

**MEMBANGUN *APPLICATION PROGRAMMING INTERFACE*
SISTEM INFORMASI VILLA TERMURAH CIATER
(Studi Kasus Simulasi Kemitraan)**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk kelulusan mata kuliah Tugas Akhir
Jenjang Strata Satu (S1)
Pada Program Teknik Informatika**

Oleh :

Jajang Nurjaman

361762009



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER
INDONESIA MANDIRI
BANDUNG
2021**

LEMBAR PENGESAHAN
MEMBANGUN *APPLICATION PROGRAMMING INTERFACE*
SISTEM INFORMASI VILLA TERMURAH CIATER
(Studi Kasus Simulasi Kemitraan)

Oleh
Jajang Nurjaman
361762009

**Skripsi ini telah diterima dan disahkan untuk memenuhi persyaratan
mencapai gelar**

SARJANA TEKNIK INFORMASI

Pada
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER
INDONESIA MANDIRI

Bandung, Juni 2021


Ketua Program Studi,

Chalifa Chazar, S.T., M.T.
NIDN: 0421098704

Dosen Pembimbing,

Chalifa Chazar, S.T., M.T.
NIDN. 0421098704

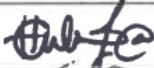
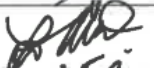
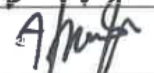
LEMBAR PERSETUJUAN REVISI

**MEMBANGUN *APPLICATION PROGRAMMING INTERFACE*
SISTEM INFORMASI VILLA TERMURAH CIATER
(Studi Kasus Simulasi Kemitraan)**

Oleh
Jajang Nurjaman
361762009

Telah melakukan sidang skripsi dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang skripsi.

**Bandung, Juni 2021
Menyetujui**

No	Nama Dosen	Keterangan	Tanda Tangan
1	Chalifa Chazar, S.T., M.T.	Pembimbing	
2	Yudhi W. Arthana R., S.T., M.Kom.	Penguji 1	
3	Moch. Ali Ramdhani, S.T., M.Kom.	Penguji 2	

**Mengetahui
Ketua Program Studi Sistem Informasi**



Chalifa Chazar, S.T., M.T.
NIDN: 0421098704

SURAT PERNYATAAN


Dengan ini saya menyatakan bahwa:

- 1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- 2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- 3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, Juni 2021
Yang membuat pernyataan




Jajang Nurjaman
361762009

ABSTRAK

MEMBANGUN *APPLICATION PROGRAMMING INTERFACE* SISTEM INFORMASI VILLA TERMURAH CIATER (Studi Kasus Simulasi Kemitraan)

Oleh

Jajang Nurjaman
361762009

Kemudahan dalam mendapatkan dan bertukar informasi adalah salah satu bagian dari perkembangan dan kemajuan teknologi yang dibangun sampai dengan saat ini, terlepas informasi apapun yang akan dibagikan dan kepada siapapun informasi itu akan disajikan, berbagai gaya perancangan atau *architectural style* pertukaran informasi diciptakan untuk membuat pertukaran informasi menjadi sangat mudah, tapi bagaimana cara pertukaran informasi itu mudah dilakukan? Disamping itu dengan adanya kemudahan dalam bertukar informasi tidak terkecuali di dalam berbagai bidang bisnis, di sini khususnya dalam bisnis jasa akomodasi atau penyewaan penginapan, ikut mengambil pemanfaatan kemudahan dalam bertukar informasi dan ikut berupaya mengembangkan usahanya. Sistem informasi villatermurahciater.com adalah salah satu sumber informasi yang menyediakan informasi-informasi penyewaan villa. *Extreme programming* (XP) digunakan untuk pengembangan sistem informasi villatermurahciater.com dari studi kasus simulasi kemitraan. Dengan arsitektur *Service Oriented Architecture* (SOA) adalah suatu gaya arsitektur *software* yang bertumpu pada komunikasi antar *service*, dengan mendefinisikan bagaimana dua bagian atau lebih *service* dapat saling berinteraksi, dimana suatu bagian dapat menjalankan bagian lainnya. API (*Application Programming Interface*) merupakan suatu teknologi antarmuka virtual yang menyediakan data-data dalam hal ini data-data untuk calon mitra Ikatan Pramuwisata Ciater (IPCI). Dengan membuat *web service* atau *web API* menggunakan gaya perancangan *REST API* diharapkan simulasi kemitraan ini dapat diimplementasikan dengan sistem informasi lain secara baik.

Kata kunci: *Service Oriented Architecture* (SOA), *Application Programming Interface* (API), *REST API*, *Web service*

ABSTRACT

BUILDING THE APPLICATION PROGRAMMING INTERFACE INFORMATION SYSTEM OF THE CHEAPEST VILLA CIATER(Case Study Simulation Partnership)

By

Jajang Nurjaman
361762009

The ease of obtaining and exchanging information is one part of technological developments and advancements that have been developed to date, regardless of what information will be shared and whoever the information will be presented, various design styles or architectural styles of information exchange were created to make information exchange a very easy, but how is the exchange of information easy to do? Besides that, the ease of exchanging information is no exception in various fields of business, especially in the accommodation or rental services business, taking advantage of the ease of exchanging information and trying to develop their business. The villatermurahciater.com information system is one source of information that provides information on villa compounding. Extreme programming (XP) was used for the development of the villatermurahciater.com information system from a partnership simulation case study. The Service Oriented Architecture (SOA) is architecture style software that rests on communication between services, by defining how two or more services can interact with each other, where one part can run other parts. API (Application Programming Interface) is a virtual interface technology that provides data, in this case, data for potential IPCI partners. By creating a web service or web API using the design REST API style, it is hoped that this partnership simulation can be implemented well with other information systems.

Keywords: *Service Oriented Architecture (SOA), Application Programming Interface (API), REST API, Web service.*

KATA PENGANTAR

Segala puji dan syukur semoga selalu terlimpah kehadiran Allah SWT., karena berkat karunia dan rahmat-Nya penulis dapat menyelesaikan skripsi dengan judul “Membangun *Application Programming Interface* Sistem Informasi Villa Termurah Ciater (Studi Kasus Simulasi Kemitraan)”. Penyusunan skripsi ini bertujuan untuk memenuhi persyaratan kelulusan pada Program Studi Sistem Informasi di Sekolah Tinggi Manajemen Informatika & Komputer Indonesia Mandiri (STMIK-IM Bandung).

Selama proses penyusunan dan penyelesaian skripsi ini penulis pun menghadapi beberapa kesulitan baik dari sisi kemampuan, waktu dan situasi yang dimiliki. Dikarenakan demikian penulis sungguh mengapresiasi dan sangat berterima kasih kepada semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini, baik kepada pihak-pihak yang telah menyediakan referensi jurnal, artikel terkait materi dibahas dan referensi dalam pemrograman.

Penyusunan skripsi ini telah dibuat dengan sebaik-baiknya, namun tentu saja masih jauh dari kata sempurna. Oleh karena itu penulis sangat menghargai jika mungkin ada sedikit saran maupun kritik apapun yang sifatnya membangun tentunya bagi penulis yang guna pembelajaran di kemudian hari.

Penulis pun berharap, semoga skripsi ini dapat memberikan manfaat dan wawasan yang lebih khususnya bagi penulis sendiri dan masyarakat pada umumnya.

Bandung, Juni 2021

Jajang Nurjaman
361762009

UCAPAN TERIMA KASIH

Pada kesempatan ini, penulis menyampaikan rasa terima kasih kepada rekan-rekan, pembimbing serta pihak yang ikut serta membantu penulis dalam penyusunan skripsi ini, antara lain:

1. Ibu Chalifa Chazar, S.T., M.T. selaku selaku Ketua Program Studi Teknik Informasi sekaligus pembimbing yang telah memberikan penulis ilmu dan masukan dalam penyusunan skripsi.
2. Ikatan Pramuwisata Ciater (IPCI) yang telah memberikan kesempatan penulis untuk melakukan kegiatan penelitian dan membuat program penyediaan layanan data guna mendukung kelancaran penyusunan skripsi.
3. Bapak Dr. Chairuddin, Ir., M.M., M.T. selaku Ketua STMIK-IM Bandung yang telah memberikan ilmu kepada penulis.
4. Kedua orang tua penulis, Emak dan Bapak yang selalu memberikan bantuan dan dukungan kepada penulis.
5. Bapak & Ibu dosen yang telah memberikan penulis ilmu selama penulis melakukan kegiatan pembelajaran di STMIK-IM Bandung.
6. Seluruh rekan dan sahabat di STMIK-IM Bandung.
7. Kepala dan seluruh staff Administrasi, BAAK, BAUK dan staff Perpustakaan STMIK-IM Bandung.
8. Teh Wiwin Widiawati yang memberikan dukungan dan dorongan doa.

9. Seluruh pihak yang tidak mungkin dapat penulis sebutkan satu demi satu.

Always do your best and don't forget to be happy, because time can't turned back. Besar sekali harapan penulis skripsi ini dapat bermanfaat bagi pembaca sekalian.

Bandung, Juni 2021

Jajang Nurjaman
361762009

DAFTAR ISI

SURAT PERNYATAAN.....	i
ABSTRAK	ii
<i>ABSTRACT</i>	iii
KATA PENGANTAR	iv
UCAPAN TERIMA KASIH.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	5
1.3 Tujuan Penelitian.....	5
1.4 Batasan Masalah.....	6
1.5 Metode Penelitian.....	6
1.5.1 Pengumpulan Data	6
1.5.2 Metode Pengembangan Sistem	7
1.6 Sistematika Penulisan.....	9
BAB II LANDASAN TEORI.....	12
2.1 <i>Application Programming Interface</i>	12
2.2 Sistem Terdistribusi.....	12
2.3 <i>REpresentational State Transfer</i>	13

2.4	<i>RESTful Application Programming Interface</i>	13
2.4.1	<i>Uniform Interface</i>	14
2.4.2	<i>Stateless</i>	15
2.4.3	<i>Chaceable</i>	15
2.4.4	<i>Client Server</i>	16
2.4.5	<i>Layered System</i>	16
2.4.6	<i>Code on Demand</i>	17
2.5	Status Kode.....	18
2.6	<i>Service Oriented Architecture (SOA)</i>	19
2.7	<i>Cloud Computing</i>	20
2.8	Portal Informasi	21
2.9	<i>Unifeid Modeling Languagege (UML)</i>	22
2.9.1	Flowchart Diagram.....	22
2.9.2	Use Case Diagram.....	23
2.9.3	Activity Diagram.....	24
2.9.4	Class Diagram	25
2.9.5	Sequence Diagram	27
2.9.6	Deployment Diagram	28
2.9.7	Entity Relational Diagram (ERD).....	29
2.10	<i>Extreme Programming</i>	31
2.11	<i>JSON</i>	31
2.12	<i>Postman</i>	31
2.13	<i>Black Box Testing</i>	32
BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM		34
3.1	Metode Pengumpulan Data	34
3.1.1	Observasi.....	34
3.1.2	Studi Pustaka.....	39
3.2	Metode Pengembangan Sistem.....	42

3.2.1	Perencanaan.....	43
3.2.2	Perancangan	45
3.3	Perancangan Antarmuka.....	90
BAB IV IMPLEMENTASI DAN UJI COBA		100
4.1	Implementasi	100
4.1.1	Implementasi Perangkat Keras.....	100
4.1.2	Implementasi Perangkat Lunak	101
4.1.3	Implementasi <i>Database</i>	101
4.1.4	Implementasi Antarmuka	104
4.2	Pengujian Sistem	111
4.2.1	Pengujian Tampilan <i>Dashboard</i> Halaman Utama	111
4.2.2	Pengujian Sistem <i>Mitra User</i>	113
4.2.3	Pengujian <i>Login Mitra User</i>	114
4.2.4	Pengujian <i>Sign up Client User</i>	115
4.2.5	Pengujian Halaman <i>Dashboard Mitra User</i>	116
4.2.6	Pengujian <i>Upload Data Vila Mitra User</i>	117
4.2.7	Pengujian <i>Edit Data Vila Mitra User</i>	118
4.2.8	Pengujian Menu <i>Logout Mitra User</i>	119
4.2.9	Pengujian <i>Login Admin User</i>	119
4.2.10	Pengujian Halaman <i>Dashboard Admin User</i>	120
4.2.11	Pengujian Halaman Menu <i>Approval Untuk User</i>	121
4.2.12	Pengujian Halaman Menu <i>Approval Untuk Client</i>	122
4.2.13	Pengujian Halaman <i>Post Data Vila</i>	122
4.2.14	Pengujian Halaman <i>Put Data Vila</i>	123
4.2.15	Pengujian Halaman <i>Form Put Data Vila</i>	124
4.2.16	Pengujian Halaman <i>Delete Data Vila</i>	125
4.2.17	Pengujian Menu <i>Logout Admin User</i>	125
BAB V PENUTUP.....		126

5.1	Kesimpulan.....	126
5.2	Saran.....	126
	DAFTAR PUSTAKA	127
	LAMPIRAN.....	130
	Lampiran 1.....	130
	Lampiran 2.....	132

DAFTAR TABEL

Tabel: 2. 1 <i>Use case diagram</i> (Hendini, 2016)	24
Tabel: 2. 2 <i>Activity diagram</i> (Hendini, 2016)	25
Tabel: 2. 3 <i>Class diagram</i> (Hendini, 2016).....	26
Tabel: 2. 4 <i>Sequence diagram</i> (Hendini, 2016)	27
Tabel: 2. 5 <i>Deployment diagram</i> (Hendini, 2016)	29
Tabel: 2. 6 <i>Entity Relationship Diagram</i> (Putri, 2020).....	30
Tabel: 3. 1 Data vila dari sistem informasi villatermurahciater.com.....	35
Tabel: 3. 2 Refrensi Jurnal Terkait <i>SOA, API, Rest dan Restful</i>	40
Tabel: 3. 3 Penjadwalan pengembangan sistem informasi villatermurahciater.com	43
Tabel: 3. 4 Analisis kebutuhan data	43
Tabel: 3. 5 Perangkat keras pengembang.....	44
Tabel: 3. 6 Spesifikasi kebutuhan perangkat lunak.....	45
Tabel: 3. 7 Identifikasi aktor	46
Tabel: 3. 8 Identifikasi <i>use case</i> mitra user.....	47
Tabel: 3. 9 Identifikasi <i>use case client user</i>	47
Tabel: 3. 10 Identifikasi <i>use case admin user</i>	48
Tabel: 3. 11 Skenario <i>use case diagram sign up</i> mitra user	51
Tabel: 3. 12 Skenario <i>use case diagram login</i> mitra user.....	52
Tabel 3. 13 Skenario <i>use case diagram upload data vila</i> mitra user.....	53
Tabel 3. 14 Skenario <i>use case diagram edit data vila</i> mitra user	53
Tabel 3. 15 Skenario <i>use case diagram view data vila</i> mitra user.....	54
Tabel 3. 16 Skenario <i>use case diagram login admin user</i>	54
Tabel: 3. 17 Skenario <i>use case diagram admin user approval data</i> mitra user...	55

Tabel 3. 18 Skenario <i>use case diagram</i> admin user approval data client user....	55
Tabel: 3. 19 Skenario <i>use case diagram</i> admin get data vila.....	56
Tabel : 3. 20 Skenario <i>Use Case Diagram</i> Admin User Put Data Vila.....	56
Tabel: 3.21 Skenario <i>use case diagram</i> admin user post data vila	57
Tabel: 3. 22 Skenario <i>use case diagram</i> admin user delete data vila	57
Tabel: 3. 23 Skenario <i>use case diagram</i> logout admin user dan mitra user	58
Tabel: 3. 24 <i>Collaboration Responssibility Card</i> (CRC).....	87
Tabel: 4. 1 Menguji tampilan <i>dashboard</i> halaman utama.....	112
Tabel: 4. 2 Pengujian <i>sign up</i> mitra user.....	113
Tabel: 4. 3 Pengujian <i>login</i> mitra user	114
Tabel: 4. 4 Menguji <i>sign up</i> mitra client user	115
Tabel: 4. 5 Pengujian halaman <i>dashboard</i> mitra user.....	116
Tabel: 4. 6 Pengujian <i>upload</i> data vila mitra user	117
Tabel: 4. 7 Pengujian <i>edit</i> data vila mitra user.....	118
Tabel: 4. 8 Pengujian menu <i>logout</i> mitra user	119
Tabel: 4. 9 Pengujian <i>login</i> admin user	119
Tabel: 4. 10 Pengujian halaman <i>dashboard</i> admin user.....	120
Tabel: 4. 11 Pengujian halaman menu <i>aapproval</i> untuk user (mitra user).....	121
Tabel: 4. 12 Pengujian halaman menu <i>approval</i> untuk client (client user)	122
Tabel: 4. 13 Pengujian halaman <i>post</i> data vila.....	122
Tabel: 4. 14 Pengujian halaman <i>put</i> data vila	123
Tabel: 4. 15 Pengujian halaman <i>form put</i> data vila.....	124
Tabel: 4. 16 Pengujian halaman <i>delete</i> data vila.....	125
Tabel: 4. 17 Pengujian menu <i>logout</i> admin user.....	125

DAFTAR GAMBAR

Gambar: 1. 1 Dasar komunikasi dalam <i>Service Oriented Architecture</i> (Alanazi dkk., 2019)	4
Gambar: 1. 2 Kerangka Kerja Extreme Programming (Sontana dkk, 2019:3)	8
Gambar: 2. 1 Simbol-simbol <i>Flowchart</i> (Ridlo, 2017).....	23
Gambar: 2. 2 Contoh gambar <i>class diagram</i> (Sontana dkk, 2019).....	26
Gambar: 3. 1 <i>Flowchart</i> pengembangan sistem informasi villatermurahciater.com	38
Gambar: 3. 2 <i>Use case diagram</i> mitra user dan client user	49
Gambar: 3. 3 <i>Use case diagram</i> admin user	50
Gambar: 3. 4 <i>Activity diagram</i> mitra user dan clientuser halaman utama.....	59
Gambar: 3. 5 <i>Activity Diagram Sign up Client User</i>	59
Gambar: 3. 6 <i>Activity diagram sign up</i> mitra user	60
Gambar: 3. 7 <i>Activity diagram login</i> mitra user	60
Gambar: 3. 8 <i>Activity diagram upload</i> data vila oleh mitra user	61
Gambar: 3. 9 <i>Activy diagram edit</i> data vila oleh mitra user	61
Gambar: 3. 10 <i>Activy diagram view</i> data oleh mitra user.....	62
Gambar: 3. 11 <i>Activity diagram logout</i> mitra user dan client user.....	62
Gambar: 3. 12 <i>Activity diagram login</i> admin user.....	63
Gambar: 3. 13 <i>Activity diagram admin view</i> data <i>approval</i> untuk mitra user	63
Gambar: 3. 14 <i>Activity diagram admin view</i> data <i>approval</i> untuk client user.....	64
Gambar: 3. 15 <i>Activity diagram get</i> data dari sistem informasi vila termurah ciater oleh admin user	64
Gambar: 3. 16 <i>Acivity diagram admin user put</i> data vila	65
Gambar: 3. 17 <i>Activity diagram admin user post</i> data vila.....	65
Gambar: 3. 18 <i>Activity diagram admin user delete</i> data vila.....	66
Gambar: 3. 19 <i>Activity diagram admin user logout</i>	66

Gambar: 3. 20 <i>Sequence diagram</i> mitra user dan <i>client user</i> halman utama.....	68
Gambar: 3. 21 <i>Sequence diagram</i> sign up mitra user	69
Gambar: 3. 22 <i>Sequence diagram</i> login mitra user.....	70
Gambar: 3. 23 <i>Sequence diagram</i> mitra user upload data vila	71
Gambar: 3. 24 <i>Sequence diagram</i> mitra user edit data vila	72
Gambar: 3. 25 <i>Sequence diagram</i> mitra user view data vila	73
Gambar: 3. 26 <i>Sequence diagram</i> logout mitra user.....	74
Gambar: 3. 27 <i>Sequence diagram</i> login admin user	75
Gambar: 3. 28 <i>Sequence diagram</i> admin view data approval untuk mitra user ...	76
Gambar: 3. 29 <i>Sequence diagram</i> admin view data approval untuk client user...	77
Gambar: 3. 30 <i>Sequence</i> get data vila dan mitra user	78
Gambar: 3. 31 <i>Sequence diagram</i> admin user put data vila	79
Gambar: 3. 32 <i>Sequence diagram</i> post data vila.....	80
Gambar: 3. 33 <i>Activity diagram</i> delete data vila.....	81
Gambar: 3. 34 <i>Sequence diagram</i> logout admin user	82
Gambar: 3. 35 <i>Sequence diagram</i> sign up client user.....	83
Gambar: 3. 36 <i>Class diagram</i> sistem informasi villatermurahciater.com	85
Gambar: 3. 37 <i>Class diagram</i> sistem admin villatermurahciater.com.....	86
Gambar: 3.38 <i>Deployment diagram</i> sistem informasi villatermurahciater.com, Admin user, Mitra user, Client user	87
Gambar: 3. 39 <i>Entity Relational Diagram</i> villatermurahciater.com.....	89
Gambar: 3. 40 Halaman utama sistem informasi villatermurahciater.com.....	90
Gambar: 3. 41 Halaman sign up mitra user	91
Gambar: 3. 42 Halaman login mitra user.....	91
Gambar: 3. 43 Halaman dashboard mitra user	92
Gambar: 3. 44 Halaman menu upload vila mitra user	92
Gambar: 3. 45 Halaman menu edit vila mitra user	93
Gambar: 3. 46 Halaman menu form edit vila mitra user.....	93

Gambar: 3. 47 Halaman menu <i>view</i> vila mitra <i>user</i>	94
Gambar: 3. 48 Halaman <i>login</i> admin <i>user</i>	94
Gambar: 3. 49 Halaman <i>dashboard</i> admin <i>user</i>	95
Gambar: 3. 50 Halaman admin aktivasi akun mitra <i>user</i>	95
Gambar: 3. 51 Halaman admin <i>approval</i> akun <i>client user</i>	96
Gambar: 3. 52 Halaman admin <i>get</i> data vila.....	96
Gambar: 3. 53 Halaman <i>put</i> data vila.....	97
Gambar: 3. 54 Halaman admin <i>form put</i> data vila	97
Gambar: 3. 55 Halaman admin <i>post</i> data vila	98
Gambar: 3. 56 Halaman admin <i>delete</i> data vila	98
Gambar: 3. 57 Halaman <i>logout</i> admin	99
Gambar: 3. 58 Halaman <i>sign up client user</i>	99
Gambar: 4. 1 Struktur utama <i>database</i> sistem informasi villatermurahciater.com	102
Gambar: 4. 2 Struktur <i>database</i> admin <i>user</i>	102
Gambar: 4. 3 Struktur <i>database</i> vila	103
Gambar: 4. 4 Struktur tabel <i>client user</i>	103
Gambar: 4. 5 Struktur tabel aktivasi	103
Gambar: 4. 6 Struktur <i>database client user</i>	104
Gambar: 4. 7 Implementasi relasi antar tabel.....	104
Gambar: 4. 8 Antarmuka halaman utama	105
Gambar: 4. 9 Antarmuka <i>sign up</i> mitra <i>user</i>	105
Gambar: 4. 10 Antarmuka <i>login</i> mitra <i>user</i>	105
Gambar: 4. 11 Antarmuka <i>dashboard</i> mitra <i>user</i>	106
Gambar: 4. 12 Antarmuka <i>upload</i> data vila mitra <i>user</i>	106
Gambar: 4. 13 Antarmuka halaman <i>edit list</i> data vila mitra <i>user</i>	106
Gambar: 4. 14 Antarmuka halaman <i>edit</i> data vila mitra <i>user</i>	107
Gambar: 4. 15 Antarmuka halaman <i>view</i> data vila mitra <i>user</i>	107

Gambar: 4. 16 Antarmuka halaman <i>logout</i> mitra <i>user</i>	107
Gambar: 4. 17 Antarmuka halaman admin	108
Gambar: 4. 18 Antarmuka halaman <i>dashboard</i> admin	108
Gambar: 4. 19 Antarmuka halaman admin <i>approve</i> mitra <i>user</i>	108
Gambar: 4. 20 Antarmuka halaman admin <i>get</i> data vila.....	109
Gambar: 4. 21 Antarmuka halaman admin <i>post</i> data vila.....	109
Gambar: 4. 22 Antarmuka halaman admin <i>put</i> data vila	109
Gambar: 4. 23 Antarmuka halaman admin <i>form put</i> data vila.....	110
Gambar: 4. 24 Antarmuka halaman admin <i>delete</i> data vila.....	110
Gambar: 4. 25 Antarmuka halaman <i>logout</i> admin <i>user</i>	110
Gambar: 4. 26 Antarmuka halaman <i>sign up client user</i>	111
Lampiran Gambar:1 Daftar vila, harga, fasilitas dan <i>sales quantity</i>	130
Lampiran Gambar:2 Wawancara tentang IPCI	131
Lampiran Gambar: 3 Wawancara penjualan vila tahun 2019 tahun 2020.....	131
Lampiran Gambar: 4 Wawancara pola pemesana vila ciater	132

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemudahan dalam mendapatkan dan bertukar informasi adalah salah satu bagian dari perkembangan dan kemajuan teknologi yang dibangun sampai dengan saat ini, berbagai *architectural style* atau gaya perancangan pertukaran informasi dan data dibangun untuk mempermudah pertukaran sebuah data ke dalam aplikasi atau sistem informasi lainnya untuk kepentingan perkembangan sebuah teknologi, bisnis, instansi, organisasi dan lain sebagainya. Akan tetapi bagaimana cara pertukaran informasi itu mudah dilakukan?

Penyajian informasi yang efektif dan tepat guna akan menjadikan performa sebuah instansi atau organisasi semakin berkualitas. Untuk menghasilkan informasi tersebut maka dibutuhkan sebuah sistem informasi yang mendukung dalam pengolahan datanya (Wijonarko dan Wahyu, 2018). Pembangan antarmuka pemograman aplikasi adalah salah satu cara agar suatu sistem informasi dapat diakses serta dimanfaatkan oleh pihak lain tanpa merubah kode utama sistem dan basis data sistem serta memudahkan komunikasi antar sistem meskipun berbeda platform (Wijonarko dan Wahyu , 2018).

Disamping adanya kemudahan dalam bertukar informasi, bisnis jasa penyewaan penginapan ikut mengambil pemanfaatan kemudahan dalam bertukar informasi dan ikut berupaya mengembangkan usahanya.

Sari Ater Hot Spring Resort atau dikenal dengan obyek wisata air panas Ciater, terletak pada kawasan pegunungan Subang, dikaki gunung tangkuban perahu, tepatnya di Desa Ciater, Kecamatan Ciater Subang, obyek wisata ini merupakan salah satu obyek terpopuler di Indonesia.

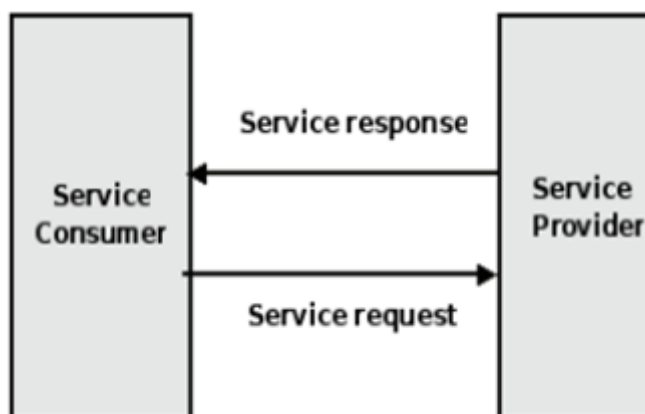
IPCI atau Ikatan Pramuwisata Ciater adalah kelompok masyarakat yang mengelola dalam jasa pemasaran vila-vila yang disewakan kepada pengunjung di tempat wisata Sari Ater yang berdiri sejak tahun 1987. Dalam pengelolaan vila-vilanya IPCI telah membuat sistem informasi yang berbasis web untuk sumber informasi vila-vila dan pemasarannya.

IPCI sekaligus pemilik sistem informasi villatermurahciater.com akan mengembangkan usaha dan sistem informasinya dengan bermitra bersama perusahaan jasa lainnya. Menyediakan data-data untuk calon mitra yang nantinya bisa digunakan oleh calon mitra bisnisnya, seperti menampilkan untuk masing-masing harga sewa vila perhari, fasilitas vila, kapasitas vila untuk menampung orang dan gambar vila. Sehingga diperlukan suatu teknologi untuk menjembatani data dari sistem informasi villatermurahciater.com kepada sistem informasi calon mitra bisnis IPCI.

API (Application Programming Interface) terdiri dari dokumentasi yang terdiri dari antarmuka, fungsi, kelas, struktur, dan sebagainya untuk membangun perangkat lunak. Dengan *API* ini, kemudian membuat programmer untuk "membongkar" perangkat lunak untuk kemudian dikembangkan atau diintegrasikan dengan perangkat lunak lain (Sontana, Rahmatulloh dan Rachman, 2019).

“*SOA is basically a group of services, where the communication involves either two or more services regarding some activity or it can involve simple data passing*” (Alanazi dkk., 2019). *SOA (Service Oriented Architecture)* merupakan arsitektur pendekatan dalam sistem komunikasi antara 2 aplikasi atau lebih, dimana data yang telah disediakan oleh layanan *web service* dapat digunakan oleh *rest client* atau suatu sistem dan aplikasi lainnya.

Dasar dari komunikasi dalam *SOA* adalah *service request message* atau pesan layanan permintaan yang dikirim dari *service customer* ke penyedia layanan atau *service provider* dan lalu *service provider* harus mengirimkan atau mengembalikan pesan respon kepada *service customer* atau penerima layanan. “*The connections and subsequent response with its request are defined in some understandable way to both service provider and the service consumer.*” (Alanazi dkk, 2019).



Gambar: 1. 1 Dasar komunikasi dalam *Service Oriented Architecture* (Alanazi dkk., 2019)

Menggunakan gaya perancangan *RESTful* merupakan gaya pengembangan *web service* yang memanfaatkan *HTTP (Hyper Text Transfer Protocol)*, dalam *service request data* atau layanan permintaan datanya menggunakan kata kerja atau *method* yaitu *GET* untuk *request client server* akan secara otomatis direspon oleh *web server* dan *method* untuk *web service* yaitu *GET, POST, PUT* dan *DELETE* dalam pengelolaan data-datanya.

Metode *REST* memiliki keunggulan lebih cepat dan efisien dibandingkan metode *SOAP*, selain itu, dalam implementasinya metode *REST* memiliki konsumsi sumberdaya lebih efisien (Wijornako dan Wahyu, 2018).

