kategori

Tabel 3. 11 Struktur tabel kategori

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_kategori	PK	NN	Int	5
2	Nama_kategori		NN	Varchar	50
3	Kategori_seo		NN	Varchar	100
4	Aktif		NN	Enum	

Sama halnya seperti tabel album yang berkaitan dengan tabel *gallery* untuk mengatur jenis foto yang *diupload* agar lebih rapi dan tertata, perbedaannya adalah tabel kategori mengatur jenis berita bukan foto.

10. Tabel komentar

Tabel 3. 12 Struktur tabel komentar

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Id_komentar	PK	NN	Int	5
2	Id_berita	FK	NN	Int	5
3	Username	FK	NN	Varchar	100
4	Isi_komentar		NN	Text	
5	Tanggal		NN	Date	
6	Jam_komentar		NN	Time	
7	Aktif		NN	Enum	

Primary key pada tabel komentar adalah *id_komentar* yang di *set auto increment*. Terdapat dua *foreign key* yaitu *id_berita* dari tabel berita dan *username* dari tabel *users*. Kolom *id_berita* merupakan penjelas bahwa komentar tersebut berada pada satu berita tertentu dan kolom *username* digunakan untuk mendapatkan informasi *member* mana yang memberikan

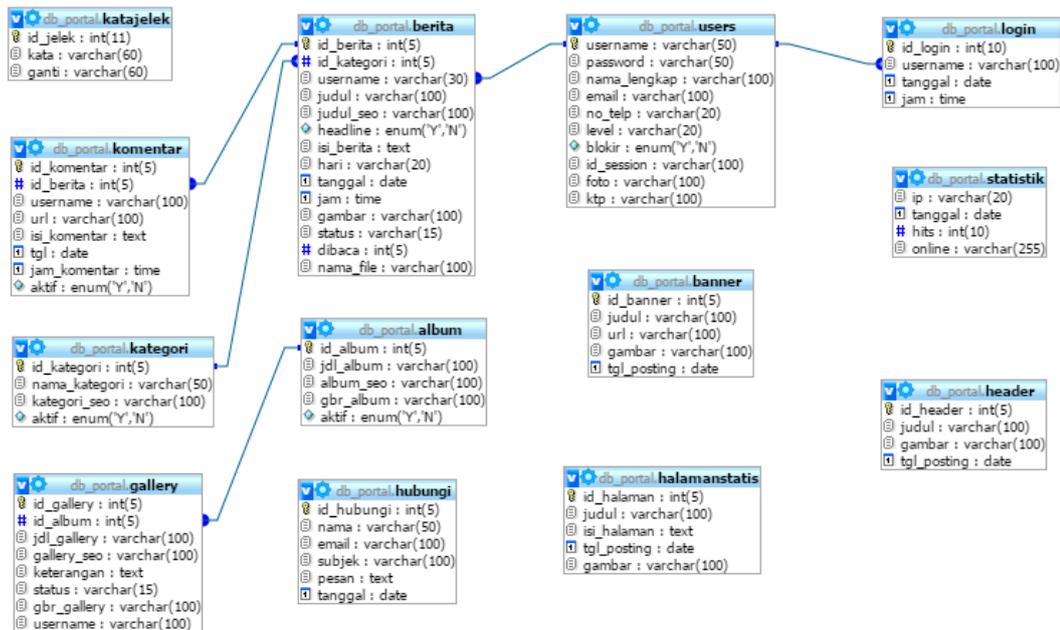
Ketika pengguna akan mendaftarkan dirinya sebagai *member*, data pada tabel *users* inilah yang harus dilengkapi kecuali tabel blokir. Tabel blokir memiliki *type* data *enum* dengan isi *string* ‘Y’ untuk diaktifkan dan ‘N’ untuk tidak diaktifkannya akun. Secara *default*, akun akan langsung aktif ketika pengguna berhasil mendaftarkan dirinya. Kolom blokir hanya bisa diakses oleh administrator. Pada tabel *users* terdapat kolom *password* yang datanya telah di *set encryption* berarti admin tidak dapat melihat isi *password* dari *member* ataupun administrator lain.

Tabel 3. 15 Struktur tabel *users*

No	Field Name	Key Type	Null	Data Type	Max. Length
1	Username	PK	NN	Varchar	5
2	Password		NN	Varchar	100
3	Nama_lengkap		NN	Varchar	100
4	Email		NN	Varchar	100
5	No_tlpn		NN	Varchar	20
6	Level		NN	Varchar	20
7	Blokir		NN	Enum	
8	Id_session		NN	Varchar	100
9	Foto		NN	Varchar	100
10	Ktp		NN	Varchar	100

3.6.5 Relasi Antar Tabel Pada Database

Dalam pembuatan basis data ada beberapa tabel yang saling berelasi dan ada beberapa tabel yang tidak memiliki relasi. Relasi antar tabel akan dijelaskan lebih detail menggunakan gambar relasi tabel basis data pada gambar 3.25.



Gambar 3. 25 Relasi Antar Tabel

Tabel-tabel yang berelasi terlihat dengan adanya simbol penghubung antara tabel satu dengan tabel lainnya. *Primary key* pada pada tabel tertentu dapat menjadi *foreign key* pada tabel yang berelasi dengan tabel tersebut. Relasi tersebut berarti adanya kaitan antara tabel yang satu dengan tabel yang lain. Dengan kata lain, tabel-tabel yang berelasi tidak dapat berdiri sendiri. Pada gambar 3.25 terlihat *main table* terdapat pada tabel *users*, tabel *galeri* dan tabel kategori. Ketiga tabel tersebut menjadi patokan untuk tabel berikutnya yang berelasi dengan salah satu dari ketiga tabel tersebut.

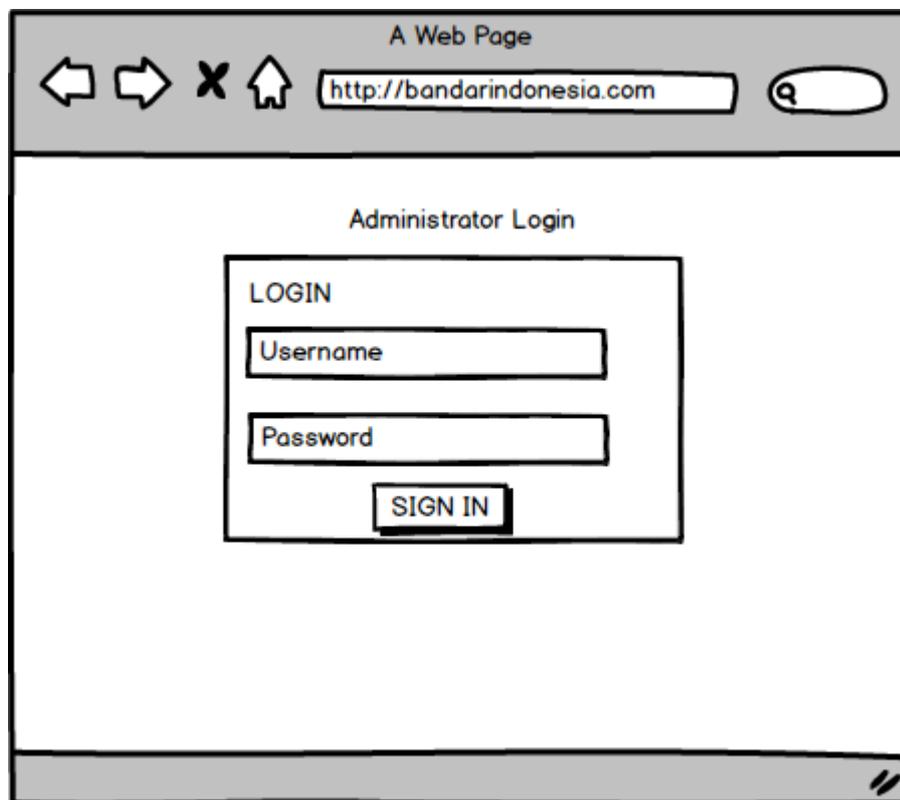
3.6.6 Perancangan Tampilan Antar Muka

Perancangan tampilan antar muka (*interface*) aplikasi portal berita berbasis *website* akan dibuat *simple* agar pengguna mudah mengerti dan nyaman digunakan (*user friendly*). Perancangan *interface* ini adalah gambaran secara

umum mengenai tampilan aplikasi. Untuk aplikasi *website* portal berita yang akan dibuat memiliki dua *website* yaitu *website* pengelola dan *website* utama.

Website pengelola hanya dapat dibuka oleh administrator dan editor. Selayaknya tugas pengelola, isi *website* pengelola semuanya berkaitan dengan pengelolaan *website*. *Website* pengelola dibagi kembali menjadi dua yaitu aktor administrator dan aktor editor. Perancangan *interface* pada *website* pengelola khusus untuk aktor administrator adalah sebagai berikut :

a. Rancangan halaman *login*



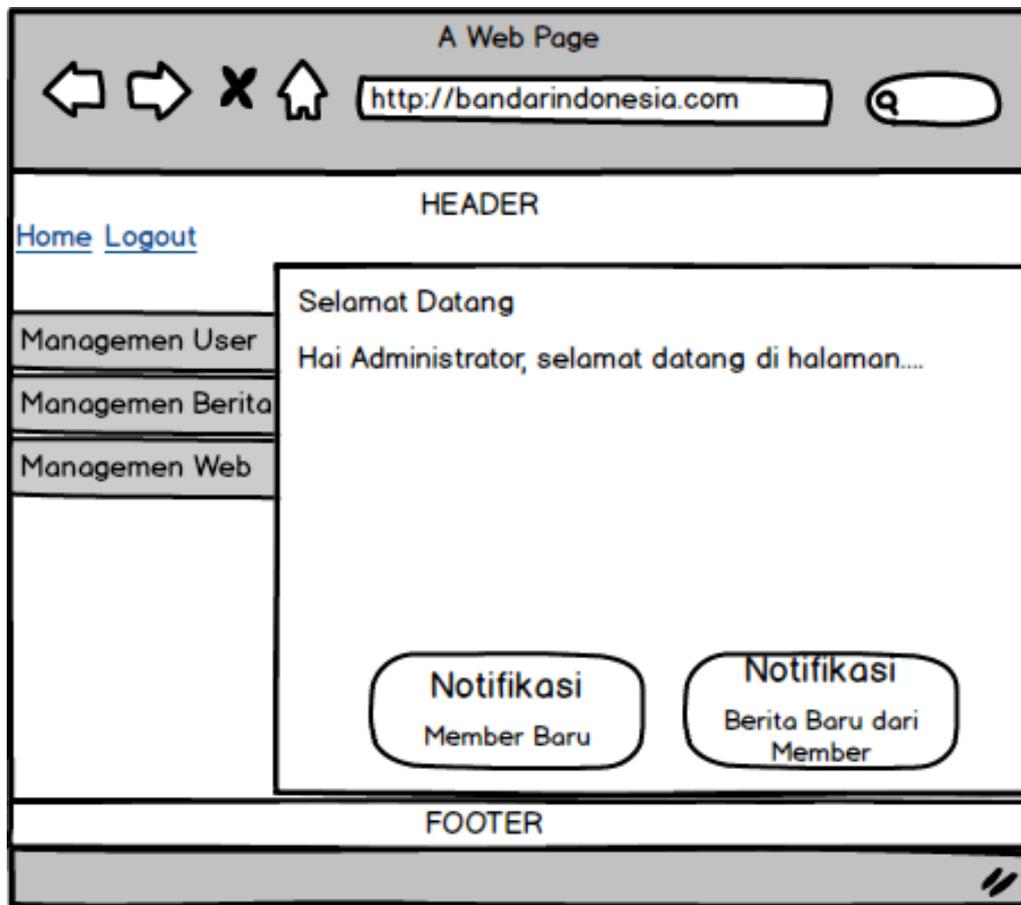
The image shows a wireframe of a web browser window. The title bar reads "A Web Page". The address bar contains "http://bandarindonesia.com". The main content area is titled "Administrator Login" and contains a form with the following elements:

- A header "LOGIN" above the input fields.
- A text input field labeled "Username".
- A text input field labeled "Password".
- A button labeled "SIGN IN" below the password field.

Gambar 3. 16 Rancangan halaman *login*

Rancangan untuk halaman *login* akan dibuat simpel, hanya ada *text box* untuk mengisi *username* dan *password*. Tersedia satu *button* bernama *sign in* untuk memberikan perintah masuk ketika pengguna sudah mengisi *username* dan *password*.

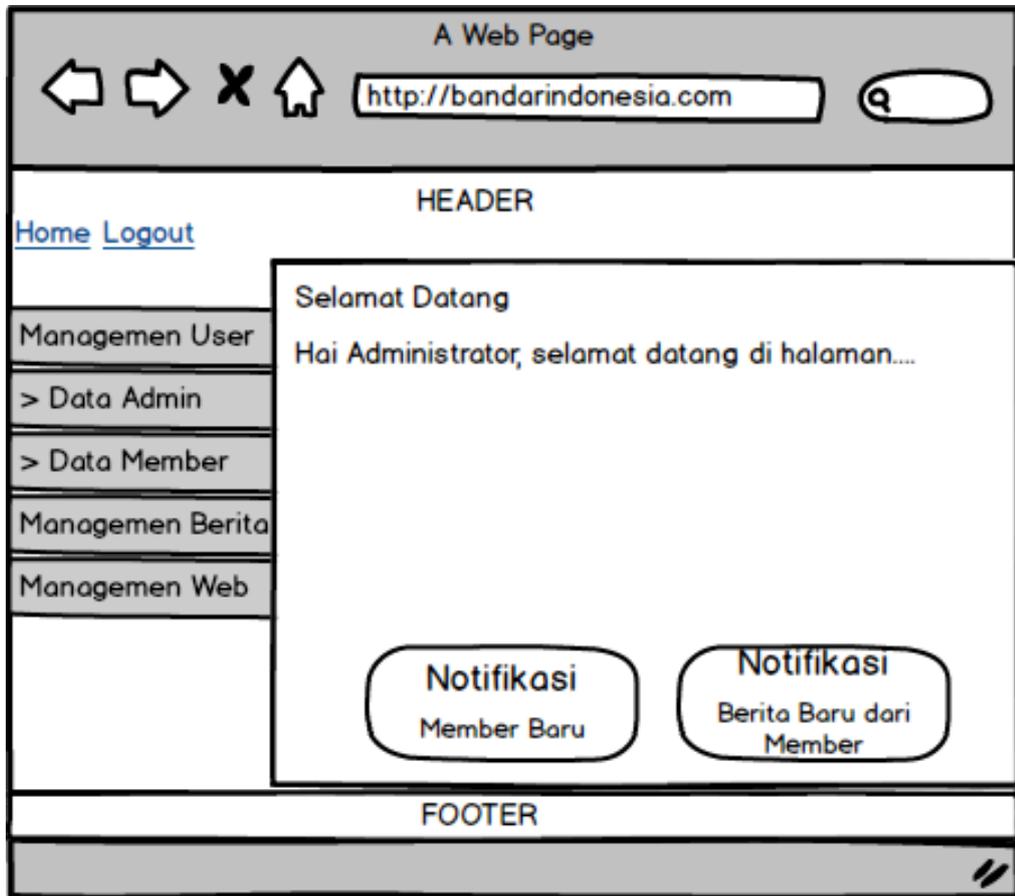
b. Rancangan halaman *home*



Gambar 3. 27 Rancangan halaman *home*

Halaman *home* akan berisi ucapan selamat datang kepada administrator ataupun pengguna lainnya. Terdapat beberapa pilihan menu utama yang dapat diakses oleh administrator. Didalam rancangan menu utama tersedia notifikasi menunjukkan terkait informasi baru.

