Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III

METODE DAN PERANCANGAN APLIKASI

3.1 Metode Penelitian

Penelitian ini menggunakan metode-metode sebagai berikut.

1. Studi Pustaka

   Studi pustaka merupakan metode penelitian yang memanfaatkan referensi terhadap buku, jurnal, artikel, dan sumber lainnya yang tersedia baik secara *online* maupun *offline* dan berhubungan dengan penelitian yang sedang dilakukan.

2. Perancangan dan Implementasi Aplikasi

   Tahapan ini meliputi analisis terhadap *requirements*, tahapan jalannya aplikasi, dan prosedur apa saja yang harus dilakukan oleh aplikasi tersebut, serta bagaimana tampilan antarmuka aplikasinya. Rancangan ini dibuat dalam bentuk diagram alir sederhana yang menggambarkan jalannya aplikasi. Setelah perancangan selesai dilakukan, tahap selanjutnya adalah implementasi aplikasi menggunakan bahasa pemrograman yang fungsionalitasnya mendukung.

3. *Testing* dan *Debugging* Aplikasi

   Tahapan ini dilakukan beriringan (paralel) dengan tahap perancangan dan implementasi aplikasi. Saat aplikasi dibangun, tahap *testing* dan *debugging* juga dilakukan untuk memastikan fitur-fitur aplikasi yang dibangun berjalan sebagaimana mestinya.

4. Pengumpulan dan Analisis Data Pengujian

   Pengumpulan data untuk menguji tingkat efektivitas penelitian ini menggunakan alat bantu, yakni Google Analytics. Analisis data untuk penentuan keberhasilan penerapan metode penelitian ini didasarkan pada jumlah *pageview* situs.

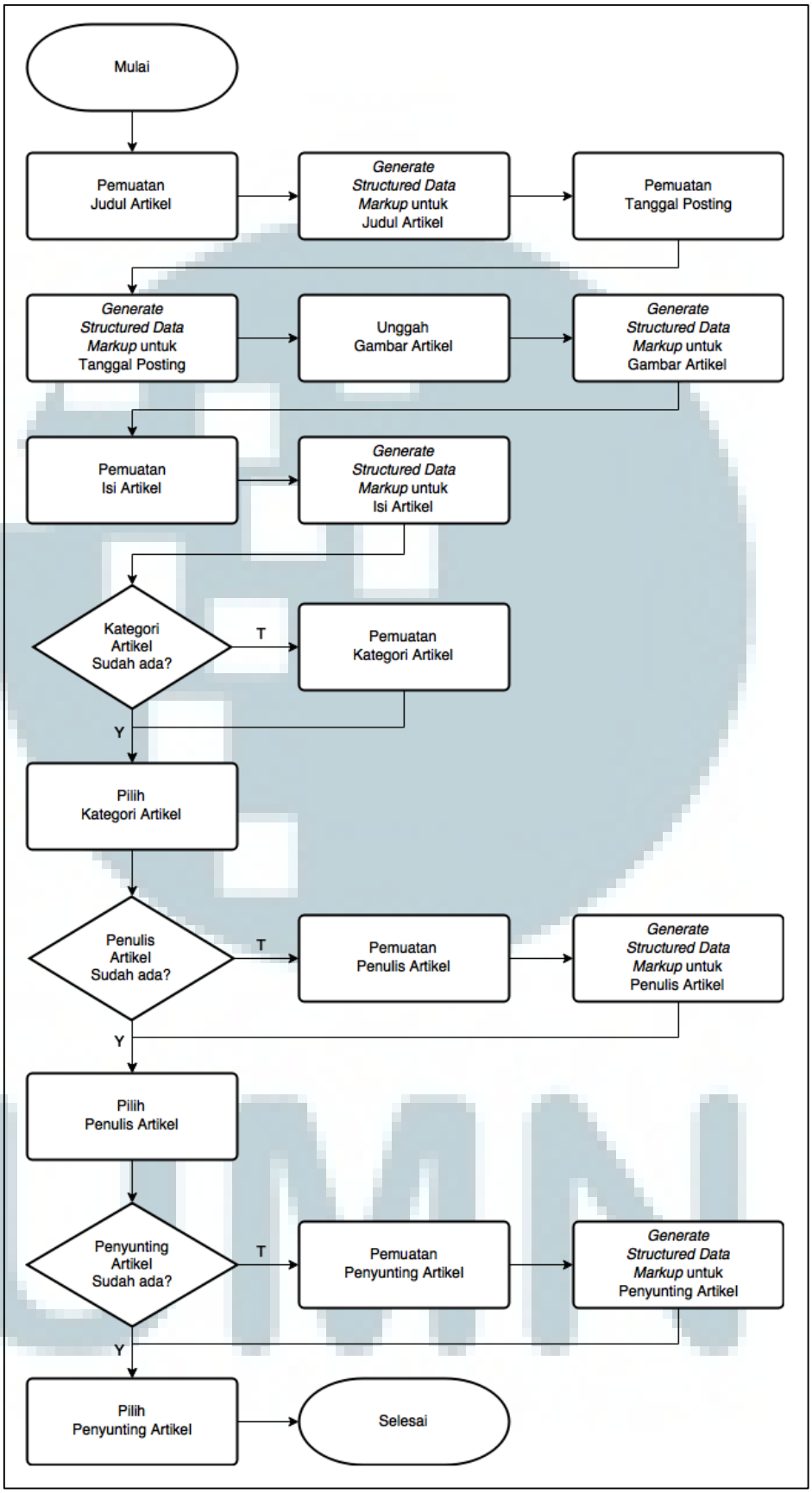
5. Penulisan Laporan

   Laporan mengenai hasil penelitian ditulis secara bertahap mulai dari studi pustaka hingga penarikan kesimpulan dari hasil pengumpulan dan analisis data penelitian. Laporan hasil penelitian ditulis berdasarkan sistematika yang ditentukan.

3.2 Rancangan Aplikasi

Secara garis besar, portal berita ini mengutamakan banyaknya jumlah *pageview* yang didapatkan situs ketika diukur menggunakan Google Analytics. Pada halaman *admin*, penulis berita mengisi konten portal seperti biasa, lalu sistem akan men-*generate structured data markup* untuk melakukan *Search Engine Optimization (SEO)*. Data yang dapat dimasukkan oleh *admin* adalah kategori, penulis berita, penyunting berita, dan artikelnya.

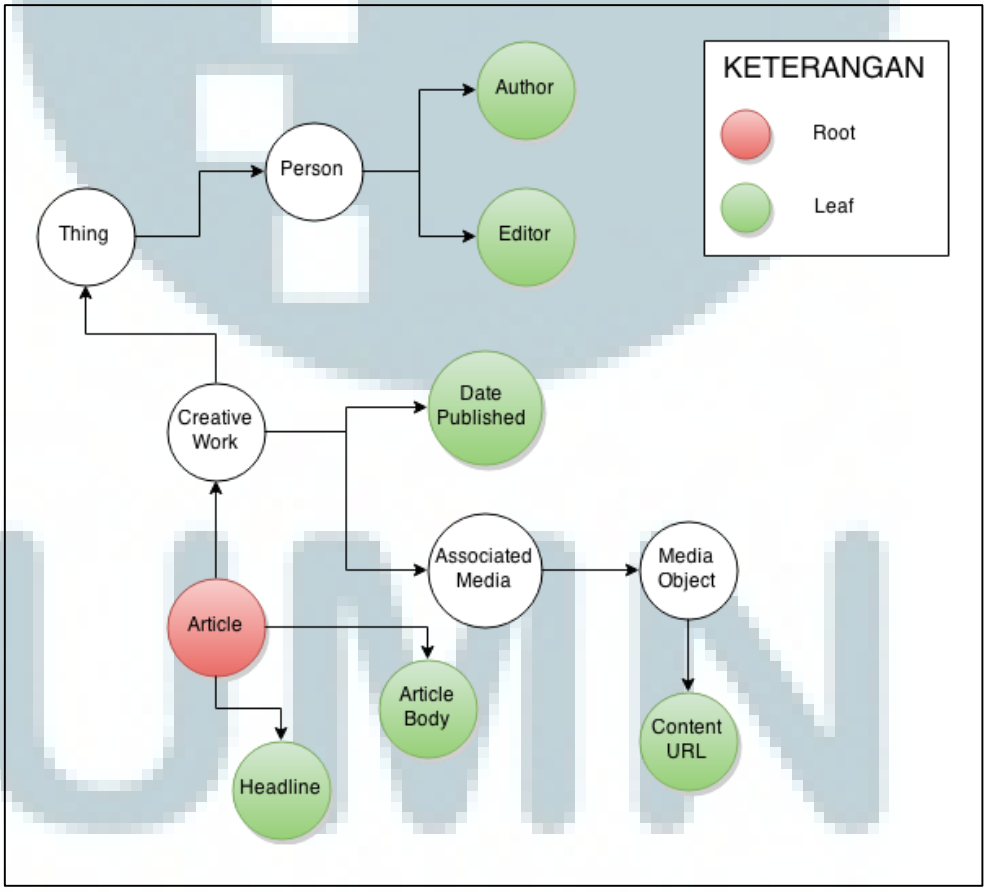
Diagram alir sederhana cara kerja *generator structured data markup* untuk portal ini dipaparkan pada gambar 3.1. Saat sistem mulai, maka judul artikel dimuat dan *structured data markup* untuk judul artikel akan di-*generate* oleh sistem. Demikian juga structured data markup untuk tanggal *posting*, gambar artikel, isi artikel, penulis artikel, dan penyunting artikel juga akan di-generate oleh sistem.



Gambar 3.1. Cara Kerja *Generator Structured Data Markup*

3.2.1 Pemilihan Skema untuk *Structured Data Markup*

Skema yang digunakan pada penelitian ini mengacu pada skema *data markup* yang dipublikasikan pada http://schema.org. Schema.org merupakan sebuah situs yang melakukan publikasi *vocabulary* yang digunakan untuk melakukan implementasi SEO menggunakan teknik *structured data markup.* Schema.org akan memberikan informasi yang dapat dimengerti oleh mesin ke laman, sehingga mesin pencari dapat meningkatkan performa pencarian (Patel-Schneider, 2014). Situs ini disponsori oleh empat mesin pencari terbesar, yakni Google, Microsoft, Yahoo, dan Yandex. Skema yang digunakan pada penelitian ini dipaparkan pada gambar 3.2 berikut.