Oleh karena itu dari paparan diatas untuk studi kasus simulasi kemitraan dalam hal ini mengintegrasikan sistem informasi villatermurahciater.com

diperlukan pembuatan *API*. Dengan membuat *web service* atau *web API* menggunakan gaya perancangan *Representational State Transfer (REST)* dan *Representational State Transfer Application Programming Interface (RESTful API)* diharapkan dapat menyediakan data-data diatas untuk calon mitra (*rest client*) dan pengelolaan data untuk *villatermurahciater.com* (*rest server*).

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka identifikasi masalah adalah bagaimana mengintegrasikan sistem informasi *villatermurahciater.com* dengan sistem informasi calon mitra dan sekaligus menyediakan layanan data untuk calon mitra?

1.3 Tujuan Penelitian

Membangun *Application Programming Interface* melalui pendekatan *Service Oriented Architectur* dengan gaya arsitektur *REST API* sehingga diharapkan sistem informasi *villatermurahciater.com* dengan sistem informasi calon mitra dapat terintegrasi dan sekaligus dapat menyediakan layanan data untuk calon mitra.

1.4 Batasan Masalah

Batasan masalah dalam penelitian yang akan dilakukan adalah sebagai berikut :

1. *API* yang dirancang adalah untuk menarik data vila hanya dari *database* sistem informasi villatermurahciater.com.
2. Masalah keamanan atau *security* tidak dibahas di dalam penelitian.
3. *API* yang dibangun adalah berbasis *PHP* dan *JavaScript*.
4. *API* menerima *request* dan *response* dalam bentuk *JSON*.
5. Untuk keperluan *demo* aplikasi, akan digunakan aplikasi *online* yang tersedia gratis bernama Postman.

1.5 Metode Penelitian

Adapun metode-metode yang digunakan dalam penelitian ini sebagai berikut:

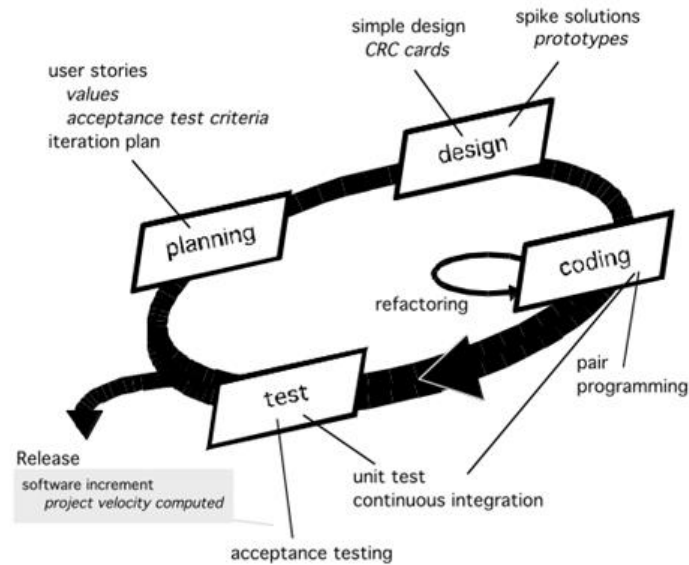
1.5.1 Pengumpulan Data

1. Kerangka penelitian dimulai dengan metode pengumpulan data dengan menanyakan data-data apa saja yang akan dibuatkan *API* untuk nantinya dibagikan kepada calon mitra.
2. Studi litelatur yaitu mengumpulkan bahan-bahan materi dari berbagai sumber misalnya jurnal, artikel, internet, dokumen penelitian-penelitian terdahulu dan lain sebagainya.

1.5.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang akan digunakan adalah *Extreme Programming (XP)*. *This model is used for developing software in an flexible environment. Due to its flexible nature the cost of change in requirements during the later development phase decreases. XP provides a simple, precise and seemingly native principles and values to guide the software development process in four main stages of software development: planning, coding, design and testing* (Singh, 2017).

Model ini digunakan untuk mengembangkan perangkat lunak dalam lingkungan yang fleksibel. Karena sifatnya yang fleksibel, biaya perubahan persyaratan selama fase pengembangan selanjutnya menurun. XP menyediakan prinsip dan nilai yang sederhana, tepat dan tampak asli untuk memandu proses pengembangan perangkat lunak dalam empat tahap utama pengembangan perangkat lunak: perencanaan, pengkodean, desain, dan pengujian



Gambar: 1. 2 Kerangka Kerja *Extreme Programming* (Sontana dkk, 2019:3)

1. *Planing* (Perencanaan)

Penjadwalan pengembangan sistem informasi, dimana dibuatkan *time table* mulai dan berakhirnya pengembangan sistem informasi atau suatu proyek.

Analisis data merupakan kebutuhan yang berdasarkan pada fungsionalitas seperti *user name* dan *password* untuk *login* yang nantinya memiliki hak akses yang dapat mengelola fitur-fitur yang ada di dalam *web service* yaitu *GET*, *POST*, *PUT* dan *DELETE*.

Spesifikasi minimum perangkat keras yang dibutuhkan untuk menjalankan aplikasi berbasis *web*, yaitu *PC* yang terhubung dengan jaringan internet. Selbihnya aplikasi ini tidak membutuhkan antarmuka perangkat keras yang spesifik.

2. *Design* (Perancangan)

Pada tahap ini memberikan identifikasi aktor, skenario, *flowchart diagram*, *use case diagram*, *class diagram* dan *CRC (class-responsibility-collaborator)*.

3. *Coding* (Pengkodean)

Pada tahap ini pengerjaan perancangan awal dilakukan, dengan tidak merubah *code* yang sudah ada melainkan mengembangkan serangkaian pengujian unit yang telah dibuat yang nantinya akan dimasukkan ke dalam tahap rilis.

4. *Test* (Pengujian)

Pada tahap ini pengujian unit dari hasil pengkodean apakah sudah sesuai dengan *output* perancangan yang telah dibuat dan dari fungsionalitas apakah sudah sesuai dengan fungsinya. Hal ini dilakukan untuk meminimalisir adanya kesalahan (*error*).

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini mempunyai beberapa sub bab dengan uraian sebagai berikut:

BAB I Pendahuluan

Bab ini merupakan bagian yang menjelaskan latar belakang masalah, identifikasi masalah, tujuan dari penelitian, batasan masalah, metode penelitian, sistematika penulisan, serta lokasi penelitian.

BAB II Landasan Teori

Bab ini menguraikan tentang teori yang didapat dari artikel, jurnal dan penelitian sebelumnya yang menyangkut dengan topik yang sedang dibuat untuk menunjang penulisan skripsi.

BAB III Analisa Masalah dan Perancangan Program

Bab ini berisi tentang proses perancangan program berdasarkan analisa permasalahan dengan metode penelitian dan metode pengembangan sistem yang digunakan.

BAB IV Implementasi dan Uji Coba

Berisi tentang penjelasan dalam perancangan dan pengoperasian sistem informasi secara bertahap, hasil dan evaluasi sistem informasi termasuk kelebihan dan kekurangannya.

BAB V Penutup

Pada bab ini menjabarkan hasil dari penelitian yang dilakukan. Berupa saran yang bertujuan untuk peningkatan topik dalam skripsi di masa yang akan datang serta gagasan yang dapat dari hasil penelitian.

BAB II

LANDASAN TEORI

2.1 *Application Programming Interface*

API atau *Application Programming Interface* terdiri dari dokumentasi yang terdiri dari antarmuka, fungsi, kelas, struktur, dan sebagainya untuk membangun perangkat lunak (Sontana dkk, 2019), merupakan suatu teknologi antarmuka virtual yang memungkinkan untuk bertukar informasi atau data antar aplikasi atau sistem (Sontana dkk, 2019). Dengan adanya teknologi ini maka penyediaan data dan proses pertukaran data atau informasi akan mudah untuk dilakukan antar aplikasi atau antar sistem informasi. Pengembangan aplikasi atau sistem informasi tentunya akan cepat berkembang, menjadi stimulus untuk berinovasi terhadap perkembangan pangsa pasar bisnis khususnya bisnis pelayanan dan jasa.

2.2 Sistem Terdistribusi

Sistem Terdistribusi adalah gabungan dari dua pendekatan pengolahan data yaitu *database* dan jaringan komputer (Mahendra dkk, 2016), yang memungkinkan komputer untuk mengkoordinasikan kegiatan-kegiatan mereka dan untuk berbagi sumber daya sistem. Pengguna sebuah sistem terdistribusi yang memandang sistem sebagai fasilitas komputasi tunggal terintegrasi.

Salah satu gaya perancangan di dalam sistem terdistribusi adalah *REST* (*REpresentational State Transfer*) yang mana gaya perancangan ini mendukung untuk pertukaran data dan informasi kedalam aplikasi atau sistem informasi yang berbeda yang disediakan oleh *web service*.

2.3 REpresentational State Transfer

REST adalah “*Architectural Styles and the Design of Network-based Software Architecture*”(Patni, 2017). “*an architectural style for distributed hypermedia systems. Put simply, REST (short for REpresentational State Transfer) is an architectural style defined to help create and organize istributed systems. The key word from that definition should be “style”*”(Patni, 2017).

Gaya arsitektur dan desain arsitektur perangkat lunak yang berbasis jaringan dimana sebuah gaya arsitektur untuk sistem hypermedia terdistribusi. sederhananya, *REpresentational State Transfer* gaya arsitektur yang didefinisikan untuk membantu membuat dan mengatur sistem terdistribusi, akan tetapi bukan berarti menjadi pedoman atau standar yang baku yang harus diikuti ketika membuat dan mengatur sistem terdistribusi.

2.4 RESTful Application Programming Interface

“*REST architectural style is gaining popularity but there is lot of debate and growing concern about modeling of REST web services (REST API)*

(Surwase, 2016). *Were originally communicated by Roy Fielding in his doctoral dissertation and define the basis of RESTful-style*” (Surwase, 2016).

Gaya arsitektur *REST* ini berfokus kepada perkembangan tentang pemodelan layanan *web* yaitu *REST (REST API)*, yang awalnya dipublikasikan oleh Roy Fielding dalam disertasi doktoralnya dan mendefinisikan dasar gaya *RESTful*. Ada enam aturan dalam membuat *RESTful API* yaitu “*Uniform Interface, Stateless, Cacheable, Client-Server, Layered System, Code on Demand*” (Surwase, 2016).

2.4.1 Uniform Interface

“It defines the interface between service consumer and service provider. It simplifies and decouples the architecture, which enables independence of each part. Resource-Based, Individual resources are identified in requests using URIs as resource identifiers”(Surwase, 2016).

Mendefinisikan antarmuka antara klien (*client*) dan penyedia layanan (*server*) dalam menyederhanakan bagian antarmuka dan memisahkan arsitektur *rest client*, yang memungkinkan independensi terhadap setiap bagian, *Resource-Based*, sumber daya secara individual diidentifikasi dalam sebuah permintaan melalui *URI* .

2.4.2 *Stateless*

“Stateless As REST deals with state transfer, statelessness is power present in REST. It implies is that the state to control the request is resides inside the request itself, whether as part of the URI, request parameters, body, or headers”(Surwase, 2016).

Stateless adalah dimana permintaan *client* kepada *server* secara terisolasi selama proses permintaan parameter itu terpenuhi, *body* atau *header* yang dikirimkan ke *URI (Uniform Resource Identifier)*.

2.4.3 *Chaceable*

“Cacheable As on the world large web, clients can cache responses. Responses have to hence, implicitly or explicitly, outline themselves as cacheable, or no longer, to avert customers reusing stale or inappropriate data in line with extra requests. Goodmanaged caching partly or thoroughly eliminates some purchaser– server interactions, extra bettering scalability and efficiency” (Surwase, 2016).

Client dapat menyimpan respon kedalam *cache*, oleh karena itu, respon secara implisit atau eksplisit dapat menguraikan diri agar disimpan kedalam *cache*, atau juga tidak disimpan agar mencegah *client* menggunakan kembali data lama atau tidak sesuai dengan permintaan lainnya. Dapat dikelola oleh *client*

sepenuhnya dengan caching dihapus seluruhnya atau beberapa interaksi pada *client-server* dan mempunyai skalabilitas lebih baik dan efisien.

2.4.4 Client Server

“Consumer–server The uniform interface separates customers from servers. This separation of considerations implies that, for instance, purchasers aren't concerned with knowledge storage, which stays inner to every server, in order that the portability of client code is improved” (Surwase, 2016).

Pada model *client server*, proses sistem terdistribusi dibagi ke dalam dua kelompok. *server* merupakan proses implementasi dari sebuah layanan tertentu, contohnya sebuah layanan sistem *file* atau layanan basis data. *Client* merupakan proses meminta layanan dari *server* dengan cara mengirim permintaan yang kemudian menunggu balasan dari *server*. Proses interaksi *client-server* disebut sebagai *request-reply behavior*.

2.4.5 Layered System

“Layered procedure API consumer cannot traditionally tell whether or not it is hooked up directly to the top server, or to an middleman along the way. Middleman servers could improve procedure scalability via enabling load-balancing and through delivering shared caches. Layers may additionally implement protection policies” (Surwase, 2016).

Agar berfungsi secara efektif *web API* harus memenuhi beberapa persyaratan fungsional dan non-fungsional. Persyaratan ini, yang pada dasarnya adalah kualitas perangkat lunak yang harus menampilkan antarmuka, bayak variasi *layer* yang digunakan tergantung besar atau kecilnya suatu aplikasi atau sistem informasi.

2.4.6 Code on Demand

“Code on demand (optional) Servers are ready to temporarily prolong or customise the performance of a consumer by way of transferring good judgment to it that it could actually execute. Examples of this may incorporate compiled add-ons equivalent to Java applets and patron-side scripts such as JavaScript” (Surwase, 2016).

Kode berdasarkan permintaan, untuk kode yang dikirimkan oleh klien ke *server* atau sebaliknya maka kode tersebut dapat dibaca dan dieksekusi berdasarkan permintaan dari perangkat lunak klien.

RESTful Sebuah metode yang dirancang mengikuti gaya arsitektur *REST* yang menekankan skalabilitas (Putra dkk, 2020), dengan memanfaatkan metode *HTTP (HyperText Transfer Protocol)* yang dapat di panggil oleh *client* dimana kata kerja yang akan digunakan yaitu *GET, POST, PUT* dan *Delete*.

1. *GET Method* ini digunakan untuk mendapatkan data dari *server*.

2. *POST Method* ini digunakan untuk mengirimkan data ke *server*.
3. *PUT Method* ini digunakan untuk meng-*update* data ke *server*.
4. *DELETE Method* ini digunakan untuk menghapus data ke *server*.

Dengan proses pertukaran data ini dilakukan secara “terisolasi” selama persyaratan yang dibutuhkan pada *server* terpenuhi yaitu autentikasi, *key*, parameter dan lain-lain maka proses pertukaran data tersebut akan berjalan dengan baik.

2.5 Status Kode

Status kode dalam *REST API* yang akan memberikan informasi untuk *client* dan *server* dimana kode status tersebut masing-masing mengidentifikasi *request* dan *response* pada *HTTP*, berikut kode status pada *HTTP* :

1. 200 – *OK*
2. 400 – *Bad Request*
3. 500 – *Internal Server Error*
4. 201 – *Created*
5. 304 – *Not Modified*
6. 404 – *Not Found*
7. 401 – *Unauthorized*
8. 403 – *Forbidden*

2.6 Service Oriented Architecture (SOA)

“Service Oriented Architecture (SOA) is an architectural style that supports service orientation based on a way of thinking in terms that are specific to services and development focused on services and the results of their implementation” (Pirna, 2016).

Gaya arsitektur yang mendukung orientasi layanan berdasarkan cara berpikir yang lebih khusus untuk melayani dan pengembangan yang berfokus pada layanan dan hasil dari implementasinya.

Suatu gaya arsitektur *software* yang bertumpu pada komunikasi antar *services*, dengan mendefinisikan bagaimana dua bagian atau lebih *service* dapat saling berinteraksi, dimana satu bagian dapat menjalankan bagian lainnya. Gaya perancangan *SOA* harus memenuhi kriteria metadata yaitu :

1. *“The metadata may be used by other applications, such as a catalog service to perform automatic discovery of services without changing the functionality of a service contract” (Pirna, 2016).*

Meta data dapat digunakan oleh aplikasi lain, seperti layanan catalog untuk melakukan performa layanan secara otomatis tanpa merubah fungsionalitas dari kontrak layanan.

2. “*Metadata should be provided in a form in which system designers can understand and manage services at a reasonable cost and effort*” (Pirna, 2016).

Meta data harus disediakan dalam bentuk dimana perancang sistem dapat memahami dan pengelolaan layanan dengan biaya yang wajar.

2.7 Cloud Computing

Cloud computing adalah paradigma komputasi terdistribusi dalam skala yang besar yang dilatar belakangi oleh faktor ekonomi, yang mana berisi kumpulan dari virtualisasi abstrak, skalabilitas yang dinamis, pengaturan kekuatan komputasi, tempat penyimpanan, platform dan layanan yang dapat diakses sesuai dengan kebutuhan oleh pelanggan eksternal melalui media Internet (Effendi, 2016).

Layanan utama yang disediakan oleh *cloud computing* dapat dibagi menjadi 3 bagian, diantaranya:

1. *IaaS (Infrastructure as a Service)*, kemampuan dalam menetapkan ketersediaan perangkat keras kepada konsumen meliputi: *processing, storage, networks and other fundamental computing resource*. Termasuk *operating systems and applications* (Jatmika dkk, 2019).
2. *PaaS (Platform as a Service)*, kemampuan dalam menyediakan layanan kepada konsumen untuk dapat membangun aplikasi yang mendukung kedalam infrastruktur *cloud computing* dengan menggunakan bahasa

pemrograman sehingga aplikasi tersebut dapat berjalan pada *platform* yang telah disediakan (Jatmika dkk, 2019).

3. *SaaS (software as a service)*, kemampuan dalam menyediakan layanan yang ditujukan kepada konsumen untuk dapat menjalankan aplikasi diatas infrastruktur *cloud computing* yang telah disediakan (Jatmika, Afwani dan Agitha, 2019).

2.8 Portal Informasi

Portal Informasi adalah sekumpulan aplikasi yang melibatkan data yang telah di proses atau diolah, sehingga data-data tersebut dapat saling melengkapi guna dihasilkan bahan informasi baru yang bermanfaat (Munawir, 2018).

Portal informasi sering juga disebut portal *web* dimana menyediakan berbagai informasi dan fasilitas bagi pengunjungnya. Beberapa portal *web* mempunyai kemampuan spesifik yaitu menyediakan sebuah informasi. Adapun elemen-elemen yang dimiliki sebuah portal informasi berbasis komputer adalah sebagai berikut (Munawir, 2018):

1. Perangkat keras (*hardware*)
2. Perangkat lunak (*software*)
3. Prosedur
4. Manusia (*brainware*)
5. Basis data

6. Jaringan komputer dan komunikasi data

Membangun sebuah komunitas dengan cara mengikat pengunjung secara sukarela ke dalam berbagai bentuk layanan yang disediakan. Dengan demikian tujuan membangun sebuah portal *web* akan tercapai

2.9 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Hendini, 2016).

Beberapa perancangan pengembangan sistem informasi yang berorientasi objek yang berbasiskan *UML* Sebagai berikut :

2.9.1 Flowchart Diagram

Flowchart adalah gambaran secara grafik dalam langkah-langkah dan urutan-urutan prosedur dari suatu program. Perancangan dan pengembangan berdasarkan analisa untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil menjadi alternatif-alternatif lain dalam pengoprasian.

Flowchart sangat mempermudah dalam penyelesaian suatu masalah khususnya kasus yang harus dipelajari dan dievaluasi lebih lanjut. Sehingga langkah-langkah tersebut akan lebih mudah dimengerti jika kita

menggambarkannya dalam suatu bagan yang dikenal dengan istilah: *flowchart* atau *bagan alir*.(Ridlo, 2017).

Berikut ini simbol-simbol *flowchart* yang biasanya di pakai adalah simbol-simbol *flowchart* yang dikeluarkan oleh ANSI dan ISO. Simbol-simbol ini disajikan pada Gambar 2.1.

SIMBOL	ARTI	CONTOH	SIMBOL	ARTI	CONTOH	SIMBOL	ARTI	CONTOH
	Merepresentasikan Input data atau Output data yang diproses atau Informasi.	Baca Jam & Tarif Upah		Keputusan dalam program	Apakah A < B =>		I/O dalam format yang dicetak	Cetak Slip Upah
	Merepresentasikan operasi	Hitung Upah Kotor		Rincian operasi berada di tempat lain	Hitung Akar Pangkat Dua		I/O yang menggunakan pita magnetik	File Upah
	Keluar ke atau masuk dari bagian lain flowchart khususnya halaman yang sama	3 Keluar 3 Masuk		Pemberian harga awal	SW = 1		I/O yang menggunakan disk magnetik	Update File Pegawai
	Merepresentasikan alur kerja	Hitung Upah Kotor 3		Awal / akhir flowchart	Start		I/O yang menggunakan drum magnetik	File Pegawai
	Digunakan untuk komentar tambahan	Urutkan Sebelum Pembayaran Berdasarkan No. Pelanggan		Input / output yang menggunakan kartu berlubang	Kartu Absen			

Gambar: 2. 1 Simbol-simbol *Flowchart* (Ridlo, 2017)

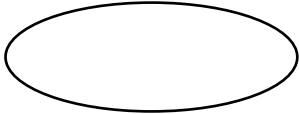
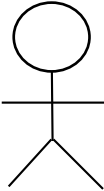


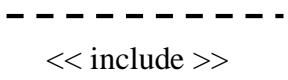
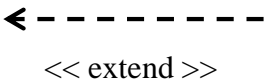
2.9.2 Use Case Diagram

Use Case diagram adalah diagram yang menggambarkan kebutuhan sistem dari sudut pandang *user*, yang memperlihatkan hubungan-hubungan yang terjadi antara *actors* dengan *use case* dalam sistem.

Use case diagram merupakan pemodelan untuk kelakuakn (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi

apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* (Hendini, 2016) yaitu




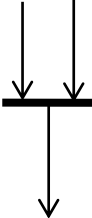
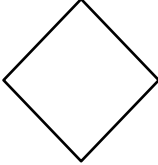
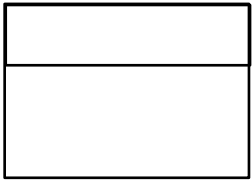
Tabel: 2. 1 *Use case diagram* (Hendini, 2016)

Gambar	Keterangan
	<i>Use Case Diagram</i> menggambarkan fungsionalitas.
	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya pemanggilan sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

2.9.3 *Activity Diagram*

Activity Diagram adalah diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu:

Tabel: 2. 2 *Activity diagram* (Hendini, 2016)

Gambar	Keterangan
	<i>Start point</i> , diletakan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , Akhir aktivitas
	<i>Activities</i> , menggambarkan suatu proses atau kegiatan bisnis
	<i>Join</i> (Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Point</i> , menggambarkan pilihan untuk pengambilan keputusan <i>true</i> atau <i>false</i>
	<i>Swimline</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

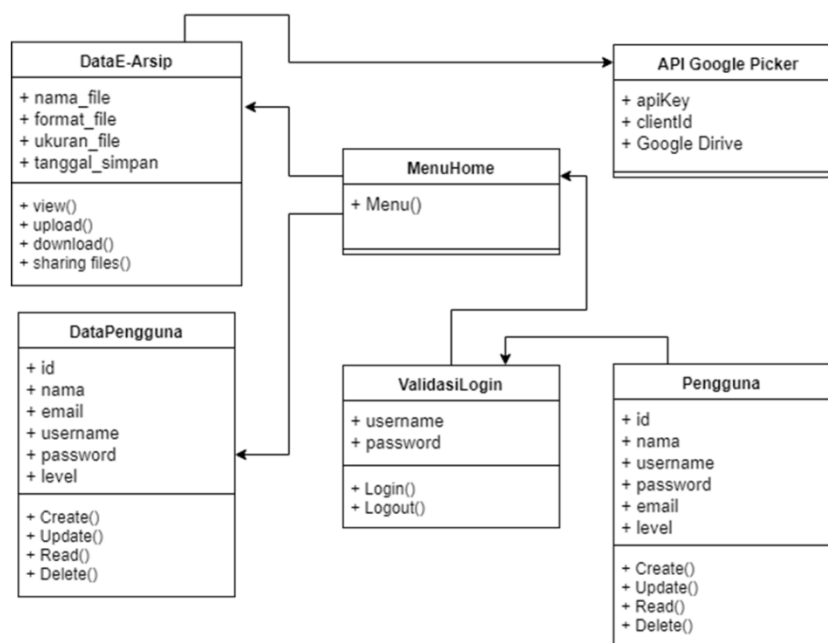
2.9.4 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode

atau fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan obyek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Tabel: 2. 3 *Class diagram* (Hendini, 2016)

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1...*	1 atau lebih
0..1	Boleh tidak ada maksimal 1
n...n	Batas antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

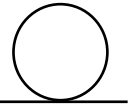
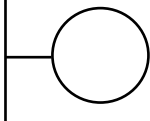
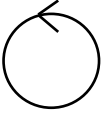


Gambar: 2. 2 Contoh gambar *class diagram* (Sontana dkk, 2019)

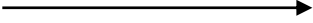

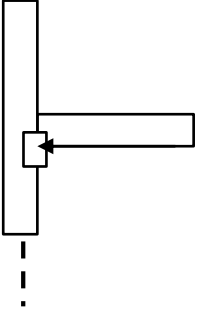
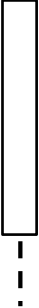
2.9.5 Sequence Diagram

Sequence diagram menggambarkan interaksi antar obyek di dalam dan di sekitar sistem (termasuk pengguna, display dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (obyek-obyek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Tabel: 2. 4 *Sequence diagram* (Hendini, 2016)

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>Boundary Class</i> , Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form</i> cetak
	<i>Control Class</i> , satu obyek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai obyek


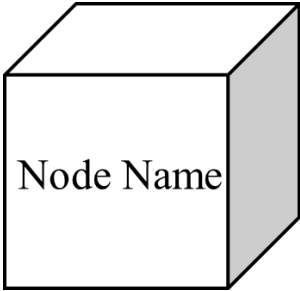

Tabel: 2. 4 *Sequence diagram* (Hendini, 2016) (lanjutan)

	<i>Message</i> , simbol mengirim pesan antar <i>class</i>
	<i>Lifeline</i> , garis titik-titik yang berhubungan dengan objek, sepanjang <i>Lifeline</i> terdapat <i>activation</i>
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi

2.9.6 *Deployment Diagram*

Deployment diagram merupakan diagram yang bisa menggambarkan detail dan menunjukkan bagaimana komponen disusun di infrastruktur sistem yang menjalankan sebuah proses eksekusi pada sebuah sistem. Simbol-simbol yang digunakan dalam *Deployment diagram* yaitu:



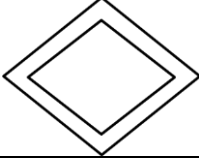
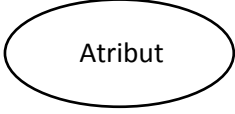
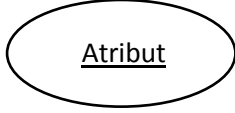

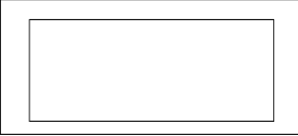
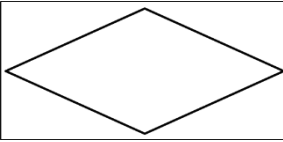
Tabel: 2. 5 *Deployment diagram* (Hendini, 2016)

Gambar	Keterangan
 <p data-bbox="502 555 707 600">Component</p>	<p data-bbox="847 495 1257 667">Pada deployment diagram komponen-komponen yang ada diletakan didalam node memastikan keberadaan posisi mereka</p>
 <p data-bbox="427 840 635 884">Node Name</p>	<p data-bbox="847 685 1249 857">Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 demensi</p>
	<p data-bbox="847 987 1270 1196">Sebuah assosiation digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-element hardware.</p>

2.9.7 *Entity Relational Diagram (ERD)*

Entity Relationship Diagram (ERD) menunjukkan hubungan antara 2 objek atau lebih yang memiliki atribut yang menggambarkan diantara mereka. Entitas adalah objek tempat menyimpan data. Suatu hubungan mendefinisikan koneksi yang diizinkan antara instasi entitas(Rahardja dkk, 2019).

Tabel: 2. 6 *Entity Relationship Diagram* (Putri, 2020)

Notasi	Deskripsi
	<i>Entity set</i>
	<i>Relationship set</i>
	Relasi pengidentifikasi
	Atribut
	<i>Key Atribut</i>
	<i>Derived atribut</i>
	<i>Entity weak set</i>
	Entitas Asosiatif

2.10 *Extreme Programming*

Metode *Extreme Programming* dalam pengembangan sistem informasi villatermurahciater.com akan berkonsentrasi pada 4 poin penting yaitu perencanaan, perancangan, pengkodean dan pengujian kemudian tahapan rilis, pada tahap perencanaan dan perancangan, penulis akan menuangkan konsep pada pengembangan sistem informasi villatermurahciater.com menjadi terpolo agar proses dari tahapan memulai proyek sampai dengan akhir proyek diharapkan berjalan dengan baik, dalam tahap perancangan penulis akan mencoba menggunakan pendekatan metode gaya arsitektur *Service Oriented Architecture*.

2.11 *JSON*

“*JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange*” (Patni, 2017).

JSON atau *JavaScript Object Notation* adalah standar terbuka berbasis teks ringan yang dirancang untuk pertukaran data yang dapat dibaca manusia, yang di dalamnya berisi pasangan antara *key* dan *value*.

2.12 *Postman*

Postman merupakan sebuah *software* yang mendukung dalam pengembangan sistem *REST API* dengan mengklasifikasikan *request* berdasarkan *request method*, *URL* dan parameter *request* lainnya pada proses mengirim dan menerima respon pada *server*.

2.13 *Black Box Testing*

Pengujian kotak hitam yang menguji perilaku sistem, berfokus kepada syarat fungsional sebuah perangkat lunak. Sebuah teknik pengujian kumpulan kondisi masukan yang sepenuhnya akan melakukan kebutuhan fungsional untuk program. Pengujian kotak hitam bukan alternatif untuk menguji kotak putih dan sebaliknya, ini merupakan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari yang diungkap oleh metode *white box testing*.

Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori tersebut:

1. Fungsi yang salah atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses basis data eksternal
4. Kesalahan perilaku atau kinerja
5. Kesalahan inisialisasi dan penghentian

Tidak seperti *white box testing*, yang dilakukan pada awal proses pengujian, pengujian kotak hitam cenderung diterapkan selama tahap-tahap struktur kendali. Pengujian kotak hitam sengaja mengabaikan struktur kendali. Pengujian kotak hitam berfokus untuk menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana validasi fungsional diuji?
2. Bagaimana perilaku dan kinerja sistem diuji?