c. Rancangan halaman manajemen *user*

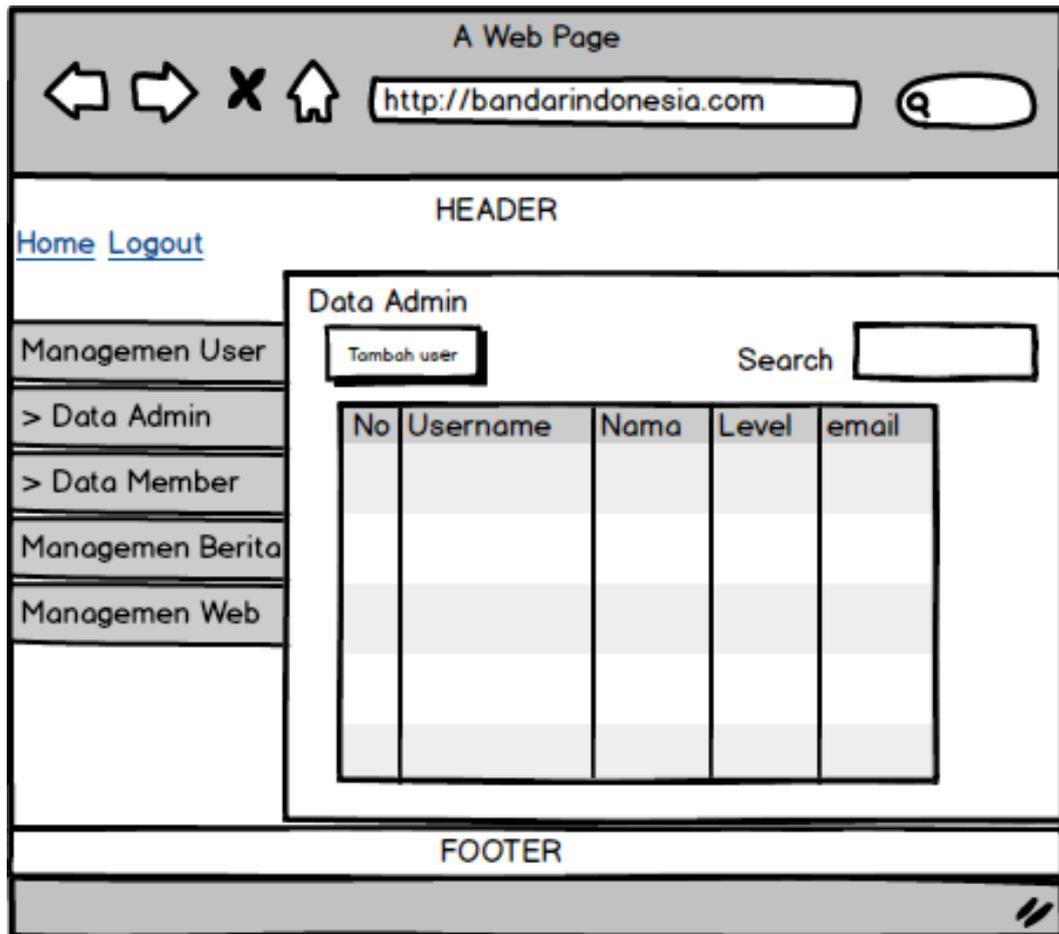


Gambar 3. 38 Rancangan halaman manajemen *user*

Halaman manajemen *user* memiliki dua pilihan fungsi yaitu data admin dan data *member* yang dapat diakses oleh administrator. Halaman manajemen *user* hanya petunjuk halaman saja dan tidak memiliki fungsi lain kecuali sebagai kategori pilihan dalam *back end website*.

d. Rancangan halaman manajemen *user* khusus administrator

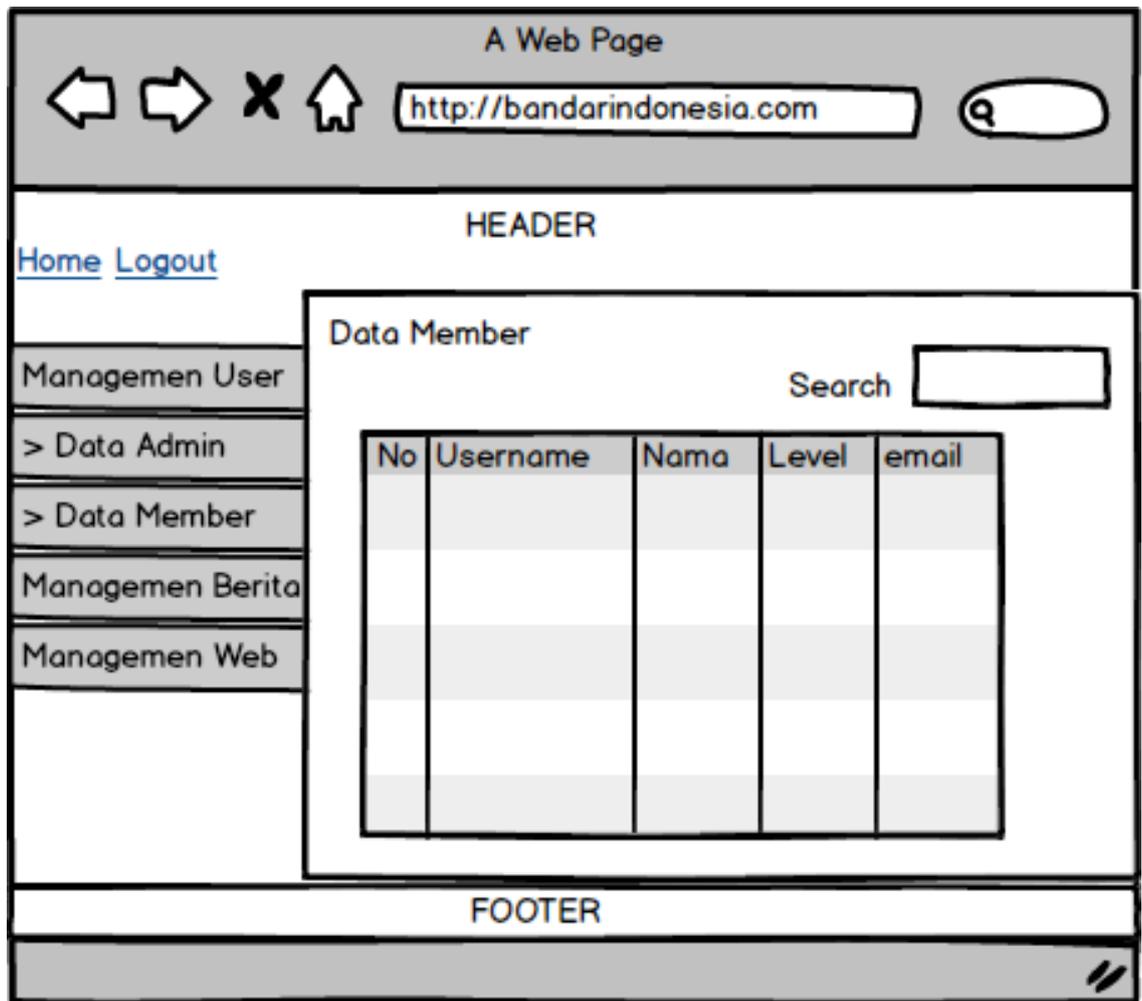
Rancangan data administrator berupa tabel data yang berisi informasi terkait administrator dan editor. Admin dapat melakukan penambahan *user* yang nantinya disesuaikan levelnya (administrator, editor atau *member*). Terdapat fungsi *search* yang dapat digunakan jika ingin mencari data.



Gambar 3. 49 Rancangan halaman data administrator

e. Rancangan halaman manajemen *user* khusus *member*

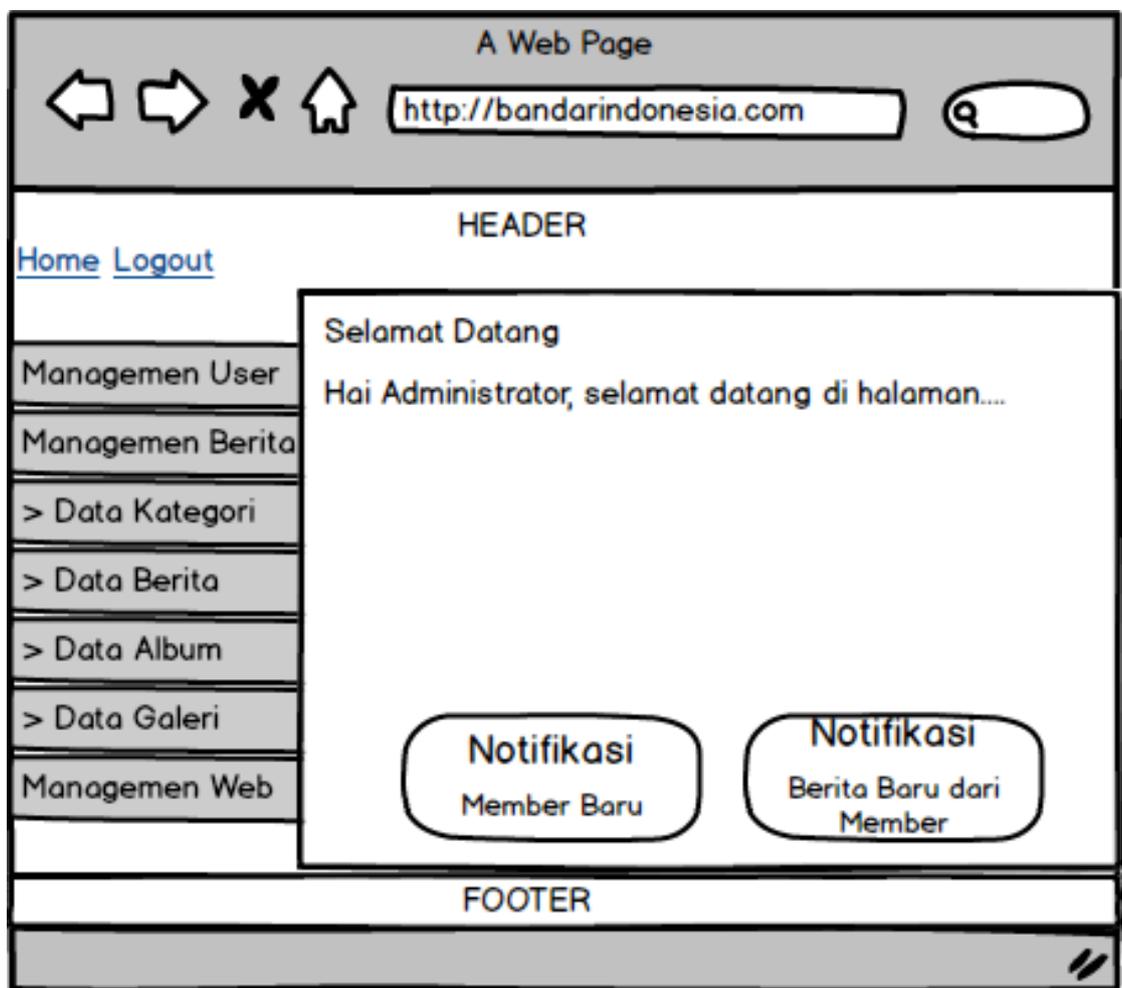
Sama dengan data admin, data *member* juga berisi *list* data dari *member* beserta informasi dari *member* terkait. Fungsi *search* pada data *member* memiliki fungsi yang sama seperti pada halaman data admin.



Gambar 3. 30 Rancangan halaman *member*

f. Rancangan halaman manajemen berita

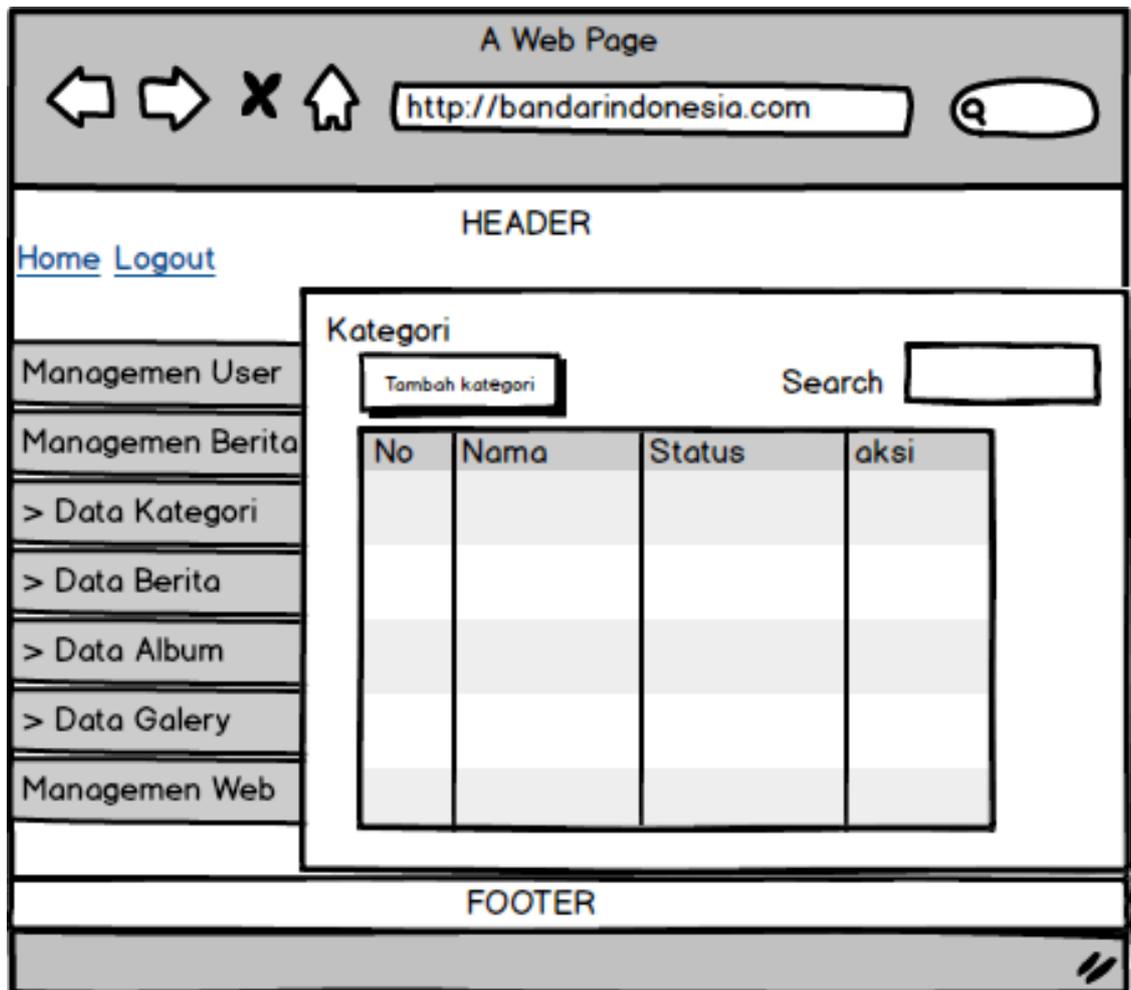
Halaman manajemen berita memiliki empat pilihan fungsi yaitu data kategori, data berita, data album dan data galeri yang semua fungsi tersebut dapat diakses oleh administrator. Halaman manajemen berita hanya petunjuk halaman saja dan tidak memiliki fungsi lain kecuali sebagai kategori pilihan dalam *back end website*.



Gambar 3. 31 Rancangan halaman manajemen berita

g. Rancangan halaman manajemen berita khusus kategori

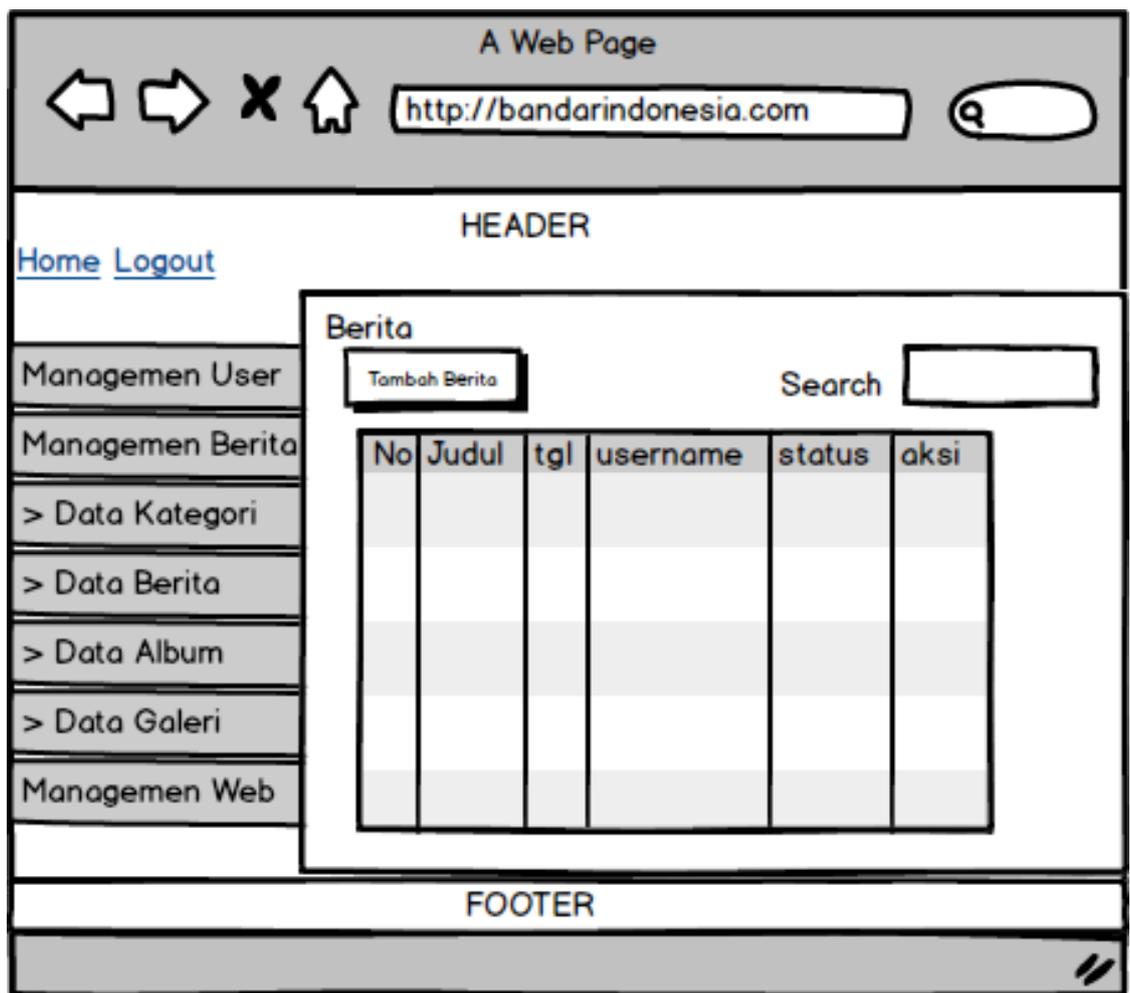
Halaman kategori akan menampilkan data lengkap terkait kategori yang telah tersedia. Selain itu, pengguna dapat menambah kategori jika diperlukan.



