PENGEMBANGAN SISTEM INFORMASI MANAJEMEN PARIWISATA TERPADU SEBAGAI MEDIA PROMOSI KABUPATEN BANYUWANGI

SKRIPSI

Disusun oleh:
Fadly Yoga Pratama
NIM:125150406111001

PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN PARIWISATA TERPADU
SEBAGAI MEDIA PROMOSI KABUPATEN BANYUWANGI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Fadly Yoga Pratama
NIM: 125150406111001

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Februari 2017
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Satrio Agung W., S.Kom, M.Kom	Fajar Pradana, S.ST, M.Eng
NIP. 19860521 201212 1 001	NIP. 19871121 201504 1 004

Mengetahui
Ketua Jurusan Sistem Informasi

Herman Tolle, Dr. Eng., S.T, M.T
NIP. 19740823 200012 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 November 2016

Fadly Yoga Pratama

NIM: 125150406111001

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, atas rahmat dan hidayat-Nya penulis dapat menyelesaikan penyusunan tugas akhir dengan judul "Pengembangan Sistem Informasi Manajemen Pariwisata Terpadu Sebagai Media Promosi Kabupaten Banyuwangi".

Tugas akhir ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Komputer di Program Studi Sistem Informasi, Jurusan Sistem Informasi, Fakultas Ilmu Komputer Universitas Brawijaya.

Melalui pengantar ini penulis mengucapkan banyak terima kasih karena dalam penulisan dan penyusunan laporan tugas akhir ini penulis mendapat bantuan dan dorongan baik secara langsung maupun tidak langsung dari berbagai pihak diantaranya:

1. Bapak Satrio Agung W., S.Kom, M.Kom., selaku dosen pembimbing I dan Bapak Fajar Pradana, S.ST, M.Eng, selaku dosen pembimbing II yang telah sabar memberikan bimbingan, arahan, ilmu dan saran dalam penulisan dan penyusunan laporan skripsi ini.
2. Bapak Yusi Tyroni Mursityo, S.Kom selaku Dosen Pembimbing Akademik yang telah memberikan tuntunan dan motivasi serta bimbingan selama penulis menjadi mahasiswa Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Suprapto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian hingga menyelesaikan penulisan skripsi ini.
4. Bapak Herman Tolle, Dr. Eng., S.T, M.T selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian hingga menyelesaikan penulisan skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian hingga menyelesaikan penulisan skripsi ini.
6. Seluruh Bapak dan Ibu Dosen serta seluruh Staff Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya yang telah memberikan ilmunya serta arahan selama masa perkuliahan.
7. Ayah tercinta Gatot Suyono dan Ibunda terkasih Puji Rahayu atas segala dukungan, kasih sayang, perhatian dan kesabarannya, serta yang senantiasa selalu memberikan do'a dan semangat serta segala pengorbanan baik materil maupun moril demi terselesaikannya skripsi ini.

8. Ilyas Habiburrahman, Hafiz Rahmadi, Jeanny Anggraeni, dan Bayu Adi Bhaskara yang telah memberikan dan mengajarkan sedikit ilmu tentang Framework CodeIgniter dan memberikan semangat dalam pengerjaan skripsi ini.
9. Teman-teman "TSC" yang tidak bisa disebutkan satu persatu yang selalu menemani di waktu senggang dalam proses pengerjaan skripsi ini.
10. Teman-teman "JCKers" meliputi Abraham Dwiasmoro, Adi Nugroho, Afandi Ahmad Syaikhu, Aldi Yudha, Agung Baskoro W., Andre Firza Setyananda, Ardhi Hanifa Rahmansyah, Bagus Setiawan, Dewi Nuraini, Dimas Hamdan, Dyah Ayu W., Ishak Novianus, Isti Dyah, Komang Ayu, Nanda Kurnia W., Prasetya M. Aditya, Rahma Sischa, Tommy Krisna dan teman-teman seperjuangan FILKOM angkatan 2012 yang tidak bisa disebutkan satu persatu yang ikut mendukung dan menyemangati selama proses penulisan dan penyusunan laporan skripsi ini.
11. Seluruh Keluarga Besar Fakultas Ilmu Komputer yang telah memberikan tempat untuk kuliah dan mencari pengalaman di Fakultas Ilmu Komputer Universitas Brawijaya.
12. Seluruh Keluarga Besar Mahasiswa Sistem Informasi atas bantuan dan semangat kebersamaan yang telah terjalin selama ini.
13. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini. Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna dan tidak terlepas dari kekurangan serta kesalahan karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Oleh karena itu, segala kritik dan saran yang bersifat membangun penulis harapkan untuk penyempurnaan skripsi ini. Penulis juga berharap mudah-mudahan skripsi ini dapat bermanfaat bagi semua pihak.

Malang, 18 November 2016

Penulis

Fadly_y@rocketmail.com

repository.ub.ac.id

ABSTRAK

Seiring dengan meningkatnya jumlah wisatawan yang berkunjung di berbagai tempat wisata di Kabupaten Banyuwangi, hal ini dibuktikan dengan jumlah wisatawan pada tahun 2015 yang mengunjungi masing-masing tempat wisata di Kabupaten Banyuwangi yang didapat dari hasil wawancara di Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi (Data Pengunjung Obyek dan Wisata Tahun 2015). Sehingga wisatawan memerlukan referensi informasi tempat wisata di Kabupaten Banyuwangi secara terperinci agar dapat mengetahui apa saja yang ada pada tempat wisata tersebut. Pada umumnya cara mengetahui destinasi tempat wisata yang ingin dikunjungi para wisatawan harus mengujungi Dinas Kebudayaan dan Pariwisata kabupaten Banyuwangi karena terdapat *Tourist Information Center* yang melayani pertanyaan seputar informasi pariwisata di Kabupaten Banyuwangi. Salah satu upaya untuk mengurangi permasalahan yaitu dengan cara menerapkan sistem informasi manajemen pariwisata terpadu dengan metode *Object Oriented Design* (OOD) sebagai upaya untuk meningkatkan dan mempertahankan jumlah wisatawan yang berkunjung di Kabupaten Banyuwangi.

Dalam pembuatan sistem informasi manajemen pariwisata terpadu penelitian diawali dengan melakukan analisis kebutuhan untuk memperoleh semua kebutuhan yang diperlukan oleh pengguna dan dilakukan pemodelan kebutuhan menggunakan *usecase diagram* dan *activity diagram*. Selanjutnya dilakukan proses perancangan menggunakan *class diagram*, *sequence diagram*, perancangan *database*, dan perancangan antarmuka yang digunakan untuk dasar implementasi kemudian dilakukan implementasi sistem dan dilakukan pengujian dan analisa hasil dengan menggunakan metode *white-box* untuk menguji detail perancangan, metode validasi menggunakan *black-box* untuk menguji validitas kebutuhan fungsional dan menggunakan metode *compability* yang menguji sistem bisa berjalan di berbagai *browser* yang berbeda.

Analisa kebutuhan menghasilkan kebutuhan fungsional dan kebutuhan non-fungsional. Dari hasil pengujian *white-box* dapat disimpulkan bahwa, pengujian validasi menggunakan *black-box* dapat disimpulkan bahwa 100% kebutuhan fungsional dari sistem telah valid dan dari hasil pengujian *compability* dapat dinyatakan bahwa sistem dapat berjalan dengan baik pada *browser chrome*, *firefox*, *safari*, *browser android* dan *IOS*.

Kata kunci: pengembangan sistem informasi manajemen, pariwisata terpadu, *object oriented design*, media promosi

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

Along with the increasing number of tourists who visit the various tourist attractions in Banyuwangi, this is evidenced by the number of tourists in 2015 who visited each tourist spots in Banyuwangi obtained from interviews in the Department of Culture and Tourism Banyuwangi (Data Visitor Destinations and Tourism 2015). So tourists need a reference information places of tourism in Banyuwangi a detailed to be able know what is in tourist destinations. In general I know tourist attractions destinations want to visit the tourists have visited the Department of Culture and Tourism Banyuwangi because there Tourist Information Center that serves questions about the tourism information in Banyuwangi. One of the effort to reduce the problem, namely by implementing an integrated tourism management information system by the method of Object Oriented Design (OOD) as an effort to increase and maintain the number of tourists visiting Banyuwangi.

In the manufacture of integrated tourism management information systems research begins with a needs analysis to obtain all the necessary requirements by the user and the modeling needs using usecase diagrams and activity diagrams. Then performed the design process using class diagrams, sequence diagrams, database design, and the design of the interface is used for basic implementations and then do the system implementation and testing and analysis of the results using the white-box to test the detail design, method validation using black-box for test the validity of the functional requirements and methods compability that test the system can run on a variety of different browsers.

Analysis requirements generates functional requirements and non-functional requirements. From the results of white-box testing can be concluded that, validation testing using black-box can be concluded that 100% functional requirements of the system have been valid and the results of testing compability can be stated that the system can run both in the browser chrome, firefox, safari, browser android and IOS.

Keywords: integrated tourism, management information system development, media promotion, object oriented design

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL xii
DAFTAR GAMBAR xiv
DAFTAR LAMPIRAN xvi
BAB 1 PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan masalah 3
1.3 Tujuan 3
1.4 Manfaat 3
1.5 Batasan masalah 4
1.6 Sistematika pembahasan 4
BAB 2 LANDASAN KEPUSTAKAAN 6
2.1 Kajian Pustaka 6
2.2 Dasar Teori 7
2.2.1 Profil Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi 7
2.2.2 Sistem 7
2.2.3 Sistem Informasi 8
2.2.4 Sistem Informasi Manajemen 8
2.2.4.1 Tujuan Sistem Informasi Manajemen 8
2.2.4.2 Proses Manajemen 8
2.2.5 Promosi 9
2.2.6 Business Process 9
2.2.7 Software Engineering 9
2.2.8 Unified Modeling Language (UML) 10
2.2.8.1 Class Diagram 10

2.2.8.2 Use Case Diagram 10
2.2.8.3 Sequence Diagram 10
2.2.8.4 Activity DIagram 11
2.2.9 System Development Life Cycle (SDLC) 12
2.2.9.1 Waterfall Model 12
2.2.10 PHP (PHP: Hypertext Preprocessor) 15
2.2.11 MySQL 15
2.2.12 Framework CodeIgniter 15
2.2.13 Pengujian *Black-Box* 16
2.2.13.1 Validation Testing 15
2.2.14 Pengujian *White-Box* 16
2.2.15 Pengujian Compatibility 16
BAB 3 METODOLOGI 18
3.1 Tempat dan Waktu Penelitian 18
3.2 Langkah-langkah Penelitian 18
3.2.1 Wawancara 19
3.2.2 Study Literatur 19
3.2.3 Analisis Kebutuhan 19
3.2.4 Perancangan 19
3.2.5 Implementasi 19
3.2.6 Pengujian dan Analisis 20
3.2.7 Kesimpulan dan Saran 20
BAB 4 ANALISIS KEBUTUHAN 21
4.1 Deskripsi Umum Sistem 21
4.2 Proses Bisnis 21
4.3 Analisis Kebutuhan 23
4.3.1 Elisitasi dan Analisis Kebutuhan 23
4.3.2 Aturan Penomoran 24
4.3.3 Identifikasi Pengguna 25
4.3.4 Kebutuhan Fungsional 25
4.3.5 Kebutuhan Non Fungsional 29
4.4 Pemodelan Kebutuhan Sistem 29

4.4.1 *Use Case Diagram* 29
4.4.2 Activity Diagram 44
BAB 5 PERANCANGAN 49
5.1 Perancangan Arsitektur 49
5.1.1 Sequence Diagram 49
5.1.2 Class Diagram 55
5.2 Perancangan Komponen 57
5.2.1 *Algoritma Fungsi register_process()* 57
5.2.2 *Algoritma Fungsi login()* 57
5.2.3 *Algoritma Fungsi tambah()* 57
5.2.4 *Algoritma Fungsi update_process()* 58
5.2.5 *Algoritma Fungsi informasi($id_event)* 58
5.3 Perancangan Data 58
5.3.1 *Physical Data Models* 58
5.4 Perancangan Antarmuka 59
5.4.1 Perancangan Antarmuka Admin 59
5.4.2 Perancangan Antarmuka Pengelola 64
5.4.3 Perancangan Antarmuka Wisatawan 67
BAB 6 IMPLEMENTASI 72
6.1 Spesifikasi Perangkat Lunak 72
6.2 Spesifikasi Perangkat Keras 72
6.3 Batasan Implementasi 73
6.4 Implementasi Antarmuka 73
BAB 7 PENGUJIAN DAN ANALISIS 87
7.1 Pengujian Sistem 87
7.1.1 Pengujian Validasi dengan *Black-Box* 87
7.1.2 Pengujian *White-Box* 101
7.1.3 Pengujian *Compatibility* 107
7.2 Analisis Sistem 109
7.2.1 Analisis Pengujian Validasi dengan *Black-Box* 110
7.2.2 Analisis Pengujian *White-Box* 110
7.2.3 Analisis Pengujian *Compatibility* 110

BAB 8 PENUTUP 111
8.1 Kesimpulan 111
8.2 Saran 111
DAFTAR PUSTAKA 112

DAFTAR TABEL

Tabel 2.1 Simbol-simbol Use Case Diagram 11
Tabel 2.2 Simbol-simbol Sequence Diagram 11
Tabel 2.3 Simbol-simbol Activity Diagram 12
Tabel 4.1 Identifikasi Pengguna 25
Tabel 4.2 Kebutuhan Fungsional Pengguna Umum 25
Tabel 4.3 Kebutuhan Fungsional Wisatawan 26
Tabel 4.4 Kebutuhan Fungsional Pengelola Wisata 27
Tabel 4.5 Kebutuhan Fungsional Admin 28
Tabel 4.6 Kebutuhan Non Fungsional Sistem 29
Tabel 4.7 Registrasi Wisatawan (SIMPADU-F-001) 31
Tabel 4.8 Registrasi Pengelola (SIMPADU-F-003) 31
Tabel 4.9 Melihat Daftar Objek Wisata (SIMPADU-F-009) 32
Tabel 4.10 Melihat Detail Info Objek Wisata (SIMPADU-F-004) 32
Tabel 4.11 Melakukan Booking Objek Wisata (SIMPADU-F-006) 33
Tabel 4.12 Melihat Riwayat Booking Objek Wisata (SIMPADU-F-007) 34
Tabel 4.13 Melihat Daftar Booking Objek Wisata (SIMPADU-F-013) 34
Tabel 4.14 Konfirmasi Booking Objek Wisata (SIMPADU-F-014) 35
Tabel 4.15 Menambah Objek Wisata (SIMPADU-F-015) 35
Tabel 4.16 Melihat Grafik Jumlah Wisatawan (SIMPADU-F-016) 36
Tabel 4.17 Melihat Grafik Objek Wisata (SIMPADU-F-022) 36
Tabel 4.18 Melihat Daftar Wisatawan (SIMPADU-F-018) 37
Tabel 4.19 Verifikasi Objek Wisata (SIMPADU-F-019) 37
Tabel 4.20 Menambah Event (SIMPADU-F-020) 38
Tabel 4.21 Melihat Daftar Event (SIMPADU-F-010) 39
Tabel 4.22 Cetak Laporan Booking Objek Wisata (SIMPADU-F-008) 39
Tabel 4.23 Melihat Detail Booking Objek Wisata (SIMPADU-F-012) 40
Tabel 4.24 Melihat Detail Info Event (SIMPADU-F-011) 40
Tabel 4.25 Melihat Daftar Administrator (SIMPADU-F-023) 41
Tabel 4.26 Menambah Administrator (SIMPADU-F-025) 41
Tabel 4.27 Menghapus Administrator (SIMPADU-F-024) 42
Tabel 4.28 Mengupdate Event (SIMPADU-F-026) 42
Tabel 4.29 Mengupdate Objek Wisata (SIMPADU-F-017) 43
Tabel 5.1 Algoritma Fungsi register_process() 57
Tabel 5.2 Algoritma Fungsi login() 57
Tabel 5.3 Algoritma Fungsi tambah() 57
Tabel 5.4 Algoritma Fungsi update_process() 58
Tabel 5.5 Algoritma Fungsi informasi() 58
Tabel 6.1 Tabel Spesifikasi Perangkat Lunak 72
Tabel 6.2 Tabel Spesifikasi Perangkat Keras 72
Tabel 7.1 Tabel Kasus Uji Pengujian pada Login 87
Tabel 7.2 Tabel Kasus Uji Pengujian pada Regitrasi Wisatawan 88
Tabel 7.3 Tabel Kasus Uji Pengujian pada Regitrasi Pengelola 88

Tabel 7.4 Tabel Kasus Uji Pengujian pada Melihat Riwayat Booking Objek Wisata 89
Tabel 7.5 Tabel Kasus Uji Pengujian pada Melihat Detail Booking Objek Wisata 89
Tabel 7.6 Tabel Kasus Uji Pengujian pada Melakukan Booking Objek Wisata 90
Tabel 7.7 Tabel Kasus Uji Pengujian pada Cetak Laporan Booking Objek Wisata 90
Tabel 7.8 Tabel Kasus Uji Pengujian pada Melihat Detail Info Event 91
Tabel 7.9 Tabel Kasus Uji Pengujian pada Melihat Detail Info Objek Wisata 92
Tabel 7.10 Tabel Kasus Uji Pengujian pada Melihat Daftar Objek Wisata..... 92
Tabel 7.11 Tabel Kasus Uji Pengujian pada Melihat Daftar Event 93
Tabel 7.12 Tabel Kasus Uji Pengujian pada Menambah Event 93
Tabel 7.13 Tabel Kasus Uji Pengujian pada Menghapus Event..... 94
Tabel 7.14 Tabel Kasus Uji Pengujian pada Mengupdate Event..... 94
Tabel 7.15 Tabel Kasus Uji Pengujian pada Melihat Daftar Administrator..... 94
Tabel 7.16 Tabel Kasus Uji Pengujian pada Menambah Administrator..... 95
Tabel 7.17 Tabel Kasus Uji Pengujian pada Menghapus Administrator 95
Tabel 7.18 Tabel Kasus Uji Pengujian pada Verifikasi Objek Wisata 96
Tabel 7.19 Tabel Kasus Uji Pengujian pada Melihat Grafik Objek Wisata 96
Tabel 7.20 Tabel Kasus Uji Pengujian pada Melihat Daftar Wisatawan 97
Tabel 7.21 Tabel Kasus Uji Pengujian pada Menambah Objek Wisata..... 97
Tabel 7.22 Tabel Kasus Uji Pengujian pada Konfirmasi Booking Objek Wisata.... 98
Tabel 7.23 Tabel Kasus Uji Pengujian pada Melihat Grafik Jumlah Wisatawan ... 98
Tabel 7.24 Tabel Kasus Uji Pengujian pada Daftar Booking Objek Wisata 98
Tabel 7.25 Tabel Kasus Uji Pengujian pada Mengupdate Objek Wisata 99
Tabel 7.26 Hasil Pengujian Validasi dengan Black-Box 100
Tabel 7.27 Hasil Pengujian Compatibility..... 108

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi 7
Gambar 2.2 Waterfall Model Sommerville 14
Gambar 3.1 Langkah-langkah Penelitian 18
Gambar 4.1 Proses Bisnis Usulan 22
Gambar 4.2 Format Penomoran Kebutuhan 24
Gambar 4.3 Use Case Diagram Simpadu 30
Gambar 4.4 Activity Diagram Registrasi Wisatawan 44
Gambar 4.5 Activity Diagram Registrasi Pengelola 45
Gambar 4.6 Activity Diagram Menambah Objek Wisata 45
Gambar 4.7 Activity Diagram Menambah Event 46
Gambar 4.8 Activity Diagram Menambah Administrator 46
Gambar 4.9 Activity Diagram Konfirmasi Booking Objek Wisata 47
Gambar 4.10 Activity Diagram Verifikasi Objek Wisata 47
Gambar 4.11 Activity Diagram Cetak Laporan Booking Objek Wisata 48
Gambar 4.12 Activity Diagram Melakukan Booking Objek Wisata 48
Gambar 5.1 Sequence Diagram Registrasi Wisatawan 49
Gambar 5.2 Sequence Diagram Registrasi Pengelola 50
Gambar 5.3 Sequence Diagram Menambah Administrator 50
Gambar 5.4 Sequence Diagram Menambah Event 51
Gambar 5.5 Sequence Diagram Menambah Objek Wisata 52
Gambar 5.6 Sequence Diagram Booking Objek Wisata 53
Gambar 5.7 Sequence Diagram Verifikasi Objek Wisata 53
Gambar 5.8 Sequence Diagram Konfirmasi Booking Objek Wisata 54
Gambar 5.9 Sequence Diagram Cetak Laporan Booking Objek Wisata 54
Gambar 5.10 Class Diagram 56
Gambar 5.11 Physical Data Models 59
Gambar 5.12 Halaman Login 59
Gambar 5.13 Halaman Grafik Objek Wisata 60
Gambar 5.14 Halaman Objek Wisata 60
Gambar 5.15 Halaman Detail Objek Wisata 61
Gambar 5.16 Halaman Event 61
Gambar 5.17 Halaman Tambah Event 62
Gambar 5.18 Halaman Detail Event 62
Gambar 5.19 Halaman Wisatawan 63
Gambar 5.20 Halaman Administrator 63
Gambar 5.21 Halaman Tambah Administrator 64
Gambar 5.22 Halaman Register Pengelola 64
Gambar 5.23 Halaman Objek Wisata 65
Gambar 5.24 Halaman Detail Objek Wisata 65
Gambar 5.25 Halaman Tambah Objek Wisata 66
Gambar 5.26 Halaman Daftar Booking 66
Gambar 5.27 Halaman Grafik Wisatawan 67

Gambar 5.28 Halaman Register Wisatawan 67
Gambar 5.29 Halaman Objek Wisata 68
Gambar 5.30 Halaman Detail Objek Wisata 68
Gambar 5.31 Halaman Event 69
Gambar 5.32 Halaman Detail Event 69
Gambar 5.33 Halaman Riwayat Booking Objek Wisata 70
Gambar 5.34 Halaman Detail Booking Objek Wisata 70
Gambar 5.35 Halaman Detail Booking Objek Wisata 71
Gambar 5.36 Halaman Booking Objek Wisata 71
Gambar 6.1 Halaman Home 73
Gambar 6.2 Halaman Login 74
Gambar 6.3 Halaman Registrasi Wisatawan 74
Gambar 6.4 Halaman Registrasi Pengelola 75
Gambar 6.5 Halaman Grafik Objek Wisata 75
Gambar 6.6 Halaman Objek Wisata (Admin) 76
Gambar 6.7 Halaman Detail Objek Wisata (Admin) 76
Gambar 6.8 Halaman Event (Admin) 77
Gambar 6.9 Halaman Detail Event (Admin) 77
Gambar 6.10 Halaman Wisatawan 78
Gambar 6.11 Halaman Administrator 78
Gambar 6.12 Halaman Tambah Administrator 79
Gambar 6.13 Halaman Objek Wisata (Wisatawan) 79
Gambar 6.14 Halaman Detail Objek Wisata (Wisatawan) 80
Gambar 6.15 Halaman Event (Wisatawan) 80
Gambar 6.16 Halaman Detail Event (Wisatawan) 81
Gambar 6.17 Halaman Riwayat Booking Objek Wisata 81
Gambar 6.18 Halaman Detail Booking Objek Wisata 82
Gambar 6.19 Halaman Cetak Laporan Booking Objek Wisata 82
Gambar 6.20 Halaman Booking Objek Wisata 83
Gambar 6.21 Halaman Objek Wisata (Pengelola) 83
Gambar 6.22 Halaman Detail Objek (Pengelola) 84
Gambar 6.23 Halaman Tambah Objek Wisata 84
Gambar 6.24 Halaman Daftar Booking Objek Wisata 85
Gambar 6.25 Halaman Grafik Wisatawan 85
Gambar 6.26 Halaman Update Event 86
Gambar 6.27 Halaman Update Objek Wisata 86
Gambar 7.1 Flowgraph Tambah Event 102
Gambar 7.2 Flowgraph Tambah Objek Wisata 104
Gambar 7.3 Flowgraph Update Informasi Objek Wisata 106
Gambar 7.4 Hasil Pengujian Compatibility dengan Sortsite 107

DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA..... 112

BAB 1 PENDAHULUAN

1.1 Latar belakang

Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi merupakan sebuah organisasi yang berdiri di bawah naungan Pemerintah Daerah Kabupaten Banyuwangi mempunyai tugas pokok untuk melaksanakan urusan pemerintah daerah di bidang kebudayaan dan pariwisata (Peraturan Bupati Banyuwangi Nomor 56, 2016).

Seiring dengan meningkatnya jumlah wisatawan yang berkunjung di berbagai tempat wisata di Kabupaten Banyuwangi, hal ini dibuktikan dengan jumlah wisatawan pada tahun 2015 yang mengunjungi masing-masing tempat wisata di Kabupaten Banyuwangi yang didapat dari hasil wawancara di Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi (Data Pengunjung Obyek dan Wisata, 2015). Oleh karena itu, wisatawan memerlukan referensi informasi tempat wisata di Kabupaten Banyuwangi secara terperinci agar dapat mengetahui apa saja yang ada pada tempat wisata tersebut.

Dari hasil wawancara penulis pada Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi, untuk saat ini Kabupaten Banyuwangi menggunakan sebuah *website* resmi dari Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi yang belum mencakup semua informasi yang diperlukan oleh wisatawan dan juga terdapat beberapa konten-konten yang belum sepenuhnya berjalan sesuai dengan kebutuhannya sebagai media promosi pariwisata yang terdapat di Kabupaten Banyuwangi (Hasil Wawancara dengan Kepala Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi, 2016). Akan tetapi dalam memperoleh informasi wisata pada *website* tersebut wisatawan sering mengalami kesulitan karena tidak mengetahui dimana dan pada siapa mereka harus mengetahui informasi destinasi wisata yang dituju dikarenakan pada website hanya terdapat informasi singkat tentang tempat wisata yang ada di Kabupaten Banyuwangi.

Pada umumnya cara mengetahui destinasi tempat wisata yang ingin dikunjungi para wisatawan harus mengunjungi Dinas Kebudayaan dan Pariwisata kabupaten Banyuwangi karena terdapat *Tourist Information Center* yang melayani pertanyaan seputar informasi pariwisata di Kabupaten Banyuwangi. Oleh karena itu, dengan alur yang demikian membuat para wisatawan kebanyakan bertanya lokasi tempat wisata yang ingin dikunjungi kepada warga sekitar destinasi tempat wisata daripada bertanya langsung di *Tourist Information Center* yang disediakan oleh Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi. Hal ini dikarenakan masih kurangnya sosialisasi yang dilakukan oleh Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi mengenai adanya *Tourist Information Center* tersebut (Hasil Wawancara dengan Kepala Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi, 2016).

Dalam melakukan promosi pariwisata pihak Dinas Kebudayaan dan Pariwisata telah bekerja sama dengan para netizen yang ada di Kabupaten Banyuwangi dan

dengan bagian humas dan protokol Pemerintah Kabupaten Banyuwangi melalui media sosial dalam memberikan informasi-informasi tentang pariwisata. Akan tetapi hal tersebut masih sedikit kurang menarik minat para wisatawan luar banyuwangi untuk berkunjung ke Kabupaten Banyuwangi dikarenakan informasi-informasi yang diberikan hanya informasi umum tentang objek wisata yang ada di Kabupaten Banyuwangi. Sehingga sangat perlu sekali untuk menerapkan sistem informasi manajemen dalam memudahkan wisatawan dalam melihat destinasi dan hal-hal lain yang terkait dengan pariwisata yang ada di Kabupaten Banyuwangi (Hasil Wawancara dengan Kepala Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi, 2016).

Singkatnya kebutuhan informasi di bidang pariwisata Kabupaten Banyuwangi masih belum berjalan dengan baik, dikarenakan media informasi saat ini masih belum dikelola dengan baik dan terperinci serta belum keseluruhan daerah menerapkan jaringan fiber optik (Hasil Wawancara dengan Kepala Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi, 2016). Disamping kesiapan dari sistem pengelola data maka orang yang membangun struktur sistem informasi ini harus benar-benar mengerti kebutuhan pengguna data tersebut, karena informasi pariwisata memiliki karakteristik data yang sangat beragam seperti objek dan daya tarik, data hotel, data sarana transportasi, dan data-data fasilitas lain, hingga ke data statistik seperti jumlah wisatawan dan pemandu wisatanya, perlu dikelola secara terintegrasi. Data-data ini juga sangat dinamis, sehingga kompleks dalam pemilahannya, serta harus diperhatikan masalah keakuratan atau kebenaran datanya. Kegunaan dari setiap data juga harus diperhatikan berdasarkan segmen pasar penggunanya. Maka diperlukan sebuah sistem informasi manajemen pariwisata yang menyediakan informasi tentang tujuan wisata, obyek wisata yang menarik, sarana yang tersedia seperti transportasi untuk mencapai daerah tujuan wisata, produk wisata yang diminati, pemesanan tiket secara online dan lain sebagainya (Hasil Wawancara dengan Kepala Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi, 2016).

Suatu Sistem Informasi Manajemen (SIM) dapat membantu baik pemerintah daerah Kabupaten Banyuwangi maupun industri/ pelaku pariwisata yang ada di Kabupaten Banyuwangi. Sejalan dengan keinginan pemerintah daerah Kabupaten Banyuwangi untuk memajukan industri pariwisata yang dimilikinya maka tentu ada keinginan besar untuk menata informasi data pariwisata sebaik-baiknya agar masyarakat yang membutuhkan dapat memperoleh dengan cepat, akurat dan dapat disebarluaskan dengan mudah. Untuk bidang pariwisata maka SIM dapat digunakan untuk mengelola data yang dapat dimanfaatkan oleh wisatawan, industri pariwisata maupun pemerintah daerah.

Berdasarkan permasalahan yang terjadi pada Pemerintah Daerah Kabupaten Banyuwangi khususnya pada Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi mengenai media promosi pariwisata pada Kabupaten Banyuwangi, penulis melakukan pengembangan sistem informasi dengan menggunakan metode *Object Oriented Design* (OOD) dalam bentuk penulisan skripsi dengan judul "**Pengembangan Sistem Informasi Manajemen Pariwisata Terpadu Sebagai**

Media Promosi Kabupaten Banyuwangi" sebagai bentuk media promosi untuk meningkatkan dan mempertahankan jumlah wisatawan yang berkunjung di Kabupaten Banyuwangi serta memberikan informasi objek wisata yang terdapat di Kabupaten Banyuwangi secara rinci.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka penulis merumuskan masalah yang ada yaitu :

1. Bagaimana proses analisis kebutuhan sistem dalam pengembangan sistem informasi manajemen terpadu sebagai media promosi Kabupaten Banyuwangi?
2. Bagaimana merancang dan mengimplementasikan menggunakan metode *Object Oriented Design* untuk mengembangkan sistem informasi manajemen pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi?
3. Bagaimana menguji sistem informasi manajemen pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi?

1.3 Tujuan

Tujuan yang ingin dicapai pada pembuatan skripsi ini, yaitu:

1. Mengetahui hasil analisis kebutuhan dalam pengembangan sistem informasi manajemen terpadu sebagai media promosi Kabupaten Banyuwangi.
2. Melakukan perancangan dan implementasi menggunakan metode *Object Oriented Design* untuk mengembangkan sistem informasi manajemen pariwisata sebagai media promosi Kabupaten Banyuwangi.
3. Mengetahui hasil pengujian sistem informasi manajemen pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi berdasarkan metode yang diterapkan.

1.4 Manfaat

1. Bagi Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi

Dengan adanya sistem informasi manajemen pariwisata terpadu, Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi dapat melakukan promosi objek wisata dan *event* yang ada di Kabupaten Banyuwangi sehingga akan meningkatkan jumlah wisatawan yang berkunjung di Kabupaten Banyuwangi.

2. Bagi Wisatawan
   a. Memberikan informasi destinasi wisata yang ada di Kabupaten Banyuwangi.
   b. Memberikan informasi *event* yang diselenggarakan di Kabupaten Banyuwangi.
   c. Dapat melakukan booking objek wisata yang ada di Kabupaten Banyuwangi.

3. Bagi Penulis

Dapat menambah pengetahuan dan wawasan serta dapat mengaplikasikan dan mensosialisasikan teori yang telah diperoleh selama perkuliahan.

1.5 Batasan masalah

Batasan penelitian uang dilakukan pada penulisan skripsi ini memiliki batasan masalah, yaitu:

1. Sistem Informasi Manajeman Pariwisata Terpadu sebagai media promosi Kabupaten Banyuwangi berbasis *website*.
2. Objek penelitian dan pengambilan data terbatas pada sektor pariwisata kabupaten Banyuwangi di Dinas Kebudayaan dan Pariwisata kabupaten Banyuwangi.
3. Sumber data yang dihasilkan diambil dari hasil wawancara di Dinas Pariwisata dan Kebudayaan kabupaten Banyuwangi.
4. Perancangan menggunakan metode *Object Oriented Design*.
5. Implementasi sistem informasi menggunakan bahasa pemrograman *PHP* dan *database MySQL*.
6. Tahap Pengujian difokuskan pada 3 metode pengujian, yaitu Pengujian *Black-Box*, Pengujian *White-Box* dan Pengujian *Compatibility*.

1.6 Sistematika pembahasan

Penyusunan penelitian ini menggunakan kerangka pembahasan yang tersusun sebagai berikut:

1. BAB 1 PENDAHULUAN
   Penjabaran tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan pada penelitian.
2. BAB 2 LANDASAN KEPUSTAKAAN
   Menguraikan isi dari tinjauan pustaka dan pokok-pokok teori yang terkait dengan penelitian dari berbagai referensi yang dijadikan sebagai bahan dasar untuk penelitian.
3. BAB 3 METODOLOGI
   Menjelaskan tentang langkah-langkah atau metode penelitian dalam study literatur, observasi dan wawancara, analisis kebutuhan, perancangan, implementasi, dan pengujian serta metode-metode lain yang digunakan untuk penelitian.
4. BAB 4 ANALISIS KEBUTUHAN
   Menjelaskan tentang analisis kebutuhan sistem informasi manajemen pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi sesuai dengan kebutuhan yang dibutuhkan dalam melakukan penelitian ini.

5. BAB 5 PERANCANGAN
   Menjelaskan tentang perancangan sistem informasi manajemen pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi menggunakan metode *object oriented design* sesuai dengan analisis kebutuhan sistem informasi dalam melakukan penelitian ini.
6. BAB 6 IMPLEMENTASI
   Menjelaskan tentang proses implementasi dalam pembangunan sistem informasi pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi menggunakan bahasa pemrograman *php* serta *database MySQL* dan hasil impementasi.
7. BAB 7 PENGUJIAN DAN ANALISIS
   Menjelaskan tahap pengujian dan analisis sistem informasi manajemen yang telah diimplementasikan dengan melihat *output* (keluaran) sistem informasi manajemen dibandingkan dengan hasil analisis kebutuhan dan perancangan.
8. BAB 8 PENUTUP
   Menjelaskan tentang kesimpulan dari pengembangan sistem informasi manajemen pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi dan saran maupun rekomendasi atas penelitian yang telah dilakukan.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian tentang Pengembangan Sistem Informasi Manajemen Pariwisata Terpadu Sebagai Media Promosi Kabupaten Banyuwangi belum pernah dilakukan pengkajian sebelumnya. Namun, penulis mengambil beberapa contoh penelitian yang memiliki teori dan konsep yang sama, akan tetapi memiliki perbedaan dengan penelitian yang dilakukan oleh penulis. Berikut ini merupakan beberapa penelitian terkait penelitian yang dilakukan oleh penulis, yaitu:

"Pengembangan Sistem Informasi Manajemen Akademik Terintegrasi (Studi Kasus Perguruan Tinggi XYZ)" oleh A'ang Subiyakto, M. Qomarul Huda, dan W.S. Prasetyo dari Program Studi Sistem Informasi Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, dimana pengembangan sistem informasi ini dilakukan dengan melakukan integrasi sistem kerja Bagian Akademik dan Kemahasiswaan sesuai dan sejalan dengan karakteristik organisasi di P.T. XYZ. Pengembangan ini menggunakan *System Development Life Cycle* (SDLC), dengan menggunakan software desain *M.S. Visio* 2002 dan *MySQLyog Enterprise Edition*, serta bahasa pemrograman yg digunakan yaitu *M.S. Visual Basic 6.0* dan *MySQL 5.0* sebagai bahasa pemrograman *database*.

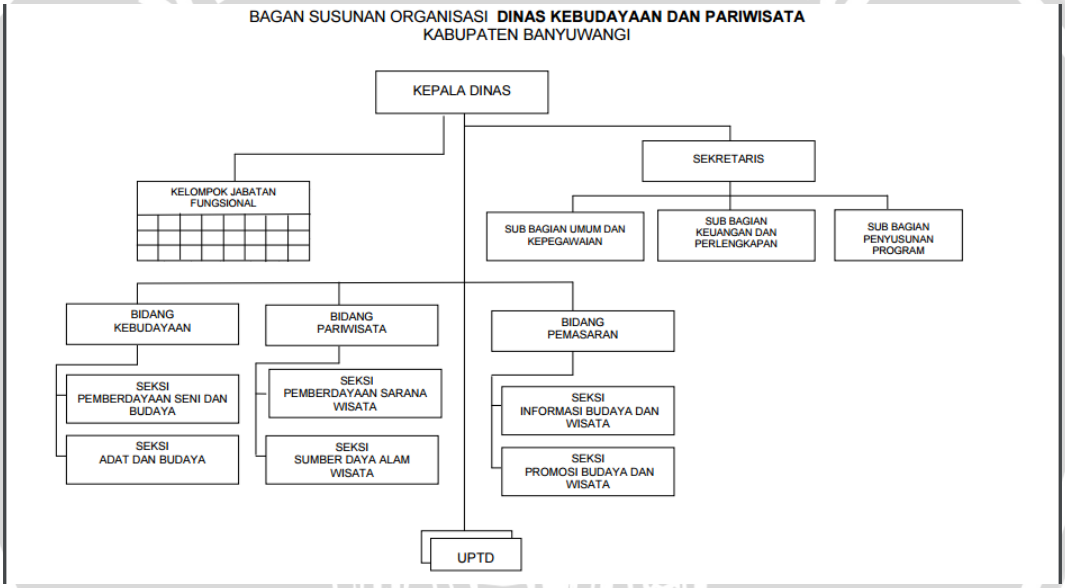
Penelitian yang kedua "Pengembangan Sistem Informasi Manajemen Pajak Berbasis Finacial Information System Model : Suatu Kajian Pendahuluan" oleh Agung Dariono dari Program Studi Teknik Informatika Universitas Islam Indonesia Yogyakarta, dimana dilakukan pengembangan model sistem informasi manajemen pajak yang bertujuan untuk menyediakan informasi perpajakan secara komprehensif sehingga dapat mencapai tujuan manajemen pajak sehingga sejalan dengan tujuan organisasi. Pengembangan ini menggunakan *System Development Life Cylce* (SDLC) dan metode *Financial Information System Model*, selanjutnya dapat digunakan untuk melakukan analisis kebutuhan pemakai dengan pendekatan user-requirement yang sesungguhnya.

Sedangkan penelitian yang dilakukan oleh penulis melakukan integritas antara pengelola wisata, wisatawan, dan juga Dinas Kebudayaan dan Pariwisata di Kabupaten Banyuwangi sehingga pengelolaan informasi objek wisata dapat lebih tertata sesuai kebutuhan pariwisata di Kabupaten Banyuwangi yang bertujuan sebagai media promosi pariwisata demi meningkatkan dan mempertahankan jumlah wisatawan yang berkunjung ke Kabupaten Banyuwangi. Pengembangan sistem informasi menggunakan *System Development Life Cycle* (SDLC) dan metode *Object Oriented Design* (OOD), dengan menggunakan software desain *Astah Professional*, serta bahasa pemograman yang digunakan yaitu bahasa pemrograman *PHP* dengan *framework CodeIgniter* dan *MySQL* sebagai bahasa pemrograman *database*.

2.2 Dasar Teori

2.2.1 Profil Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi

Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi merupakan dinas yang berdiri di bawah naungan Pemerintah Daerah Kabupaten Banyuwangi yang memiliki tugas pokok melakukan urusan pemerintahan daerah di bidang Kebudayaan dan Pariwisata Kabupaten Banyuwangi. Dinas Kebudayaan dan Pariwisata dikepalai oleh Kepala Dinas yang memiliki jajaran dibawahnya, yaitu: Sekretariat, Bidang Kebudayaan, Bidang Pariwisata, Bidang Pemasaran, UPTD, dan Kelompok Jabatan Fungsional (Peraturan Bupati Banyuwangi Nomor 56, 2016). Sistem informasi manajemen pariwisata terpadu digunakan oleh bidang pemasaran yang membawahi seksi informasi budaya dan wisata dan juga seksi promosi budaya dan wisata. Untuk lebih jelasnya dapat dilihat pada struktur organisasi Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi pada gambar 2.1.



Gambar 2.1 Struktur Organisasi Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi

Sumber: Peraturan Daerah Kabupaten Banyuwangi Nomor 6 Tahun 2011

2.2.2 Sistem

Sistem merupakan kumpulan dari beberapa elemen yang saling melakukan interaksi untuk mencapai suatu tujuan yang diharapkan. Di dalam sistem juga dapat digambarkan dari suatu kejadian dan kesatuan yang nyata (Jogiyanto, 2005).

Jaringan kerja dari beberapa prosedur yang saling terkait, berkumpul menjadi satu dalam melakukan suatu kegiatan untuk menyelesaikan suatu sasaran tertentu merupakan sebuah sistem (Andi Kristanto, 2007).

