

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Analisis Sistem**

Analisis terhadap suatu sistem sangat diperlukan untuk mengetahui kegiatan-kegiatan yang sedang berjalan dalam suatu sistem, untuk memahami dan mengerti jalannya sistem serta hambatan-hambatan yang terdapat dalam sistem tersebut.

Pada tahap ini yang perlu dilakukan adalah analisis terhadap sistem yang sedang berjalan dan bagaimana aliran dokumen yang sedang berjalan yang akan digambarkan dalam bentuk analisis pengkodean, analisis kebutuhan fungsional, analisis kebutuhan non-fungsional dan analisis pengguna.

##### **3.1.1 Analisis Masalah**

Berdasarkan hasil observasi, penelitian dan pengamatan yang telah dilakukan, diperlukan sistem informasi pariwisata Kelurahan Langenharjo berbasis *web* untuk mempermudah pihak Kelurahan Langenharjo untuk mempromosikan objek-objek wisata dan potensi-potensi wisata lainnya, maka dari itu mahasiswa dalam penulisan laporan skripsi ini mengidentifikasi masalah sebagai berikut :

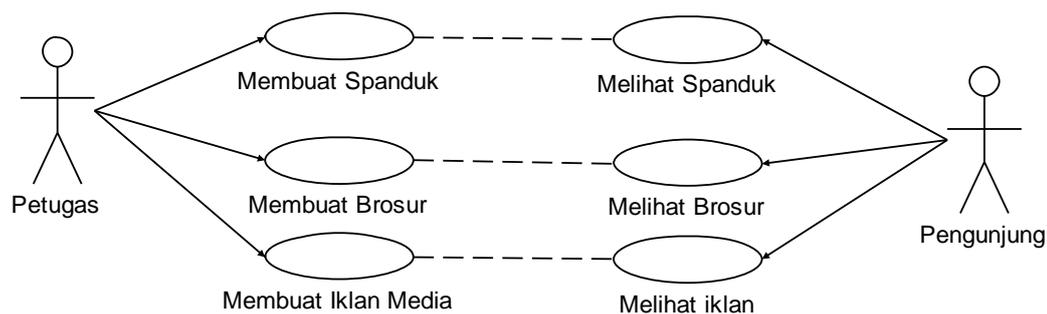
- 1) Pemberitahuan mengenai informasi melalui brosur, media *online*, radio lokal, dan media cetak.
- 2) Belum tersedianya media seperti *website* tentang pariwisata Kelurahan Langenharjo.

##### **3.1.2 Sistem yang berjalan saat ini**

Berdasarkan hasil observasi dan wawancara yang telah dilakukan di desa wisata Langenharjo, penyampaian informasi desa wisata Langenharjo dalam hal memperkenalkan atau mempromosikan pariwisata. Karena dalam pengolahan data informasi desa wisata Langenharjo belum berjalan secara optimal dengan keterbatasan penyampaian informasi. Dalam hal ini juga penyampaian berita serta *event-event* pariwisata masih belum terpublikasikan dengan baik dikarenakan

banyak kendala masyarakat masih banyak yang tidak mengetahui wisata apa saja yang ada di desa tersebut.

Untuk mendapatkan informasi masyarakat hanya bisa mengetahui informasi dari masyarakat yang pernah pergi ke tempat wisata itu dan brosur yang berada di depan kelurahan, jadi di perlukan sebuah sistem informasi tentang potensi wisata yang berada di Kel. Langenharjo. Sistem manual dapat dilihat pada Gambar 3.1



**Gambar 3.1 Sistem Manual**

### 3.1.3 Kelemahan sistem yang berjalan saat ini

Berdasarkan hasil observasi dan wawancara yang telah dilakukan di Kelurahan Langenharjo terdapat beberapa kelemahan sistem yang berjalan saat ini yaitu :

- 1) Belum adanya *website* khusus untuk mempromosikan wisata yang ada
- 2) Masyarakat sulit mencari potensi wisata yang berada di Kelurahan Langenharjo dengan akurat
- 3) Dibutuhkan waktu yang cukup lama untuk mengakses informasi pariwisata Kelurahan Langenharjo yang masih manual
- 4) Informasi pariwisata dalam bentuk tercetak tidak tahan lama, jika ada penambahan informasi baru tidak dapat secara langsung untuk diketahui, dan harus membuat lagi yang baru sehingga tidak ekonomis.

### 3.1.4 Analisis sistem yang baru

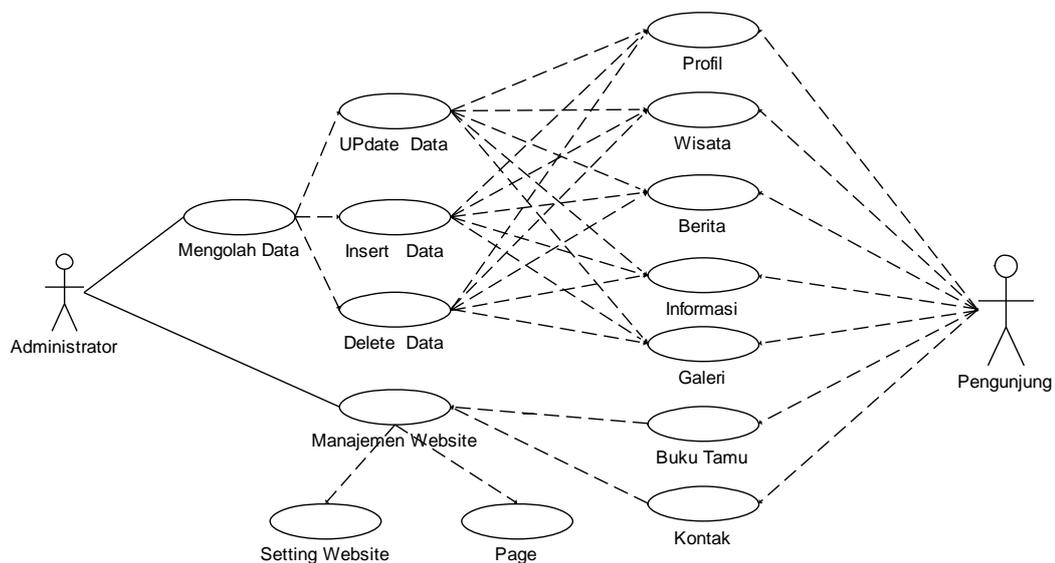
Sistem yang baru merupakan pengembangan dari sistem yang mulanya manual, dan kini dikembangkan menjadi sistem yang berbasis *web*, secara prinsip sama, perbedaannya terletak pada proses pengolahannya, sehingga informasi yang diberikan bisa diterima oleh masyarakat umum tanpa harus menyebarkan brosur. Pada sistem yang baru dibuat menggunakan bahasa pemrograman *PHP* dan *database MySQL*.