Gambar 3. 35 Rancangan halaman kategori

h. Rancangan halaman manajemen berita khusus berita

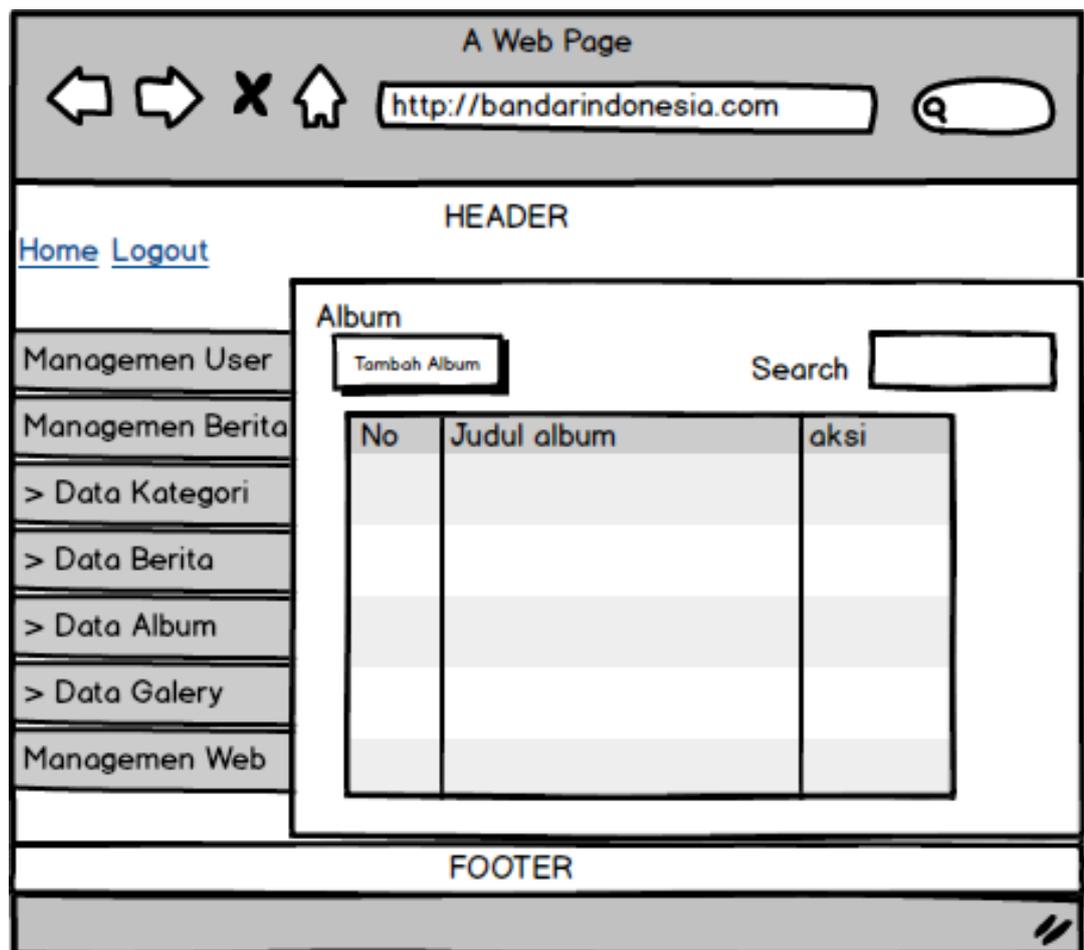
Rancangan halaman berita berisi data berita yang tersedia didalam *database*. Administrator dapat menerima dan menolak berita yang dikirim oleh *member* dari halaman ini. Selain itu, administrator dapat menambahkan berita yang sumber berita berasal dari administrator.



Gambar 3. 33 Rancangan halaman berita

i. Rancangan halaman manajemen berita khusus album

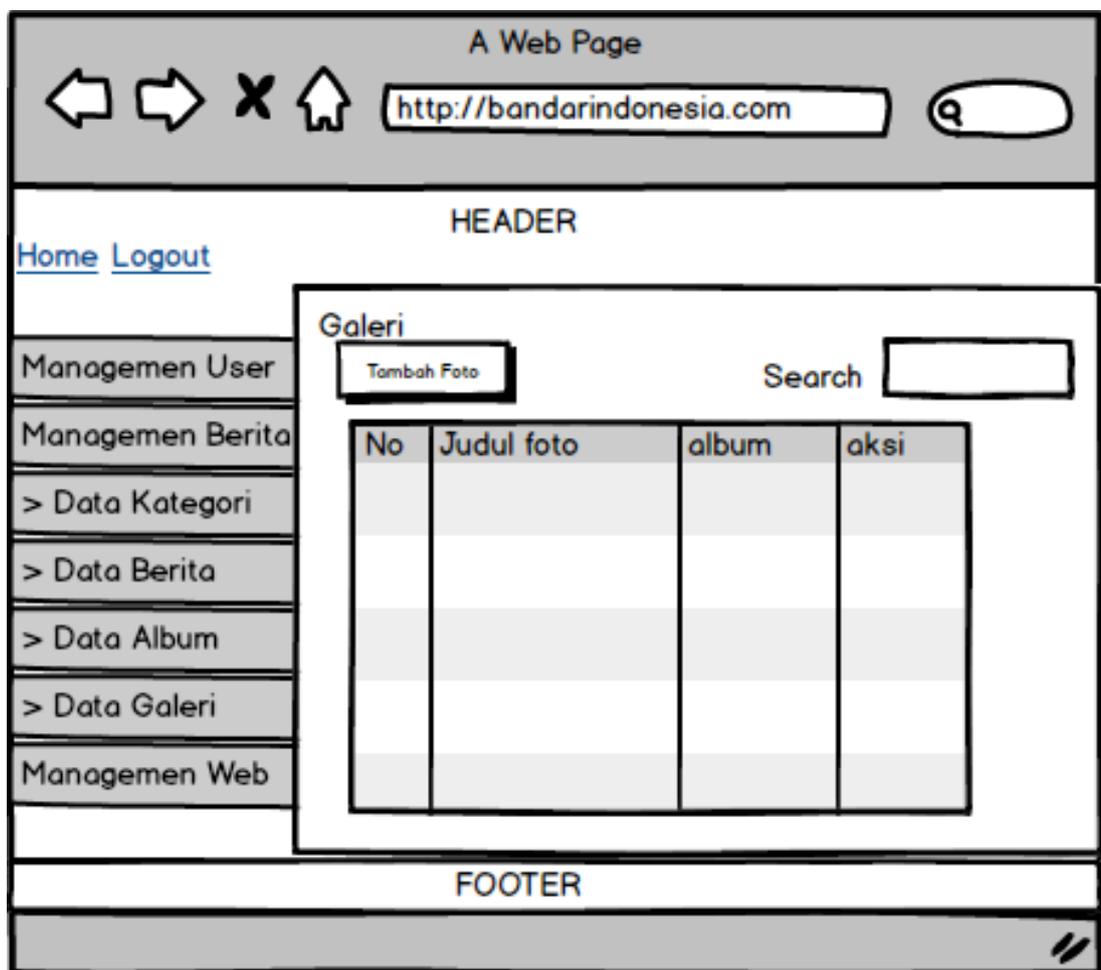
Album merupakan wadah penyusun untuk foto atau galeri. Jadi data album pada gambar 3.34 merupakan data yang berisi album-album foto. Pada halaman ini, administrator dapat menambahkan album-album baru dan menghapus album lama.



Gambar 3. 34 Rancangan halaman album

j. Rancangan halaman manajemen berita khusus *gallery*

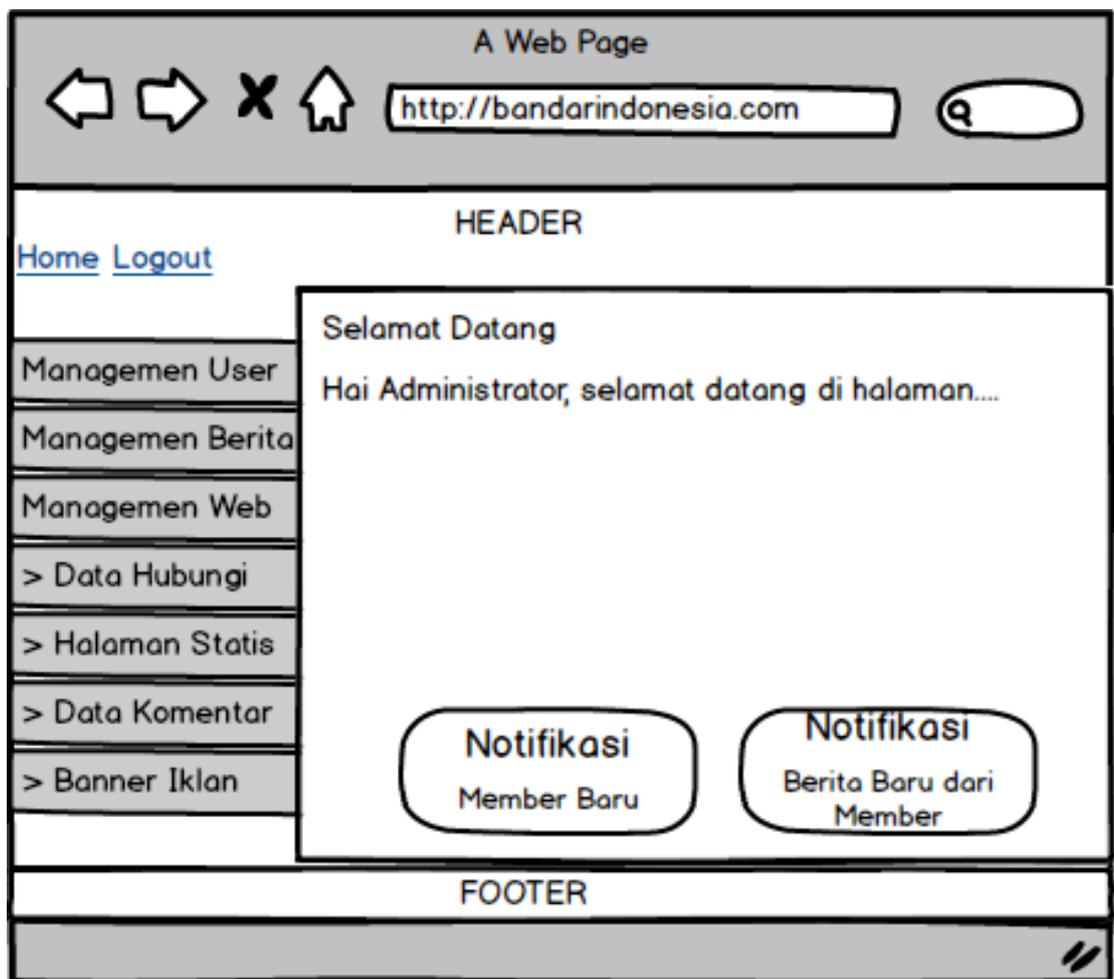
Data-data foto yang ada didalam *database* akan ditampilkan pada tabel halaman galeri. Foto-foto akan disusun berdasarkan album foto yang telah tersedia. Selain mengelola foto yang datangnya dari *member*, administrator dapat menambahkan foto pada halaman galeri untuk ditampilkan pada *front end website* atau *website* utama.



Gambar 3. 35 Rancangan halaman galeri

k. Rancangan halaman manajemen web

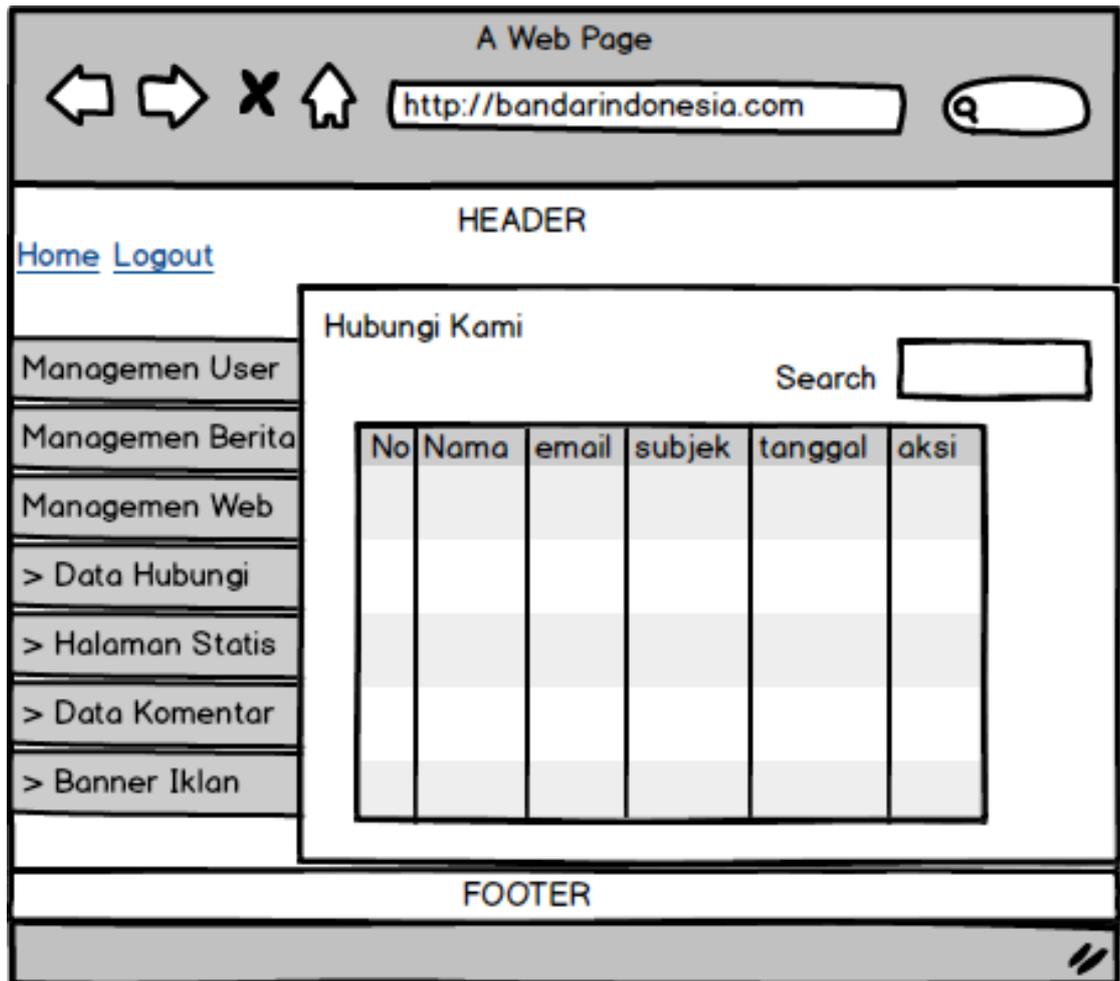
Halaman manajemen web memiliki empat pilihan fungsi yaitu data hubungi, halaman statis, data komentar dan banner iklan yang semua fungsi tersebut dapat diakses oleh administrator. Halaman manajemen web hanya petunjuk halaman saja dan tidak memiliki fungsi lain kecuali sebagai kategori pilihan dalam *back end website*.