Berdasarkan pendapat diatas maka penulis dapat menyimpulkan bahwa sistem adalah kumpulan dari beberapa elemen yang saling melakukan interaksi dan saling terkait, berkumpul menjadi satu untuk mencapai suatu tujuan tertentu.

2.2.3 Sistem Informasi

Menurut Jogiyanto (2005) didalam bukunya yang berjudul "Analisis dan Desain Sistem informasi" bahwa suatu sistem yang berjalan pada sebuah organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung semua operasi, bersifat manajerial dan menyediakan pihak luar tertentu dengan kumpulan laporan yang dibutuhkan disebut juga dengan sistem informasi.

2.2.4 Sistem Informasi Manajemen

Sistem Informasi Manajemen menurut Turban, McLean, serta Waterbe (1999) didalam bukunya yang berjudul *Information Technology for Management: Making Connection for Strategic Advantages* adalah suatu sistem yang digunakan untuk mengumpulkan (*collect*), memproses (*processing*), menyimpan (*save*), menganalisa (*analyze*), dan juga menyebarkan (*spread*) sebuah kumpulan data yang berbentuk informasi untuk tujuan yang lebih spesifik.

Sedangkan Bodnar dan Hopwood (1993) didalam bukunya yang berjudul *Accounting Information System* berpendapat bahwa suatu kumpulan perangkat keras dan perangkat lunak yang dirancang untuk dapat mentransformasikan sebuah data dalam bentuk informasi yang dapat berguna bagi *user* disebut sistem informasi manajemen.

2.2.4.1 *Tujuan Sistem Informasi Manajemen*

1. Dapat menyediakan suatu informasi yang akan digunakan dalam suatu perhitungan harga pokok jasa, produk, serta tujuan lainnya yang diinginkan manajemen.
2. Menyediakan suatu informasi yang digunakan dalam suatu perencanaan, pengendalian, pengevaluasian, serta juga perbaikan berkelanjutan.
3. Menyediakan suatu informasi untuk pengambilan suatu keputusan oleh suatu perusahaan.

2.2.4.2 *Proses Manajemen*

1. Perencanaan, untuk dapat menggapai suatu tujuan akhir tertentu dapat diperoleh dari suatu aktivitas manajemen yang disebut dengan perencanaan. Oleh sebab itu , perencanaan memiliki suatu penetapan tujuan serta juga identifikasi metode untuk dapat mencapai tujuan tersebut.
2. Pengendalian, setelah suatu rencana tersebut telah dibuat, rencana itu harus diimplementasikan, serta seseorang juga harus dapat melakukan monitor proses implementasinya untuk dapat memastikan bahwa rencana tersebut berjalan sesuai dengan rencana.

3. Pengambilan Keputusan, sebuah proses pemilihan dari berbagai *alternative* yang ada disebut dengan proses pengambilan keputusan. Hal ini dilakukan agar dapat melakukan pengambilan keputusan dari suatu jalinan antara perencanaan serta juga pengendalian.

2.2.5 Promosi

Menurut Boone dan Kurtz (2002) promosi merupakan sebuah proses yang digunakan untuk menginformasikan, membujuk, dan mempengaruhi suatu keputusan pembelian.

Sedangkan menurut Swastha (2000), promosi dipandang sebagai sebuah arus informasi atau komunikasi satu arah yang dibuat untuk mempengaruhi seseorang atau organisasi terhadap tindakan yang menciptakan pertukaran dalam pemasaran.

2.2.6 Business Process

Davenport (1993) berpendapat bahwa sebuah proses aktivitas yang terukur dan terstruktur untuk melakukan produksi *output* tertentu bagi kalangan pelanggan tertentu lebih dikenal dengan proses bisnis. Suatu proses bisnis merupakan sebuah urutan yang lebih spesifik dari aktivitas kerja lintas waktu dan ruang, dengan suatu awalan dan akhiran, dan dapat secara jelas mendefinisikan *input* dan *output*.

Dalam pembuatan sebuah proses bisnis dapat dilakukan dengan cara menggunakan *Business Process Modeling Notation* (BPMN) yang merupakan sebuah notasi grafis yang menggambarkan logika dari langkah-langkah dalam proses bisnis yang akan digambarkan.

Ada Lima (5) kategori dasar elemen dalam BPMN, yaitu:

1. *Flow Objects* adalah elemen grafis utama untuk menentukan perilaku aktor dan sistem dalam proses bisnis. Ada tiga (3) elemen *Flow Objects*, yaitu *Events*, *Activities*, dan *Gateways*.
2. *Data* dapat direpresentasikan dengan empat (4) elemen Data, yaitu *Data Objects*, *Data Inputs*, *Data Outputs*, dan *Data Stores*.
3. *Connecting Objects* merupakan cara menghubungkan arus informasi satu sama lain atau lainnya. Ada empat (4) elemen *Connecting Objects*, yaitu *Sequence Flows*, *Message Flows*, *Associations*, dan *Data Associations*.
4. *Swimlanes* merupakan cara pengelompokan unsur –unsur pemodelan utama. Ada dua (2) bentuk, yaitu *Pools* dan *Lanes*.
5. *Artifacts* digunakan untuk memberikan informasi tambahan tentang proses. Ada dua (2) model, yaitu *Group* dan *Text Annotation*.

2.2.7 Software Engineering

Menurut Ian Sommerville (2011) dalam bukunya yang berjudul *Software Engineering*, *Software Engineering* salah satu disiplin ilmu yang melakukan

pembahasan semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi dari sebuah sistem sampai pemeliharaan dari sebuah sistem setelah digunakan, dimana ada dua (2) kata kunci yaitu:

1. Disiplin Rekayasa, dimana perekayasa menerapkan teori, metode dan alat bantu yang sesuai serta penggunaan secara selektif dan selalu mencoba mencari solusi terhadap permasalahan walaupun tidak ada teori atau metode yang mendukung.
2. Semua aspek produksi perangkat lunak, rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode, dan teori untuk mendukung produksi perangakt lunak.

2.2.8 Unified Modeling Language (UML)

Unified Modeling Language (UML), sebuah metodologi yang umum digunakan untuk menentukan dan menggambarkan proyek yang kompleks. Metodologi yang paling sering digunakan untuk saat ini sebagai analisa dan perancangan sistem yaitu dengan menggunakan sebuah metodologi yang berorientasi objek (Nugroho, 2009).

UML juga menyediakan 9 (sembilan) standar diagram yang memiliki sifat statis dan dinamis dan juga dapat digunakan untuk menggambarkan sebuah sistem dari berbagai sudut pandang. Diagram tersebut antara lain : *class diagram, object diagram, use case diagram, sequence diagram, collaboration diagram, state chart diagram, activity diagram, component diagram,* dan *deployment diagram.*

2.2.8.1 *Class Diagram*

Sekumpulan dari objek-objek yang sejenis biasanya disebut *Class diagram*. Dalam sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* adalah dimana kondisi objek tersebut dinyatakan dalam *attribute/properties*. Sedangkan perilaku (*behavior*) digunakan untuk mendefinisikan bagaimana sebuah objek dapat melakukan aksi dan juga memberikan reaksi (Munawar, 2005).

Dalam setiap kelas memiliki tiga area pokok:

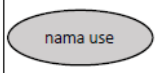
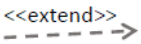
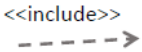
- Nama dan *stereotype*
- Atribut merupakan variabel yang dimiliki oleh suatu kelas.
- Metode atau operasi merupakan fungsi yang dimiliki oleh suatu kelas.

2.2.8.2 *Use Case Diagram*

Use Case diagram juga digunakan dalam mendeskripsikan interaksi antara aktor dari sistem dengan sistem, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Dan dapat digunakan juga untuk menunjukkan aktor mana yang menggunakan *use case* mana, *uses case* mana

yang memasukkan *use case* lain serta hubungan antara aktor dan *use case* (Martin Fowler, 2005).

Tabel 2.1 Simbol-simbol *Use Case Diagram*

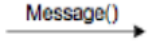
Nama	Simbol	Deskripsi
Use case		Use case menggambarkan tindakan yang di lakukan oleh aktor. Use case di tulis dengan mengunakan kata kerja di awal.
Aktor		Orang, sistem yang berinteraksi dengan sistem. Aktor dijadikan sebagai peranan yang dapat berinteraksi dengan sistem.
Relationship		Relationship menggambarkan komunikasi antara actor dan usecase atau usecase dengan actor yang ada pada sistem.
Extend Relationship		Hubungan antara use case dengan use case tambahan yang bersifat optional. Use casa dapat berdiri sendiri tanpa ada nya use case tambahan.
Include Relationship		*Include* merupakan hubungan antara use case dengan use case lain yang mengharuskan sebuah use case dijalankan terlebih dahulu agar dapat menjalankan use case yang lain.

Sumber: Booch (2007)

2.2.8.3 *Sequence Diagram*

Sequence diagram adalah diagram yang digambarkan dalam sebuah grafik yang berbentuk dua dimensi dimana sebuah obyek ditunjukkan dalam posisi horizontal, sedangkan *lifeline* ditunjukkan dalam posisi vertikal (Munawar, 2005).

Tabel 2.2 Simbol-simbol *Sequence Diagram*

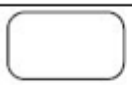
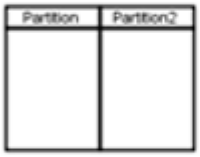
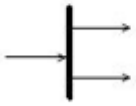
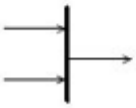
No.	Gambar	Nama	Keterangan
1		*Actor*	Mengambarkan orang yang berinteraksi dengan sistem.
2		*Boundary*	Digunakan untuk menggambarkan semuah form
3		*Control*	Mengatur, menangani tugas utama dari alur sistem
4		*Entity*	Menggambarkan informasi yang harus disimpan oleh sistem
5		*Lifeline*	Mengindikasikan keberadaan sebuah objek dalam basis awktu tertentu
6		*Message*	Hhubunga antar objek satu dengan objek lain nya yang menunjukan suatu kejadian

Sumber: Prayitno (2005)

2.2.8.4 *Activity Diagram*

Activity diagram merupakan cara yang digunakan untuk menunjukkan sebuah gambaran dari sebuah logika prosedural, proses bisnis, dan jalur kerja dari sebuah sistem (Martin Fowler, 2005).

Tabel 2.3 Simbol-simbol *Activity Diagram*

Simbol	Nama	Deskripsi
	InitialNode	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	*Action*	Aktivitas yang dilakukan oleh sistem, aktivitas biasayang diawali dengan kata kerja
	ActivityFinal	Status akhir yan dilakukan oleh sistem, sebuah diagram aktivitas memiliki seuah status akhir
	*Swimlane*	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
	*ForkNode*	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel
	*JoinNode*	Digunakan untuk menunjukkan kegiatan yang digabungkan
	ControlFlow	Menunjukkan bagaimana kendali suatu aktivitas terjadi pada aliran kerja dalam tindakan tertentu

Sumber: Dennis (2012)

2.2.9 System Development Life Cycle (SDLC)

System development life cycle (SDLC), sebuah model pendekatan yang memiliki tahapan-tahapan dalam melakukan sebuah analisa dan membangun rancangan sistem dengan menggunakan siklus yang terperinci terhadap kegiatan penggunanya (Kendall & Kendall, 2006).

Pada dasarnya *system development life cycle* (SDLC) terdiri dari lima fase, yaitu perencanaan, analisis, desain, implementasi dan perawatan. Berikut penjelasan 5 (lima) fase menurut Kendall & Kendall (2006):

1. Perencanaan

   Pada fase perencanaan akan dibagi dan dijelaskan menjadi beberapa bagian, yaitu identifikasi masalah, peluang dan tujuan.

a. Identifikasi masalah dilakukan dengan cara melihat kenyataan yang terjadi dalam suatu perusahaan.
b. Identifikasi peluang dilakukan untuk mengetahui proses apa saja yang dapat diubah menjadi lebih baik dengan adanya sistem terkomputerisasi.
c. Identifikasi tujuan dilakukan untuk mengetahui tujuan apa yang ingin dicapai perusahaan.

2. Analisis

Pada fase analisis sistem dibagi menjadi dua tahap:

a. *Determining Information Requirements*

Dilakukan proses analisis sistem untuk menentukan syarat-syarat informasi. Pengumpulan data mengenai kebutuhan user dapat dilakukan dengan cara wawancara, kuisioner, sampling, dan menganalisis hard data.

b. *Analyzing System Needs*

Menganalisis kebutuhan sistem dengan menguraikan suatu sistem informasi yang utuh ke dalam komponen-komponennya untuk mengevaluasi peluang, dan hambatan yang terjadi.

3. Desain

Dilakukan untuk membuat sebuah desain prosedur bagi pengguna untuk melakukan entri data secara akurat dan efektif, membuat desain interface bagi pengguna, membuat desain database yang akan digunakan untuk menyimpan data yang diperlukan oleh pengambil keputusan, membuat desain output baik onscreen atau tercetak, dan yang terakhir membuat desain kontrol dan prosedur backup.

4. Implementasi

Pada fase implementasi dibagi menjadi 3 bagian yang terdiri dari:

a. *Developing and Documenting Software*

Dilakukan proses untuk melakukan pengembangan dan dokumentasi sebuah sistem dimana aktivitas yang dilakukan adalah mengembangkan sistem melalui kerjasama sistem analis dengan programer, sedangkan dalam mengembangkan dokumentasi melalui kerjasama antara sistem analis dengan pengguna.

b. *Testing and Maintaining The Sistem*

Sebelum sebuah sistem dapat digunakan oleh user, maka harus dilakukan sebuah pengujian yang dilakukan oleh programer dan sistem analis, jika terjadi sebuah masalah akan segera dapat dilakukan perbaikan sistem.

c. *Implemeting and Evaluating The Sistem*

Pada fase ini dilakukan 3 proses, yaitu proses instalasi sistem, konversi sistem, dan evaluasi sistem.

5. Perawatan

Pada fase perawatan dibagi menjadi 2 proses, yaitu:

a. *Testing and Maintaining The Sistem*

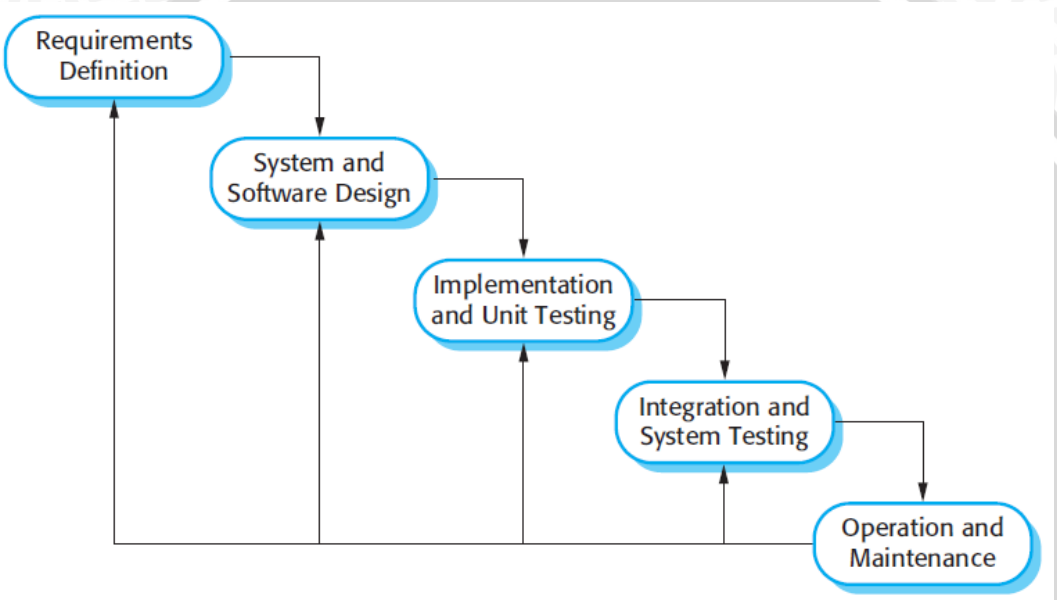
Proses yang dilakukan dengan melakukan pengetasan pada sistem apakah sistem mengalami kegagalan (*bug*) dan sekaligus dilakukan perawatan terhadap sistem.

b. Implemeting and Evaluating The Sistem

Proses yang dilakukan dengan melakukan pelatihan terhadap user yang akan ditugaskan untuk menggunakan sistem, melakukan konversi sistem, dan melakukan evaluasi sistem.

2.2.7.1 Waterfall Model

Waterfall model digunakan dalam kegiatan proses dasar spesifikasi, pengembangan, validasi, dan pembaharuan sistem dan juga mewakili kegiatan tersebut sebagai sebuah fase proses terpisah seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian, dan sebagainya (Sommerville, 2011).



Gambar 2.2 Waterfall Model Sommerville

Sumber: Sommerville (2011)

Tahapan-tahapan pengembangan perangkat lunak yang terdapat dalam *waterfall model* antara lain :

1. ***Requirement Analysis and Definition***
   Merupakan tahap penetapan *fitur*, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem.
2. ***System and Software Design***
   Dalam tahap ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar sistem perangkat lunak dan hubungan-hubungannya.
3. ***Implementation and Unit Testing***
   Dalam tahap ini, hasil dari desain perangkat lunak akan disatukan sebagai satu *set program* atau *unit program*. Setiap *unit* akan diuji apakah sudah memenuhi spesifikasinya.
4. ***Integration and System Testing***

Dalam tahap ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada.

5. ***Operation and Maintenance***
   Dalam tahap ini, sistem sudah siap untuk diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan.

2.2.10 PHP (PHP: Hypertext Preprocessor)

Sebuah bahasa server-side-scripting yang telah menyatu dengan HTML pada halaman web yang dinamis biasa disebut Hypertext Preprocessor (PHP). Karena PHP menggunakan bahasa server-side-scripting maka sintaks dan perintah-perintah PHP akan diesksekusi oleh server yang kemudian hasilnya akan dikirimkan ke browser dengan format HTML (Arief, 2011).

Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman website lebih terjamin. PHP dirancang untuk membuat halaman website yang dinamis, yaitu halaman website yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman website.

2.2.11 MySQL

MySQL, salah satu jenis *database server* yang sangat terkenal di kalangan programer dan juga telah banyak digunakan dalam membangun sebuah aplikasi berbasis *website* dengan menggunakan *database* sebagai sumber dan pengolahan datanya (Arief, 2011).

Database yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (*PHP* dan *Perl*) yaitu *MySQL*. *MySQL* lebih sering digunakan dalam membangun aplikasi berbasis *website*, pada umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *PHP*.

Sebagai salah satu *database server*, *MySQL* lebih unggul dibandingkan dengan *database server* lainnya dalam hal *query* data (Andri Kristanto, 2010).

2.2.12 Framework CodeIgniter

CodeIgniter, sebuah *framework PHP* dengan ukuran *footprint* yang sangat kecil dan dibangun untuk digunakan oleh pengembang yang membutuhkan *toolkit* sederhana dan elegan dalam membuat sebuah aplikasi berbasis *website* dengan fitur yang lengkap (Mutasim Ridlo, 2016).

Dengan menggunakan *framework CodeIgniter*, pemrogram tidak lagi perlu membuat dari awal (*from scratch*), karena *CodeIgniter* telah menyediakan banyak *library* yang diperlukan oleh pemrogram, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk dapat mengakses librarinya.

CodeIgniter sendiri dibangun menggunakan konsep *Model-View-Controller development pattern*. *CodeIgniter* merupakan salah satu *framework* tercepat dibandingkan dengan *framework* lainnya.

2.2.13 Pengujian *Black-Box*

Pengujian *Black-Box,* sebuah pengujian yang fokus pada kebutuhan fungsional dari sebuah perangkat lunak untuk memperoleh kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional sebuah program (Pressman, 2010).

Pengujian *Black Box* digunakan untuk menemukan kesalahan dalam kategori berikut:

1. Menganalisa fungsi yang tidak benar atau fungsi yang hilang.
2. Menganalisa kesalahan antarmuka.
3. Menganalisa kesalahan dalam struktur data atau akses *database* eksternal.
4. Menganalisa kesalahan perilaku (*behavior*) atau kesalahan kinerja.
5. Melakukan inisialisasi dan pemutusan kesalahan.

2.2.13.1 Validation Testing

Validation testing dilaksanakan ketika sebuah perangkat lunak telah tersusun sempurna menjadi satu kesatuan yang utuh. Kegiatan pengujian berfokus pada aksi-aksi yang dapat dilakukan oleh user serta output yang dapat dikenali oleh user.

2.2.14 Pengujian *White-Box*

Pengujian menggunakan *white-box* merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan yang telah dibuat, menggunakan struktur kontrol dari desain program sesuai dengan prosedur untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara singkat dapat diambil kesimpulan bahwa pengujian *white-box* merupakan petunjuk untuk mendapatkan program yang benar secara 100% (Jackpamot, 2013).

Penggunaan metode pengujian *white-box* dilakukan untuk :

1. Memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali.
2. Menggunakan semua keputusan logis untuk semua kondisi *true* atau *false*.
3. Mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.
4. Menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

2.2.15 Pengujian Compatibility

Pengujian *compatibility* digunakan untuk melakukan sebuah pengujian bahwa semua fitur yang ada pada suatu produk sistem dapat bekerja di *browser* yang berbeda. Pengujian ini berfungsi untuk menentukan hasil yang diharapkan oleh sistem yang dikembangkan dapat berjalan sesuai dengan kebutuhan fungsional.

Jika sistem dapat berjalan di banyak jenis *browser* yang berbeda, maka aspek *compatibility* sistem semakin baik. Pengujian *compatibility* pada sisi *browser* dapat dilakukan dengan menggunakan sebuah *tools* atau alat bantu berupa perangkat lunak, diantaranya adalah *SortSite, Secure Platform, Virtual Desktops, Ranorex Studio,* dan lain sebagainya.

Tools yang akan digunakan pada penelitian ini adalah *Sortsite.* Perangkat lunak *SortSite* digunakan karena mampu menguji sebuah perangkat lunak berbasis *web* secara keseluruhan pada setiap halaman yang dimiliki. Selain itu, *SortSite* juga mampu menguji perangkat lunak berbasis *web* pada beberapa *checkpoints* diantaranya *Accessibility, Broken Links, Compatibility, Search Engine Optimation, Privacy, Web Standart,* dan *Usability* (Rogers, 2016).

BAB 3 METODOLOGI

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan pada Dinas Pariwisata dan Kebudayaan Kabupaten Banyuwangi yang berlokasi di Jalan A. Yani nomor 78, Kabupaten Banyuwangi dan Laboratorium Sistem Informasi Program Teknologi Informasi Dan Ilmu Komputer Universitas Brawijaya. Penelitian dilaksanakan pada Bulan Maret 2016 sampai dengan Bulan Juni 2016.

3.2 Langkah-langkah Penelitian

Penelitian ini merupakan tipe penelitian implementatif – pengembangan (development), penulis mengembangkan Sistem Informasi Manajemen Pariwisata Terpadu pada Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi. Tahapan penelitian dalam pembuatan Sistem Informasi Manajemen Pariwisata Terpadu pada Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi disajikan pada Gambar 3.1.

Gambar 3.1 Langkah-langkah Penelitian

3.2.1 Wawancara

Dalam proses ini kegiatan yang dilakukan adalah melakukan diskusi serta tanya jawab dengan sumber yang dianggap memiliki pengetahuan yang lebih dalam dari permasalahan penelitian. Tujuan dari wawancara yaitu untuk mengetahui apa saja permasalahan yang terjadi di Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi dalam melakukan promosi dan menyebarkan informasi terkait destinasi wisata dan pagelaran-pagelaran yang ada di Kabupaten Banyuwangi. Pada tahap wawancara peneliti mewawancarai Kepala Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi.

3.2.2 Study Literatur

Penelitian ini dilakukan dengan cara mengumpulkan beberapa sumber data yang diperoleh dari penelitian-penelitian yang sudah ada sebelumnya yang berkaitan dengan sistem informasi manajemen yang merupakan fokus dalam penelitian ini. Pada tahap ini dilakukan juga pengumpulan teori-teori yang berkaitan dengan penelitian sebagai landasan dalam mengembangkan sistem informasi manajemen pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi. Data-data tersebut didapatkan dengan membaca referensi, mengkaji penelitian yang sudah ada, dan membuat kesimpulan, sehingga diperoleh teori ilmiah yang melandasi penelitian.

3.2.3 Analisis Kebutuhan

Pada tahap analisis kebutuhan ini memiliki tujuan untuk mengetahui kebutuhan dalam pengembangan sistem informasi manajemen pariwisata terpadu. Hasil dari analisis kebutuhan didapatkan kebutuhan fungsional dan kebutuhan non-fungsional yang mengacu pada tujuan pembuatan sistem informasi manajemen pariwisata terpadu dan melakukan pemodelan kebutuhan sistem yang dibutuhkan.

3.2.4 Perancangan

Setelah analisis kebutuhan sistem telah didefinisikan pada tahap sebelumnya, maka tahap selanjutnya adalah membuat rancangan sistem yang akan dibangun dalam tahap ini dengan membuat perancangan sistem. Pada analisa kebutuhan didapatkan fitur-fitur yang harus ada di dalam sistem. Pada tahap ini akan dirancang arsitektur sistem dan alur jalannya sistem sesuai dengan kebutuhan fungsional sistem. Di dalam tahap perancangan juga akan dirancang bagaimana struktur penyimpanan data di dalam *database*.

3.2.5 Implementasi

Pada tahap implementasi dilakukan pengkodean menggunakan bahasa pemrograman *PHP* dan *database MySQL* untuk membuat sistem berbasis *website*. Implementasi sistem menggunakan pemrograman berorientasi objek yang mengacu pada hasil dari tahap analisis kebutuhan dan tahap perancangan serta dikembangkan dengan memanfaatkan *framework CodeIgniter.* Hasil dari tahapan ini merupakan sebuah sistem informasi manajemen pariwisata terpadu sebagai media promosi Kabupaten Banyuwangi yang berbasis *website* termasuk *user interface* dan pembuatan tabel-tabel di dalam *database*.

3.2.6 Pengujian dan Analisis

Sebelum sistem diterapkan, sistem harus melewati tahap pengujian. Hal ini dilakukan untuk mengidentifikasi bila terjadi kesalahan pada sistem. Dan dari hasil pengujian dapat dilakukan analisis dari sistem yang telah diimplementasikan. Pada pengujian ini dilakukan oleh peneliti sendiri dikarenakan agar peneliti juga dapat mengetahui bahwa pengujian berjalan sesuai dengan skenario yang sudah ditentukan.

Pada penelitian ini, pengujian menggunakan 3 (tiga) tahap pengujian, yaitu pengujian *black-box,* pengujian *white-box* dan pengujian *compatibility*. Pada tahap pengujian *black-box* digunakan untuk menguji apakah sistem berjalan sesuai dengan spesifikasi kebutuhan yang telah dirancang sebelumnya. Tahap pengujian white-box digunakan untuk melakukan pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Sedangkan pada pengujian *compatibility* digunakan untuk mengetahui kinerja sistem pada *browser* yang berbeda.

3.2.7 Kesimpulan dan Saran

Kesimpulan dan saran merupakan tahapan terakhir dalam penelitian ini. Penarikan dan pembuatan kesimpulan dilakukan setelah hasil pengujian sistem yang telah diimplementasikan didapatkan. Tujuan dari penarikan dan pembuatan kesimpulan yaitu mengetahui tentang kelebihan dan kekurangan sistem yang telah penulis kembangkan. Dengan kesimpulan tersebut penulis dapat memberikan saran apabila akan dilakukan penelitian lebih lanjut.

BAB 4 ANALISIS KEBUTUHAN

4.1 Deskripsi Umum Sistem

Sistem Informasi Manajemen Pariwisata Terpadu Kabupaten Banyuwangi merupakan sebuah website yang dibangun untuk menyelesaikan permasalahan yang terdapat pada pariwisata di kabupaten Banyuwangi, yaitu permasalahan wisatawan yang kurang mengetahui info tempat wisata yang spesifik dan masih belum ada sistem yang melakukan pemesanan secara *online* untuk mengunjungi tempat-tempat wisata yang ada di kabupaten Banyuwangi. Sistem ini dibangun untuk memudahkan wisatawan di kabupaten Banyuwangi mauapun wisatawan di luar kabupaten Banyuwangi dapat mengetahui informasi tempat wisata yang spesifik dan dapat melakukan pemesanan online untuk mengunjungi tempat-tempat wisata yang ada di kabupaten Banyuwangi. Dan juga sistem ini juga menjembatani para pengelola wisata untuk mendaftarkan tempat wisata yang dikelola agar dapat mempromosikan tempat wisata dan juga melakukan interaksi dengan wisatawan.

Terdapat beberapa fitur-fitur yang dapat diakses, seperti melihat *event*, melihat daftar objek wisata, dan melihat detail objek wisata yang harus diimplementasikan dalam sistem agar dapat membantu wisatawan dalam mengetahui potensi wisata yang ada di kabupaten Banyuwangi.

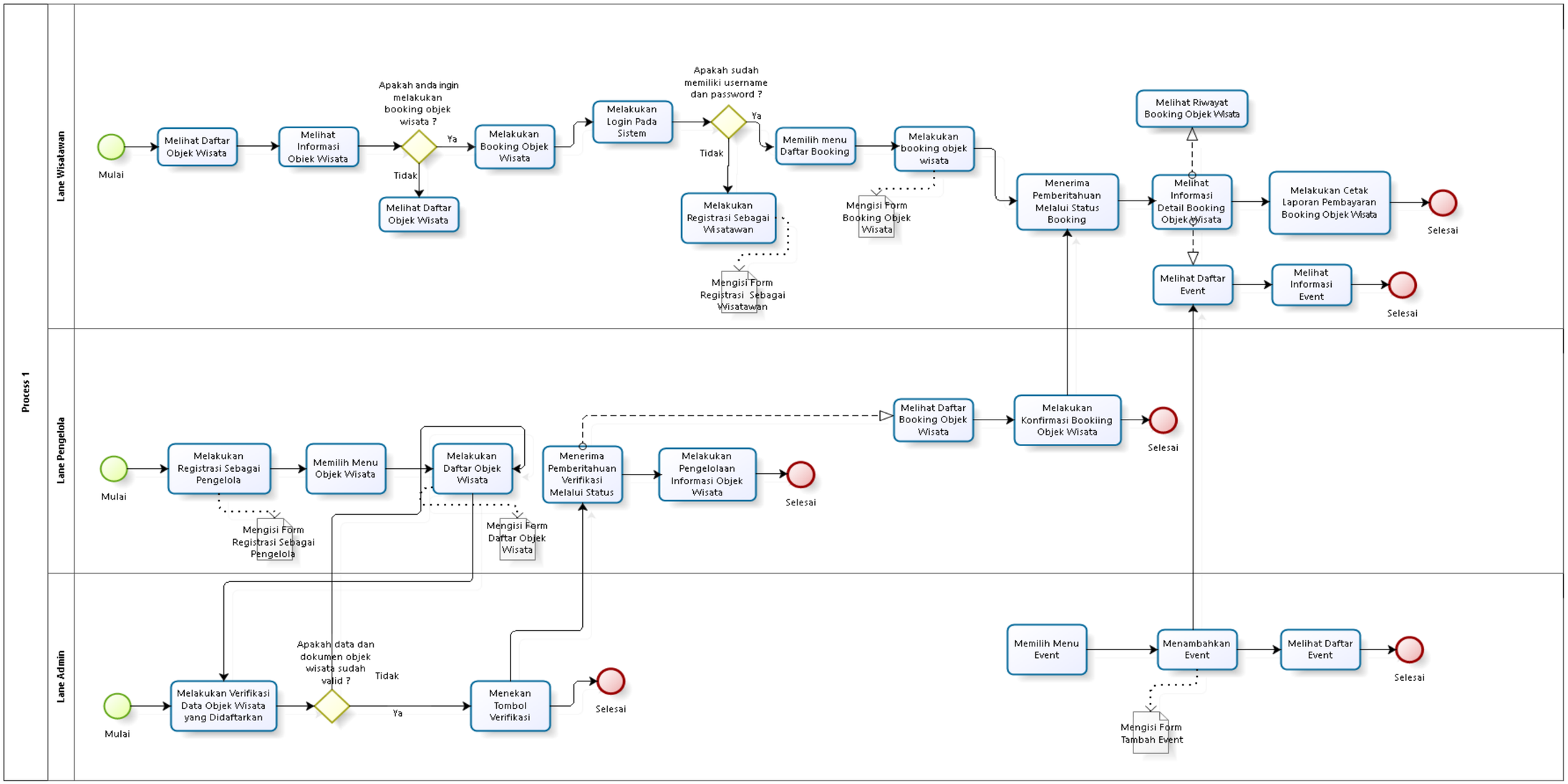
Sistem Informasi Manajemen Pariwisata Terpadu Kabupaten Banyuwangi dibangun dengan memiliki kemampuan untuk:

1. Menghindari antrian panjang wisatawan dalam melakukan pembelian tiket terutama pada saat hari libur sekolah.
2. Menyediakan informasi-informasi seputar *event* dan objek wisata yang ada di Kabupaten Banyuwangi.
3. Mengubah pelayanan dari sistem manual menjadi otomatis, sehingga proses pelayanan menjadi lebih cepat dan mudah.
4. Menjembatani antara pihak Dinas Kebudayaan dan Pariwisata dengan para pengelola wisata yang ada di Kabupaten Banyuwangi dalam mempromosikan tempat wisata yang ada di Kabupaten Banyuwangi.
5. Menyediakan fasilitas *booking* objek wisata yang dapat diakses oleh wisatawan.

4.2 Proses Bisnis

Dari hasil wawancara dengan pihak Dinas Kebudayaan dan Priwisata kabupaten Banyuwangi tentang proses bisnis yang sudah dijalankan dalam melakukan promosi pariwisata masih belum diterapkan untuk saat ini, sehingga penulis membuat analisis proses bisnis yang akan diterapkan dan sesuai dengan kebutuhan yang dibutuhkan oleh wisatawan. Berikut ini merupakan proses bisnis yang penulis usulkan kepada pihak Dinas Kebudayaan dan Pariwisata kabupaten Banyuwangi.

repository.ub.ac.id



Gambar 4.1 Proses Bisnis Usulan

Pada gambar 4.1 merupakan alur proses bisnis usulan yang terjadi pada sistem informasi manajemen pariwisata terpadu. Berikut penjelasan yang akan penulis jabarkan:

1. Pengelola melakukan registrasi sebagai pengelola objek wisata.
2. Pengelola wisata mendaftarkan objek wisata yang dikelola pada sistem sesuai dengan data dan berkas yang valid.
3. Setelah pengelola wisata mendaftarkan objek wisata, maka *admin* akan melakukan pengecekan terhadap data dan berkas objek wisata yang sudah didaftarkan.
4. Jika data dan berkas objek wisata sudah valid maka admin akan melakukan konfirmasi dan pengelola wisata akan menerima notifikasi *verified*, tetapi jika data dan berkas objek wisata yang didaftarkan belum memenuhi kriteria yang sudah ditetapkan maka pengelola harus kembali mendaftarkan objek wisata yang dikelolanya sesuai dengan kriteria yang sudah ditetapkan pada *form* pendaftaran objek wisata.
5. Setelah pengelola menerima notifikasi *verified*, maka pengelola wisata mengelola detail objek wisata yang dikelola seperti menambahkan informasi objek wisata dan mengupdate informasi objek wisata.
6. Wisatawan yang sudah terdaftar pada sistem dapat melihat daftar objek wisata yang telah terdaftar, kemudian wisatawan dapat melakukan booking objek wisata dengan mengisi *form booking* objek wisata yang tersedia.
7. Kemudian pengelola wisata akan melihat daftar *booking* objek wisata yang masuk dan pengelola akan melakukan konfirmasi *booking* wisata.
8. Sehingga wisatawan akan menerima notifikasi berhasil dan dapat mencetak laporan *booking* untuk dibawa ke objek wisata yang sudah dibooking, wisatawan juga dapat melihat daftar riwayat *booking* wisata yang pernah dilakukan.
9. Admin akan melakukan menambahkan, mengupdate dan menghapus *event* yang ada.
10. Tetapi jika wisatawan tidak melakukan *booking* wisata maka wisatawan akan kembali ke menu melihat daftar tempat wisata dan dapat juga melihat daftar *event* serta dapat melihat informasi *event* .

4.3 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan yang harus dilakukan sebelum melakukan pemodelan kebutuhan sistem. Tahap analisis kebutuhan pada penelitian ini diawali dengan elisitasi dan analisis kebutuhan, identifikasi aktor dalam sistem dan pendefinisian daftar kebutuhan fungsional dan non-fungsional sistem.

4.3.1 Elisitasi dan Analisis Kebutuhan

Pada tahap ini digunakan untuk mengetahui masalah yang perlu dipecahkan dan mengenali aktor-aktor dari sistem. Langkah-langkah yang dilakukan yaitu, mulai dari wawancara secara langsung dengan wisatawan dan staff IT Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi, kemudian dilanjutkan

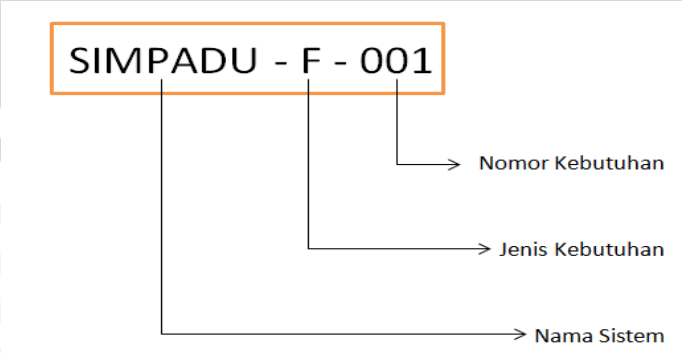
dengan melakukan pengamatan terhadap sistem yang sudah ada dan terhadap aktivitas promosi tempat wisata yang berjalan saat ini.

Berdasarkan hasil wawancara dengan wisatawan dan staff IT Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi yang telah dilakukan, didapatkan beberapa hasil yang dapat digunakan untuk perencanaan dalam membangun sistem :

1. Sistem dapat menampilkan daftar objek wisata
   Sistem dapat menampilkan daftar objek wisata yang ada di Kabupaten Banyuwangi dan informasi detail objek wisata.
2. Sistem menyediakan fasilitas untuk melakukan *booking* objek wisata
   Sistem menyediakan fasilitas *booking* dengan tujuan wisatawan tidak perlu melakukan antrian untuk mendapatkan tiket objek wisata.
3. Sistem dapat menampilkan daftar *event*
   Sistem dapat menampilkan daftar *event* dengan tujuan agar wisatawan dapat mengetahui informasi *event* yang ada di Kabupaten Banyuwangi.
4. Sistem menyediakan fasilitas untuk mengelola objek wisata
   Sistem menyediakan fasilitas untuk mengelola objek wisata meliputi menambahkan informasi objek wisata.
5. Sistem menyediakan fasilitas untuk mengelola *event*
   Sistem menyediakan fasilitas untuk mengelola *event* meliputi menambahkan *event* dan menghapus *event*.
6. Sistem dapat menampilkan jumlah wisatawan
   Sistem dapat menampilkan jumlah wisatawan dengan tujuan dapat dijadikan patokan perbedaan jumlah wisatawan per tahun mengalami kenaikan atau penurunan dalam bentuk grafik.
7. Sistem dapat menampilkan daftar wisatawan.
   Sistem dapat menampilkan daftar wisatawan dengan tujuan dapat mengetahui informasi tentan wisatawan.

4.3.2 Aturan Penomoran

Aturan penomoran kebutuhan harus ditulis sesuai dengan format yang sudah ditetapkan untuk menjaga keteraturan penulisan. Berikut contoh format penulisan penomoran yang digunakan pada penelitian ini adalah :



Gambar 4.2 Format Penomoran Kebutuhan

Format penomoran kebutuhan yang digunakan dalam penelitian ini tersusun atas 3 bagian seperti yang tertera pada Gambar 4.2.1. Nama sistem yang digunakan dalam format penomoran adalah SIMPADU yang merupakan akronim dari Sistem Informasi Manajemen Pariwisata Terpadu. Jenis kebutuhan dapat ditulis dengan huruf F untuk mewakili kebutuhan fungsional atau huruf NF untuk mewakili kebutuhan non fungsional. Nomor kebutuhan ditulis dengan nomor urut dan dimulai dari angka 1.

4.3.3 Identifikasi Pengguna

Identifikasi pengguna adalah tahap menganalisis aktor yang berinteraksi dengan sistem, pada tahap ini peneliti mengacu dari tahapan elisitas dan analisis kebutuhan secara umum yang telah dilakukan sebelumnya. Pada tabel 4.1 berikut menjabarkan aktor yang berinteraksi dengan sistem.

Tabel 4.1 Identifikasi Pengguna

No.	Kategori Pengguna	Deskripsi
1.	Admin	Admin adalah pengguna yang memiliki hak akses paling tinggi dibandingkan pengguna lain.
2.	Pengelola Wisata	Pengelola Wisata adalah pengguna yang mengelola wisata yang sudah terdaftar didalam sistem dan memiliki hak akses untuk mengelola tempat wisata pada sistem.
3.	Wisatawan	Wisatawan adalah pengguna yang terdaftar didalam sistem dan memiliki hak akses pada sistem.
4.	Pengguna Umum	Pengguna umum adalah pengguna yang belum terdaftar didalam sistem dan memiliki hak akses terbatas.