3. Kelas-kelas masukan apakah yang akan membentuk *test case* yang baik?
4. Apakah sistem sangat sensitif pada nilai masukan tertentu?
5. Bagaimana batas-batas kelas data diisolasi?
6. Berapa kecepatan dan volume data yang dapat ditolerir oleh sistem?
7. Apa pengaruh kombinasi spesifik data pada operasi sistem?

Dengan menerapkan teknik pengujian kotak hitam maka akan mendapatkan serangkaian *test case* yang memenuhi kriteria sebagai berikut:

1. *test case* yang mengetahui jumlah yang lebih besar dari suatu jumlah *test case* sehingga dapat dikelola dengan mengurangi atau menambahkan sehingga mendapatkan rancangan pengujian yang wajar.
2. *test case* yang dapat mengetahui suatu kelas yang hilang dan kesalahan yang terkait dengan pengujian khusus yang telah dibuat atau dirancang.

BAB III

ANALISA MASALAH DAN PERANCANGAN PROGRAM

3.1 Metode Pengumpulan Data

Kebutuhan dalam penyusunan skripsi ini mempunyai cara dalam mengumpulkan data-data untuk kebutuhan analisa, kebutuhan dalam penyediaan data vila, tentunya hal tersebut akan disediakan untuk keperluan *service data* yang menangani *service request data* dari *rest client*. Pada tahap pengumpulan data dilakukan dengan menggunakan metode observasi, studi pustaka dan studi literatur.

3.1.1 Observasi

Hasil dari metode ini adalah kumpulan data-data yang akan disediakan untuk calon mitra bisnis IPCI yaitu tipe atau nama vila, harga masing-masing sewa vila per-hari, jumlah kapasitas orang yang bisa ditampung untuk masing-masing vila, fasilitas untuk masing-masing vila, gambar masing-masing vila dan kontak *person* yang akan melayani pesanan vila, data-data tersebut akan menjadi data utama. Data-data yang sudah diperoleh akan dijadikan layanan data yang disediakan oleh *web service villatermurahciater.com* yang nantinya layanan data tersebut bisa diakses atau “dikonsumsi” oleh *rest client* atau calon mitra.

3.1.1.1 Analisa Data Vila

Data vila adalah data utama dalam *service provider data* yang nantinya *service costumer* akan mengirim *service request* dan kemudian *service request* tersebut harus direspon oleh *service provider*.

Adapun data vila yang di sajikan pada Tabel 3.1 yaitu sebagai berikut:

Tabel: 3. 1 Data vila dari sistem informasi villatermurahciater.com

No	Nama Vila	Fasilitas	Kapasitas	Harga Sewa Weekday	Harga Sewa Weekend
1	F1	-Kolam Renang -3 Kamar -Dapur -TV -Ruang Tamu	20 Orang	2,800,000	3,500,000
2	F2	-Kolam Renang -3 Kamar -Dapur -TV -Ruang Tamu	15 Orang	2,000,000	1,500,000
3	D1	-4 Kamar -Dapur -TV -Ruang Tamu	30 Orang	2,500,000	4,000,000
4	D2	-Kolam Renang -4 Kamar -Dapur -TV -Ruang Tamu	20 Orang	3,000,000	4,000,000
5	D3	-Kolam Renang -2 Kamar -TV -Ruang Tamu	10 Orang	1,250,000	1,500,000

Tabel: 3.1 Data vila dari sistem informasi villatermurahciater.com (lanjutan)

No	Nama Vila	Fasilitas	Kapasitas	Harga Sewa Weekday	Harga Sewa Weekend
6	S1	-Kolam Renang -3 Kamar -Dapur -TV -Ruang Tamu	15 Orang	1,500,000	2,000,000
7	S2	-Kolam Renang -3 Kamar -Dapur -TV -Ruang Tamu	15 Orang	1,5,00,000	2,000,000
Nomor Telepon PIC 085860178701					

3.1.1.2 Analisa Pemesanan Vila

Pada pola pemesanan vila calon penyewa vila menghubungi langsung kepada nomor kontak personal pelayanan pemesanan vila untuk langsung melakukan pemesanan vila kemudian calon penyewa vila dan pic jasa pengantar vila membuat janji untuk bertemu dan melakukan pembayaran sewa vila ditempat.

3.1.1.3 Analisa Pemasaran Vila

Untuk pemasaran vila, selain pemanfaatan web site untuk pemasaran dan penyediaan informasi vila-vilanya, pemilik informasi villatermurahciater.com juga memanfaatkan sosial media seperti facebook, instagram dan menawarkan secara langsung kepada calon penyewa vila.

Pemasaran yang efektif tentunya akan membuka peluang pasar dalam memberikan informasi kepada masyarakat khususnya calon penyewa vila.

Memberikan inovasi pola pemasaran dirasa sangat dibutuhkan dengan kondisi dampak krisis ekonomi disaat pandemi didalam bidang jasa pariwisata khususnya hotel dan penginapan seperti vila.

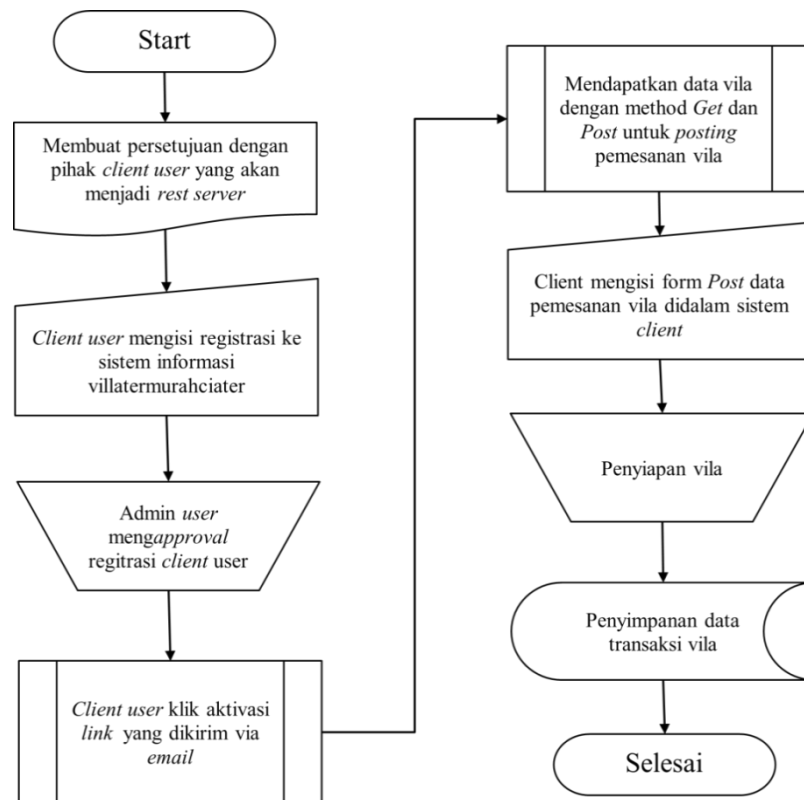
Dalam pengembangan sistem informasinya pemilik *web site* villatermurahciater.com akan membangun kemitraan dengan perusahaan jasa lainnya tentunya sebelum melakukan kemitraan tersebut pemilik ingin membuat simulasi kemitraan di dalam menyediakan data-data informasi vila untuk calon mitranya.

3.1.1.4 Analisa Pengembangan Sistem Informasi Vila

Dari metode observasi yang telah dilakukan, maka akan ada poin-poin pengembangan sistem informasi villatermurahciater.com dalam penyediaan layanan data vila dan beberapa aturan dasar untuk menjadi *rest server* villatermurahciater.com, dimana data vila tersebut dapat *direusable* atau digunakan kembali kedalam sistem calon mitra untuk keperluan informasi dan tujuan bisnis di dalam sistem informasi calon mitra secara fleksibel, sebagai gambaran simulasi layanan data vila itu berjalan, penulis mencoba menggambarkan proses awal membuat persetujuan, proses registrasi menjadi *rest server* dan penggunaan data vila yang telah didapatkan, akan tetapi gambaran aturan bermitra ini bukan menjadi sebuah aturan baku dalam pembuatan aturan bisnis dari ke 2 belah pihak.

Poin-poin pengembangan sistem informasi tersebut akan disajikan pada

Gambar 3.1.



Gambar: 3. 1 Flowchart pengembangan sistem informasi villatermurahciater.com

Dari gambar 3.1 diatas penjelasannya sebagai berikut:

1. Membuat persetujuan kemitraan dengan aturan bisnis yang dibuat adalah membuat harga paket perjalanan bus dan penginapan vila per 1 malam.
2. *Client user* mendaftarkan kepada sistem informasi villatermurahciater.com sebagai *rest server*.
3. *Admin user* akan *approval* registrasi tersebut dan mengirim *key* dan *link* aktivasi melalui *email*.

4. *Client user* menerima *email* dari *villatermurahciater.com* dan mengklik *link* tersebut untuk aktivasi akun.
5. *Client user* akan mendapatkan aksesnya yaitu *GET* untuk mendapatkan dan mengelola data khususnya data harga paket perjalanan dan vila. Untuk pemesanan vila, hak akses mitra akan ditambah yaitu *POST* yang secara sistem dari pihak mitra nantinya bisa membuat pemesanan vila kedalam sistem informasi *villatermurahciater.com*. Pengiriman data tersebut menyangkut dengan nomor transfer transaksi pemesanan vila, tanggal pemesanan vila dan lama penyewaan vila.
6. Data pemesanan vila *dipost* kedalam sistem *villatermurahciater.com* yang nantinya data tersebut bisa dimonitoring oleh *admin user* sehingga penyiapan vila ditanggal tersebut bisa disiapkan.
7. Penyiapan pesanan vila, data-data yang dikirim oleh *client* akan diproses sesuai dengan tanggal pemesanan vila dan lama hari penyewaan vila.
8. Penyimpanan data transaksi akan mempermudah dalam monitoring dan evaluasi baik evaluasi bulanan atau evaluasi tahunan.

3.1.2 Studi Pustaka

Studi pustaka yang dilakukan dengan cara mencari sumber-sumber dan informasi yang relevan dengan topik penelitian yang sedang dilakukan yaitu diantaranya jurnal, internet dan disertasi yang telah dipublikasikan.

Hasil dari metode studi pustaka ini menghasilkan informasi-informasi yang menjadi acuan atau dasar pengetahuan penelitian yang sedang dilakukan, terdapat contoh-contoh masalah dan penyelesaian masalah yang ada pada jurnal-jurnal atau pada penelitian-penelitian terdahulu.

Tabel: 3. 2 Refrensi Jurnal Terkait *SOA*, *API*, *Rest* dan *Restful*

No	Penulis	Tahun	Judul	Isi Jurnal
1	Dwi Wijornako, Betta Wahyu Retna Mulya	2018	Pengembangan Antarmuka Pemograman Aplikasi Menggunakan Metode Restful pada Sistem Informasi Akademik Politeknik Kota Malang	Jurnal ini menjelaskan tentang penelitian pengembangan sistem informasi SIAKAD dalam interpolasi antar platform yang berbeda agar bisa berkomunikasi atau menukar data dengan metode <i>Restful</i>
2	Indra Sontana, Alam Rahmatullah, Andi Nur Rahman	2019	<i>Application Programming Interface Google Picker</i> sebagai Penyimpanan Data Sistem Informasi Arsip Berbasis <i>Cloud</i>	Jurnal ini menjelaskan tentang pengembangan sistem E-Arsip yang memanfaatkan <i>API Google Picker</i> dan layanan <i>could</i> untuk menyimpan data arsip perusahaan dengan menggunakan metode pengembangan aplikasi yaitu <i>Extreme Programming (XP)</i>

Tabel: 3. 2 Refrensi Jurnal Terkait *SOA, API, Rest* dan *Restful* (lanjutan)

o	Penulis	Tahun	Judul	Isi Jurnal
3	Sultan T. Al. Anazi, Nibras Abdulah, Mohammad Anbar, Ola A. Al- wesabi	2019	<i>Evaluation Approches of Service Oriented Architecture (SOA) – A Survey</i>	Jurnal ini menjelaskan tentang evaluasi arsitektur SOA dengan biaya estimasi yang dikeluarkan oleh pengembangan dari arsitektur <i>SOA</i> sesuai dari tingkat kompleksifitas sebuah program
4	Amninder Singh	2017	<i>A Simulation Medel For InCremental Software Depelopment Life Cycle Model</i>	Jurnal ini menjelaskan tentang beberapa <i>life cycle model</i> mulai dari karakter setiap <i>life cycle</i> , rekomendasi untuk setiap <i>life cycle</i> dan kecocokan dalam pemembuat atau pengembangan sebuah sistem serta membuat comparasi untuk setiap <i>life cycle</i>
5	I putu Mahendra Pramadhitya, I Ketut Adi Purnawan, Ni Kadek Dwi Rusjyanthi	2016	Rancang Bangun Sistem Terdistribusi Pada Apotek	Jurnal ini menjelaskan tentang rancangan dalam membuat sistem yang terdistribusi dalam sistem apotek yang berfokus kepada mengatasi permasalahan ketersediaan data yang integrasi data pusat

Tabel: 3. 2 Refrensi Jurnal Terkait *SOA*, *API*, *Rest* dan *Restful* (lanjutan)

o	Penulis	Tahun	Judul	Isi Jurnal
6	Sanjay Patni	2017	<i>Pro RESTful APIs : Design, Build and Integrate with REST, JSON, XML, and JAR-RS</i>	Jurnal ini menjelaskan tentang pengenalan <i>RESTful APIs</i> dari mendesain sebuah <i>REST API</i> membangun dan menitegrasikan <i>REST</i> dengan <i>JSON</i> dan beberapa pengenalan dan komparasi diantaranya <i>SOAP</i> vs <i>REST</i> , <i>JSON</i> vs <i>XML</i>

Dari 6 jurnal di atas memberikan refrensi untuk saya dalam pengembangan dan dalam perancangan sistem informasi dengan mengadopsi metode *SOA* dan gaya perancangan menggunakan baik *REST* atau *RESTful* serta dapat meningkatkan fleksibilitas dan efektifitas untuk pengembangan *cross platform*.

3.2 Metode Pengembangan Sistem

Metode pengembangan sistem *villatermurahciater.com* akan berkonsentrasi pada 4 poin penting yaitu *planing* (perencanaan), *design* (perancangan), *coding* (pengkodean) dan *testing* (pengujian) kemudian tahapan *release*, pada tahap perencanaan dan perancangan, penulis akan menuangkan konsep pada bab ini dalam pengembangan sistem informasi *villatermurahciater.com* menjadi terpola agar proses dari tahapan memulai proyek sampai dengan akhir proyek diharapkan berjalan dengan baik, dalam tahap perancangan penulis akan mencoba menggunakan pendekatan metode *Service Oriented Architecture*.

Penjadwalan pengembangan sistem informasi villatermurahciater.com disajikan pada tabel 3.3 dimulai dari minggu pertama pada bulan September sampai dengan bulan Desember

Tabel: 3. 3 Penjadwalan pengembangan sistem informasi villatermurahciater.com

Bulan	September				Oktober				November				Desember			
Minggu Ke	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
Perencanaan																
Perancangan																
Pengkodean																
Pengujian																

3.2.1 Perencanaan

Analisis kebutuhan fungsionalitas data untuk pengembangan sistem informasi villatermurahciater.com. disajikan pada tabel 3.4.

Tabel: 3. 4 Analisis kebutuhan data

No	Nama Data	Atribut
1	Data Pengelola <i>web API</i>	<i>email dan password</i>
2	Data vila	Tipe atau nama vila, harga sewa vila, fasilitas dan gambar vila.

Table 3.4 merupakan analisa kebutuhan data yang berdasarkan observasi pada tahap pengumpulan data yang nantinya akan disajikan oleh sistem informasi villatermurahciater.com yang menjadi portal informasi dalam pengembangan sistem informasinya dan data tersebut bisa *direuseable* atau digunakan kembali oleh aplikasi lain atau calon mitra villatermurahciater.com.

3.2.1.1 Analisa Kebutuhan Perangkat Keras

Pada bagian analisis kebutuhan perangkat keras ini akan diuraikan kebutuhan-kebutuhan non fungsional yang berhubungan dengan spesifikasi kebutuhan perangkat keras yang mampu mendukung perangkat lunak yang dibutuhkan agar dapat berjalan dengan baik. Adapun analisis spesifikasi kebutuhan perangkat keras yang dibutuhkan untuk dapat menjalankan aplikasi yang akan dibangun adalah sebagai berikut :

Tabel: 3. 5 Perangkat keras pengembang

No	Perangkat Keras	Spesifikasi
1	Procesor	2,4 GHz
2	RAM	1 GB
3	VGA	1 GB
4	Hardisk	100 GB
5	Monitoring Resolusi	1366 x 768

Dilihat dari analisis spesifikasi kebutuhan perangkat keras (tabel 3.5) yang harus disiapkan penulis sudah mempunyai dan memenuhi standar.

3.2.1.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya. Berdasarkan hasil pengamatan yang ada, perangkat lunak yang dimiliki oleh pemngembang dapat dilihat pada tabel 3.6.

Tabel: 3. 6 Spesifikasi kebutuhan perangkat lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 7
2	Browser	Google Chrome dan Fire Fox

3.2.2 Perancangan

Pada tahap ini perancangan pengembangan sistem informasi villatermurahciater.com akan membuat sebuah *Use Case diagram* kemudian dilanjutkan sampai ketahap *Class diagram* dan *Deployment diagram*.

3.2.2.1 Perancangan *Use Case Diagram*

Use Case Diagram adalah gambaran grafis dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, *actor*, dan sistem, Berikut adalah tabel dan beberapa gambar *use case diagram* sistem.

Tabel: 3. 7 Identifikasi aktor

No	Aktor	Deskripsi
1	<i>Client User</i>	Orang yang mendaftar sebagai <i>REST client</i> . Sebelum <i>client user</i> tersebut meminta dan mendapatkan data informasi villatermurahciater.com. dengan menggunakan metode <i>HTTP</i> yaitu <i>Get</i> .
2	<i>Mitra User</i>	Orang yang dapat melihat informasi tentang vila, dapat mendaftar sebagai mitra villatermurahciater.com, dapat mengupload sejumlah data-data vila dan merubah atau <i>edit</i> data-data vila sesuai dengan data vila yang telah diupload
3	<i>Admin User</i>	Orang yang dapat melihat informasi vila, informasi <i>client user</i> , mitra <i>user</i> dan dapat melihat data vila sesuai yang diupload oleh mitra <i>user</i> , serta dapat mengelola data vila dengan menggunakan metode <i>HTTP</i> yaitu <i>Get</i> , <i>Put</i> , <i>Post</i> dan <i>Delete</i> .

3.2.2.2 Identifikasi Use Case

Tabel: 3. 8 Identifikasi use case mitra user

No	Nama Use Case	Deskripsi
1	Melihat halaman utama web site	Use Case menggambarkan kegiatan mitra user dan client user dapat melihat halaman utama villatermurahciater.com
2	Sign up Sebagai mitra	Use Case menggambarkan kegiatan memasukan email dan password oleh mitra user untuk mendaftar sebagai member dari sistem informasi villatermurahciater.com, kemudian mendapatkan link aktivasi dikirim ke email user
3	Login	Use Case menggambarkan kegiatan memansukan email dan password oleh mitra user untuk masuk kedalam sistem informasi villatermurah.com
4	Upload data vila	Use Case menggambarkan kegiatan user mengunggah data vila ke villatermurahciater.com
5	Edit data vila	Use Case menggambarkan kegiatan user mengubah data vila yang sudah diunggah sebelumnya
6	View Data vila	Use Case mengggambarkan kegiatan user melihat data vila dengan memilih sub menu edit
7	Logout	Use Case menggambarkan kegiatan user keluar dari sistem informasi villatermurahciater.com

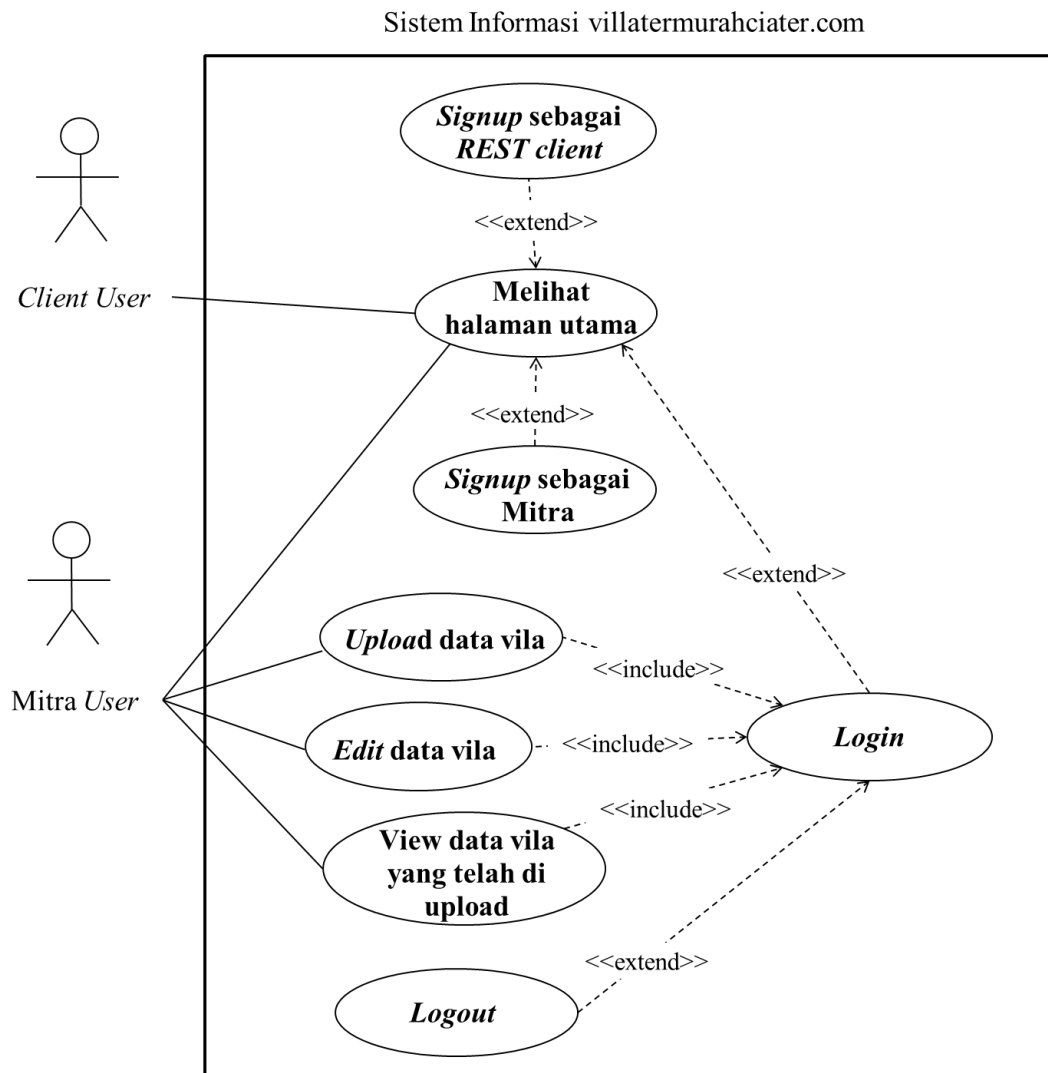
Tabel: 3. 9 Identifikasi use case client user

No	Nama Use Case	Deskripsi
1	Sign up sebagai Rest client	Use Case menggambarkan kegiatan user mendaftar sebagai Rest client.

Tabel: 3. 10 Identifikasi *use case* admin *user*

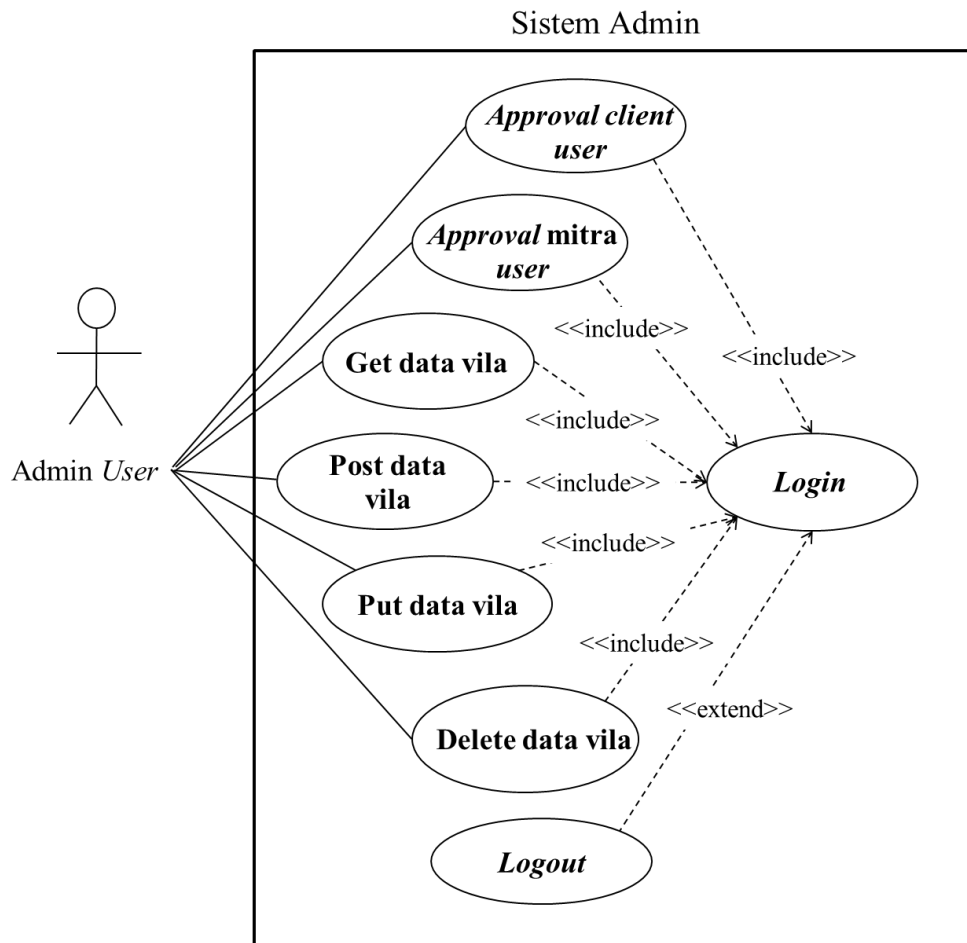
No	Nama Use Case	Deskripsi
1	<i>Login</i>	<i>Use Case</i> menggambarkan kegiatan memasukan <i>email</i> dan <i>password</i> oleh Admin untuk masuk kedalam sistem informasi admin <i>villatermurahciater.com</i>
2	<i>Approval</i> registrasi mitra <i>user</i>	<i>Use Case</i> menggambarkan kegiatan admin memproses <i>approval</i> registrasi mitra <i>user</i> dengan cara masuk ke dalam sistem informasi admin <i>villatermurahciater.com</i> .
3	<i>Approval</i> registrasi <i>client user</i>	<i>Use Case</i> menggambarkan kegiatan admin memproses <i>approval client user</i> dengan cara masuk ke dalam sistem informasi admin <i>villatermurahciater.com</i> .
4	<i>Get</i> data vila	<i>Use Case</i> menggambarkan kegiatan admin meminta data vila ke <i>web server</i> <i>villatermurahciater.com</i>
5	<i>Put</i> data vila	<i>Use Case</i> menggambarkan kegiatan admin mengubah data vila ke <i>web server</i> <i>villatermurahciater.com</i>
6	<i>Post</i> data vila	<i>Use Case</i> menggambarkan kegiatan admin mengunggah data vila ke <i>web server</i>
7	<i>Delete</i> data vila	<i>Use Case</i> menggambarkan kegiatan admin menghapus data vila ke <i>webs server</i>
8	<i>Logout</i>	<i>Use Case</i> menggambarkan kegiatan admin keluar dari sistem

1. Use Case Diagram Client User dan Mitra User



Gambar: 3. 2 Use case diagram mitra user dan client user

2. Use Case Diagram Admin User



Gambar: 3. 3 Use case diagram admin user

3.2.2.2.1 Skenario Use Case

Skenario pada gambar *Use Case Diagram* diatas disajikan pada tabel 3.11 sampai dengan tabel 3.24

1. Skenario Use Case Mitra User

Tabel: 3. 11 Skenario use case diagram sign up mitra user

<p>ID : 1.1 Mitra User Nama Use Case : <i>Sign up</i> Aktor : <i>Mitra User</i> Tujuan Keputusan : Pengguna telah mengakses aplikasi namun belum memilih fitur menu di dalam sistem Deskripsi : Pengguna memasukan <i>email</i> dan <i>password</i> kedalam <i>form sign up</i></p>		
No	Aksi Aktor	Reaksi Sistem
1	Mitra User mengisi <i>form sign up</i> dengan memasukan <i>email</i> dan <i>password</i> yang salah lalu menekan tombol <i>sign up</i>	Menampilkan pesan <i>email</i> yang valid dan konfirmasi <i>password</i> harus sama dengan <i>password</i> utama
2	Mitra User mengisi <i>form sign up</i> dengan memasukan <i>email</i> dan <i>password</i> yang kosong menekan tombol <i>sign up</i>	Menampilkan <i>form sign up</i>
3	Mitra User mengisi <i>form sign up</i> dengan memasukan <i>email</i> dan <i>password</i> lalu menekan tombol <i>sign up</i> yang benar	Menampilkan informasi kesalahan <i>sign up email</i> dan <i>password field</i> harus diisi
4		Mengirim <i>email link</i> verifikasi

Tabel: 3. 12 Skenario *use case diagram* login mitra user

ID : 1.2 Mitra User Nama Use Case : Login Aktor : Mitra User Tujuan Keputusan : Pengguna telah mengakses aplikasi namun belum memilih fitur menu di dalam sistem Deskripsi : Pengguna memasukan <i>username</i> dan <i>password</i> untuk mengakses menu aplikasi atau sistem		
No	Aksi Aktor	Reaksi sistem
1	Mengklik <i>link</i> aktivasi di dalam <i>email</i> yang telah di kirim	<i>Direct</i> ke halaman <i>login</i>
2	Mitra User mengisi <i>form login</i> dengan memasukan <i>username</i> dan <i>password</i> yang salah lalu menekan tombol <i>login</i>	Melakukan verifikasi <i>username</i> dan <i>password</i> yang dimasukan oleh User
3	Memasukan <i>username</i> dan <i>password</i> yang benar	Menampilkan informasi kesalahan <i>login username</i> dan <i>password</i> salah
4		Melakukan verifikasi <i>username</i> dan <i>password</i> yang dimasukan oleh User
5		Menampilkan halaman pengelolaan data vila untuk <i>Upload</i> data vila, <i>Edit</i> data vila dan melihat data vila yang telah di <i>upload</i>