Gambar 3. 36 Rancangan halaman manajemen web

1. Rancangan halaman manajemen *web* khusus hubungi

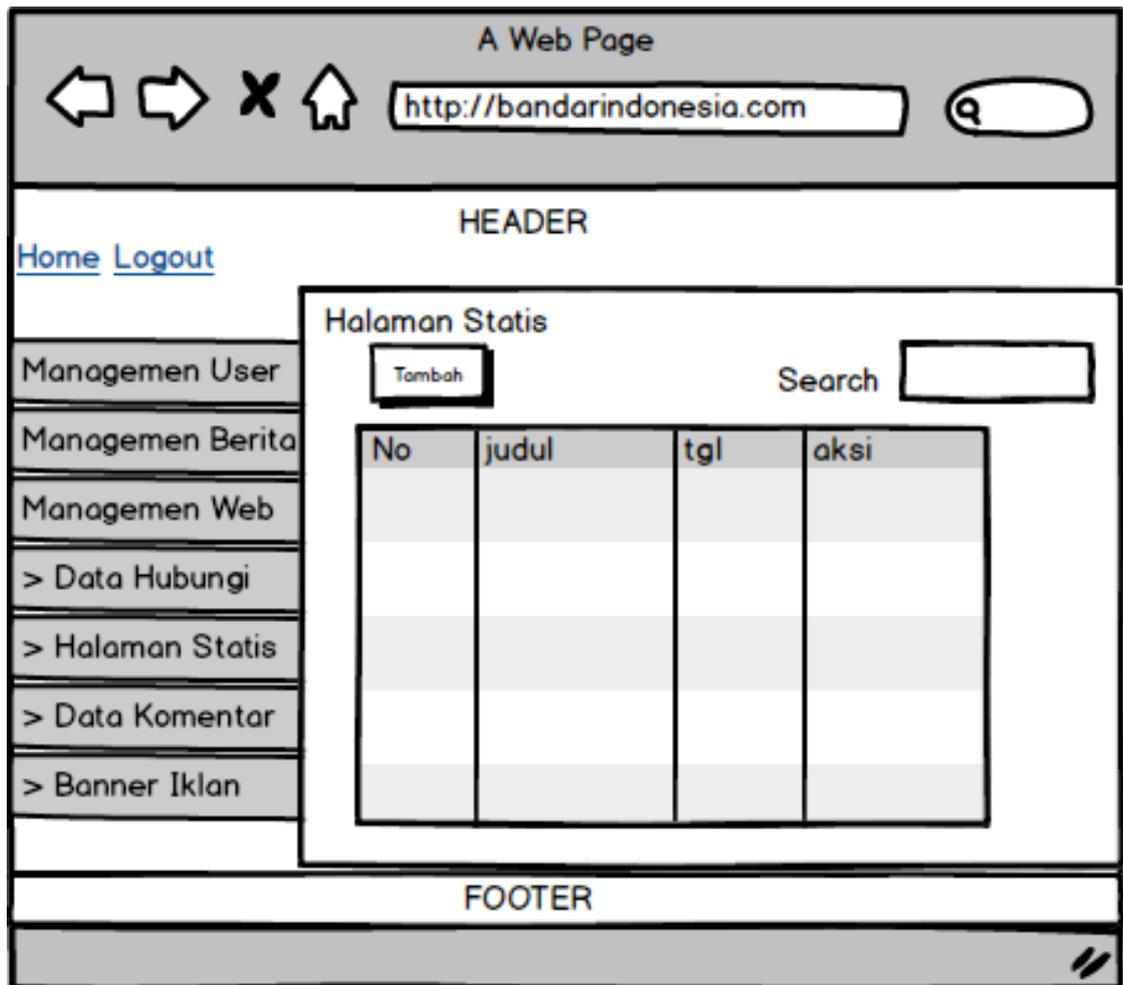
Pesan-pesan dari *member* akan terdata didalam *database* dan ditampilkan pada halaman data hubungi. Dari halaman data hubungi, administrator dapat menjawab pertanyaan dari *member*.



Gambar 3. 37 Rancangan halaman data hubungi

m. Rancangan halaman manajemen *web* khusus halaman statis

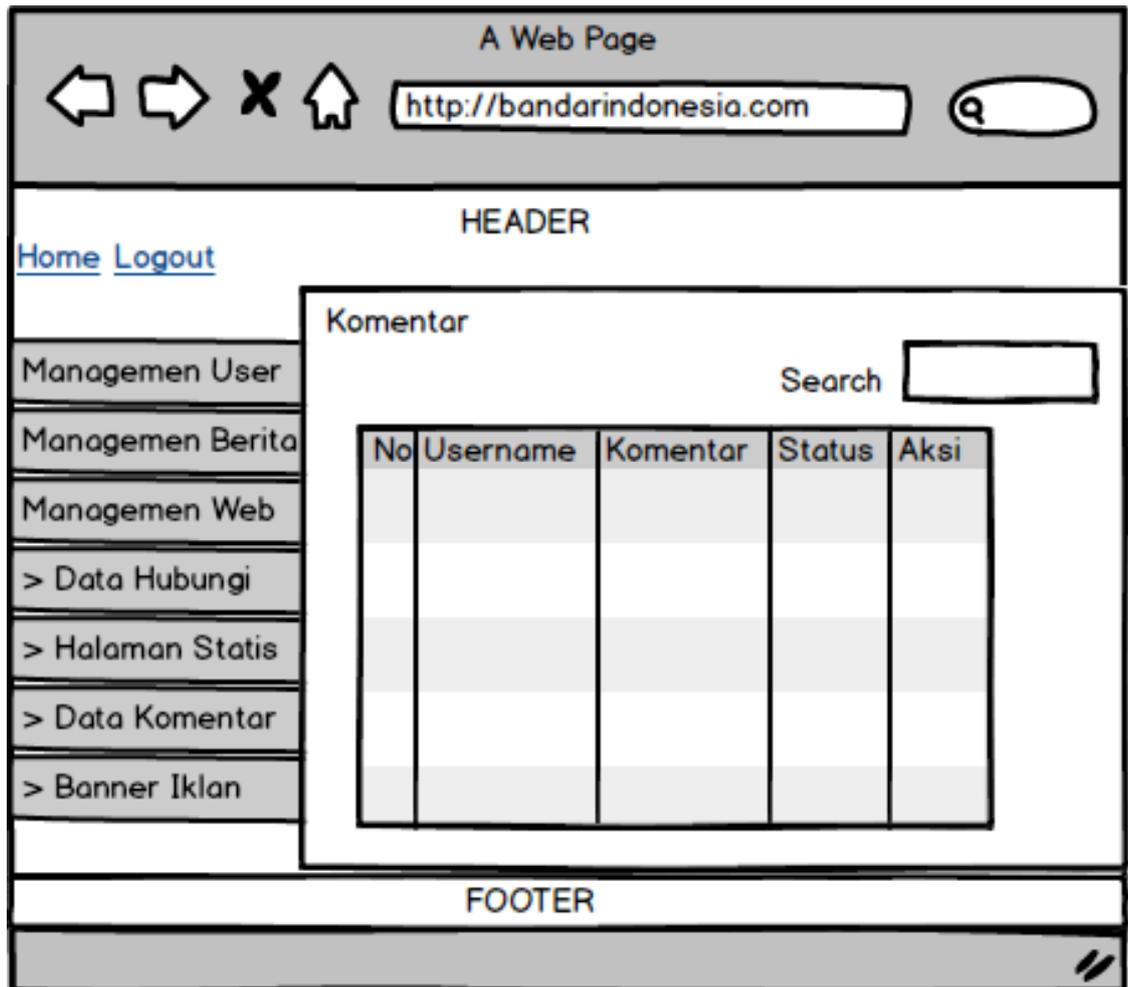
Data halaman statis berisi tutorial singkat terkait tata cara *mengupload* berita dan foto. Halaman statis dapat *diupdate* dan ditambahkan jumlahnya. Selain itu, halaman statis juga bisa diaktifkan dan tidak diaktifkan



Gambar 3. 38 Rancangan halaman statis

n. Rancangan halaman manajemen *web* khusus komentar

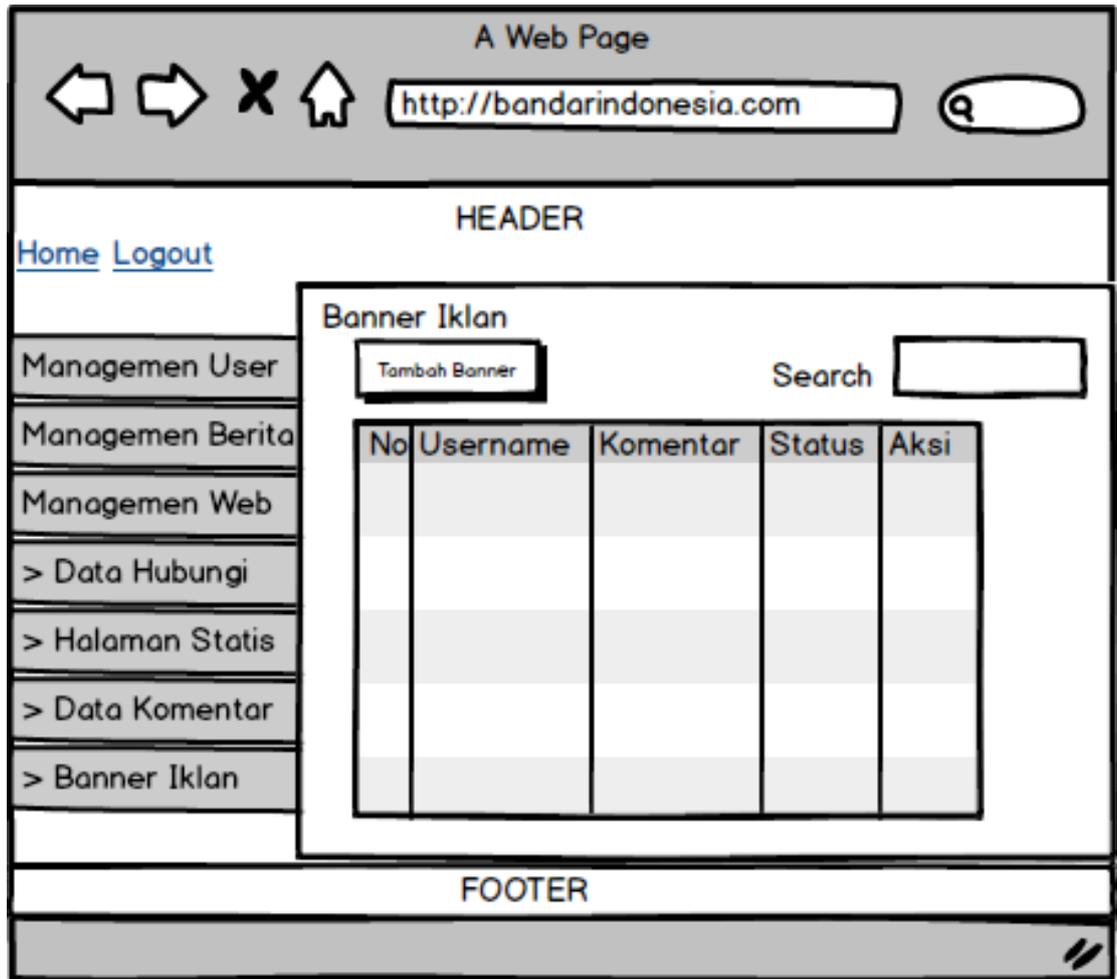
Halaman komentar berisi data komentar yang ketika ada komentar bermasalah, administrator dapat mempertimbangkan untuk menghapus komentar dari *member* tersebut. Pengaturan terkait komentar tadi dapat dilakukan oleh administrator pada halaman komentar.



Gambar 3. 39 Rancangan halaman komentar

o. Rancangan halaman manajemen *web* khusus *banner*

Tidak ada ubahnya dengan halaman lain, halaman *banner* berisi data yang berasal dari *database*. *Banner* iklan dapat ditampilkan dan dimasukkan kedalam *front end website* dari halaman *banner* di *back end website*.



Gambar 3. 40 Rancangan halaman *banner*

Perancangan *interface* pada *website* pengelola khusus untuk aktor editor adalah sebagai berikut :

a. Rancangan halaman *login*

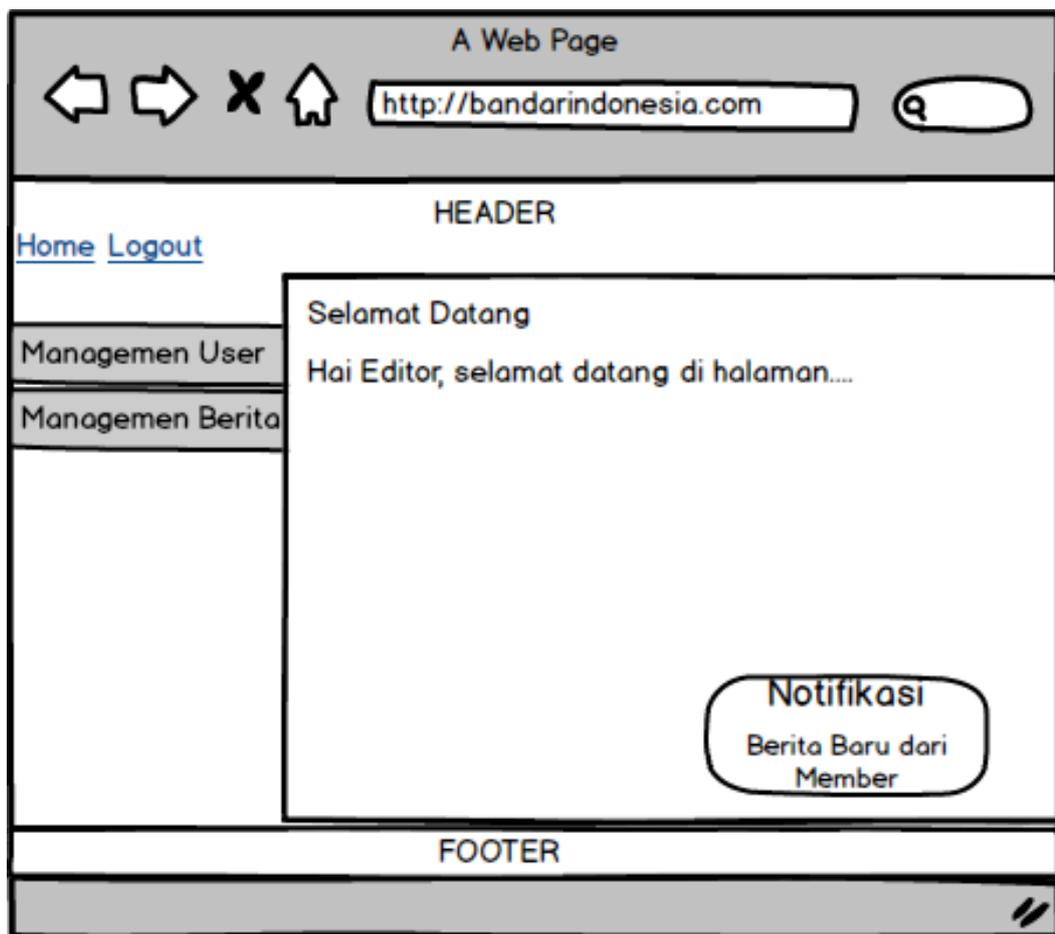
The image shows a wireframe of a web browser window. The title bar reads "A Web Page". The address bar contains the URL "http://bandarindonesia.com". The main content area is titled "Administrator Login" and features a "LOGIN" form. The form includes two input fields: "Username" and "Password", followed by a "SIGN IN" button. The browser window also shows navigation icons (back, forward, stop, home) and a search icon.