Gambar 3.2. Skema yang Digunakan pada Portal Berita

Node berwarna merah merupakan *root* yang berperan merangkum dan menghubungkan semua atribut dalam sebuah berita, sedangkan *node* berwarna hijau menggambarkan properti yang akan dipakai pada *markup* untuk memberikan *metadata* pada konten.

Untuk memperjelas penerapan skema pada gambar 3.2, berikut merupakan sebuah berita yang sudah dibubuhi *metadata*. Pada elemen terluar, ditunjukkan bahwa tipe *item* merupakan *article*. Selanjutnya, *article* memiliki properti, seperti *headline*, *article body*, *date published*, *author*, *editor*, dan *content URL* yang berisi pranala gambar pada berita tersebut.

```
<div itemscope itemtype="http://schema.org/Article" class="blog-post">
    <h2 itemprop="headline" class="blog-post-title news-title">
        Dying Light Eksklusif Untuk Next-Gen Console
    </h2>
    <p class="blog-post-meta font-open-sans">
        <span itemprop="datePublished" content="2014-10-29">
            29 October 2014
        </span> oleh
        <a href="#"><span itemprop="author">elmer</span></a>
    </p>
    <div class="row">
        <div itemprop="associatedMedia" class="col-sm-12 col-md-12">
            <span itemscope itemtype="http://schema.org/MediaObject">
                <a href="#" class="thumbnail">
                    <img itemprop="contentURL"
                        style="width: 640px; height: 320px;"
                        src="http://cdn.inigame.com/wp-content/..."
                        alt="placeholder">
                </a>
            </span>
        </div>
    </div>
    <p itemprop="articleBody" class="news-content">
        Seperti yang kita ketahui bahwa game...
    </p>
    <hr>
    <p class="blog-post-meta font-open-sans">
        Editor: <a href="#"><span itemprop="editor">elmer</span></a>
    </p>
</div>
```

Gambar 3.3. *Markup* HTML5 dengan *Structured Data Markup*

Setelah berita divalidasi, maka hasil keluarannya terpapar pada gambar 3.4 berikut.

MediaObject (1) Tidak ada masalah

MediaObject	
contentUrl:	http://cdn.inigame.com/wp-content/uploads/2014/08/dying-light-inigame-cover.jpg

Article (1) Tidak ada masalah

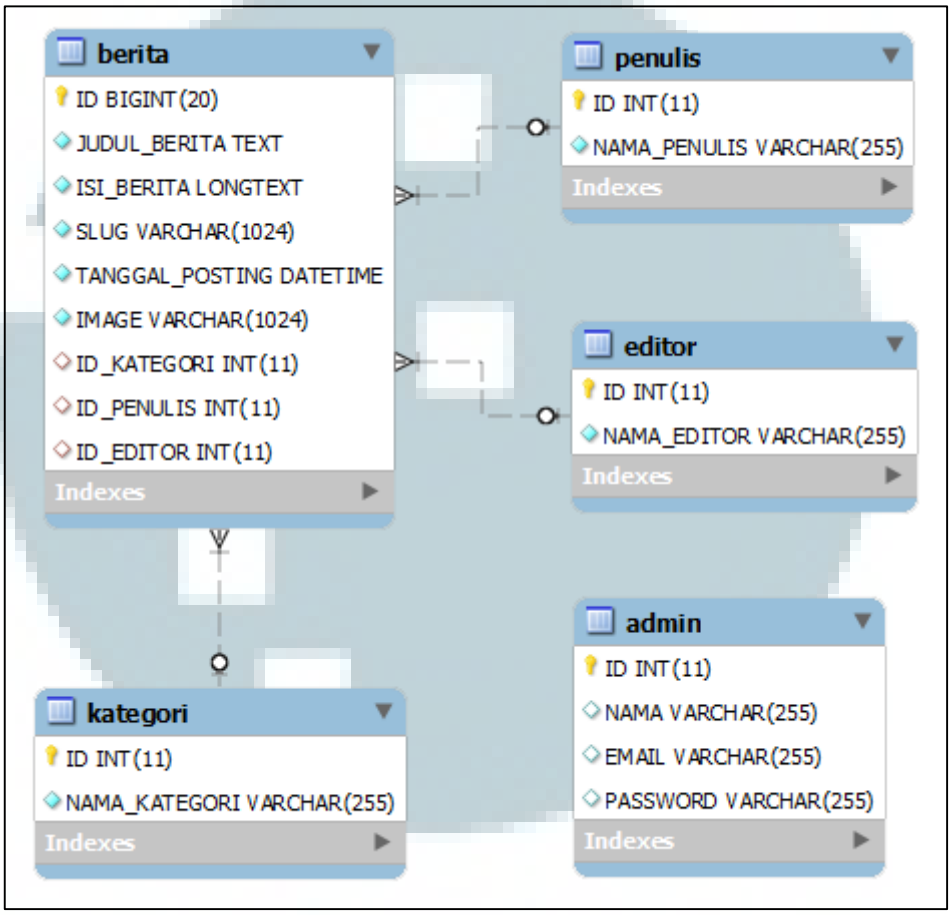
Article	
headline:	Dying Light Eksklusif Untuk Next-Gen Console
datePublished:	2014-10-29
articleBody:	Seperti yang kita ketahui bahwa game Dying Light merupakan salah satu game yang menarik yang ditunggu kehadirannya Januari 2015 mendatang. Bahkan game ini dikabarkan akan diluncurkan untuk last-gen console, next-gen console dan PC. Namun, sepertinya beberapa gamer harus mendengar kabar yang mengecewakan dari Dying Light kalau game ini tidak jadi diluncurkan untuk last-gen console. Game ini hanya akan dirilis untuk PlayStation 4, Xbox One dan PC. Beberapa alasan yang dikemukakan adalah bahwa teknologi dari last-gen console tidak mampu untuk mengeluarkan hasil yang ingin dicapai oleh developer seperti 200.000 object yang dapat ditampilkan secara langsung. Kemudian efek realistis yang didapat melalui physics-based lighting. Natural Movement yang memungkinkan karakter dalam game dapat bergerak seperti manusia aslinya dan masih banyak lagi teknologi untuk memaksimalkan game tersebut dan hanya bisa dicapai dengan next-gen console.
author [Thing]:	
name:	elmer
editor [Person]:	
name:	elmer

Gambar 3.4. Keluaran *Structured Data Markup* Setelah Divalidasi

3.2.2 Struktur Basis Data Aplikasi

Gambar 3.5 berikut merupakan *Entity Relationship Diagram* (ERD) dan penjelasannya dari portal berita ini. Terdapat 5 buat tabel pada basis data yang digunakan pada portal ini. Kelima tabel berikut adalah tabel penulis, editor, kategori, berita, dan admin. Tabel berita berelasi dengan tiga buah tabel, yaitu tabel penulis, tabel editor, dan tabel kategori. Tabel berita dan penulis direferensikan oleh

kolom ID_PENULIS pada tabel berita. Tabel berita dan kategori diferensikan oleh kolom ID_KATEGORI pada tabel berita. Tabel berita dan editor direferensikan oleh kolom ID_EDITOR pada tabel berita. Nama kolom, tipe data, dan fungsi masing-masing kolom dipaparkan pada penjelasan di bawah ini.



Gambar 3.5. *Entity Relationship Diagram* pada Portal Berita Guratpena

Tabel 3.1 merupakan struktur tabel kategori yang berfungsi menyimpan kategori berita pada basis data. Tabel ini terdiri dari 2 kolom, yakni ID dan NAMA_KATEGORI. Kolom ID merupakan *primary key* tabel ini dan menggunakan atribut *auto increment*.

Tabel 3.1. Struktur Tabel Kategori

Nama Kolom	Tipe Data	Fungsi
ID	INT	Menyimpan *primary key* dari tabel kategori.
NAMA_KATEGORI	VARCHAR	Menyimpan nama kategori.

Tabel 3.2 merupakan struktur tabel penulis yang berfungsi menyimpan penulis berita pada basis data. Tabel ini terdiri dari 2 kolom, yakni ID dan NAMA_PENULIS. Kolom ID merupakan *primary key* tabel ini dan menggunakan atribut *auto increment*.

Tabel 3.2. Struktur Tabel Penulis

Nama Kolom	Tipe Data	Fungsi
ID	INT	Menyimpan *primary key* dari tabel penulis.
NAMA_PENULIS	VARCHAR	Menyimpan nama penulis.

Tabel 3.3 merupakan struktur tabel editor yang berfungsi menyimpan penyunting *(editor)* berita pada basis data. Tabel ini terdiri dari 2 kolom, yakni ID dan NAMA_EDITOR. Kolom ID merupakan *primary key* tabel ini dan menggunakan atribut *auto increment*.

Tabel 3.3. Struktur Tabel Editor

Nama Kolom	Tipe Data	Fungsi
ID	INT	Menyimpan *primary key* dari tabel *editor*.
NAMA_EDITOR	VARCHAR	Menyimpan nama penyunting *(editor)*.

Tabel 3.4 merupakan struktur tabel berita yang berfungsi menyimpan berita pada basis data. Tabel ini terdiri dari 9 kolom, yakni ID, JUDUL_BERITA, ISI_BERITA, SLUG, TANGGAL_POSTING, IMAGE, ID_KATEGORI, ID_PENULIS dan ID_EDITOR. Kolom ID merupakan *primary key* tabel ini dan

menggunakan atribut *auto increment*. Kolom ID_KATEGORI, ID_PENULIS, dan ID_EDITOR merupakan *foreign key* untuk tabel kategori, penulis dan editor.

Tabel 3.4. Struktur Tabel Berita

Nama Kolom	Tipe Data	Fungsi
ID	BIGINT	Menyimpan *primary key* dari tabel berita.
JUDUL_BERITA	TEXT	Menyimpan judul berita.
ISI_BERITA	LONGTEXT	Menyimpan isi berita.
SLUG	VARCHAR	Menyimpan *slug* (URL) berita.
TANGGAL_POSTING	DATETIME	Menyimpan tanggal dan waktu berita dipublikasikan.
IMAGE	VARCHAR	Menyimpan *path image* berita yang akan ditampilkan.
ID_KATEGORI	INT	Menyimpan *foreign key* yang direferensikan dengan tabel "kategori".
ID_PENULIS	INT	Menyimpan *foreign key* yang direferensikan dengan tabel "penulis".
ID_EDITOR	INT	Menyimpan *foreign key* yang direferensikan dengan tabel "editor".

