

2.2. Configuration Items Screenshot

2.2.1. Baseline 1



Gambar diatas merupakan tampilan dari halaman Sistem SIAMANG berdasarkan Baseline 1 yang baru memiliki fitur : *Login*, *Melihat acara terbaru*, *Tambah acara* dan *Logout*

BAB III

USER MANUAL (PANDUAN PENGGUNA)

3.1 Informasi Umum

3.1.1. Deskripsi Sistem

Sistem Informasi Acara Musik Malang (SIAMANG) adalah perangkat lunak berbasis web yang terkait pada bidang musik dibagian hiburan yang digunakan untuk membantu ataupun mempermudah peminat acara musik untuk mengetahui informasi mengenai acara musik yang ada di kota malang. Terdapat dua jenis pengguna pada sistem ini, yaitu Admin dan Pengguna.

Aplikasi ini dibangun dengan database MySQL dan menggunakan bahasa pemrograman PHP, serta dapat berjalan pada browser seperti Google Chrome dan Mozilla Firefox. Aplikasi ini dikembangkan oleh Kelompok 5 Kelas MKPL-B Fakultas Ilmu Komputer Universitas Brawijaya.

3.1.2. Referensi

Berikut adalah daftar referensi yang digunakan dalam pengembangan sistem:

1. Code Igniter
2. Apache Web Server

3.1.3. Bantuan

a. Kontak Pengembang

Jika terdapat pertanyaan atau keluhan, pengguna dapat menghubungi di email kelompok5mkplb@ub.ac.id.

3.1.4. Struktur Panduan Pengguna

Panduan ini terbagi atas 3 bagian, yaitu:

- 3.1 Informasi Sistem: Terdiri atas informasi dasar mengenai program, seperti fungsi dan deskripsi program.
- 3.2 Rangkuman Sistem: Terdiri atas rangkuman umum sistem.
- 3.3 Memulai Sistem: Terdiri atas panduan umum mengenai cara kerja sistem dari awal hingga akhir.

3.1.5. Akronim dan Istilah

| Istilah dan | Uraian |
|-------------|--|
| SIA | Sistem Informasi Acara Musik Malang |
| SKPL | Adalah spesifikasi dari suatu produk atau program yang melakukan suatu fungsi tertentu pada lingkungan tertentu. |
| Admin | Adalah orang yang menjalankan dan mengelola suatu sistem secara kontinu. |

| | |
|--|---|
| Input | Adalah data atau perintah yang yang dimasukkan kedalam memori komputer untuk diproses selanjutnya. |
| Pengguna | Adalah orang yang akan langsung menjalankan sistem. Pengguna dan pelanggan umumnya adalah orang yang sama, walaupun bisa juga berbeda (IEEE Std 610.12-1990). |
| <i>Object Oriented Programming (OOP)</i> | Adalah suatu metode pemrograman yang berorientasi objek. OOP biasa digunakan dalam perancangan suatu perangkat, yang melibatkan diagram-diagram seperti usecase diagram, usecase <i>Scenario</i> , <i>Sequence</i> diagram, dan class diagram. |
| <i>Use Case</i> | <i>Use Case / Use Case Diagram</i> adalah suatu digram yang dibentuk berdasarkan kebutuhan-kebutuhan fungsional dari suatu sistem yang dirancang. <i>Use Case Diagram</i> disusun berdasarkan Attribut yang dapat dilakukan oleh para Aktor |
| <i>Use Case Scenario</i> | <i>Use Case Scenario</i> adalah bentuk alur dari attribut yang terdapat pada <i>Use Case Diagram</i> berdasarkan aktor. <i>Use Case Scenario</i> berisi tentang Aktor yang melakukannya, Pre-Kondisi, Tahapan/ <i>Main Flow</i> , Alternatif Flow dan Post-Kondisi. |
| <i>Sequence Diagram</i> | <i>Sequence Diagram</i> adalah bentuk alur dari <i>Use Case Scenario</i> berdasarkan fungsi. <i>Sequence Diagram</i> berisi tentang Aktor, <i>Boundary/Batasan</i> , <i>Control</i> , dan Database. |
| <i>Class Diagram</i> | Class Diagram adalah bentuk alur dari Diagram-diagram sebelumnya berdasarkan <i>method</i> per kelas/form. Class Diagram berisi tentang nama Diagram/nama kelas dan isi/ <i>method</i> |
| <i>Security</i> | Istilah dalam rekayasa Perangkat Lunak yang berarti sistem harus mempunyai autentifikasi, keamanan jaringan, dan keamanan data. |