Gambar 3. 41 Rancangan halaman *login*

Tidak ada perbedaan rancangan desain halaman *login* untuk administrator dan editor. Perbedaannya akan terlihat ketika editor tersebut sudah berhasil *login*.

b. Rancangan halaman *home*

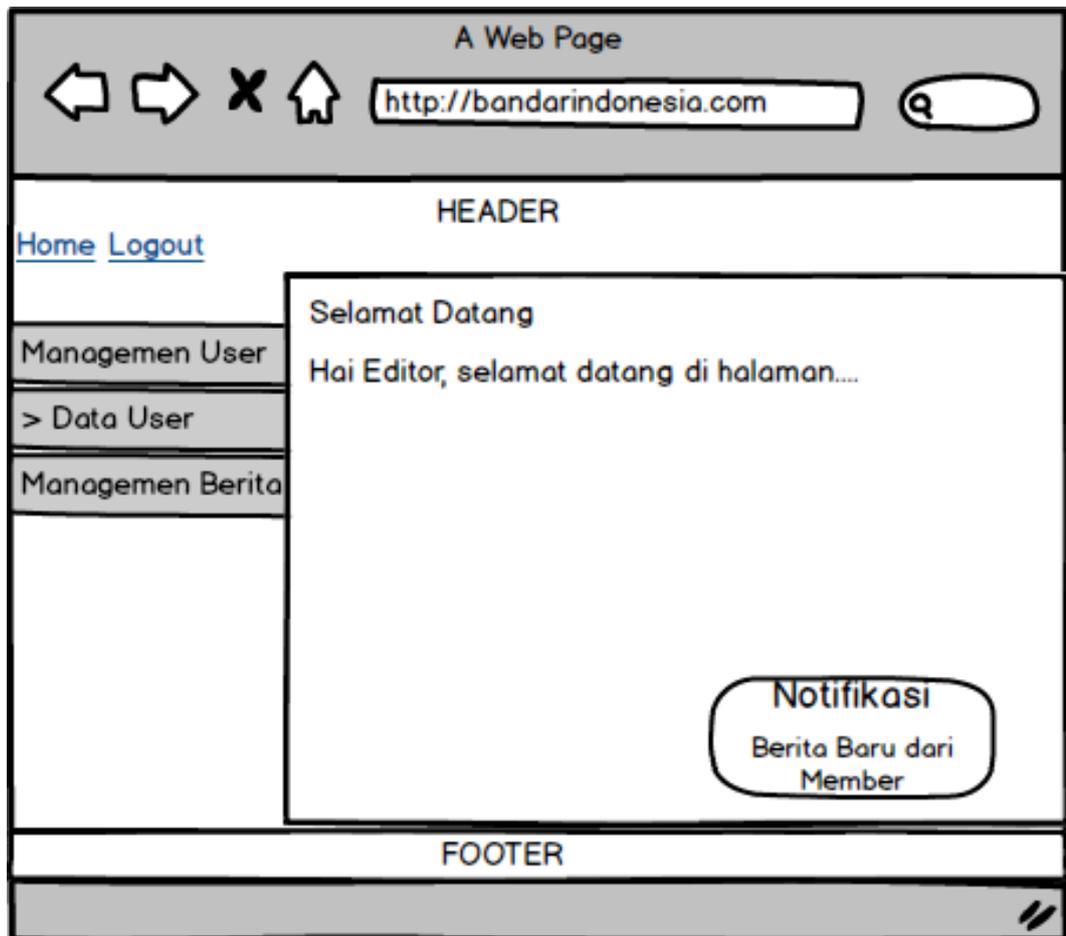
Pada gambar 3.42 terlihat perbedaan antara rancangan halaman *home* untuk administrator dengan halaman *home* untuk editor. Pilihan dari halaman *home* untuk editor lebih sedikit dibandingkan dengan halaman *home* untuk administrator.



Gambar 3. 42 Rancangan halaman *home*

c. Rancangan halaman manajemen *user*

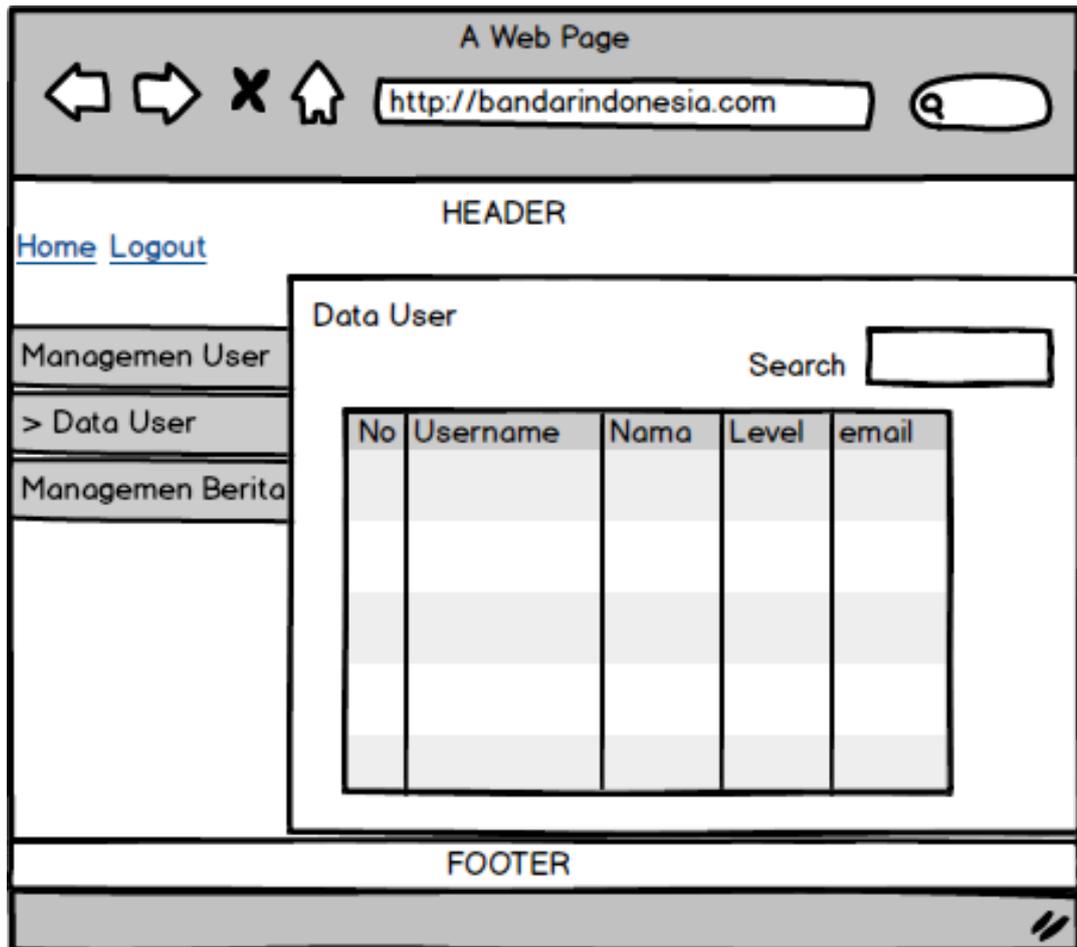
Berbeda dengan manajemen *user* pada rancangan halaman administrator, halaman *user* pada editor hanya memiliki satu pilihan fungsi yaitu data *user*.



Gambar 3. 43 Rancangan halaman manajemen *user*

d. Rancangan halaman manajemen *user* khusus data *user*

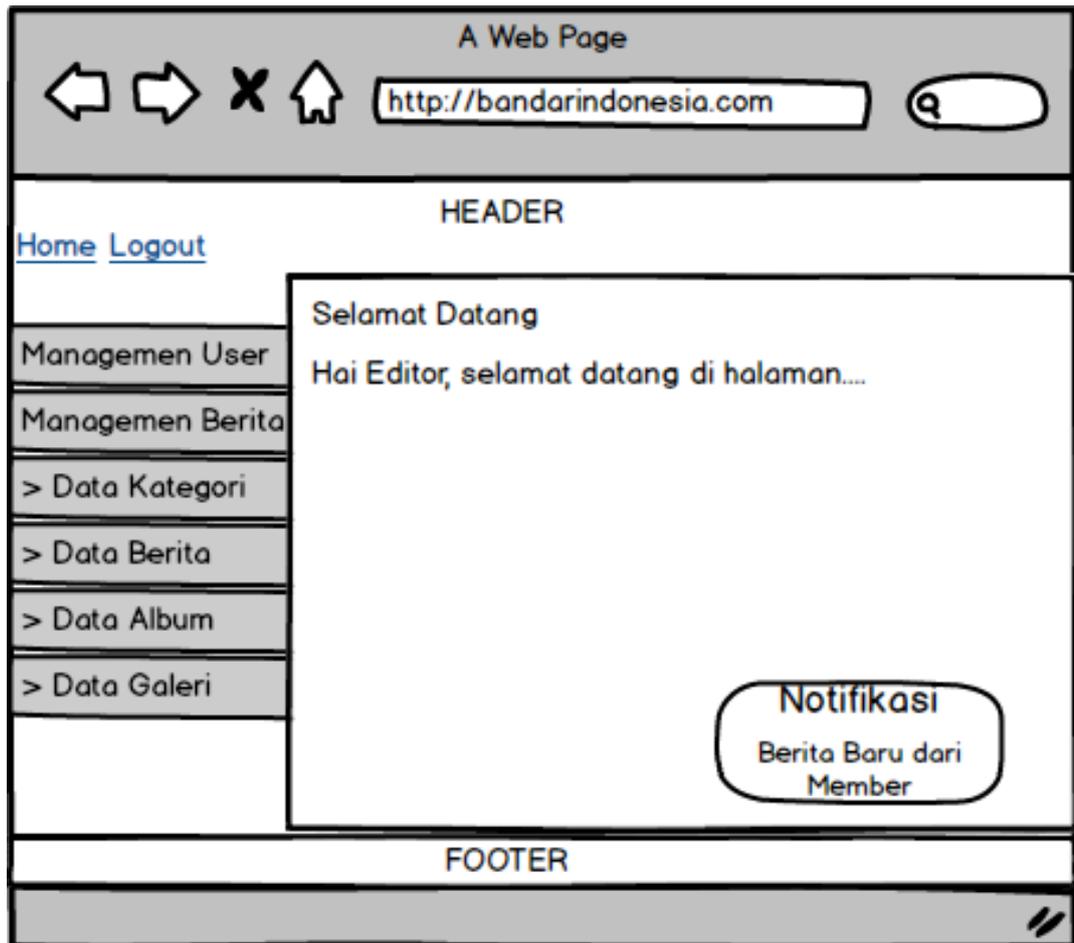
Data yang ditampilkan pada gambar 3.44 hanya data milik editor itu sendiri yang dapat diubah isi datanya. Halaman data *user* memiliki fungsi yang hampir sama dengan edit profil.



Gambar 3. 44 Rancangan halaman data *user*

e. Rancangan halaman manajemen berita

Rancangan halaman manajemen berita untuk editor memiliki jumlah pilihan kegiatan yang sama dengan halaman manajemen berita untuk administrator. Untuk isi dari pilihan yang tersedia juga tidak terdapat perbedaan.



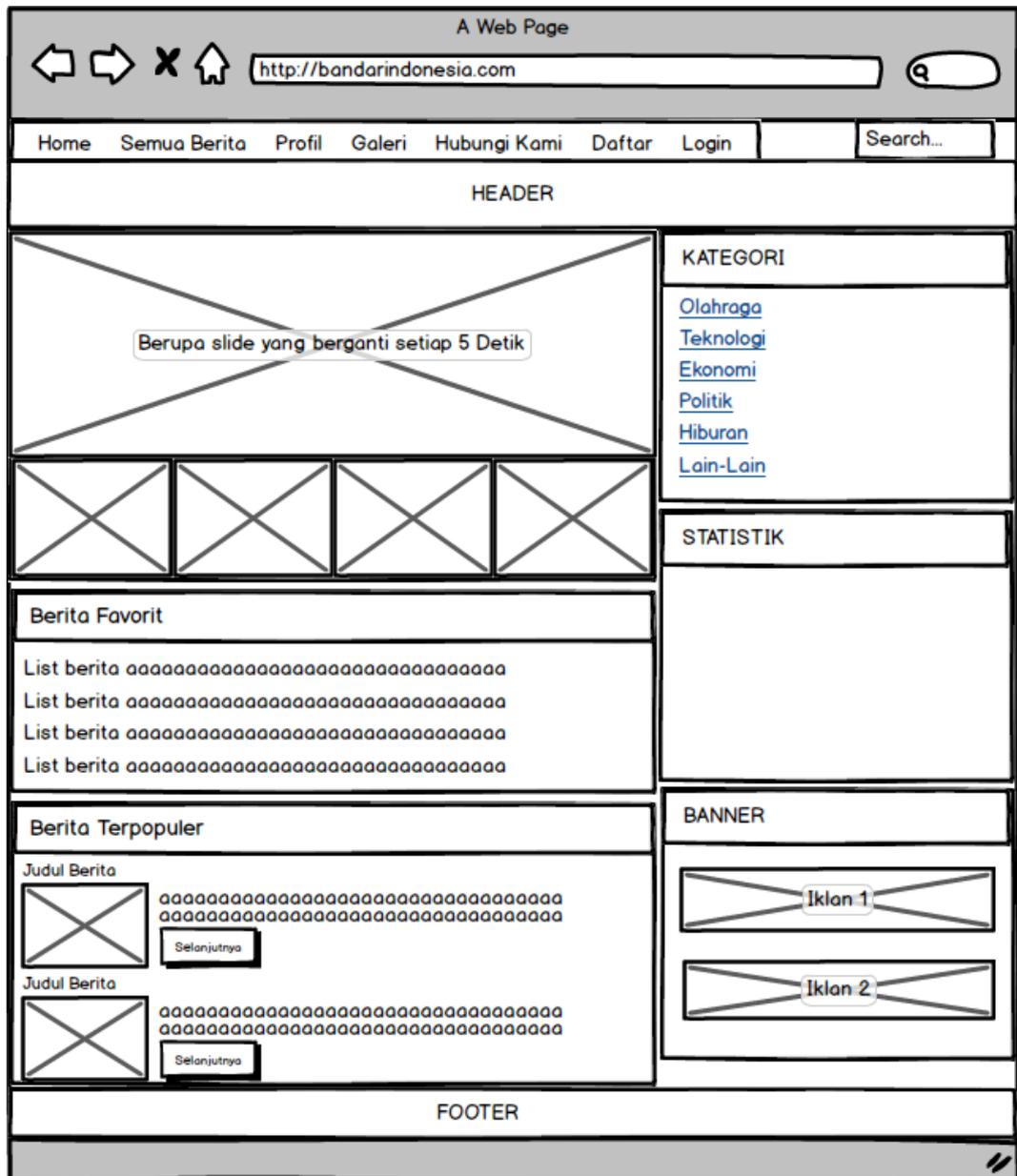
Gambar 3. 45 Rancangan halaman manajemen berita

Berbeda dengan *website* pengelola atau *back end website*, *website* utama atau *front end website* adalah *website* yang diperuntukkan untuk pengguna yang berperan sebagai aktor *user* biasa atau aktor *member*. Perancangan *interface* pada *website* utama khusus untuk aktor *user* biasa adalah sebagai berikut :

a. Rancangan halaman *home*

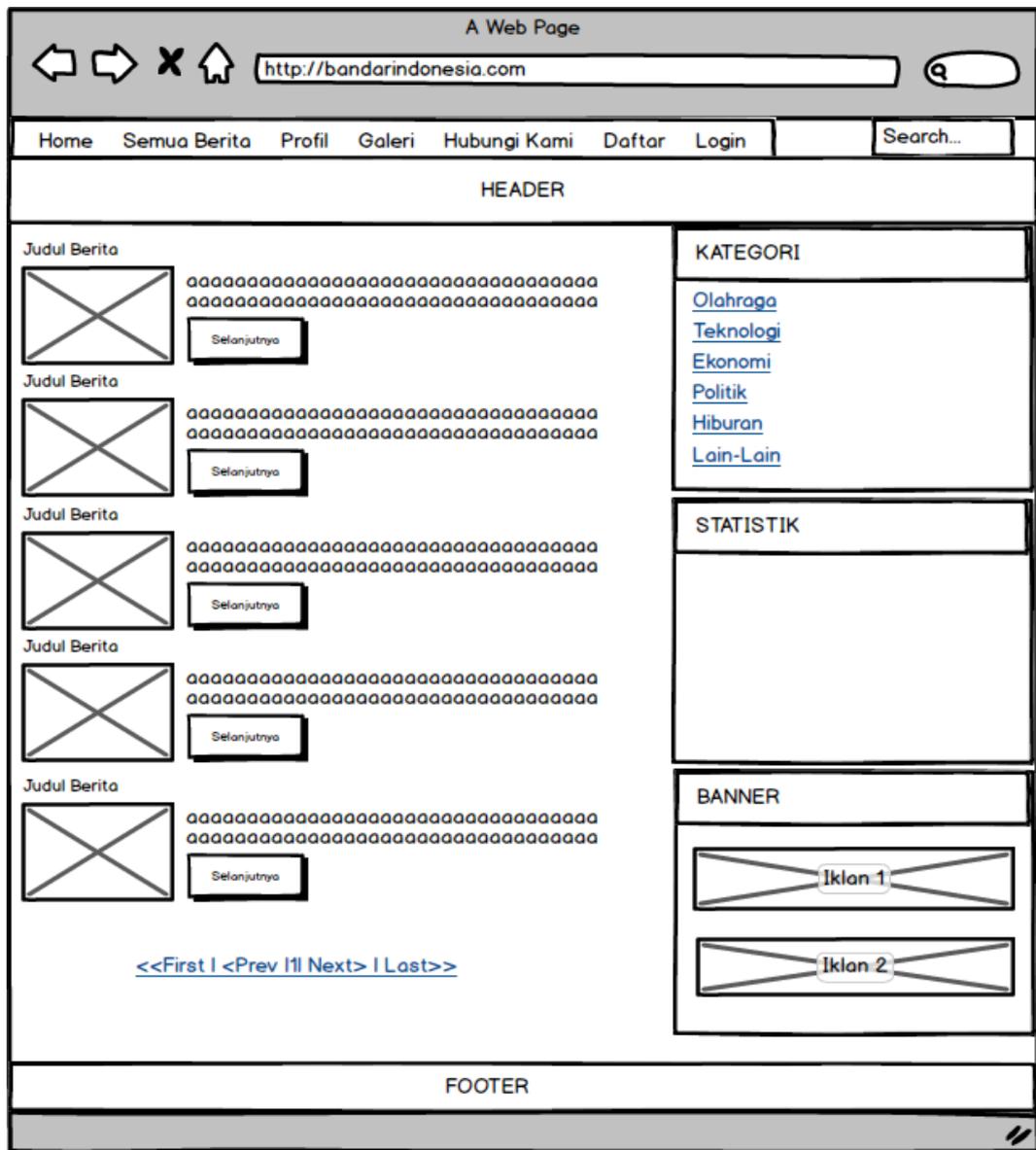
Rancangan halaman *home* pada *front end website* merupakan tampilan awal ketika pengguna membuka situs *website*. Terdapat beberapa bagian pada halaman *home*. Bagian-bagian tersebut antara lain :