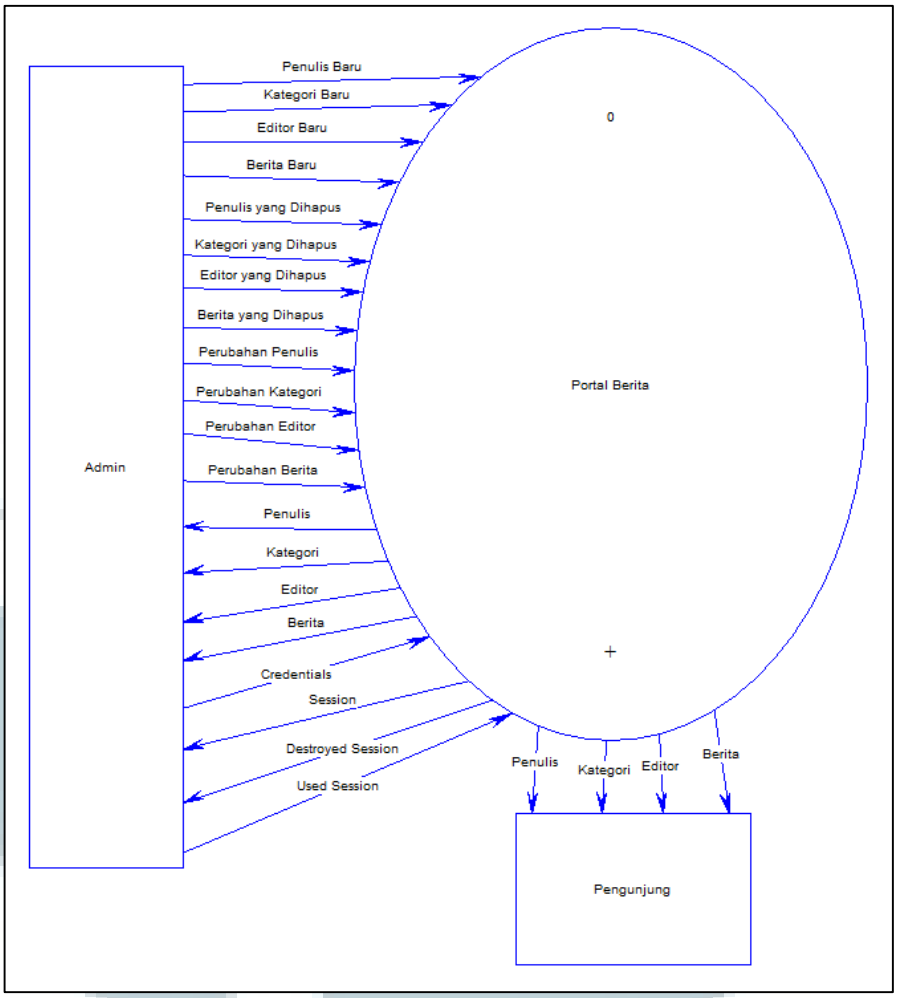
Tabel 3.5 merupakan struktur tabel admin yang berfungsi menyimpan identitas dan *credential admin* pada basis data. Tabel ini terdiri dari 4 kolom, yakni ID, NAMA, EMAIL dan PASSWORD. Kolom ID merupakan *primary key* tabel ini dan menggunakan atribut *auto increment*. Kolom EMAIL dan PASSWORD berfungsi menyimpan *credential admin* untuk melakukan login pada *back office* portal berita.

Tabel 3.5. Struktur Tabel Admin

Nama Kolom	Tipe Data	Fungsi
ID	INT	Menyimpan *primary key* dari tabel *admin*.
NAMA	VARCHAR	Menyimpan nama *admin*.
EMAIL	VARCHAR	Menyimpan email *admin*.
PASSWORD	VARCHAR	Menyimpan sandi *(password) admin* yang digunakan untuk melakukan login.

3.2.3 Alur Aplikasi

Dalam portal berita ini, ada dua entitas yang berperan penting mendukung jalannya sistem. Kedua entitas tersebut adalah *admin* dan pengunjung situs. *Admin* memiliki peran untuk melakukan manajemen data pada portal berita, sedangkan pengunjung situs berperan melihat berita yang dipublikasikan pada portal. Gambar 3.6 merupakan *context diagram* dari *Data Flow Diagram (DFD)* untuk penggambaran alur sistem secara keseluruhan.



Gambar 3.6. *Context Diagram* Portal Berita

Terdapat dua entitas gambar 3.6, yakni *admin* dan pengunjung. Entitas *admin* dapat memberikan data ke dalam sistem, sedangkan pengunjung hanya dapat menerima data dari sistem. Masukan data yang dilakukan *admin* adalah sebagai berikut

1. Penulis baru, merupakan data penulis yang dimasukkan oleh *admin* ke dalam sistem. Data penulis yang dimasukkan adalah nama penulis.
2. Kategori baru, merupakan data kategori yang dimasukkan oleh *admin* ke dalam sistem. Data kategori yang dimasukkan adalah nama kategori.

3. *Editor* baru, merupakan data penyunting berita yang dimasukkan oleh *admin* ke dalam sistem. Data penyunting yang dimasukkan adalah nama penyunting berita.
4. Berita baru, merupakan data berita yang dimasukkan oleh *admin* ke dalam sistem. Data berita yang dimasukkan meliputi judul berita, isi berita, gambar berita, dan tanggal posting.
5. Penulis yang dihapus, merupakan data penulis yang akan dihapus dari sistem. *Admin* mengirimkan ID penulis yang akan dihapus dari sistem.
6. Kategori yang dihapus, merupakan data kategori yang akan dihapus dari sistem. *Admin* mengirimkan ID kategori yang akan dihapus dari sistem.
7. *Editor* yang dihapus, merupakan data penyunting berita yang akan dihapus dari sistem. *Admin* mengirimkan ID penyunting yang akan dihapus dari sistem.
8. Berita yang dihapus, merupakan berita yang akan dihapus dari sistem. *Admin* mengirimkan ID berita yang akan dihapus dari sistem.
9. Perubahan penulis, merupakan data penulis yang akan diubah pada sistem. Data yang dikirimkan *admin* pada sistem meliputi ID penulis dan nama penulis yang baru.
10. Perubahan kategori, merupakan data kategori yang akan diubah pada sistem. Data yang dikirimkan *admin* pada sistem meliputi ID kategori dan nama kategori yang baru.
11. Perubahan editor, merupakan data penyunting berita yang akan diubah pada sistem. Data yang dikirimkan *admin* pada sistem meliputi ID penyunting berita dan nama penyunting berita yang baru.

12. Perubahan berita, merupakan detail berita yang akan diubah pada sistem. Data yang dikirimkan *admin* pada sistem meliputi ID berita dan detail berita yang baru. Detail berita tersebut terdiri dari judul berita yang baru, isi berita yang baru, penulis yang baru, penyunting berita yang baru, dan gambar berita yang baru.
13. *Credentials*, merupakan data *credentials* yang dikirimkan *admin* untuk melakukan manajemen portal berita. Data *credentials* ini meliputi email dan kata sandi.
14. *Used session*, merupakan *session* yang sedang dipakai oleh *admin* untuk melakukan manajemen berita. *Admin* perlu mendapatkan *session* untuk masuk ke sistem dan melakukan tugasnya.

Keluaran data yang didapatkan oleh *admin* adalah sebagai berikut

1. Penulis, merupakan data penulis yang dapat dilihat oleh *admin*. Data penulis yang dapat dilihat oleh *admin* terdiri dari nama penulis.
2. Kategori, merupakan data penulis yang dapat dilihat oleh *admin*. Data kategori yang dapat dilihat oleh *admin* terdiri dari nama kategori.
3. Editor, merupakan data penyunting berita yang dapat dilihat oleh *admin*. Data penyunting berita yang dapat dilihat oleh *admin* terdiri dari nama penyunting berita.
4. Berita, merupakan detail berita yang dapat dilihat oleh *admin*. Detail berita yang dapat dilihat oleh *admin* terdiri dari judul berita, isi berita, gambar berita, penulis berita, penyunting berita, dan tanggal posting berita.

5. *Session*, merupakan *session* yang didapatkan oleh *admin* ketika ia dapat dengan tepat memasukkan *credentials* ke dalam sistem untuk dapat melakukan manajemen data di dalamnya.
6. *Destroyed session*, merupakan *session* yang telah dihilangkan oleh sistem karena *admin* melakukan *logout* dari sistem.