Tabel 3. 13 Skenario *use case diagram upload data vila mitra user*

ID	: 1.3 Mitra User	
Nama Use Case	: Upload Data Vila	
Aktor	: Mitra User	
Tujuan Keputusan	: Pengguna memilih fitur <i>Upload</i> menu di dalam sistem villatermurahciater.com	
Deskripsi	: Pengguna melihat dan menambahkan data vila sesuai dengan hak aksesnya	
No	Aksi Aktor	Reaksi Sistem
1	Mitra User mengisi <i>form</i> data vila yang salah lalu menekan tombol <i>Upload</i>	Menampilkan <i>field</i> dalam <i>form</i> harus sesuai dengan nama <i>field</i> masing-masing
2	Mitra User mengisi <i>form</i> data vila yang benar lalu menekan tombol <i>Upload</i>	Menerima dan menyimpan data vila yang telah diupload ke dalam <i>data base</i> villatermurahciater.com
3		Menampilkan <i>upload</i> telah berhasil

Tabel 3. 14 Skenario *use case diagram edit data vila mitra user*

ID	: 1.4 Mitra User	
Nama Use Case	: Edit Data Vila	
Aktor	: Mitra User	
Tujuan Keputusan	: Pengguna memilih fitur <i>Edi</i> di dalam sistem villatermurahciater.com	
Deskripsi	: Pengguna melihat dan menambahkan data vila sesuai dengan hak aksesnya	
No	Aksi Aktor	Reaksi Sistem
1	Mitra User memilih menu <i>Edit</i>	Menampilkan data vila yang telah diupload

Tabel 3.14 Skenario *use case diagram edit data vila mitra user* (lanjutan)

No	Aksi Aktor	Reaksi Sistem
2	Mitra <i>User</i> memilih data vila mana yang akan diubah lalu klik pada tombol <i>edit</i> vila	Menampilkan data vila berupa <i>form</i> untuk mitra <i>user</i> merubah data vila tersebut
3	Mitra <i>User</i> merubah data vila secara lengkap dan memilih tombol ' <i>Edit</i> '	Membawa ke halaman pilih data ubah dan menampilkan pesan data telah diubah

Tabel 3. 15 Skenario *use case diagram view data vila mitra user*

ID	: 1.5 Mitra <i>User</i>
Nama <i>Use Case</i>	: <i>View Data Vila</i> Mitra <i>User</i>
Aktor	: Mitra <i>User</i>
Tujuan Keputusan	: Pengguna memilih fitur <i>View Data Vila</i> di dalam sistem villatermurahciater.com
Deskripsi	: Pengguna melihat data vila sesuai dengan hak aksesnya

No	Aksi Aktor	Reaksi Sistem
1	Mitra <i>User</i> memilih tombol <i>ViewData Vila</i>	Menampilkan data vila yang telah <i>diupload</i> atau di <i>edit</i>

2. Skenario *Use Case Admin User*

Tabel 3. 16 Skenario *use case diagram login admin user*

ID	: 1.1 Admin <i>User</i>
Nama <i>Use Case</i>	: <i>Login</i>
Aktor	: Admin <i>User</i>
Tujuan Keputusan	: Pengguna telah mengakses aplikasi namun belum memilih fitur menu di dalam sistem
Deskripsi	: Pengguna memasukan <i>user id</i> dan <i>password</i> untuk mengakses menu aplikasi atau sistem

Tabel3.16 Skenario *use case diagram login admin user* (lanjutan)

No	Aksi Aktor	Reaksi sistem
1	Pengguna mengisi <i>form login</i> dengan memasukan <i>user id</i> dan <i>password</i> lalu menekan tombol <i>login</i>	Melakukan verifikasi <i>user id</i> dan <i>password</i> yang dimasukan oleh pengguna
2	Memasukan <i>user id</i> dan <i>password</i> yang tidak valid	Menampilkan informasi kesalahan <i>login username</i> dan <i>password</i> salah
3	Memasukan <i>user id</i> dan <i>password</i> yang valid	Menampilkan halaman pengelolaan data <i>API</i>

Tabel: 3. 17 Skenario *use case diagram admin user approval data mitra user*

ID	: 1.2 Admin User	
Nama Use Case	: Approval data mitra user	
Aktor	: Admin User	
Tujuan Keputusan	: Admin user memberikan Approval kepada mitra user	
Deskripsi	: Admin memilih menu <i>user</i> pada menu <i>approval data mitra user</i>	
No	Aksi Aktor	Reaksi Sistem
1	Admin user memilih menu <i>user</i> pada menu nav bar <i>approval</i>	Menampilkan data mitra <i>user</i>
2	Admin user memilih tombol "Approve"	Menampilkan data mitra <i>user</i> berhasil dikirim <i>email</i> aktivasi

Tabel 3. 18 Skenario *use case diagram admin user approval data client user*

ID	: 1.3 Admin User	
Nama Use Case	: Approval Data Client User	
Aktor	: Admin User	
Tujuan Keputusan	: Admin user memberikan Approval kepada client user	
Deskripsi	: Admin memilih menu <i>client</i> pada menu <i>approval data client user</i>	

Tabel 3.18 Skenario *use case diagram* admin user approval data client user

(lanjutan)

No	Aksi Aktor	Reaksi Sistem
1	Admin user memilih menu <i>client</i> pada menu nav bar <i>approval</i>	Menampilkan data <i>client user</i>
2	Admin user memilih tombol “ <i>Approve</i> ”	Menampilkan data <i>client user</i> berhasil dikirim <i>email</i> aktivasi

Tabel: 3. 19 Skenario *use case diagram* admin get data vila

ID	: 1.4 Admin User	
Nama Use Case	: Admin User Get data vila	
Aktor	: Admin User	
Tujuan Keputusan	: Admin user dapat melihat detail data vila yang ada pada sistem informasi vilatermurahciater.com	
Deskripsi	: Admin user dapat melihat detail data vila yang ada pada sistem informasi vilatermurahciater.com	
No	Aksi Aktor	Reaksi Sistem
1	Admin user memilih sub menu <i>Get</i> pada menu <i>icon burger</i> di nav bar	Menampilkan detail data vila

Tabel : 3. 20 Skenario *Use Case Diagram* Admin User Put Data Vila

ID	: 1.5 Admin User	
Nama Use Case	: Admin User Put data vila	
Aktor	: Admin User	
Tujuan Keputusan	: Admin User dapat merubah data vila yang ada pada sistem informasi vilatermurahciater.com	
Deskripsi	: Admin User merubah data vila sesuai dengan hak aksesnya	
No	Aksi Aktor	Reaksi Sistem

Tabel: 3.20 Skenario *use case diagram* admin user put data vila (lanjutan)

1	Admin User memilih menu <i>Put</i>	Menampilkan <i>form</i> untuk mengubah
2	Memilih memilih dan mengisi data vila yang akan di ubah	
3	Mengisi <i>form</i> dan klik tombol “ <i>PUT</i> ”	Menampilkan data berhasil dirubah

Tabel: 3.21 Skenario *use case diagram* admin user post data vila

ID	: 1.6 Admin User	
Nama Use Case	: Admin User Post data vila	
Aktor	: Admin User	
Tujuan Keputusan	: Admin User dapat menambah data vila yang ada pada sistem informasi vilatermurahciater.com	
Deskripsi	: Admin menambah data vila sesuai dengan hak aksesnya	
No	Aksi Aktor	Reaksi Sistem
1	Admin User memilih menu <i>Post</i>	Menampilkan <i>form</i> untuk menambah data vila
2	Mengisi data vila yang akan di tambah dan menekan tombol “ <i>POST</i> ”	Menampilkan data vila telah ditambahkan

Tabel: 3. 22 Skenario *use case diagram* admin user delete data vila

ID	: 1.7 Admin User	
Nama Use Case	: Admin User Delete data vila	
Aktor	: Admin User	
Tujuan Keputusan	: Admin User dapat menghapus data vila yang ada pada sistem infomasi vilatermurahciater.com	
Deskripsi	: Admin menghapus data vila sesuai dengan hak aksesnya	
No	Aksi Aktor	Reaksi Sistem
1	Admin User memilih menu <i>Delete</i>	Menampilkan data vila yang akan di hapus

Tabel: 3.22 Skenario *use case diagram* admin user delete data vila (lanjutan)

No	Aksi Aktor	Reaksi Sistem
2	Memilih data vila yang akan di hapus dan menekan tombol "Delete"	Menampilkan data vila telah terhapus

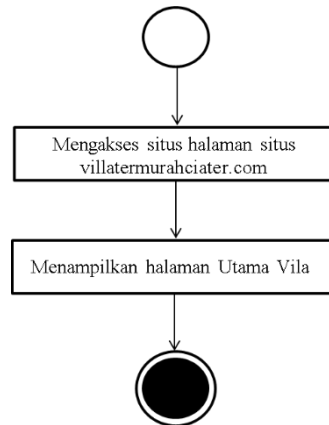
Tabel: 3. 23 Skenario *use case diagram* logout admin user dan mitra user

ID	: 1.8 Admin User dan Mitra User	
Nama Use Case	: Logout	
Aktor	: Pengguna (Admin User, Mitra User)	
Tujuan Keputusan	: Pengguna memilih fitur Logout untuk keluar dari sistem	
Deskripsi	: Admin user dan mitra user keluar dari sistemvillatermurahciater.com	
No	Aksi Aktor	Reaksi Sistem
1	Memilih menu <i>logout</i>	Menampilkan halaman utama sistem

3.2.2.3 Activity Daigram

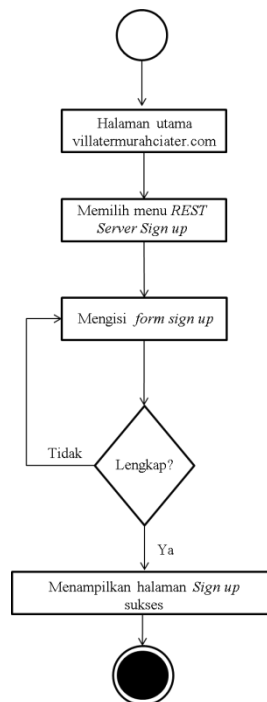
Activity diagram, sesuai dengan namanya diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Berikut adalah *activity diagram* sistem informasi villatermurahciater.com. *Activity diagram* disajikan dalam bentuk gambar 3.4 sampai dengan gambar 3.19.

1. Activity Diagram Mira User dan Client User Halaman Utama



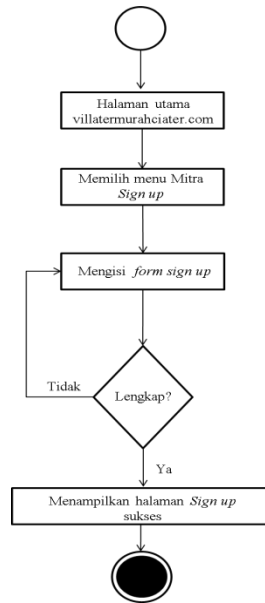
Gambar: 3. 4 Activity diagram mitra user dan client user halaman utama

2. Activity Diagram Sign up Client User



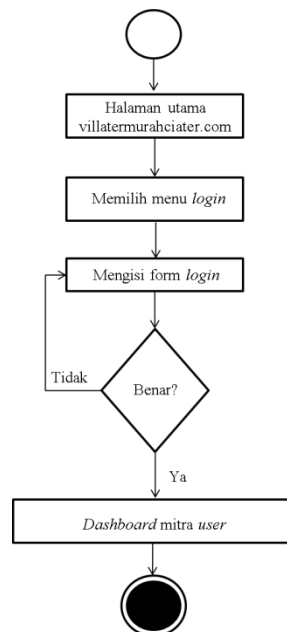
Gambar: 3. 5 Activity Diagram Sign up Client User

3. Activity Diagram Sign up Mitra User



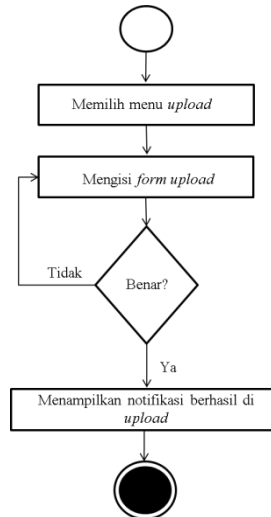
Gambar: 3. 6 Activity diagram sign up mitra user

4. Activity Diagram Login Mitra User



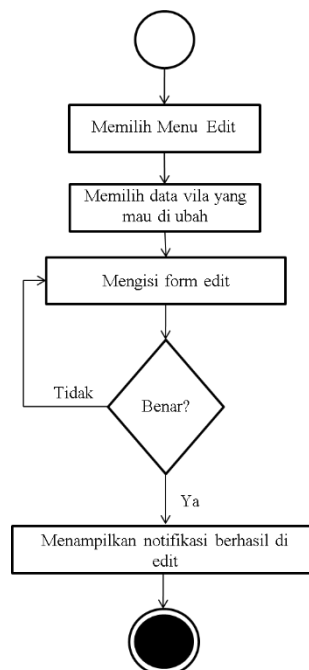
Gambar: 3. 7 Activity diagram login mitra user

5. Activity Diagram Mitra User Upload Data Vila



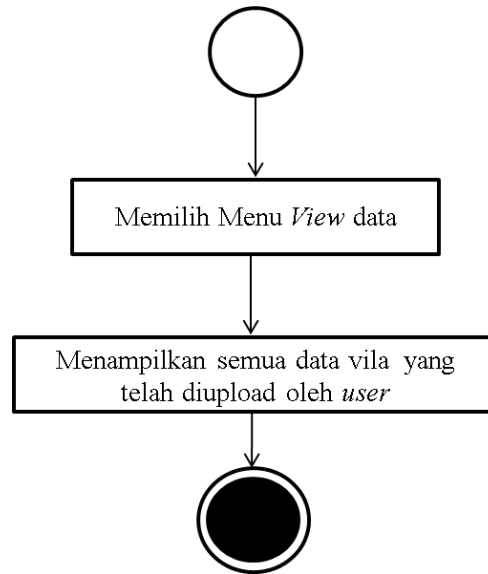
Gambar: 3. 8 Activity diagram upload data vila oleh mitra user

6. Activity Diagram Edit Mitra User Data Vila



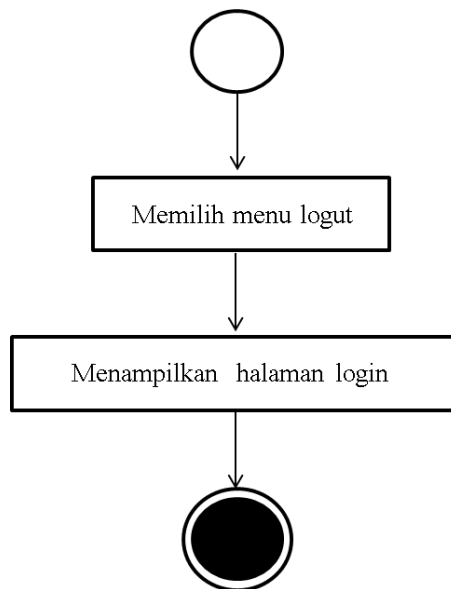
Gambar: 3. 9 Activity diagram edit data vila oleh mitra user

7. Activity Diagram Mitra User View Data



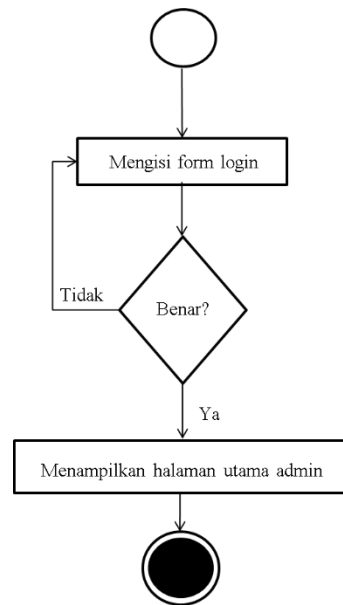
Gambar: 3. 10 Activity diagram view data oleh mitra user

8. Activity Diagram Logout Mitra User



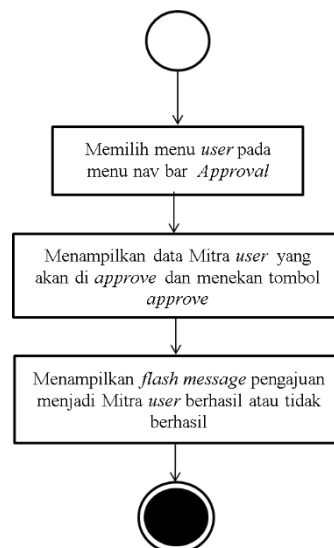
Gambar: 3. 11 Activity diagram logout mitra user dan client user

9. Activity Diagram Login Admin User



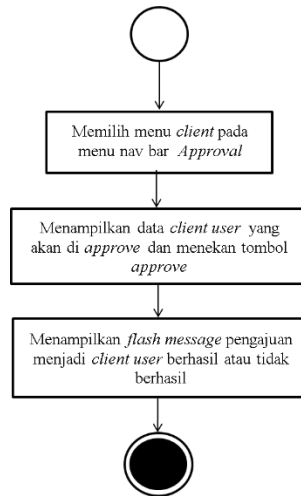
Gambar: 3. 12 Activity diagram login admin user

10. Activity Diagram Admin View Data Approval Untuk Mitra User



Gambar: 3. 13 Activity diagram admin view data approval untuk mitra user

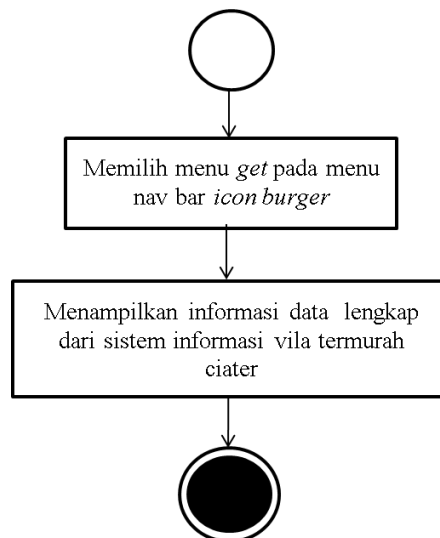
11. Activity Diagram Admin View Data Approval Untuk Client User



Gambar: 3. 14 Activity diagram admin view data approval untuk client user

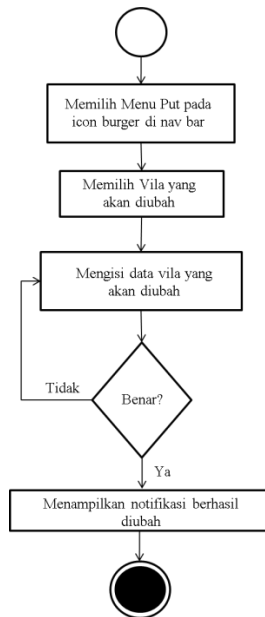
12. Activity Diagram Get Data Dari Sistem Informasi Vila Termurah Ciater

Oleh Admin User



Gambar: 3. 15 Activity diagram get data dari sistem informasi vila termurah ciater oleh admin user

13. Activity Diagram Admin User Put Data Vila



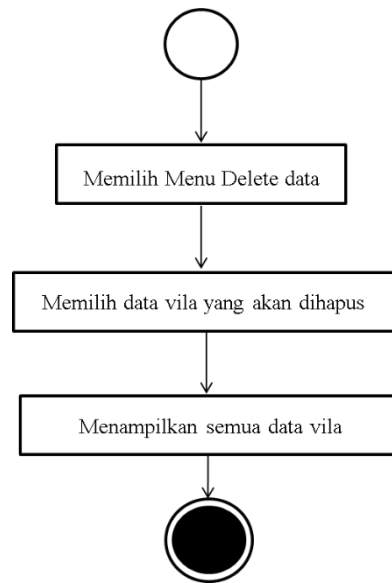
Gambar: 3. 16 Acivity diagram admin user put data vila

14. Activity Diagram Admin User Post Data Vila



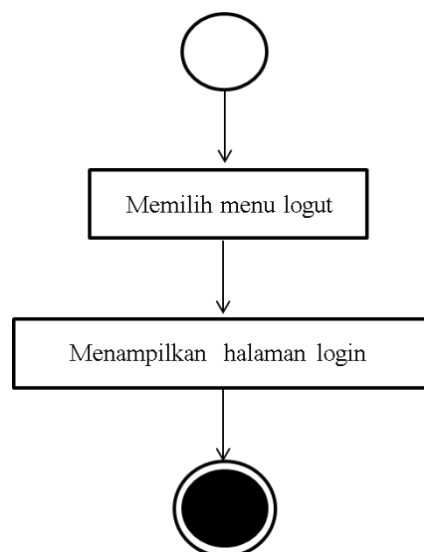
Gambar: 3. 17 Activity diagram admin user post data vila

15. Activity Diagram Admin User Delete Data vila



Gambar: 3. 18 Activity diagram admin user delete data vila

16. Activity Diagram Admin User Logout



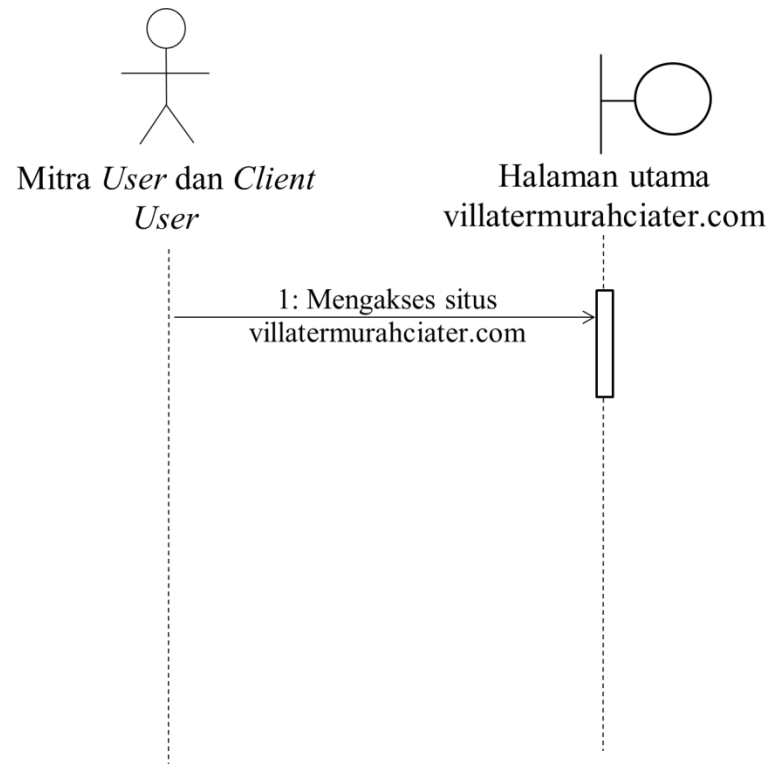
Gambar: 3. 19 Activity diagram admin user logout

3.2.2.4 *Sequence Diagram*

Sequence Diagram adalah salah satu dari diagram - diagram yang ada pada UML, *sequence diagram* ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object. Berikut adalah *sequence diagram* sistem informasi villatermurahciater.

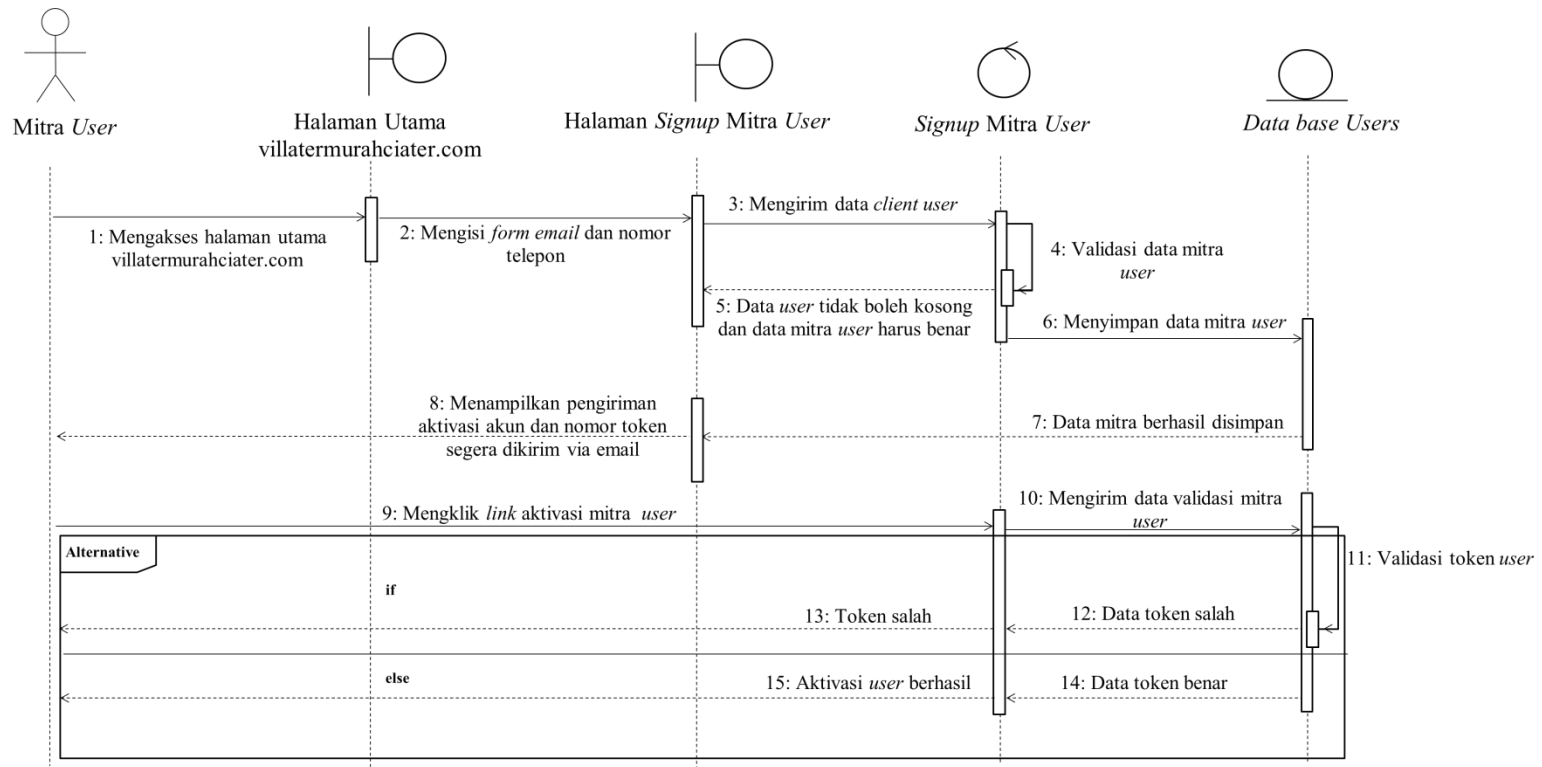
Sequence diagram disajikan dalam bentuk gambar 3.20 sampai dengan gambar 3.35.