Dalam sistem yang baru ini yang mulanya petugas menjadi seorang *admin website* yang *mengupdate* atau memasukkan data informasi mengenai pariwisata Kelurahan Langenharjo. Dengan adanya sistem informasi pariwisata Kelurahan Langenharjo berbasis *web*, kantor Kelurahan tidak susah lagi menginformasikan pariwisata dan *even-even* pariwisata karena sudah mempunyai sistem informasi berbasis *web*.

### 3.2 Perancangan Sistem

Perancangan sistem adalah merancang atau *mendesain* suatu sistem yang baik, yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur untuk mendukung operasi sistem.

Pada Kelurahan Langenharjo akan dibuat sebuah aplikasi sistem informasi pariwisata Kelurahan Langenharjo berbasis *web*. Perancangan aplikasi ini dibangun bertujuan untuk memudahkan masyarakat umum untuk mengetahui potensi wisata apa saja yang berada di Kelurahan Langenharjo. Perancangan sistem bisa dilihat pada Gambar 3.2.



Gambar 3.2 Perancangan Sistem

Perancangan aplikasi sistem informasi pariwisata Kelurahan Langenharjo berbasis *web* ini bersifat *object oriented* (berorientasi objek) dengan menggunakan

*Unified Modeling Language (UML)* sebagai bahasa pemodelan. Pembangunan *website* sistem informasi pariwisata Kelurahan Langenharjo berbasis *web* ini dilakukan dengan menggunakan :

- 1) PHP sebagai *frame work* aplikasi untuk membaca pemograman PHP
- 2) MySQL sebagai *Database server*
- 3) PHP *MyAdmin* sebagai *tools* untuk *Database* berbasis *web*

### 3.2.1 *Usecase Diagram*

*Usecase Diagram* digunakan untuk memodelkan dan menyatakan unit fungsi / layanan yang disediakan oleh sistem: sub sistem atau *class* ke pemakai. *Usecase* dapat dilengkapi dengan batasan sistem yang diberi label nama sistem. *Usecase* adalah sesuatu yang menyediakan hasil yang dapat diukur ke pemakai atau sistem eksternal.

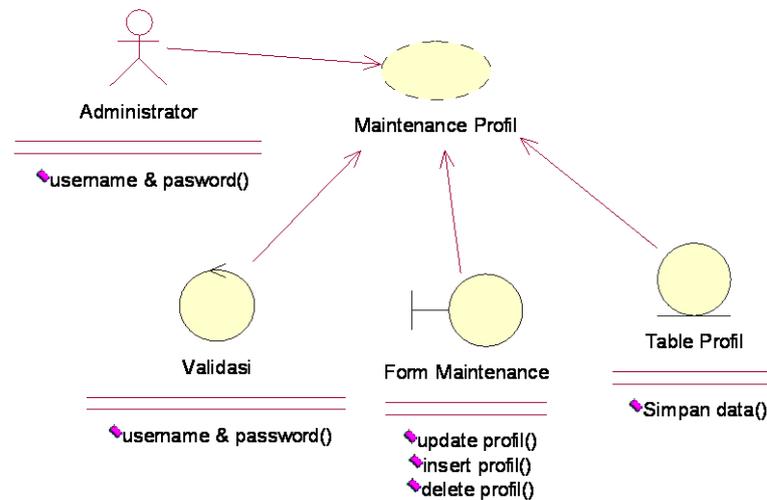
*Usecase diagram* dalam sistem ini memiliki dua aktor utama yaitu *administrator* dan pengunjung sistem yang dilambangkan dengan simbol orang dengan label *administrator* dan pengunjung. Diagram diatas menggambarkan tentang tindakan-tindakan yang dapat dilakukan oleh *administrator* dan pengunjung di dalam sistem, yaitu *admin* mengelola berita mengelola informasi, mengelola dan mengatur *pages*, mengelola galeri, *setting website*, mengelola profil wisata *website*, dimana semua proses tersebut hanya dapat dilakukan setelah *administrator* sistem melakukan proses login terlebih dahulu dengan cara memasukkan *username* dan *password*, sedangkan pengunjung hanya bisa mencari informasi-informasi yang terdapat di sistem informasi pariwisata Kelurahan Langenharjo berbasis web ini dan pengunjung dapat mengisi buku tamu dan mengirim pesan.

### 3.2.2 *Usecase Realization*

*Usecase Realization* menggambarkan hubungan aktor dengan sistem pada saat proses yang ada dalam sistem.

### 3.2.2.1 Usecase Realization Maintenance Profil

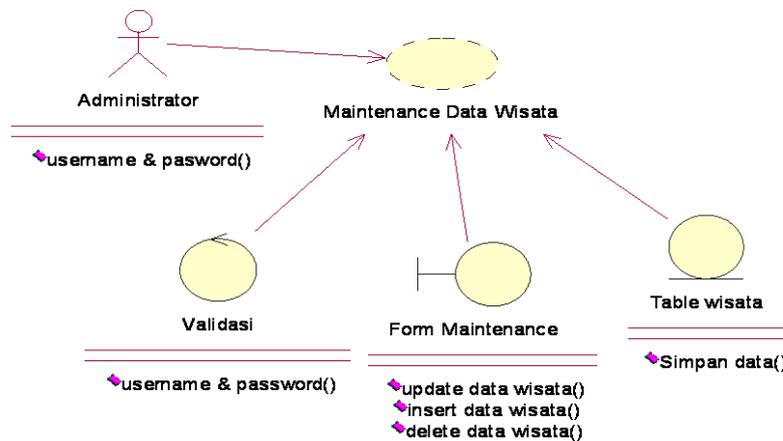
*Usecase realization maintenance* profil adalah proses yang dilakukan oleh *administrator* saat melakukan *login*, *administrator* melakukan *login* dengan cara memasukkan *username* dan *password*, setelah *valid admin* akan menuju menu utama *admin* dan mengelola data profil pada *form maintenance* profil setelah proses *maintenance* data selesai kemudian disimpan dalam *table* profil. *Usecase Realization Maintenance* profil bisa dilihat pada Gambar 3.3