3.2 Rangkuman Sistem

3.2.1. Konfigurasi Sistem

Pengguna dapat berinteraksi dengan database sistem untuk menjalankan perintah-perintah seperti melihat data, memasukkan data baru, mengubah data, menghapus data, dan mencari data.

3.2.2. Alur Data

Untuk Tambah Acara, data dari acara yang telah ditambahkan oleh admin disimpan dalam database, yang nantinya akan dipanggil oleh method tambahAcara(), kemudian ditampilkan dalam halaman Admin dan halaman Pengguna.

Untuk Edit Acara, data dari acara yang telah diedit oleh admin disimpan dalam database, yang nantinya akan dipanggil oleh method editACara(), kemudian ditampilkan dalam halaman Admin dan halaman Pengguna sebagai acara yang telah dilakukan perubahan.

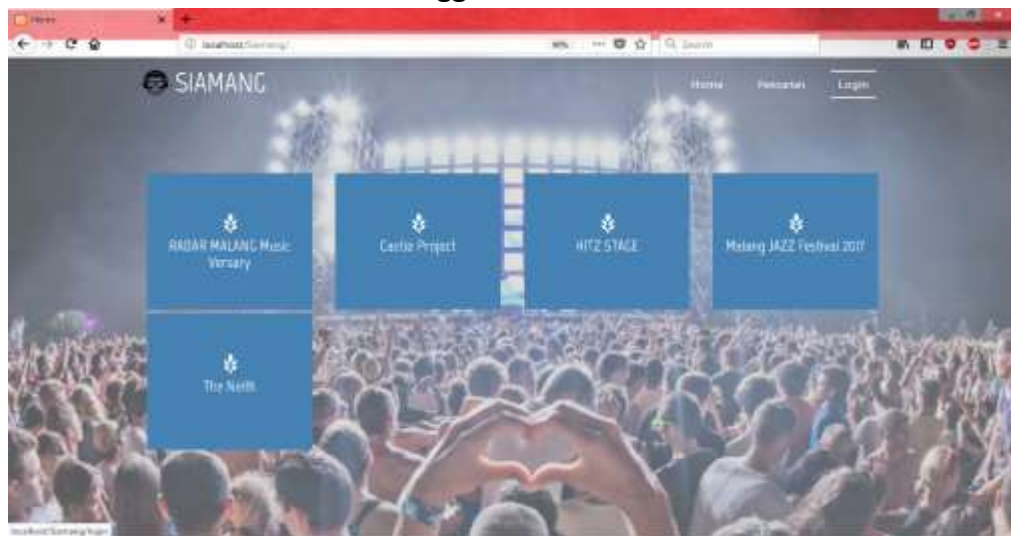
3.2.3. Kemungkinan dan Metode Alternatif

Lakukan pencadangan data secara reguler, dikarenakan jika terjadi kondisi darurat dapat terjadi kehilangan data.