1. Sequence Diagram Mitra User dan Client User Halaman Utama



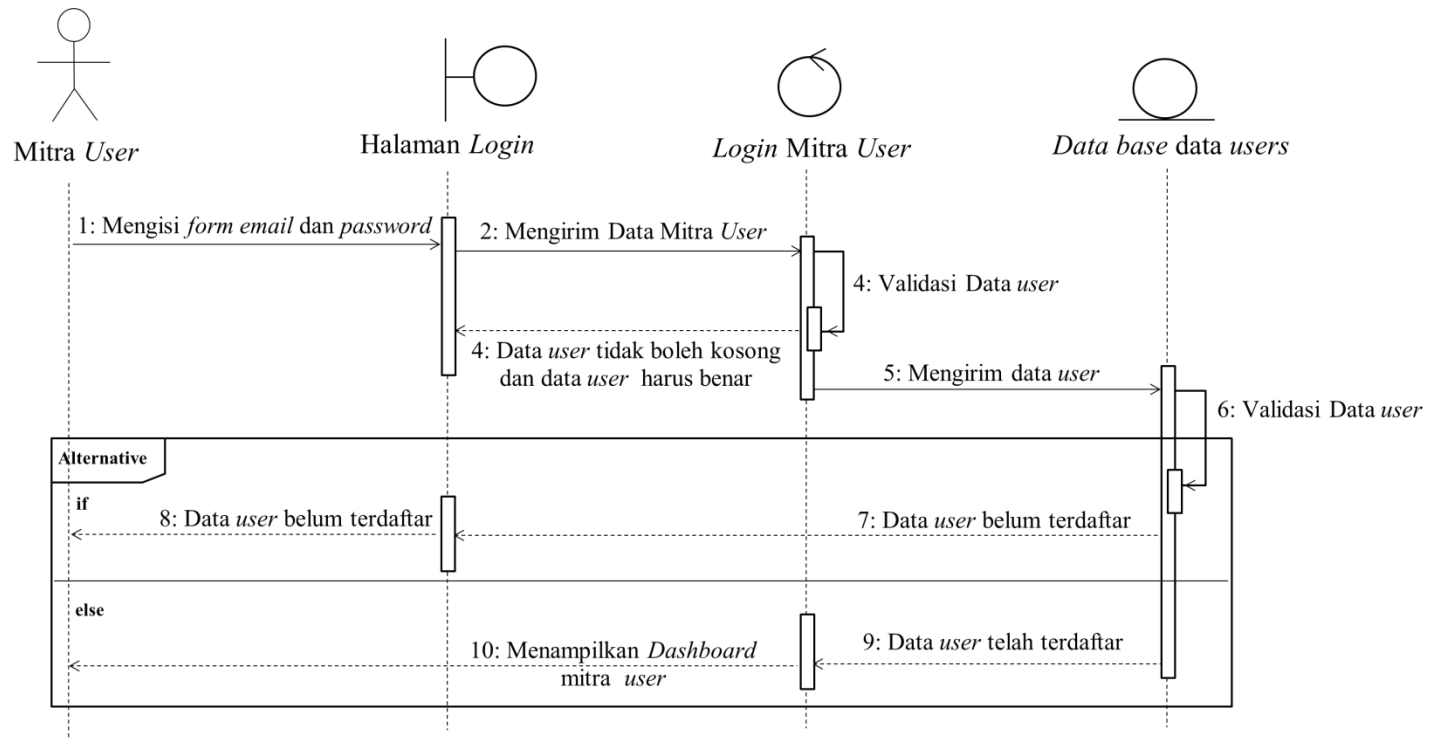
Gambar: 3. 20 Sequence diagram mitra user dan client user halman utama

2. Sequence Diagram Sign up Mitra User



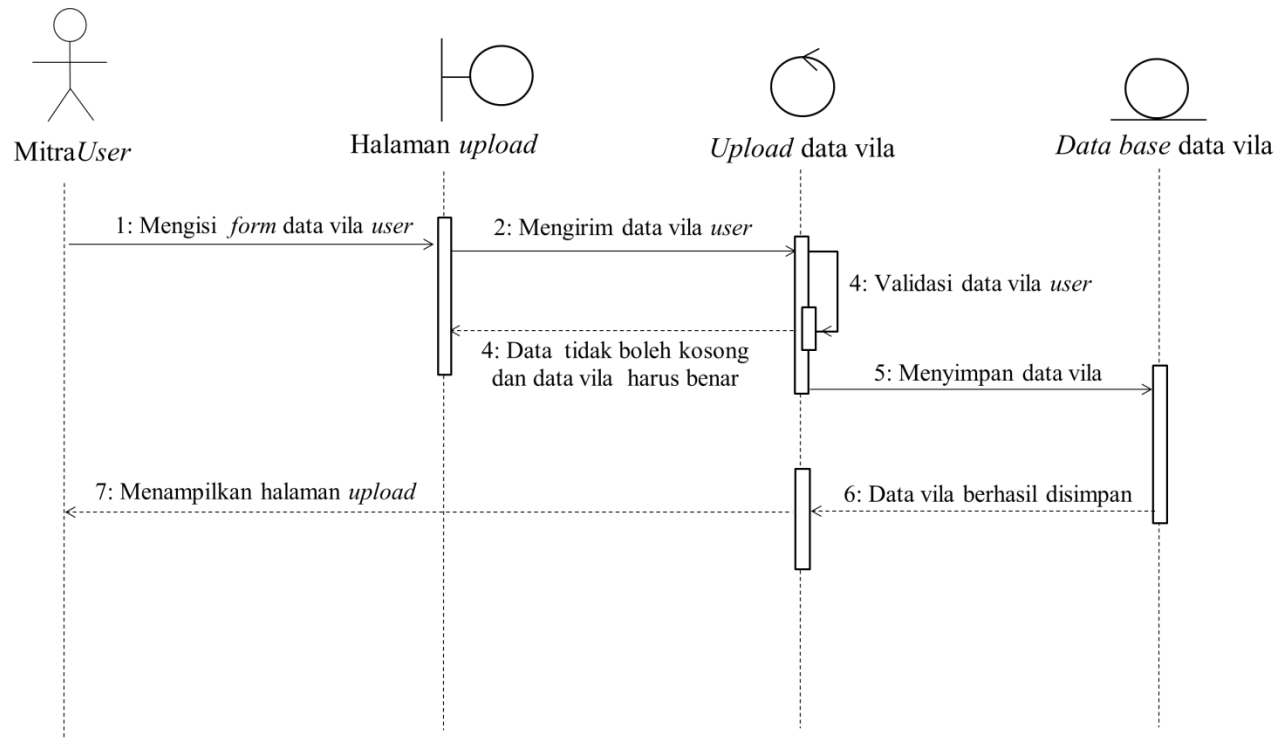
Gambar: 3. 21 Sequence diagram sign up mitra user

3. Sequence Diagram Login Mitra User



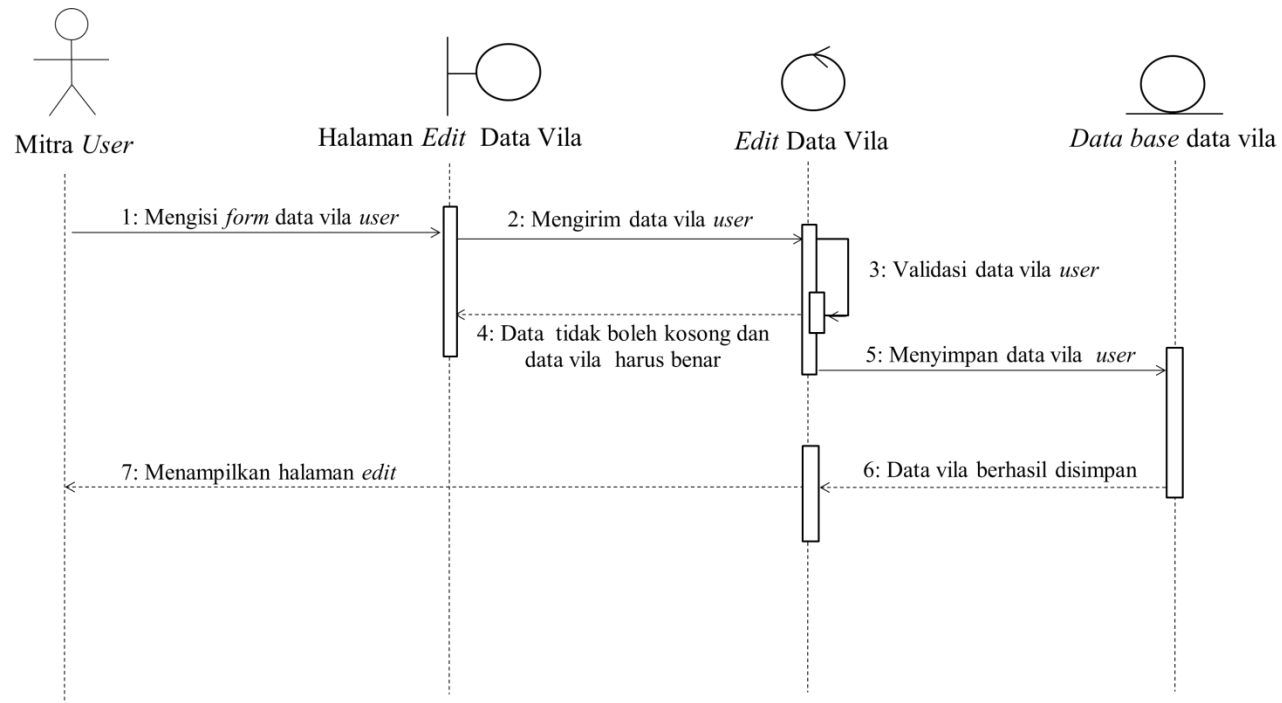
Gambar: 3. 22 Sequence diagram login mitra user

4. Sequence Diagram Mitra User Upload Data Vila



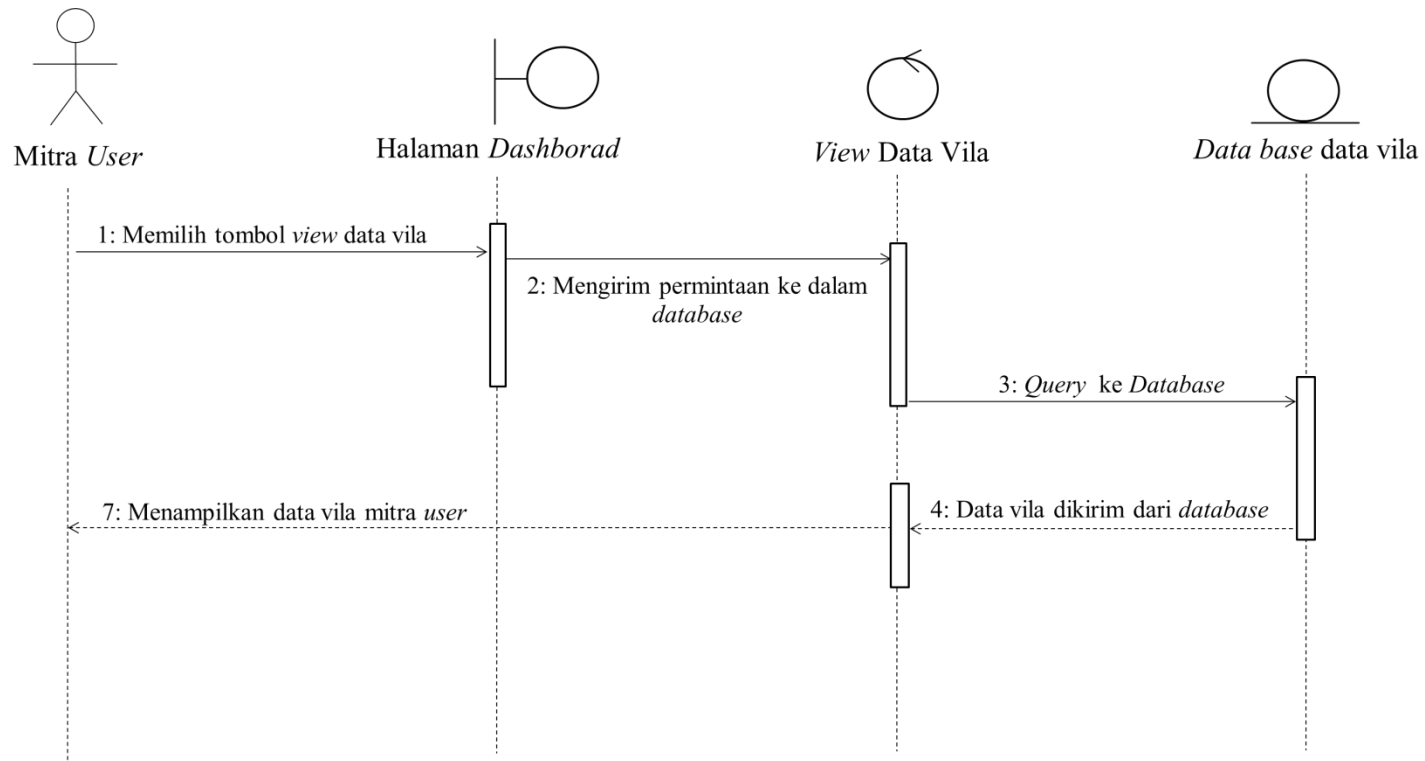
Gambar: 3. 23 Sequence diagram mitra user upload data vila

5. Sequence Diagram Mitra User Edit Data Vila



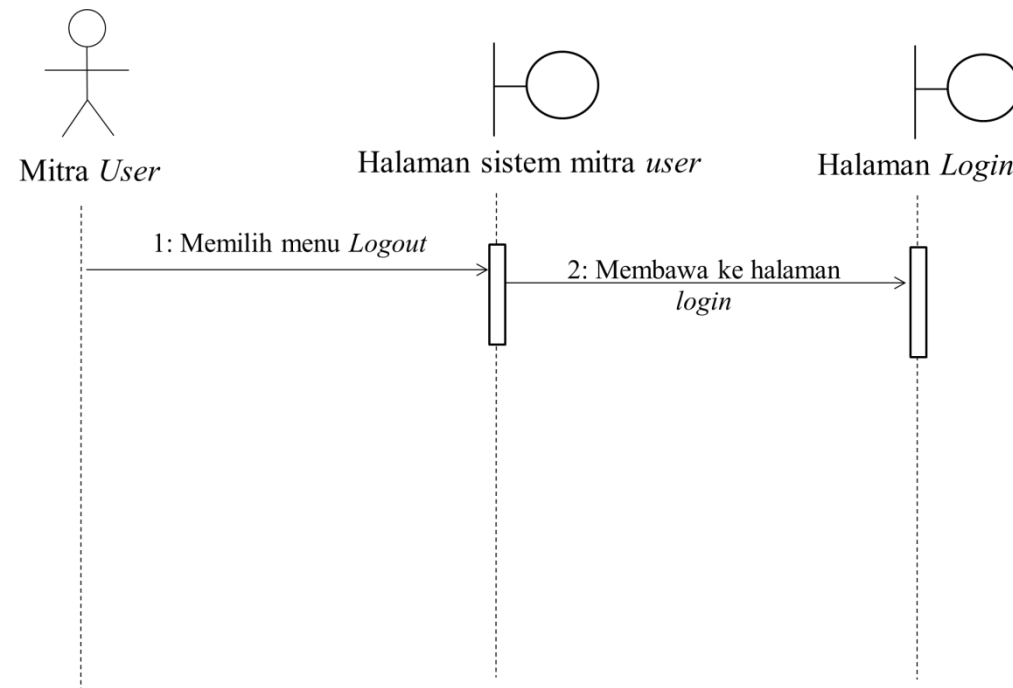
Gambar: 3. 24 Sequence diagram mitra user edit data vila

6. Sequence Diagram Mitra User View Data Vila



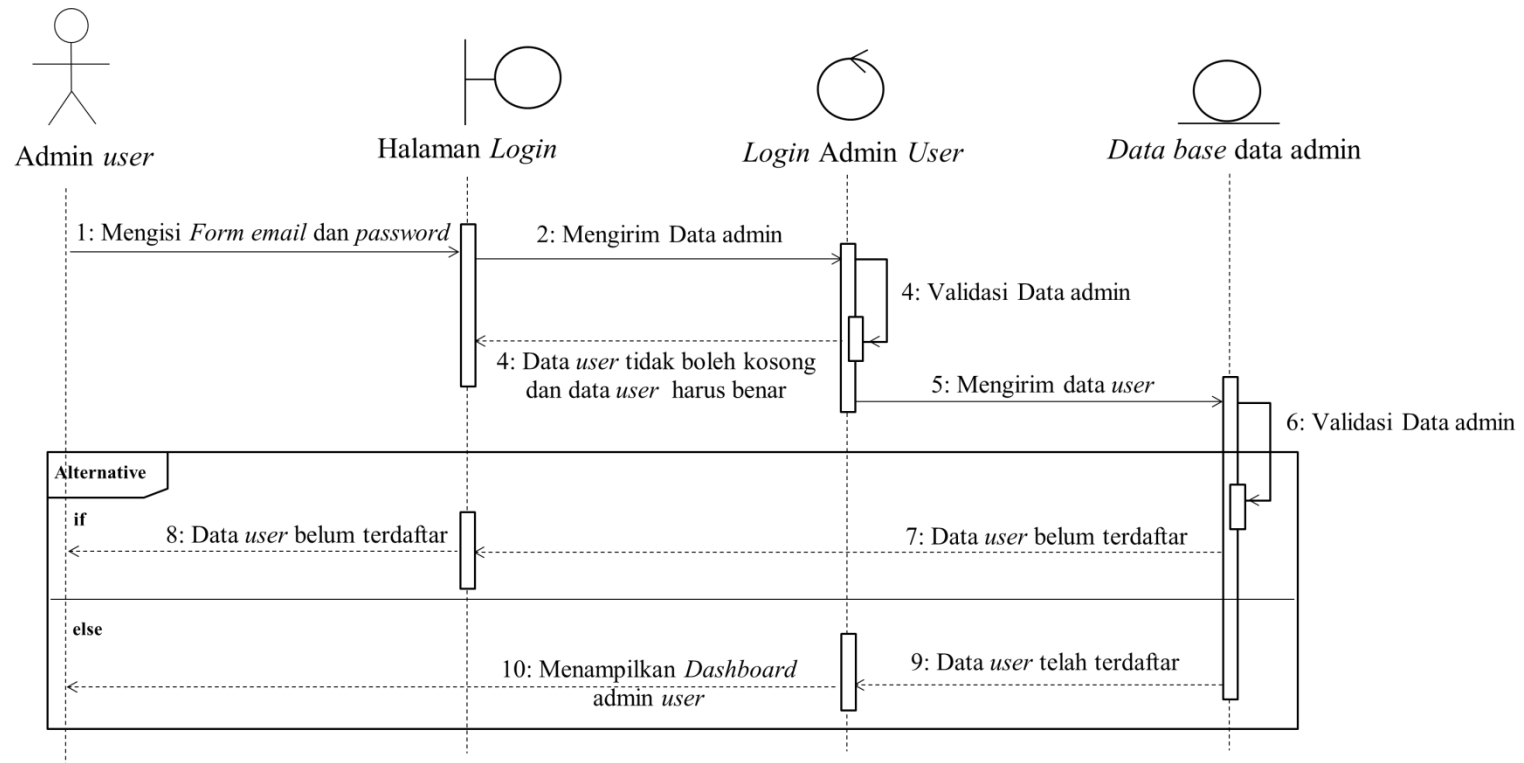
Gambar: 3. 25 Sequence diagram mitra user view data vila

7. Sequence Diagram Mitra User Logout



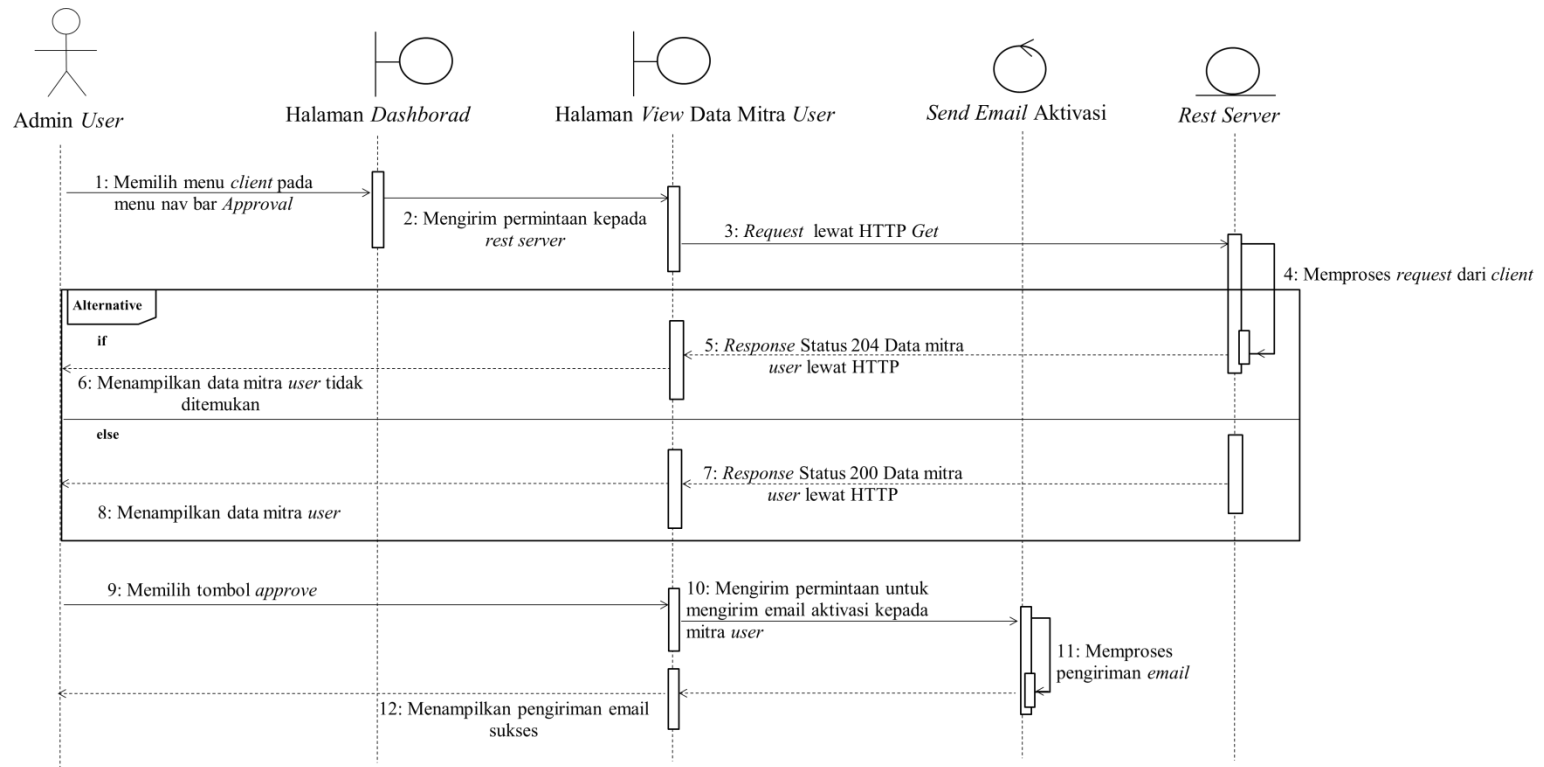
Gambar: 3. 26 Sequence diagram logout mitra user

8. Sequence Diagram Login Admin User



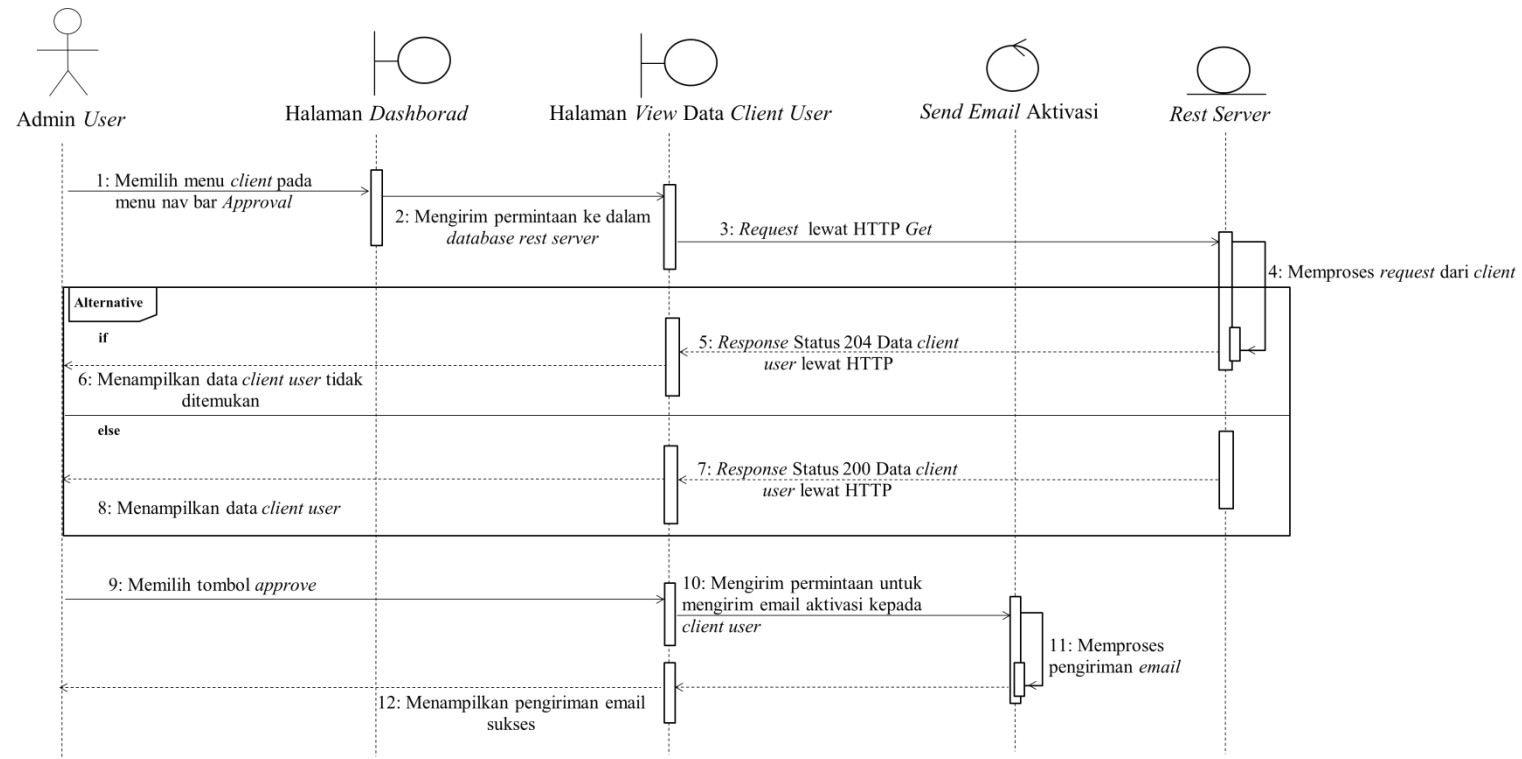
Gambar: 3. 27 Sequence diagram login admin user

9. Sequence Diagram Admin View Data Approval Untuk Mitra User



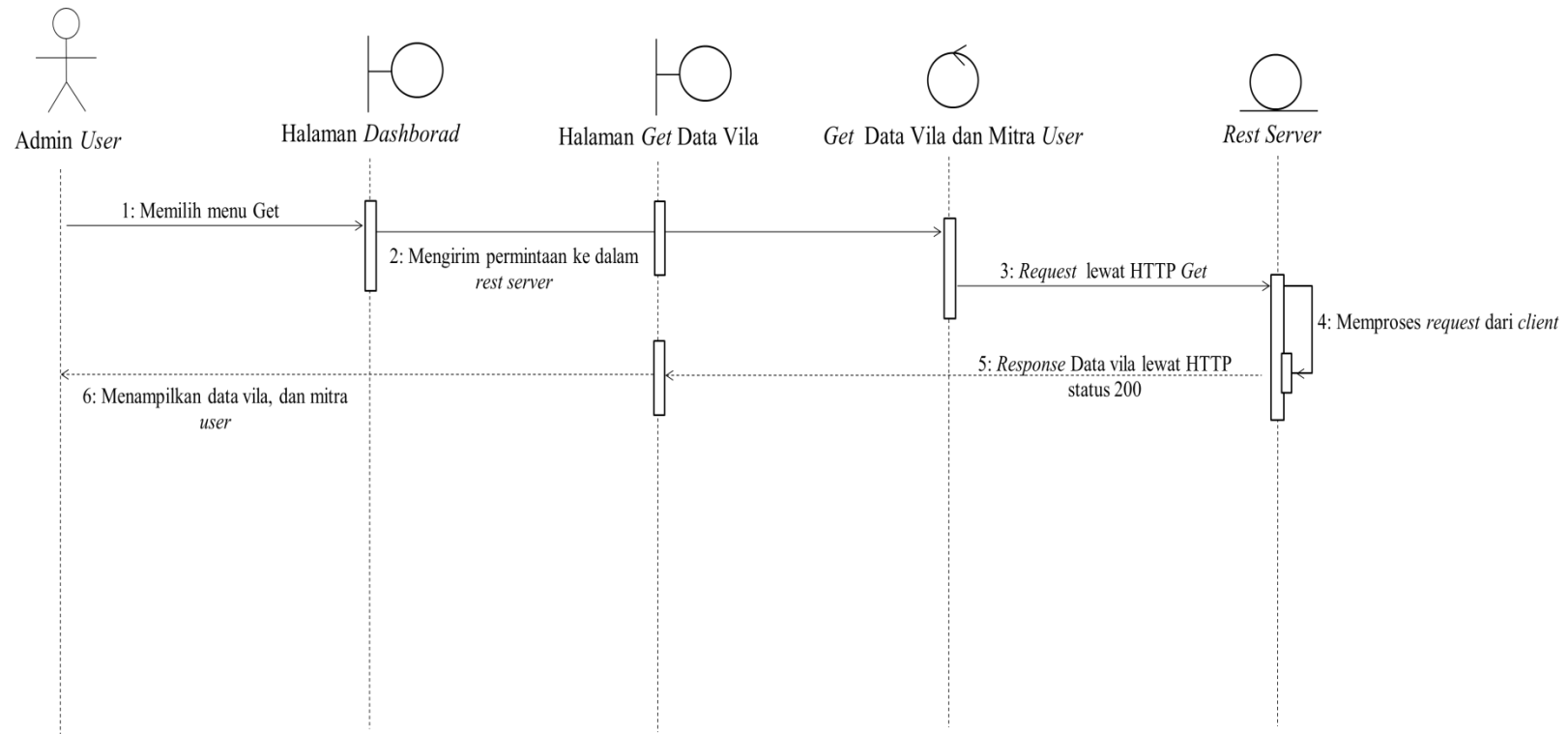
Gambar: 3. 28 Sequence diagram admin view data approval untuk mitra user

10. Sequence Diagram Admin View Data Approval Untuk Client User



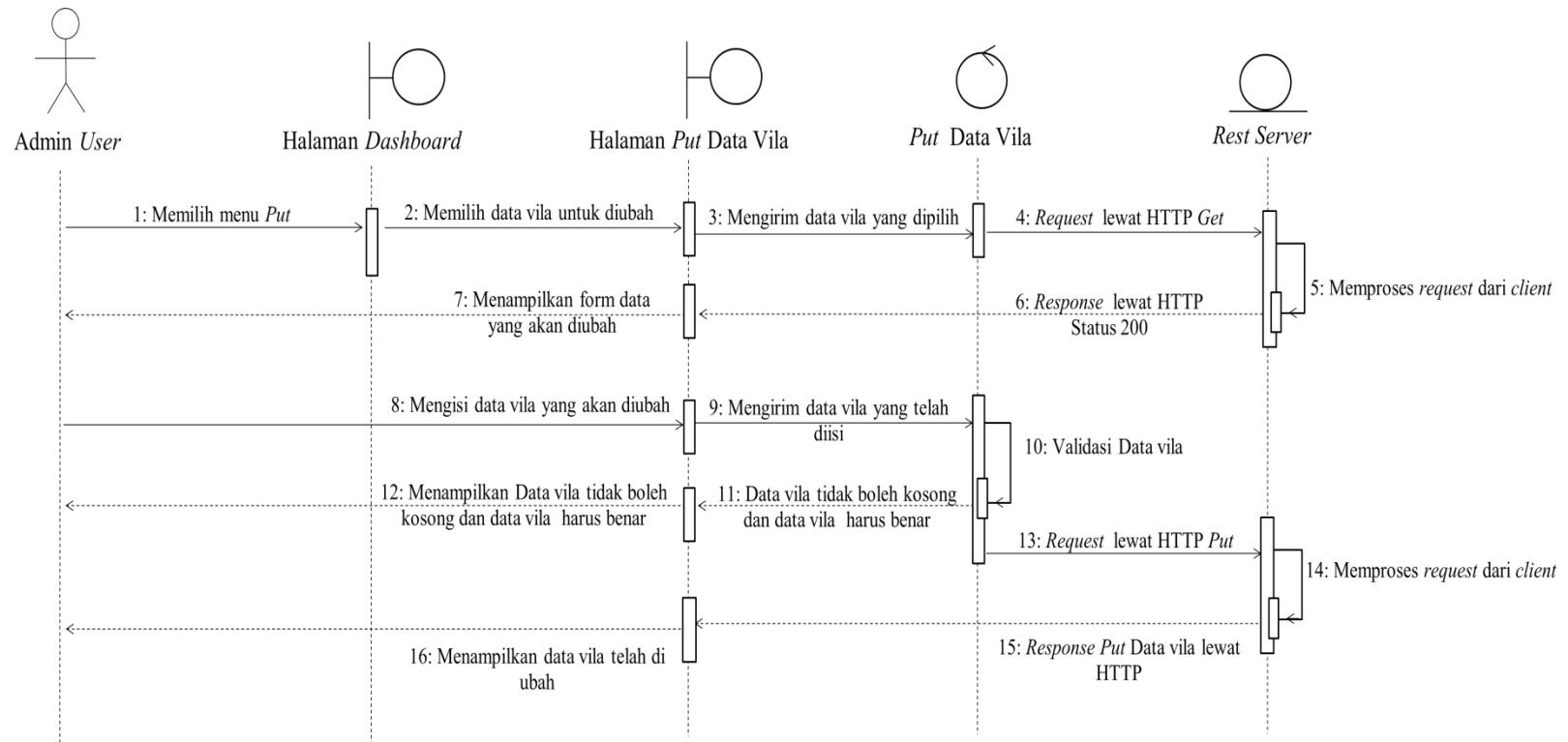
Gambar: 3. 29 Sequence diagram admin view data approval untuk client user