Gambar 3.3 *Usecase Realization Maintenance* profil

### 3.2.2.2 Usecase Realization Maintenance data Wisata

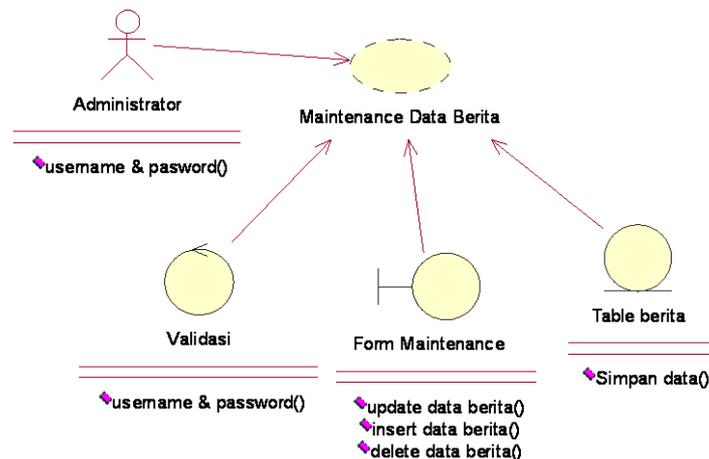
*Usecase realization maintenance* data wisata adalah proses yang dilakukan oleh *administrator* saat melakukan *login*, *administrator* melakukan *login* dengan cara memasukkan *username* dan *password*, setelah *valid admin* akan menuju menu utama *admin* dan mengelola data wisata pada *form maintenance* data wisata setelah proses *maintenance* data selesai kemudian disimpan dalam *table*. *Usecase Realization Maintenance* wisata bisa dilihat pada Gambar 3.4



Gambar 3.4 Usecase Realization Maintenance data wisata

### 3.2.2.3 Usecase Realization Maintenance Data Berita

*Usecase realization maintenance data berita* adalah proses yang dilakukan oleh *administrator* saat melakukan *login*, *administrator* melakukan *login* dengan cara memasukkan *username* dan *password*, setelah *valid admin* akan menuju menu utama *admin* dan mengelola data berita pada *form maintenance* berita setelah proses *maintenance* data selesai kemudian disimpan dalam table berita. *Usecase Realization Maintenance* berita bisa dilihat pada Gambar 3.5

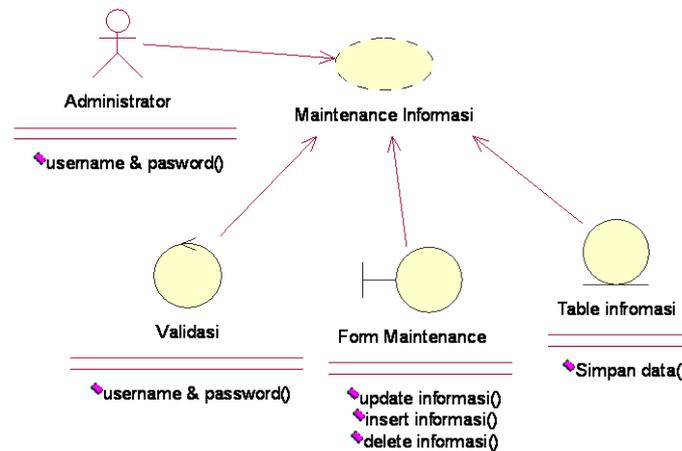


Gambar 3.5 Usecase Realization Maintenance Data Berita

### 3.2.2.4 Usecase Realization Maintenance Informasi

*Usecase realization maintenance Informasi* adalah proses yang dilakukan oleh *administrator* saat melakukan *login*, *administrator* melakukan *login* dengan cara memasukkan *username* dan *password*, setelah *valid admin* akan menuju

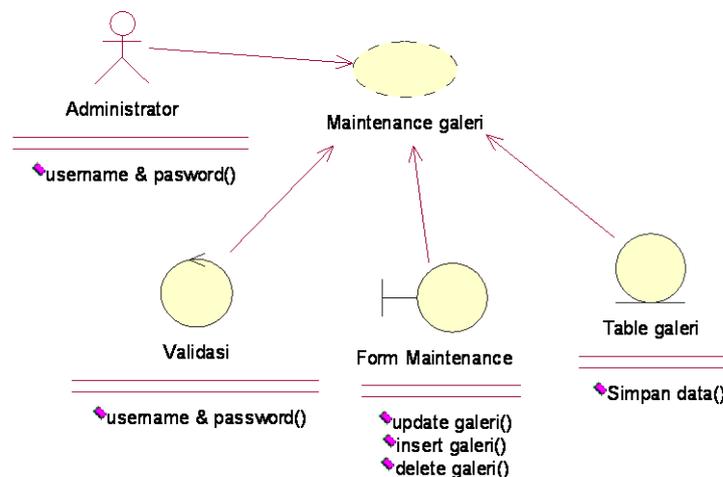
menu utama *admin* dan mengelola data Informasi pada *form maintenance* Informasi setelah proses *maintenance* selesai disimpan dalam table Informasi. *Usecase Realization Maintenance* Informasi bisa dilihat pada Gambar 3.6



Gambar 3.6 *Usecase Realization Maintenance* Informasi

### 3.2.2.5 *Usecase Realization Maintenance* Galeri

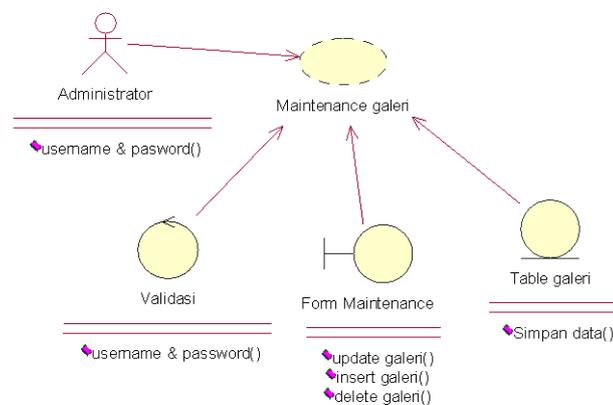
*Usecase realization maintenance* galeri adalah proses yang dilakukan oleh *administrator* saat melakukan *login*, *administrator* melakukan *login* dengan cara memasukkan *username* dan *password*, setelah *valid admin* akan menuju menu utama *admin* dan mengelola galeri pada *form maintenance* galeri setelah proses *maintenance* selesai disimpan dalam table galeri. *Usecase Realization Maintenance* galeri bisa dilihat pada Gambar 3.7.