4.3.4 Kebutuhan Fungsional

Daftar kebutuhan fungsional ditulis dalam bentuk tabel dan dikelompokkan berdasarkan aktornya sesuai dengan nama fungsi dan juga deskripsinya. Berikut merupakan kebutuhan fungsional tiap aktor dari sistem informasi yang akan dikembangkan dalam penelitian ini antara lain:

1. **Pengguna Umum**

Tabel 4.2 Kebutuhan Fungsional Pengguna Umum

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIMPADU-F-001	Registrasi wisatawan	Pengguna umum harus melakukan daftar sebagai wisatawan agar dapat melakukan *login* pada sistem dengan mengisi *form* registrasi wisatawan. Pada *form* registrasi wisatawan terdapat atribut nama, alamat, nomor hp, jenis kelamin, *email*, *username*, dan *password*.

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
2.	SIMPADU-F-002	*Login*	Pengguna umum harus melakukan *login* agar dapat mengakses fitur-fitur yang disediakan oleh sistem.
3.	SIMPADU-F-003	Registrasi pengelola	Pengguna umum dapat melakukan daftar sebagai pengelola agar dapat mendaftarkan objek wisatanya pada sistem dengan mengisi *form* registrasi pengelola. Pada *form* registrasi pengelola terdapat atribut nama pengelola, nomor hp, *email*, *username*, dan *password*.

2. Wisatawan

Tabel 4.3 Kebutuhan Fungsional Wisatawan

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIMPADU-F-004	Melihat Detail Info objek wisata	Wisatawan dapat melihat detail info objek wisata yang ingin diketahuinya pada sistem.
2.	SIMPADU-F-005	*Logout*	Wisatawan dapat melakukan *logout* jika berkeinginan keluar dari sistem.
3.	SIMPADU-F-006	Melakukan *booking* objek wisata	Wisatawan dapat melakukan *booking* objek wisata dengan cara mengisi *form booking* objek wisata dengan ketentuan yang terdapat pada sistem. Pada *form booking* terdapat atribut tanggal *booking*, jumlah tiket, dan nama objek wisata.
4.	SIMPADU-F-007	Melihat riwayat *booking* objek wisata	Wisatawan dapat melihat riwayat *booking* objek wisata yang pernah dilakukan pada sistem.
5.	SIMPADU-F-008	Cetak laporan *booking* objek wisata	Wisatawan dapat melakukan cetak laporan *booking* objek wisata setelah melakukan booking dan dikonfirmasi oleh pengelola objek wisata pada sistem.
6.	SIMPADU-F-009	Melihat daftar objek wisata	Wisatawan dapat melihat daftar objek wisata yang telah terdaftar pada sistem.
7.	SIMPADU-F-010	Melihat daftar *event*	Wisatawan dapat melihat daftar *event* yang akan berlangsung pada sistem.

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
8.	SIMPADU-F-011	Melihat detail info *event*	Wisatawan dapat melihat detail info *event* yang akan berlangsung pada sistem.
9.	SIMPADU-F-012	Melihat Detail *booking* objek wisata	Wisatawan dapat melihat detail informasi *booking* objek wisata yang telah dilakukannya pada sistem.

3. Pengelola Wisata

Tabel 4.4 Kebutuhan Fungsional Pengelola Wisata

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIMPADU-F-005	*Logout*	Pengelola dapat melakukan *logout* jika berkeinginan keluar dari sistem.
2.	SIMPADU-F-009	Melihat daftar objek wisata	Pengelola dapat melihat daftar objek wisata yang dikelolanya pada sistem.
3.	SIMPADU-F-013	Melihat daftar *booking* objek wisata	Pengelola dapat melihat daftar booking tempat wisata yang masuk pada sistem.
4.	SIMPADU-F-014	Konfirmasi booking objek wisata	Pengelola dapat melakukan konfirmasi *booking* objek wisata melalui sistem.
5.	SIMPADU-F-015	Menambah objek wisata	Pengelola dapat menambahkan objek wisata yang sesuai dengan daerah objek wisata yang dikelola pada sistem dengan mengisi *form* tambah objek wisata. Pada *form* tambah objek wisata terdapat atribut nama objek, inisial, alamat, harga tiket, pilihan wisata, gambar, dokumen, dan koordinat.
6.	SIMPADU-F-004	Melihat detail info objek wisata	Pengelola dapat melihat detail info objek wisata yang dikelolanya pada sistem.
7.	SIMPADU-F-016	Melihat grafik jumlah wisatawan	Pengelola dapat melihat jumlah wisatawan yang mengujungi objek wisata yang dikelola dalam bentuk grafik pada sistem.
8.	SIMPADU-F-017	Mengupdate objek wisata	Pengelola dapat mengupdate informasi objek wisata yang dikelola pada sistem dengan mengisi *form* update objek wisata. Pada form update objek wisata terdapat atribut deskripsi, harga tiket, dan dokumen.

4. Admin

Tabel 4.5 Kebutuhan Fungsional Admin

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIMPADU-F-005	*Logout*	Admin dapat melakukan *logout* jika berkeinginan keluar dari sistem.
2.	SIMPADU-F-009	Melihat daftar objek wisata	Admin dapat melihat daftar objek wisata yang telah terdaftar pada sistem.
3.	SIMPADU-F-018	Melihat daftar wisatawan	Admin dapat melihat daftar wisatawan yang terdaftar pada sistem.
4.	SIMPADU-F-019	Verifikasi objek wisata	Admin dapat melakukan verifikasi objek wisata yang didaftarkan oleh pengelola pada sistem jika sudah sesuai dengan prosedur.
5.	SIMPADU-F-020	Menambah *event*	Admin dapat menambahkan *event* terbaru yang akan berlangsung pada sistem dengan mengisi *form* tambah *event*. Pada form tambah event terdapat atribut nama event, lokasi, tanggal, tanggal2, htm, gambar, dan deskripsi.
6.	SIMPADU-F-021	Menghapus *event*	Admin dapat menghapus *event* jika *event* telah terlaksana atau berakhir pada sistem.
7.	SIMPADU-F-022	Melihat grafik objek wisata	Admin dapat melihat jumlah wisatawan tiap objek wisata yang terdaftar per bulan dan juga objek wisata yang paling diminati dalam bentuk grafik pada sistem.
8.	SIMPADU-F-010	Melihat daftar *event*	Admin dapat melihat daftar *event* yang akan berlangsung pada sistem.
9.	SIMPADU-F-011	Melihat detail info *event*	Admin dapat melihat detail info *event* yang akan berlangsung pada sistem.
10.	SIMPADU-F-004	Melihat detail info objek wisata	Admin dapat melihat detail info objek wisata yang sudah terdaftar maupun belum terdaftar sebagai proses dari verifikasi objek wisata sebelum diverifikasi pada sistem.
11.	SIMPADU-F-023	Melihat daftar *administrator*	Admin dapat melihat daftar *administrator* yang aktif pada sistem.
12.	SIMPADU-F-024	Menghapus *administrator*	Admin dapat menghapus administrator lainnya jika diperlukan pada sistem.

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
13.	SIMPADU-F-025	Menambah *administrator*	Admin dapat menambahkan *administrator* jika itu diperlukan pada sistem dengan mengisi *form* tambah *administrator*. Pada *form* tambah *administrator* terdapat atribut nama *administrator*, *username*, dan *password*.
14.	SIMPADU-F-026	Mengupdate event	Admin dapat mengupdate info *event* yang akan berlangsung pada sistem dengan mengisi *form update event*. Pada form update event terdapat atribut lokasi, tanggal, tanggal2, deskripsi, dan htm.

4.3.5 Kebutuhan Non Fungsional

Tabel 4.6 Kebutuhan Non Fungsional Sistem

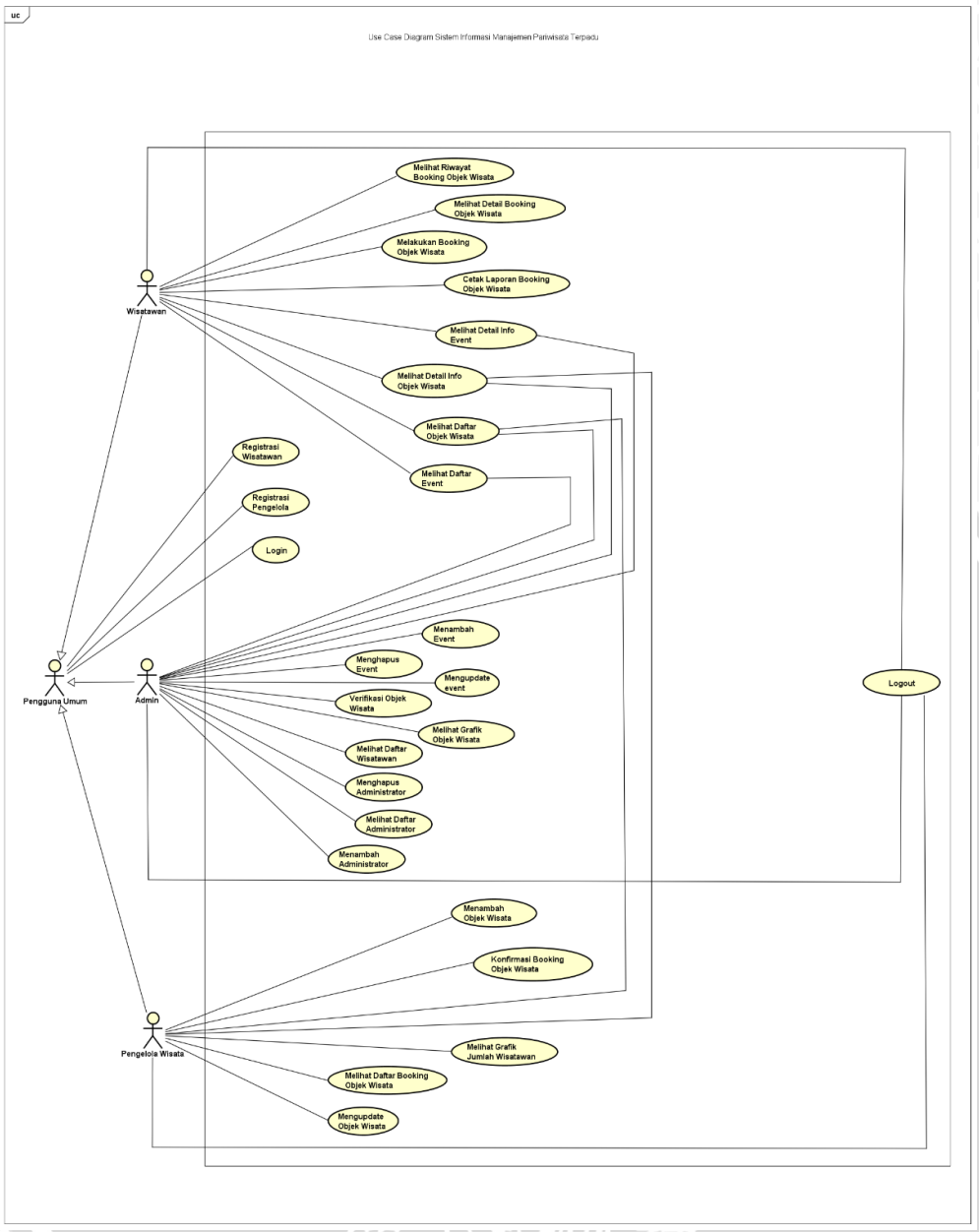
No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIMPADU-NF-001	*Security*	Sistem akan menggunakan enkripsi md5 untuk keamanan password bagi pengguna sistem dalam melakukan sign in pada sistem.
2.	SIMPADU-NF-002	*Usability*	Sistem dapat memberikan kemudahan dan kenyaman dalam penggunaan sistem bagi pengguna dalam mengaksesnya.
3.	SIMPADU-NF-003	*Compatibility*	Sistem dapat dijalankan dengan baik pada *browser* yang berbeda dan bersifat *responsif*.

4.4 Pemodelan Kebutuhan Sistem

Pada tahap pemodelan kebutuhan sistem dilakukan dengan mengacu terhadap tahap analisis kebutuhan yang telah dilakukan sebelumnya, tahap pemodelan kebutuhan sistem merupakan rancangan atau perencanaan dari sistem yang akan dibangun agar proses pembangunan sistem dapat berjalan dengan baik dan sesuai dengan apa yang di harapkan. Pada pemodelan kebutuhan sistem terdiri *use case diagram dan activity diagram.*

4.4.1 *Use Case Diagram*

Use case diagram pada gambar 4.3 menggambarkan 3 aktor yang terdiri dari wisatawan, admin, dan pengelola merupakan generalisasi dari pengguna umum.



Gambar 4.3 Use Case Diagram Simpadu

4.4.1.1 *Use Case Scenario*

Use case scenario dari setiap kebutuhan fungsional yang telah digambarkan dalam *use case diagram* pada gambar 4.1 akan dideskripsikan dalam bentuk tabel. Tabel pada *use case scenario* akan berisi beberapa keterangan seperti nama *use case*, deskripsi singkat *use case*, aktor yang berperan, *pre condition, post condition, basic flow of event,* dan *alternative flow.*

1. *Use Case Scenario* Registrasi Wisatawan (SIMPADU-F-001)

Berdasarkan Tabel 4.7, *use case* registrasi wisatawan bertujuan untuk melakukan registrasi sebagai wisatawan. Aktor dari *use case* ini yaitu pengguna umum.

Tabel 4.7 Registrasi Wisatawan (SIMPADU-F-001)

Brief Description	*Use case* ini memungkinkan aktor untuk melakukan registrasi wisatawan pada sistem.
Actor	Pengguna Umum
Pre-condition	1. Aktor belum memiliki *username* dan *password*. 2. Sistem menampilkan halaman awal (halaman *login*).
Basic Flow of Event	1. Aktor memilih fungsi registrasi wisatawan. 2. Sistem menampilkan form registrasi wisatawan. 3. Aktor mengisi data pada form registrasi wisatawan. 4. Aktor menekan tombol "Registrasi". 5. Sistem menyimpan data wisatawan didalam *database* dan menampilkan pesan "Registrasi Berhasil". 6. Aktor berhasil melakukan registrasi wisatawan dan dapat melakukan *login.*
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	1. Data pada basisdata mengalami *update* otomatis. 2. Sistem melakukan *redirect* (beralih) ke halaman *login*.

2. *Use Case Scenario* Registrasi Pengelola (SIMPADU-F-003)

Berdasarkan Tabel 4.8, *use case* registrasi pengelola bertujuan untuk melakukan registrasi sebagi pengelola. Aktor dari *use case* ini yaitu pengelola.

Tabel 4.8 Registrasi Pengelola (SIMPADU-F-003)

Brief Description	*Use case* ini memungkinkan aktor untuk melakukan registrasi pengelola objek wisata
Actor	Pengelola
Pre-condition	Sistem menampilkan halaman utama (halaman *login*).
Basic Flow Of Event	1. Aktor memilih menu Registrasi Pengelola pada sistem. 2. Sistem menampilkan *form* registrasi pengelola 3. Aktor mengisi *form* registrasi pengelola. 4. Aktor menekan tombol "registrasi". **{A1}** 5. Sistem menampilkan pesan "register berhasil".
Alternative Flow	**A1: kolom wajib belum diisi**
	1. Sistem menampilkan pesan peringatan "kolom wajib diisi". 2. Aktor melihat pesan peringatan.

	3. Sistem menampilkan halaman *form* registrasi tempat wisata.
Subflows	Tidak ada
Post-condition	1. Data pada basisdata mengalami *update* otomatis. 2. Sistem melakukan *redirect* (beralih) ke halaman *login*.

3. *Use Case Scenario* Melihat Daftar Objek Wisata (SIMPADU-F-009)

Berdasarkan Tabel 4.9, *use case* melihat daftar objek wisata memungkinkan aktor untuk melihat daftar objek wisata pada sistem. Aktor dari *use case* ini yaitu wisatawan, *admin*, dan pengelola.

Tabel 4.9 Melihat Daftar Objek Wisata (SIMPADU-F-009)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat daftar objek wisata
Actor	Wisatawan, Pengelola, *Admin*
Pre-condition	Aktor sudah masuk di *dashboard* masing-masing-masing aktor.
Basic Flow Of Event	1. Aktor memilih menu Objek Wisata pada sistem. 2. Sistem menampilkan daftar objek wisata yang sudah terdaftar pada sistem. 3. Aktor melihat daftar objek wisata.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat daftar objek wisata sesuai dengan halaman daftar objek wisata masing-masing aktor.

4. *Use Case Scenario* Melihat *Detail* Info Objek Wisata (SIMPADU-F-004)

Berdasarkan Tabel 4.10, *use case* melihat *detail* info objek wisata memungkinkan aktor untuk melihat *detail* informasi objek wisata pada sistem. Aktor dari *use case* ini yaitu wisatawan, *admin*, dan pengelola.

Tabel 4.10 Melihat Detail Info Objek Wisata (SIMPADU-F-004)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat *detail* informasi objek wisata
Actor	Wisatawan, Pengelola, *Admin*

Pre-condition	Aktor sudah masuk di dashboard masing-masing aktor.
Basic Flow Of Event	1. Aktor memilih menu Objek Wisata pada sistem. 2. Sistem menampilkan daftar objek wisata yang sudah terdaftar pada sistem. 3. Aktor menekan tombol "*view detail*" dari salah satu dari daftar objek wisata. 4. Sistem menampilkan detail informasi objek wisata. 5. Aktor melihat detail informasi objek wisata.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat *detail* info objek wisata sesuai dengan halaman daftar objek wisata masing-masing aktor.

5. *Use Case Scenario* Melakukan *Booking* Objek Wisata (SIMPADU-F-006)

Berdasarkan Tabel 4.11, *use case* melakukan *booking* objek wisata memungkinkan aktor untuk melakukan booking objek wisata pada sistem. Aktor dari *use case* ini yaitu wisatawan.

Tabel 4.11 Melakukan *Booking* Objek Wisata (SIMPADU-F-006)

Brief Description	*Use case* ini memungkinkan aktor untuk melakukan *booking* objek wisata
Actor	Wisatawan
Pre-condition	1. Aktor sudah masuk di *dashboard* Wisatawan.
Basic Flow Of Event	1. Aktor memilih menu *Booking* pada sistem. 2. Sistem menampilkan riwayat *booking* objek wisata yang pernah dilakukan. 3. Aktor menekan tombol "*Booking*". 4. Sistem menampilkan *form booking*. 5. Aktor mengisi *form booking*. 6. Aktor menekan tombol "*Booking*". 7. Sistem menampilkan pesan "Selamat *booking* Anda berhasil. Silakan tunggu konfirmasi." 8. Aktor berhasil melakukan *booking*.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melakukan *booking* objek wisata

6. *Use Case Scenario* Melihat Riwayat Booking Objek Wisata (SIMPADU-F-007)

Berdasarkan Tabel 4.12, *use case* melihat riwayat booking objek wisata memungkinkan aktor untuk melihat riwayat booking objek wisata yang telah dilakukannya pada sistem. Aktor dari *use case* ini yaitu wisatawan.

Tabel 4.12 Melihat Riwayat *Booking* Objek Wisata (SIMPADU-F-007)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat riwayat *booking* objek wisata.
Actor	Wisatawan
Pre-condition	Aktor sudah masuk di *dashboard* Wisatawan.
Basic Flow Of Event	1. Aktor memilih menu *Booking* pada sistem. 2. Sistem menampilkan riwayat *booking* objek wisata. 3. Aktor melihat riwayat *booking*.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat riwayat *booking* yang pernah dilakukan.

7. *Use Case Scenario* Melihat Daftar Booking Objek Wisata (SIMPADU-F-013)

Berdasarkan Tabel 4.13, *use case* melihat daftar *booking* objek wisata memungkinkan aktor untuk melihat daftar *booking* objek wisata yang belum dikonfirmasi ataupun yang sudah dikonfirmasi pada sistem. Aktor dari *use case* ini yaitu pengelola.

Tabel 4.13 Melihat Daftar *Booking* Objek Wisata (SIMPADU-F-013)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat daftar *booking* objek wisata yang dikelola
Actor	Pengelola
Pre-condition	Aktor sudah masuk di *dashboard* Pengelola.
Basic Flow Of Event	1. Aktor memilih menu *Booking* pada sistem. 2. Sistem menampilkan daftar *booking* objek wisata yang sudah masuk pada sistem. 3. Aktor melihat daftar *booking* objek wisata.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat daftar *booking* objek wisata.

8. *Use Case Scenario* Konfirmasi Booking Objek Wisata (SIMPADU-F-014)

Berdasarkan Tabel 4.14, *use case* konfirmasi *booking* objek wisata memungkinkan aktor untuk melakukan konfirmasi booking objek wisata yang sudah masuk pada sistem. Aktor dari *use case* ini yaitu pengelola.

Tabel 4.14 Konfirmasi *Booking* Objek Wisata (SIMPADU-F-014)

Brief Description	*Use case* ini memungkinkan aktor untuk melakukan konfirmasi *booking* objek wisata
Actor	Pengelola
Pre-condition	Aktor sudah masuk di *dashboard* Pengelola.
Basic Flow Of Event	1. Aktor memilih menu *Booking* pada sistem. 2. Sistem menampilkan daftar *booking* objek wisata yang sudah masuk pada sistem. 3. Aktor menekan tombol konfirmasi pada daftar *booking* objek wisata. 4. Sistem meperbarui halaman dan merubah status menjadi *approved*.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melakukan konfirmasi *booking* objek wisata yang sudah masuk pada sistem.

9. *Use Case Scenario* Menambah Objek Wisata (SIMPADU-F-015)

Berdasarkan Tabel 4.15, *use case* menambah objek wisata memungkinkan aktor untuk menambahkan objek wisata yang dikelolanya pada sistem. Aktor dari *use case* ini yaitu pengelola.

Tabel 4.15 Menambah Objek Wisata (SIMPADU-F-015)

Brief Description	*Use case* ini memungkinkan aktor untuk objek wisata yang masih dikelolanya
Actor	Pengelola
Pre-condition	Aktor sudah masuk di *dashboard* Pengelola.
Basic Flow Of Event	1. Aktor memilih menu Objek Wisata pada sistem. 2. Sistem menampilkan halaman daftar objek wisata yang dikelola. 3. Aktor menekan tombol “Tambah”. 4. Sistem menampilkan *form* tambah objek wisata. 5. Aktor mengisi *form* tambah objek wisata sesuai dengan *form* yang ada. 6. Aktor menekan tombol “Tambah”.

	7. Sistem menampilkan pesan "Objek Wisata berhasil ditambahkan. Silakan tunggu konfirmasi dari admin". 8. Aktor melihat pesan.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil menambahkan objek wisata yang dikelola.

10. *Use Case Scenario* Melihat Grafik Jumlah Wisatawan (SIMPADU-F-016)

Berdasarkan Tabel 4.16, *use case* melihat grafik jumlah wisatawan memungkinkan aktor untuk melihat grafik jumlah wisatawan setiap bulan pada sistem. Aktor dari *use case* ini yaitu pengelola.

Tabel 4.16 Melihat Grafik Jumlah Wisatawan (SIMPADU-F-016)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat jumlah wisatawan dalam bentuk grafik
Actor	Pengelola
Pre-condition	Aktor sudah masuk di *dashboard* Pengelola.
Basic Flow Of Event	1. Aktor memilih menu Grafik Wisatawan pada sistem. 2. Sistem menampilkan halaman Grafik Wisatawan dalam bentuk grafik. 3. Aktor melihat jumlah wisatawan dalam bentuk grafik.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat jumlah wisatawan dalam bentuk grafik.

11. *Use Case Scenario* Melihat Grafik Objek Wisata (SIMPADU-F-022)

Berdasarkan Tabel 4.17, *use case* melihat grafik objek wisata memungkinkan aktor untuk melihat jumlah wisatawan tiap objek wisata setiap bulannya dan objek wisata yang paling diminati pada sistem. Aktor dari *use case* ini yaitu *admin*.

Tabel 4.17 Melihat Grafik Objek Wisata (SIMPADU-F-022)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat jumlah wisatawan tiap objek wisata per bulan dan objek wisata yang paling diminati dalam bentuk grafik
Actor	*Admin*
Pre-condition	Aktor sudah masuk di *dashboard* Admin.

Basic Flow Of Event	1. Aktor memilih menu Grafik Objek Wisata pada sistem. 2. Sistem menampilkan halaman Grafik Objek Wisata dalam bentuk grafik. 3. Aktor melihat jumlah wisatawan per objek wisata per bulan dan objek wisata yang paling diminati dalam bentuk grafik.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat melihat jumlah wisatawan tiap objek wisata per bulan dan objek wisata yang paling diminati dalam bentuk grafik.

12. *Use Case Scenario* Melihat Daftar Wisatawan (SIMPADU-F-018)

Berdasarkan Tabel 4.18, *use case* melihat daftar wisatawan memungkinkan aktor untuk melihat daftar wisatawan yang terdaftar pada sistem. Aktor dari *use case* ini yaitu *admin*.

Tabel 4.18 Melihat Daftar Wisatawan (SIMPADU-F-018)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat daftar wisatawan
Actor	*Admin*
Pre-condition	Aktor sudah masuk di *dashboard* Admin.
Basic Flow Of Event	1. Aktor memilih menu Wisatawan pada sistem. 2. Sistem menampilkan daftar wisatawan. 3. Aktor melihat daftar wisatawan dalam bentuk tabel.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat daftar wisatawan dalam bentuk tabel.

13. *Use Case Scenario* Verifikasi Objek Wisata (SIMPADU-F-019)

Berdasarkan Tabel 4.19, *use case* verifikasi objek wisata memungkinkan aktor untuk melakukan verifikasi objek wisata yang sudah didaftarkan pada sistem. Aktor dari *use case* ini yaitu *admin*.

Tabel 4.19 Verifikasi Objek Wisata (SIMPADU-F-019)

Brief Description	*Use case* ini memungkinkan aktor untuk melakukan verifikasi objek wisata
Actor	*Admin*

Pre-condition	Aktor sudah masuk di *dashboard* Admin.
Basic Flow Of Event	1. Aktor memilih menu Objek Wisata pada sistem. 2. Sistem menampilkan daftar objek wisata yang ditambahkan. 3. Aktor menekan tombol "*view detail*". 4. Sistem menampilkan detail informasi objek wisata. 5. Aktor menekan tombol "kembali" pada pesan. 6. Sistem kembali ke halaman daftar objek wisata. 7. Aktor menekan tombol "verifikasi" jika objek wisata telah memenuhi persyaratan. 8. Sistem merperbarui halaman dan status objek wisata akan menjadi verified. 9. Aktor meihat daftar objek wisata.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melakukan verifikasi objek wisata.

14. *Use Case Scenario* Menambah *Event* (SIMPADU-F-020)

Berdasarkan Tabel 4.20, *use case* menambah *event* memungkinkan aktor untuk melakukan tambah *event* pada sistem. Aktor dari *use case* ini yaitu *admin*.

Tabel 4.20 Menambah *Event* (SIMPADU-F-020)

Brief Description	*Use case* ini memungkinkan aktor untuk menambah info event
Actor	*Admin*
Pre-condition	Aktor sudah masuk di *dashboard Admin*.
Basic Flow Of Event	1. Aktor memilih menu *Event* pada sistem. 2. Sistem menampilkan halaman daftar *Event*. 3. Aktor menekan tombol "tambah" . 4. Sistem menampilkan *form* tambah *event*. 5. Aktor mengisi informasi *event*. 6. Aktor menekan tombol "Tambah". 7. Sistem menampilkan pesan "*Event* berhasil ditambahkan". 8. Aktor melihat pesan.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil menambahkan *event*.

15. *Use Case Scenario* Melihat Daftar *Event* (SIMPADU-F-010)

Berdasarkan Tabel 4.21, *use case* melihat daftar *event* memungkinkan aktor untuk melihat daftar *event* yang terdapat pada sistem. Aktor dari *use case* ini yaitu *admin* dan wisatawan.

Tabel 4.21 Melihat Daftar *Event* (SIMPADU-F-010)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat daftar *event*
Actor	*Admin* dan Wisatawan
Pre-condition	Aktor sudah masuk di *dashboard Admin*.
Basic Flow Of Event	1. Aktor memilih menu *Event* pada sistem. 2. Sistem menampilkan halaman daftar *Event*. 3. Aktor melihat daftar *event*.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat daftar *event*.

16. *Use Case Scenario Cetak Laporan Booking Objek Wisata (SIMPADU-F-008)*

Berdasarkan Tabel 4.22, *use case* cetak laporan *booking* objek wisata memungkinkan aktor untuk melakukan cetak laporan *booking* objek wisata pada sistem. Aktor dari *use case* ini yaitu wisatawan.

Tabel 4.22 Cetak Laporan *Booking* Objek Wisata (SIMPADU-F-008)

Brief Description	*Use case* ini memungkinkan aktor untuk mencetak laporan *booking* objek wisata
Actor	Wisatawan
Pre-condition	Aktor sudah masuk di *dashboard* Wisatawan.
Basic Flow Of Event	1. Aktor memilih menu *Booking* pada sistem. 2. Sistem menampilkan halaman riwayat *booking*. 3. Aktor menekan tombol "*view detail*". 4. Sistem menampilkan halaman *view detail* laporan *booking* objek wisata. 5. Aktor menekan tombol "cetak" 6. Sistem akan menampilkan halaman cetak laporan *booking* objek wisata 7. Aktor melakukan cetak laporan *booking* tempat wisata.
Alternative Flow	Tidak ada
Subflows	Tidak ada

Post-Condition	Aktor berhasil melakukan cetak laporan *booking* tempat wisata.

17. *Use Case Scenario* Melihat Detail Booking Objek Wisata (SIMPADU-F-012)

Berdasarkan Tabel 4.23, *use case* melihat detail *booking* objek wisata memungkinkan aktor untuk melihat detail informasi *booking* objek wisata yang telah dilakukannya pada sistem. Aktor dari *use case* ini yaitu wisatawan.

Tabel 4.23 Melihat Detail *Booking* Objek Wisata (SIMPADU-F-012)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat detail *booking* objek wisata
Actor	Wisatawan
Pre-condition	Aktor sudah masuk di *dashboard* Wisatawan.
Basic Flow Of Event	1. Aktor memilih menu *Booking* pada sistem. 2. Sistem menampilkan halaman riwayat *booking*. 2. Aktor menekan tombol "*view detail*". 3. Sistem menampilkan halaman *view detail* laporan booking objek *wisata*.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat *detail booking* objek wisata.

18. *Use Case Scenario* Melihat Detail Info *Event* (SIMPADU-F-011)

Berdasarkan Tabel 4.24, *use case* melihat detail info *event* memungkinkan aktor untuk melihat detail informasi *event* yang akan berlangsung ataupun yang belum pada sistem. Aktor dari *use case* ini yaitu wisatawan.

Tabel 4.24 Melihat Detail Info *Event* (SIMPADU-F-011)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat detail info *event*
Actor	*Admin* dan Wisatawan
Pre-condition	Aktor sudah masuk di masing-masing *dashboard*.
Basic Flow Of Event	1. Aktor memilih menu *Event* pada sistem. 2. Sistem menampilkan halaman daftar *event*. 3. Aktor menekan tombol "*view detail*" di salah satu *event*. 4. Sistem menampilkan halaman *view detail* info *event.*
Alternative Flow	Tidak ada

Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat *detail* info *event*.

19. *Use Case Scenario* Melihat Daftar *Administrator* (SIMPADU-F-023)

Berdasarkan Tabel 4.25, *use case* melihat daftar *administrator* memungkinkan aktor untuk melihat daftar *administrator* yang aktif pada sistem. Aktor dari *use case* ini yaitu *admin*.

Tabel 4.25 Melihat Daftar *Administrator* (SIMPADU-F-023)

Brief Description	*Use case* ini memungkinkan aktor untuk melihat daftar *administrator*
Actor	*Admin*
Pre-condition	Aktor sudah masuk di *dashboard Admin*.
Basic Flow Of Event	1. Aktor memilih menu Halaman *Administrator* pada sistem. 2. Sistem menampilkan halaman daftar *administrator*. 3. Aktor melihat daftar *administrator*.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil melihat daftar *administrator*.

20. *Use Case Scenario* Menambah *Administrator* (SIMPADU-F-025)

Berdasarkan Tabel 4.26, *use case* menambah *administrator* memungkinkan aktor untuk menambahkan *administrator* baru pada sistem. Aktor dari *use case* ini yaitu *admin*.

Tabel 4.26 Menambah *Administrator* (SIMPADU-F-025)

Brief Description	*Use case* ini memungkinkan aktor untuk menambahkan *administrator* baru pada sistem.
Actor	*Admin*
Pre-condition	Aktor sudah masuk di *dashboard Admin*.
Basic Flow Of Event	1. Aktor memilih menu Halaman *Administrator* pada sistem. 2. Sistem menampilkan halaman daftar *administrator*. 2. Aktor menekan tombol "*tambah*" pada salah satu *administrator*. 3. Sistem menampilkan halaman tambah *administrator*. 4. Aktor mengisi data pada *form* tambah *administrator*. 5. Aktor menekan tombol "tambah".

	6. Sistem menampilkan pesan "berhasil menambahkan *administrator*" dan refresh ke halaman daftar *administrator*. 7. Aktor melihat pesan berhasil dan daftar *administrator*.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil menambah *administrator*.

21. *Use Case Scenario* Menghapus *Administrator* (SIMPADU-F-024)

Berdasarkan Tabel 4.27, *use case* menghapus *administrator* memungkinkan aktor untuk menghapus *administrator* yang tidak aktif pada sistem. Aktor dari *use case* ini yaitu *admin*.

Tabel 4.27 Menghapus *Administrator* (SIMPADU-F-024)

Brief Description	*Use case* ini memungkinkan aktor untuk menghapus *administrator*
Actor	*Admin*
Pre-condition	Aktor sudah masuk di *dashboard Admin*.
Basic Flow Of Event	1. Aktor memilih menu Halaman *Administrator* pada sistem. 2. Sistem menampilkan halaman daftar *administrator*. 3. Aktor menekan tombol "hapus" pada salah satu *administrator* pada daftar *administrator*. 4. Sistem refresh ke halaman daftar *administrator* .
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil menghapus *administrator*.

22. *Use Case Scenario* Mengupdate *Event* (SIMPADU-F-026)

Berdasarkan Tabel 4.28, *use case* mengupdate *event* memungkinkan aktor untuk melakukan *update* informasi *event* jika ada peubahan pada sistem. Aktor dari *use case* ini yaitu *admin*.

Tabel 4.28 Mengupdate *Event* (SIMPADU-F-026)

Brief Description	*Use case* ini memungkinkan aktor untuk mengupdate informasi *event*
Actor	*Admin*
Pre-condition	Aktor sudah masuk di *dashboard Admin*.

Basic Flow Of Event	1. Aktor memilih menu Halaman *Event* pada sistem. 2. Sistem menampilkan halaman daftar *event*. 3. Aktor menekan tombol "view detail" pada salah satu *event* pada daftar *event*. 4. Sistem menampilkan detail informasi *event*. 5. Aktor menekan tombol "*update*" pada halaman detail informasi *event*. 6. Sistem menampilkan form *update event*. 7. Aktor menekan tombol "*OK*". 8. Sistem refresh ke halaman daftar *event*.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil mengupdate informasi *event*.

23. *Use Case Scenario* Mengupdate Objek Wisata (SIMPADU-F-017)

Berdasarkan Tabel 4.29, *use case* mengupdate objek wisata memungkinkan aktor untuk melakukan *update* informasi objek wisata pada sistem. Aktor dari *use case* ini yaitu pengelola.

Tabel 4.29 Mengupdate Objek Wisata (SIMPADU-F-017)

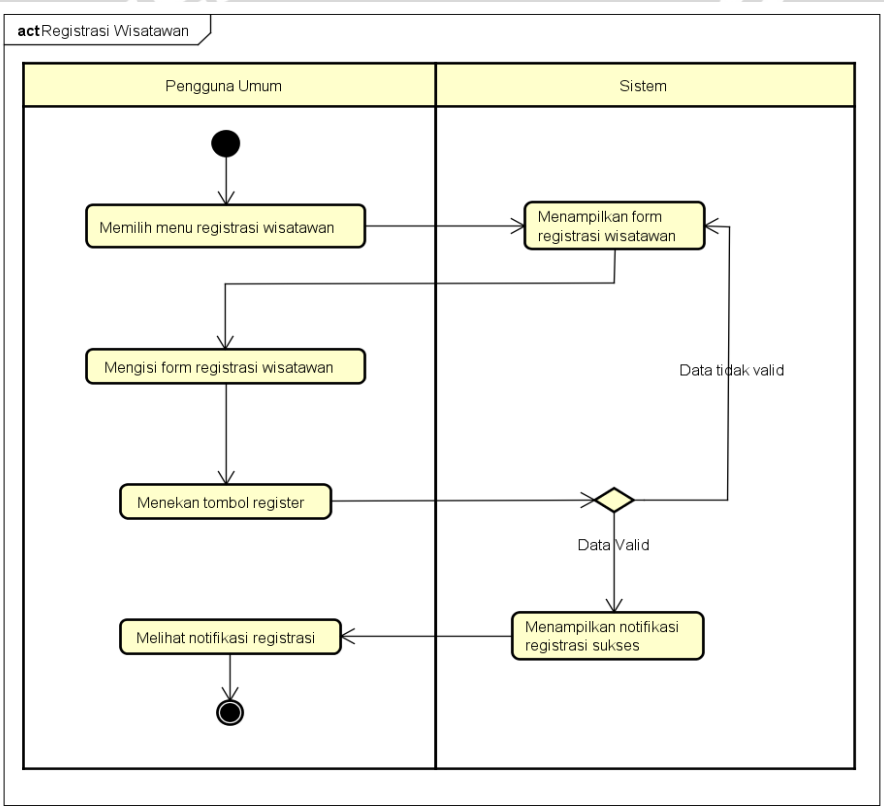
Brief Description	*Use case* ini memungkinkan aktor untuk mengupate informasi objek wisata
Actor	Pengelola
Pre-condition	Aktor sudah masuk di *dashboard* Pengelola.
Basic Flow Of Event	1. Aktor memilih menu Halaman Objek Wisata pada sistem. 2. Sistem menampilkan halaman daftar objek wisata. 3. Aktor menekan tombol "view detail" pada salah satu objek wisata pada daftar objek wisata. 4. Sistem menampilkan detail informasi objek wisata. 5. Aktor menekan tombol "*update*" pada halaman detail informasi objek wisata. 6. Sistem menampilkan form *update* objek wisata. 7. Aktor menekan tombol "*OK*". 8. Sistem refresh ke halaman daftar objek wisata.
Alternative Flow	Tidak ada
Subflows	Tidak ada
Post-Condition	Aktor berhasil menghapus *administrator*.

4.4.2 Activity Diagram

Activity diagram berisi penggambaran aliran kerja antara *user* dengan sistem. Perancangan *activity diagram* ini bertujuan untuk menggambarkan urutan aktivitas dari proses setiap *use case* yang sudah ada. Berikut ini merupakan gambar dari *activity diagram* dari sistem informasi manajemen pariwisata terpadu.

1. *Activity Diagram* Registrasi Wisatawan

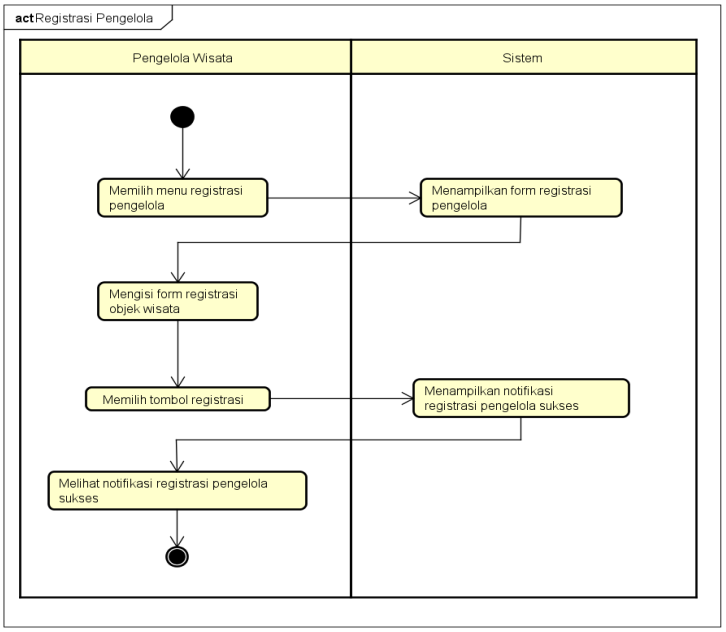
Activity diagram registrasi pengelola adalah *event* ketika user ingin mendaftarkan sebagai wisatawan untuk dapat melakukan aktivitas yang terdapat pada sistem. *Activity Diagram* ini dimulai ketika pengunjung memilih fungsi *register* sebagai wisatawan dan berakhir ketika sistem menampilkan pesan data sukses disimpan di basis data. *Activity diagram* registrasi wisatawan dapat dilihat pada Gambar 4.4.