11. Sequence Diagram Admin User Get Data Vila



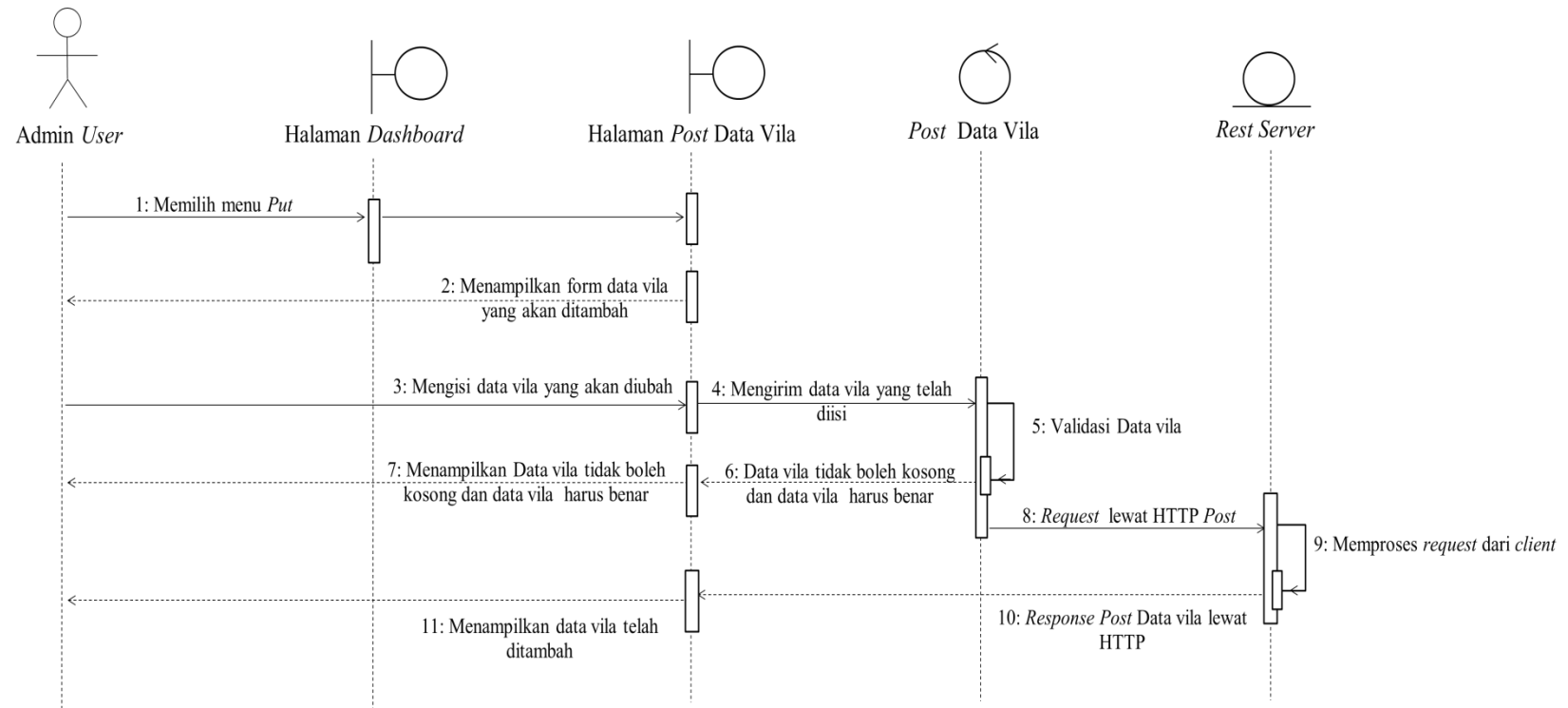
Gambar: 3. 30 Sequence get data vila dan mitra user

12. Sequence Diagram Admin User Put Data Vila



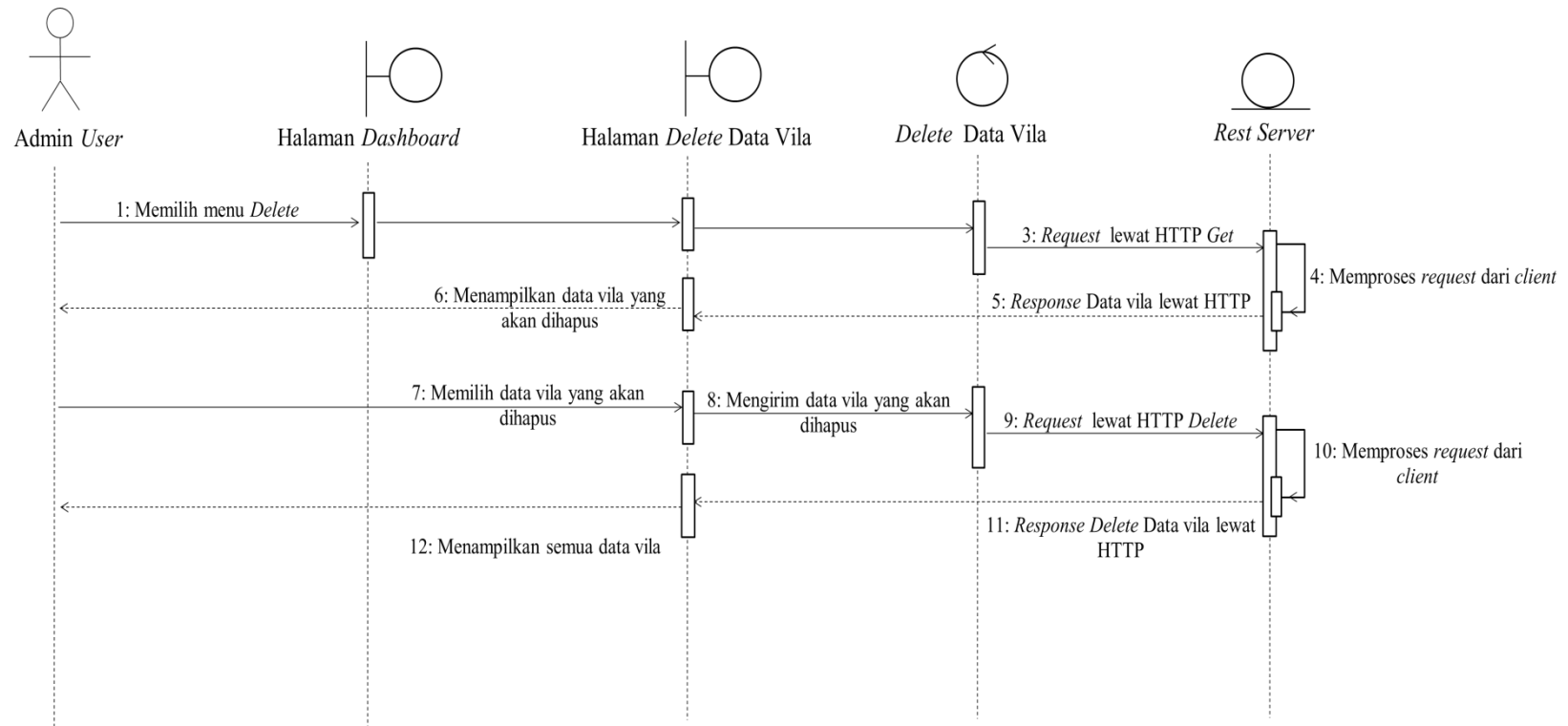
Gambar: 3. 31 Sequence diagram admin user put data vila

13. Sequence Diagram Admin User Post data vila



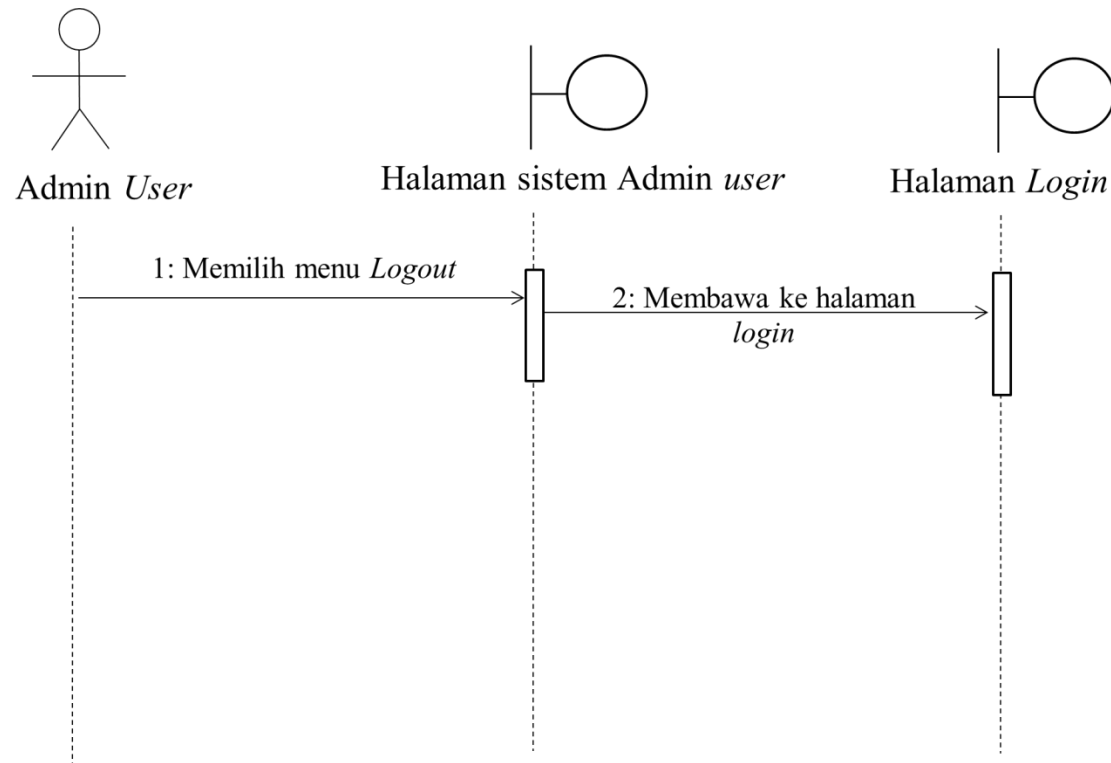
Gambar: 3. 32 Sequence diagram post data vila

14. Sequence Diagram Admin User Delete data vila



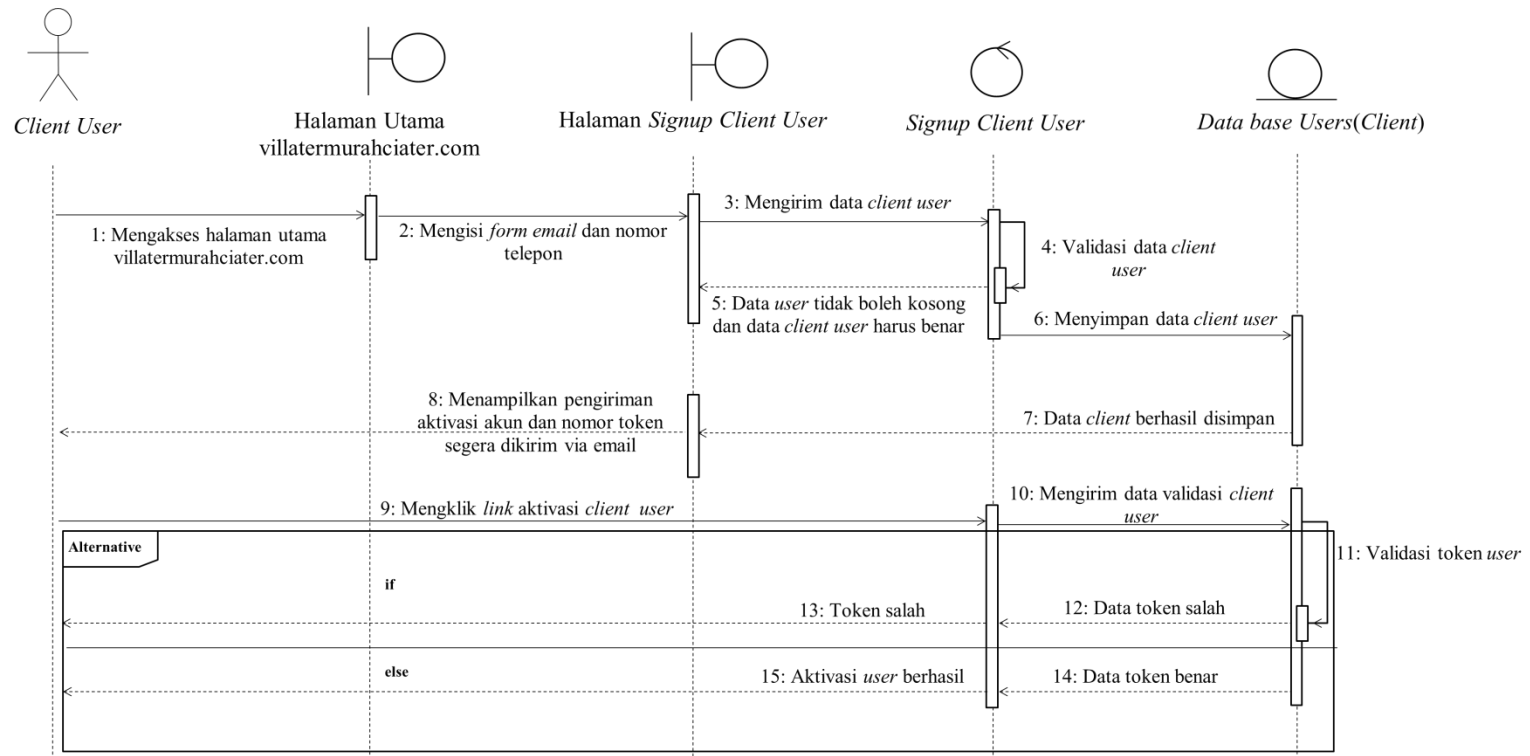
Gambar: 3. 33 Activity diagram delete data vila

15. Sequence Diagram Logout Admin User



Gambar: 3. 34 Sequence diagram logout admin user

16. Sequence Diagram Sign up Client User

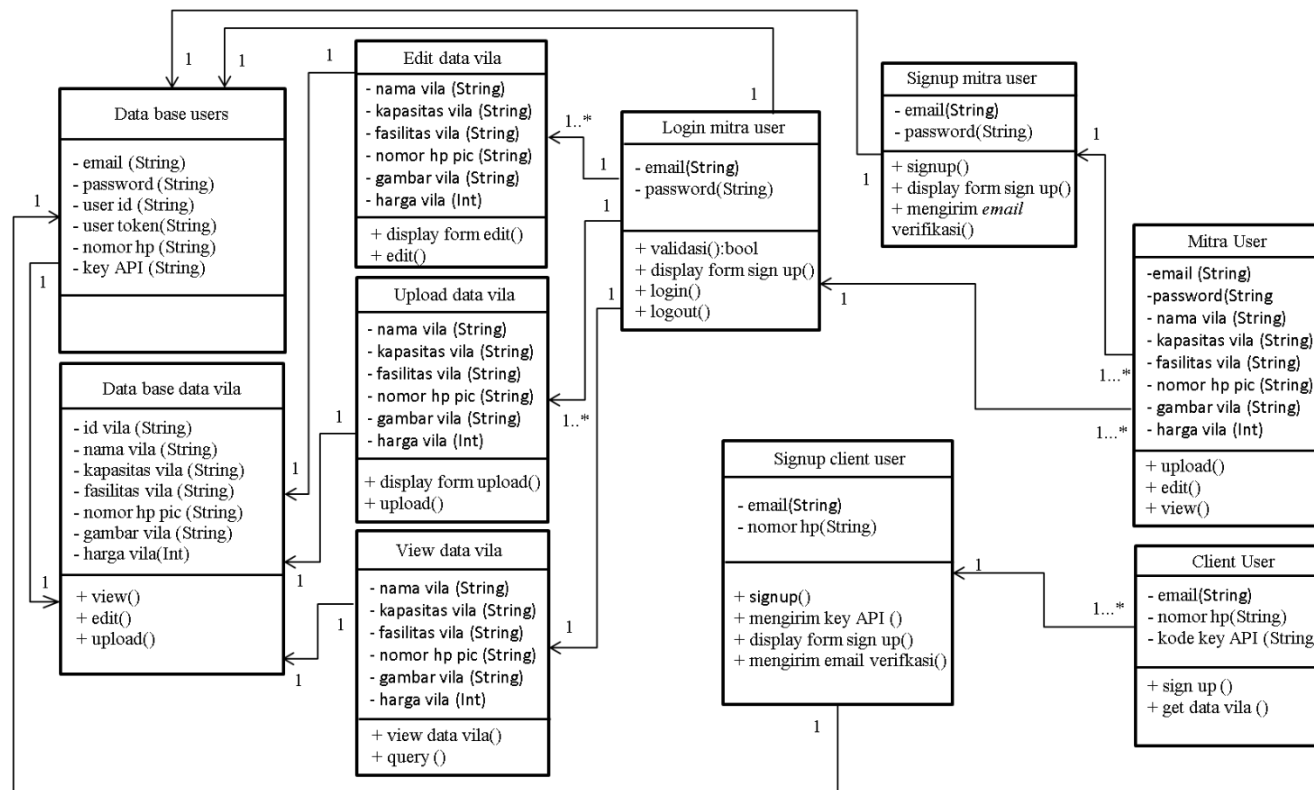


Gambar: 3. 35 Sequence diagram sign up client user

3.2.2.5 Class Diagram

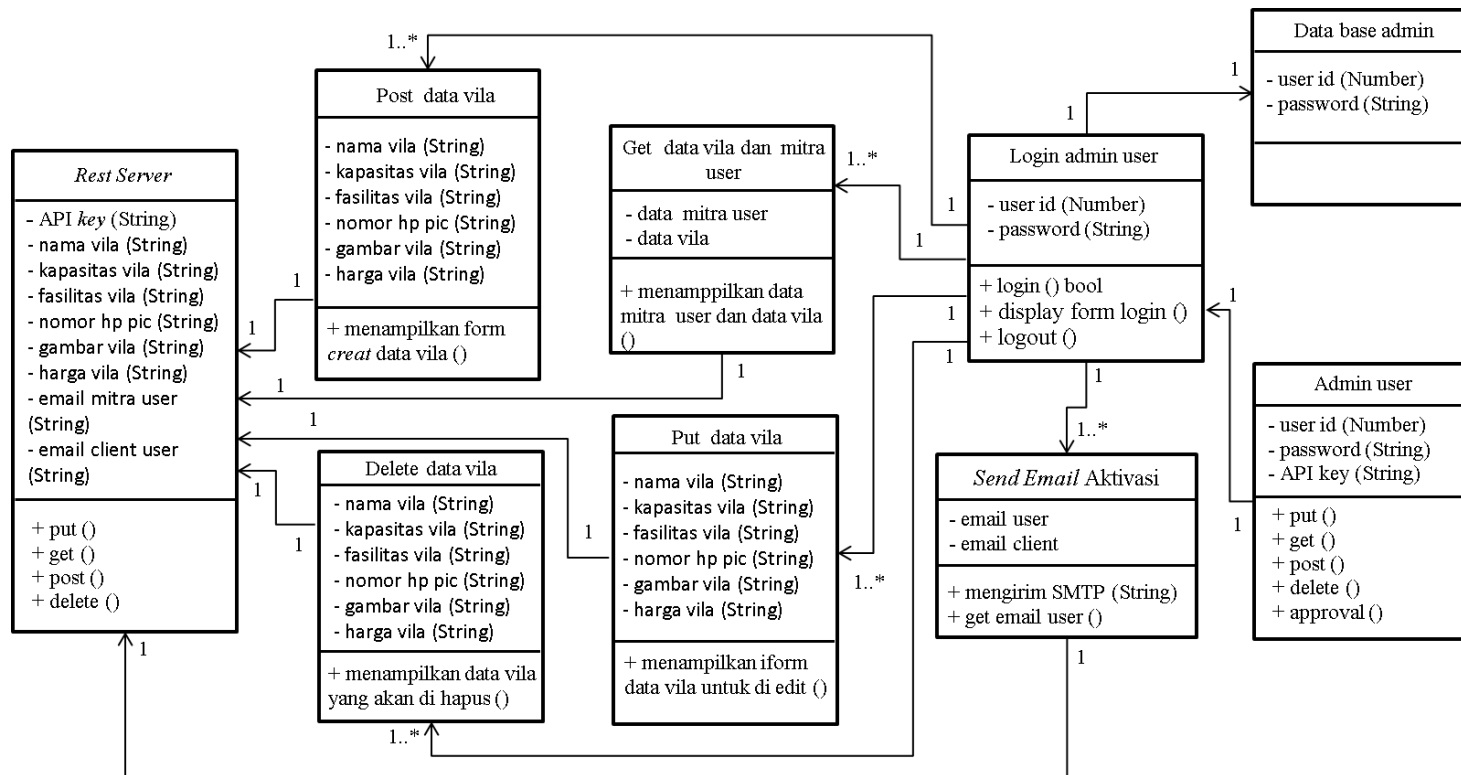
Class Diagram menggambarkan serta deskripsi atau penggambaran dari class, atribut, dan objek disamping itu juga hubungan satu sama lain seperti pewarisan, containmet, asosiasi dan lainnya, Berikut adalah gambar *class diagram* sistem.

1. Class Diagram Mitra User dan Client User



Gambar: 3. 36 Class diagram sistem informasi villatermurahciater.com

2. Class Diagram Sistem Admin villatermurahciater.com



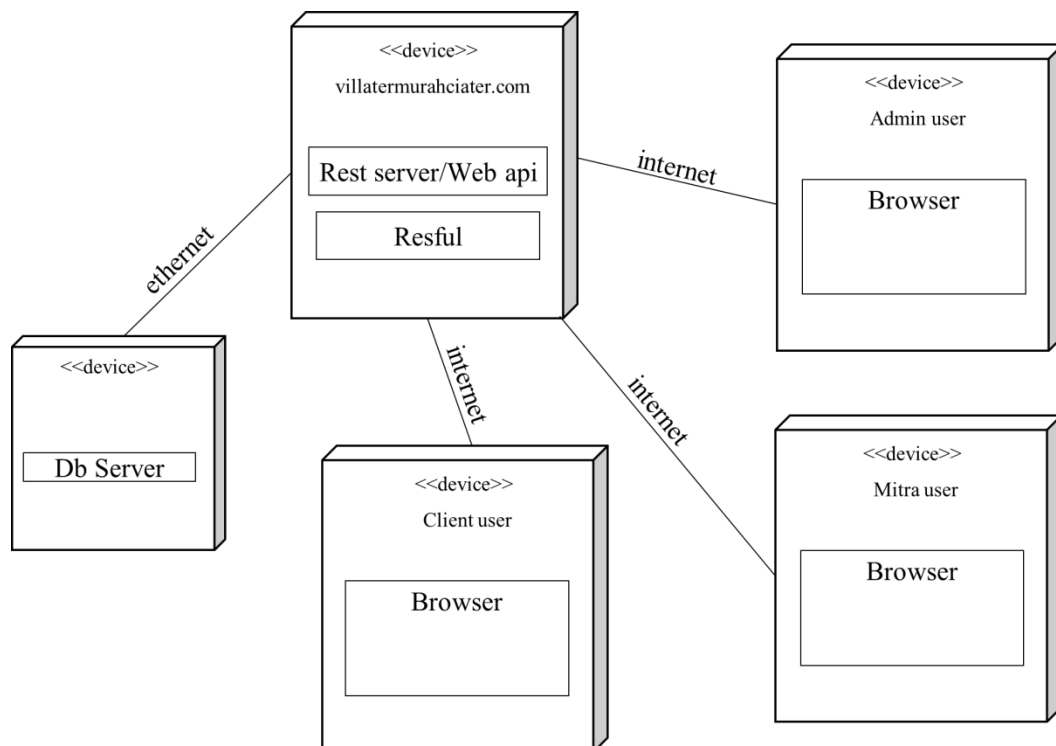
Gambar: 3. 37 Class diagram sistem admin villatermurahciater.com

3.2.2.6 Class Responsibility collaborations (CRC)

Tabel: 3. 24 *Collaboration Responsibility Card (CRC)*

Menu Utama	
<i>Responsibilitie</i>	<i>Collabolator</i>
Melihat data vila	Database vila
upload data vila	Class upload, Database vila
edit data vila	Class edit, Database vila
sign up	Class sign up, Database vila
login	Class login, Database vila
logout	Class logout

3.2.2.7 Analisis *Deployment Diagram*

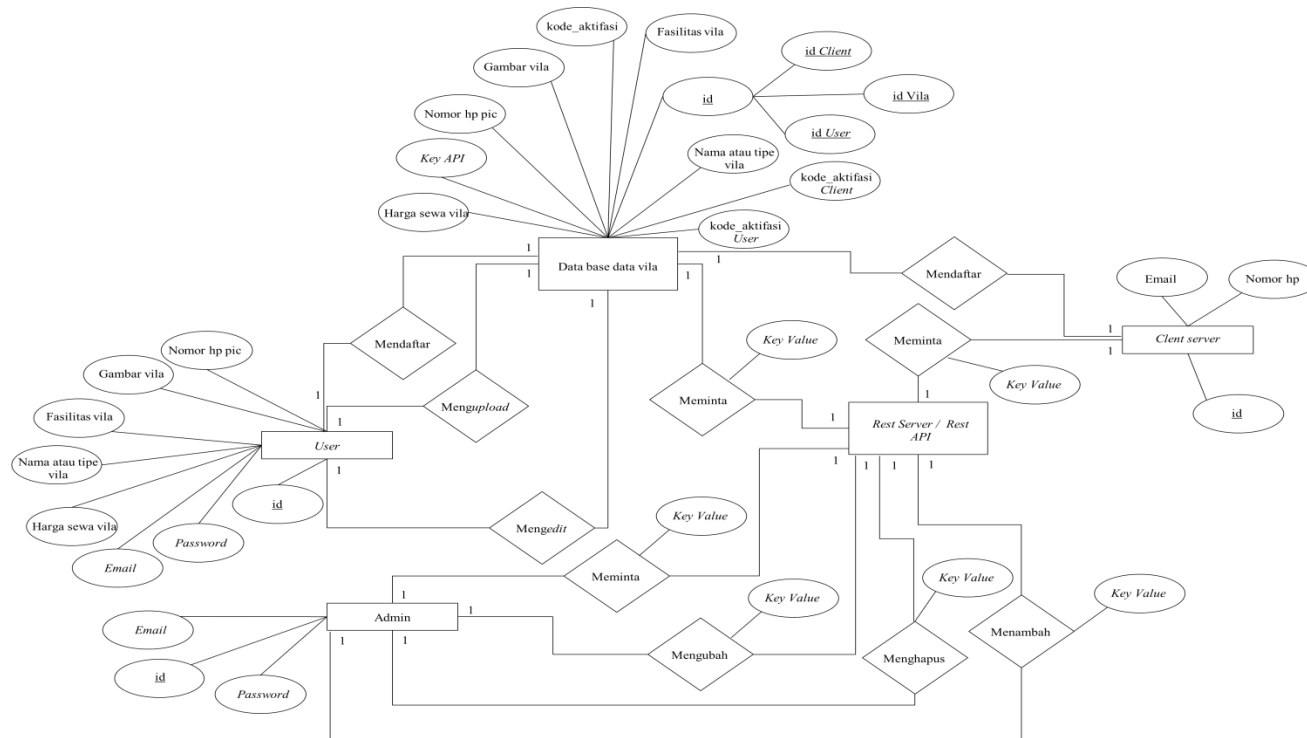


Gambar: 3.38 *Deployment diagram* sistem informasi villatermurahciater.com, Admin user, Mitra user, Client user

3.2.2.8 Analisis Database

Analisis basis data merupakan salah satu pendukung aplikasi yang penting sebagai penyimpanan data. Analisis basis data membahas tentang perancangan basis data mulai dari ERD sampai skema relasi. ERD yang digunakan dalam merancang basis data untuk menggambarkan relasi atau hubungan antar tabel. ERD terdiri dari komponen utama yaitu entitas, relasi, kardinalitas, atribut. Komponen tersebut dideskripsikan lebih jauh melalui atribut-atribut atau properti. ERD dapat dilihat pada gambar 3. 39 sebagai berikut:

1. Entity Relational Diagram



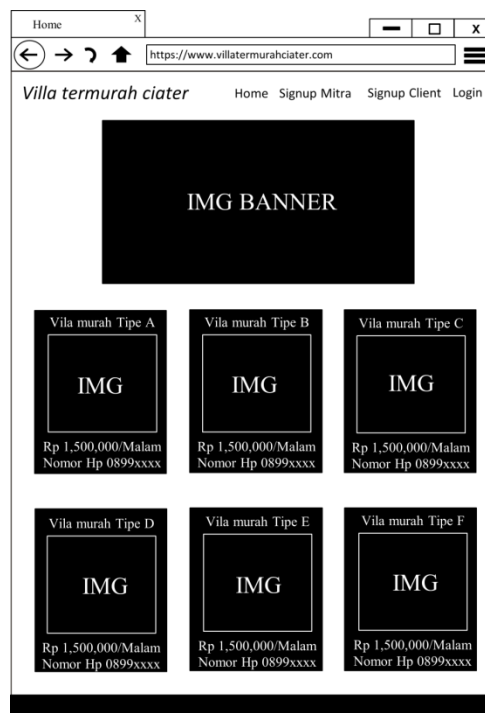
Gambar: 3. 39 Entity Relational Diagram villatermurahciater.com

3.3 Perancangan Antarmuka

Perancangan antarmuka adalah sebuah gambaran dimana nantinya bagaimana tampilan sistem yang akan dibangun, hal ini sangat diperlukan sebagai acuan pengembang dalam membangun sistem dan agar nantinya pengguna sistem tidak kebingungan dalam mengoperasikan sistem. Berikut adalah perancangan antarmuka sistem.