Gambar 3.7 *Usecase Realization Maintenance* Galeri

### 3.2.2.6 Usecase Realization Maintenance Galeri

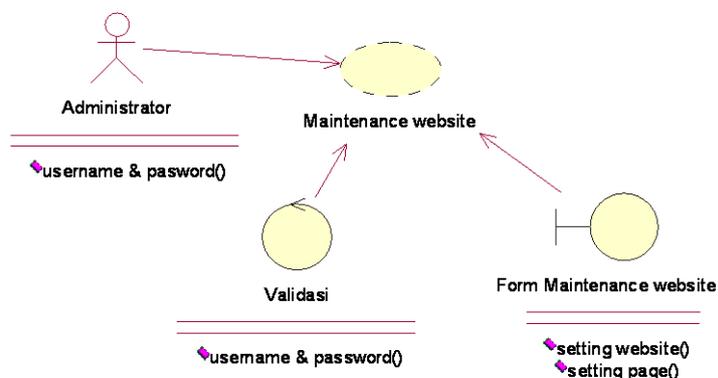
*Usecase realization Maintenance Galeri* menggambarkan hubungan admin sebagai aktor dengan sistem pada saat proses mengelola gambar atau foto pada galeri. Admin melakukan login dan mengelola data informasi pada form maintenance galeri. Kemudian sistem melakukan *validasi* setelah *validasi valid* sistem selanjutnya melakukan proses penyimpanan atau mengunggah ke dalam table galeri. *Usecase realization Maintenance Galeri* bisa dilihat pada Gambar 3.8.



Gambar 3.8 Usecase Realization Maintenance Galeri

### 3.2.2.7 Usecase Realization Maintenance Website

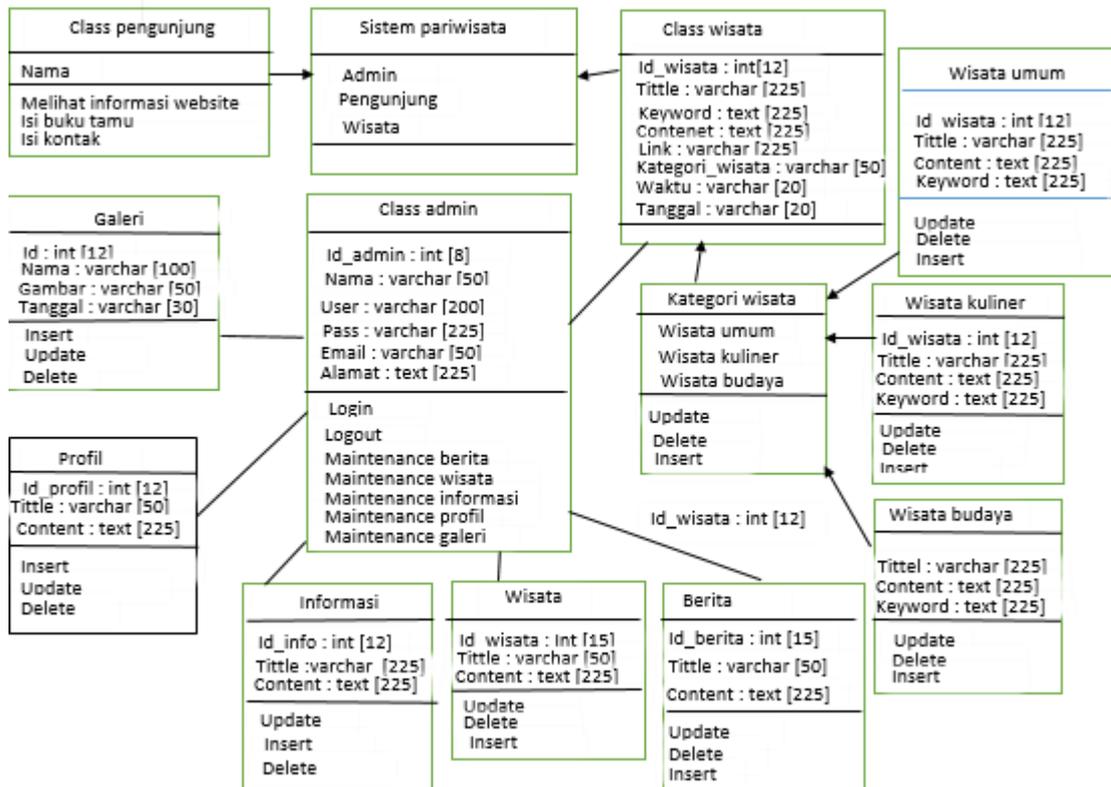
*Usecase Realization Maintenance website* menggambarkan hubungan admin sebagai aktor dengan sistem pada saat proses setting *website* dan *page* pada *form maintenance website*. *Usecase Realization Maintenance* profil Wisata bisa dilihat pada Gambar 3.9



Gambar 3.9 Usecase Realization Maintenance Website

### 3.2.3 Class Diagram

*Class Diagram* adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang digunakan. *Class Diagram* memberikan gambaran (diagram statis) tentang sistem/ perangkat lunak dan relasi-relasi yang ada di dalamnya.



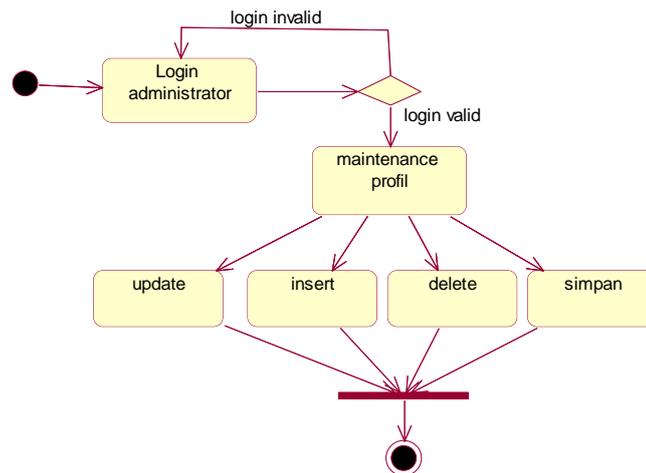
Gambar 3.10 Class diagram

### 3.2.4 Activity Diagram

*Activity Diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *usecase*. *Activity diagram* menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas sejenisnya seperti *use case*. Berikut ini *Activity Diagram* dari sistem informasi pariwisata Kabupaten Sukoharjo berbasis *web*.