Gambar 4.4 *Activity Diagram* Registrasi Wisatawan

2. *Activity Diagram* Registrasi Pengelola

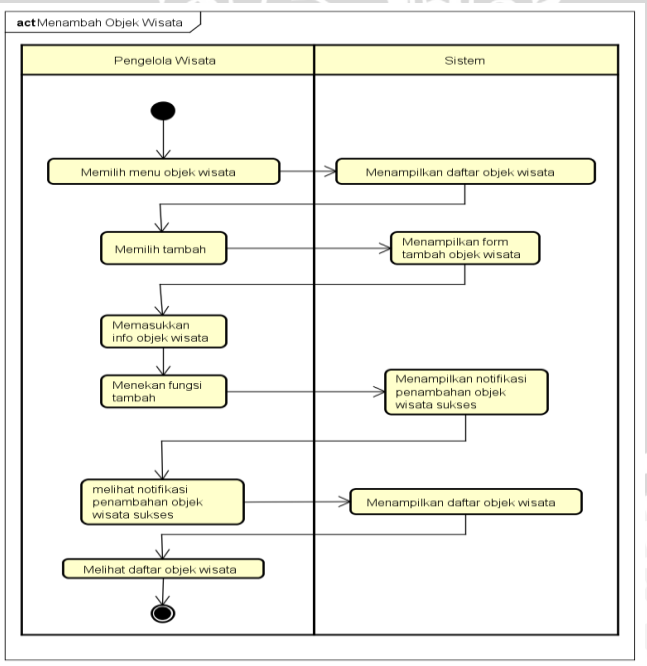
Activity diagram registrasi pengelola adalah *event* ketika *user* ingin mendaftarkan sebagai pengelola untuk dapat melakukan aktivitas yang terdapat pada sistem dan mendaftarkan objek wisata yang dikelolanya pada sistem. *Activity Diagram* ini dimulai ketika pengunjung memilih fungsi *register* sebagai pengelola dan berakhir ketika sistem menampilkan pesan data sukses disimpan di basis data. *Activity diagram* registrasi wisatawan dapat dilihat pada Gambar 4.5.



Gambar 4.5 *Activity Diagram* Registrasi Pengelola

3. *Activity Diagram* Menambah Objek Wisata

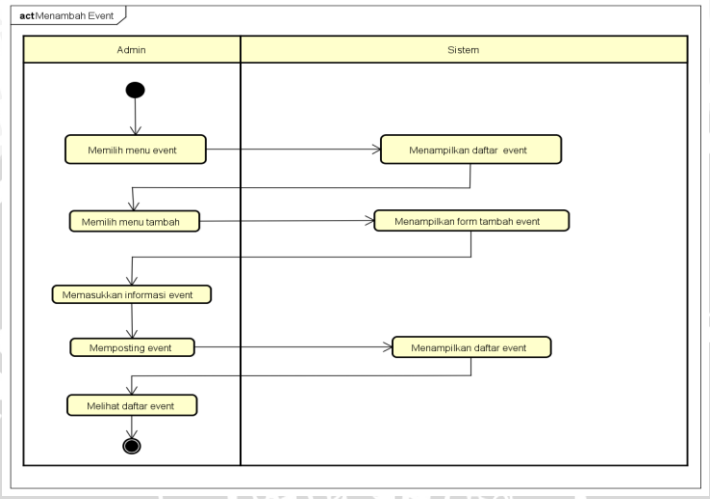
Activity diagram menambah objek wisata adalah *event* dimana *user* yang *login* sebagai pengelola dapat menambahkan objek wisata di Kabupaten Banyuwangi yang dikelolanya ke dalam sistem. *Activity diagram* ini dimulai ketika pengelola memilih menu objek wisata kemudian menekan tombol tambah dan berakhir ketika sistem menampilkan pesan tambah objek wisata sukses terkirim. *Activity diagram* menambah objek wisata dapat dilihat pada Gambar 4.6.



Gambar 4.6 *Activity Diagram* Menambah Objek Wisata

4. *Activity Diagram* Menambah *Event*

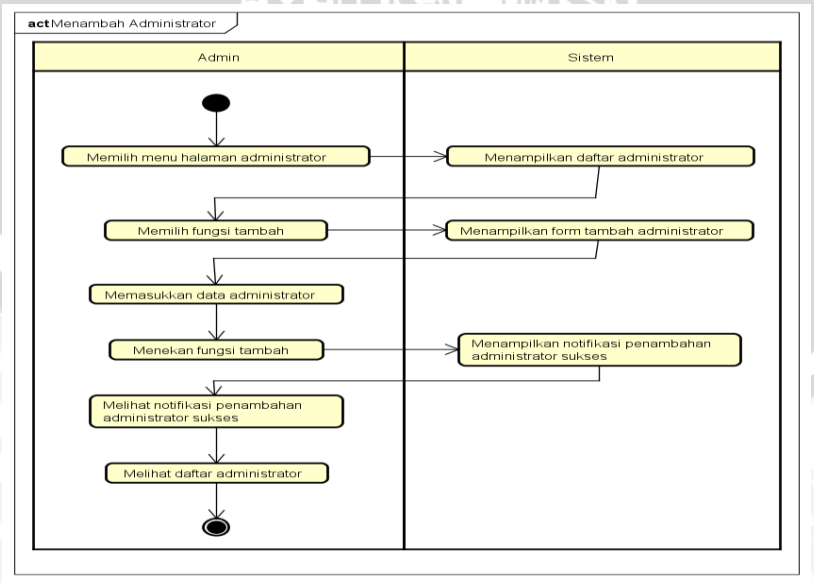
Activity diagram menambah *event* adalah *event* dimana *user* yang *login* sebagai *admin* dapat menambah *event* yang akan berlangsung di Kabupaten Banyuwangi setiap bulannya pada sistem. *Activity diagram* ini dimulai ketika *admin* memilih menu *event* kemudian menekan tombol tambah dan berakhir ketika sistem menampilkan pesan tambah *event* sukses terkirim. *Activity diagram* menambah *event* dapat dilihat pada Gambar 4.7.



Gambar 4.7 *Activity Diagram* Menambah *Event*

5. *Activity Diagram* Menambah *Administrator*

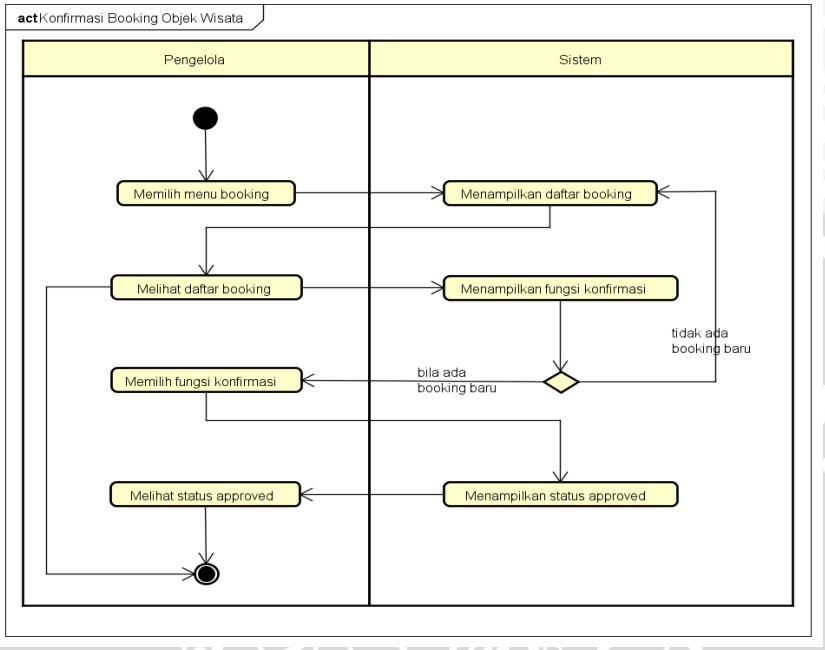
Activity diagram menambah *administrator* adalah *event* dimana *user* yang *login* sebagai *admin* dapat menambah *administrator* baru sesuai dengan kebutuhan yang dibutuhkan oleh *admin* untuk mengakses sistem. *Activity diagram* menambah *administrator* dapat dilihat pada Gambar 4.8.



Gambar 4.8 *Activity Diagram* Menambah *Administrator*

6. *Activity Diagram* Konfirmasi *Booking* Objek Wisata

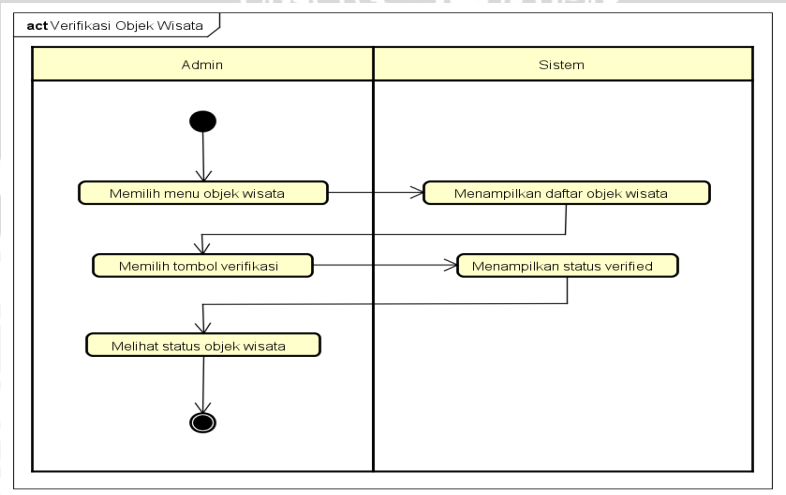
Activity diagram konfirmasi *booking* objek wisata adalah *event* dimana *user* yang *login* sebagai pengelola dapat melakukan konfirmasi *booking* objek wisata yang masuk pada sistem. *Activity diagram* konfirmasi *booking* objek wisata dapat dilihat pada gambar 4.9.



Gambar 4.9 *Activity Diagram* Konfirmasi *Booking* Objek Wisata

7. *Activity Diagram* Verifikasi Objek Wisata

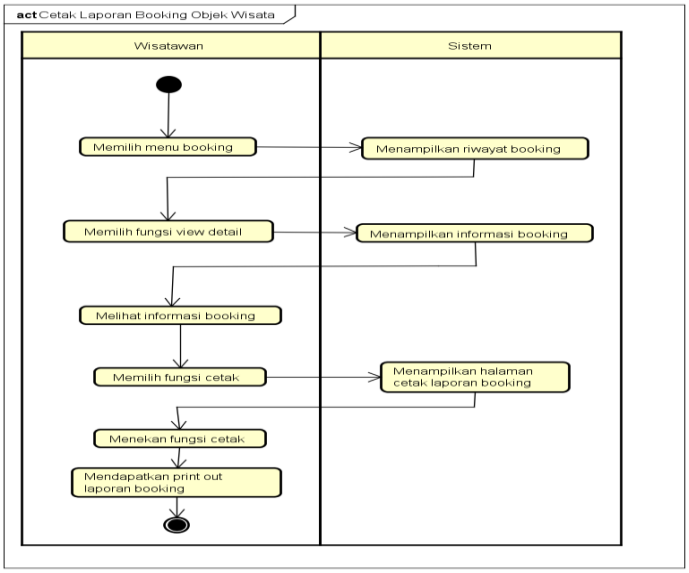
Activity diagram verifikasi objek wisata adalah *event* dimana *user* yang *login* sebagai *admin* dapat melakukan verifikasi objek wisata yang masuk sesuai dengan data yang valid dan dapat dipertanggungjawabkan pada sistem. *Activity diagram* verifikasi objek wisata dapat dilihat pada gambar 4.10.



Gambar 4.10 *Activity Diagram* Verifikasi Objek Wisata

8. *Activity Diagram* Cetak Laporan *Booking* Objek Wisata

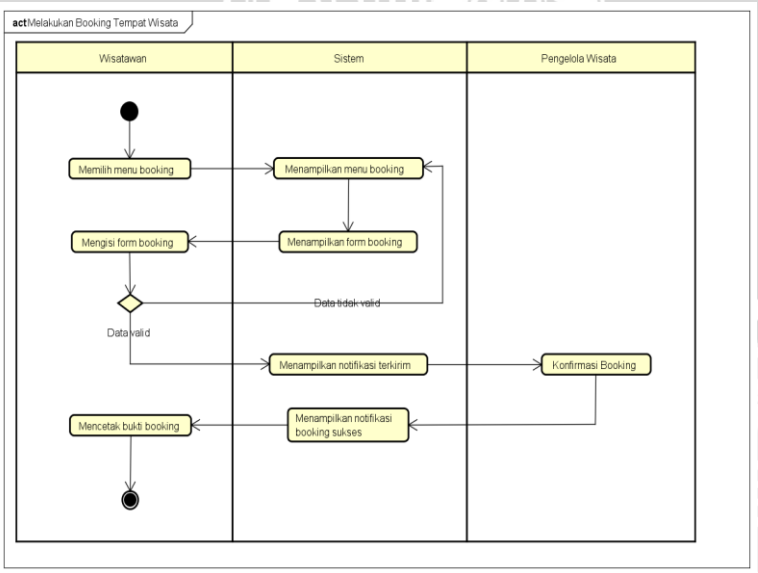
Activity diagram cetak laporan *booking* objek wisata adalah *event* dimana *user* yang *login* sebagai wisatawan dapat melakukan cetak laporan *booking* objek wisata setelah *booking* yang dilakukan sebelumnya sudah dikonfirmasi oleh pengelola objek wisata pada sistem. *Activity diagram* cetak laporan *booking* objek wisata dapat dilihat pada Gambar 4.11.



Gambar 4.11 *Activity Diagram* Cetak Laporan *Booking* Objek Wisata

9. *Activity Diagram Booking* Objek Wisata

Activity diagram booking objek wisata adalah *event* dimana *user* yang *login* sebagai wisatawan dapat melakukan *booking* terhadap objek wisata yang akan dikunjungi melalui sistem. *Activity diagram booking* objek wisata dapat dilihat pada Gambar 4.12.



Gambar 4.12 *Activity Diagram* Melakukan *Booking* Objek Wisata

BAB 5 PERANCANGAN

5.1 Perancangan Arsitektur

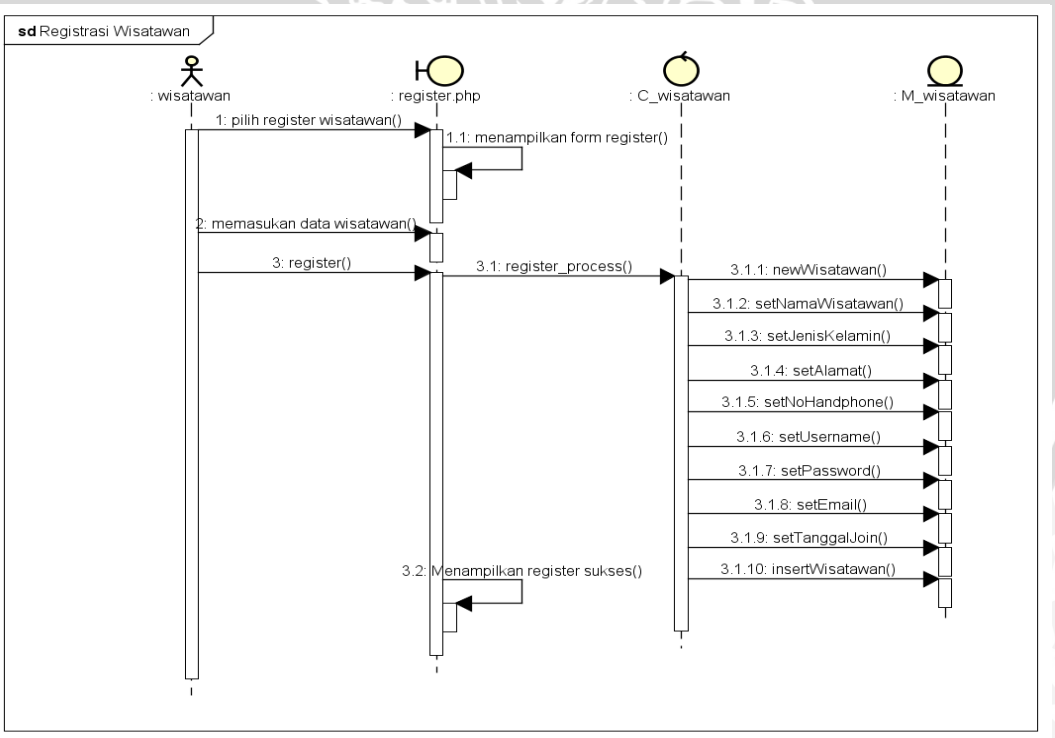
Pada perancangan arsitektur ini penulis menggunakan sequence diagram dan class diagram yang dirancang dari penjabaran analisis kebutuhan dari sistem.

5.1.1 Sequence Diagram

Sequence diagram digambarkan interaksi antara aktor dengan komponen sistem dan interaksi antara komponen antar sistem dalam bentuk diagram. *Sequence diagram* dirancang dengan mengacu pada *framework* CodeIgniter dimana komponen sistem disusun sebagai *boundary object* yang merupakan representasi dari *view (interface), control object* merupakan representasi dari *controller* (penengah antara *boundary* dan *entity*) dan *entity object* merupakan representasi dari model (sistem data).

1. *Sequence Diagram* Registrasi Wisatawan

Sequence diagram registrasi wisatawan adalah *event* dimana *user* melakukan registrasi sebagai wisatawan agar terdaftar pada sistem. *Sequence diagram* registrasi wisatawan dapat dilihat pada gambar 5.6.

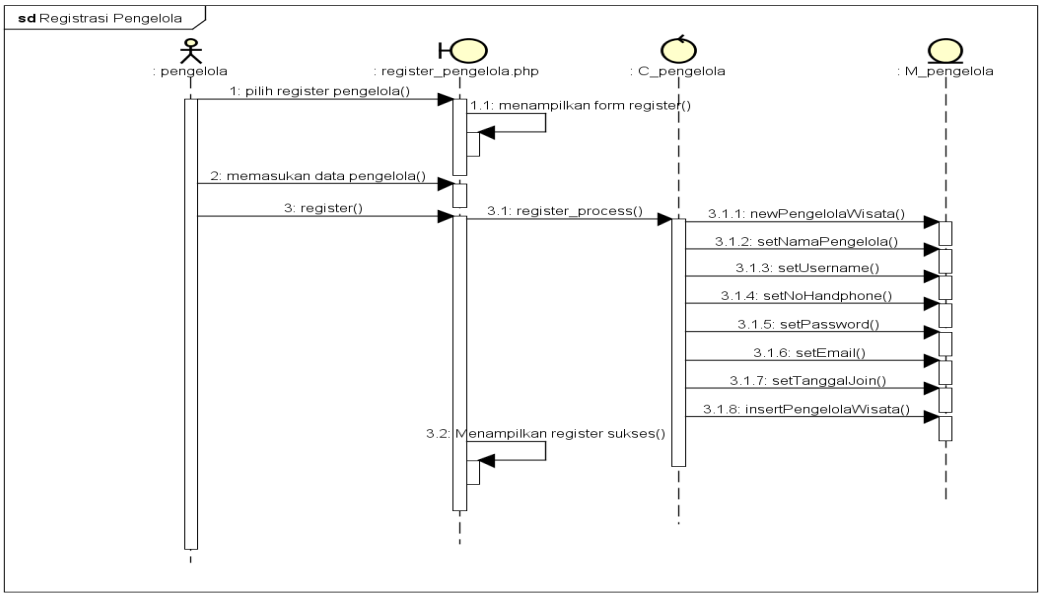


Gambar 5.1 *Sequence Diagram* Registrasi Wisatawan

2. *Sequence Diagram* Registrasi Pengelola

Sequence diagram registrasi pengelola adalah *event* dimana *user* melakukan registrasi sebagai pengelola agar terdaftar pada sistem. Registrasi pengelola

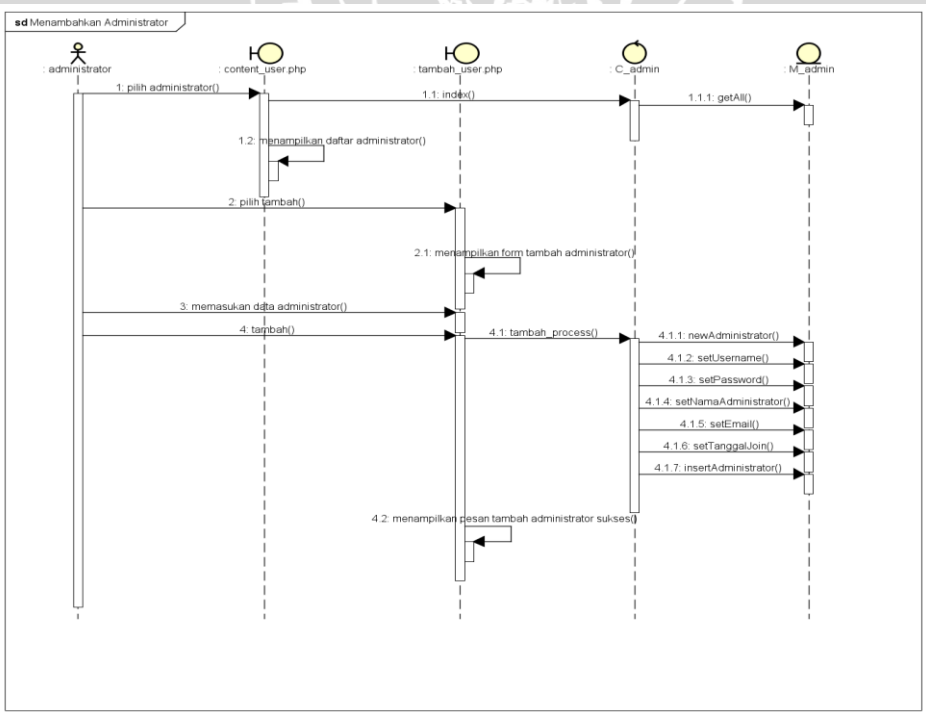
dikhususkan pada user yang memiliki objek wisata dibawah pengelolaannya. *Sequence diagram* registrasi wisatawan dapat dilihat pada gambar 5.6.



Gambar 5.2 *Sequence Diagram* Registrasi Pengelola

3. Sequence Diagram Menambah *Administrator*

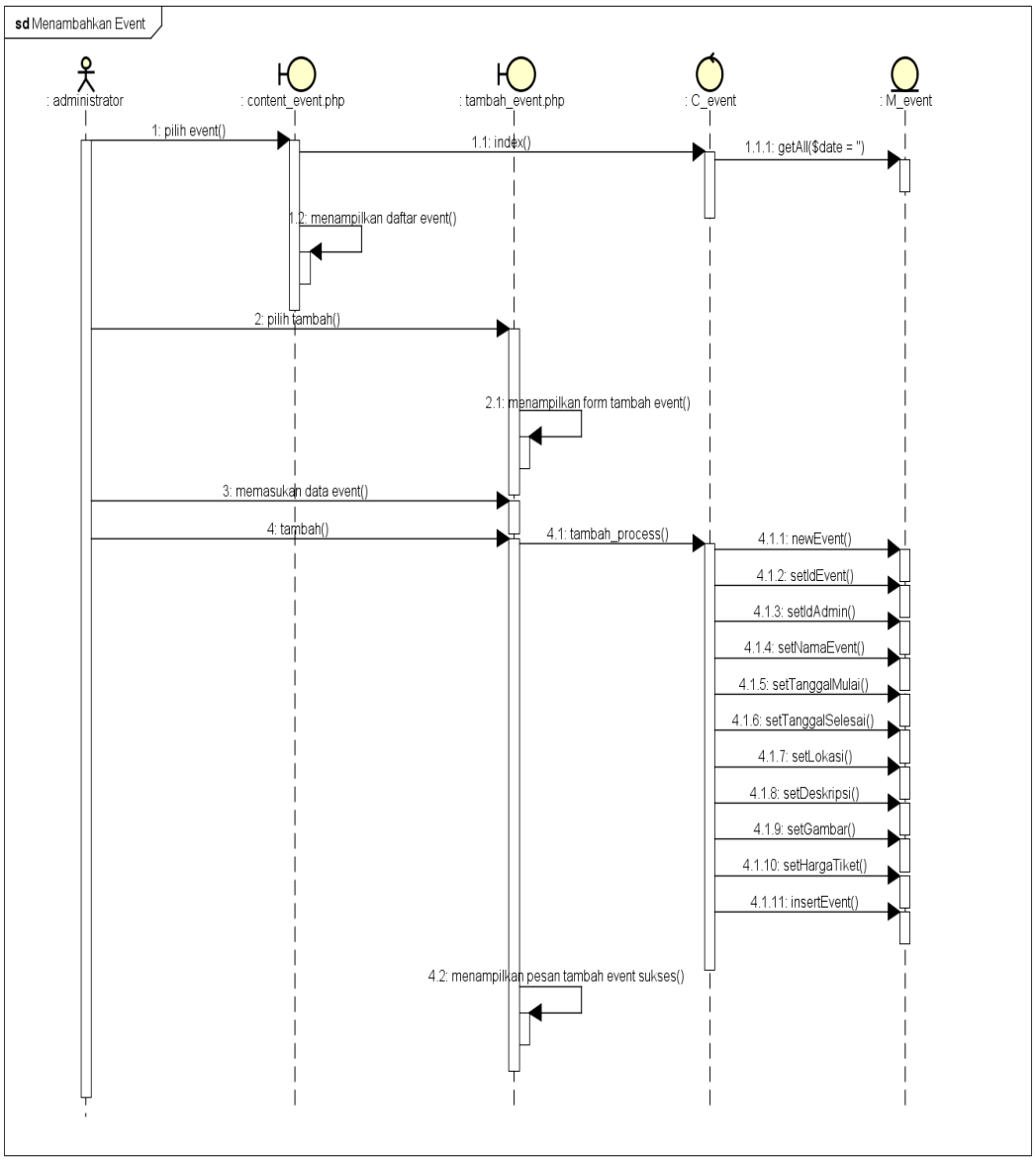
Sequence diagram menambah *administrator* adalah *event* dimana *user* sebagai *admin* dapat menambah *administrator* baru pada sistem. *Sequence diagram* menambah*administraor* dapat dilihat pada gambar 5.18.



Gambar 5.3 Sequence Diagram Menambah Administrator

4. *Sequence Diagram* Menambah *Event*

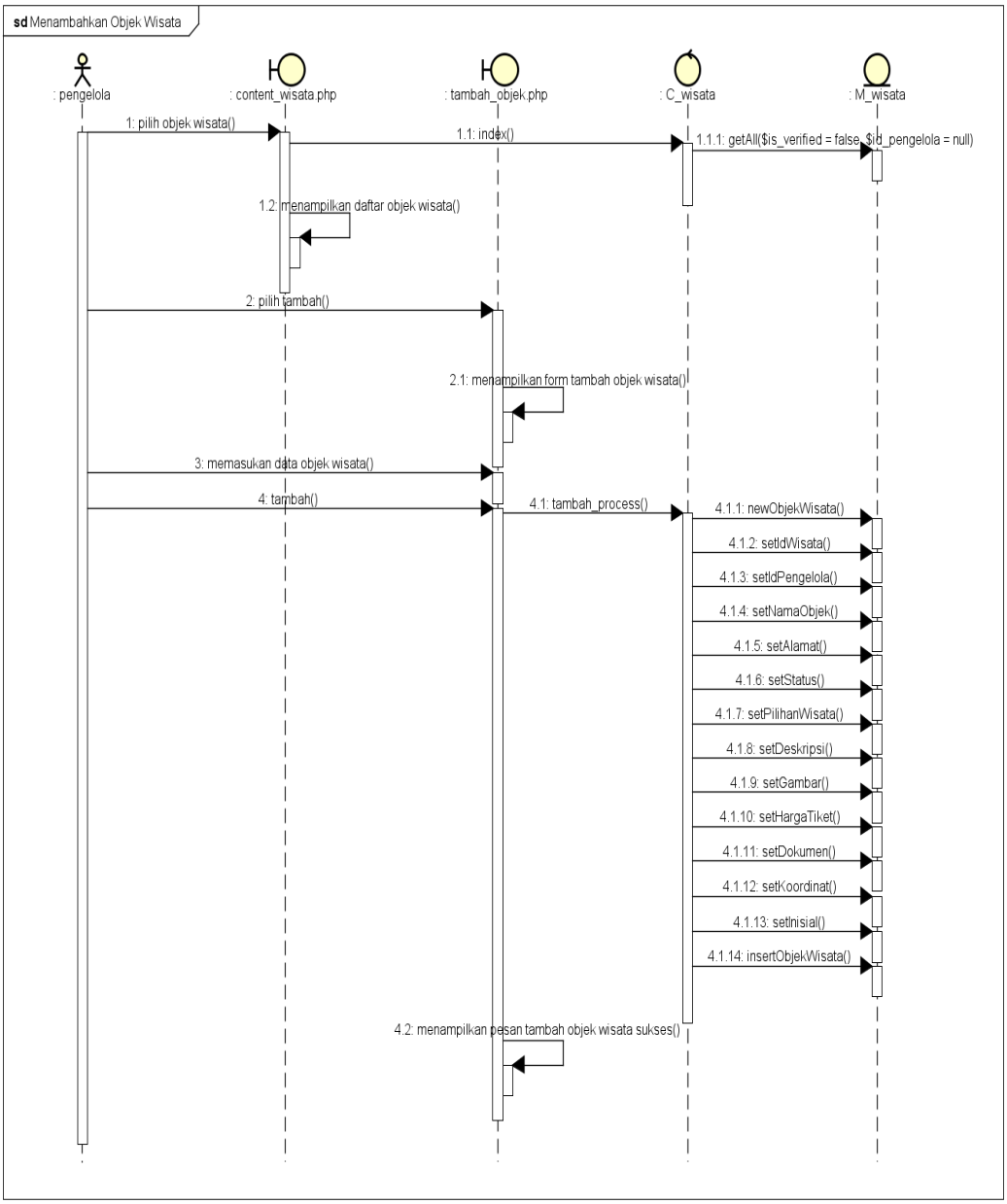
Sequence diagram menambah *event* adalah *event* dimana *user* sebagai admin dapat menambah *event* yang akan berlangsung pada sistem. *Sequence diagram* menambah *event* dapat dilihat pada gambar 5.19.



Gambar 5.4 *Sequence Diagram* Menambah *Event*

5. *Sequence Diagram* Menambah Objek Wisata

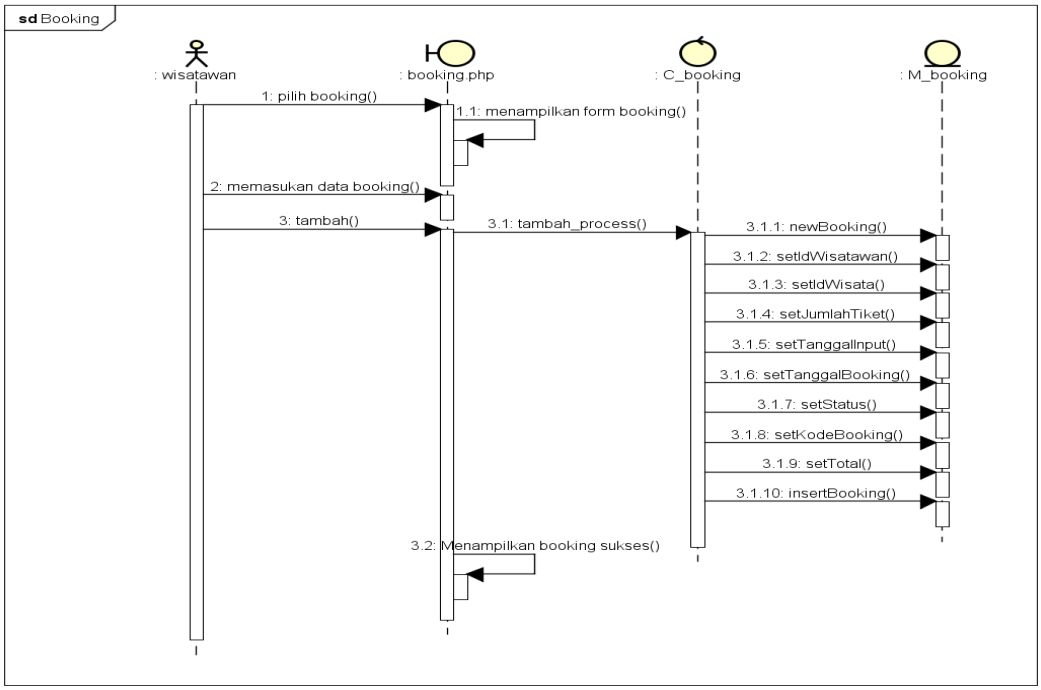
Sequence diagram menambah objek wisata adalah *event* dimana *user* sebagai pengelola dapat menambahkan objek wisata yang masih dikelolanya pada sistem. *Sequence diagram* melihat daftar objek wisata dapat dilihat pada gambar 5.20.



Gambar 5.5 *Sequence Diagram* Menambah Objek Wisata

6. *Sequence Diagram Booking* Objek Wisata

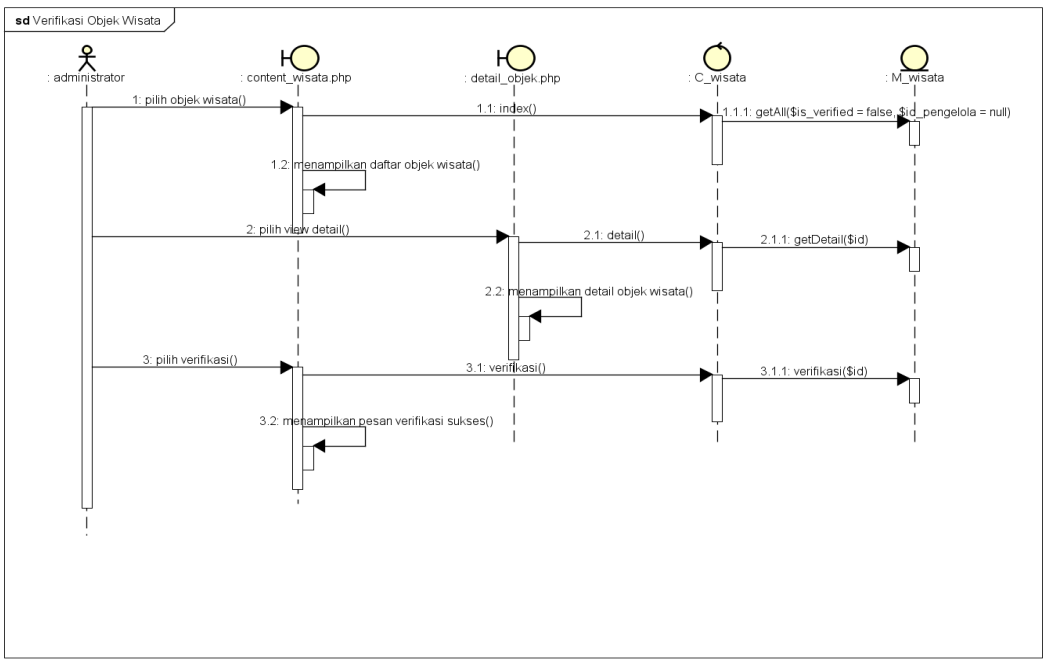
Sequence diagram booking objek wisata adalah *event* dimana *user* sebagai wisatawan dapat melakukan *booking* objek wisata yang ingin dikunjungi pada sistem. *Sequence diagram booking* objek wisata dapat dilihat pada gambar 5.23.



Gambar 5.6 *Sequence Diagram Booking* Objek Wisata

7. *Sequence Diagram* Verifikasi Objek Wisata

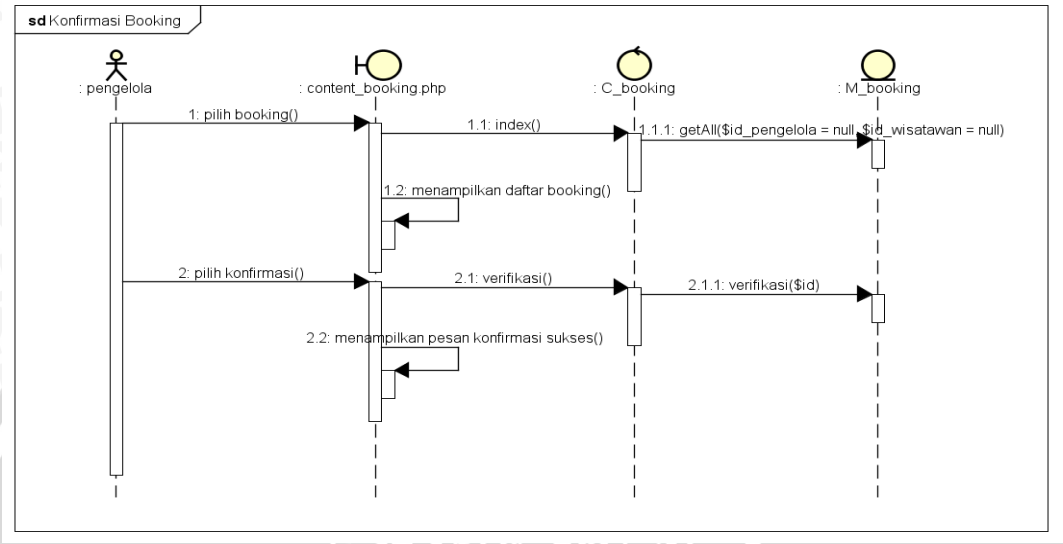
Sequence diagram Verifikasi objek wisata adalah *event* dimana *user* sebagai *admin* dapat melakukan verifikasi objek wisata yang sudah sesuai dengan data-data yang dibutuhkan pada sistem. *Sequence diagram* verifikasi objek wisata dapat dilihat pada gambar 5.24.



Gambar 5.7 *Sequence Diagram* Verifikasi Objek Wisata

8. *Sequence Diagram* Konfirmasi *Booking* Objek Wisata

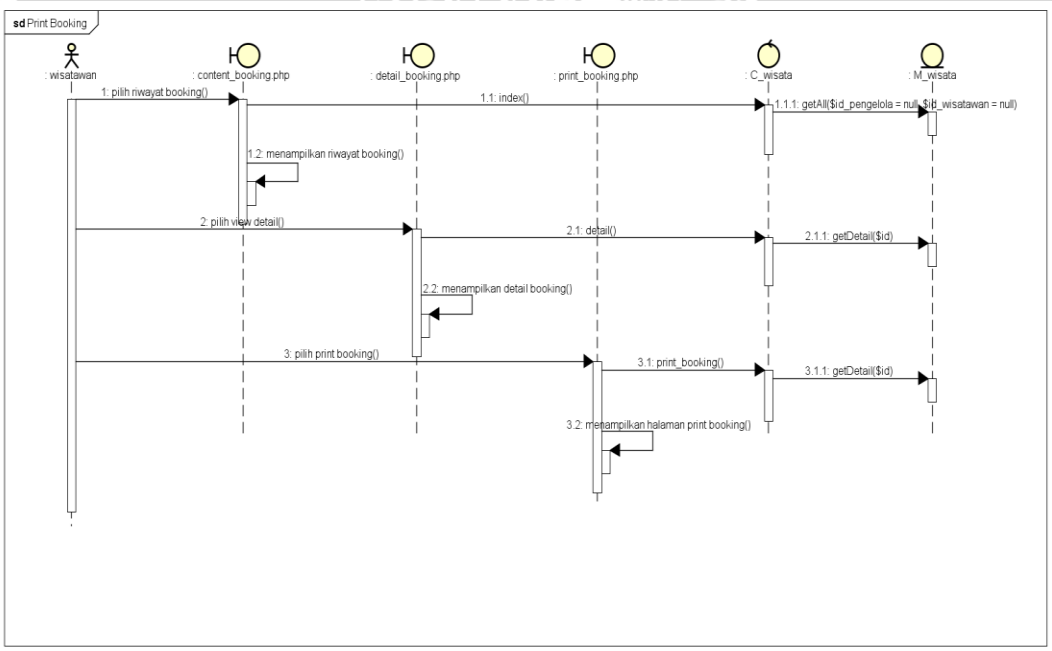
Sequence diagram konfirmasi *booking* objek wisata adalah *event* dimana *user* sebagai pengelola dapat melakukan konfirmasi *booking* objek wisata sesuai dengan objek wisata yang dikelolanya pada sistem. *Sequence diagram* menghapus *administrator* dapat dilihat pada gambar 5.25.



Gambar 5.8 *Sequence Diagram* Konfirmasi *Booking* Objek Wisata

9. *Sequence Diagram* Cetak Laporan *Booking* Objek Wisata

Sequence diagram cetak laporan *booking* objek wisata adalah *event* dimana *user* sebagai wisatawan dapat melakukan cetak laporan booking objek wisata untuk bukti saat menuju objek wisata yang dikunjungi pada sistem. *Sequence diagram* cetak laporan *booking* objek wisata dapat dilihat pada gambar 5.26.



Gambar 5.9 *Sequence Diagram* Cetak Laporan *Booking* Objek Wisata

5.1.2 Class Diagram

Class diagram merupakan sebuah *diagram* yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas terdiri dari 3 bagian utama yaitu *attribute*, *operation*, dan *name*.