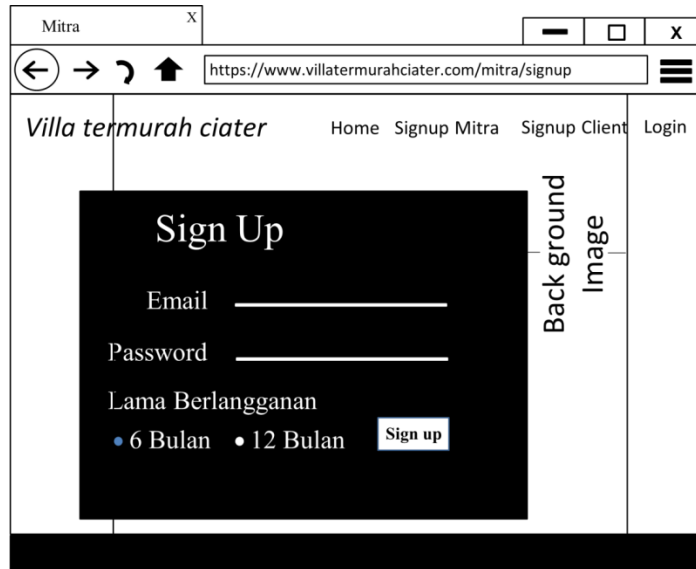
Interface pada sistem disajikan dalam bentuk gambar 3.40 sampai dengan gambar 3.58

1. Halaman utama *web site villatermurahciater.com*



Gambar: 3. 40 Halaman utama sistem informasi villatermurahciater.com

2. Halaman Sign Up Mitra User



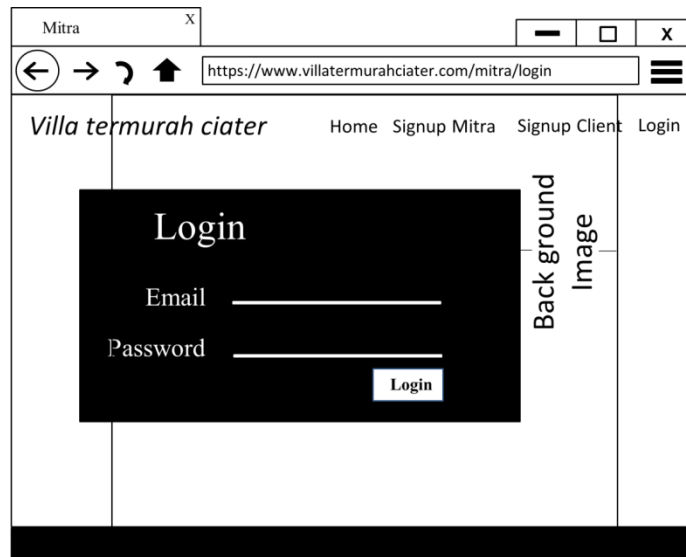
The screenshot shows a web browser window with the URL <https://www.villatermurahciater.com/mitra/signup>. The page title is "Villa termurah ciater". The navigation menu includes "Home", "Signup Mitra", "Signup Client", and "Login". The main content area features a black sign-up form with the following fields and options:

- Sign Up** (Title)
- Email** (Text input field)
- Password** (Text input field)
- Lama Berlangganan** (Subscription duration options):
 - 6 Bulan
 - 12 Bulan
- Sign up** (Submit button)

A vertical label "Back ground Image" is positioned to the right of the form.

Gambar: 3. 41 Halaman *sign up* mitra user

3. Halaman Login Mitra User



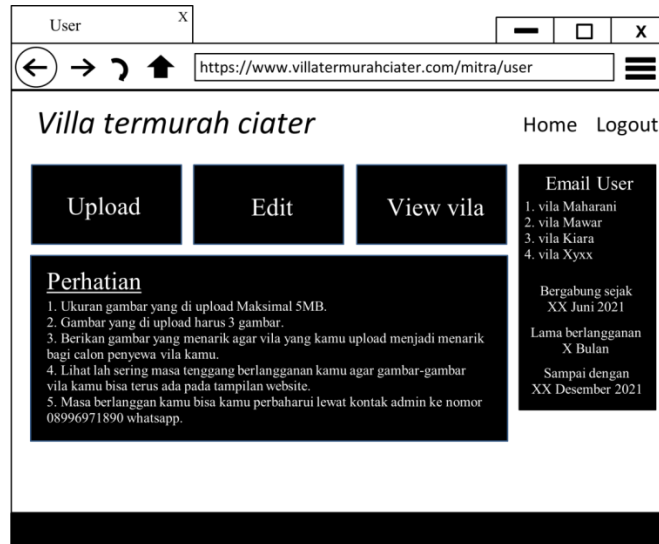
The screenshot shows a web browser window with the URL <https://www.villatermurahciater.com/mitra/login>. The page title is "Villa termurah ciater". The navigation menu includes "Home", "Signup Mitra", "Signup Client", and "Login". The main content area features a black login form with the following fields and options:

- Login** (Title)
- Email** (Text input field)
- Password** (Text input field)
- Login** (Submit button)

A vertical label "Back ground Image" is positioned to the right of the form.

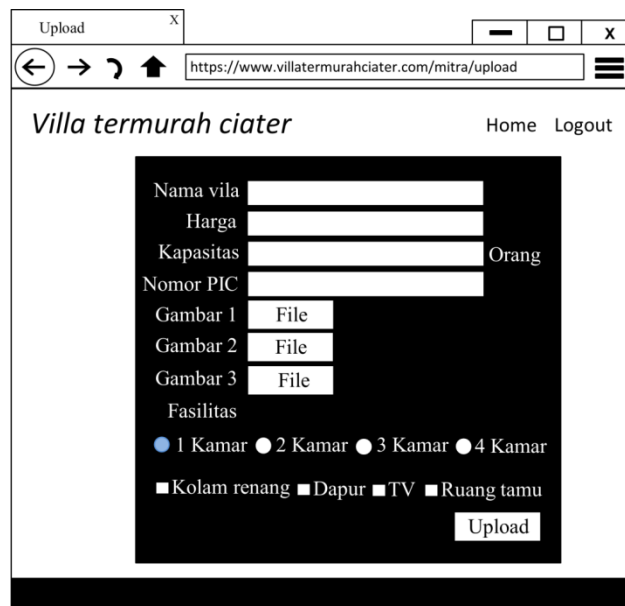
Gambar: 3. 42 Halaman *login* mitra user

4. Halaman *Dashboard Mitra User*



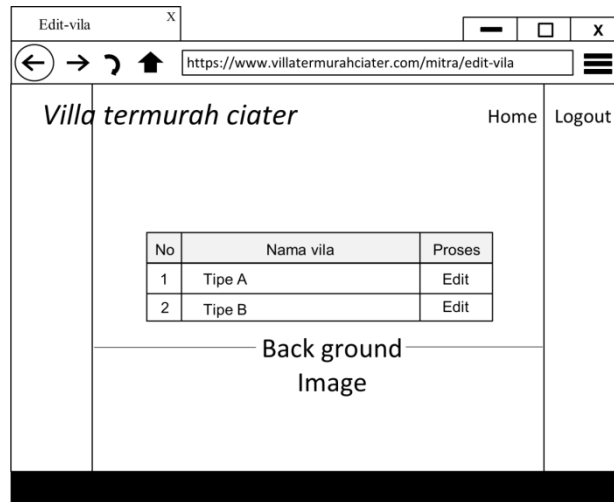
Gambar: 3. 43 Halaman *dashboard mitra user*

5. Halaman Menu *Upload Mitra User*



Gambar: 3. 44 Halaman menu *upload vila mitra user*

6. Halaman Menu *Edit Vila Mitra User*



Gambar: 3. 45 Halaman menu *edit vila mitra user*

7. Halaman Menu *Form Edit Vila Mitra User*

Nama vila

Harga

Kapasitas Orang

Nomor PIC

Gambar 1 File

Gambar 2 File

Gambar 3 File

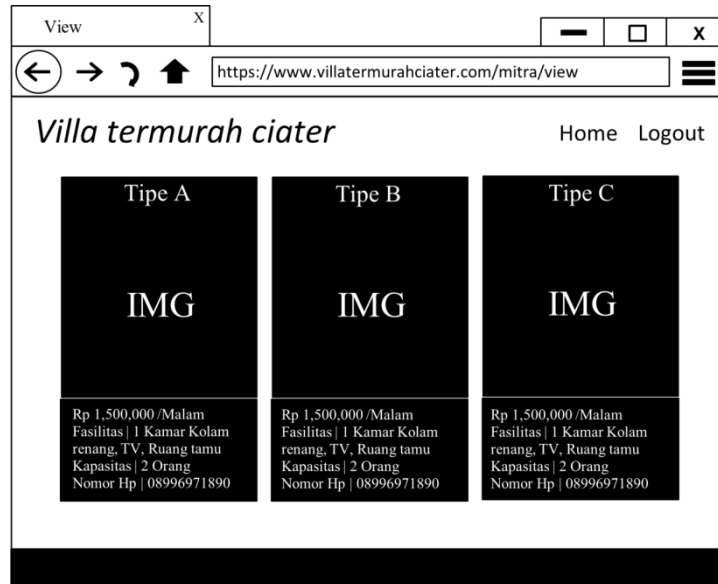
Fasilitas

1 Kamar 2 Kamar 3 Kamar 4 Kamar

Kolam renang Dapur TV Ruang tamu

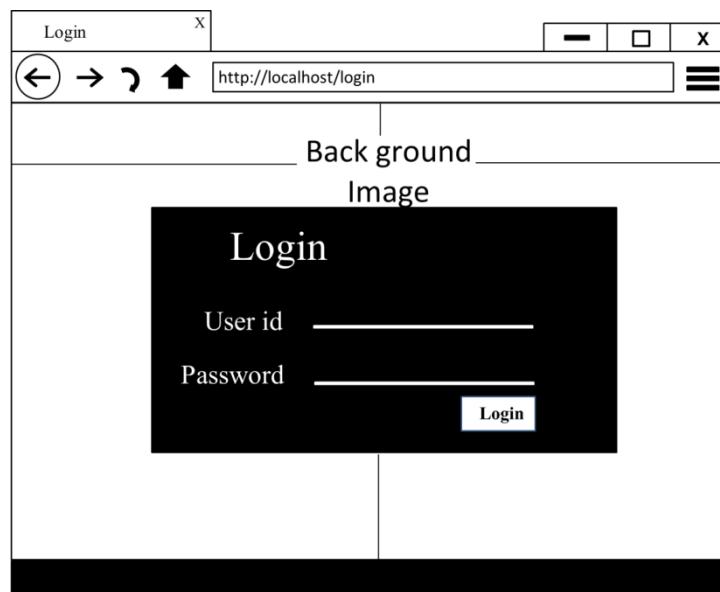
Gambar: 3. 46 Halaman menu *form edit vila mitra user*

8. Halaman Menu *View Vila Mitra User*



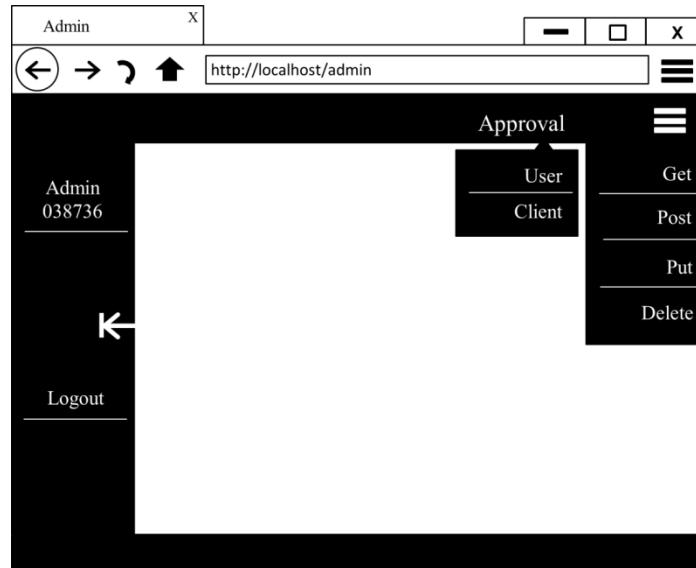
Gambar: 3. 47 Halaman menu *view vila mitra user*

9. Halaman *Login Admin User*



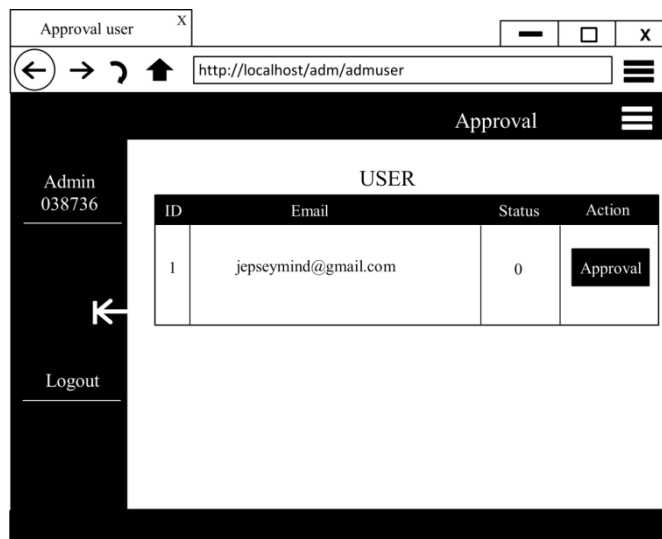
Gambar: 3. 48 Halaman *login admin user*

10. Halaman *Dashboard Admin User*



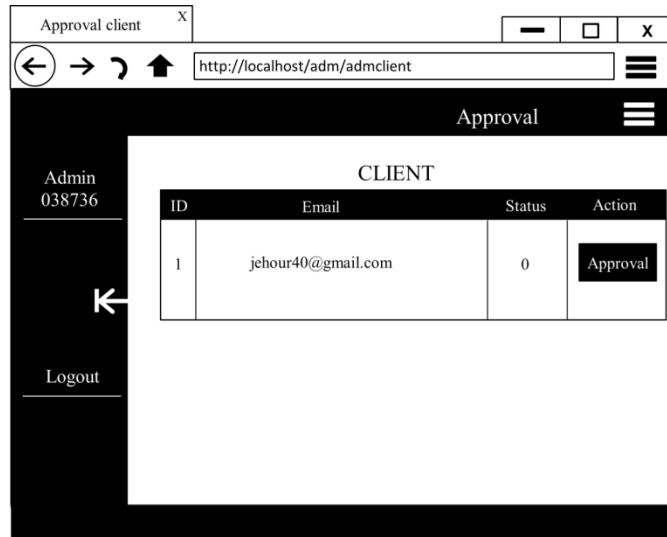
Gambar: 3. 49 Halaman *dashboard* admin user

11. Halaman *Approval Akun Mitra User*



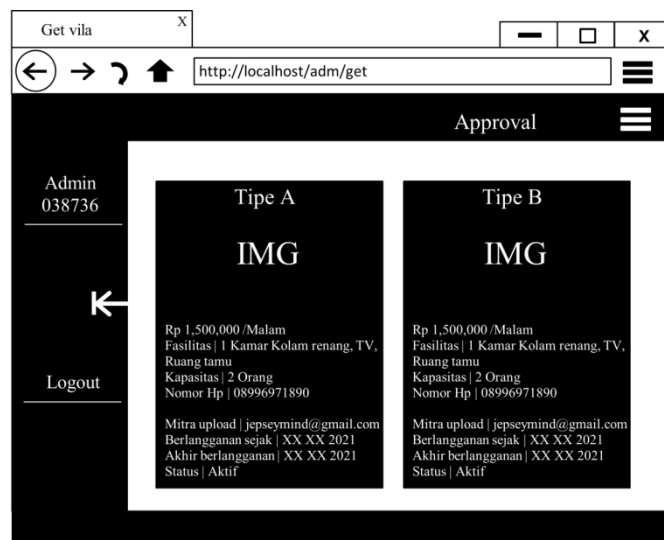
Gambar: 3. 50 Halaman admin aktivasi akun mitra user

12. Halaman Admin *Approval Akun Client User*



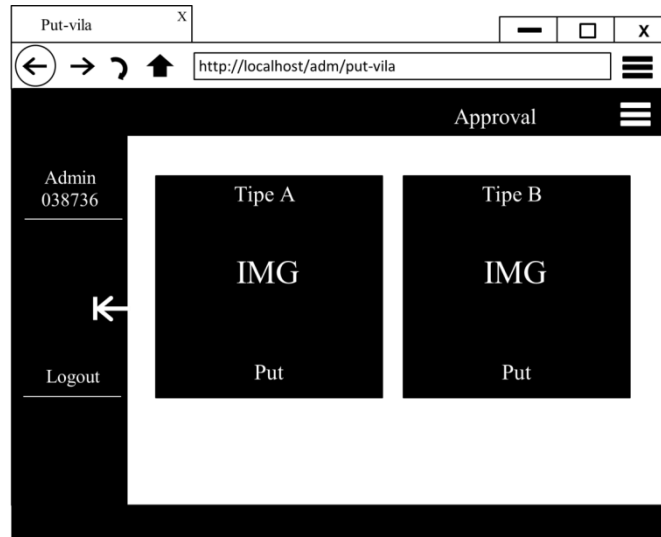
Gambar: 3. 51 Halaman admin *approval* akun *client user*

13. Halaman Admin *Get Data Vila*



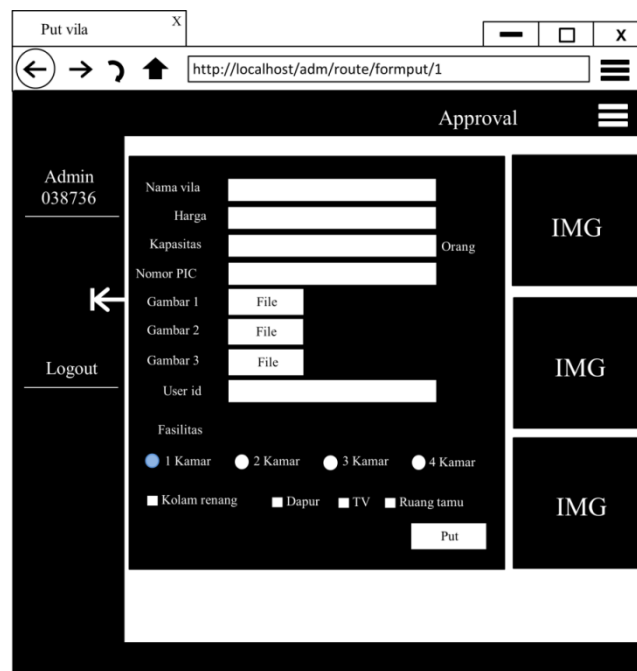
Gambar: 3. 52 Halaman admin *get* data vila

14. Halaman Admin *Put* Data Vila



Gambar: 3. 53 Halaman *put* data vila

15. Halaman Admin *Form Put* Data Vila



Gambar: 3. 54 Halaman admin *form put* data vila

16. Halaman Admin *Post* Data Vila

Post vila

Approval

Admin 038736

Logout

Nama vila

Harga

Kapasitas Orang

Nomor PIC

Gambar 1

Gambar 2

Gambar 3

User id

Fasilitas

1 Kamar 2 Kamar 3 Kamar 4 Kamar

Kolam renang Dapur TV Ruang tamu

Post

Gambar: 3. 55 Halaman admin *post* data vila

17. Halaman Admin *Delete* Data Vila

Delete vila

Approval

Admin 038736

Logout

Tipe A

IMG

Delete

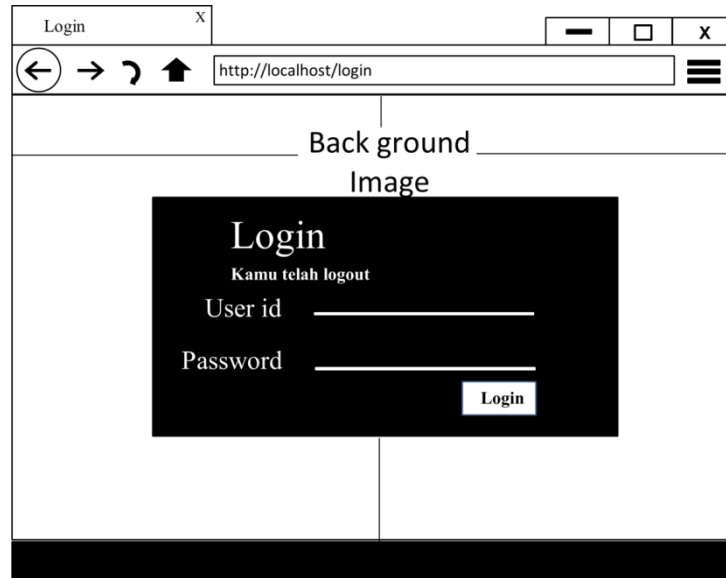
Tipe B

IMG

Delete

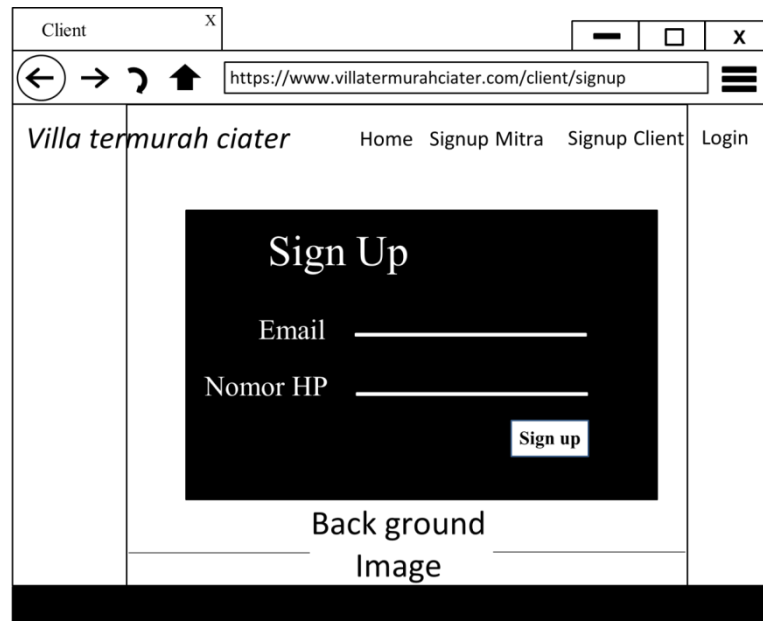
Gambar: 3. 56 Halaman admin *delete* data vila

18. Halaman *Logout Admin*



Gambar: 3. 57 Halaman *logout admin*

19. Halaman *Sign up Client User*



Gambar: 3. 58 Halaman *sign up client user*

BAB IV

IMPLEMENTASI DAN UJI COBA

4.1 Implementasi

Pada tahap ini penulis melakukan implementasi terhadap rancangan dan desain sistem yang telah dibuat sebelumnya kemudian tahapan selanjutnya sistem tersebut akan diuji dengan metode pengujian sistem yaitu *blackbox testing* untuk menguji sistem yang telah dibuat.

Dalam implementasi sistem penulis menggunakan dukungan yaitu *Text Editor Sublime text, Apache 7.2, DBMS MySQL, Frame work Code Igniter, Compossor Json, Library Rest API, Web browser Firefox dan Crome* .

4.1.1 Implementasi Perangkat Keras

Adapun spesifikasi minimum dan perangkat keras yang digunakan untuk mendukung implementasi sistem dalam implementasi perancangan diatas sebagai berikut:

1. Laptop

Spesifikasi laptop yang digunakan dalam implementasi sistem sebagai berikut:

- OS : Windows 7 Ultimate
- RAM: 4GB
- Processor: Intel (R) Core (TM) i3-2330M CPU @ 2.20GHz

4.1.2 Implementasi Perangkat Lunak

Adapun perangkat lunak pendukung dalam implementasi perancangan sistem diatas yaitu:

- *Fire fox Last Version* 84.0.2 (64-bit)
- *Google Chrome* Version 87.0.4280.141 (Official Build) (32-bit)

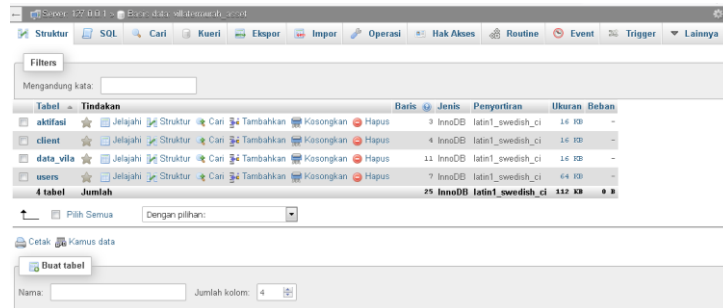
4.1.3 Implementasi Database

Berdasarkan perancangan *database* dan diagram yang telah dipaparkan sebelumnya, maka berikut adalah implementasi *Data Base Management System (DBMS)* untuk sistem informasi villatermurahciater.com sebagai berikut:

4.1.3.1 Struktur Tampilan Database Utama

Struktur ini diset untuk memenuhi data-data vila yang menampung semua data

1. Implementasi struktur basis data vila



Gambar: 4. 1 Struktur utama *database* sistem informasi villatermurahciater.com

4.1.3.2 Struktur Tabel *Database Admin User*

Struktur tabel *database* ini merupakan struktur yang menyiapkan informasi admin *user*.

1. Implementasi *database* admin *user*



Gambar: 4. 2 Struktur *database* admin *user*

4.1.3.3 Struktur Tabel *Database Vila*

Merupakan struktur tabel *database* yang menyiapkan data-data dan informasi tentang vila.

1. Implementasi Tabel *Database* Data Vila

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	namavila	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	harga	varchar(12)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	kapasitas	varchar(11)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	hp	varchar(12)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
6	gambar1	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
7	gambar2	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
8	gambar3	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
9	fasilitas	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
10	userid	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar: 4. 3 Struktur *database* vila

2. Implementasi Tabel *Client* User

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	email	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	password	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	is_active	varchar(11)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	lamasewa	varchar(11)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
6	createdate	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar: 4. 4 Struktur tabel *client* user

3. Struktur Tabel Database Aktivasi Akun

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	email	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	kode_aktivasi	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	create_date	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

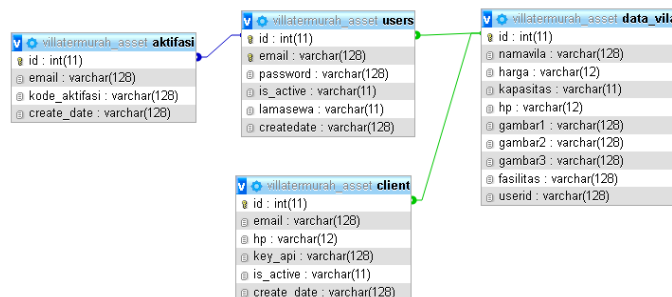
Gambar: 4. 5 Struktur tabel aktivasi

4. Struktur Tabel *Database Client User*

#	Nama	Jenis	Penyortiran	Atribut	Tak Terbilang	Batasan	Komentar Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Lainnya
2	email	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
3	hp	varchar(12)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
4	key_api	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
5	is_active	varchar(11)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
6	create_date	varchar(128)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya

Gambar: 4. 6 Struktur *database client user*

5. Implementasi Relasi Antar Tabel



Gambar: 4. 7 Implementasi relasi antar tabel

4.1.4 Implementasi Antarmuka

Pada tahap ini yang dilakukan berdasarkan hasil perancangan diatas, adapun antarmuka yang diimplementasikan dalam sistem sebagai berikut:

4.1.4.1 Antarmuka Halaman Utama

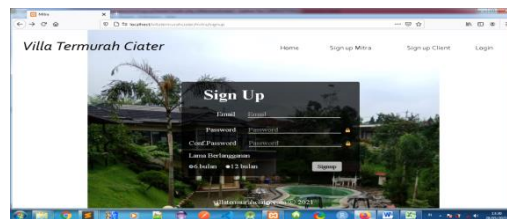
Antarmuka yang ditampilkan oleh sistem pada saat calon penyewa vila dan admin akan masuk ke dalam sistem.



Gambar: 4. 8 Antarmuka halaman utama

4.1.4.2 Antarmuka Halaman *Sign up Mitra User*

Halaman ini setelah *user* memilih menu *sign up*.



Gambar: 4. 9 Antarmuka *sign up* mitra user

4.1.4.3 Antar Muka Halaman *Login Mitra User*

Halaman ini setelah *user* memilih menu *login*.



Gambar: 4. 10 Antarmuka *login* mitra user

4.1.4.4 Antarmuka Halman *Dashboard* Mitra User

Halaman ini akan bisa diakses jika *user* telah terdaftar dan sudah *sign up* dan *login*.



Gambar: 4. 11 Antarmuka *dashboard* mitra *user*

4.1.4.5 Antarmuka Halaman *Upload Data Vila* Mitra User

Halaman ini untuk *user* mengupload data vila.



Gambar: 4. 12 Antarmuka *upload* data vila mitra *user*

4.1.4.6 Antarmuka Halaman *Edit List Data Vila* Mitra User

Halaman ini mitra *user* dapat melihat *list* data yang telah diupload



Gambar: 4. 13 Antramuka halaman *edit list* data vila mitra *user*

4.1.4.7 Antarmuka Halaman *Edit Data Vila Mitra User*

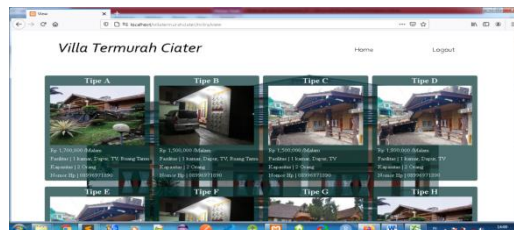
Halaman ini mitra *user* akan mengubah data vila yang telah diupload.



Gambar: 4. 14 Antarmuka halaman *edit* data vila mitra *user*

4.1.4.8 Antarmuka Halaman *View Data Vila Mitra User*

Halaman ini untuk melihat data vila jika *user* telah mengupload data vila.



Gambar: 4. 15 Antarmuka halaman *view* data vila mitra *user*

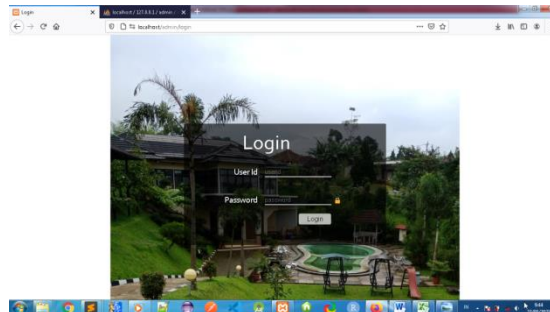
4.1.4.9 Antarmuka Halaman *Logout* Mitra *User*

Halaman antarmuka ini jika *user* telah melakukan *logout* dan membawa ke halaman *login*.