### 3.2.4.1 Activity Diagram Maintenance Profil

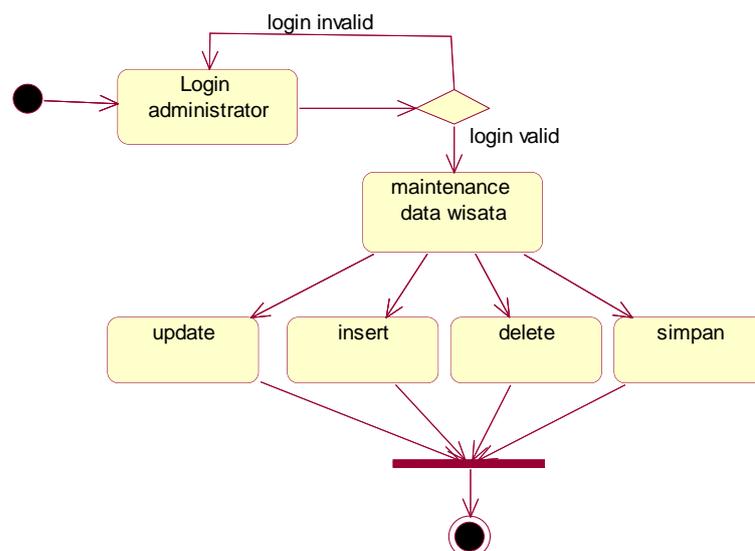
Activity Diagram yang pertama adalah *activity diagram maintenance Profil*. Dimulai dengan memasukkan *login administrator*, kemudian apabila setelah divalidasi dan login data *valid* terdapat beberapa pilihan yaitu : *update, insert, Delete, simpan*. *diagram maintenance* profil ditunjukkan dalam Gambar 3.11.



Gambar 3.11 Activity Diagram Maintenance Profil

### 3.2.4.2 Activity Diagram Maintenance Data Wisata

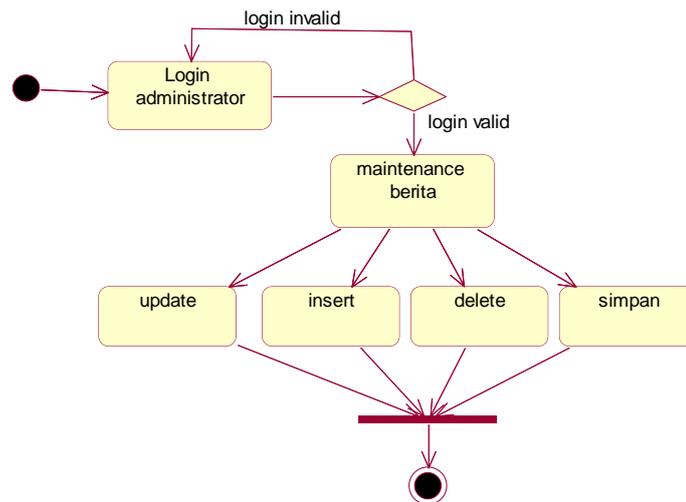
Activity Diagram *maintenance data wisata* dimulai dengan memasukkan *login administrator*, kemudian apabila setelah divalidasi dan login data *valid* terdapat beberapa pilihan yaitu : *update, insert, Delete, simpan*. *diagram maintenance* data wisata ditunjukkan dalam Gambar 3.12.



Gambar 3.12 Activity Diagram Maintenance Data Wisata

### 3.2.4.3 Activity Diagram Maintenance Berita

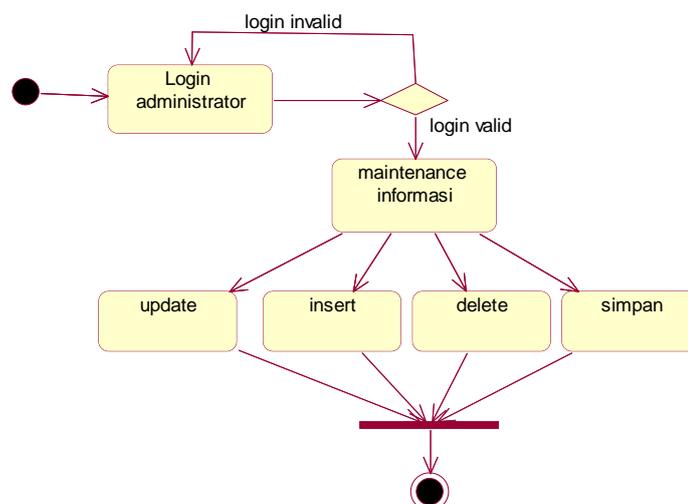
Activity Diagram *maintenance data wisata* dimulai dengan memasukkan *login administrator*, kemudian apabila setelah divalidasi dan login data *valid* terdapat beberapa pilihan yaitu : *update, insert, Delete, simpan*. diagram *maintenance data berita* ditunjukkan dalam Gambar 3.13.



Gambar 3.13 Activity Diagram Maintenance Berita

### 3.2.4.4 Activity Diagram Maintenance Informasi

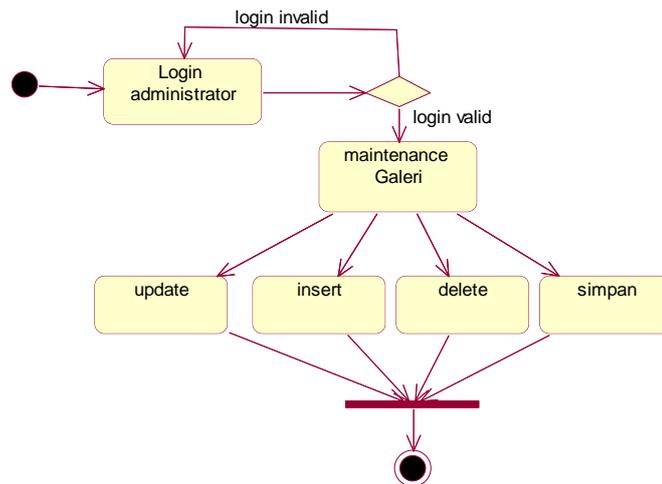
Activity Diagram *maintenance infromasi* dimulai dengan memasukkan *login administrator*, kemudian apabila setelah divalidasi dan login data *valid* terdapat beberapa pilihan yaitu : *update, insert, Delete, simpan*. diagram *maintenance informasi* ditunjukkan dalam Gambar 3.14.