1. Bagian berita utama
2. Bagian berita favorit
3. Bagian berita terpopuler
4. Bagian kategori
5. Bagian banner iklan



Gambar 3. 46 Rancangan halaman *home*

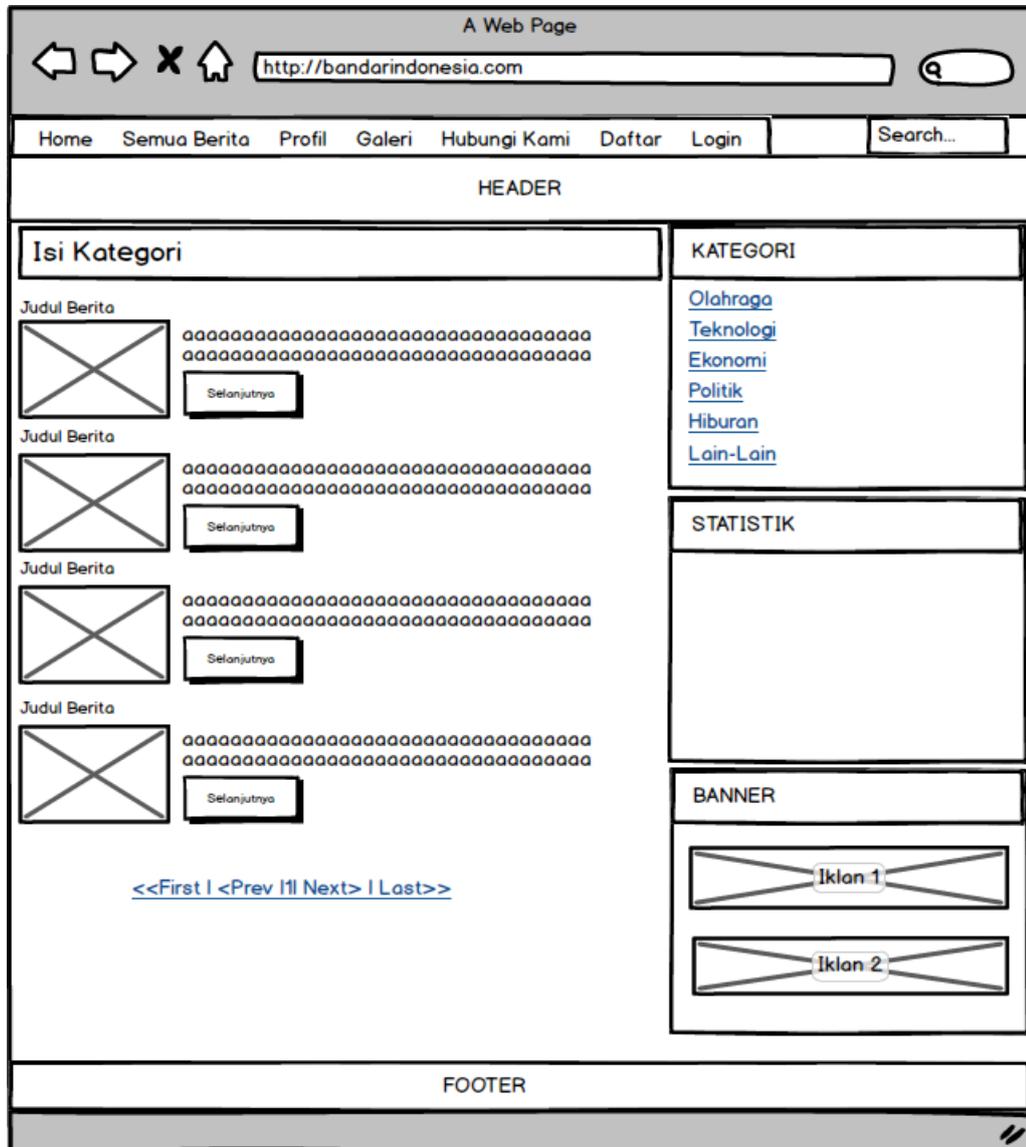
b. Rancangan halaman semua berita



Gambar 3. 47 Rancangan halaman semua berita

Rancangan halaman semua berita akan tampil ketika aktor *user* biasa atau *member* memilih *tab* semua berita pada *menu bar* di bagian paling atas. Pada gambar 3.50 terlihat jelas bahwa semua berita ditampilkan dan pengurutan berita yang ditampilkan berdasarkan waktu. Jadi berita terbaru akan muncul pada bagian paling atas.

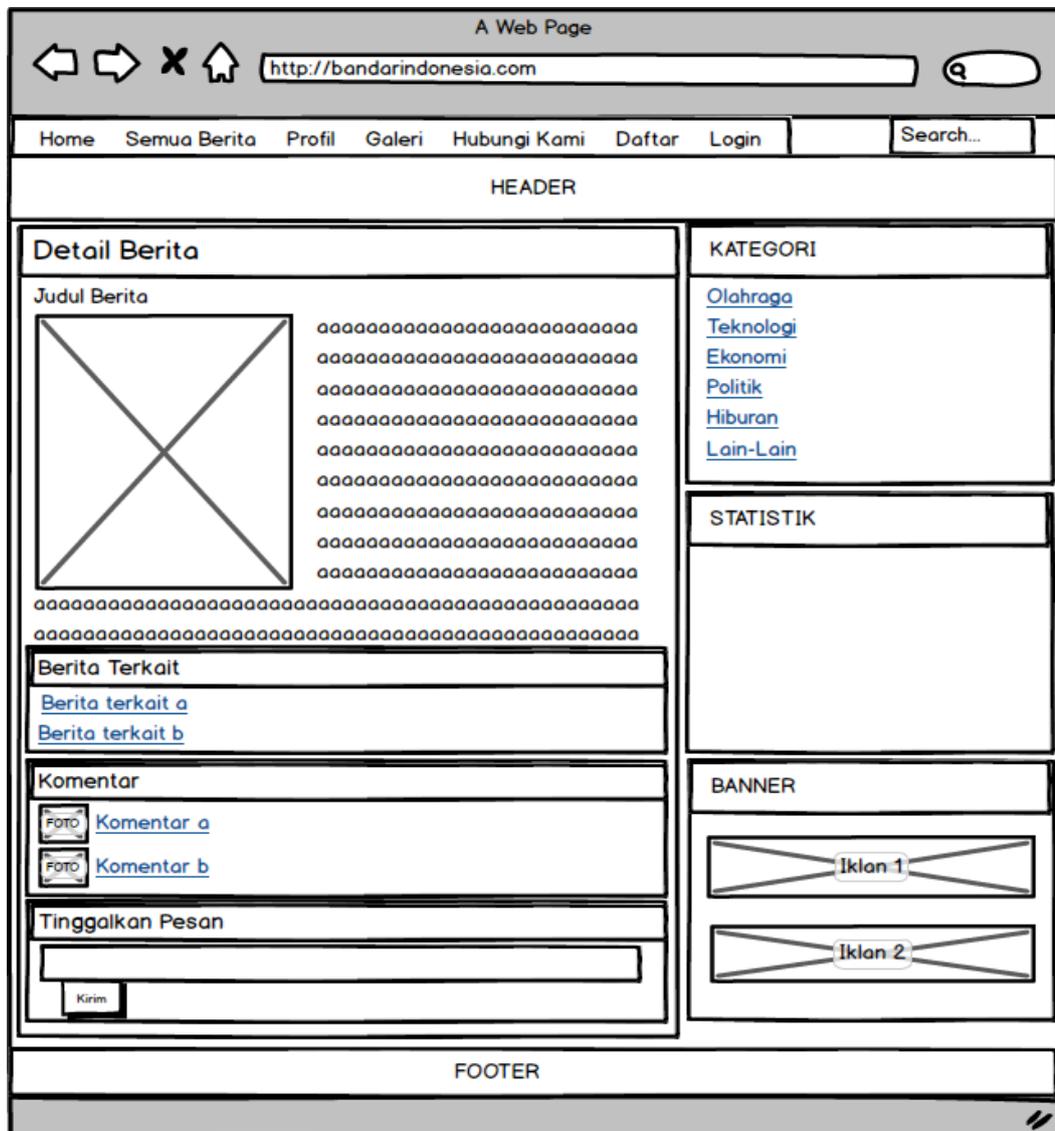
c. Rancangan halaman isi kategori



Gambar 3. 48 Rancangan halaman isi kategori

Tidak ada bedanya dengan rancangan halaman semua berita jika dilihat dari sisi tampilan, namun isi dari rancangan halaman isi kategori hanya sebatas berita dari kategori terpilih saja yang ditampilkan.

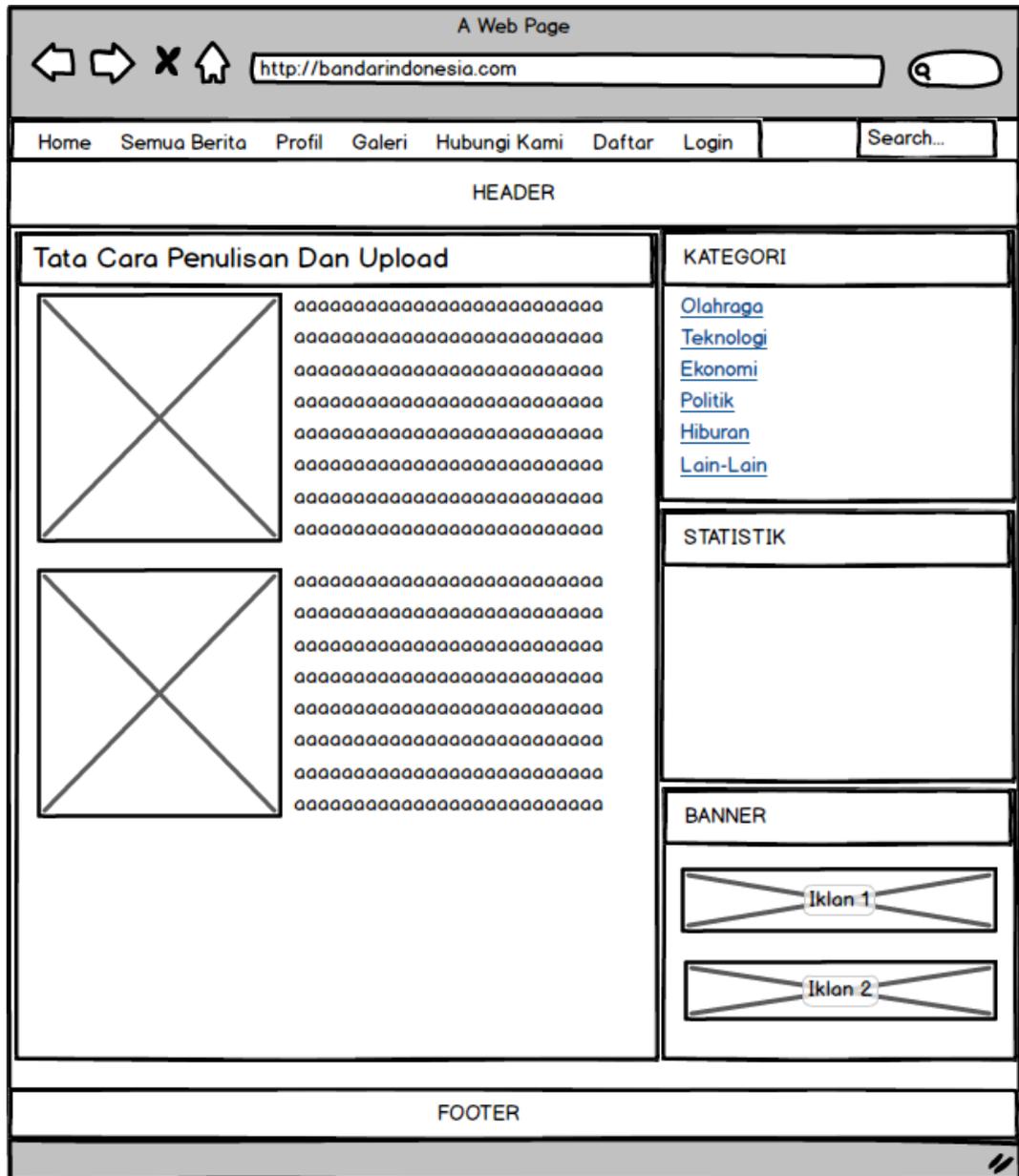
d. Rancangan halaman isi berita



Gambar 3. 49 Rancangan halaman isi berita

Ketika *user* biasa atau *member* membuka salah satu berita, rancangan tampilan yang akan muncul akan tampak seperti pada gambar 3.49. Terdapat foto dari berita dan isi berita terkait. Selain itu, *website* akan menyediakan berita terkait yang memiliki kesamaan kategori berita.

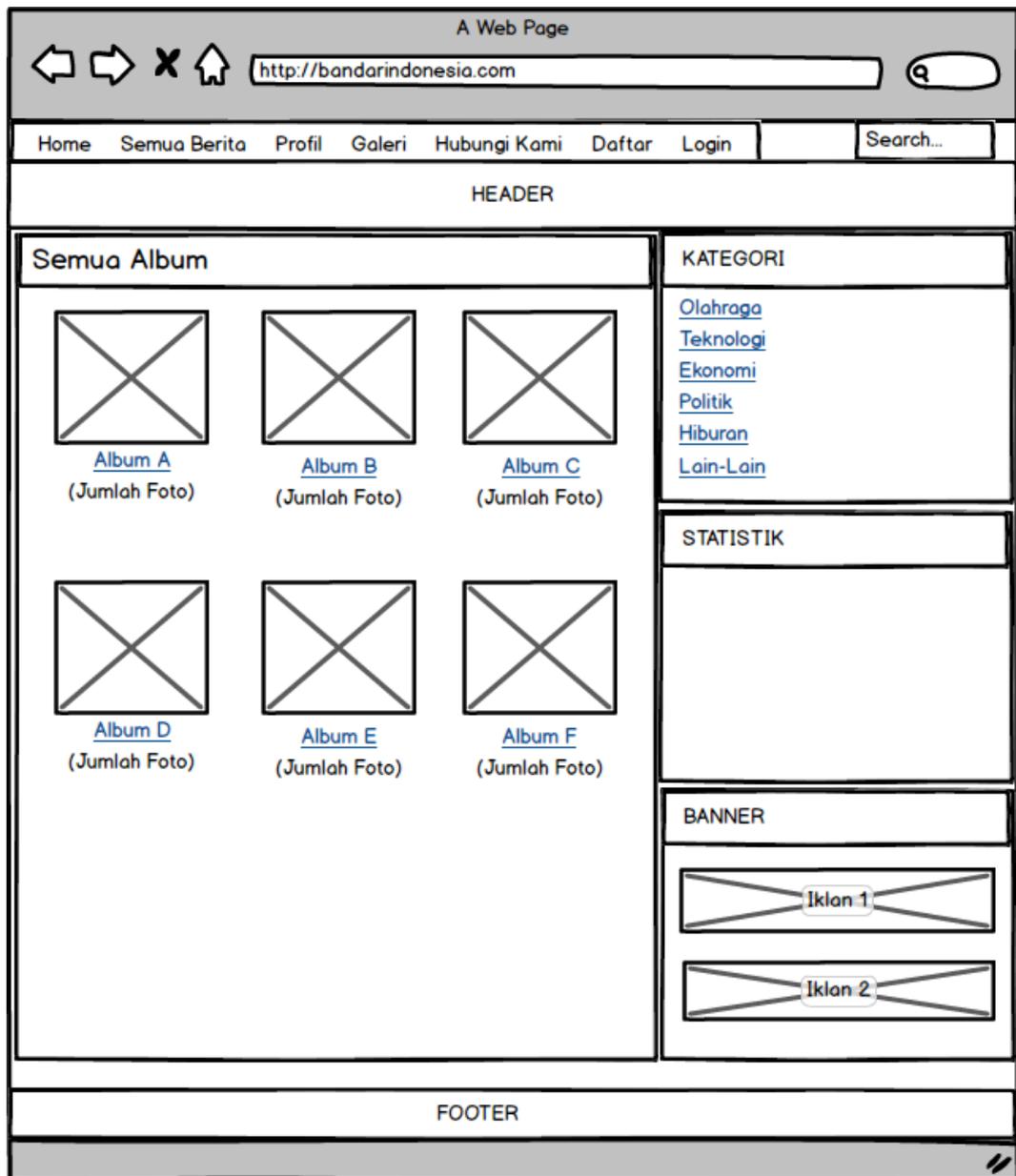
e. Rancangan halaman tata cara



Gambar 3. 50 Rancangan halaman tata cara

Rancangan halaman tata cara akan menampilkan tata cara *upload* berita dan *upload* foto.