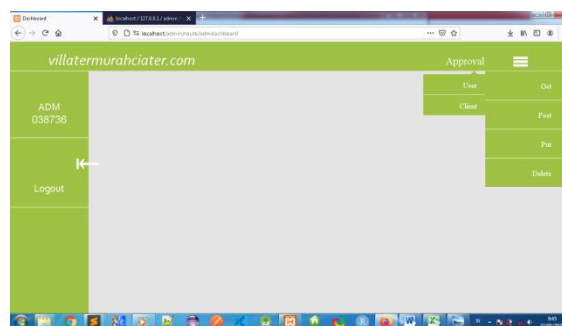
Gambar: 4. 16 Antarmuka halaman *logout* mitra *user*

4.1.4.10 Antarmuka Halaman *Login Admin*



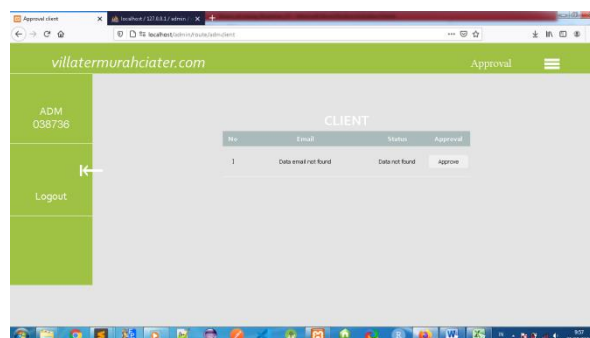
Gambar: 4. 17 Antarmuka halaman admin

4.1.4.11 Antarmuka Halaman *Dashboard Admin*



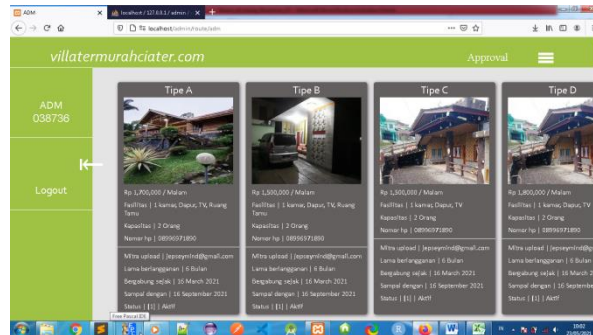
Gambar: 4. 18 Antarmuka halaman *dashboard* admin

4.1.4.12 Antarmuka Halaman Admin *Approve Mitra User*



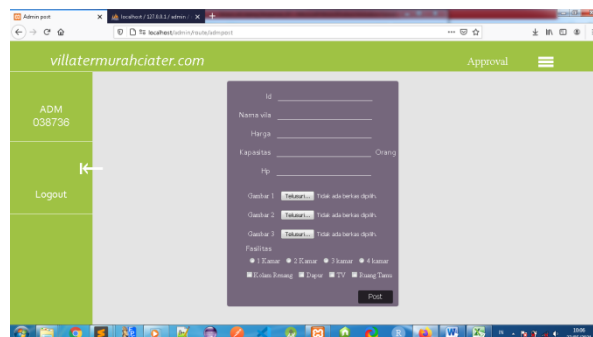
Gambar: 4. 19 Antarmuka halaman admin *approve mitra user*

4.1.4.13 Antarmuka Halaman Admin *Get* data Vila



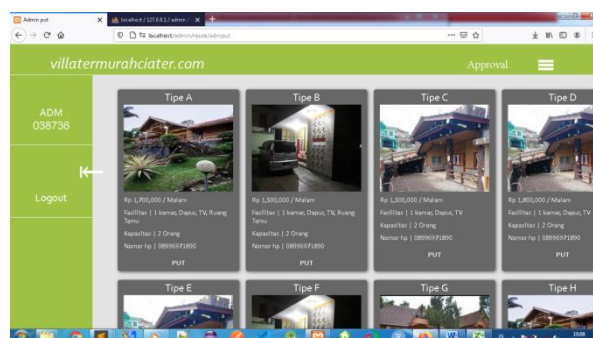
Gambar: 4. 20 Antarmuka halaman admin *get* data vila

4.1.4.14 Antarmuka Halaman Admin *Post* Data Vila



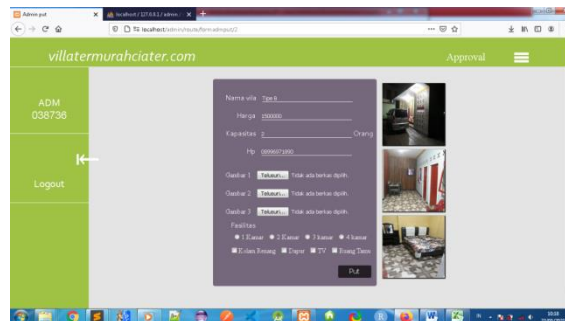
Gambar: 4. 21 Antarmuka halaman admin *post* data vila

4.1.4.15 Antarmuka Halaman Admin *Put* Data Vila



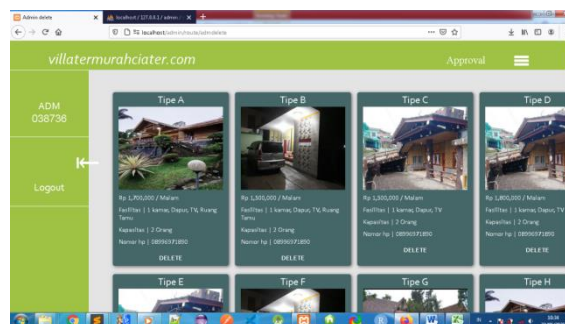
Gambar: 4. 22 Antarmuka halaman admin *put* data vila

4.1.4.16 Antarmuka Halaman Admin *Form Put Data Vila*



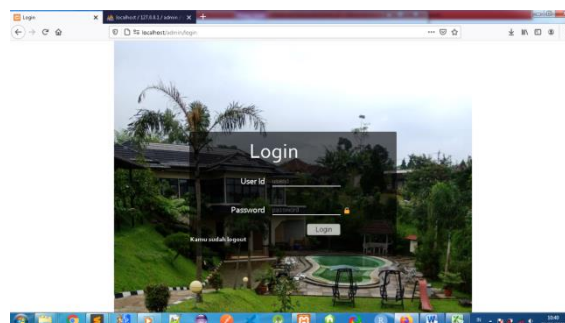
Gambar: 4. 23 Antarmuka halaman admin *form put data vila*

4.1.4.17 Antarmuka Halaman Admin *Delete Data Vila*



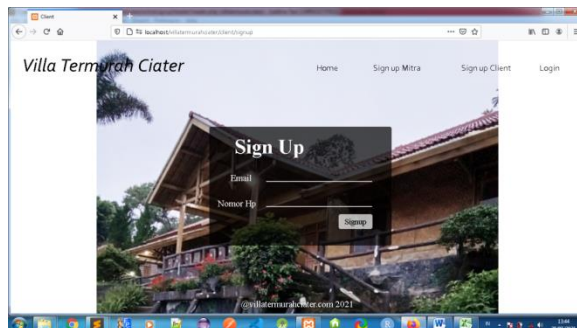
Gambar: 4. 24 Antarmuka halaman admin *delete data vila*

4.1.4.18 Antarmuka Halaman *Logout Admin use*



Gambar: 4. 25 Antarmuka halaman *logout admin user*

4.1.4.19 Antarmuka Halaman *Sign up Client User*



Gambar: 4. 26 Antarmuka halaman *sign up client user*

4.2 Pengujian Sistem

Pada tahap pengujian sistem dilakukan dengan menggunakan metode *Blackbox testing*. Pengujian dilakukan dengan mengakses situs *website villatermurahciater.com* oleh laptop, melihat fitur-fitur sistem dan reaksi *output* yang dihasilkan oleh sistem telah sesuai dengan hasil yang diharapkan. Berikut adalah hasil dari pengujian menggunakan metode *Blackbox Testing*.

4.2.1 Pengujian Tampilan *Dashboard* Halaman Utama

Melakukan pengujian fungsional pada gambar yang ada pada *dashboard* halaman depan. Hasil yang didapatkan disajikan pada tabel dibawah sebagai berikut:

Tabel: 4. 1 Menguji tampilan *dashboard* halaman utama

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji fungsi klik pada gambar	Klik pada gambar	Menampilkan informasi vila dan beberapa gambar vila	Sesuai
2	Menguji fungsi klik pada tombol <i>next image</i>	Klik pada <i>icon</i> '>' dan '<'	Menampilkan gambar berikutnya	Sesuai
3	Menguji fungsi klik pada nomor whatsapp pic	Klik pada nomor whatsapp pic	<i>Direct</i> ke aplikasi whatsapp	Sesuai
4	Menguji fungsi klik pada tombol <i>close</i>	Klik pada <i>icon</i> 'x'	Menampilkan halaman utama	Sesuai
5	Menguji fungsi navigasi halaman utama	Klik menu Home	Menampilkan Halaman utama	Sesuai
6	Menguji fungsi navigasi halaman utama	Klik menu More Information	Tidak menampilkan halaman	Sesuai
7	Menguji fungsi navigasi halaman utama	Klik menu API	Menampilkan halaman API	Sesuai

4.2.2 Pengujian Sistem Mitra User

Melakukan pengujian pada *account user*. Hasil yang didapatkan disajikan pada tabel dibawah ini:

Tabel: 4. 2 Pengujian *sign up* mitra user

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menuji <i>sign up</i> jika <i>email</i> yang dimasukan salah	Memasukan <i>email</i> yang salah	Menampilkan “ <i>The Email field must contain a valid email address.</i> ”	Sesuai
2	Menguji jika <i>email</i> sudah terdaftar	Memasukan <i>email</i> yang telah terdaftar	Menampilkan “ <i>This email has already registered!</i> ”	Sesuai
3	Menguji jika <i>password</i> utama dan konfirmasi <i>password</i> berbeda	Memasukan <i>password</i> utama dengan konfirmasi <i>password</i> berbeda	Menampilkan “ <i>The Password field does not match the Password field.</i> ”	Sesuai
4	Menguji jika kolom <i>email</i> dan <i>password</i> kosong	Tidak memasukan <i>email</i> dan <i>password</i> kosong	Menampilkan “ <i>The Email field is required</i> dan <i>The Password field does not match the Password field.</i> ”	Sesuai

Table:4.2 Pengujian *sign up* mitra user (lanjutan)

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
5	Menguji jika kolom <i>email, password</i> utama dan konfirmasi <i>password</i> benar	Memasukan <i>email, password</i> utama dan konfirmasi <i>password</i> benar	Menampilkan link verifikasi untuk aktivasi akun	Sesuai

4.2.3 Pengujian *Login Mitra User*

Tabel: 4. 3 Pengujian *login* mitra user

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji jika <i>email</i> kosong dan <i>password</i> kosong	Tidak mengisi <i>form login</i>	Menampilkan “ <i>isi isian ini.</i> ”	Sesuai
2	Menguji jika <i>email</i> salah	Mengisi <i>email</i> yang salah	Menampilkan “ <i>Email harus benar.</i> ”	Sesuai
3	Menguji jika <i>email</i> dan <i>password</i> salah	Mengisi <i>email</i> dengan <i>password</i> salah	Menampilkan “ <i>Password salah</i> ”	Sesuai
4	Menguji jika <i>email</i> belum daftar	Mengisi jika <i>email</i> belum terdaftar	Menampilkan “ <i>Email belum terdaftar!</i> ”	Sesuai

Tabel: 4.3 Pengujian *login* mitra *user* (lanjutan)

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
5	Menguji jika <i>email</i> dan <i>password</i> benar	Mengisi <i>email</i> dan <i>password</i> benar	Menampilkan halaman <i>dashboard</i> mitra <i>user</i>	Sesuai

4.2.4 Pengujian *Sign up Client User*

Tabel: 4. 4 Menguji *sign up* mitra *client user*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji jika <i>email</i> dan nomor telepon kosong	Tidak memasukan <i>email</i> dan nomor telepon	Menampilkan “ <i>isi isian ini</i> ”	Sesuai
2	Menguji jika <i>email</i> benar dan nomor telepon kurang dari 11 digit	Memasukan jika <i>email</i> benar dan nomor telepon kurang dari 11 digit	Menampilkan “ <i>Nomor minimal 11 digit</i> ”	Sesuai
3	Menguji jika kolom nomor telepon diisi dengan huruf	Memasukan huruf pada kolom nomor telepon	Menampilkan “ <i>Kolom harus diisi degan nomor</i> ”	Sesuai

Tbale: 4. 4 Menguji *sign up* mitra *client user* (lanjutan)

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
4	Menguji jika kolom <i>email</i> sudah terdaftar	Memasukan <i>email</i> udah terdaftar	Menampilkan “ <i>Email sudah terdaftar</i> ”	Sesuai
5	Menguji mengisi <i>email</i> dan nomer benar	Mengisi <i>email</i> dan nomor telepon	Menampilkan “Registrasi berhasil, email aktivasi akun segera dikirim”	Sesuai

4.2.5 Pengujian Halaman *Dashboard* Mitra *User*

Tabel: 4. 5 Pengujian halaman *dashboard* mitra *user*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji tombol menu <i>upload</i>	Klik tombol menu <i>upload</i>	Menampilkan <i>form upload</i>	Sesuai
2	Menguji tombol menu <i>edit</i>	Klik tombol menu <i>edit</i>	Menampilkan halaman <i>edit</i> data vila	Sesuai
3	Menguji tombol menu <i>view</i> vila	Klik tombol menu <i>view</i> vila	Menampilkan halaman data vila yang sudah diupload	Sesuai

4.2.6 Pengujian *Upload Data Vila Mitra User*

Tabel: 4. 6 Pengujian *upload data vila mitra user*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji jika semua kolom <i>form upload</i> kosong	Tidak memasukan data vila di dalam <i>form upload</i>	Menampilkan “ <i>isi isian ini</i> ”	Sesuai
2	Menguji jika kolom harga, kapasitas, nomor telepon diisi dengan huruf	Memasukan kolom harga, kapasitas dan nomor telepon diisi dengan huruf	Menampilkan “ <i>Kolom harus diisi dengan angka</i> ”	Sesuai
3	Menguji jika kolom nomor telepon diisi dengan huruf	Memasukan huruf pada kolom nomor telepon	Menampilkan “ <i>Kolom harus diisi degan nomor</i> ”	Sesuai
4	Menguji jika kolom gambar diisi dengan bukan <i>file</i> gambar	Memasukan <i>file</i> bukan gambar	Menampilkan “ <i>File yang kamu upload bukan gambar</i> ”	Sesuai
5	Menguji mengisi kolom <i>upload</i> benar	Mengisi kolom <i>upload</i> benar	Menampilkan “ <i>Upload data berhasil</i> ”	Sesuai

4.2.7 Pengujian *Edit Data Vila Mitra User*

Tabel: 4. 7 Pengujian *edit data vila mitra user*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji jika semua kolom <i>form edit</i> kosong	Tidak memasukan data vila di dalam <i>form upload</i>	Menampilkan “ <i>isi isian ini</i> ”	Sesuai
2	Menguji jika kolom harga, kapasitas, nomor telepon diisi dengan huruf	Memasukan kolom harga, kapasitas dan nomor telepon diisi dengan huruf	Menampilkan “ <i>Kolom harus diisi dengan angka</i> ”	Sesuai
3	Menguji jika kolom nomor telepon diisi dengan huruf	Memasukan huruf pada kolom nomor telepon	Menampilkan “ <i>Kolom harus diisi dengan nomor</i> ”	Sesuai
4	Menguji jika kolom gambar diisi dengan bukan <i>file</i> gambar	Memasukan <i>file</i> bukan gambar	Menampilkan “ <i>File yang kamu upload bukan gambar</i> ”	Sesuai
5	Menguji mengisi kolom <i>upload</i> benar	Mengisi kolom <i>edit</i> benar	Menampilkan “ <i>Edit data berhasil</i> ”	Sesuai

4.2.8 Pengujian Menu *Logout Mitra User*

Tabel: 4. 8 Pengujian menu *logout mitra user*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji tombol <i>logout</i>	Klik menu <i>logout</i>	Menampilkan halaman <i>login</i>	Sesuai

4.2.9 Pengujian *Login Admin User*

Tabel: 4. 9 Pengujian *login admin user*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji jika <i>email</i> kosong dan <i>passwod</i> kosong	Tidak mengisi <i>form login</i>	Menampilkan “ <i>isi isian ini.</i> ”	Sesuai
2	Menguji jika <i>user id</i> salah	Mengisi <i>user id</i> yang salah	Menampilkan “ <i>User id harus benar.</i> ”	Sesuai
3	Menguji jika <i>email</i> dan <i>password</i> salah	Mengisi <i>user id</i> dengan <i>password</i> salah	Menampilkan “ <i>Password salah</i> ”	Sesuai

Tabel: 4.9 Pengujian *login admin user* (lanjutan)

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
4	Menguji jika <i>user id</i> belum daftar	Mengisi jika <i>user id</i> belum terdaftar	Menampilkan “ <i>User id belum terdaftar!</i> ”	Sesuai
5	Menguji jika <i>user id</i> dan <i>password</i> benar	Mengisi <i>user id</i> dan <i>password</i> benar	Menampilkan halaman <i>dashboard admin</i>	Sesuai

4.2.10 Pengujian Halaman *Dashboard Admin User*

Tabel: 4. 10 Pengujian halaman *dashboard admin user*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji <i>link</i> submenu <i>user</i> pada menu navbar <i>approval</i>	Klik menu <i>user</i>	Menampilkan halaman <i>approval user</i>	Sesuai
2	Menguji <i>link</i> submenu <i>client</i> pada menu navbar <i>approval</i>	Klik menu <i>client</i>	Menampilkan halaman <i>approval client</i>	Sesuai

Tabel: 4.10 Pengujian halaman *dashboard* admin *user* (lanjutan)

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
3	Menguji <i>link</i> submenu <i>get</i> pada menu <i>icon burger</i>	Klik menu <i>get</i>	Menampilkan halaman <i>get data vila</i>	Sesuai
4	Menguji <i>link</i> submenu <i>put</i> pada menu <i>icon burger</i>	Klik menu <i>put</i>	Menampilkan halaman <i>put data vila</i>	Sesuai
5	Menguji <i>link</i> submenu <i>post</i> pada menu <i>icon burger</i>	Klik menu <i>post</i>	Menampilkan halaman <i>post data vila</i>	Sesuai
6	Menguji <i>link</i> submenu <i>delete</i> pada menu <i>icon burger</i>	Klik menu <i>delete</i>	Menampilkan halaman <i>delete data vila</i>	Sesuai

4.2.11 Pengujian Halaman Menu Approval Untuk *User*

Tabel: 4. 11 Pengujian halaman menu *approval* untuk *user* (mitra *user*)

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji tombol <i>approve</i>	Klik tombol <i>approve</i>	Menampilkan “ <i>email telah dikirim</i> ”	Sesuai

4.2.12 Pengujian Halaman Menu *Approval* Untuk *Client*

Tabel: 4. 12 Pengujian halaman menu *approval* untuk *client* (*client user*)

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji tombol <i>approve</i>	Klik tombol <i>approve</i>	Menampilkan “ <i>email telah dikirim</i> ”	Sesuai

4.2.13 Pengujian Halaman *Post Data Vila*

Tabel: 4. 13 Pengujian halaman *post data vila*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji jika semua kolom <i>form post</i> kosong	Tidak memasukan data vila di dalam <i>form post</i>	Menampilkan “ <i>isi isian ini</i> ”	Sesuai
2	Menguji jika kolom harga, kapasitas, nomor telepon diisi dengan huruf	Memasukan kolom harga, kapasitas dan nomor telepon diisi dengan huruf	Menampilkan “ <i>Kolom harus diisi dengan angka</i> ”	Sesuai
3	Menguji jika kolom nomor telepon diisi dengan huruf	Memasukan huruf pada kolom nomor telepon	Menampilkan “ <i>Kolom harus diisi dengan nomor</i> ”	Sesuai

Tabel: 4. 13 Pengujian halaman *post* data vila (lanjutan)

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
4	Menguji jika kolom gambar diisi dengan bukan <i>file</i> gambar	Memasukan <i>file</i> bukan gambar	Menampilkan “ <i>File yang kamu upload bukan gambar</i> ”	Sesuai
5	Menguji mengisi kolom <i>post</i> benar	Mengisi kolom <i>post</i> benar	Menampilkan “ <i>Post data berhasil</i> ”	Sesuai

4.2.14 Pengujian Halaman *Put* Data Vila

Tabel: 4. 14 Pengujian halaman *put* data vila

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji tombol <i>put</i> pada <i>card</i> info data vila	Klik tombol <i>put</i> pada <i>card</i> info data vila	Menampilkan <i>form put</i> data vila yang telah dipilih	Sesuai

4.2.15 Pengujian Halaman *Form Put Data Vila*

Tabel: 4. 15 Pengujian halaman *form put data vila*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji jika semua kolom <i>form put</i> kosong	Tidak memasukan data vila di dalam <i>form upload</i>	Menampilkan “ <i>isi isian ini</i> ”	Sesuai
2	Menguji jika kolom harga, kapasitas, nomor telepon diisi dengan huruf	Memasukan kolom harga, kapasitas dan nomor telepon diisi dengan huruf	Menampilkan “ <i>Kolom harus diisi dengan angka</i> ”	Sesuai
3	Menguji jika kolom nomor telepon diisi dengan huruf	Memasukan huruf pada kolom nomor telepon	Menampilkan “ <i>Kolom harus diisi dengan nomor</i> ”	Sesuai
4	Menguji jika kolom gambar diisi dengan bukan <i>file</i> gambar	Memasukan <i>file</i> bukan gambar	Menampilkan “ <i>File yang kamu upload bukan gambar</i> ”	Sesuai
5	Menguji mengisi kolom <i>upload</i> benar	Mengisi kolom <i>edit</i> benar	Menampilkan “ <i>Put data berhasil</i> ”	Sesuai

4.2.16 Pengujian Halaman *Delete Data Vila*

Tabel: 4. 16 Pengujian halaman *delete* data vila

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji tombol <i>delete</i> pada <i>card</i> info data vila	Klik tombol <i>delete</i> pada <i>card</i> info data vila	Menampilkan data vila berhasil dihapus dan menampilkan halaman <i>delete</i>	Sesuai

4.2.17 Pengujian Menu *Logout Admin User*

Tabel: 4. 17 Pengujian menu *logout* admin *user*

No	Deskripsi Pengujian	Masukan	Hasil yang Diharapkan	Hasil yang Didapat
1	Menguji tombol <i>logout</i>	Klik menu <i>logout</i>	Menampilkan halaman <i>login</i>	Sesuai

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pembahasan dan penelitian yang telah dilakukan, maka didapatkan beberapa kesimpulan sebagai berikut:

Adanya gaya arsitektur *SOA (Service Oriented Architecture)* serta gaya perancangan *REST (Representational State Transfer)* dan *RESTful* sangatlah membantu untuk menentukan konsep penyediaan layanan informasi serta data *villatermurahciter.com* bagi calon mitra yang akan berkerja sama dalam usaha dibidang layanan jasa. Pertukaran informasi dan data menjadi sangat mudah, efisien, fleksibel, bisa digunakan oleh *cross platform*. Sehingga dapat dikembangkan kedalam sistem informasi lain yang lebih kompleks dan besar.

5.2 Saran

Melihat dari kesimpulan diatas penulis ingin memberikan saran dalam penelitian ini yang bertujuan untuk pengembangan penelitian selanjutnya dalam pengamanan pertukaran data dimana *key* yang telah dibuat masih bisa dipakai oleh banyak *client user* tanpa memberikan autentikasi yang khusus untuk *client user*.

DAFTAR PUSTAKA

- Alanazi, S. T. *et al.* (2019) 'Evaluation Approaches of Service Oriented Architecture (SOA) - A Survey', *2nd International Conference on Computer Applications and Information Security, ICCAIS 2019*, pp. 1–6. doi: 10.1109/CAIS.2019.8769543.
- Claudiu Pirnau, M. A. B. (2016) 'Service-Oriented Architecture (SOA) and Web Services', *21st International Conference on Data Engineering ICDE05, (Icde)*, pp. 1147–1147. Available at: <http://java.sun.com/developer/technicalArticles/WebServices/soa/>.
- Dwi Wijonarko¹, B. W. R. M. (2018) 'Indexing Pada Ebook Perancangan dan Pembuatan Aplikasi Website E-²', *Pengembangan Antarmuka Pemrograman Aplikasi Menggunakan Metode RESTful pada Sistem Informasi Akademik Politeknik Kota Malang*, 63–66.
- Hendini, A. (2016) 'Permodelan Uml Sistem Informasi Moniotring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)', *Crop Science*, 23(2), pp. 201–205. doi: 10.2135/cropsci1983.0011183x002300020002x.
- Jatmika, A. H., Afwani, R. and Agitha, N. (2019) 'Perancangan Software As A Service (SAAS) untuk Sistem Pelayanan Kesehatan Ibu dan Anak

- (PKIA) pada Puskesmas Se-Kota Mataram Berbasis Cloud Computing’,
Jurnal Teknologi Informasi dan Ilmu Komputer, 6(5), p. 485. doi:
10.25126/jtiik.2019651589.
- Kadek, N. *et al.* (2016) ‘Rancang Bangun Sistem Terdistribusi pada Apotek’,
*Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi
Informasi)*, 4(1), pp. 10–21.
- M. Ridwan Effendi (2016) ‘Penerapan Teknologi Cloud Computing Di
Universitas (Studi Kasus: Fakultas Teknologi Informasi Universitas
Bayangkara Jakarta)’, *Jurnal Teknologi Informasi*12, 12(1), pp. 7–14.
- Munawir, E. (2018) ‘Perancangan Portal Informasi Gampong Pada Gampong
Lambeugak Kecamatan Kuta Cot Glie Aceh Besar’, 01, pp. 74–81. doi:
10.1017/CBO9781107415324.004.
- Patni, S. (2017) *Pro RESTful APIs, Pro RESTful APIs*. doi: 10.1007/978-1-4842-
2665-0.
- Putra, B. W. *et al.* (2020) ‘Implementasi Framework CodeIgniter dan Restful API
pada Sistem Informasi Manajemen Tugas Akhir’, *Annual Research
Seminar (ARS)*, 5(1), pp. 307–313.
- Putri, R. A. (2020) ‘Diktat Sistem Basis Data’, p. 65.
- Rahardja, U., Febriyanto2, E. and Aldiya, M. A. (2019) ‘Penerapan Central Event

Information Untuk Mencetak Sertifikat dan Verifikasi Dengan QR Code Menggunakan Global Extreme Programming', *Jurnal Informatika Upgris*, 4(2), pp. 2–7. doi: 10.26877/jiu.v4i2.2843.

Ridlo, I. A. (2017) 'Pedoman Pembuatan Flowchart', *Academia.Edu*, p. 14.

Available at:

https://www.academia.edu/34767055/Pedoman_Pembuatan_Flowchart.

Singh, A. (2017) 'a Simulation Model for Incremental Software Development Life Cycle Model', *International Journal of Advanced Research in Computer Science*, 8(7), pp. 126–132. doi: 10.26483/ijarcs.v8i7.4136.

Sontana, I., Rahmatulloh, A. and Rachman, A. N. (2019) 'Application Programming Interface Google Picker Sebagai Penyimpanan Data Sistem Informasi Arsip Berbasis Cloud', *Jurnal Nasional Teknologi dan Sistem Informasi*, 5(1), pp. 25–32. doi: 10.25077/teknosi.v5i1.2019.25-32.

Surwase, V. (2016) 'REST API Modeling Languages -A Developer's Perspective', *IJSTE -International Journal of Science Technology & Engineering*, 2(10), pp. 634–637.

LAMPIRAN

Lampiran 1

Daftar Pertanyaan dan Hasil Wawancara

Responden : Dadan

Jabatan : *Owner website villatermurahciater.com*

No. Telepon : 085860178701

Tanggal : September 2020

Tempat : Ciater – Kab. Subang

Tujuan : Mengetahui data-data vila yang akan menjadi data API *website*

villatermurahciater.com dan data-data untuk kebutuhan penelitian

Pertanyaan 1

Pertanyaan Apakah ada daftar vila lengkap dengan harga-harganya?

Jawban Ada tetapi saya akan sebutkan satu per satu untuk daftar vila

vilanya.

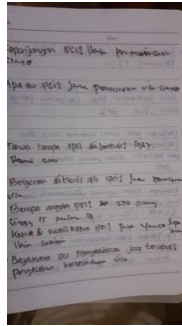
No	Nama Vila	Lokasi	Harga	Jumlah
1	DA
2	DA
3	DA
4	DA
5	DA
6	DA
7	DA

Lampiran Gambar:1 Daftar vila, harga, fasilitas dan *sales quantity*

Pertanyaan 2

Pertanyaan Bisa du jelaskan mengenai tentang apa itu IPCI?

Jawban Jasa pemasaran vila ciater



Lampiran Gambar:2 Wawancara tentang IPCI

Pertanyaan 3

Pertanyaan Berapa penjualan selama tahun 2019 samapai dengan 2020?

Jawaban Penjualan tahun ini sangatlah berat berbeda dengan tahun 2019

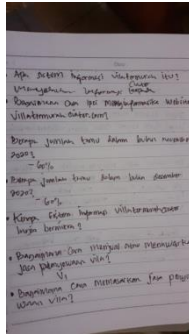


Lampiran Gambar: 3 Wawancara penjualan vila tahun 2019 tahun 2020

Pertanyaan 4

Pertanyaan Bagaimana pola pemesanan vila ciater?

Jawaban Secara langsung kepada calon penyewa vila



Lampiran Gambar: 4 Wawancara pola pemesana vila ciater

Lampiran 2

Kode Program

1. *Controller Home Untuk Menampilkan Halaman Utama Pada Website*

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Home extends CI_Controller {

public function __construct () {

parent::__construct();

//load library untuk pembuatan form login dan sign up

$this->load->library('form_validation');

$this->load->helper('url', 'form');

$this->load->model('modelhome/Home_model');

//session jika user belum login }

//halaman/page home/dashboard halaman utama (controller)

public function index() {
```



```

//menampilkan halaman api-rest

$data['judul'] = 'Home';

$data['vila'] = base_url('img/upload');

$data['datavila'] = $this->Home_model->datavila();

$this->load->view('home/header/header', $data);

$this->load->view('home/index', $data);

$this->load->view('home/footer/footer');

    }

```

2. *Controller Mitra Untuk Menampilkan Halaman Sign up Mitra User, Login Dan Logout*

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Mitra extends CI_Controller {

public function __construct () {

    parent::__construct();

    //load library untuk pembuatan form login dan sign up

    $this->load->library('form_validation');

    $this->load->helper('url', 'form');

    $this->load->model('modelmitra/Mitra_model');

    $this->load->model('modelupload/Upload_model');

    $this->load->model('modeleedit/Edit_model');

```

```

$this->load->model('modelview/View_model');

    //session jika user belum login }

public function signup() {

    //mengecek jika user sudah login ingin ke halaman sign up

    if ($this->session->userdata('email')) {

        redirect('mitra/user');

    }
}

```

3. *Controller Untuk Menampilkan Halaman Sign up Client*

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Client extends CI_Controller {

    public function __construct () {

        parent::__construct();

        //load library untuk pembuatan form login dan sign up

        $this->load->library('form_validation');

        $this->load->helper('url', 'form');

        //link ke database

        $this->load->model('modelclient/Signupclient_model');

        //session jika user belum login

    }

    //halaman/page client (controller)
}

```

```
public function signup()
{
$this->form_validation->set_rules('email', 'Email',
'required|trim|valid_email|is_unique[client.email]', ['valid_email'=> 'Email
harus benar','is_unique'=> 'Email sudah terdaftar']);

$this->form_validation->set_rules('hp', 'Nomor',
'required|trim|numeric|min_length[11]', ['numeric'=> 'Kolom harus di isi
dengan nomor', 'min_length'=>'Nomor minimal 11 digit']);
```