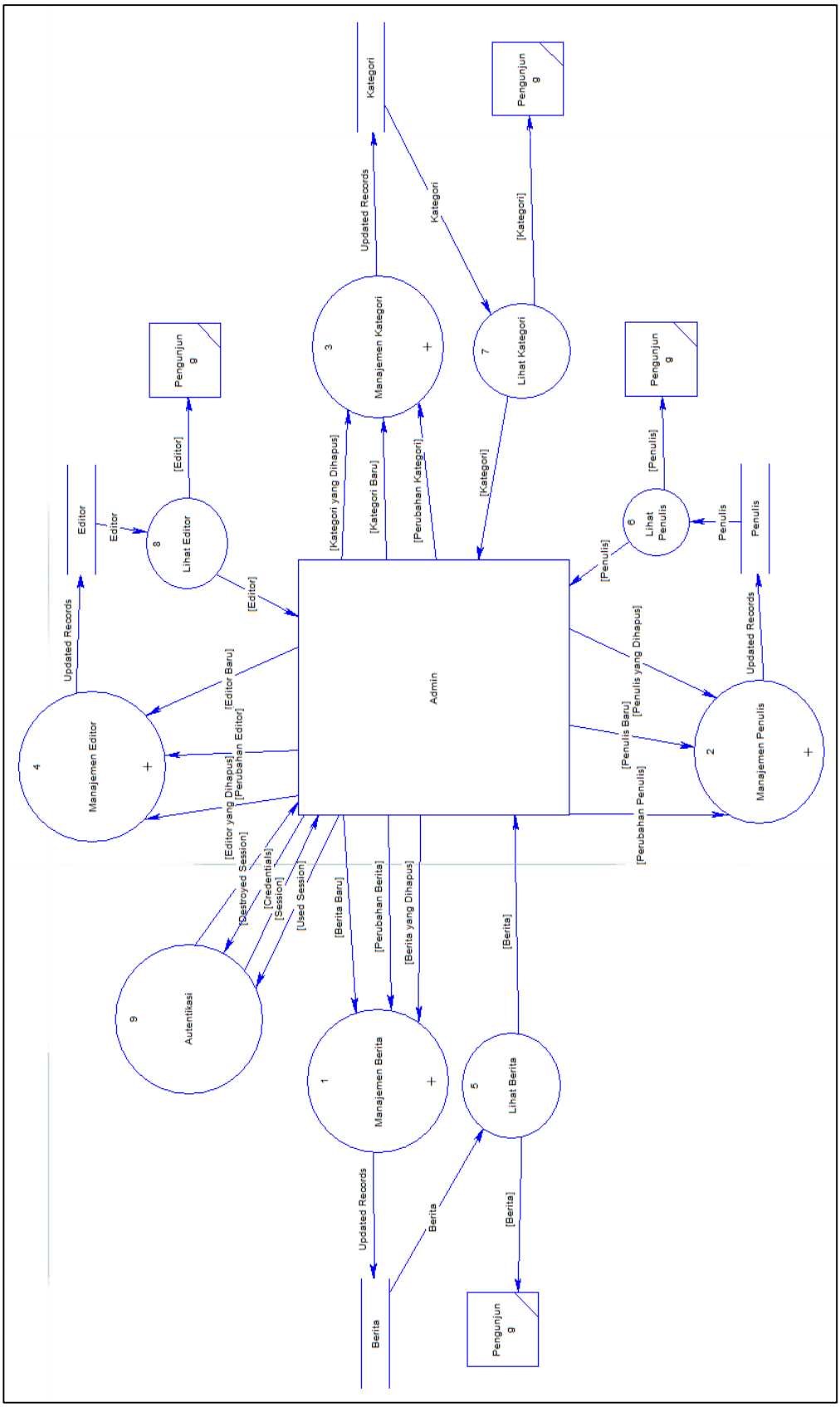
Entitas pengunjung memiliki hak untuk mendapatkan empat keluaran dari sistem, yakni penulis, kategori, editor, dan berita. Keluaran ini merupakan data yang sama dengan data yang diterima oleh *admin* pada nomor 1, 2, 3, dan 4.

Context diagram pada gambar 3.6 dapat diperdalam lagi menjadi diagram seperti pada gambar 3.7. Portal berita pada *context diagram* dijabarkan menjadi 9 proses yang berinteraksi dengan *admin* dan pengunjung. Sembilan proses tersebut adalah sebagai berikut.

1. Manajemen berita, merupakan proses dimana *admin* melakukan penambahan berita, pengubahan berita, dan penghapusan berita. Proses manajemen berita juga memiliki alur data menuju *data store* berita. Data yang disimpan berupa *updated records*, yakni *record* basis data yang berhasil diperbaharui. Pembaharuan yang dimaksud dapat berupa penambahan jumlah *record* akibat penambahan data baru, perubahan nilai *record*, maupun pengurangan jumlah *record* akibat penghapusan.
2. Manajemen penulis, merupakan proses dimana admin melakukan penambahan penulis, pengubahan penulis, dan penghapusan penulis. Proses manajemen penulis juga memiliki alur data menuju *data store* penulis. Data yang disimpan berupa *updated records*, yakni *record* basis data yang berhasil diperbaharui.

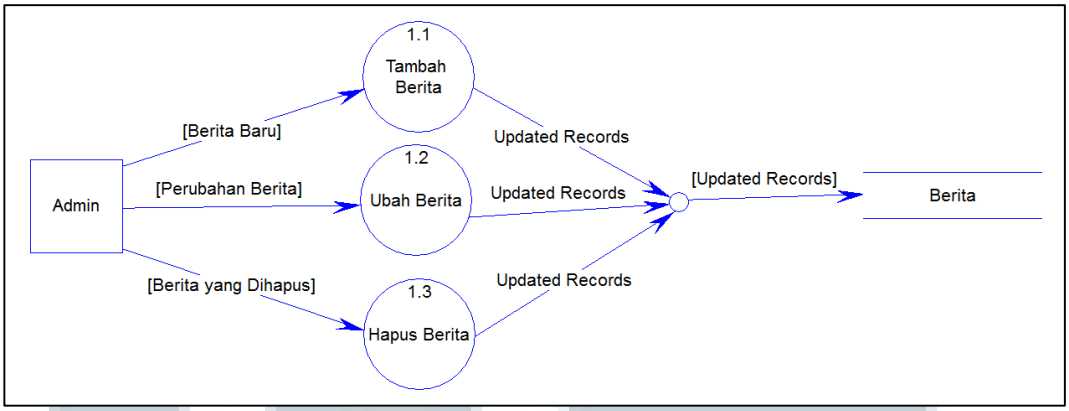
3. Manajemen kategori, merupakan proses dimana *admin* melakukan penambahan kategori, pengubahan kategori, dan penghapusan kategori. Proses manajemen kategori juga memiliki alur data menuju *data store* kategori. Data yang disimpan berupa *updated records*, yakni *record* basis data yang berhasil diperbaharui.
4. Manajemen editor, merupakan proses dimana *admin* melakukan penambahan penyunting berita, pengubahan penyunting berita, dan penghapusan penyunting berita. Proses manajemen editor juga memiliki alur data menuju *data store* editor. Data yang disimpan berupa *updated records*, yakni *record* basis data yang berhasil diperbaharui.
5. Lihat berita, merupakan proses untuk melakukan pengambilan berita dari *data store* berita sehingga berita yang diambil dapat dilihat oleh *admin* dan pengunjung. Masukan dari proses ini merupakan berita dari *data store* berita dan memiliki keluaran berita yang mengalir pada admin dan pengunjung.
6. Lihat penulis, merupakan proses untuk melakukan pengambilan data penulis dari *data store* penulis sehingga data penulis yang diambil dapat dilihat oleh admin dan pengunjung. Masukkan dari proses ini merupakan berita dari *data store* penulis dan memiliki keluaran data penulis yang mengalir pada *admin* dan pengunjung.
7. Lihat kategori, merupakan proses untuk melakukan pengambilan data kategori dari *data store* kategori sehingga data kategori yang diambil dapat dilihat oleh *admin* dan pengunjung. Masukan dari proses ini merupakan kategori dari *data store* kategori dan memiliki keluaran data kategori yang mengalir pada *admin* dan pengunjung.

8. Lihat editor, merupakan proses untuk melakukan pengambilan data penyunting berita dari *data store* editor sehingga data penyunting yang diambil dapat dilihat oleh *admin* dan pengunjung. Masukan dari proses ini merupakan data penyunting dari *data store* editor dan memiliki keluaran data penyunting berita yang mengalir pada *admin* dan pengunjung.
9. Autentikasi, merupakan proses yang hanya melibatkan *admin*. Proses ini bertugas menerima masukkan berupa *credentials admin* dan memberikan *session* kembali kepada *admin*. *Admin* hanya bisa memasuki *back office* portal berita ketika berhasil mendapatkan *session* dari proses *login*. Setelah admin selesai bertugas, maka ia akan keluar dari halaman *back office* dengan memberikan *used session* (*session* yang sudah dipakai) sebagai masukkan dari proses autentikasi, lalu proses autentikasi akan memberikan *destroyed session* (*session* yang sudah dihancurkan dan tidak dapat digunakan). Setelah melalui proses ini, admin tidak dapat melakukan aktivitas kembali pada halaman *backoffice.*



Gambar 3.7. Level 1 *Data Flow Diagram* Portal Berita

Proses dengan nomor 1, 2, 3 dan 4 dapat dijabarkan lagi ke *level* dua *data flow diagram*. Setiap proses dapat dijabarkan menjadi tiga proses. Gambar 3.8 menunjukkan penjabaran proses nomor 1.

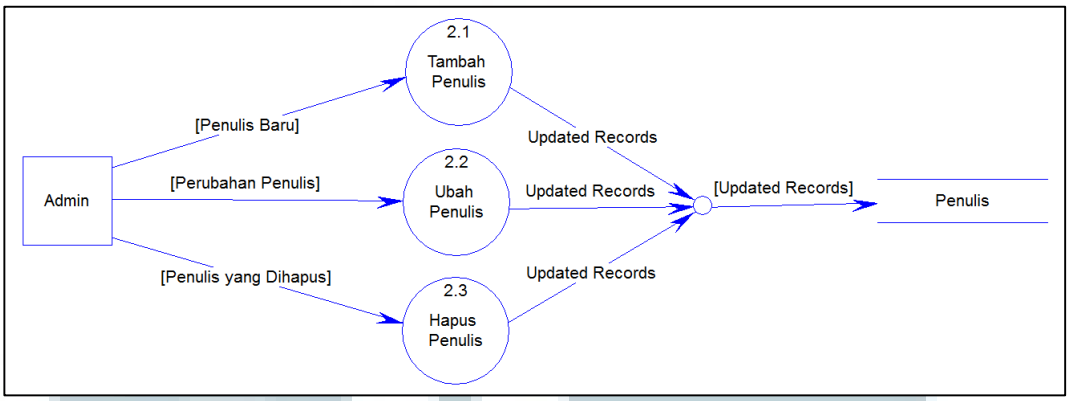


Gambar 3.8. Level 2 DFD Proses Manajemen Berita (Proses 1)

Pada gambar 3.8, proses manajemen berita dijabarkan menjadi tiga subproses. Ketiga subproses tersebut adalah

1. Proses tambah berita, berperan menambahkan berita baru ke dalam *data store* berita. Proses ini menerima masukan berupa berita baru dan menghasilkan keluaran *updated records*.
2. Proses ubah berita, berperan melakukan perubahan berita yang telah tersimpan di dalam *data store* berita dan menyimpan perubahannya ke dalam *data store*. Proses ini menerima masukan berupa perubahan berita dari *admin*, lalu menghasilkan keluaran *updated records* untuk disimpan.
3. Proses hapus berita, berperan melakukan penghapusan berita yang telah tersimpan di dalam *data store* berita. Proses ini menerima masukan berupa berita yang akan dihapus dan menghasilkan keluaran berupa *updated records*, yakni *records* baru tanpa mengandung data yang telah dihapus.

Gambar 3.9 merupakan penjabaran dari proses manajemen penulis (proses 2). Proses manajemen penulis dibagi menjadi tiga buah subproses seperti pada gambar 3.9.