3.3 Memulai Sistem

Baseline 1 : Versi 1.0

3.3.1. Antarmuka Halaman Pengguna



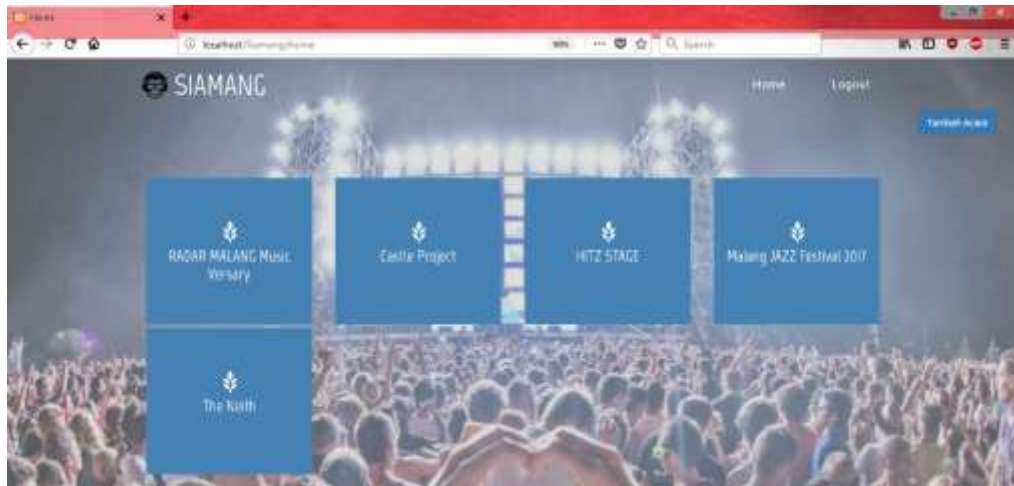
Tampilan halaman pengguna dengan beberapa acara yang telah ditambahkan sebelumnya oleh Admin

3.3.2. Antarmuka Halaman Login



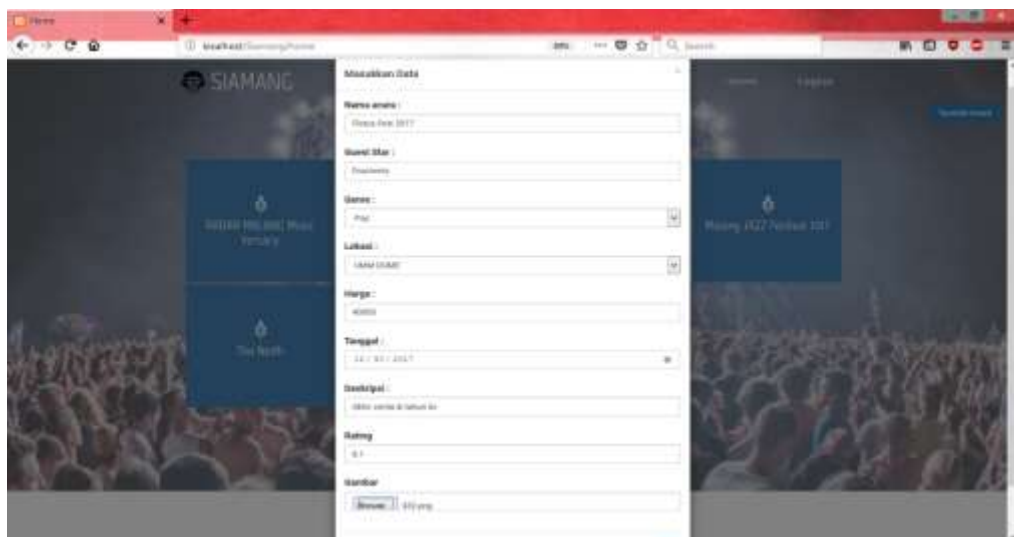
Pada halaman Login, Aktor dapat memasukkan *username* dan *password* untuk masuk kedalam sistem sebagai Admin