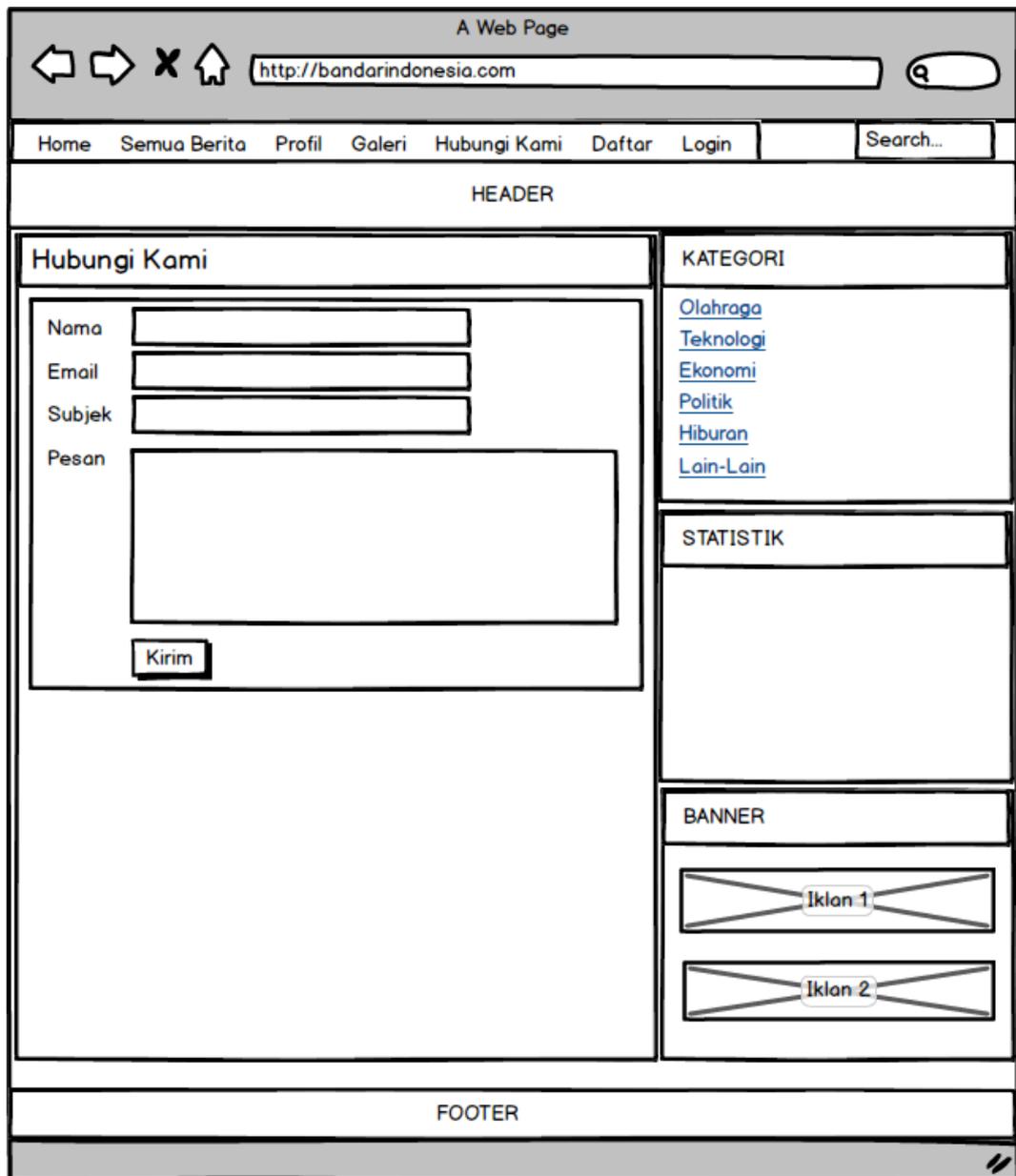
f. Rancangan halaman *gallery*



Gambar 3. 51 Rancangan halaman *gallery*

Pada gambar 3.51 terlihat album-album foto berisi foto-foto yang *diupload* oleh administrator maupun *member website*. Akan ada nama album dan jumlah foto pada album tersebut.

g. Rancangan halaman hubungi kami



Gambar 3. 52 Rancangan halaman hubungi

Halaman hubungi merupakan halaman tempat *user* biasa atau *member* berinteraksi dengan administrator untuk mengajukan pertanyaan, memberikan usulan dan mengkritik terkait isi *website*.

h. Rancangan halaman daftar *member*

A Web Page

http://bandarindonesia.com

Home Semua Berita Profil Galeri Hubungi Kami Daftar Login Search...

HEADER

Daftar Sebagai Member

Username

Password

Ulangi Pssword

Nama Lengkap

Email

No.Telp / HP

Foto

Scan KTP

KATEGORI

[Olahraga](#)

[Teknologi](#)

[Ekonomi](#)

[Politik](#)

[Hiburan](#)

[Lain-Lain](#)

STATISTIK

BANNER

Iklan 1

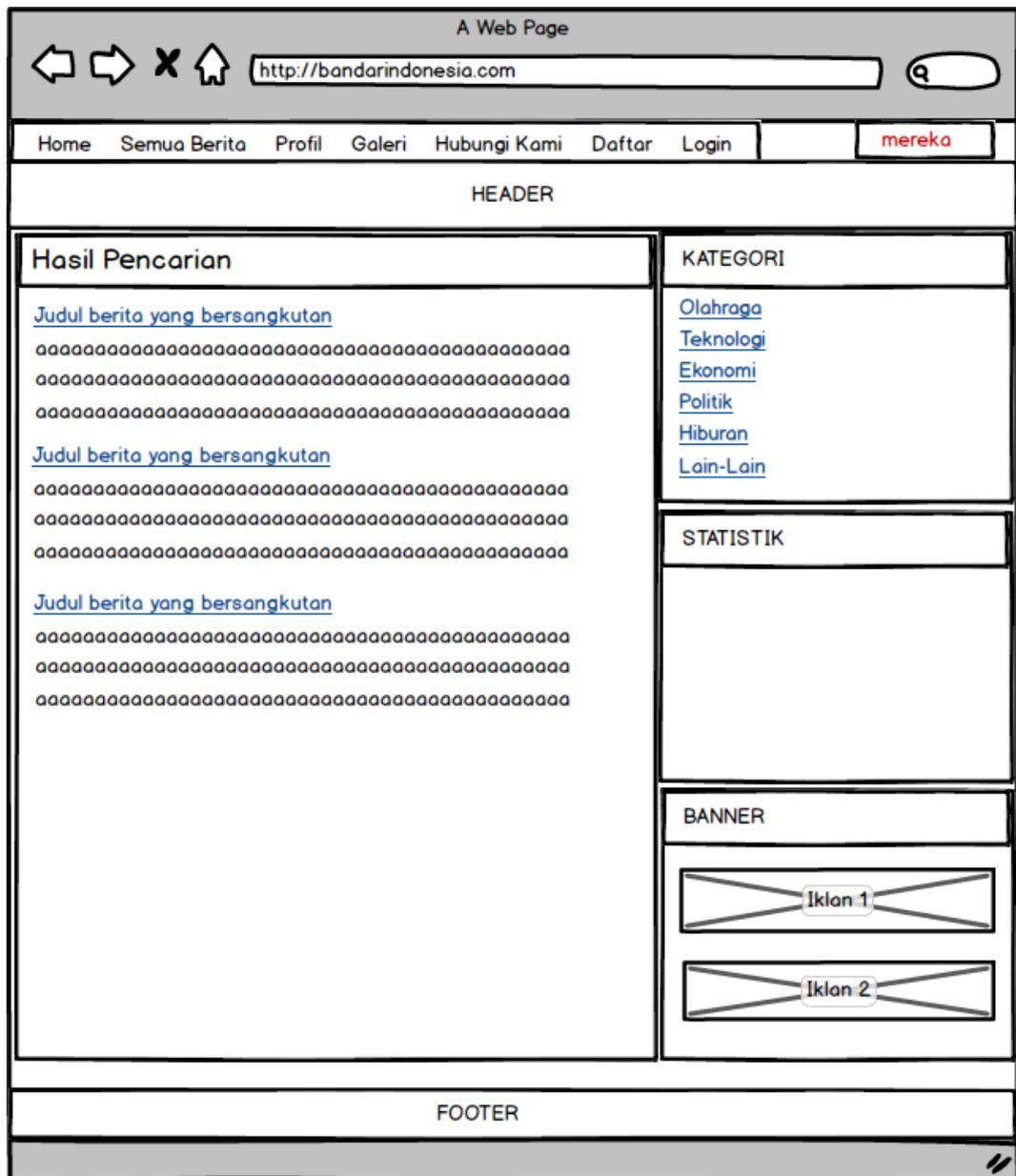
Iklan 2

FOOTER

Gambar 3. 53 Rancangan halaman daftar *member*

Untuk menjadi *member*, *user* biasa harus mendaftarkan dirinya melalui halaman daftar *member* yang rancangannya tergambar pada gambar 3.53.

i. Rancangan halaman *search*

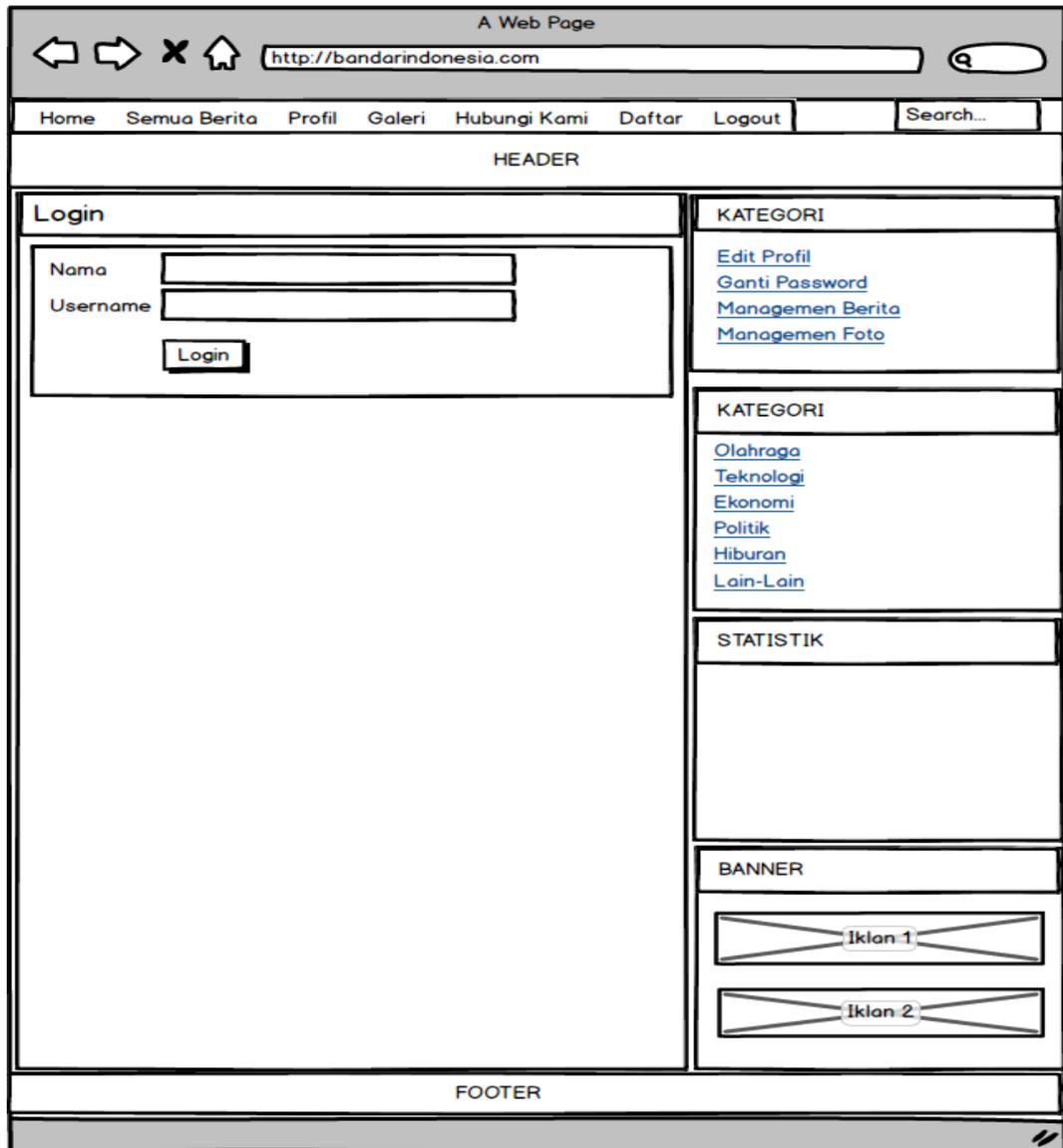


Gambar 3. 54 Rancangan halaman *search*

Fungsi *search* dapat dilihat jelas pada gambar 3.54. Terdapat *tools search* pada pojok kanan atas untuk memasukkan *key word* yang ingin dicari. Selanjutnya halaman akan menampilkan hasil dari *key word* yang dimasukkan.

Semua yang dilakukan oleh aktor *user* biasa dapat juga dilakukan oleh aktor *member*, namun sebaliknya ada beberapa kegiatan yang tidak dapat dilakukan oleh aktor *user* biasa. Perancangan *interface* pada *website* utama khusus untuk aktor *member* adalah sebagai berikut :

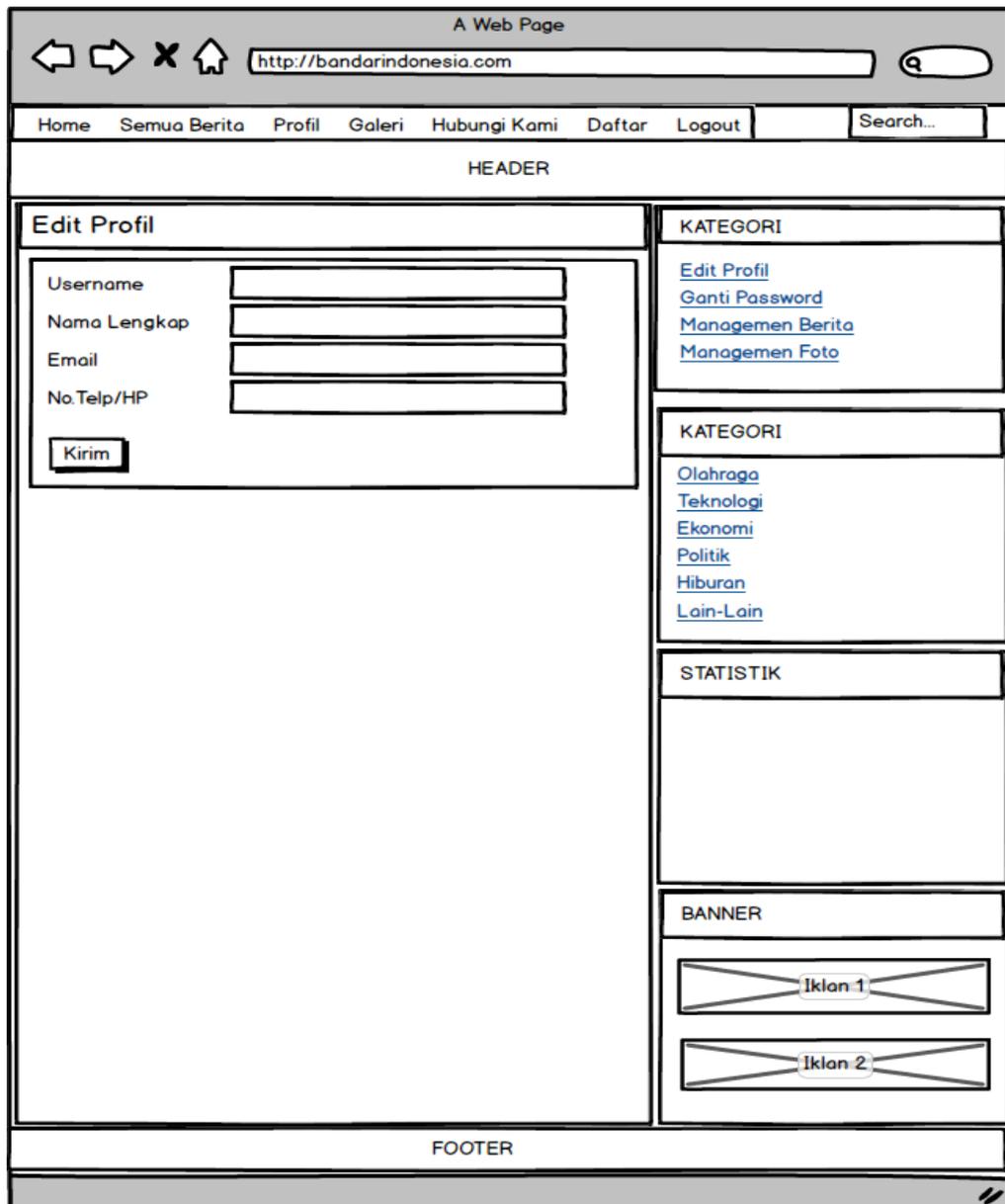
a. Rancangan halaman *login*



Gambar 3. 55 Rancangan halaman *login*

Rancangan halaman *login* hanya dapat berfungsi jika pengguna sudah terdaftar sebagai *member* yang secara otomatis memiliki *username* dan *password* yang diisi ketika mendaftar.