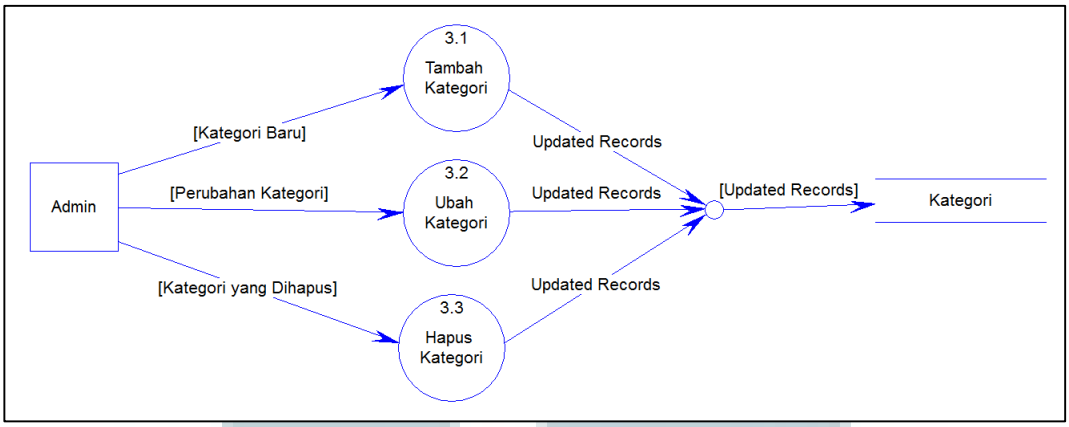


Gambar 3.9. Level 2 DFD Proses Manajemen Penulis (Proses 2)

Ketiga subproses yang terdapat pada proses manajemen penulis adalah

1. Proses tambah penulis, berperan menambahkan penulis baru ke dalam *data store* penulis. Proses ini menerima masukan berupa penulis baru dan menghasilkan keluaran *updated records*.
2. Proses ubah penulis, berperan melakukan perubahan penulis yang telah tersimpan di dalam *data store* penulis dan menyimpan perubahannya ke dalam *data store*. Proses ini menerima masukan berupa perubahan penulis dari *admin*, lalu menghasilkan keluaran *updated records* untuk disimpan.
3. Proses hapus penulis, berperan melakukan penghapusan penulis yang telah tersimpan di dalam *data store* berita. Proses ini menerima masukan berupa penulis yang akan dihapus dan menghasilkan keluaran berupa *updated records*, yakni *records* baru tanpa mengandung data yang telah dihapus.

Gambar 3.10 merupakan penjabaran dari proses manajemen kategori (proses 3). Proses manajemen kategori dibagi menjadi tiga buah subproses seperti pada gambar 3.10.

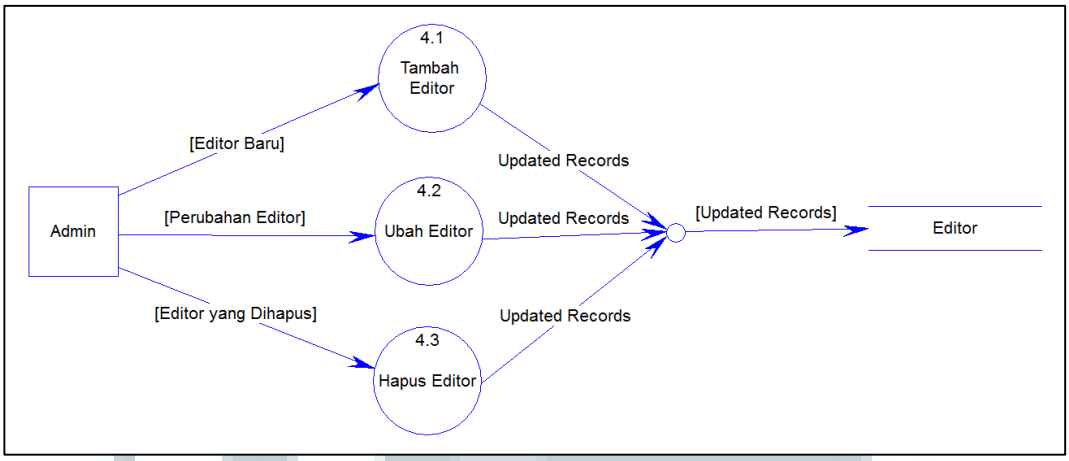


Gambar 3.10. Level 2 DFD Proses Manajemen Kategori (Proses 3)

Ketiga subproses yang terdapat pada proses manajemen kategori adalah

1. Proses tambah kategori, berperan menambahkan kategori baru ke dalam *data store* kategori. Proses ini menerima masukan berupa kategori baru dan menghasilkan keluaran *updated records*.
2. Proses ubah kategori, berperan melakukan perubahan kategori yang telah tersimpan di dalam *data store* kategori dan menyimpan perubahannya ke dalam *data store*. Proses ini menerima masukan berupa perubahan kategori dari *admin*, lalu menghasilkan keluaran *updated records* untuk disimpan.
3. Proses hapus kategori, berperan melakukan penghapusan kategori yang telah tersimpan di dalam *data store* berita. Proses ini menerima masukan berupa kategori yang akan dihapus dan menghasilkan keluaran berupa *updated records*, yakni *records* baru tanpa mengandung data yang telah dihapus.

Gambar 3.11 merupakan penjabaran dari proses manajemen editor (proses 4). Proses manajemen editor dibagi menjadi tiga buah subproses seperti pada gambar 3.11.



Gambar 3.11. Level 2 DFD Proses Manajemen *Editor* (Proses 4)

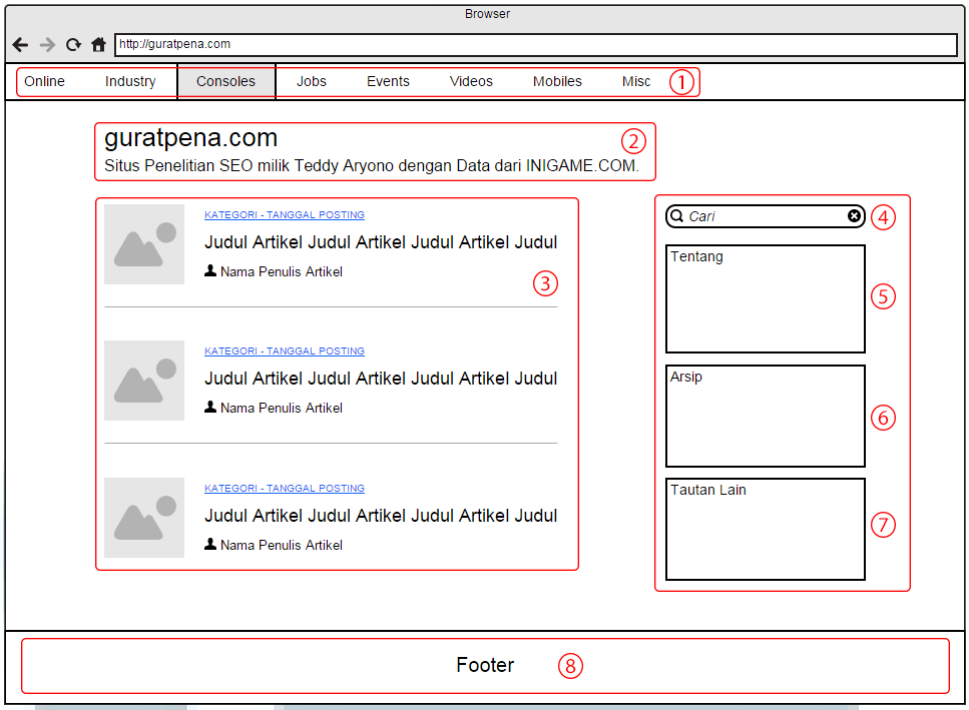
Ketiga subproses yang terdapat pada proses manajemen *editor* adalah

1. Proses tambah *editor*, berperan menambahkan *editor* baru ke dalam *data store* editor. Proses ini menerima masukan berupa *editor* baru dan menghasilkan keluaran *updated records*.
2. Proses ubah *editor*, berperan melakukan perubahan penyunting berita yang telah tersimpan di dalam *data store editor* dan menyimpan perubahannya ke dalam *data store*. Proses ini menerima masukan berupa perubahan *editor* dari *admin*, lalu menghasilkan keluaran *updated records* untuk disimpan.
3. Proses hapus *editor*, berperan melakukan penghapusan penyunting berita yang telah tersimpan di dalam *data store editor*. Proses ini menerima masukan berupa *editor* yang akan dihapus dan menghasilkan keluaran berupa *updated records*, yakni *records* baru tanpa mengandung data yang telah dihapus.

3.3 Rancangan Tampilan Antarmuka Aplikasi

Tampilan antarmuka pada portal berita ini dibagi menjadi 2 secara umum, yakni tampilan untuk pengunjung situs dan tampilan untuk administrator seperti halnya

Content Management System (CMS) pada umumnya. Rancangan tampilan halaman beranda untuk pengunjung ditunjukkan oleh gambar 3.12.



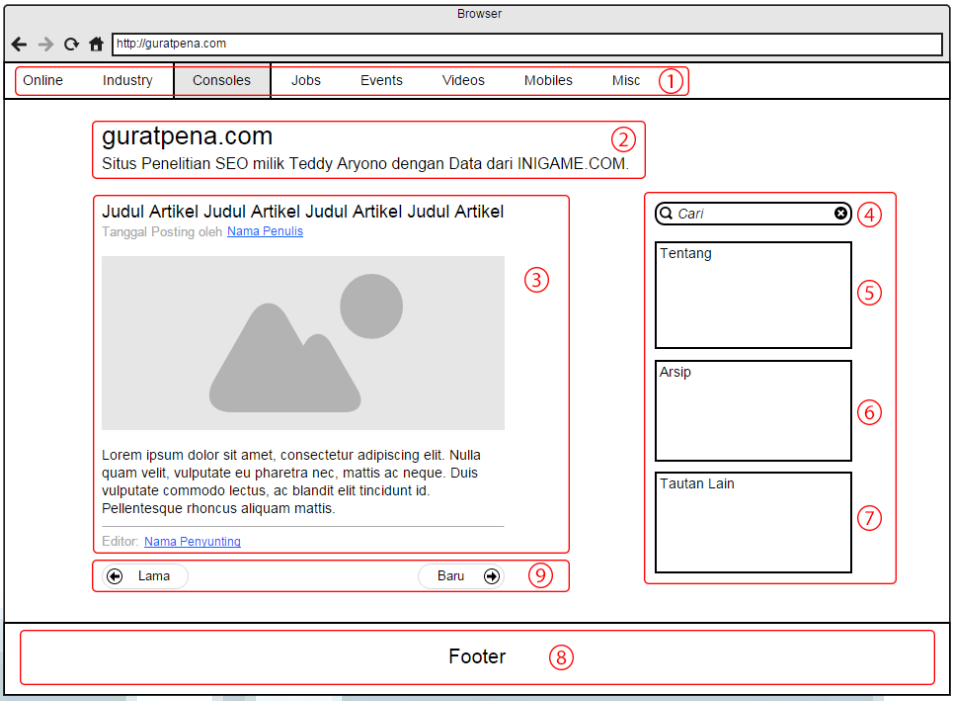
Gambar 3.12. Rancangan Layar Beranda pada Pengunjung

Secara garis besar, berita pada tampilan diatur dengan menggunakan *lazy load*, sehingga pengunjung bisa melihat dan menelusuri berita dengan terus melakukan *scrolling* pada layar. Berita diurutkan berdasarkan waktu postingnya. Tampilan ini memiliki komponen-komponen seperti yang ditunjukkan dengan angka yang dilingkari pada gambar. Berikut merupakan penjelasan komponen-komponen pada tampilan.