3.3.3. Antarmuka Halaman Admin



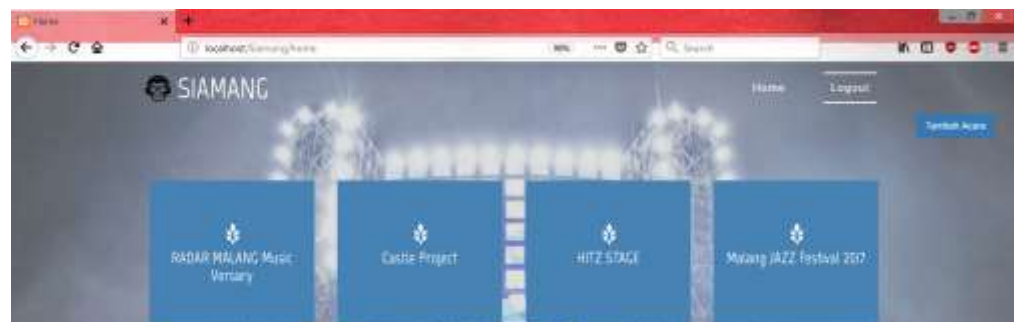
Tampilan diatas adalah tampilan ketika User telah berhasil Login kedalam sistem sebagai Admin.

3.3.4. Antarmuka Tambah Acara



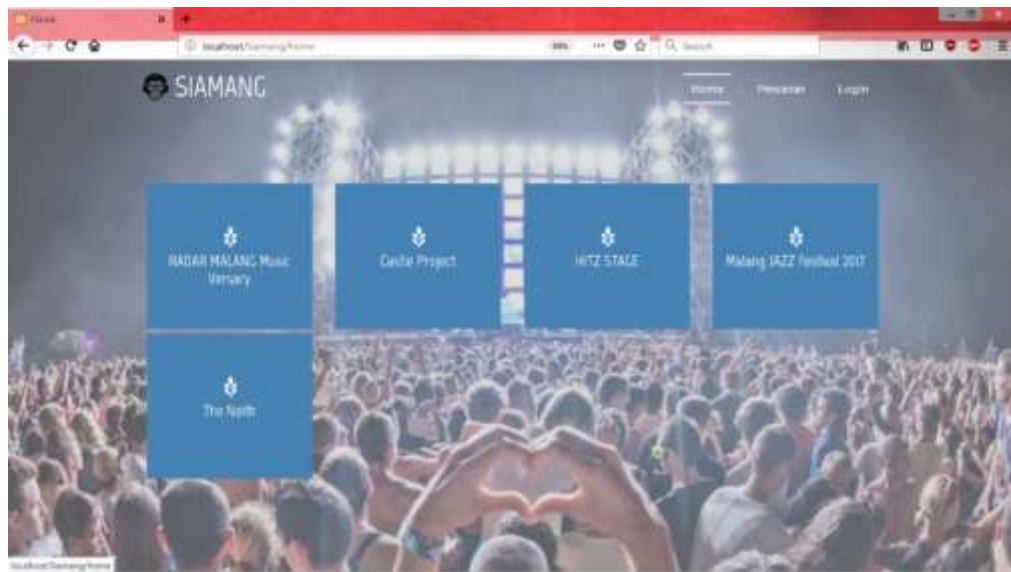
Tampilan diatas adalah tampilan ketika Admin menambahkan sebuah acara musik.

3.3.5. Antarmuka Logout



Tampilan diatas adalah tampilan ketika Admin ingin melakukan Logout.

Antarmuka Melihat Acara Musik Terbaru



Tampilan diatas adalah tampilan ketika Pengguna melihat acara musik yang terbaru, yaitu yang baru diupload oleh Admin.