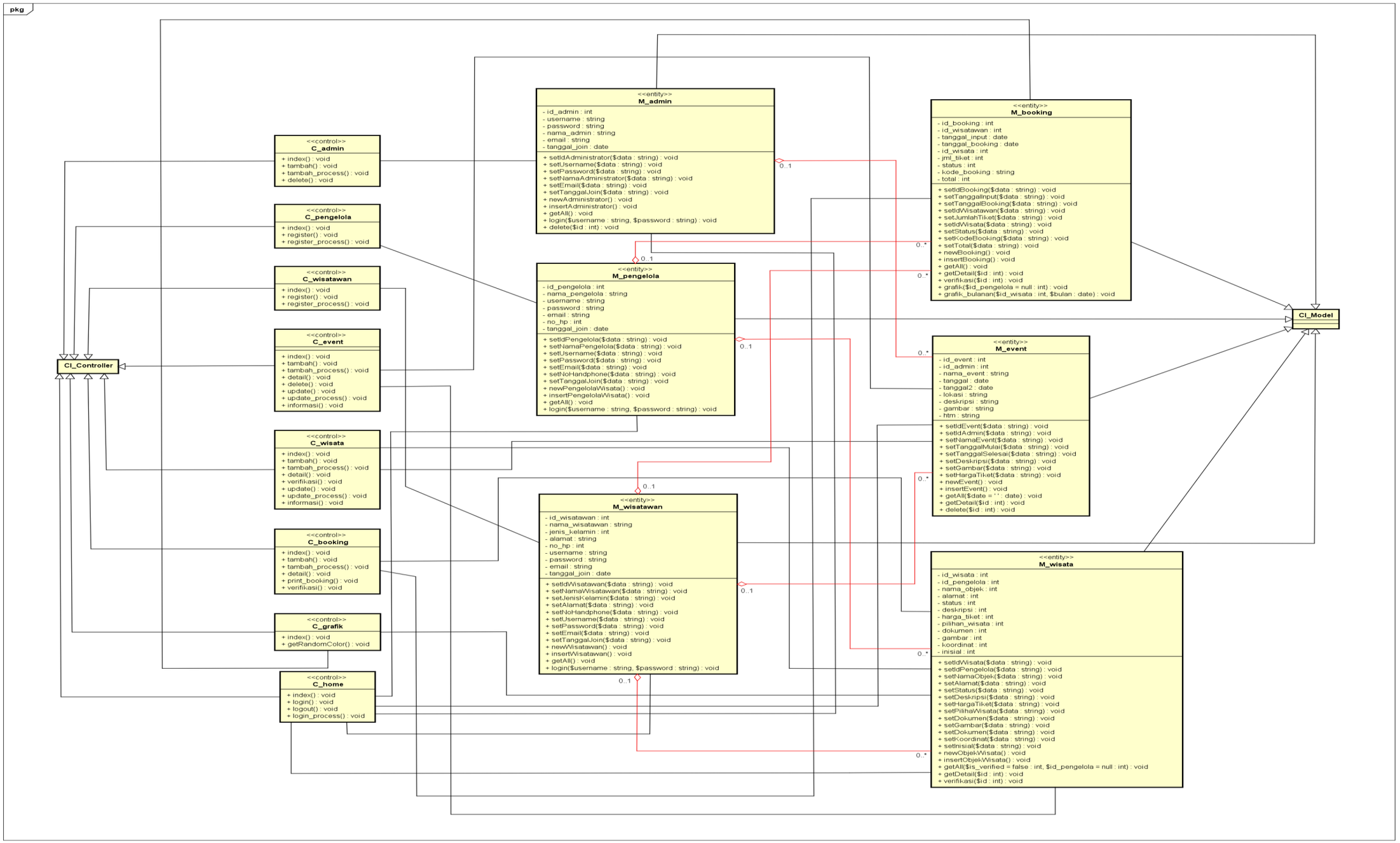
Class diagram ini dirancang dengan konsep MVC (*Model View Controller*) menggunakan *framework CodeIgniter*. Perancangan class diagram mengacu pada sequence diagram. Didalam kelas-kelas tersebut terdapat operasi-operasi yang disesuaikan dengan sequence diagram yang nantinya digunakan dalam proses implementasi sistem.

Setiap class yang memiliki *stereotype <<control>>* akan digunakan untuk menangani proses-proses tertentu sesuai instruksi dari pengguna sistem. Selanjutnya, *<<entity>>* mempunyai peran penting dalam proses interaksi dengan database.

Pada Gambar 5.1, garis penghubung antara *class controller* dan *class model* menggunakan garis *association* berwarna hitam sedangkan garis penghubung antara *class model* dan *class model* menggunakan garis *aggregation* berwarna merah. Dalam konsep MVC, class dengan *stereotype <<entity>>* biasa disebut dengan model. Sedangkan *class* dengan *stereotype <<control>>* bisa juga disebut dengan *controller* dan *class* dengan *stereotype <<boundary>>* dikenal sebagai *view*. Selain itu semua *class controller* akan digeneralalisasi ke *class CI_Controller* dan semua *class model* akan digeneralisasikan ke *class CI_Model*. Akan tetapi pada *class diagram* yang peneliti rancang tidak ditampilkan *class* dengan *stereotype <<boundary>>.*

Dalam *framework CodeIgniter* merupakan salah satu *framework* dalam bahasa pemrograman PHP yang menerapkan pola pemrograman MVC. Pada *framework CodeIgniter*, seluruh *view* langsung berinteraksi dengan *controller* tertentu dengan *class* dan fungsi yang telah siap digunakan untuk mengolah perintah tersebut. Setelah itu pada *class controller* akan mengakses *class model* untuk menerima dan memanggil data pada *database*.

repository.ub.ac.id



Gambar 5.10 *Class Diagram*

5.2 Perancangan Komponen

Dalam perancangan komponen dilakukan untuk mendefinisikan algoritma yang digunakan pada fungsi-fungsi yang telah didefinisikan sebelumnya. Selain itu dapat digunakan untuk mengecek apakah perangkat lunak yang akan dibuat sudah sesuai dan konsisten dengan perancangan. Perancangan komponen Sistem Informasi Manajemen Pariwisata Terpadu adalah sebagai berikut :

5.2.1 *Algoritma Fungsi register_process()*

Tabel 5.1 *Algoritma Fungsi register_process()*

1	Mulai
2	Memanggil data wisatawan baru dengan variabel “null” (kosong)
3	Menampung data masukan berupa “nama”
4	Menampung data masukan berupa “jenis_kelamin”
5	Menampung data masukan berupa “alamat”
6	Menampung data masukan berupa “no_handphone”
7	Menampung data masukan berupa “username”
8	Menampung data masukan berupa “password”
9	Menampung data masukan berupa “email”
10	Melakukan setting “tanggal_join” secara otomatis
11	Selesai

5.2.2 *Algoritma Fungsi login()*

Tabel 5.2 *Algoritma Fungsi login()*

1	Mulai
2	Menampilkan halaman header
3	Menampilkan halaman login
4	Selesai

5.2.3 *Algoritma Fungsi tambah()*

Tabel 5.3 *Algoritma Fungsi tambah()*

1	Mulai
2	Menampilkan halaman header
3	Menampilkan halaman menu
4	Menampilkan halaman formulir tambah event

5	Menampilkan halaman footer
6	Selesai

5.2.4 *Algoritma Fungsi update_process()*

Tabel 5.4 *Algoritma Fungsi update_process()*

1	Mulai
2	Menampung data informasi event kedalam bentuk array, yaitu:
3	Memasukkan data “lokasi
4	Memasukkan data “tanggal”
5	Memasukkan data “tanggal2”
6	Memasukkan data “deskripsi”
7	Memasukkan data “htm”
8	Memanggil data event yang memiliki id sesuai dengan parameter
9	Dimana id_event harus sesuai dengan data yang dimasukkan
10	Melakukan perubahan data informasi event sesuai dengan tabel
11	Mengarahkan ke halaman detail event sesuai dengan id_event yang telah diupdate
12	Selesai

5.2.5 *Algoritma Fungsi informasi($id_event)*

Tabel 5.5 *Algoritma Fungsi informasi()*

1	Mulai
2	Menampung data infomasi event ke dalam bentuk array, yaitu:
3	Memanggil informasi detail event yang sesuai dengan id_event
4	Memanggil semua objek wisata yang terdapat pada daabase
5	Memanggil semua event yang terdapat pada database
6	Menampilkan halaman informasi event
7	Selesai

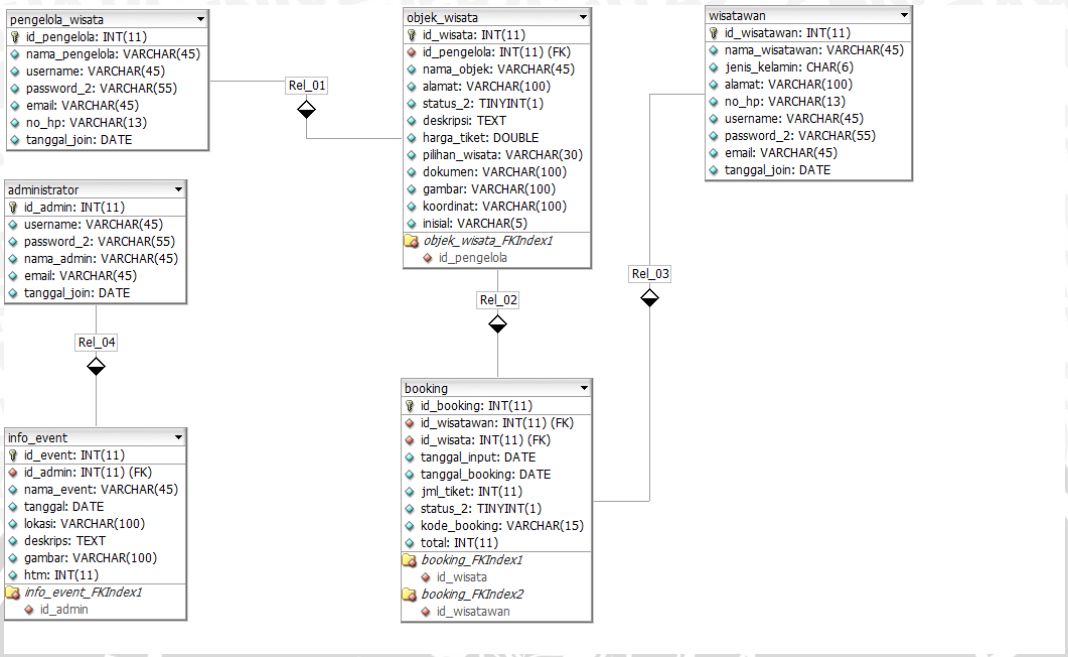
5.3 Perancangan Data

Dalam perancangan data ini penulis menggunakan *physical data models* pada sistem data sesuai dengan sistem manajemen *database* yang diinginkan.

5.3.1 *Physical Data Models*

Dalam *physical data models*, sebuah *database* merupakan sekumpulan relasi antar tabel yang saling terhubung satu sama lainnya. Terdapat nama tabel,

attribut, dan relasi. Berikut ini merupakan *physical data models* pada sistem informasi manajemen pariwisata terpadu seperti pada Gambar 5.29.



Gambar 5.11 *Physical Data Models*

5.4 Perancangan Antarmuka

Perancangan antarmuka digunakan untuk menggambarkan perancangan antarmuka sistem informasi manajemen pariwisata terpadu. Perancangan antarmuka pada penelitian ini dibagi menjadi 3 yaitu admin, pengelola dan wisatawan.

5.4.1 Perancangan Antarmuka Admin

Berikut merupakan antarmuka untuk Admin.

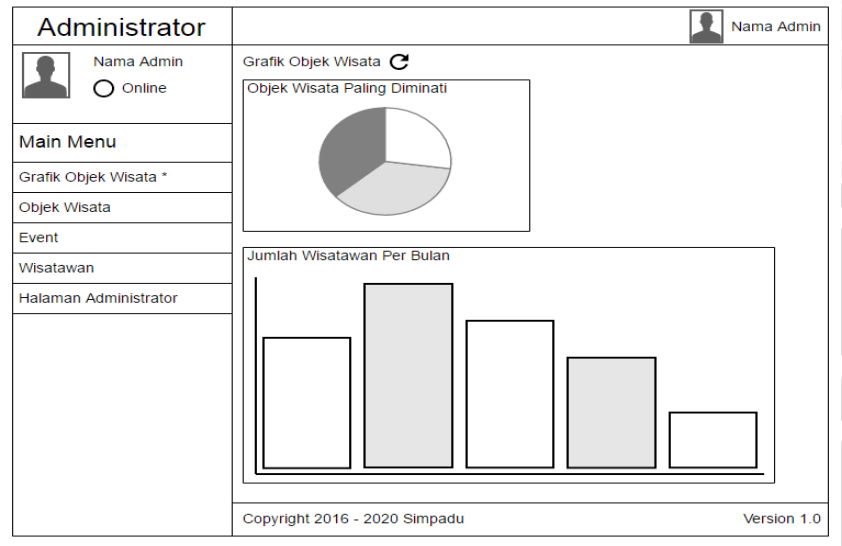
1. Halaman *Login*

Halaman *login* merupakan halaman pengguna umum. Halaman ini menampilkan *form login* untuk masuk ke dalam sistem sesuai dengan hak akses yang diberikan oleh sistem. Berikut ini merupakan rancangan antarmuka halaman *login* pada gambar 5.30.

Gambar 5.12 Halaman *Login*

2. Halaman Grafik Objek Wisata

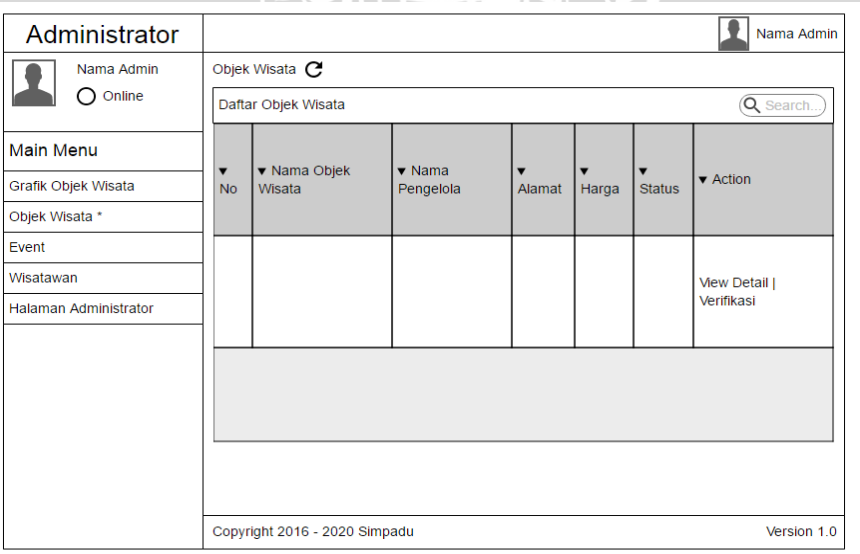
Halaman grafik objek wisata merupakan halaman dari *admin*. Halaman ini menampilkan jumlah wisatawan setiap bulannya pada masing-masing objek wisata dan menampilkan objek wisata yang paling sering dikunjungi oleh wisatawan dalam bentk grafik. Berikut ini merupakan rancangan antarmuka halama grafik objek wisata pada Gambar 5.31.



Gambar 5.13 Halaman Grafik Objek Wisata

3. Halaman Objek Wisata

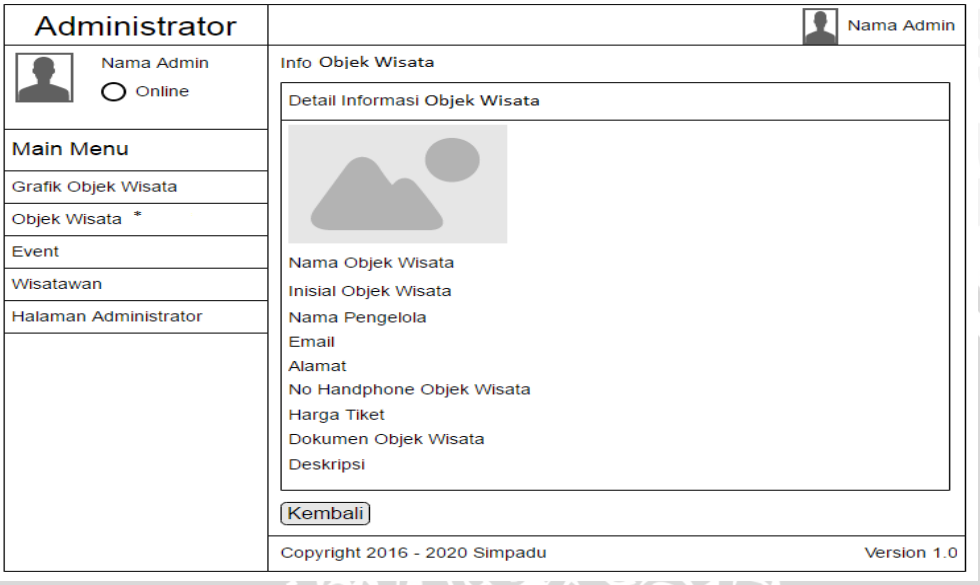
Halaman objek wisata merupakan halaman dari *admin*. Halaman ini menampilkan daftar objek wisata yang sudah diverifikasi maupun belum diverifikasi oleh *admin*. Berikut ini merupakan rancangan antarmuka halaman objek wisata pada Gambar 5.32.



Gambar 5.14 Halaman Objek Wisata

4. Halaman Detail Objek Wisata

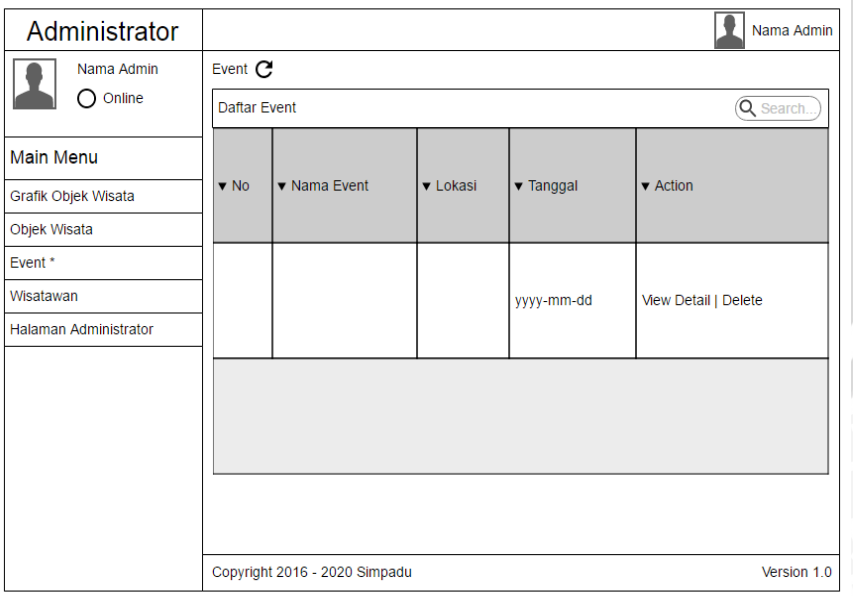
Halaman detail objek wisata merupakan halaman dari *admin*. Halaman ini menampilkan detail informasi objek wisata yang akan dilakukan verifikasi oleh *admin* dengan melihat data-data yang telah diinputkan. Berikut ini merupakan rancangan antarmuka halaman detail objek wisata pada Gambar 5.33.



Gambar 5.15 Halaman Detail Objek Wisata

5. Halaman *Event*

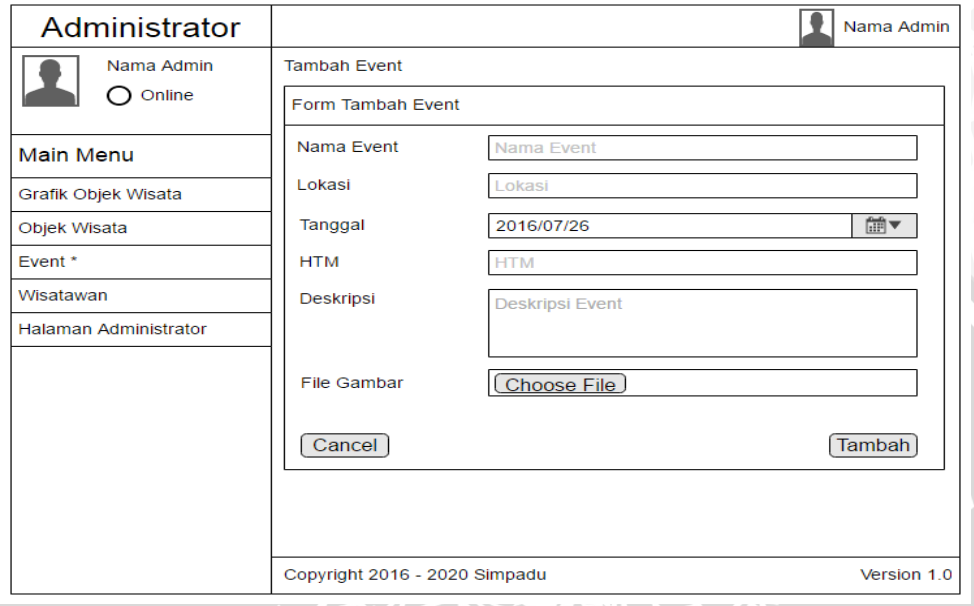
Halaman *event* merupakan halaman dari *admin*. Halaman ini menampilkan daftar *event* yang akan berlangsung dan diinputkan setiap 1 (satu) bulan sekali. Berikut ini merupakan rancangan antarmuka halaman *event* pada Gambar 5.34.



Gambar 5.16 Halaman *Event*

6. Halaman Tambah *Event*

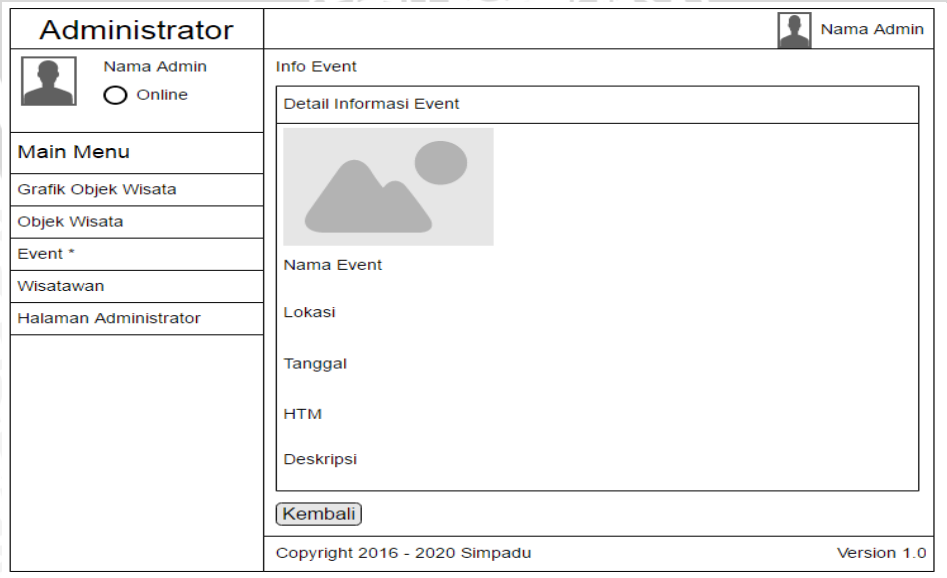
Halaman tambah *event* merupakan halaman dari *admin*. Halaman ini menampilkan halaman tambah *event* dalam bentuk *form* yang akan diisi oleh *admin* sesuai dengan data-data yang ada. Berikut ini merupakan rancangan antarmuka halaman tambah *event* pada Gambar 5.35.



Gambar 5.17 Halaman Tambah *Event*

7. Halaman Detail *Event*

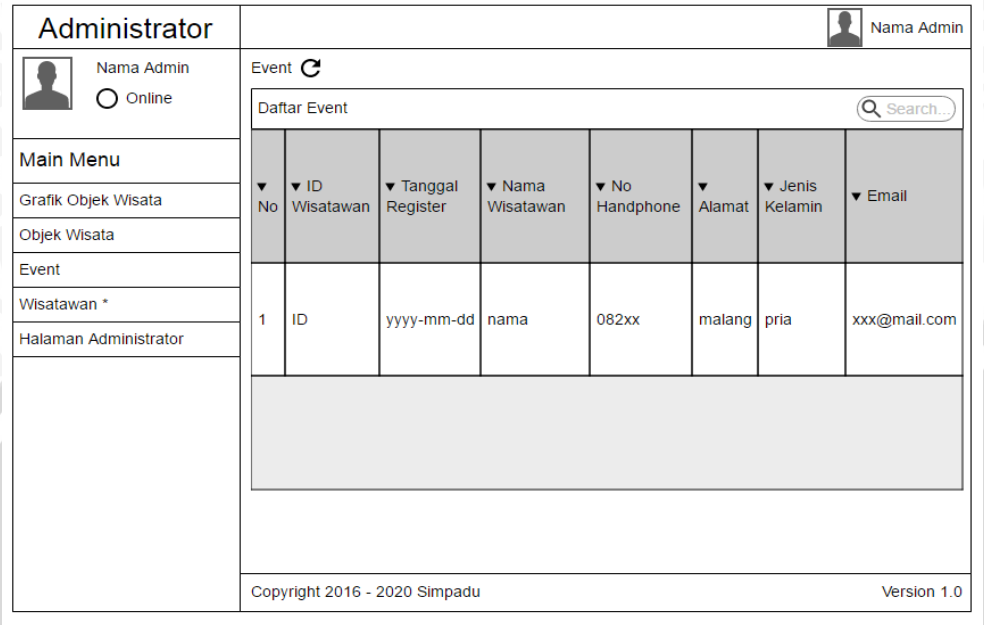
Halaman detail *event* merupakan halaman dari *admin*. Halaman ini menampilkan detail informasi *event* yang akan berlangsung. Berikut ini merupakan rancangan antarmuka halaman detail *event* pada Gambar 5.36.



Gambar 5.18 Halaman Detail *Event*

8. Halaman Wisatawan

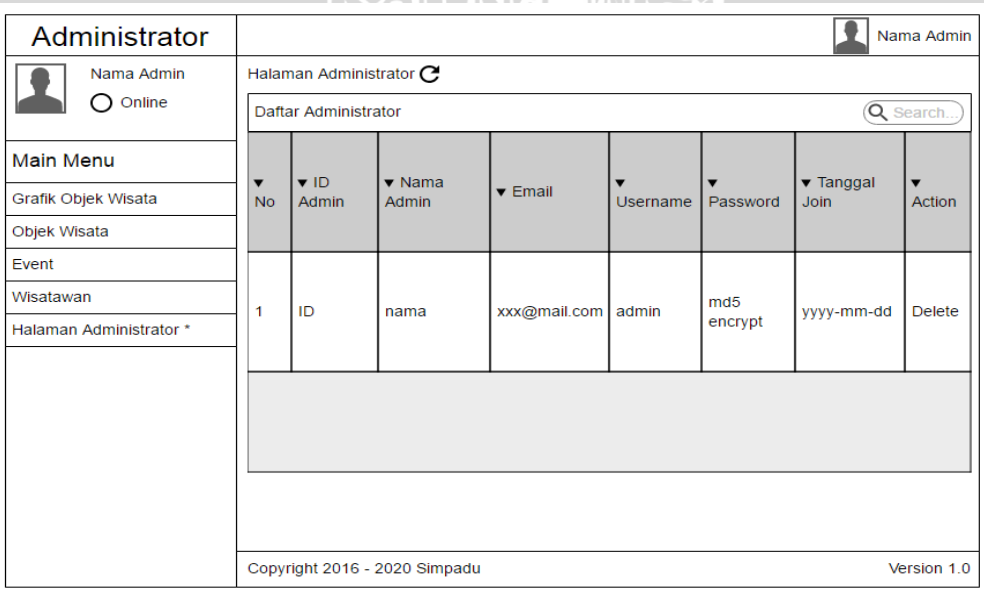
Halaman wisatawan merupakan halaman dari *admin*. Halaman ini menampilkan daftar wisatawan yang telah terdaftar pada sistem. Berikut ini merupakan rancangan antarmuka halaman wisatawan pada Gambar 5.37.



Gambar 5.19 Halaman Wisatawan

9. Halaman *Administrator*

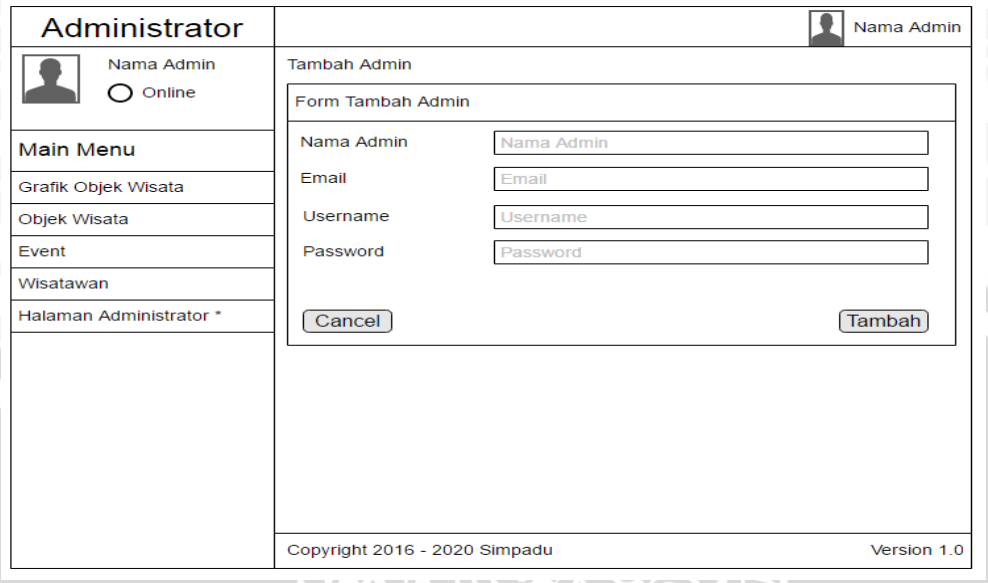
Halaman *administrator* merupakan halaman dari *admin*. Halaman ini menampilkan daftar *administrator* yang aktif. Berikut ini merupakan rancangan antarmuka halaman *administrator*pada Gambar 5.38.



Gambar 5.20 Halaman *Administrator*

10. Halaman Tambah *Administrator*

Halaman tambah *administrator* merupakan halaman dari *admin*. Halaman ini menampilkan halaman tambah *administrator* dalam bentuk *form* yang akan diisi oleh *admin* sebelumnya. Berikut ini merupakan rancangan antarmuka halaman tambah administrator pada Gambar 5.39.



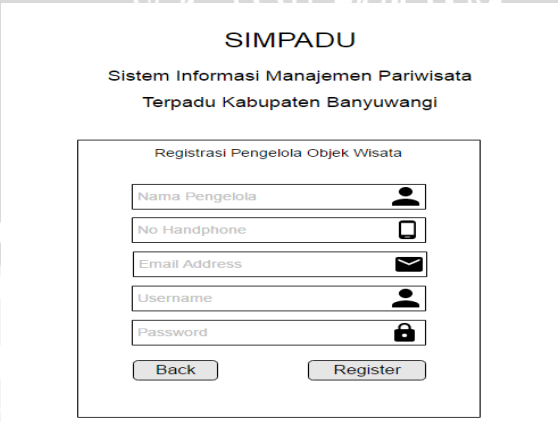
Gambar 5.21 Halaman Tambah *Administrator*

5.4.2 Perancangan Antarmuka Pengelola

Berikut merupakan antarmuka untuk Pengelola.

1. Halaman Registrasi Pengelola

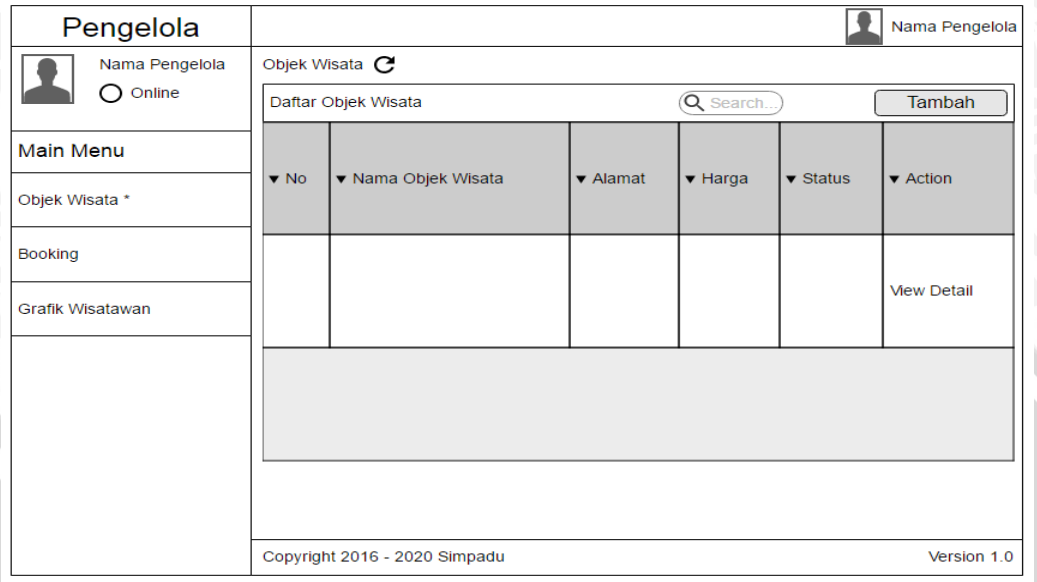
Halaman registrasi pengelola merupakan halaman *user* sebagai pengelola objek wisata. Halaman ini digunakan untuk melakukan registrasi pengelola untuk bisa melakukan aktivitas yang ada di *dashboard* pengelola pada sistem. Berikut ini merupakan rancangan antarmuka halaman registrasi pengelola pada gambar 5.40.



Gambar 5.22 Halaman Register Pengelola

2. Halaman Objek Wisata

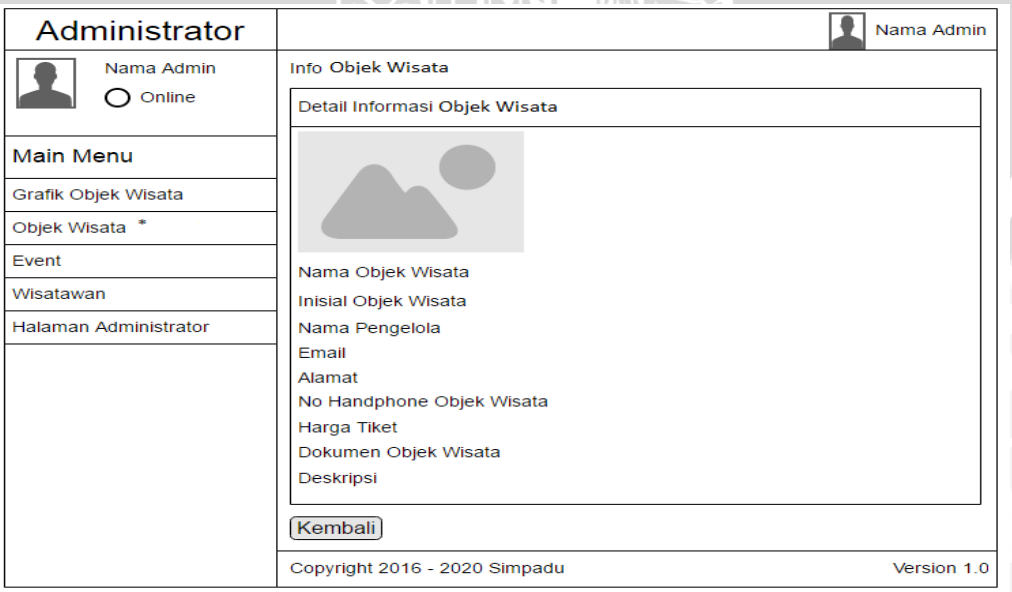
Halaman objek wisata merupakan halaman *user* sebagai pengelola objek wisata. Halaman ini menampilkan daftar objek wisata yang dikelolanya. Berikut ini merupakan rancangan antarmuka halaman objek wisata pada gambar 5.41.



Gambar 5.23 Halaman Objek Wisata

3. Halaman Detail Objek Wisata

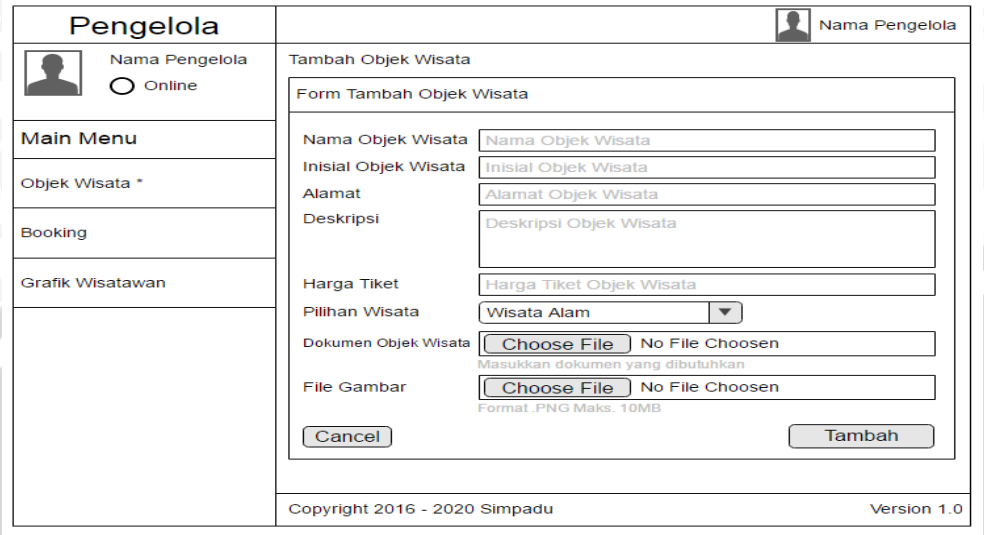
Halaman detail objek wisata merupakan halaman *user* sebagai pengelola objek wisata. Halaman ini menampilkan detail informasi objek wisata yang dikelolanya. Berikut ini merupakan rancangan antarmuka halaman detail objek wisata pada gambar 5.42.



Gambar 5.24 Halaman Detail Objek Wisata

4. Halaman Tambah Objek Wisata

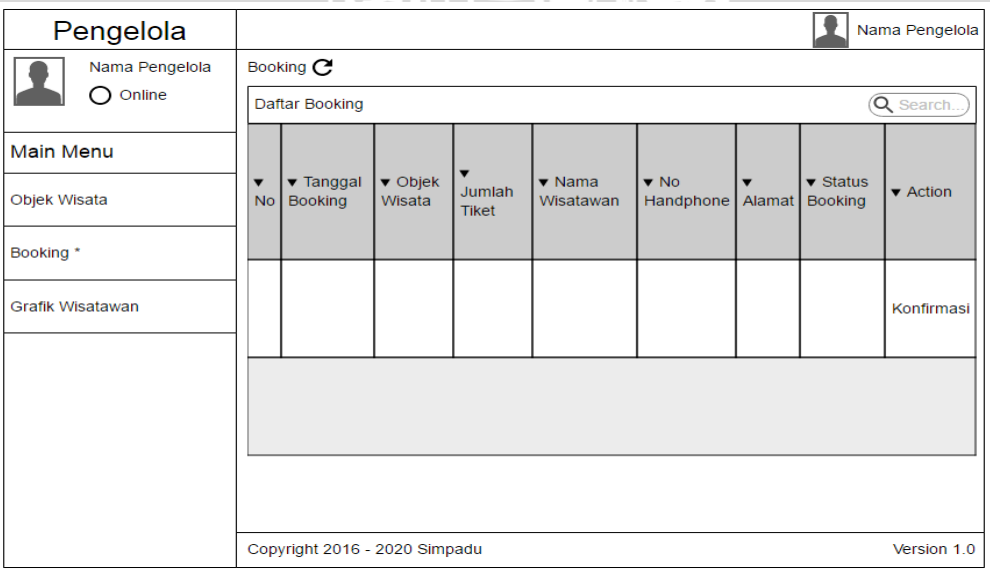
Halaman tambah objek wisata merupakan halaman *user* sebagai pengelola objek wisata. Halaman ini menampilkan halaman tambah objek wisata dalam bentuk *form* yang diisi sesuai dengan data-data objek wisata yang dikelolanya. Berikut ini merupakan rancangan antarmuka halaman tambah objek wisata pada gambar 5.43.



Gambar 5.25 Halaman Tambah Objek Wisata

5. Halaman Daftar *Booking* Objek Wisata

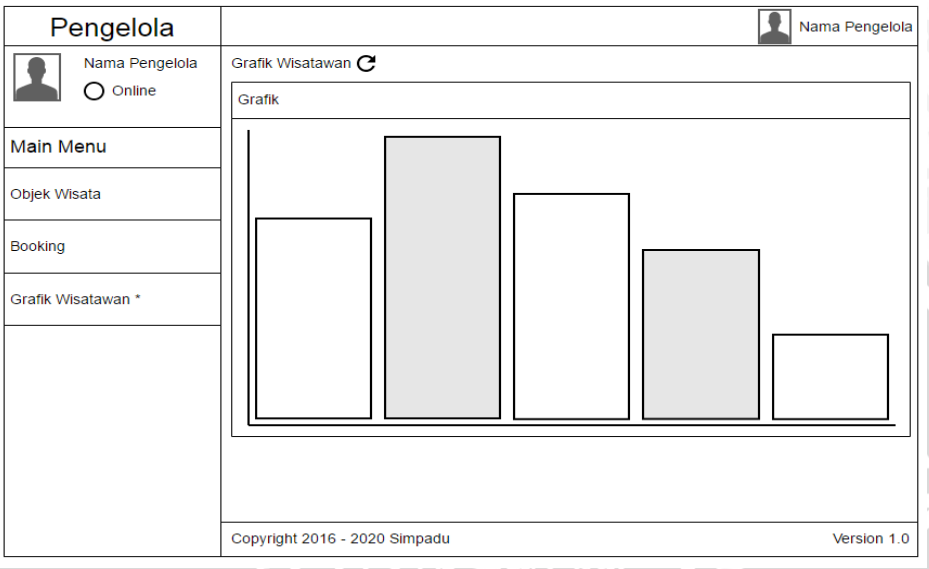
Halaman daftar *booking* objek wisata merupakan halaman *user* sebagai pengelola objek wisata. Halaman ini menampilkan daftar *booking* objek wisata yang masuk. Berikut ini merupakan rancangan antarmuka halaman daftar *booking* objek wisata pada gambar 5.44.