1. Baris navigasi. Tiap pengunjung situs bisa melihat berita berdasarkan kategori yang dipilih di navigasi ini.
2. Baris judul, berisikan judul dan *tagline* situs.
3. Daftar berita yang disusun menurut *layout* yang digambarkan. Dalam *layout* ini, tiap berita memiliki atribut: gambar kategori berita, tanggal posting berita, judul berita, dan penulis berita.

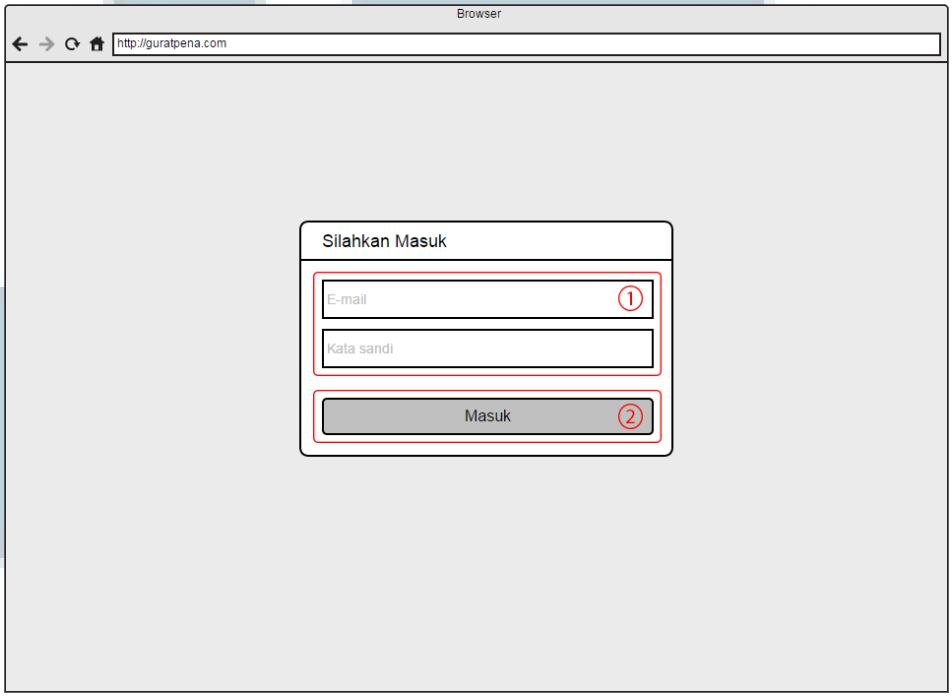
4. Kotak pencarian. Tiap pengunjung bisa mencari berita melalui kotak pencarian ini.
5. Kotak tentang, berisikan deskripsi situs secara umum.
6. Kotak arsip, berisikan kumpulan berita-berita lama yang dapat dibaca oleh pengunjung situs. Kumpulan berita ini dikelompokkan per bulan dan diurutkan berdasarkan waktu postingnya.
7. Kotak ke tautan lain, berisikan tautan-tautan yang berkaitan dengan situs.
8. *Footer*, berisikan *footer* situs.

Gambar 3.13 merupakan rancangan tampilan ketika pengunjung membaca berita yang dikehendakinya. Komponen 1, 2, 4, 5, 6, 7 dan 8 sama dengan tampilan pada gambar 3.12, namun pada komponen 3, ditampilkan berita secara detail. Atribut berita yang ditampilkan antara lain judul berita, tanggal posting, nama penulis artikel, gambar artikel, isi artikel, dan nama penyunting artikel. Selain itu, pada komponen 9, ada navigasi berbentuk anak panah yang memudahkan pengunjung untuk berpindah ke artikel sebelumnya atau artikel sesudahnya.



Gambar 3.13. Rancangan Layar Detail Berita pada Pengunjung

Selain pengunjung, situs ini juga digunakan oleh *admin* yang bertugas melakukan manajemen portal berita, antara lain melakukan manajemen penulis, penyunting, dan berita. Tampilan untuk *login admin* ditampilkan pada gambar 3.14.

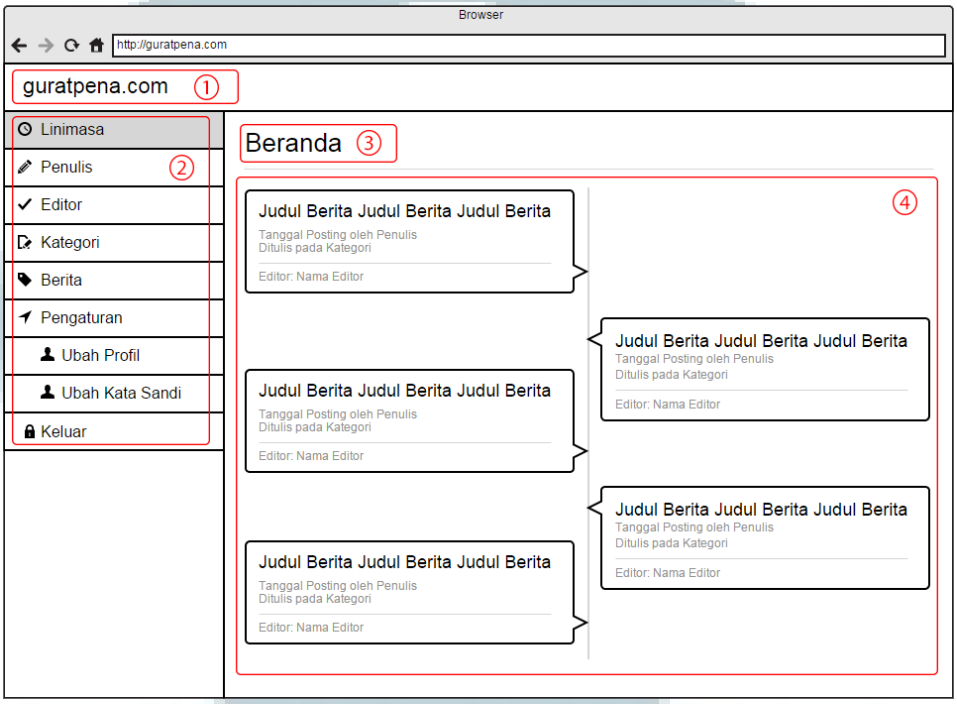


Gambar 3.14. Rancangan Layar *Login* pada *Admin*

Berikut merupakan penjelasan komponen-komponen pada gambar 3.14.

1. Kolom input untuk *email* dan *password*.
2. Tombol untuk melakukan *login*.

Setelah *admin* melakukan *login*, halaman beranda *admin* akan ditampilkan. Pada halaman ini, *admin* dapat melihat delapan berita terbaru yang ditampilkan dalam format linimasa dari berita terbaru ke terlama.



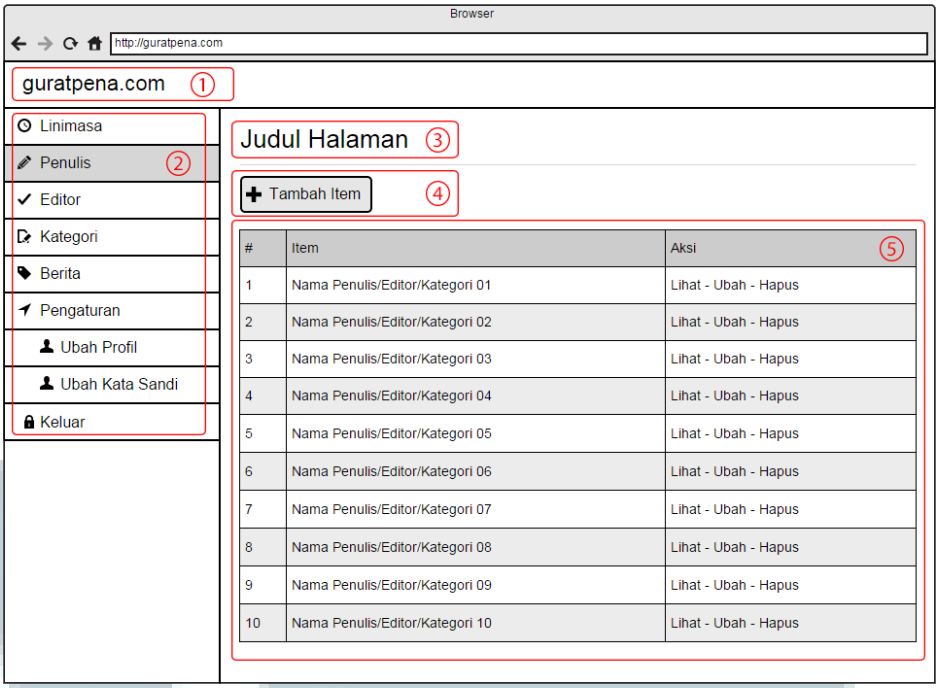
Gambar 3.15. Rancangan Layar Beranda pada *Admin*

Berikut merupakan penjelasan komponen-komponen pada gambar 3.15.