BAB IV

CODES

4.1 Controller

4.1.1. Controller Acara

```
class Acara extends CI_Controller{
    function __construct(){
        parent::__construct();
        $this->load->model('m_acara');
    }
    function index(){
        $id_acara = $_GET['id_acara'];
        $data['acara'] = $this->m_acara->
        >tampilAcara($id_acara)->result();

        $this->load->view('v_acara', $data);
    }

    function tambahAcara(){
        $nama_acara = $this->input->post('nama_acara');
        $gs = $this->input->post('gs');
        $genre = $this->input->post('genre');
        $lokasi = $this->input->post('lokasi');
        $harga = $this->input->post('harga');
        $tanggal = $this->input->post('tanggal');
        $deskripsi = $this->input->post('deskripsi');
        $rating = $this->input->post('rating');
        $gambar = $this->input->post('gambar');
        $data=array(
            'nama_acara'=>$nama_acara,
            'gs' => $gs,
            'genre' => $genre,
            'lokasi' => $lokasi,
            'harga' => $harga,
            'tanggal' => $tanggal,
            'deskripsi' => $deskripsi,
            'rating' => $rating,
            'gambar' => $gambar
        );
        $this->m_acara->tambahAcara($data,'acara');
        redirect('home');
    }
}
```

4.1.2. Controller Home

```
class Home extends CI_Controller{
    function __construct(){
        parent::__construct();
        $this->load->model('m_home');
    }
    function index(){
        $data['acara'] = $this->m_home->tampilAcara()
        >result();
        $this->load->view('v_home', $data);
    }
}
```

4.1.3. Controller Login

```
class Login extends CI_Controller{
    function __construct(){
        parent::__construct();
        $this->load->model('m_login');}
    function index(){
        $this->load->view('v_login');}
    function aksi_login(){
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        if (empty($username) || empty($password)){
            echo "<script>alert('username atau password
belum diisi')
            window.location='http://localhost/Siamang/login';</scri
pt>";
        }else{
            $where = array(
                'username' => $username,
                'password' => $password
            );
            $cek = $this->m_login->cek_login("user",$where)->num_rows();
            if($cek > 0){
                $row = $this->m_login->getData('user',$where)->row();
                $_SESSION['role'] = $row->role;
                $_SESSION['id_admin'] = $row->id_admin;
                $_SESSION['username'] = $username;
                $_SESSION['nama'] = $row->nama;
                redirect(base_url("home"));
            }else{
                echo "<script>alert('username atau
password salah')
                window.location='http://localhost/Siamang/login';</scri
pt>";
            }
        }
        function logout(){
            $this->session->sess_destroy();
            redirect(base_url('login'));}
    }
```

4.2 View

4.2.1. View Home

```
<title>Home</title>
</head>
<body>
<!--header strat here-->
<div class="banner agileits">
    <div class="header">
        <div class="container">
            <div class="header-main">
                <div class="logo">
                    <h1><span class="books"></span> <a
href="/Siamang/home">SIAMANG</a></h1>
                </div>
```



```

<div class="top-nav">
    <span class="menu"> </span>
    <nav class="cl-effect-21" id="cl-effect-21">
        <ul class="res">
            <li><a href="/Siamang/home"
class="active">Home</a></li>
            <li><?php
if(!isset($_SESSION['id_admin'])) { ?>
                <div class="dropdown">
                    <button class="dropbtn
active">Pencarian</button>
                    <div class="dropdown-
content">
                        <a href="#modalLokasi"
data-toggle="modal">Lokasi</a>
                        <a href="#modalGenre"
data-toggle="modal">Genre</a>
                        <a
href="/Siamang/pencarian/limateratas">5 Teratas</a>
                    </div>
                </div> <?php } ?>
            </li>
            <li><?php
.....

```

4.3 Model

4.3.1. Model Acara, Home dan Login,

```

class M_acara extends CI_Model{
    function tampilAcara($id){
        return $this->db->query("SELECT * FROM acara
WHERE $id = id_acara");
    }
    function tambahAcara($data,$table){
        $this->db->insert($table,$data);
    }
}

class M_home extends CI_Model{
    function tampilAcara(){
        return $this->db->query("SELECT * FROM acara
ORDER BY id_acara DESC");
    }
}

class M_login extends CI_Model{
    function cek_login($table,$where){
        return $this->db->get_where($table,$where);
    }
    function getData($table,$where){
        return $this->db->get_where($table,$where);
    }
}

```