Gambar 5.26 Halaman Daftar *Booking*

6. Halaman Grafik Wisatawan

Halaman grafik wisatawan merupakan halaman *user* sebagai pengelola objek wisata. Halaman ini menampilkan jumlah wisatawan yang berkunjung ke objek wisata yang dikelolanya dalam bentuk grafik. Berikut ini merupakan rancangan antarmuka halaman grafik wisatawan pada gambar 5.45.



Gambar 5.27 Halaman Grafik Wisatawan

5.4.3 Perancangan Antarmuka Wisatawan

Berikut merupakan antarmuka untuk Wisatawan.

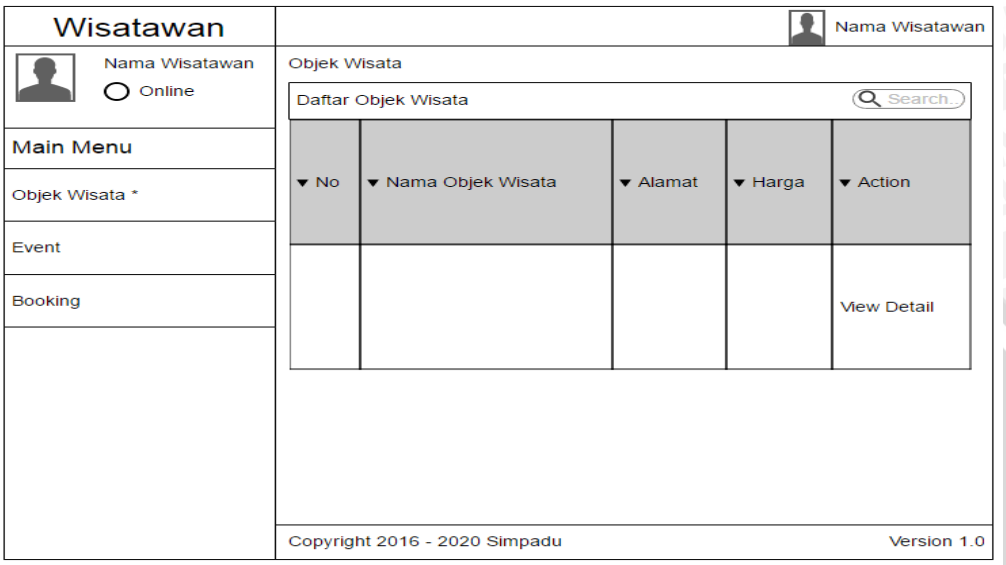
1. Halaman Registrasi Wisatawan

Halaman registrasi wisatawan merupakan halaman *user* sebagai pengguna umum. Halaman ini menampilkan *form* registrasi sebagai wisatawan. Berikut ini merupakan rancangan antarmuka halaman registrasi wisatawan pada gambar 5.46.

Gambar 5.28 Halaman Register Wisatawan

2. Halaman Objek Wisata

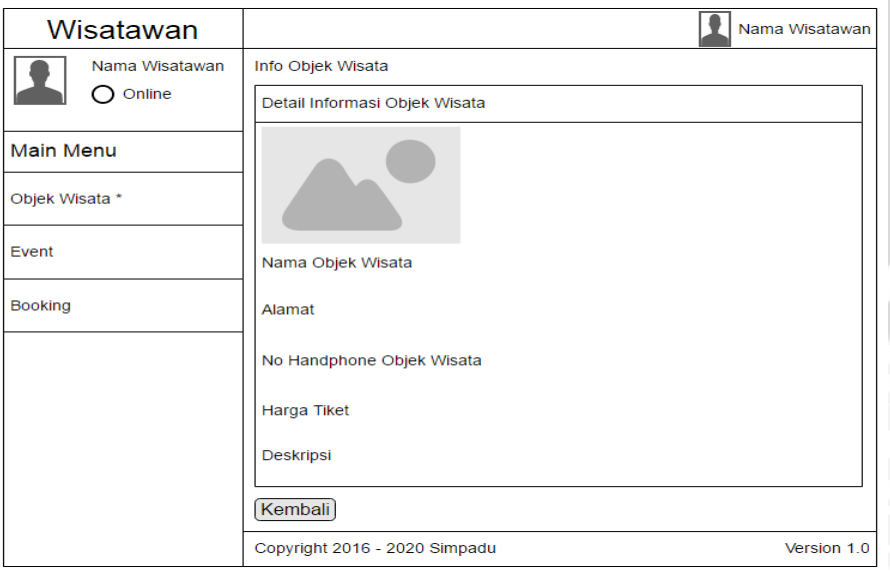
Halaman objek wisata merupakan halaman *user* sebagai wisatawan. Halaman ini menampilkan daftar objek wisata yang sudah terdaftar dan dapat dikunjungi oleh wisatawan. Berikut ini merupakan rancangan antarmuka halaman objek wisata pada gambar 5.47.



Gambar 5.29 Halaman Objek Wisata

3. Halaman Detail Objek Wisata

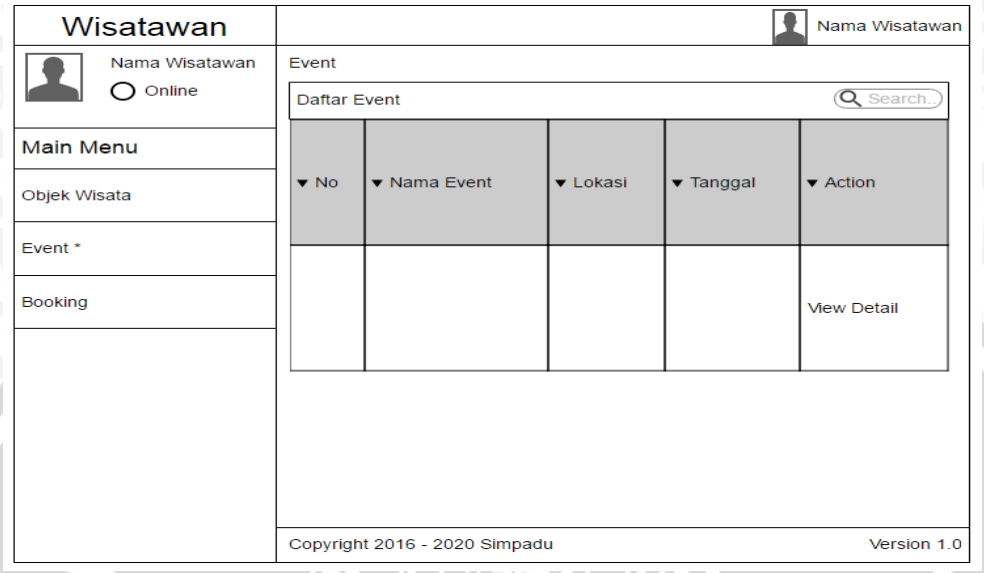
Halaman detail objek wisata merupakan halaman *user* sebagai wisatawan. Halaman ini menampilkan detail informasi objek wisata yang ingin diketahui dan dikunjungi. Berikut ini merupakan rancangan antarmuka halaman detail objek wisata pada gambar 5.48.



Gambar 5.30 Halaman Detail Objek Wisata

4. Halaman Event

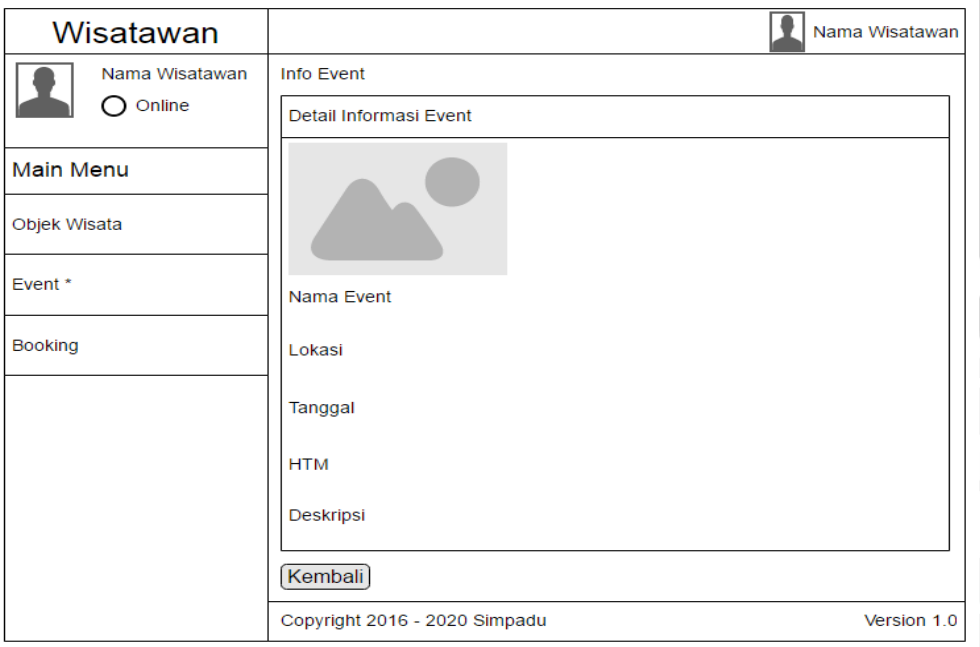
Halaman *event* merupakan halaman *user* sebagai wisatawan. Halaman ini menampilkan daftar *event* yang akan berlangsung setiap bulannya. Berikut ini merupakan rancangan antarmuka halaman *event* pada gambar 5.49.



Gambar 5.31 Halaman *Event*

5. Halaman Detail *Event*

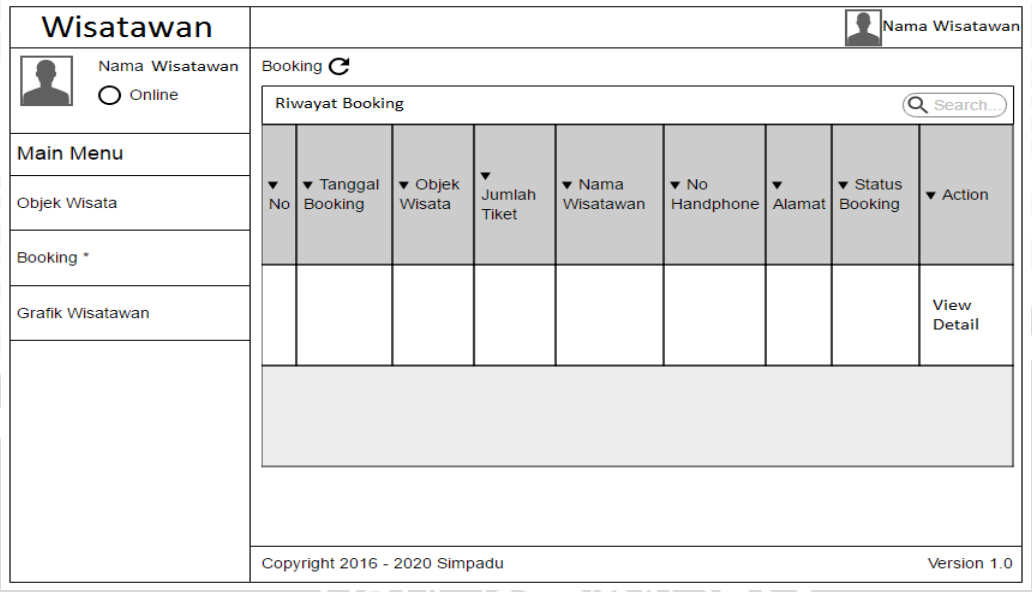
Halaman detail *event* merupakan halaman *user* sebagai wisatawan. Halaman ini menampilkan detail informasi *event* yang akan berlangsung. Berikut ini merupakan rancangan antarmuka halaman detail *event* pada gambar 5.50.



Gambar 5.32 Halaman Detail *Event*

6. Halaman Riwayat *Booking* Objek Wisata

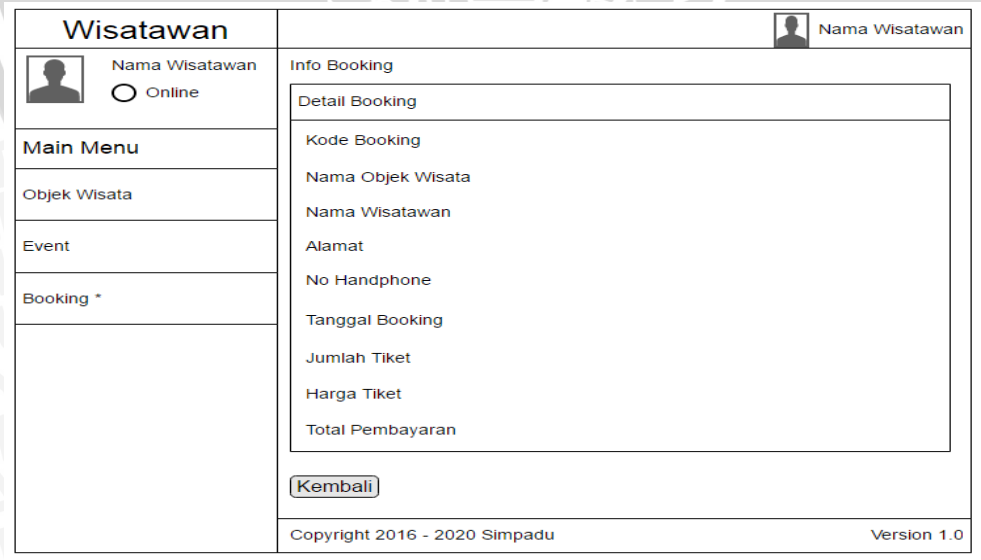
Halaman riwayat *booking* objek wisata merupakan halaman *user* sebagai wisatawan. Halaman ini menampilkan riwayat *booking* objek wisata yang pernah dilakukan oleh wisatawan. Berikut ini merupakan rancangan antarmuka halaman riwayat *booking* objek wisata pada gambar 5.51.



Gambar 5.33 Halaman Riwayat *Booking* Objek Wisata

7. Halaman Detail *Booking* Objek Wisata

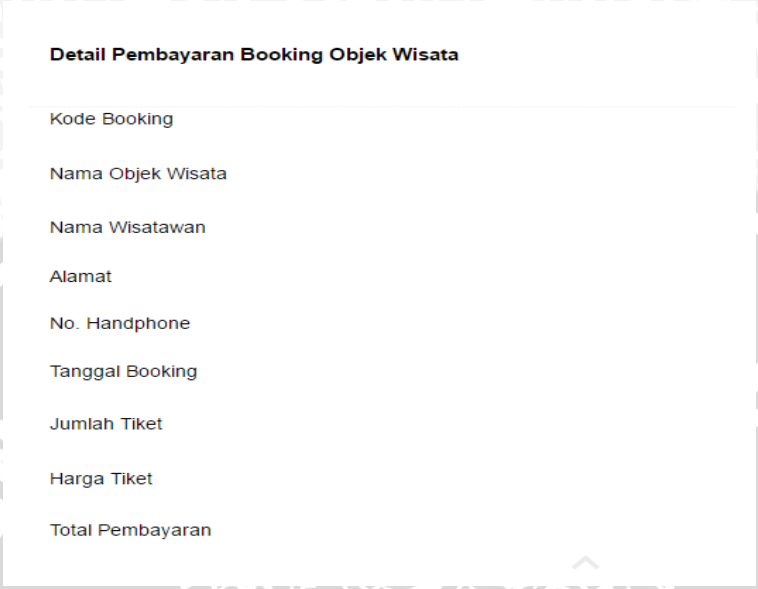
Halaman detail *booking* objek wisata merupakan halaman *user* sebagai wisatawan. Halaman ini menampilkan detail informasi *booking* objek wisata yang telah dilakukan sebelumnya. Berikut ini merupakan rancangan antarmuka halaman detail *booking* objek wisata pada gambar 5.52.



Gambar 5.34 Halaman Detail *Booking* Objek Wisata

8. Halaman Cetak Laporan *Booking* Objek wisata

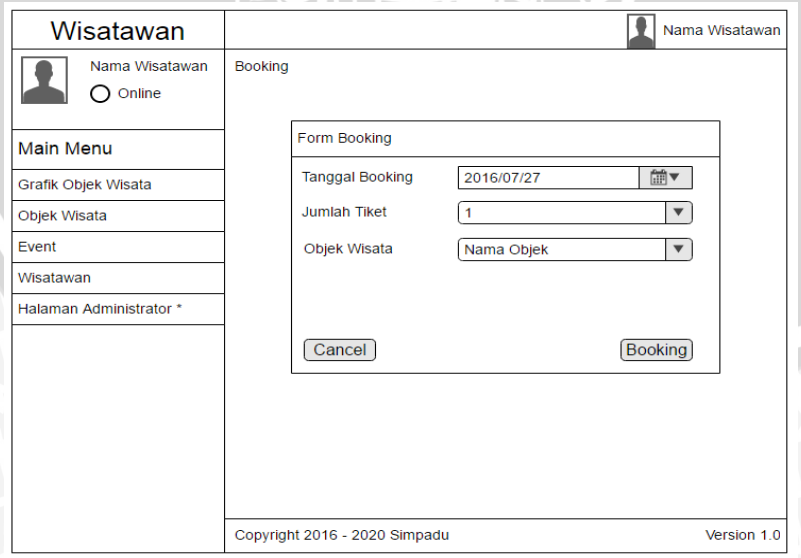
Halaman cetak laporan *booking* objek wisata merupakan halaman *user* sebagai wisatawan. Halaman ini menampilkan detail cetak dan melakukan cetak laporan *booking* objek wisata. Berikut ini merupakan rancangan cetak laporan *booking* objek wisata pada gambar 5.53.



Gambar 5.35 Halaman Detail *Booking* Objek Wisata

9. Halaman *Booking* Objek Wisata

Halaman *booking* objek wisata merupakan halaman *user* sebagai wisatawan. Halaman ini menampilkan *form booking* objek wisata yang akan diisi oleh wisatawan. Berikut ini merupakan rancangan antarmuka *booking* objek wisata pada gambar 5.54.



Gambar 5.36 Halaman *Booking* Objek Wisata

BAB 6 IMPLEMENTASI

Implementasi sistem informasi manajemen pariwisata terpadu merupakan tahap penerapan dari hasil proses analisis kebutuhan dan perancangan sistem yang telah dilakukan. Pembahasan pada bab implementasi meliputi spesifikasi perangkat lunak dan perangkat keras, batasan implementasi, implementasi antarmuka sistem.

6.1 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam implementasi aplikasi sistem informasi manajemen pariwisata terpadu Kabupaten Banyuwangi, sebagai berikut:

Tabel 6.1 Tabel Spesifikasi Perangkat Lunak

No	Komponen	Spesifikasi
1.	*Operating System*	*Windows 8.1 64-bit.*
2.	*Web Server and RDBMS*	*Xampp versi v3.2.2*
3.	*Application*	*Sublime Text 3*
4.	*Browser*	*Google Chrome versi 51.0*
5.	*Framework*	*Framework CodeIgniter (CI) versi 3.0.6*

6.2 Spesifikasi Perangkat Keras

pesifikasi perangkat keras yang digunakan dalam implementasi aplikasi sistem informasi manajemen pariwisata terpadu Kabupaten Banyuwangi, sebagai berikut:

Tabel 6.2 Tabel Spesifikasi Perangkat Keras

No	Komponen	Spesifikasi
1.	*Type*	*ASUS VivoBook*
2.	*Processor*	*Intel Core i3-3217u CPU @1.80GHz*
3.	*Memory (RAM)*	*4.00 GB*
4.	*Mouse*	*Microsoft Pack Model BT-792W*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6.3 Batasan Implementasi

Batasan dalam implementasi sistem informasi manajemen pariwisata terpadu adalah sebagai berikut :

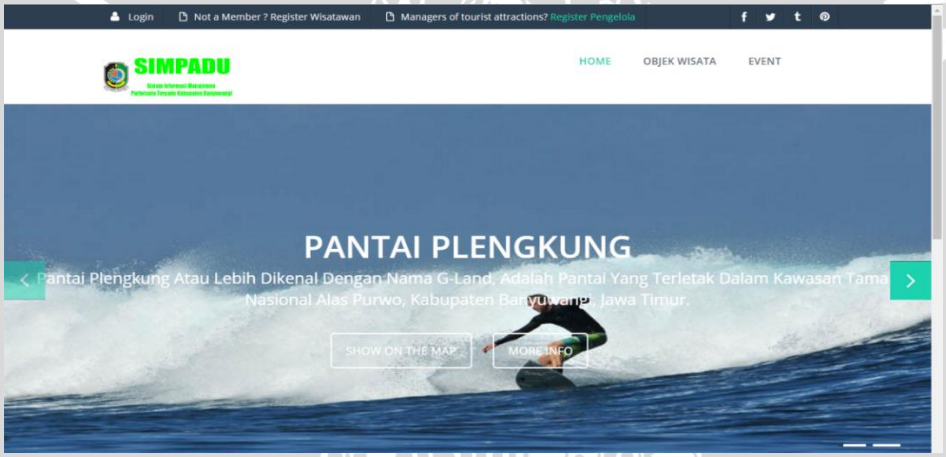
a. *Web server* yang digunakan yaitu *apache* pada *Xampp v3.2.2*.
b. *Database Management System* menggunakan *MySQL* dan *phpMyAdmin pada Xampp v3.2.2*.
c. *Framework* yang digunakan sebagai implementasi sistem yaitu *framework CodeIgniter* versi 3.0.6.

6.4 Implementasi Antarmuka

Implementasi antarmuka sistem informasi manajemen pariwisata terpadu yang berdasarkan perancangan antarmuka. Berikut merupakan implementasi antarmuka sistem informasi manajemen pariwisata terpadu.

a. Halaman *Home*

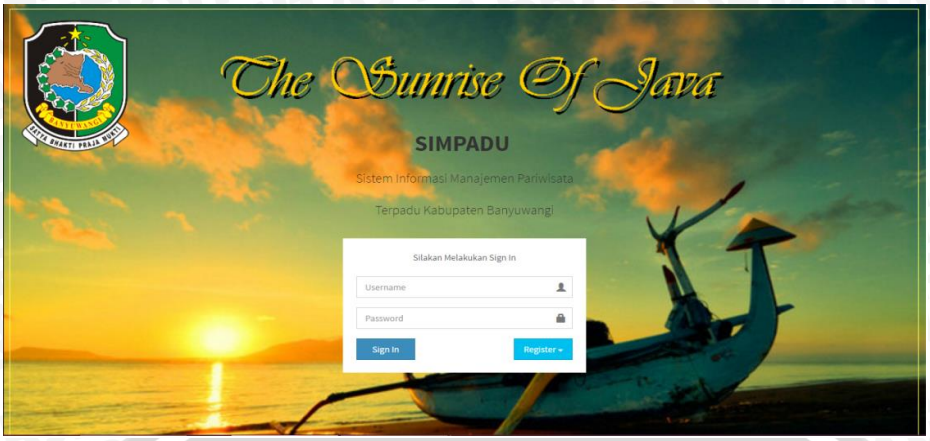
Gambar 6.1 menggambarkan hasil implementasi dari halaman login pada sistem informasi pariwisata terpadu Kabupaten Banyuwangi. Halaman ini digunakan untuk melakukan login ke sistem.



Gambar 6.1 Halaman *Home*

b. Halaman *Login*

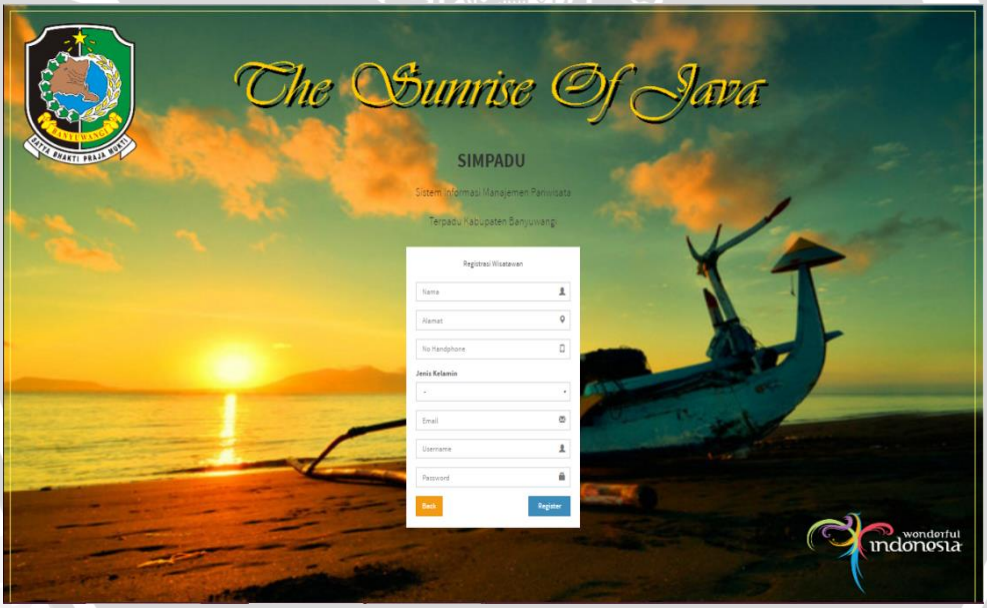
Gambar 6.2 menggambarkan hasil implementasi dari halaman *login* pada sistem informasi pariwisata terpadu Kabupaten Banyuwangi. Halaman ini digunakan untuk melakukan *login* ke sistem.



Gambar 6.2 Halaman *Login*

c. Halaman Register Wisatawan

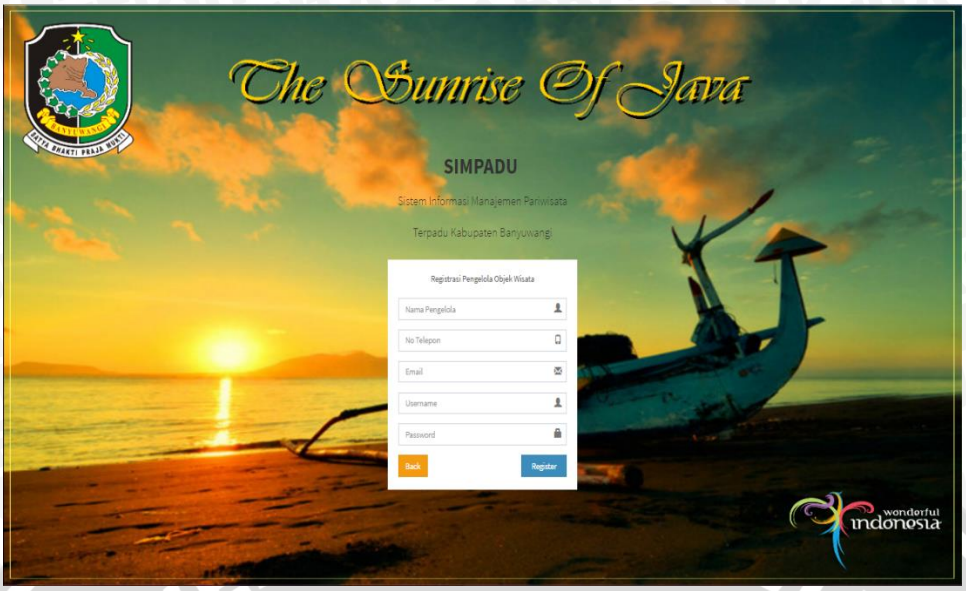
Gambar 6.3 menggambarkan hasil implementasi dari tampilan halaman register wisatawan pada sistem informasi manajemen pariwisata terpadu Kabupaten Banyuwangi. Halaman ini digunakan oleh pengguna umum untuk melakukan registrasi sebagai wisatawan.



Gambar 6.3 Halaman Registrasi Wisatawan

d. Halaman Register Pengelola

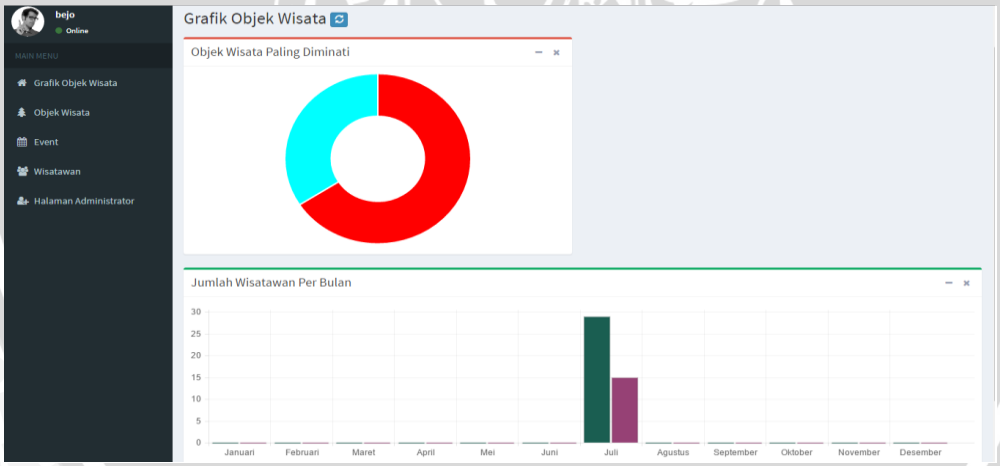
Gambar 6.4 menggambarkan hasil implementasi dari tampilan halaman register pengelola objek wisata pada sistem informasi manajemen pariwisata terpadu Kabupaten Banyuwangi. Halaman ini digunakan untuk melakukan registrasi pengelola yang mengelola objek wisata tertentu.



Gambar 6.4 Halaman Registrasi Pengelola

e. Halaman Grafik Objek Wisata

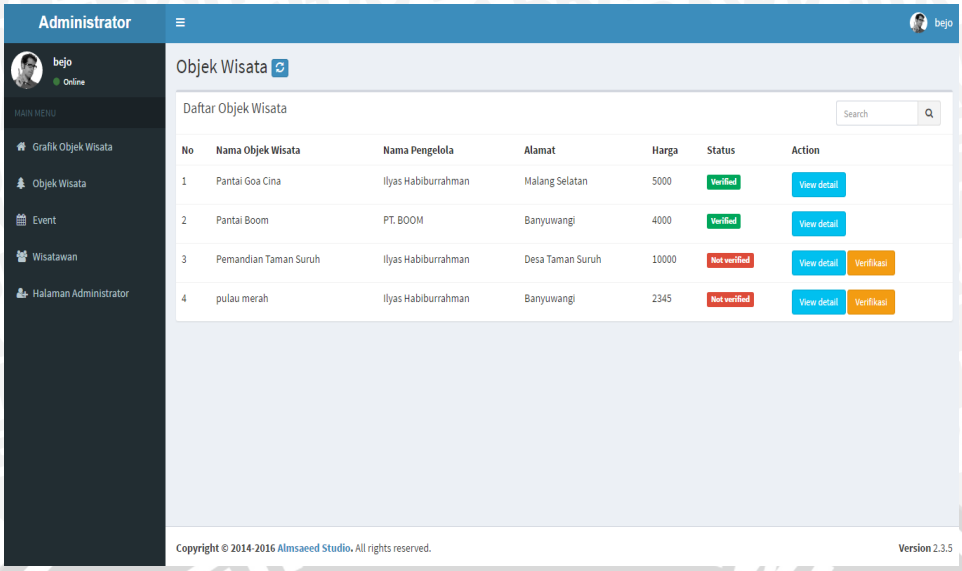
Gambar 6.5 menggambarkan hasil implementasi dari tampilan halaman grafik objek wisata dimana yang dapat mengakses halaman tersebut hanya *admin*, digunakan untuk melihat jumlah wisatawan setiap bulannya pada masing-masing objek wisata dan melihat objek wisata.



Gambar 6.5 Halaman Grafik Objek Wisata

f. Halaman Objek Wisata (*Admin*)

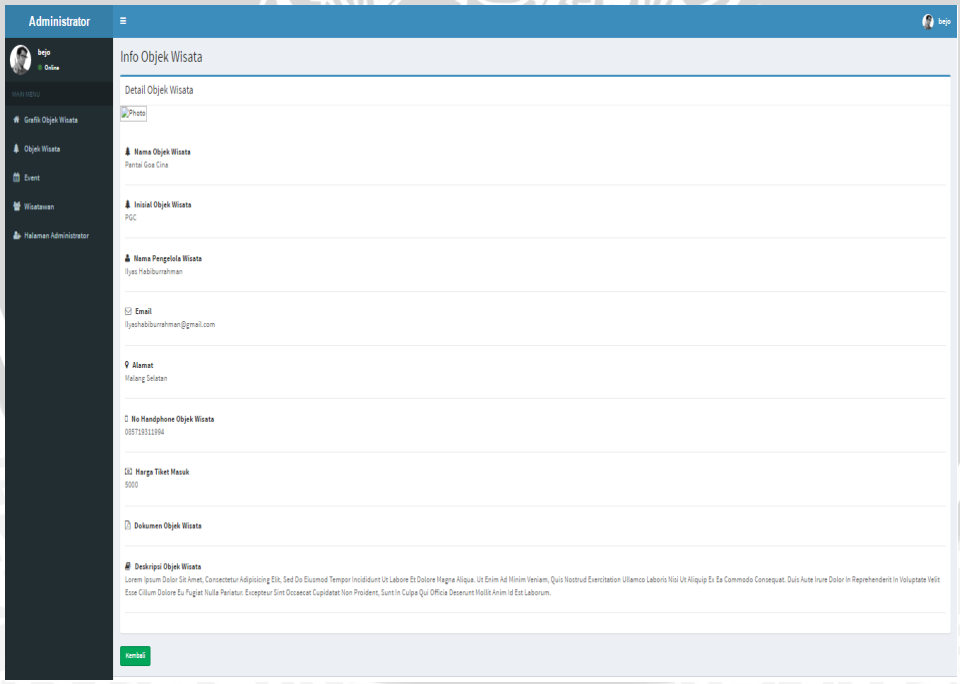
Gambar 6.6 menggambarkan hasil implementasi dari halaman objek wisata yang hanya bisa diakses oleh *admin*. Halaman ini digunakan oleh *admin* untuk melihat daftar objek wisata dan melakukan verifikasi terhadap objek wisata dengan melihat terlebih detail informasi objek wisata yang telah didaftarkan pada sistem.



Gambar 6.6 Halaman Objek Wisata (*Admin*)

g. Halaman Detail Objek Wisata (*Admin*)

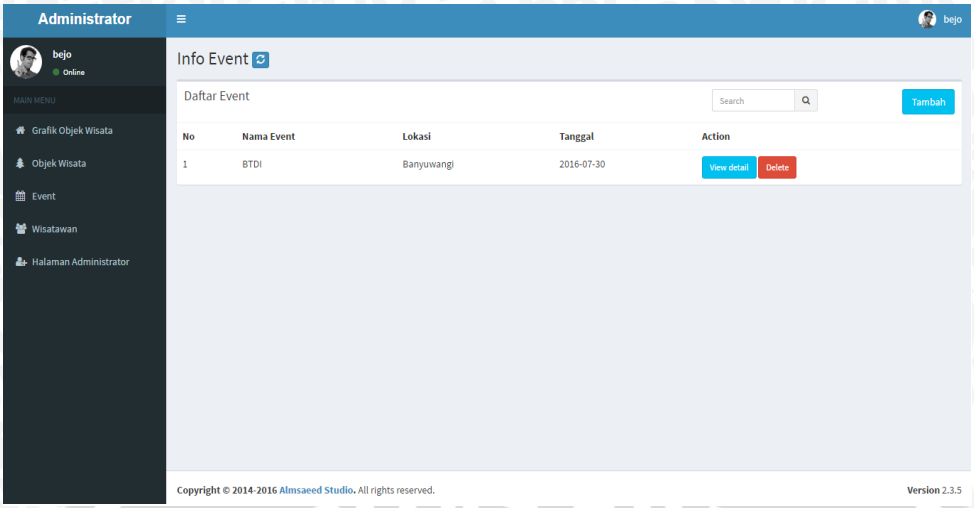
Gambar 6.7 menggambarkan hasil implementasi dari halaman objek wisata yang hanya bisa diakses oleh *admin*. Halaman ini digunakan oleh *admin* untuk melihat detail informasi objek wisata sebelum *admin* melakukan verifikasi terhadap objek wisata dengan menekan tombol "verifikasi" pada sistem.



Gambar 6.7 Halaman Detail Objek Wisata (*Admin*)

h. Halaman *Event* (*Admin*)

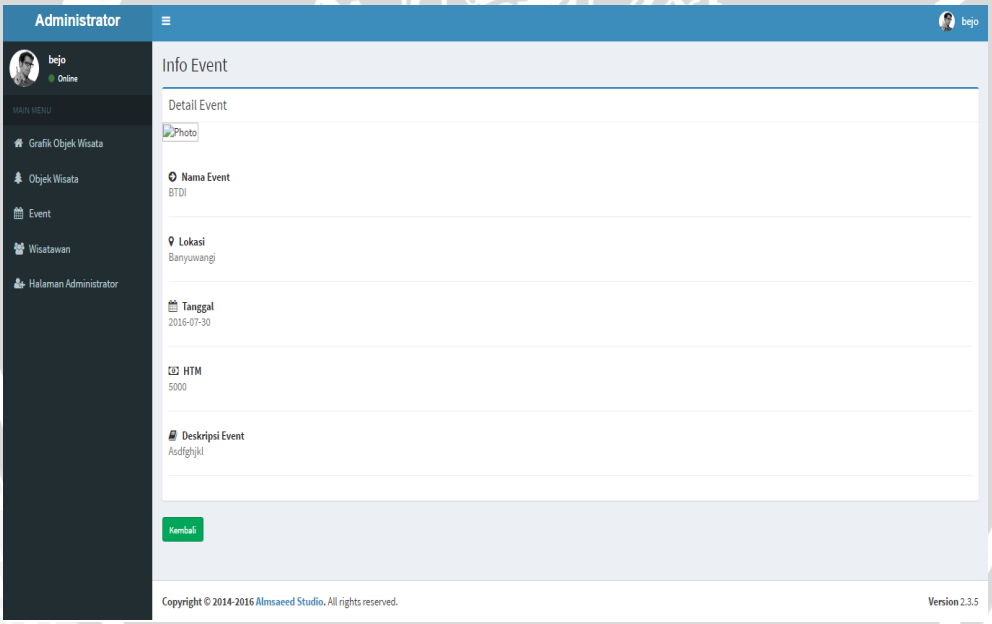
Gambar 6.8 menggambarkan hasil implementasi dari halaman *event* pada *dashboard admin*. Halamain ini digunakan untuk melihat daftar *event* dan menghapus *event* yang sudah selesai berlangsung.



Gambar 6.8 Halaman *Event* (*Admin*)

i. Halaman Detail *Event* (*Admin*)

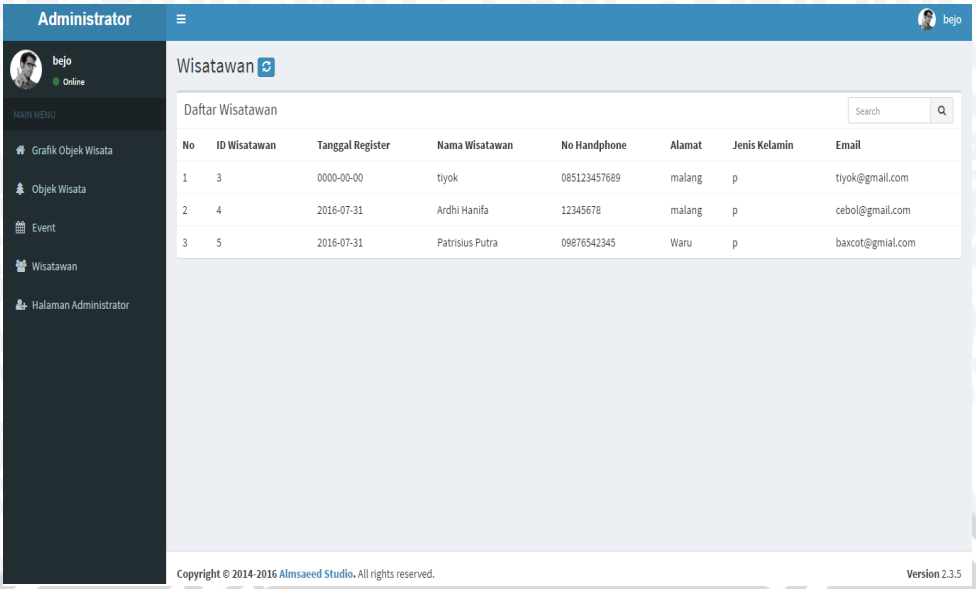
Gambar 6.9 menggambarkan hasil implementasi dari halaman detail *event* pada *dashboard admin*. Halaman ini digunakan untuk melihat detail informasi *event* yang akan berlangsung.