Gambar 3.14 Activity Diagram Maintenance Informasi

### 3.2.4.5 Activity Diagram Maintenance Galeri

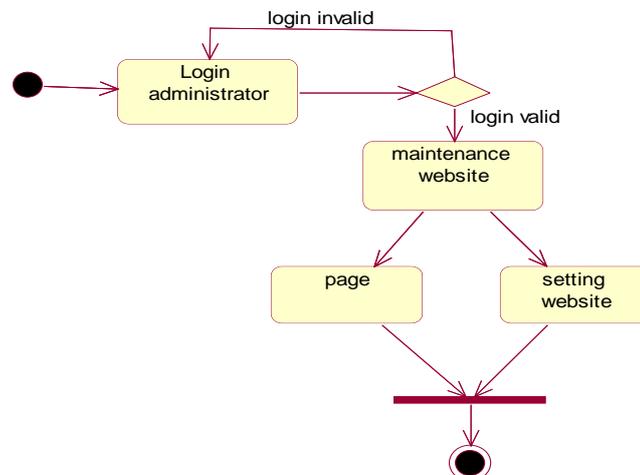
*Activity Diagram maintenance galeri* dimulai dengan memasukkan *login administrator*, kemudian apabila setelah divalidasi dan login data *valid* terdapat beberapa pilihan yaitu : *update*, *insert*, *Delete*, *simpan*. *diagram maintenance galeri* ditunjukkan dalam Gambar 3.15



Gambar 3.15 Activity Diagram Maintenance Galeri

### 3.2.4.6 Activity Diagram Maintenance Website

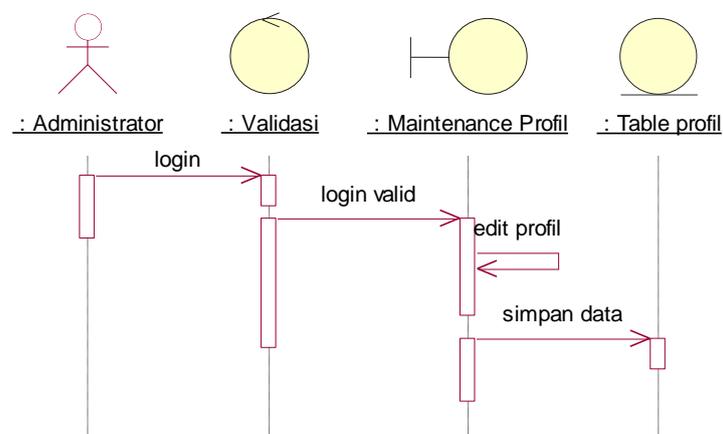
*Activity Diagram maintenance website* dimulai dengan memasukkan *login administrator*, kemudian apabila setelah divalidasi dan login data *valid* terdapat beberapa pilihan yaitu : *maintenance page* dan *seting website*. *diagram maintenance galeri* ditunjukkan pada Gambar 3.16



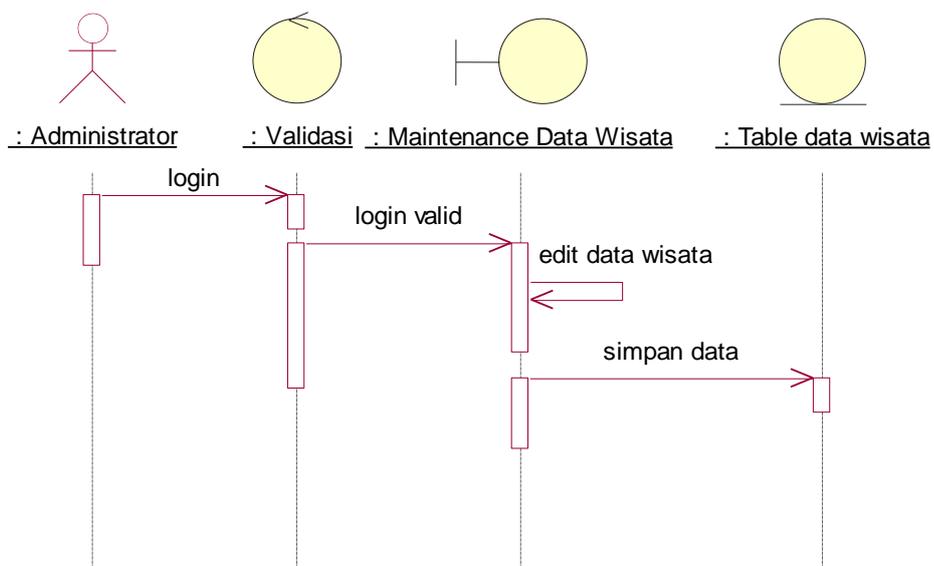
Gambar 3.16 Activity Diagram Maintenance Website