1. Baris judul situs, berisi judul situs.
2. Baris navigasi, berisi navigasi untuk berpindah ke halaman lain.
3. Baris judul halaman, berisi judul halaman yang sedang aktif saat ini.
4. Tampilan delapan berita terbaru dalam format linimasa. Atribut berita yang ditampilkan adalah judul berita, nama penulis, tanggal posting, kategori, dan nama penyunting.

Halaman manajemen untuk penulis, penyunting, dan kategori memiliki beberapa kesamaan. Halaman manajemen tersebut terdiri dari halaman yang

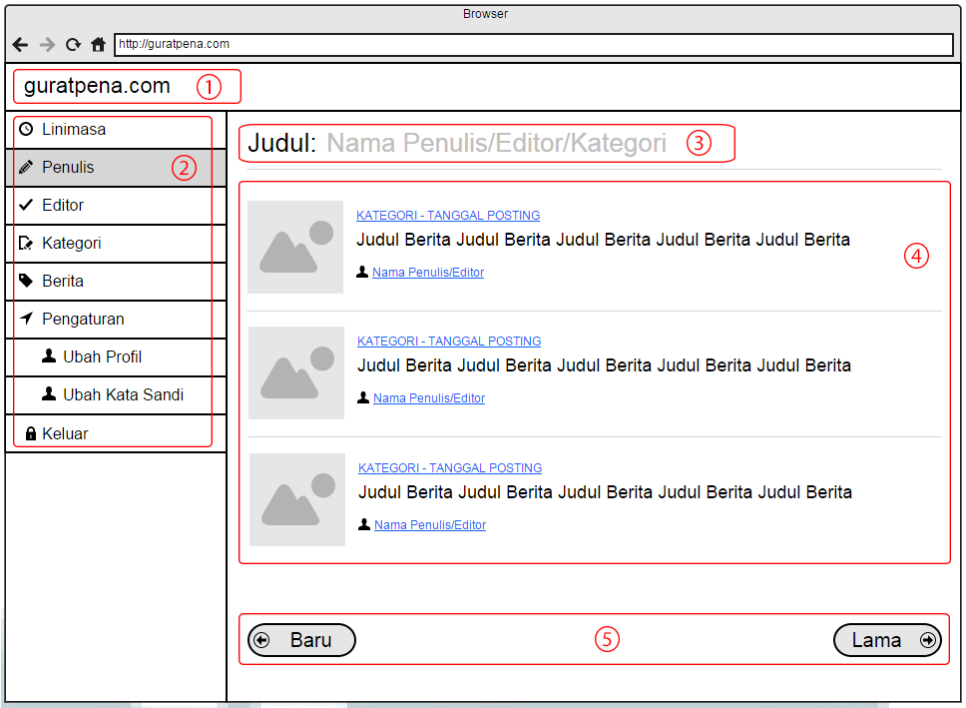
digunakan untuk melihat daftar *item* dan halaman yang digunakan untuk menambahkan *item* baru. Halaman-halaman tersebut dipaparkan sebagai berikut.



Gambar 3.16. Rancangan Layar Penulis pada *Admin*

Gambar 3.16 merupakan rancangan halaman dimana admin bisa melihat daftar penulis, penyunting, dan kategori pada basis data. Komponen 1, 2, dan 3 memiliki fungsi yang sama dengan gambar 3.15. Komponen 4 merupakan tombol yang digunakan untuk menambahkan penulis, penyunting, dan kategori. Proses penambahan penulis, penyunting, dan kategori dilakukan melalui *form* yang ditampilkan dalam bentuk *popup*. Komponen 5 menampilkan daftar penulis, penyunting, dan kategori dalam bentuk tabel. Dalam tabel tersebut juga dilengkapi tombol aksi lihat, ubah dan hapus *item*.

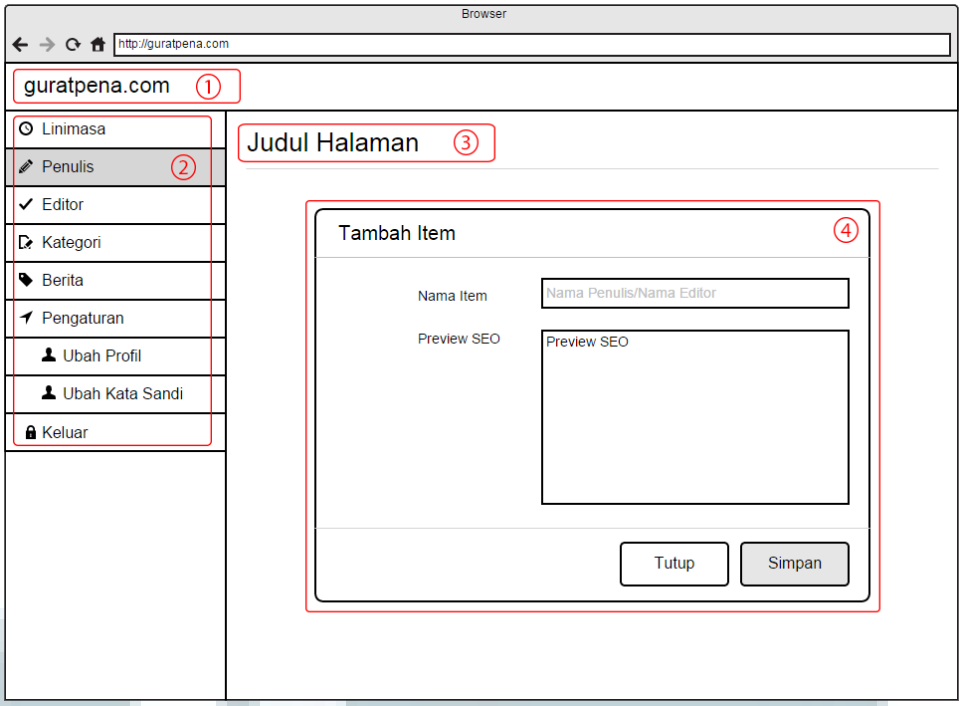
Saat tombol lihat ditekan, maka *admin* akan diarahkan ke halaman yang menampilkan daftar berita berdasarkan penulis, penyunting ataupun kategorinya. Halaman ini dipaparkan pada gambar 3.17.



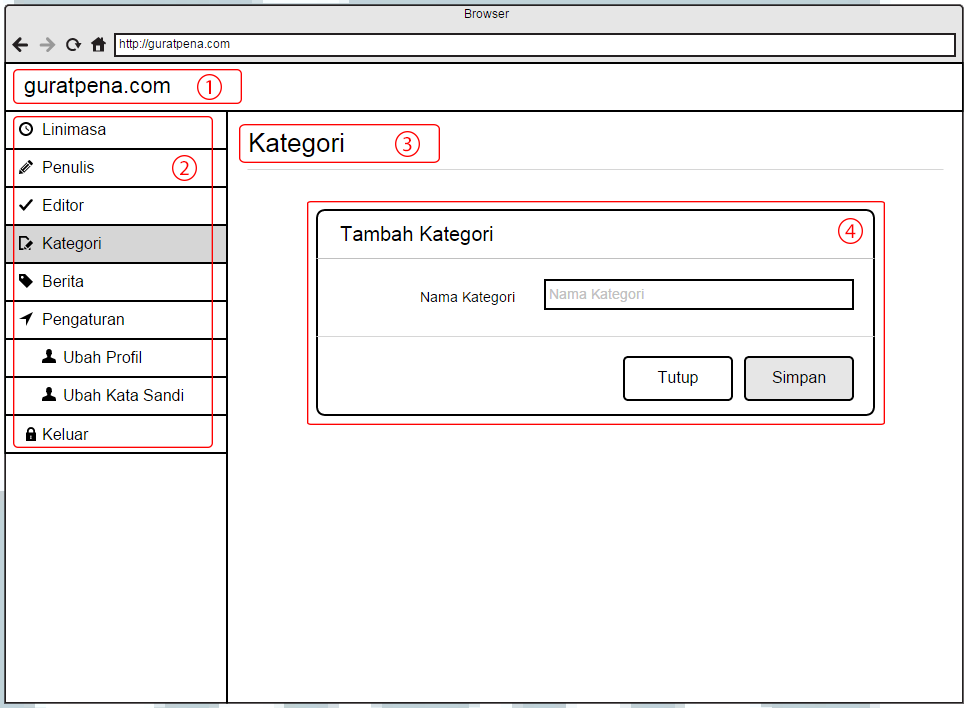
Gambar 3.17. Rancangan Layar Lihat Berita Berdasarkan Penulis, Editor, dan Kategori pada *Admin*

Fungsi komponen 1, 2, dan 3 pada halaman ini sama dengan pada komponen di gambar 3.15. Komponen 4 pada halaman ini berisi daftar berita yang dapat dilihat oleh *admin*. Komponen 5 merupakan *pagination* yang dapat digunakan untuk berpindah antar halaman.

Gambar 3.18 memaparkan tampilan layar saat *admin* ingin menambahkan penulis dan penyunting berita, sedangkan gambar 3.19 memaparkan tampilan layar saat *admin* ingin menambahkan kategori ke dalam basis data. Komponen 1, 2, dan 3 pada tampilan ini memiliki fungsi yang sama dengan tampilan pada gambar 3.15. Komponen 4 merupakan *form* untuk melakukan masukan guna menambahkan data baru.