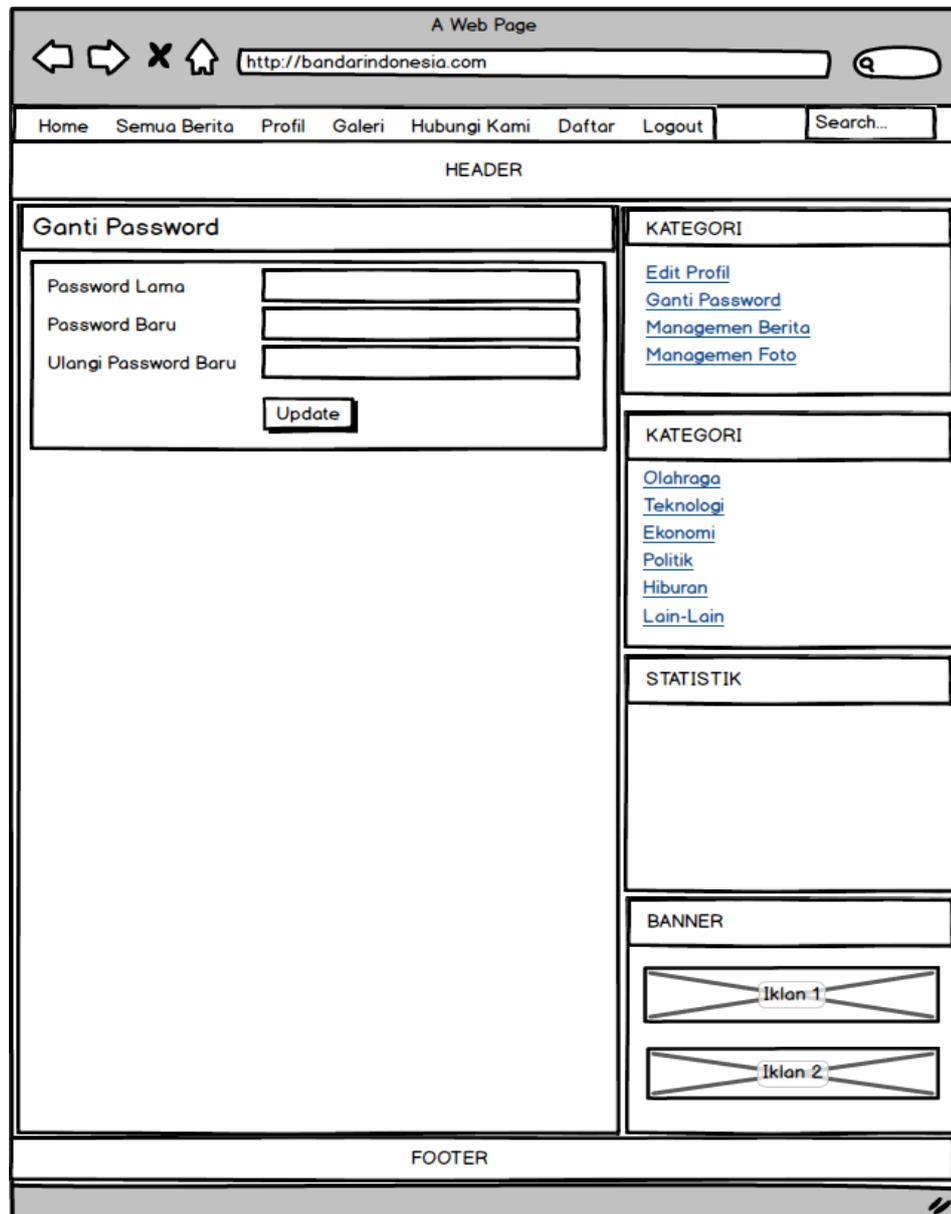
b. Rancangan halaman edit profil



Gambar 3. 56 Rancangan halaman edit profil

Halaman edit profil terdapat beberapa data yang dapat diperbarui paska pembuatan akun. Halaman edit profil dapat diakses dan tampil ketika *member* telah melakukan *login*.

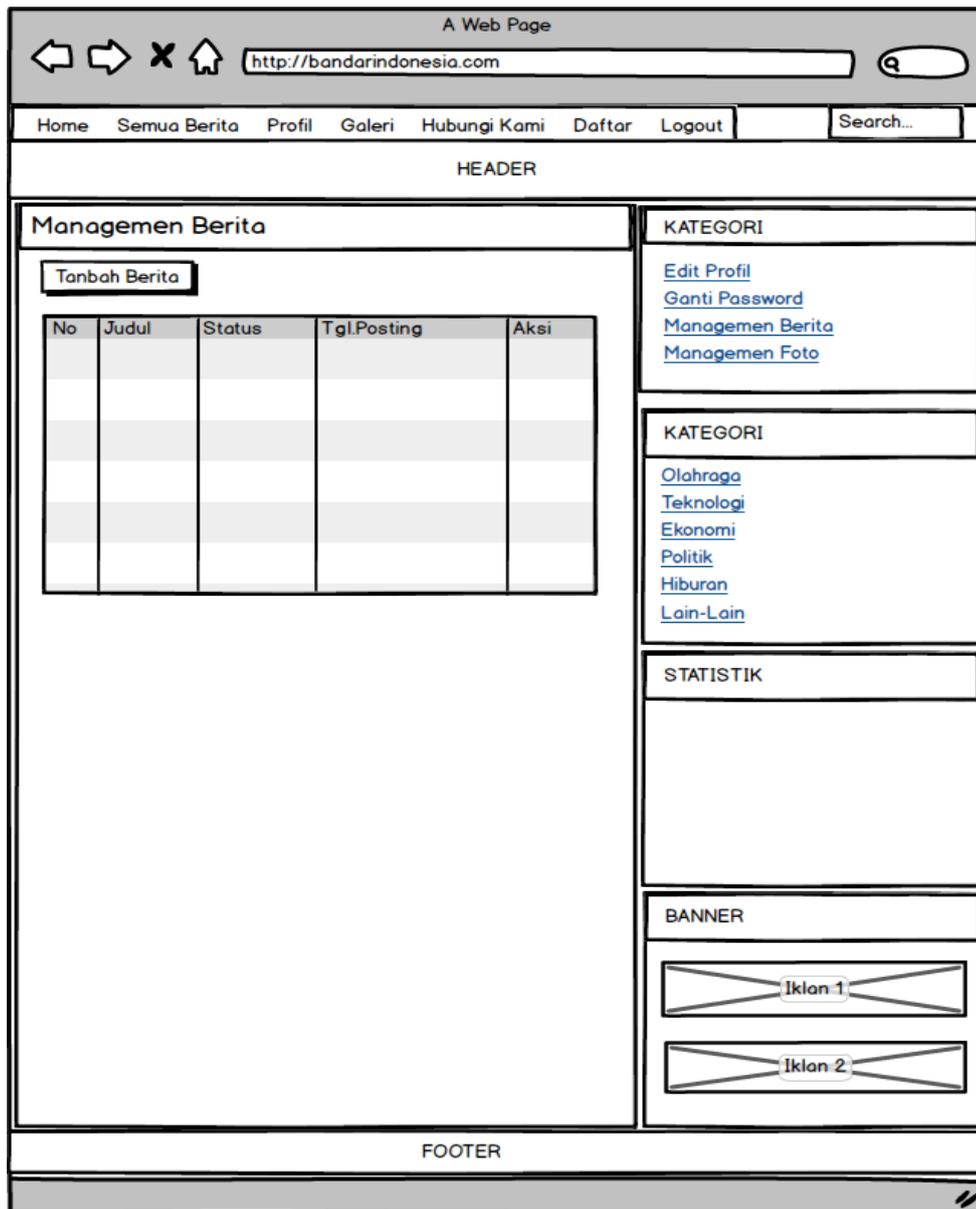
c. Rancangan halaman ganti *password*



Gambar 3. 57 Rancangan halaman ganti *password*

Untuk mengganti *password* yang lama, pengguna harus memasukkan *password* yang lama terlebih dahulu sebagai verifikasi bahwa pengguna tersebut adalah benar pemilih *username* terkait.

d. Rancangan halaman manajemen berita



Gambar 3. 58 Rancangan halaman manajemen berita

Dalam manajemen berita terdapat tabel berita dan *button* tambah berita. Tabel berita berisi berita yang pernah di *post* oleh *member*.

e. Rancangan halaman tambah berita

A Web Page
http://bandarindonesia.com

Home Semua Berita Profil Galeri Hubungi Kami Daftar Logout Search...

HEADER

Form Tambah Berita

Judul

Kategori

Isi Berita

Gambar

File

KATEGORI

[Edit Profil](#)
[Ganti Password](#)
[Managemen Berita](#)
[Managemen Foto](#)

KATEGORI

[Olahraga](#)
[Teknologi](#)
[Ekonomi](#)
[Politik](#)
[Hiburan](#)
[Lain-Lain](#)

STATISTIK

BANNER

Iklan 1

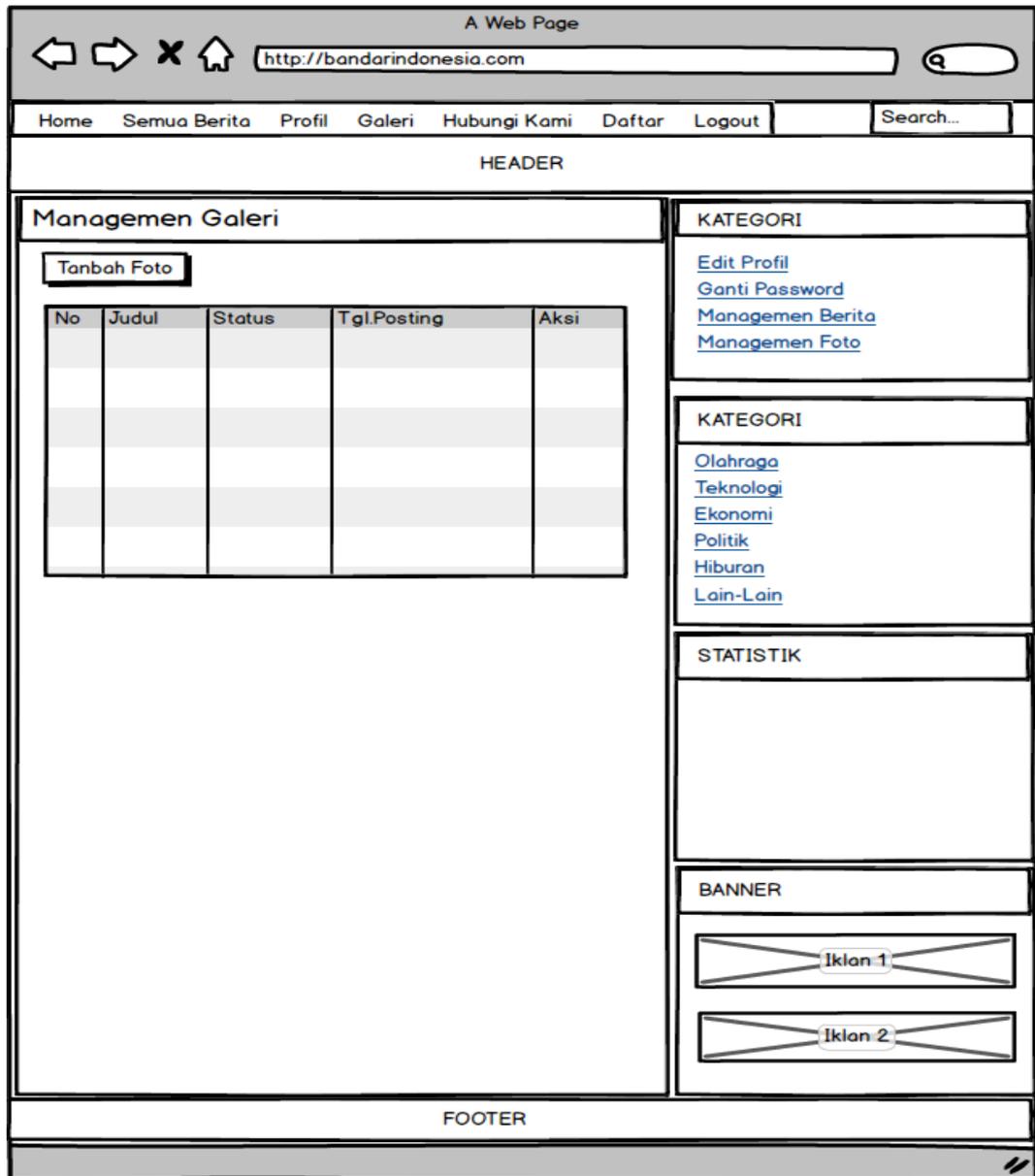
Iklan 2

FOOTER

Gambar 3. 59 Rancangan halaman tambah berita

Member dapat melakukan penambahan berita dan akan masuk kedalam tabel di halaman manajemen berita. Pada gambar 3.59 terlihat ada beberapa *button* pilih *file* yang berguna untuk mengambil data gambar dan *file* dari penyimpanan lokal.

f. Rancangan halaman manajemen *gallery*



Gambar 3. 60 Rancangan halaman manajemen *gallery*

Tidak ada bedanya dengan manajemen berita, manajemen galeri juga berisi data-data foto yang pernah di *upload* oleh pengguna terkait.

g. Rancangan halaman tambah foto

The image shows a wireframe of a web page for adding a photo. The page layout includes a header with navigation links and a search bar. The main content area is split into a form for adding a photo and a sidebar with various menu items and advertisements. The form has fields for title, image, and file, with buttons for selecting files, saving, and canceling. The sidebar lists categories, statistics, and banners.

Gambar 3. 61 Rancangan halaman tambah foto

Foto yang akan ditambahkan oleh pengguna harus diberi judul terlebih dahulu. *Member* dapat mengambil gambar langsung dari lokal. Terlihat pada gambar

3.61 terdapat *button* pilih *file* yang berarti memilih atau mengambil *file* dari *local storage*.

3.7 Metode Pengujian

Dalam pengujian aplikasi penulis menggunakan metode *blackbox*. Dimana aplikasi akan diuji fungsi pada setiap halaman. Penulis menggunakan metode *blackbox testing* untuk pengujian aplikasi dikarenakan dinilai memiliki beberapa keunggulan yaitu :

- Bisa memilih *subset test* secara efektif dan efisien.
- Dapat menemukan cacat atau *error*.
- Memaksimalkan *testing investmen*.

Pengujian aplikasi portal berita dengan sumber berita dua arah berbasis website menggunakan metode *blackbox* akan menguji semua halaman yang berada dalam aplikasi. Halaman-halaman yang akan diuji adalah :

<i>Website Utama</i>	<i>Website Pengelola</i>
- Halaman utama	- Halaman <i>login</i>
- Halaman semua berita	- Halaman utama
- Halaman kategori	- Halaman manajemen <i>user</i> (admin)
- Halaman isi berita	- Halaman manajemen <i>user</i> (<i>member</i>)
- Halaman tata cara	- Halaman manajemen berita (kategori)
- Halaman galeri	- Halaman manajemen berita (berita)
- Halaman hubungi	- Halaman manajemen berita (galeri)
- Halaman daftar	- Halaman manajemen berita (album)
- Halaman <i>search</i>	- Halaman manajemen <i>web</i> (hubungi)
- Halaman <i>login</i>	- Halaman manajemen <i>web</i> (komentar)
- Halaman edit profil	- Halaman manajemen <i>web</i> (<i>banner</i>)
- Halaman ganti <i>password</i>	- Halaman manajemen <i>web</i> (halaman statis)
- Halaman manajemen berita	
- Halaman tambah berita	

- Halaman manajemen galeri
- Halaman tambah foto galeri

Hasil pengujian aplikasi berupa tabel yang akan di jelaskan lebih detail pada bab IV. Sementara cara menyimpulkan hasil pengujian dengan mencoba setiap fungsi yang ada dalam aplikasi dan apabila semua fungsi aplikasi berfungsi dengan benar maka pengujian dikatakan berhasil.