### 3.2.5 Sequence Diagram

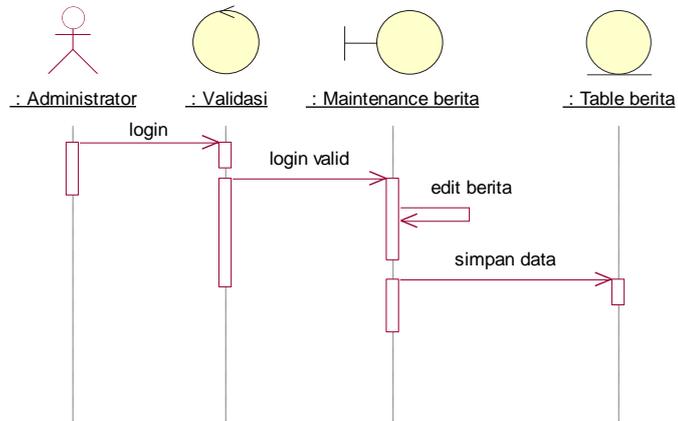
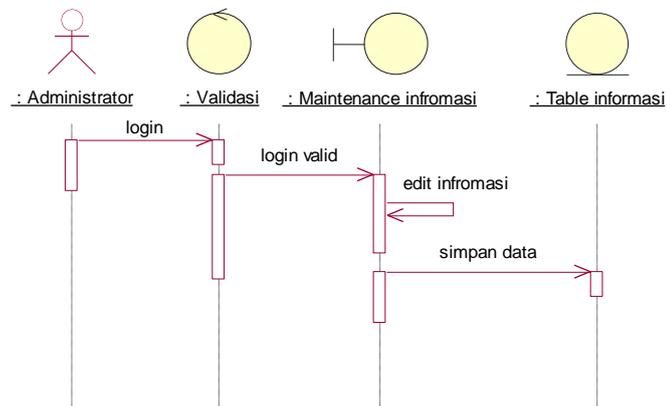
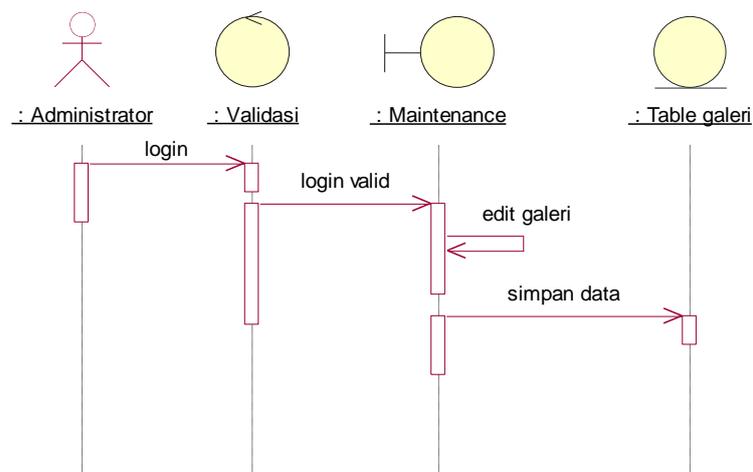
*Sequence Diagram* adalah suatu diagram yang menggambarkan interaksi antar obyek dan menghasilkan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya di taruh di paling kiri dari diagram.



Gambar 3.17 *Sequence Diagram Maintenance Profil*

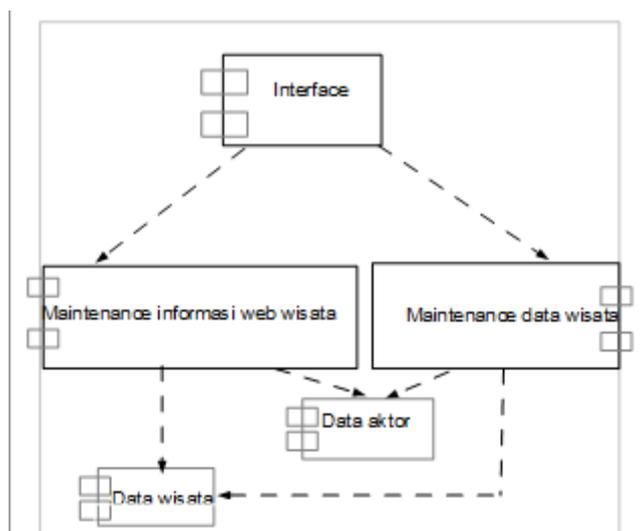


Gambar 3.18 *Sequence Diagram Maintenance Data Wisata*

Gambar 3.19 *Sequence Diagram Maintenance Berita*Gambar 3.20 *Sequence Diagram Maintenance Informasi*Gambar 3.21 *Sequence Diagram Maintenance Galeri*

### 3.2.6. Component Diagram

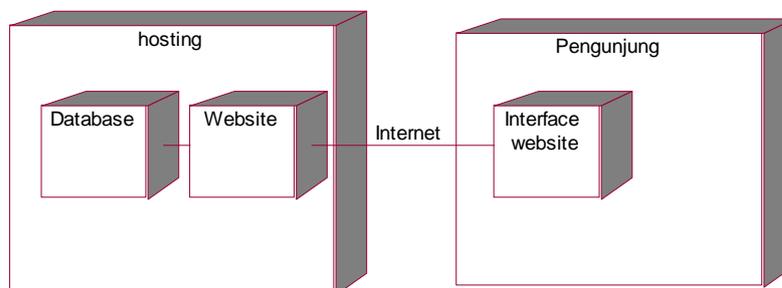
*Component diagram* menggambarkan struktur dan hubungan antar komponen perangkat lunak termasuk ketergantungan satu dengan lainnya. Dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Seperti terlihat pada Gambar 3.22.



Gambar 3.22 *Component Diagram*

### 3.2.7. Deployment diagram

*Deployment diagram* jika diartikan dalam bahasa Indonesia berarti diagram pendistribusian. Sesuai dengan kebutuhan sistem yang akan diterapkan pada *Website Desa Wisata Langenharjo* adalah menggunakan *server / hosting*, yang artinya website ini sudah di-*upload* pada *hosting* dan domain, maka bentuk *deployment diagram* adalah seperti ditunjukkan pada Gambar 3.23.

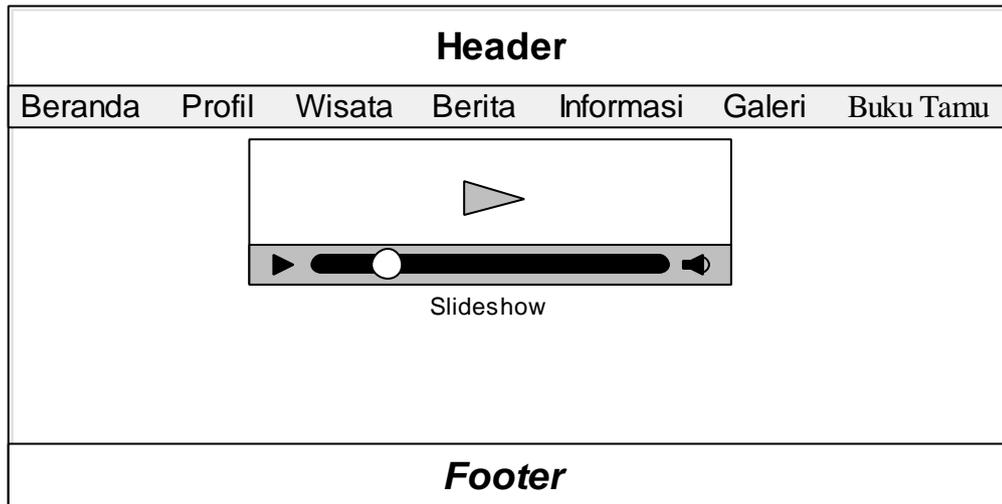


Gambar 3.23. *Deployment Diagram*

### 3.3 Desain layout *form*, proses *input* dan *output*

#### 3.3.1 Desain Halaman Utama *Website* ( Untuk pengunjung)

Desain halaman utama aplikasi adalah tampilan tatap muka pertama antara admin dengan aplikasi. Terdapat tampilan yaitu logo, header, menu utama, konten-konten aplikasi, tanggal, dan *footer*