Gambar 3.18. Rancangan Layar Tambah Penulis dan Editor pada *Admin*



Gambar 3.19. Rancangan Layar Tambah Kategori pada *Admin*

Data paling penting yang harus ada adalah berita itu sendiri. Oleh karena itu, *admin* dapat melakukan manajemen berita dari CMS yang tersedia. Pada gambar 3.20, *admin* dapat melihat berita yang sudah ditambahkan dalam bentuk *list* yang

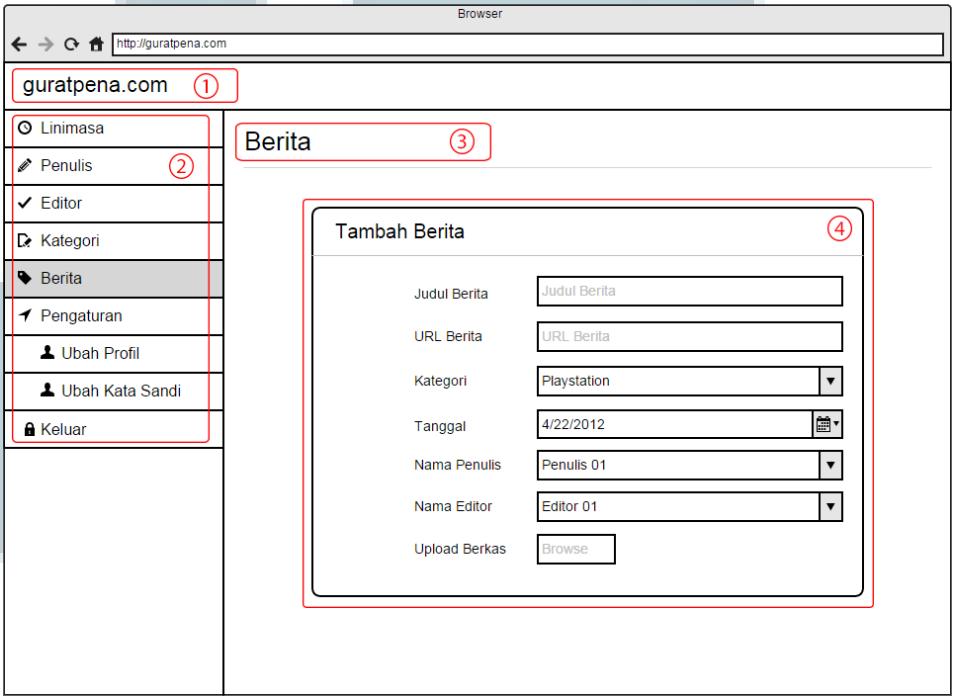
ditampilkan pada komponen 5. *Admin* dapat bernavigasi antar halaman dengan menggunakan tombol pagination yang ditunjukkan pada komponen 7. Atribut berita yang ditampilkan antara lain judul berita, nama penulis berita, kategori berita, tanggal posting, dan tiga buah tombol aksi, yakni lihat, ubah, dan hapus yang ditampilkan pada komponen 6. Komponen 1, 2, dan 3 pada gambar 3.20 memiliki fungsi yang sama dengan gambar 3.15. Komponen 4 merupakan tombol untuk menambahkan berita.

Gambar 3.20. Rancangan Layar Berita pada *Admin*

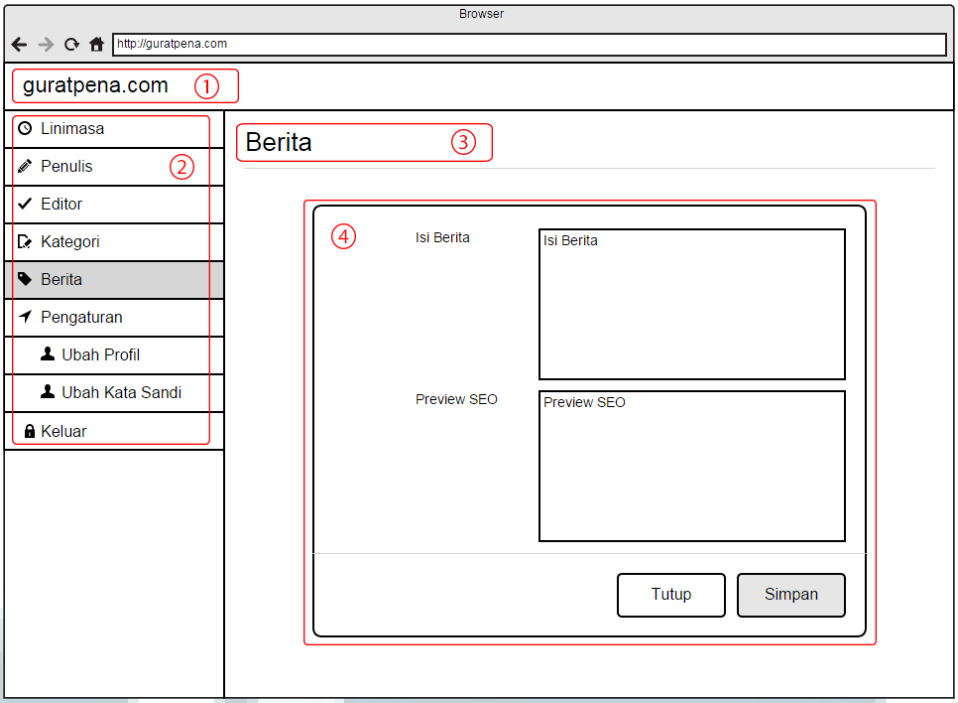
Saat *admin* menekan tombol lihat, ia akan diarahkan pada tampilan seperti gambar 3.21. Pada tampilan tersebut, *admin* dapat melihat berita secara detail. Komponen 1, 2, dan 3 berfungsi sama dengan komponen pada gambar 3.15. Komponen 4 berfungsi menampilkan berita secara detail pada layar. Tombol ubah dan hapus digunakan untuk mengubah dan menghapus berita bersangkutan.

Gambar 3.21. Rancangan Layar Lihat Detail Berita pada *Admin*

Gambar 3.22 dan 3.23 merupakan tampilan layar saat *admin* ingin menambahkan berita baru ke dalam basis data. Komponen 1, 2, 3, dan 4 memiliki fungsi yang sama dengan komponen pada gambar 3.18 dan 3.19.

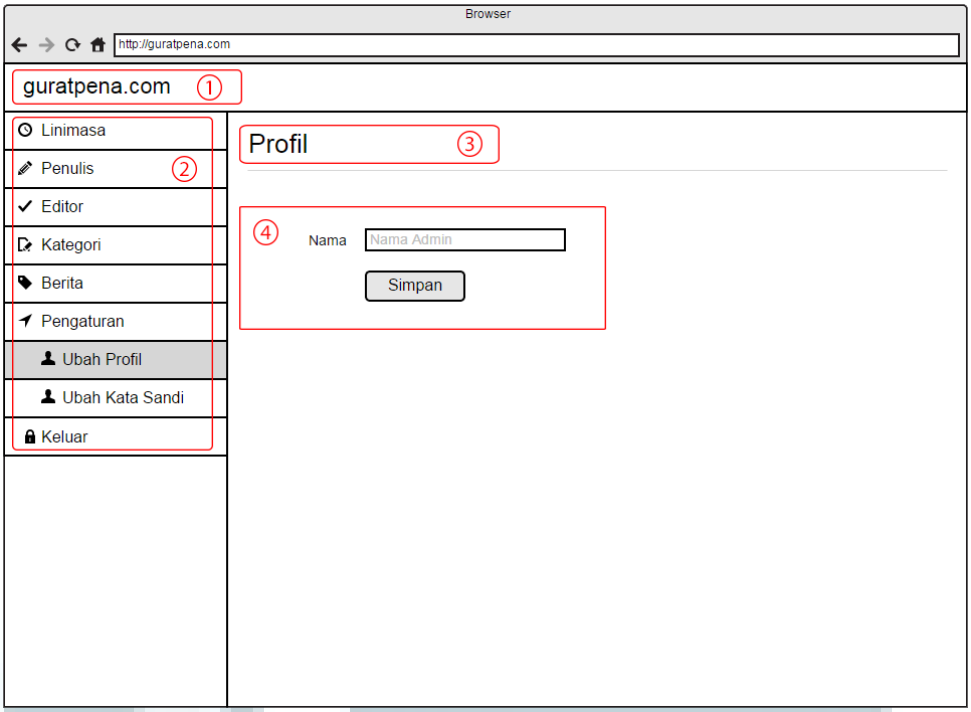


Gambar 3.22. Rancangan Layar Tambah Berita pada *Admin* (Bagian 1)

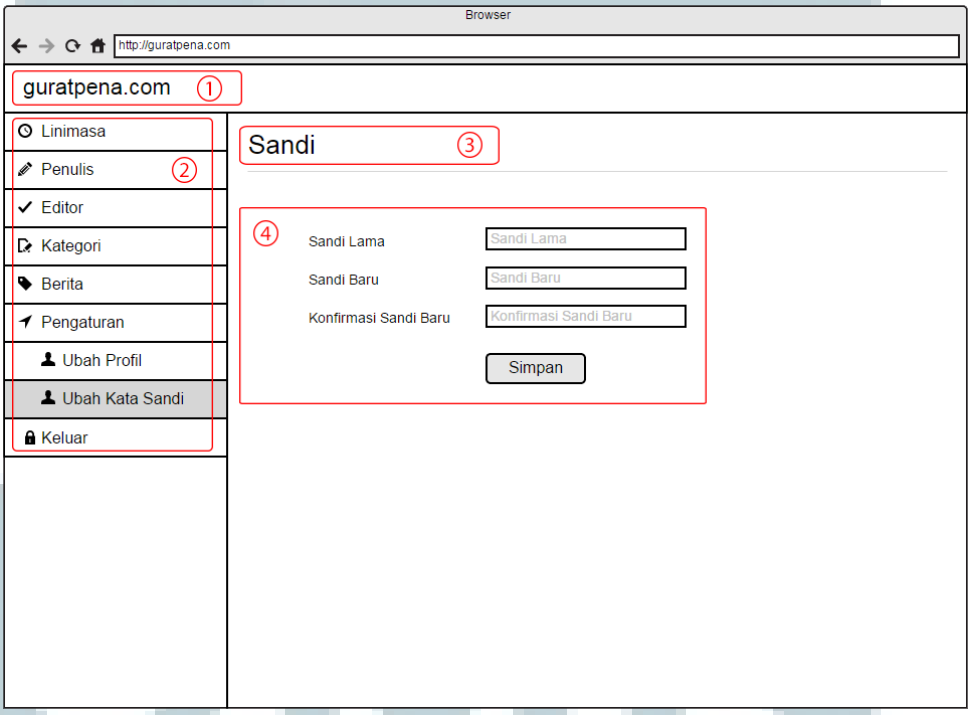


Gambar 3.23. Rancangan Layar Tambah Berita pada *Admin* (Bagian 2)

Selain melakukan manajemen konten, *admin* dapat juga melakukan manajemen profilnya sendiri. Atribut yang dapat diubah oleh *admin* adalah nama dan kata sandinya. Tampilan untuk melakukan hal ini dipaparkan pada gambar 3.24 dan 3.25. Komponen 1, 2, dan 3 memiliki fungsi yang sama dengan gambar 3.15. Komponen 4 merupakan *form* yang digunakan untuk melakukan masukan.



Gambar 3.24. Rancangan Layar Profil pada *Admin*



Gambar 3.25. Rancangan Layar Ubah Sandi pada *Admin*