Gambar 6.9 Halaman Detail *Event* (*Admin*)

j. Halaman Wisatawan

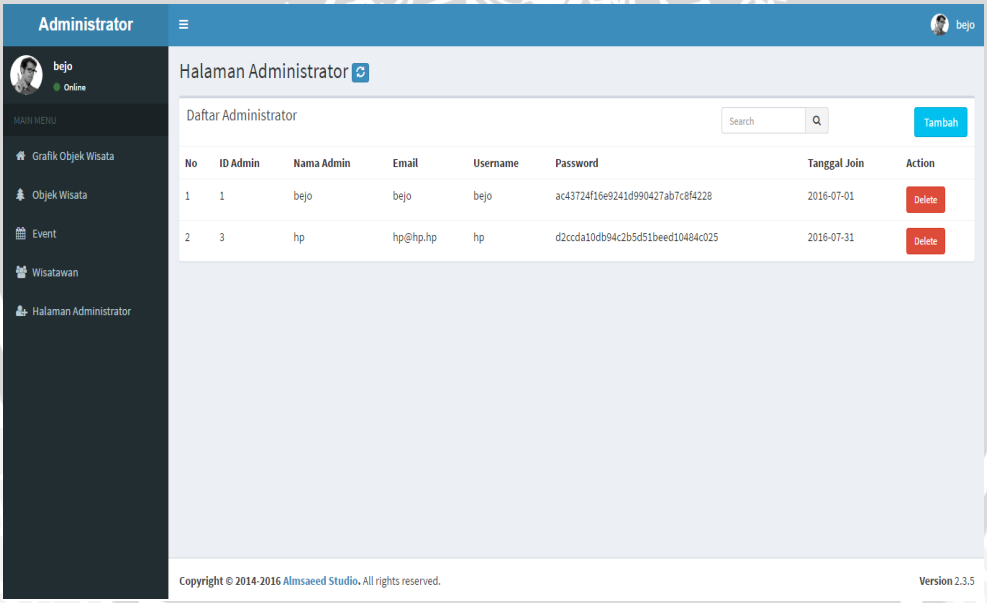
Gambar 6.10 menggambarkan hasil implementasi dari halaman wisatawan pada *dashboard admin*. Halaman ini digunakan untuk melihat daftar wisatawan yang terdaftar pada sistem.



Gambar 6.10 Halaman Wisatawan

k. Halaman *Administrator*

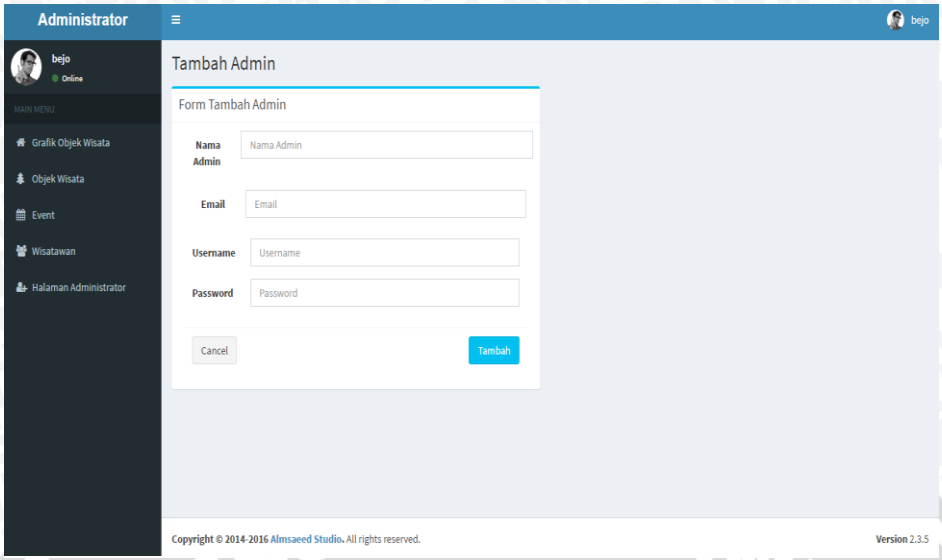
Gambar 6.11 menggambarkan hasil implementasi dari halaman *administrator* pada *dashboard admin*. Halaman ini digunakan untuk melihat daftar *administrator* yang terdaftar dan aktif pada sistem.



Gambar 6.11 Halaman *Administrator*

l. Halaman Tambah *Administrator*

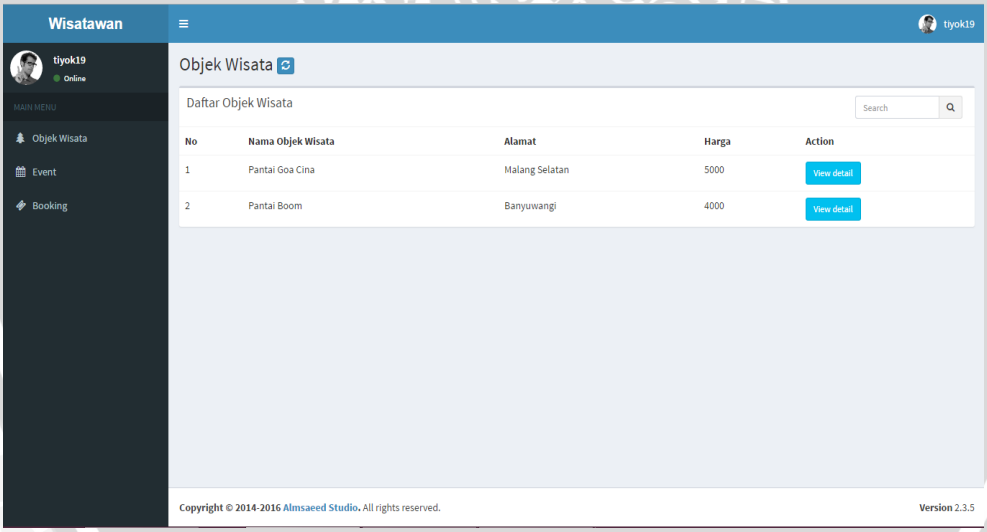
Gambar 6.12 menggambarkan hasil implementasi dari halaman tambah *administrator* pada *dashboard admin*. Halaman ini digunakan untuk menambahkan *administrator* baru sesuai dengan kebutuhan pada sistem.



Gambar 6.12 Halaman Tambah *Administrator*

m. Halaman Objek Wisata (Wisatawan)

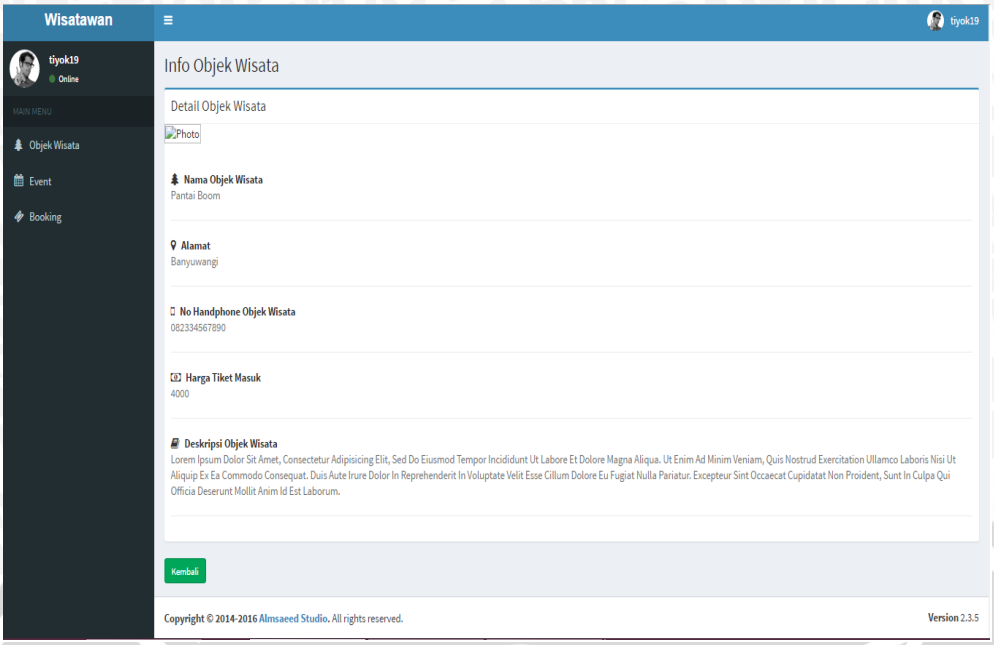
Gambar 6.13 menggambarkan hasil implementasi dari halaman objek wisata pada *dashboard* wisatawan. Halaman ini digunakan untuk melihat daftar objek wisata yang terdaftar pada sistem.



Gambar 6.13 Halaman Objek Wisata (Wisatawan)

n. Halaman Detail Objek Wisata (Wisatawan)

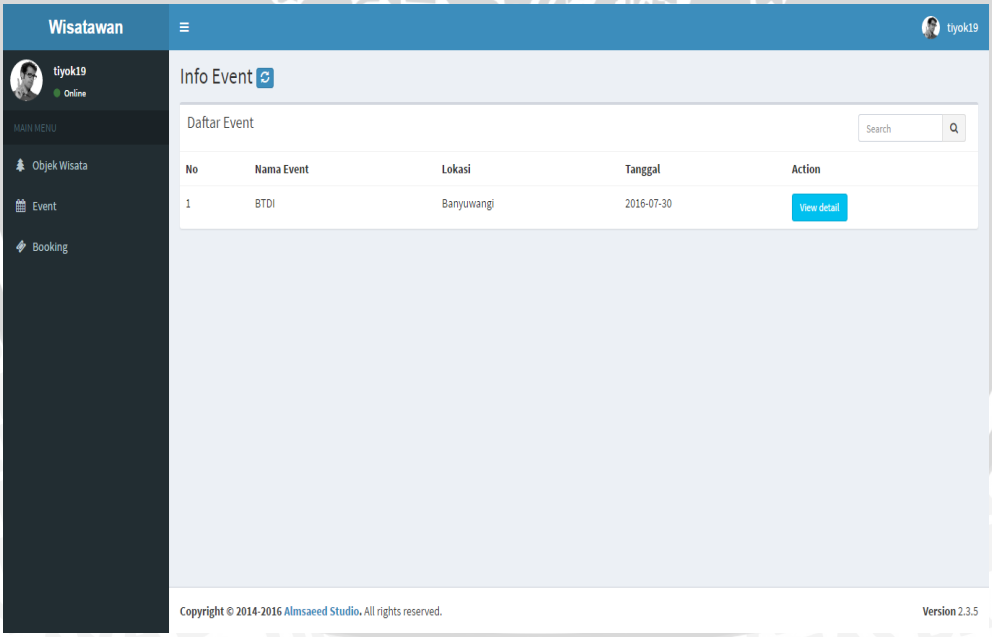
Gambar 6.14 menggambarkan hasil implementasi dari halaman detail objek wisata pada *dashboard* wisatawan. Halaman ini digunakan untuk melihat detail informasi objek wisata yang terdaftar pada sistem.



Gambar 6.14 Halaman Detail Objek Wisata (Wisatawan)

o. Halaman *Event* (Wisatawan)

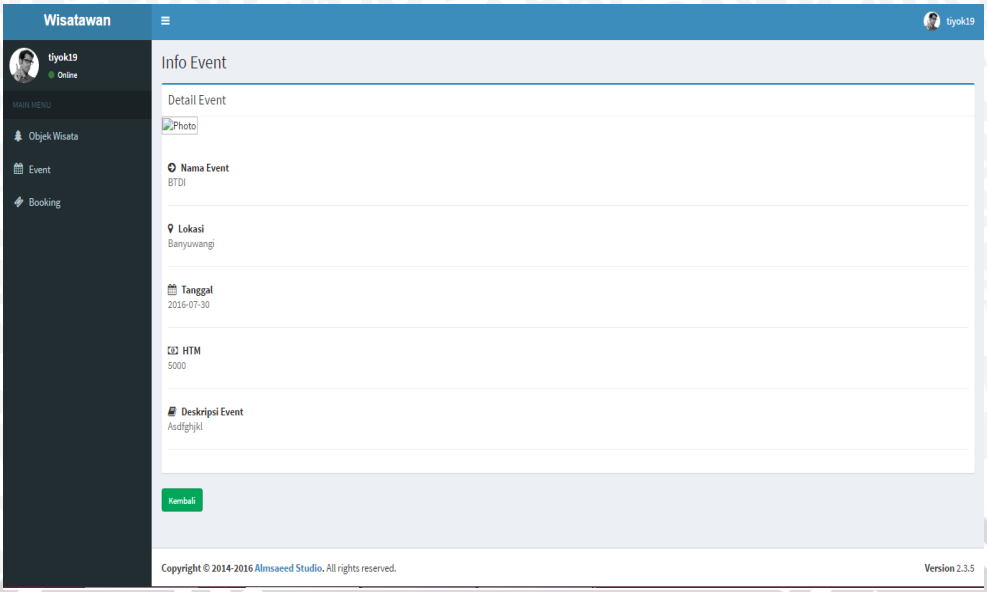
Gambar 6.15 menggambarkan hasil implementasi dari halaman *event* pada *dashboard* wisatawan. Halaman ini digunakan untuk melihat daftar *event* yang akan berlangsung pada sistem.



Gambar 6.15 Halaman *Event* (Wisatawan)

p. Halaman Detail *Event* (Wisatawan)

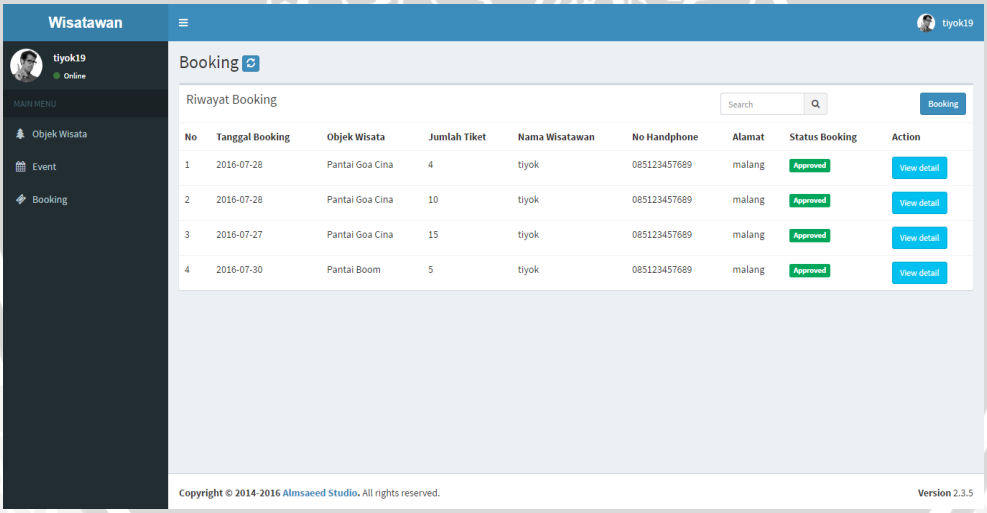
Gambar 6.16 menggambarkan hasil implementasi dari halaman detail *event* pada *dashboard* wisatawan. Halaman ini digunakan untuk melihat detail informasi *event* yang akan berlangsung pada sistem.



Gambar 6.16 Halaman Detail *Event* (Wisatawan)

q. Halaman Riwayat *Booking* Objek Wisata

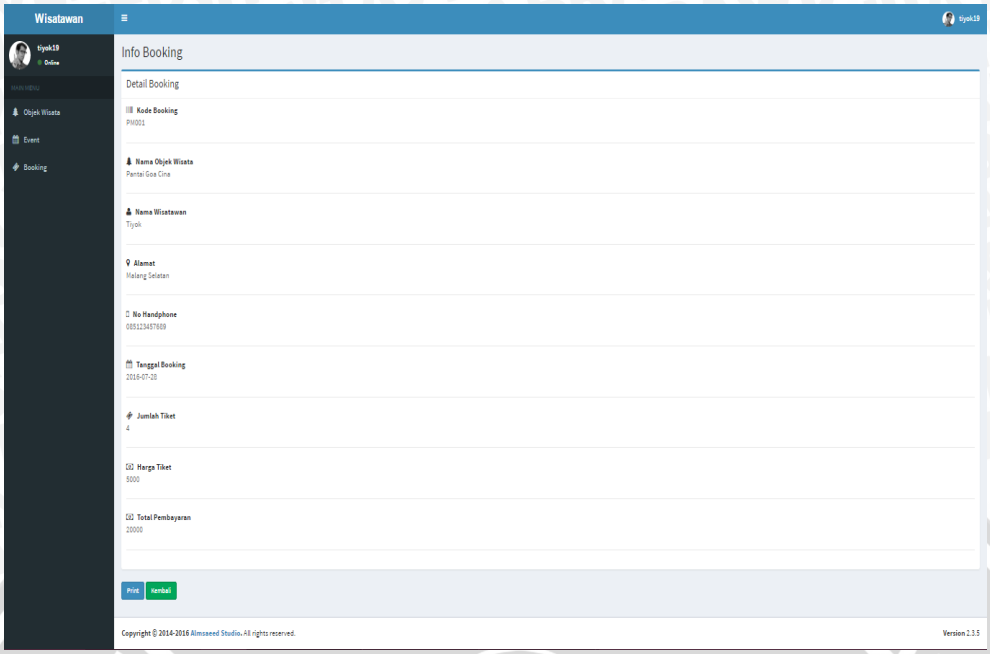
Gambar 6.17 menggambarkan hasil implementasi dari halaman riwayat *booking* objek wisata pada *dashboard* wisatawan. Halaman ini digunakan untuk melihat riwayat *booking* objek wisata yang pernah dilakukan sebelumnya dan tersimpan pada sistem.



Gambar 6.17 Halaman Riwayat *Booking* Objek Wisata

r. Halaman Detail *Booking* Objek Wisata

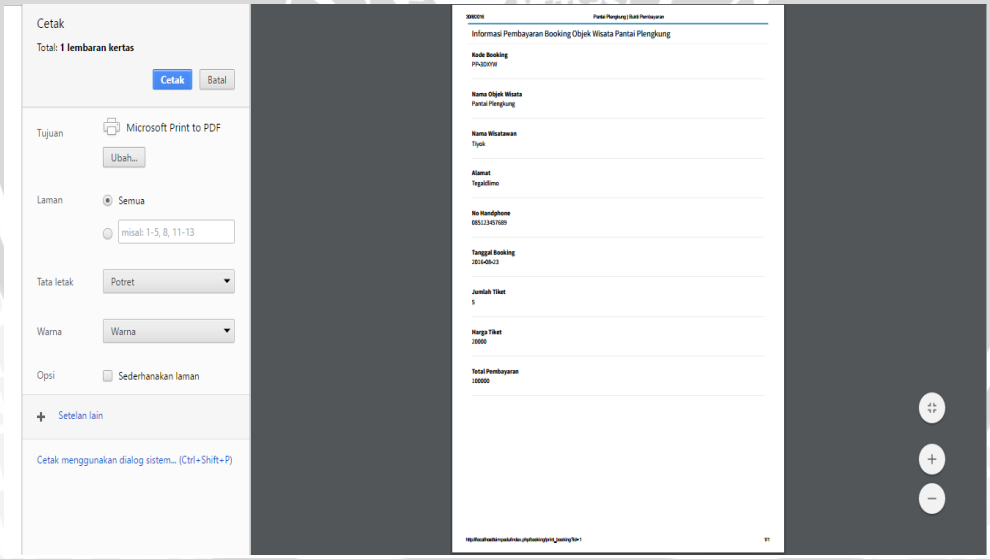
Gambar 6.18 menggambarkan hasil implementasi dari halaman detail *booking* objek wisata pada *dashboard* wisatawan. Halaman ini digunakan untuk melihat detail *booking* objek wisata yang pernah dilakukan sebelumnya dan untuk melakukan cetak laporan *booking* objek wisata pada sistem.



Gambar 6.18 Halaman Detail *Booking* Objek Wisata

s. Halaman Cetak Laporan *Booking* Objek Wisata

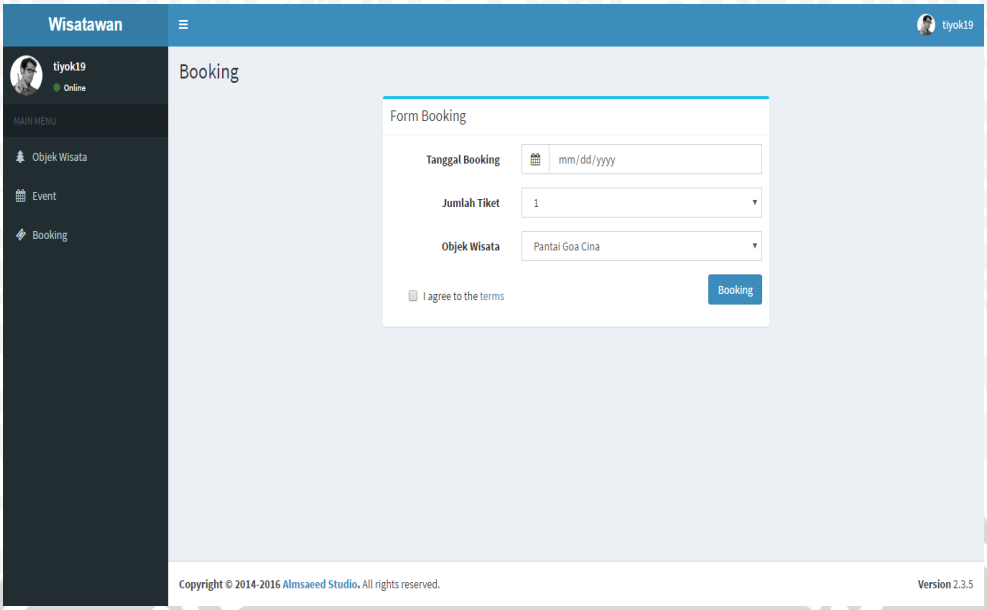
Gambar 6.19 menggambarkan hasil implementasi dari halaman cetak laporan *booking* objek wisata pada *dashboard* wisatawan. Halaman ini digunakan untuk melihat detail *booking* objek wisata dan melakukan cetak laporan *booking* objek wisata pada sistem.



Gambar 6.19 Halaman Cetak Laporan *Booking* Objek Wisata

t. Halaman *Booking* Objek Wisata

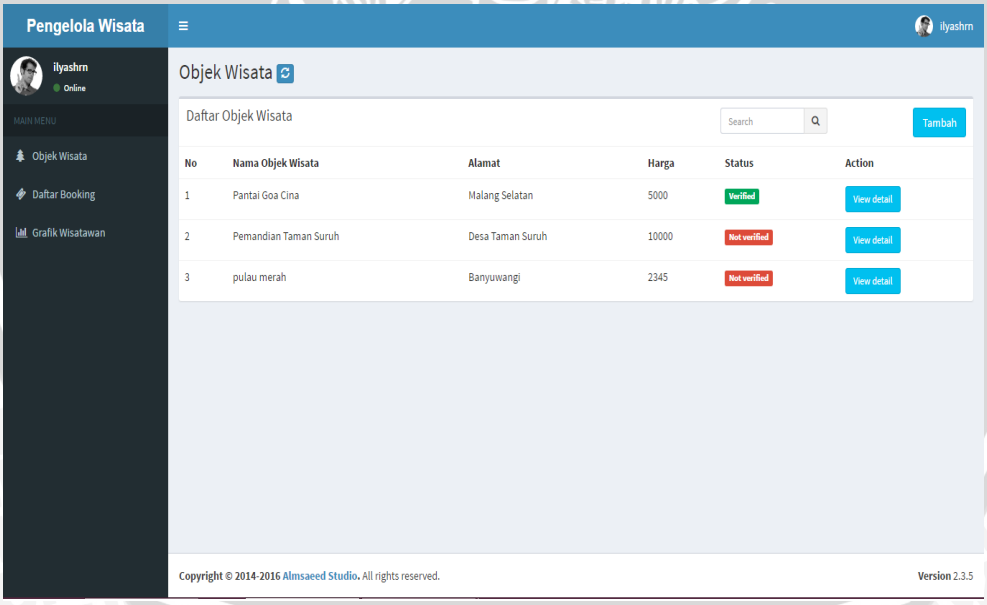
Gambar 6.20 menggambarkan hasil implementasi dari halaman *booking* objek wisata pada *dashboard* wisatawan. Halaman ini digunakan untuk melakukan *booking* objek wisata yang akan dikunjungi dengan mengisi *form booking* yang sudah tersedia pada sistem.



Gambar 6.20 Halaman *Booking* Objek Wisata

u. Halaman Objek Wisata (Pengelola)

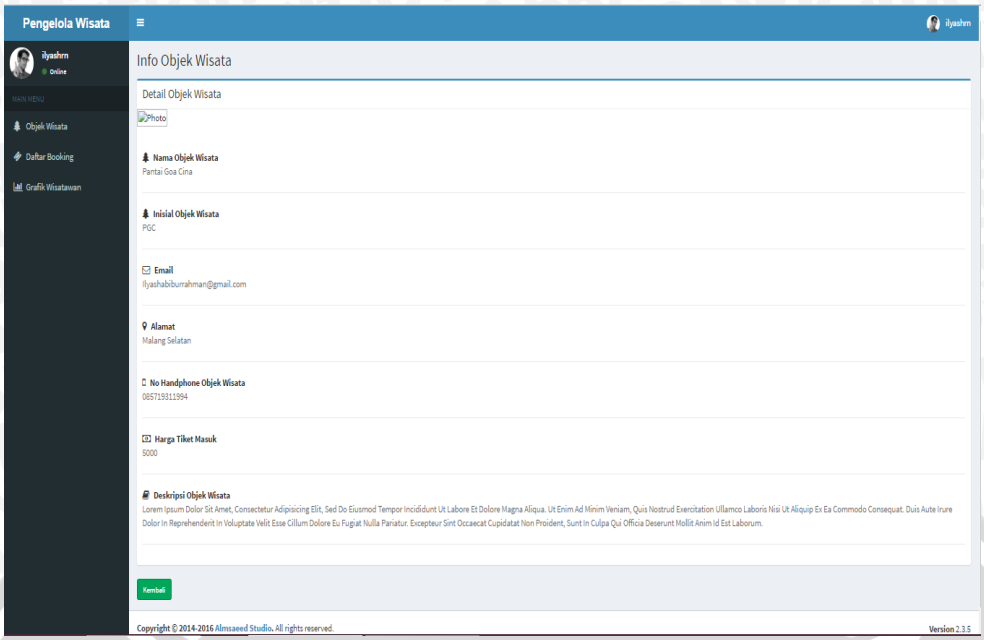
Gambar 6.21 menggambarkan hasil implementasi dari halaman objek wisata pada *dashboard* pengelola. Halaman ini digunakan untuk melihat daftar objek wisata yang dikelolanya dan terdaftar pada sistem.



Gambar 6.21 Halaman Objek Wisata (Pengelola)

v. Halaman Detail Objek Wisata (Pengelola)

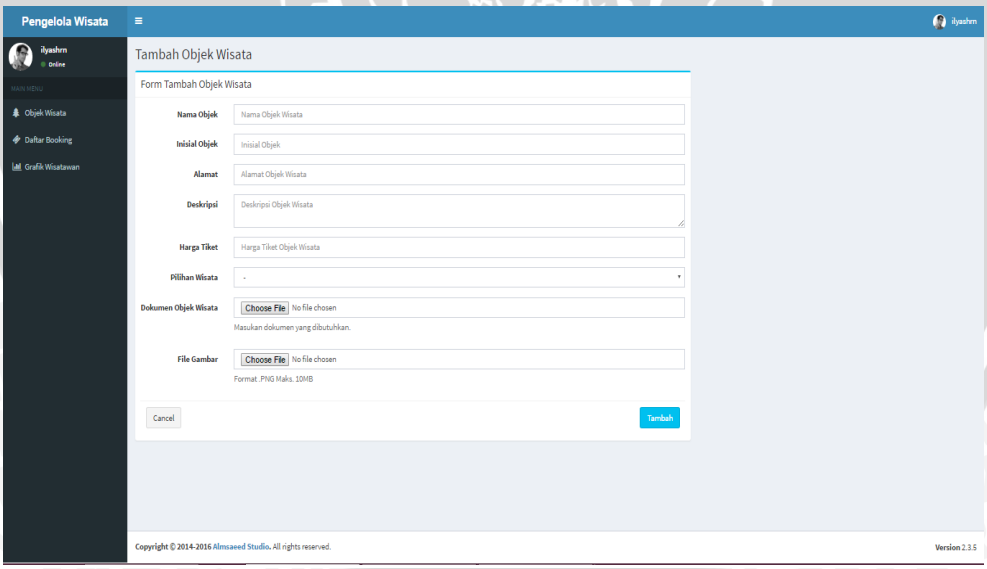
Gambar 6.22 menggambarkan hasil implementasi dari halaman detail objek wisata pada *dashboard* pengelola. Halaman ini digunakan untuk melihat detail informasi objek wisata yang dikelolanya pada sistem.



Gambar 6.22 Halaman Detail Objek (Pengelola)

w. Halaman Tambah Objek Wisata

Gambar 6.23 menggambarkan hasil implementasi dari halaman tambah objek wisata pada *dashboard* pengelola. Halaman ini digunakan untuk melakukan penambahan objek wisata yang masih dikelolanya dengan mengisi *form* tambah objek wisata sesuai dengan data-data yang benar dan dapat dipertanggungjawabkan pada sistem.

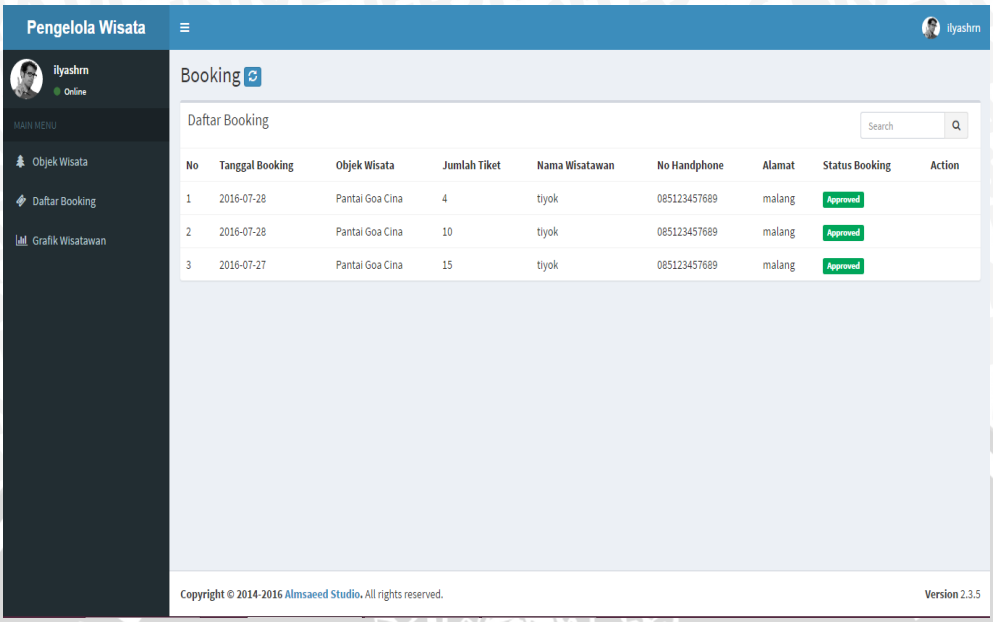


Gambar 6.23 Halaman Tambah Objek Wisata

x. Halaman Daftar *Booking* Objek Wisata

Gambar 6.24 menggambarkan hasil implementasi dari halaman daftar *booking* objek wisata pada *dashboard* pengelola. Halaman ini digunakan untuk melihat

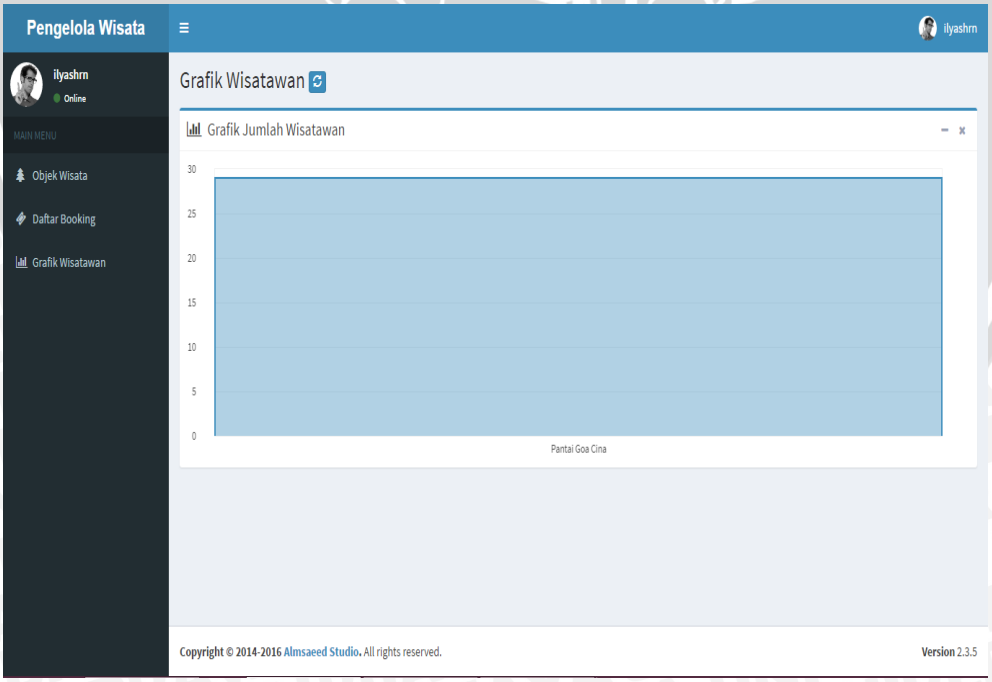
daftar *booking* objek wisata yang sudah masuk dan dapat melakukan konfirmasi booking pada sistem.



Gambar 6.24 Halaman Daftar *Booking* Objek Wisata

y. Halaman Grafik Wisatawan

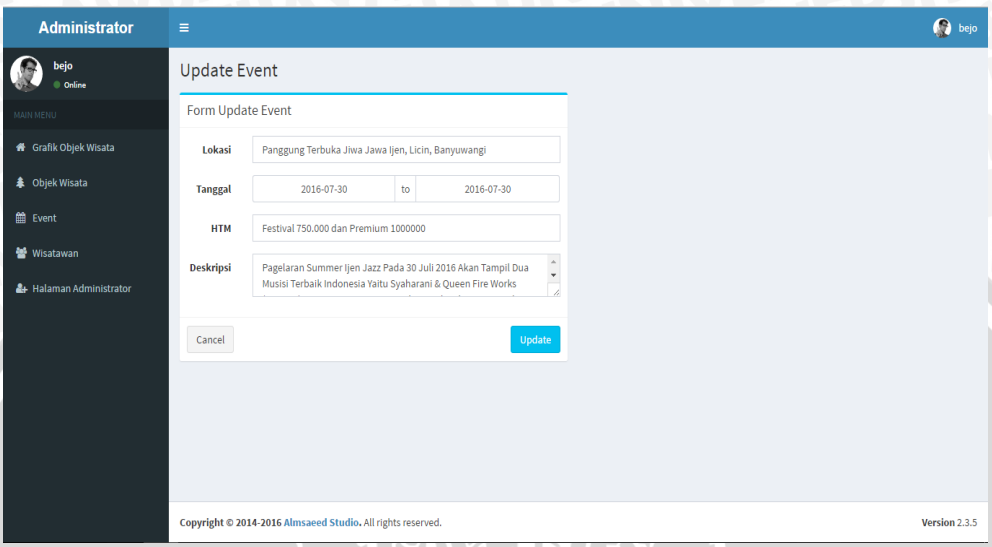
Gambar 6.25 menggambarkan hasil implementasi dari halaman grafik wisatawan pada *dashboard* pengelola. Halaman ini digunakan untuk melihat jumlah wisatawan yang mengunjungi objek wisata yang dikelolanya dalam bentuk grafik pada sistem.



Gambar 6.25 Halaman Grafik Wisatawan

z. Halaman *Update Event*

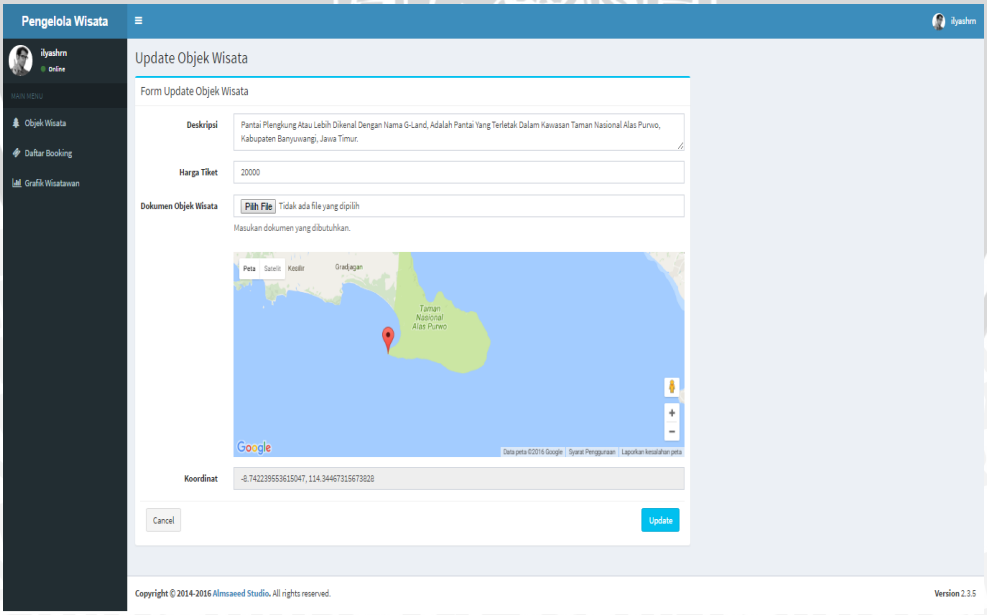
Gambar 6.26 menggambarkan hasil implementasi dari halaman *update event* pada *dashboard admin*. Halaman ini digunakan untuk mengupdate informasi event yang akan berlangsung pada sistem.



Gambar 6.26 Halaman *Update Event*

aa. Halaman *Update* Objek Wisata

Gambar 6.27 menggambarkan hasil implementasi dari halaman *update* objek wisata pada *dashboard* pengelola. Halaman ini digunakan untuk mengupdate informasi objek wisata yang dikelolanya pada sistem.



Gambar 6.27 Halaman *Update* Objek Wisata

BAB 7 PENGUJIAN DAN ANALISIS

Bab ini membahas tahapan pengujian dan analisis Sistem Informasi Manajemen Pariwisata Terpadu Sebagai Media Promosi Kabupaten Banyuwangi. Pembahasan dalam bab ini terdiri atas penjelasan tentang metode pengujian yang digunakan pada penelitian beserta hasil pengujiannya dan analisa hasil dari hasil pengujian yang didapatkan.

7.1 Pengujian Sistem

Pengujian sistem dilakukan dengan 3 (tiga) tahapan yaitu pengujian *black box*, pengujian *white-box* dan pengujian *compatibility*. Pada seluruh proses pengujian ini dilakukan dengan menggunakan perangkat lunak dan perngkat keras sesuai dengan sub bab 6.1 dan sub bab 6.2 yang telah dijelaskan sebelumnya. Pengujian *black box* dilakukan dengan melakukan uji satu persatu fitur yang ada sesuai dengan kebutuhan fungsionalnya, Pengujian *white-box* dilkukan dengan melakukan pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural dan pengujian *compatibility* dilakukan dengan melakukan uji sistem pada *browser* yang berbeda.

7.1.1 Pengujian Validasi dengan *Black-Box*

7.1.1.1 Tujuan

Tujuan dilakukan pengujian validasi adalah untuk mengetahui apakah setiap kebutuhan fungsional yang sudah didefinisikan telah sesuai dengan yang diharapkan dengan cara menguji seluruh skenario dari *use case* yang telah dibuat pada bab dari analisis kebutuhan sebelumnya.

7.1.1.2 Mekanisme

Setiap skenario dari *use case* yang didefinisikan pada tahap analisis kebutuhan akan diujikan pada pengujian validasi ini dengan cara mendefinisikan kasus uji dan hasil yang didapatkan dari setiap kebutuhan fungsional. Kasus uji pada sistem informasi manajemen pariwisata terpadu sebagai berikut.

Tabel 7.1 Tabel Kasus Uji Pengujian pada Login

Nomor Kasus Uji	PV_01
Nama Kasus Uji	Login
Nomor Use Case	SIMPADU-F-002
Prosedur Uji	1. Penguji membuka sistem 2. Penguji memilih menu login pada sistem 3. Penguji memasukkan *username* dan *password*

	4. Penguji menekan tombol *login* untuk masuk ke dalam sistem
Yang Diharapkan	Sistem dapat melakukan pengecekan data inputan yang dilakukan oleh penguji di *database* apakah *username* dan *password* yang dimasukkan oleh penguji sudah terdaftar dan benar agar dapat masuk ke dalam sistem informasi manajemen pariwisata terpadu.
Hasil yang Didapatkan	Sistem dapat melakukan validasi *username* dan *password* agar dapat masuk ke dalam sistem informasi manajemen pariwisata terpadu.

Tabel 7.2 Tabel Kasus Uji Pengujian pada Regitrasi Wisatawan

Nomor Kasus Uji	PV_02
Nama Kasus Uji	Registrasi Wisatawan
Nomor Use Case	SIMPADU-F-001
Prosedur Uji	1. Penguji membuka sistem 2. Penguji memilih menu register wisatawan 3. Penguji memasukkan data diri pada form registrasi wisatawan 4. Penguji menekan tombol "*register*"
Yang Diharapkan	Sistem dapat menampilkan pesan "Register Anda Berhasil" dan data inputan tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan pesan "Register Anda Berhasil" dan data inputan tersimpan di *database* dalam sistem.