Gambar 3.24 Halaman Utama *Website*

### 3.3.2 Desain tampilan halaman utama Admin

Desain utama admin ini akan muncul setelah admin melakukan *login* dengan benar, setelah benar admin akan menuju halaman ini. Di dalam halaman ini terdapat menu seperti berita, wisata, informasi, pages, galeri, profile, setting website. Halaman utama admin dapat dilihat pada Gambar 3.26

<i>Administrator Header</i>		<input type="button" value="Logout"/>
<i>Profil</i>		
<i>Berita</i>		
<i>Wisata</i>		
<i>Informasi</i>		
<i>Galeri</i>		
<i>Pages</i>		
<i>Setting</i>		
<b>Footer</b>		

Gambar 3.25 Desain Halaman Administrator

### 3.3.3 Desain *maintenance* berita baru

Desain halaman berita baru diakses oleh admin setelah melalui proses login terlebih dahulu menggunakan *username* dan *password*. Pada tampilan admin mempunyai hak akses untuk melakukan *maintenance* data berita baru atau membuat berita baru sesuai kategori yang ada.

Tambah Berita

Title

Kategori

Text (200)

Gambar 3.26 Desain Maintenance Berita Baru

### 3.3.4 Desain *maintenance* kategori berita

Desain *maintenance* kategori diakses oleh admin untuk menambahkan kategori dalam pembuatan berita baru dengan cara admin melalui proses *login* terlebih dahulu menggunakan *username* dan *password*. Pada tampilan ini *admin* bisa menambahkan kategori berita, kategori berita ini digunakan dalam pembuatan kelompok-kelompok berita, atau class-class berita.

Tambah Kategori Berita

Title

Varchar (100)

Kategori

Varchar (50)

No	Title	Action	
		Update	Delete

Simpan

Batal

Gambar 3.27 Desain *Maintenance* Kategori Berita

### 3.3.5 Desain *maintenance* data wisata baru

Desain halaman *maintenance* data wisata baru berita ini diakses oleh admin setelah melalui proses *login* terlebih dahulu menggunakan *username* dan *password*. Pada tampilan ini *admin* mempunyai hak akses untuk melakukan *maintenance* data wisata baru atau membuat daftar wisata baru sesuai kategori yang ada dalam menu kategori wisata. Desain tambah wisata baru dapat dilihat pada Gambar 3.28

Tambah Wisata

Tittle

Varchar (200)

Kategori

Varchar (80)

Text (200)

Simpan      Batal

Gambar 3.28 Desain Maintenance Data Wisata Baru

### 3.3.6 Desain post wisata

Desain post wisata diakses oleh admin untuk menampilkan semua informasi wisata yang sudah di unggah, atau semua wisata yang sudah pernah dibuat dari awal hingga akhir. Pada tampilan ini admin mempunyai hak akses untuk melihat semua wisata yang sudah dibuat dan dapat melakukan update, dan delete berita. Menu post wisata dapat dilihat pada Gambar 3.29

Wisata

Varchar (100)      Search

No	<i>Tittle</i>	<i>Action</i>	
int (10)	Var (100)	<i>Update</i>	<i>Delete</i>

Simpan      Batal

Gambar 3.29 Desain Post Wisata

### 3.3.7 Desain *maintenance* informasi baru

Desain halaman maintenance data informasi baru ini di akses oleh admin setelah melalui proses login terlebih dahulu menggunakan username dan password. Pada tampilan ini admin mempunyai hak akses untuk melakukan maintenance data informasi baru atau membuat daftar informasi baru sesuai katategori yang ada dalam menu kategori informasi. Desain tambah wisata baru pada Gambar 3.30

Tambah Informasi

Title

Varchar (200)

Kategori

Varchar (80)

Text (200)

Simpan      Batal

Gambar 3.30 Desain *Maintenance* Informasi Baru

### 3.3.8 Desain Post Informasi

Menu post informasi diakses oleh admin untuk menampilkan semua berita yang sudah di unggah, atau semua informasi yang sudah pernah dibuat dari awal hingga akhir. Pada tampilan ini admin mempunyai hak akses untuk melihat semua informasi yang sudah dibuat dan dapat melakukan update, dan delete informasi. Desain post berita dapat dilihat pada Gambar 3.31

Informasi

Varchar (100)      Search

No	Title	Action	
int (10)	Var (100)	Update	Delete

Simpan      Batal

Gambar 3.31 Desain Post Informasi

### 3.3.9 Desain *maintenance new pages*

Desain *new pages* adalah menu yang digunakan untuk pembuatan halaman baru dalam menu utama *website*. Judul dari *pages* ini dapat juga di gunakan untuk membuat sub menu dalam menu utama *website*. Desain *maintenance new pages* dapat dilihat pada Gambar 3.32

Tambah Pages

Title

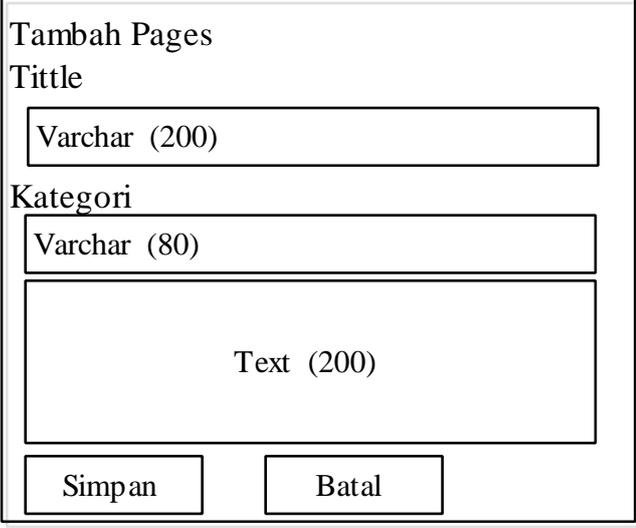
Varchar (200)

Kategori

Varchar (80)

Text (200)

Simpan      Batal



Gambar 3.32 Desain Menu Post Pages