Tabel 7.3 Tabel Kasus Uji Pengujian pada Regitrasi Pengelola

Nomor Kasus Uji	PV_03
Nama Kasus Uji	Registrasi Pengelola
Nomor Use Case	SIMPADU-F-003
Prosedur Uji	1. Penguji membuka sistem

	2. Penguji memilih menu registrasi pengelola 3. Penguji memasukkan data diri pada form registrasi pengelola 4. Penguji menekan tombol "*register*"
Yang Diharapkan	Sistem dapat menampilkan pesan "Register Anda Berhasil" dan data inputan tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan pesan "Register Anda Berhasil" dan data inputan tersimpan di *database* dalam sistem.

Tabel 7.4 Tabel Kasus Uji Pengujian pada Melihat Riwayat Booking Objek Wisata

Nomor Kasus Uji	PV_04
Nama Kasus Uji	Melihat Riwayat *Booking* Objek Wisata
Nomor Use Case	SIMPADU-F-007
Prosedur Uji	1. Penguji login ke sistem sebagai wisatawan 2. Penguji memilih menu *booking* 3. Penguji melihat riwayat *booking* yang telah dilakukan
Yang Diharapkan	Sistem dapat menampilkan riwayat *booking* objek wisata yang tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan riwayat *booking* objek wisata yang tersimpan di *database* dalam sistem.

Tabel 7.5 Tabel Kasus Uji Pengujian pada Melihat Detail Booking Objek Wisata

Nomor Kasus Uji	PV_05
Nama Kasus Uji	Melihat Detail *Booking* Objek Wisata
Nomor Use Case	SIMPADU-F-012
Prosedur Uji	1. Penguji login ke sistem sebagai wisatawan 2. Penguji memilih menu *booking*

	3. Penguji menekan tombol "view detail" pada salah satu *booking* objek wisata yang telah dilakukan 4. Penguji melihat tampilan halaman detail info *booking* objek wisata
Yang Diharapkan	Sistem dapat menampilkan detail informasi *booking* objek wisata yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan detail informasi *booking* objek wisata yang telah tersimpan di *database* dalam sistem.

Tabel 7.6 Tabel Kasus Uji Pengujian pada Melakukan Booking Objek Wisata

Nomor Kasus Uji	PV_06
Nama Kasus Uji	Melakukan *Booking* Objek Wisata
Nomor Use Case	SIMPADU-F-006
Prosedur Uji	1. Penguji login ke sistem sebagai wisatawan 2. Penguji memilih menu *booking* 3. Penguji menekan tombol "*booking*" 4. Penguji mengisi *form booking* 5. Penguji menekan tombol "*booking*"
Yang Diharapkan	Sistem dapat menampilkan *form booking* objek wisata dan data *booking* objek wisata tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan *form booking* objek wisata dan data *booking* objek wisata tersimpan di *database* dalam sistem.

Tabel 7.7 Tabel Kasus Uji Pengujian pada Cetak Laporan Booking Objek Wisata

Nomor Kasus Uji	PV_07
Nama Kasus Uji	Cetak Laporan *Booking* Objek Wisata
Nomor Use Case	SIMPADU-F-008
Prosedur Uji	1. Penguji login ke sistem sebagai wisatawan 2. Penguji memilih menu *booking*

	3. Penguji menekan tombol "view detail" pada salah satu *booking* objek wisata yang telah dilakukan 4. Penguji melihat tampilan halaman detail *booking* objek wisata 5. Penguji menekan tombol "*print*"
Yang Diharapkan	Sistem dapat menampilkan halaman cetak laporan *booking* objek wisata dan melakukan cetak laporan *booking* objek wisata yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan halaman cetak laporan *booking* objek wisata dan melakukan cetak laporan *booking* objek wisata yang telah tersimpan di *database* dalam sistem.

Tabel 7.8 Tabel Kasus Uji Pengujian pada Melihat Detail Info *Event*

Nomor Kasus Uji	PV_08
Nama Kasus Uji	Melihat Detail Info *Event*
Nomor Use Case	SIMPADU-F-011
Prosedur Uji	1. Penguji login ke sistem sebagai wisatawan 2. Penguji memilih menu *event* 3. Penguji menekan tombol "view detail" pada salah satu *event* yang ada 4. Penguji melihat tampilan halaman detail info *event* objek wisata
Yang Diharapkan	Sistem dapat menampilkan detail informasi *event* yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan detail informasi *event* yang telah tersimpan di *database* dalam sistem.

Tabel 7.9 Tabel Kasus Uji Pengujian pada Melihat Detail Info Objek Wisata

Nomor Kasus Uji	PV_09
Nama Kasus Uji	Melihat Detail Info Objek Wisata
Nomor Use Case	SIMPADU-F-004
Prosedur Uji	1. Penguji login ke sistem sebagai wisatawan 2. Penguji memilih menu objek wisata 3. Penguji menekan tombol "view detail" pada salah satu objek wisata yang ada pada daftar objek wisata 4. Penguji melihat tampilan halaman detail informasi objek wisata
Yang Diharapkan	Sistem dapat menampilkan detail informasi objek wisata yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan detail informasi objek wisata yang telah tersimpan di *database* dalam sistem.

Tabel 7.10 Tabel Kasus Uji Pengujian pada Melihat Daftar Objek Wisata

Nomor Kasus Uji	PV_10
Nama Kasus Uji	Melihat Daftar Objek Wisata
Nomor Use Case	SIMPADU-F-009
Prosedur Uji	1. Penguji login ke sistem sebagai wisatawan 2. Penguji memilih menu objek wisata 3. Penguji melihat tampilan halaman daftar objek wisata
Yang Diharapkan	Sistem dapat menampilkan daftar objek wisata yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan daftar objek wisata yang telah tersimpan di *database* dalam sistem.

Tabel 7.11 Tabel Kasus Uji Pengujian pada Melihat Daftar *Event*

Nomor Kasus Uji	PV_11
Nama Kasus Uji	Melihat Daftar *Event*
Nomor Use Case	SIMPADU-F-010
Prosedur Uji	1. Penguji login ke sistem sebagai wisatawan 2. Penguji memilih menu *event* 3. Penguji melihat tampilan halaman daftar *event*
Yang Diharapkan	Sistem dapat menampilkan daftar *event* yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan daftar *event* yang telah tersimpan di *database* dalam sistem.

Tabel 7.12 Tabel Kasus Uji Pengujian pada Menambah *Event*

Nomor Kasus Uji	PV_12
Nama Kasus Uji	Menambah *Event*
Nomor Use Case	SIMPADU-F-020
Prosedur Uji	1. Penguji login ke sistem sebagai *admin* 2. Penguji memilih menu *event* 3. Penguji menekan tombol “tambah” 4. Penguji melihat halaman *form* tambah *event* 5. Penguji memasukkan data *event* 6. Penguji menekan tombol “tambah”
Yang Diharapkan	Sistem dapat menampilkan pesan “Penambahan *event* berhasil” dan data masukkan tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan pesan “Penambahan *event* berhasil” dan data masukkan tersimpan di *database* dalam sistem.

Tabel 7.13 Tabel Kasus Uji Pengujian pada Menghapus *Event*

Nomor Kasus Uji	PV_13
Nama Kasus Uji	Menghapus *Event*
Nomor Use Case	SIMPADU-F-021
Prosedur Uji	1. Penguji login ke sistem sebagai *admin* 2. Penguji memilih menu *event* 3. Penguji melihat tampilan halaman daftar *event* 4. Penguji menekan tombol "hapus" pada salah satu *event*
Yang Diharapkan	Sistem dapat menghapus *event* yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menghapus *event* yang telah tersimpan di *database* dalam sistem.

Tabel 7.14 Tabel Kasus Uji Pengujian pada Mengupdate *Event*

Nomor Kasus Uji	PV_14
Nama Kasus Uji	Mengupdate *Event*
Nomor Use Case	SIMPADU-F-026
Prosedur Uji	1. Penguji login ke sistem sebagai *admin* 2. Penguji memilih menu *event* 3. Penguji melihat tampilan halaman daftar *event*
Yang Diharapkan	Sistem dapat menampilkan *form update event* dan data masukkan telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan *form update event* dan data masukkan telah tersimpan di *database* dalam sistem.

Tabel 7.15 Tabel Kasus Uji Pengujian pada Melihat Daftar *Administrator*

Nomor Kasus Uji	PV_15
Nama Kasus Uji	Melihat Daftar *Administrator*
Nomor Use Case	SIMPADU-F-023
Prosedur Uji	1. Penguji login ke sistem sebagai *admin*

	2. Penguji memilih menu halaman *administrator* 3. Penguji melihat tampilan halaman daftar *administrator*
Yang Diharapkan	Sistem dapat menampilkan daftar *administrator* yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan daftar *administrator* yang telah tersimpan di *database* dalam sistem.

Tabel 7.16 Tabel Kasus Uji Pengujian pada Menambah *Administrator*

Nomor Kasus Uji	PV_16
Nama Kasus Uji	Menambah *Administrator*
Nomor Use Case	SIMPADU-F-025
Prosedur Uji	1. Penguji login ke sistem sebagai *admin* 2. Penguji memilih menu halaman *administrator* 3. Penguji menekan tombol “tambah” 4. Penguji mengisi *form* tambah *administrator* 5. Penguji menekan tombol “tambah”
Yang Diharapkan	Sistem dapat menampilkan pesan “Penambahan admin sukses” dan data masukkan telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan pesan “Penambahan admin sukses” dan data masukkan telah tersimpan di *database* dalam sistem.

Tabel 7.17 Tabel Kasus Uji Pengujian pada Menghapus *Administrator*

Nomor Kasus Uji	PV_17
Nama Kasus Uji	Menghapus *Administrator*
Nomor Use Case	SIMPADU-F-024
Prosedur Uji	1. Penguji login ke sistem sebagai *admin*

	2. Penguji memilih menu halaman *administrator* 3. Penguji menekan tombol “hapus”
Yang Diharapkan	Sistem dapat menghapus salah satu *administrator* dan data telah terhapus di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menghapus salah satu *administrator* dan data telah terhapus di *database* dalam sistem.

Tabel 7.18 Tabel Kasus Uji Pengujian pada Verifikasi Objek Wisata

Nomor Kasus Uji	PV_18
Nama Kasus Uji	Verifikasi Objek Wisata
Nomor Use Case	SIMPADU-F-019
Prosedur Uji	1. Penguji login ke sistem sebagai *admin* 2. Penguji memilih menu objek wisata 3. Penguji menekan tombol “verifikasi”
Yang Diharapkan	Sistem dapat menampilkan status “*verified*” dan data tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan status “*verified*” dan data tersimpan di *database* dalam sistem.

Tabel 7.19 Tabel Kasus Uji Pengujian pada Melihat Grafik Objek Wisata

Nomor Kasus Uji	PV_19
Nama Kasus Uji	Melihat Grafik Objek Wisata
Nomor Use Case	SIMPADU-F-022
Prosedur Uji	1. Penguji login ke sistem sebagai *admin* 2. Penguji memilih menu halaman grafik objek wisata 3. Penguji melihat grafik objek wisata
Yang Diharapkan	Sistem dapat menampilkan grafik objek wisata dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan grafik objek wisata dalam sistem.

Tabel 7.20 Tabel Kasus Uji Pengujian pada Melihat Daftar Wisatawan

Nomor Kasus Uji	PV_20
Nama Kasus Uji	Melihat Daftar Wisatawan
Nomor Use Case	SIMPADU-F-018
Prosedur Uji	1. Penguji login ke sistem sebagai *admin* 2. Penguji memilih menu halaman wisatawan 3. Penguji melihat daftar wisatawan
Yang Diharapkan	Sistem dapat menampilkan daftar wisatawan yang telah terdaftar dan tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan daftar wisatawan yang telah terdaftar dan tersimpan di *database* dalam sistem.

Tabel 7.21 Tabel Kasus Uji Pengujian pada Menambah Objek Wisata

Nomor Kasus Uji	PV_21
Nama Kasus Uji	Menambah Objek Wisata
Nomor Use Case	SIMPADU-F-015
Prosedur Uji	1. Penguji login ke sistem sebagai pengelola 2. Penguji memilih menu halaman objek wisata 3. Penguji menekan tombol “tambah” 4. Penguji mengisi *form* tambah objek wisata 5. Penguji menekan tombol “tambah”
Yang Diharapkan	Sistem dapat menampilkan pesan “Penambahan info wisata berhasil. Silahkan tunggu verifikasi dari *Administrator*.” dan data masukkan telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan pesan “Penambahan info wisata berhasil. Silahkan tunggu verifikasi dari *Administrator*.” dan data masukkan telah tersimpan di *database* dalam sistem.

Tabel 7.22 Tabel Kasus Uji Pengujian pada Konfirmasi *Booking* Objek Wisata

Nomor Kasus Uji	PV_22
Nama Kasus Uji	Konfirmasi *Booking* Objek Wisata
Nomor Use Case	SIMPADU-F-014
Prosedur Uji	1. Penguji login ke sistem sebagai pengelola 2. Penguji memilih menu halaman daftar *booking* 3. Penguji menekan tombol "konfirmasi"
Yang Diharapkan	Sistem dapat menampilkan status *booking* "*approved*" dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan status *booking* "*approved*" dalam sistem.

Tabel 7.23 Tabel Kasus Uji Pengujian pada Melihat Grafik Jumlah Wisatawan

Nomor Kasus Uji	PV_23
Nama Kasus Uji	Melihat Grafik Jumlah Wisatawan
Nomor Use Case	SIMPADU-F-016
Prosedur Uji	1. Penguji login ke sistem sebagai pengelola 2. Penguji memilih menu halaman grafik wisatawan 3. Penguji melihat grafik jumlah wisatawan
Yang Diharapkan	Sistem dapat menampilkan grafik jumlah wisatawan dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan grafik jumlah wisatawan dalam sistem.

Tabel 7.24 Tabel Kasus Uji Pengujian pada Daftar *Booking* Objek Wisata

Nomor Kasus Uji	PV_24
Nama Kasus Uji	Melihat Daftar *Booking* Objek Wisata
Nomor Use Case	SIMPADU-F-013
Prosedur Uji	1. Penguji login ke sistem sebagai pengelola

	2. Penguji memilih menu halaman daftar *booking* 3. Penguji melihat daftar *booking* objek wisata
Yang Diharapkan	Sistem dapat menampilkan daftar *booking* objek wisata yang telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan daftar *booking* objek wisata yang telah tersimpan di *database* dalam sistem.

Tabel 7.25 Tabel Kasus Uji Pengujian pada Mengupdate Objek Wisata

Nomor Kasus Uji	PV_25
Nama Kasus Uji	Mengupdate Objek Wisata
Nomor Use Case	SIMPADU-F-017
Prosedur Uji	1. Penguji login ke sistem sebagai pengelola 2. Penguji memilih menu halaman objek wisata 3. Penguji menekan tombol "view detail" 4. Penguji melihat halaman detail informasi objek wisata 5. Penguji menekan tombol "*update*" 6. Penguji mengisi form update informasi objek wisata 7. Penguji menekan tombol "*update*"
Yang Diharapkan	Sistem dapat menampilkan detail informasi objek wisata yang telah diupdate dan data masukkan telah tersimpan di *database* dalam sistem.
Hasil yang Didapatkan	Sistem dapat menampilkan detail informasi objek wisata yang telah diupdate dan data masukkan telah tersimpan di *database* dalam sistem.

7.1.1.3 Hasil Pengujian

Hasil pengujian *black-box* dengan cara validasi dapat dilihat pada Tabel 7.26 berikut.

Tabel 7.26 Hasil Pengujian Validasi dengan *Black-Box*

Nomor Kasus Uji	Fungsi	Status
PV_01	Login	Valid
PV_02	Registrasi Wisatawan	Valid
PV_03	Registrasi Pengelola	Valid
PV_04	Melihat Riwayat *Booking* Objek Wisata	Valid
PV_05	Melihat Detail *Booking* Objek Wisata	Valid
PV_06	Melakukan *Booking* Objek Wisata	Valid
PV_07	Cetak Laporan *Booking* Objek Wisata	Valid
PV_08	Melihat Detail Info *Event*	Valid
PV_09	Melihat Detail Info Objek Wisata	Valid
PV_10	Melihat Daftar Objek Wisata	Valid
PV_11	Melihat Daftar *Event*	Valid
PV_12	Menambah *Event*	Valid
PV_13	Menghapus *Event*	Valid
PV_14	Mengupdate *Event*	Valid
PV_15	Melihat Daftar *Administrator*	Valid
PV_16	Menambah *Administrator*	Valid
PV_17	Menghapus *Administrator*	Valid
PV_18	Verifikasi Objek Wisata	Valid
PV_19	Melihat Grafik Jumlah Wisatawan	Valid
PV_20	Melihat Daftar Wisatawan	Valid
PV_21	Menambah Objek Wisata	Valid
PV_22	Konfirmasi *Booking* Objek Wisata	Valid
PV_23	Melihat Grafik Jumlah Wisatawan	Valid
PV_24	Melihat Daftar *Booking* Objek Wisata	Valid
PV_25	Mengupdate Objek Wisata	Valid

7.1.2 Pengujian *White-Box*

7.1.2.1 Tujuan

Tujuan dilakukan pengujian *white-box* adalah untuk memberikan jaminan bahwa jalur independen pada modul digunakan minimal satu kali, menggunakan semua keputusan logis untuk kondisi *true* atau *false*.

7.1.2.2 Mekanisme

Mekanisme pengujian white-box dapat dilakukan dengan cara sebagai berikut:

1. Mendefinisikan semua alur logika.
2. Membangun kasus untuk digunakan dalam pengujian.
3. Mengevaluasi semua hasil pengujian.
4. Melakukan pengujian secara menyeluruh.

7.1.2.3 Hasil Pengujian

1. **Tambah *Event***

```
public function tambah_process()
{
    $config['upload_path']   = './uploads/gambar/';
    $config['allowed_types'] = 'png|jpg|jpeg';
    $config['max_size']      = 10240;

    $this->load->library('upload', $config);

if ( ! $this->upload->do_upload('gambar')) {
$this->session->set_flashdata('error', $this->upload->display_errors().' gambar');
    redirect('event/tambah','refresh');
    }
    else {
        $gambar = $this->upload->data();

    $data = array(
            'nama_event' => $this->input->post('nama_event'),
            'lokasi' => $this->input->post('lokasi'),
```

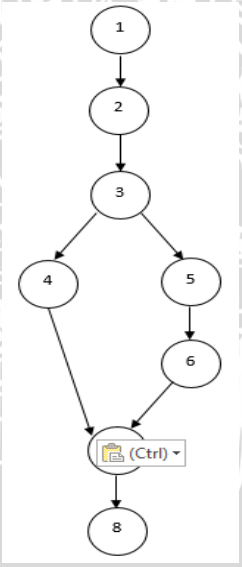
1
2
3
4
5
6

```
            'tanggal' => $this->input->post('tanggal'),
            'tanggal2' => $this->input->post('tanggal2'),
            'gambar' => $gambar['file_name'],
            'deskripsi' => $this->input->post('deskripsi'),
            'htm' => $this->input->post('htm'),
            'id_admin' => $this->session->userdata('id')
        );

        $this->info_event_model->insert($data);
        $this->session->set_flashdata('success', true);
        redirect('event','refresh');
    }
}
```

7

8

Flowgraph:



Gambar 7.1 *Flowgraph* Tambah *Event*

Kompleksitas:
V (G) = E – N + 2 = 8 – 8 + 2 = 2
V (G) = 2 *region*
V (G) = *Predicate node* + 1 = 1 + 1 = 2
Independen *Path*:

1. 1-2-3-4-7-8
2. 1-2-3-5-6-7-8

2. Tambah Objek Wisata

```
public function tambah_process()
{
        $config['upload_path']   = './uploads/gambar/';
        $config['allowed_types'] = 'png|jpg|jpeg';
        $config['max_size']      = 10240;
        $this->load->library('upload', $config);

        if ( ! $this->upload->do_upload('gambar')) {
          $this->session->set_flashdata('error',        $this->upload-
>display_errors().' gambar');
        redirect('wisata/tambah','refresh');
}
        else {
        $gambar = $this->upload->data();
        $config2['upload_path']   = './uploads/dokumen/';
        $config2['allowed_types'] = 'pdf';
        $config2['max_size']      = 10240;
        // $config['max_width']     = 1024;
        // $config['max_height']    = 768;
        // $this->load->library('upload', $config2);
        $this->upload->initialize($config2);

        if ( ! $this->upload->do_upload('dokumen')) {
                $this->session->set_flashdata('error', $this->upload-
>display_errors().' dokumen');
                redirect('wisata/tambah','refresh');
        }
        else {
                $dokumen = $this->upload->data();
                $data = array(
                  'nama_objek' => $this->input->post('nama_objek'),
```

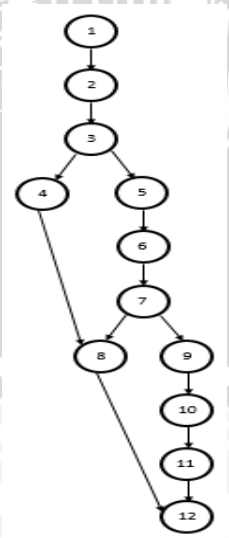
```
                'inisial'=>strtoupper($this->input->post('inisial_objek')),
                'alamat' => $this->input->post('alamat'),
                'deskripsi' => $this->input->post('deskripsi'),
                'harga_tiket' => $this->input->post('harga_tiket'),
                'status' => 0,
                'id_pengelola' => $this->session->userdata('id'),
                'pilihan_wisata'=>$this->input->post('pilihan_wisata'),
                'dokumen' => $dokumen['file_name'],
                'gambar' => $gambar['file_name'],
                'koordinat' => $this->input->post('koordinat'),
            );

            $this->objek_wisata_model->insert($data);
            $this->session->set_flashdata('success', true);
            redirect('wisata','refresh');
        }
    }
}
```

10

11

12

Flowgraph:



Gambar 7.2 *Flowgraph* Tambah Objek Wisata

Kompleksitas:
V (G) = E – N + 2 = 13 – 12 + 2 = 3
V (G) = 3 *region*
V (G) = *Predicate node* + 1 = 2 + 1 = 3
Independen *Path*:

1. 1-2-3-4-8-12
2. 1-2-3-5-6-7-8-12
3. 1-2-3-5-6-7-9-10-11-12

4. *Update* Objek Wisata

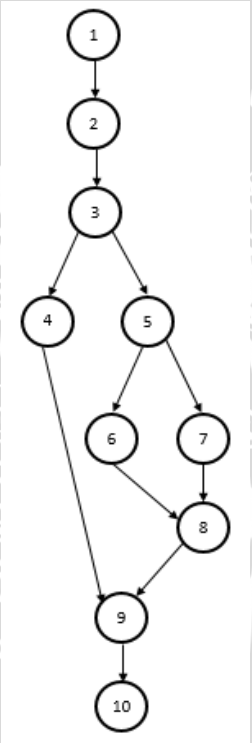
```
public function update_process()   1
{
    $data = array(
        'deskripsi' => $this->input->post('deskripsi'),   2
        'harga_tiket' => $this->input->post('harga_tiket'),
        'id_pengelola' => $this->session->userdata('id')
    );

if($this->input->post('dokumen')){   3
        $config2['upload_path']   = './uploads/dokumen/';
        $config2['allowed_types'] = 'pdf';
        $config2['max_size']      = 10240;
        // $config['max_width']     = 1024;   4
        // $config['max_height']    = 768;
        // $this->load->library('upload', $config2);
        $this->upload->initialize($config2);

    if ( ! $this->upload->do_upload('dokumen')) {   5
        $this->session->set_flashdata('error', $this->upload-
>display_errors().' dokumen');
        redirect('wisata/update?id='.$this->input-   6
>post('id_wisata'),'refresh');
    }
    else{   7
        $dokumen = $this->upload->data();   8
```

```
                $data['dokumen'] = $dokumen['file_name'];
            }
        }
                $this->db->where('id_wisata',$this->input->post('id_wisata'));
                $this->db->update('objek_wisata', $data);
                redirect('wisata/detail?id='.$this->input->post('id_wisata'),'refresh');
    }
```

9

10

Flowgraph:



Gambar 7.3 *Flowgraph Update* Informasi Objek Wisata

Kompleksitas:
V (G) = E – N + 2 = 11 – 10 + 2 = 3
V (G) = 3 *region*
V (G) = *Predicate node* + 1 = 2 + 1 = 3
Independen *Path*:

1. 1-2-3-4-9-10
2. 1-2-3-5-6-8-9-10
3. 1-2-3-5-7-8-9-10

7.1.3 Pengujian *Compatibility*

7.1.3.1 Tujuan

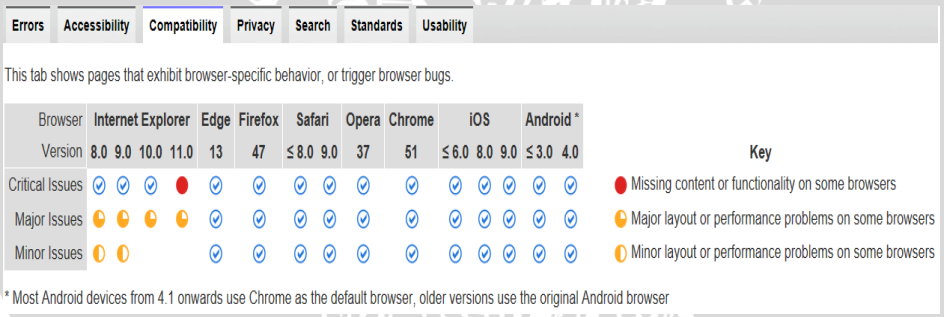
Tujuan dilakukan pengujian *compatibility* adalah untuk mengetahui apakah sistem bisa berjalan normal pada *browser* yang berbeda-beda karena banyaknya pengguna sistem.

Pengujian dilakukan dengan menggunakan beberapa *browser* yang berbeda yaitu *Mozilla Firefox, Google Chrome, Internet Explorer, Microsoft Edge, Safari, browser Android,* dan *browser iOS.* Pada penelitian ini, pengujian *compability* dilakukan menggunakan aplikasi *SortSite trial version* 5.22.764.0. Aplikasi *Sortsite* dapat menguji perangkat lunak berbasis *web* secara keseluruhan pada setiap halaman yang dipilih.

7.1.3.2 Mekanisme

Mekanisme pengujian ini dengan melakukan pendefinisian daftar kebutuhan yang diperlukan untuk menguji sistem dan membuat sebuah skenario pengujian serta hasil yag akan diperoleh terhadap setiap pengujian tersebut.

Hasil pengujian akan di dapatkan ketika perangkat lunak dipilih kemudian aplikasi *sortsite* akan mengecek setiap halaman dan menampilkan hasil dari pengujian dalam bentuk tabel seperti pada gambar 7.6.


Errors | Accessibility | Compatibility | Privacy | Search | Standards | Usability

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	Internet Explorer				Edge	Firefox	Safari		Opera	Chrome	iOS			Android *		Key
Version	8.0	9.0	10.0	11.0	13	47	≤ 8.0	9.0	37	51	≤ 6.0	8.0	9.0	≤ 3.0	4.0	
Critical Issues	✓	✓	✓	●	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	● Missing content or functionality on some browsers
Major Issues	◔	◔	◔	◔	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	◔ Major layout or performance problems on some browsers
Minor Issues	◐	◐			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	◐ Minor layout or performance problems on some browsers

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 7.4 Hasil Pengujian *Compatibility* dengan *Sortsite*

7.1.3.3 Hasil Pengujian

Hasil pengujian didapatkan setelah proses pengujian dilakukan yang ditampilkan dalam bentuk tabel berdasarkan kriteria-kriteria pada pengujian *compability*. Hasil pengujian menunjukkan bahwa fungsionalitas sistem informasi manajemen pariwisata terpadu dapat berjalan dengan baik diberbagai *platfom* seperti *internet explorer, edge, browser iOS, browser android, mozilla firefox, opera* dan *chrome*. Tetapi terdapat *manor issuses* yang mengakibatkan layout utama seperti css pada sistem informasi manajemen pariwisata terpadu tidak berjalan baik pada *platform internet explorer* 8.0, 9.0, 10.0, 11.0 dan juga terdapat *minor issuses* yang mengakibatkan properti layout pada sistem informasi manajemen pariwisata terpadu tidak berjalan dengan baik

pada *platform internet explorer* 8.0, 9.0. Hasil pengujian *compability* secara detail dapat dilihat pada tabel 7.27.

Dari hasil pengujian pada tabel 7.27 dapat disimpulkan bahwa sistem informasi manajemen pariwisata terpadu sudah *compatible* dengan 8 *browser* yang sudah dilakukan pengujian sehingga telah memenuhi persyaratan non-fungsional yang sudah di tetapkan.

Tabel 7.27 Hasil Pengujian *Compatibility*

Browser	**Hasil**	**Masalah**	**Keterangan**
Internet Explorer 8.0, 9.0, 10.0, 11.0	Terdapat *major issues*	- *CSS pseudo selector* - *CSS star filter*	- *Internet Explorer 8.0, 9.0, 10.0, dan 11.0* belum didukung dengan selektor *CSS pseudo*. - Filter *CSS star* tidak lagi bekerja di *Internet Explorer 7.0*.
Internet Explorer 11.0	Terdapat *critical issues*	*XdomainRequest object*	*Internet Explorer 11.0* belum mendukung *XdomainRequest object*
Internet Explorer 8.0, 9.0	Terdapat *minor issues*	- *CSS property border-radius* - *CSS selector for INPUT TYPE="SEARCH"* - *Box-shadow CSS property* - *Opacity CSS3 property* - *Required attribute* - *Text-shadow CSS property*	- *Internet Explorer 8.0, 9.0* tidak mendukung properti *CSS* seperti *border-radius*. - *CSS* selektor untuk input type seperti *search* diabaikan di *Internet Explorer 8.0, 9.0*. - *Internet Explorer 8.0, 9.0* tidak mendukung

			properti *CSS* seperti *box-shadow*. - *Internet Explorer 8.0, 9.0* tidak mendukung properti *CSS* seperti *opacity CSS 3*. - *Internet Explorer 8.0, 9.0* tidak mendukung atribut *required*. - *Internet Explorer 8.0, 9.0* tidak mendukung properti *CSS* seperti *text-shadow*.
Edge 13	Dapat berjalan dengan baik	-	-
Firefox 47	Dapat berjalan dengan baik	-	-
Safari ≤ 8.0, 9.0	Dapat berjalan dengan baik	-	-
Opera 37	Dapat berjalan dengan baik	-	-
iOS ≤ 6.0, 8.0, 9.0	Dapat berjalan dengan baik	-	-
Android ≤ 3.0, 4.0	Dapat berjalan dengan baik	-	-

7.2 Analisis Sistem

Pada sub bab ini akan dilakukan seluruh analisis sistem dari semua pengujian yang telah dilakukan pada sub bab sebelumnya, analisis bertujuan untuk

menganalisa data dari hasil pengujian hingga menghasilkan kesimpulan. Analisis ini dilakukan berdasarkan hasil implementasi hingga pengujian yang telah dilakukan pada sistem dan mengacu pada dasar teori. Proses analisis yang dilakukan antara lain adalah analisis pengujian validasi dengan *black-box*, analisis pengujian *white-box* dan analisa hasil pengujian *compatibility*.

7.2.1 Analisis Pengujian Validasi dengan *Black-Box*

Berdasarkan hasil pengujian validasi dengan *black-box*, implementasi sistem dapat dikatakan layak digunakan dan berhasil serta sesuai dengan kebutuhan fungsional yang telah dijabarkan, hasil pengujian menunjukkan dari 25 kasus uji pengujian yang diuji minimal 2 kali menunjukkan bahwa 100 % kebutuhan fungsional dari sistem telah valid ketika dilakukan pengujian. Sehingga dapat disimpulkan bahwa Sistem Informasi Manajemen Pariwisata Terpadu Sebagai Media Promosi Kabupaten Banyuwangi yang telah diimplementasikan telah berjalan dengan baik sesui dengan yang diharapkan.

7.2.2 Analisis Pengujian *White-Box*

Berdasarkan hasil pengujian *white-box*, dengan cara menganalisis dan menghitung menggunakan *basis path* dan *cyclomatic complexity* dapat dikatakan sudah sesuai dengan logika yang digunakan dalam proses implementasi sistem. Sehingga Sistem Informasi Manajemen Pariwisata Terpadu Sebagai Media Promosi Kabupaten Banyuwangi sudah dapat digunakan dan sejauh ini masih terminimalisir dari kesalahan-kesalahan logika, asumsi, dan kesalahan ketik.

7.2.3 Analisis Pengujian *Compatibility*

Berdasarkan hasil pengujian *compability* yang dilakukan dengan melihat hasil kinerja sistem pada *browser edge 13, firefox 47, chrome, safari, opera, browser iOS,* dan *browser Android*. Berdasarkan hasil pengujian *compability* dapat disimpulkan bahwa Sistem Informasi Manajemen Pariwisata Terpadu Sebagai Media Promosi Kabupaten Banyuwangi dapat berjalan dengan baik dan normal pada 8 *browser* yang berbeda.

BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

1. Dalam melakukan analisis kebutuhan Sistem Informasi Manajemen Pariwisata Terpadu sebagai Media Promosi Kabupaten Banyuwangi telah sesuai dengan hasil observasi dan wawancara yang dilakukan.
2. Dalam perancangan dan implementasi Sistem Informasi Manajemen Pariwisata Terpadu sebagai Media Promosi Kabupaten Banyuwangi menggunakan metode *object oriented design* (OOD) telah sesuai dengan kebutuhan pengguna.
3. Berdasarkan hasil pengujian yang telah dilakukan pada Sistem Informasi Manajemen Pariwisata Terpadu sebagai Media Promosi Kabupaten Banyuwangi dapat diperoleh kesimpulan bahwa secara fungsionalitas dari 25 pengujian kebutuhan fungsional sistem ini berfungsi 100% valid dan logika dari fungsi-fungsi secara keseluruhan berjalan sesuai dengan skenario *use case* yang telah dilakukan sebelumnya dan sistem berjalan secara normal pada 8 *browser* yang berbeda.

8.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah:

1. Untuk penelitian selanjutnya, diharapkan sistem ini dapat mencakup semua aspek pariwisata yang terdapat di Kabupaten Banyuwangi seperti *booking* hotel, menampilkan daftar wisata kuliner, menampilkan daftar *tour guide*, booking *tour guide*, menampilkan *travel agent*, dan booking *travel agent*.
2. Pengimplementasian Sistem Informasi Manajemen Pariwisata Terpadu sebagai Media Promosi Kabupaten Banyuwangi dapat dilakukan dengan mengunakan sistem operasi *mobile* seperti *android* dan *iOS*.
3. Pengembangan sistem dapat dilakukan dengan menambahkan fitur sms *gateway* untuk informasi *event*, informasi objek wisata, dan *booking* objek wisata.

DAFTAR PUSTAKA

Arief, M.Rudianto. 2011. Pemrograman Web Dinamis Menggunakan Php dan Mysql. Yogyakarta: ANDI.

Basu Swastha. 2000. Manajemen Pemasaran Modern Edisi Kedelapan. Yogyakarta: Liberty.

Booch, G., 2007. *Object-Oriented Analysis and Design with Applications Third Edition.* Westford.

Boone, Louis E dan Kurtz, David L. 2002. Pengantar Bisnis (Jilid 2). Jakarta: Erlangga.

Darono, Agung. 2009. Pengembangan sistem informasi manajemen pajak berbasis Financial information system model : suatu kajian pendahuluan. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi.

Davenport, Thomas. 1993. *Process Innovation: Reengineering work through information technology*. Boston: Harvard Business School Press.

Dennis, Alan. 2012. Systems analysis design, UML version 2.0 : an object oriented approach. United State : Wiley.

Fowler, Martin. 2005. UML Distilled (Edisi 3). Yogyakarta: ANDI.

Jackpamot. 2013. [online] tersedia di: http://universitaspendidikan.com/pengertian-white-box-dan-contoh-white-box-testing/ [diakses 20 Oktober 2016]

Jogiyanto, Hartono. 2005. Analisis dan Desain Sistem Informasi (Edisi III). Yogyakarta: ANDI.

Kendall, Kenneth E. dan Kendall, Julie E. 2006. Analisis dan Perancangan Sistem, Jakarta: PT. Indeks.

Kristanto, Andi. 2010. Kupas Tuntas PHP & MYSQL. Jam Menguasai PHP dan MYSQL Dengan Mudah & Cepat. Klaten: Cable Book.

Munawar. 2005. Pemodelan Visual dengan UML. Yogyakarta: Graha Ilmu.

Nugroho, Adi. 2009. Rekayasa Perangkat Lunak Menggunakan UML & Java. Yogyakarta: ANDI.

Pemerintah Daerah Kabupaten Banyuwangi. 2016. Peraturan Bupati Banyuwangi Nomor 56 Tahun 2016. Banyuwangi: Sekretaris Daerah Kabupaten Banyuwangi.

Pemerintah Daerah Kabupaten Banyuwangi. 2011. Peraturan Daerah Nomor 6 Tahun 2011. Banyuwangi: Bupati Banyuwangi.

Prayitno, W., 2005. *Desain Model Sistem Perangkat Lunak dengan UML.*

Pressman, R.S. 2010. Software Engineering : a practitioner's approach. New York: McGraw-Hill.

Rogers, M. 2016. [online] tersedia di: http://www.powermapper.com/products/ [di akses 20 Oktober 2016].

Sommerville, Ian. 2011. Software Engineering (Rekayasa Perangkat Lunak). Jakarta: Erlangga.

Subiyakto, A'ang., Huda, M. Qomarul., dan Prasetyo, W.S. 2007. Pengembangan Sistem Informasi Manajemen (S.I.M) Akademik Terintegrasi (Studi Kasus Di Perguruan Tinggi XYZ). Jakarta Universitas Islam Negeri Syarif Hidayatullah.

Turban, Efraim., McClean, Ephraim., Wetherbe. James. 1999. *Information Technology for Management Making: Connections for Strategic Advantage. 2 nd Edition*, Canada: John Wiley & Sons.Inc.

LAMPIRAN A WANCARA

IDENTIFIKASI MASALAH SISTEM INFORMASI MANAJEMEN PARIWISATA DI KABUPATEN BANYUWANGI

Pertanyaan-pertanyaan di bawah ini peneliti ajukan kepada Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi agar dapat mengidentifikasi permasalahan sistem informasi manajemen pariwisata yang sudah berjalan. Jawaban dari pertanyaan-pertanyaan tersebut akan peneliti gunakan sebagai salah satu landasan untuk menggagas sebuah penelitian guna untuk menyelesaikan permasalahan sistem informasi manajemen pariwisata di Kabupaten Banyuwangi.

1. **Bagaimana alur proses bisnis yang sudah dilakukan oleh Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi?**
   - Melakukan promosi dengan kontinue dan disesuaikan dengan kebutuhan wisatawan.
   - Menggelar event dan festival agar wisatawan bisa setiap saat berkunjung ke banyuwangi.
2. **Apakah untuk saat ini sudah memiliki sebuah media yang mewadahi semua informasi terkait pariwisata di Kabupaten Banyuwangi?**
   Ada, banyuwangikab.go.id dan banyuwangitourism.com
3. **Selain media informasi seperti website, apakah ada media lainnya untuk mempromosikan dan memberikan informasi terkait pariwisata di Kabupaten Banyuwangi?**
   - Twitter
   - Facebook
   - Instagram
   - Grup media sosial kumpulan para netizen banyuwangi
   - Bagian Humas dan Protokol Pemkab Banyuwangi
   - Dinas Infokom Kabupaten Banyuwangi
4. **Apakah media informasi yang saat ini telah digunakan sebagai media promosi dan informasi terkait pariwisata sudah berjalan dengan baik?**
   Belum sepenuhnya, dikarenakan media informasi saat ini masih belum dikelola dengan baik dan terperinci.
5. **Apa saja keterbatasan yang dimiliki oleh media informasi yang digunakan saat ini?**
   - Masih hanya berupa informasi umum terkait destinasi pariwisata
   - Belum seluruh desa terkoneksi jaringan FO (Fiber Optik)
   - Masih kurangnya sosialisasi terkait media informasi pariwisata
6. **Apakah perlu adanya kerjasama antara pihak pengelola objek wisata yang ada di Kabupaten Banyuwangi dengan pihak Pemerintah Daerah Kabupaten Banyuwangi khususnya Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi?**
   Sangat perlu dan diwadahi dalam organisasi POKDARWIS (Kelompok Sadar Wisata)
7. **Apakah perlu diterapkan sistem informasi manajemen untuk mengolah data dan informasi pariwisata yang ada di Kabupaten Banyuwangi?**
   Sangat perlu, untuk memudahkan wisatawan melihat destinasi dan hal-hal lain terkait pariwisata
8. **Jika sistem informasi manajemen diperlukan, apa saja fitur yang diharapkan ada di dalam sistem informasi manajemen tersebut?**
   Semua hal yang ada informasi terkait kebutuhan wisatawan, yaitu:
   - Hotel
   - Restoran
   - Perbankan
   - Wisata
   - Budaya
   - Event
   - Pertsar
   - Travel agent
   - dll

Mengetahui dengan asli
Plt. Kepala Dinas Kebudayaan dan Pariwisata Kabupaten Banyuwangi

PEMERINTAH KABUPATEN BANYUWANGI
DINAS KEBUDAYAAN DAN PARIWISATA

M.Y. BRAMUDA, S.Sos. MBA, MM
Pembina Tingkat I
NIP. 19730120 199201 1 001