

**RANCANG BANGUN SISTEM INFORMASI SPORT CLUB
HOUSE PERTAMINA - UB MENGGUNAKAN PENDEKATAN
*EVOLUTIONARY PROTOTYPING***

**SKRIPSI
KONSENTRASI REKAYASA PERANGKAT LUNAK**

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh
FAUZIAH PUSPITASARI
NIM. 115090601111005

**PROGRAM STUDI INFORMATIKA/ ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2015

LEMBAR PERSETUJUAN

**RANCANG BANGUN SISTEM INFORMASI SPORT CLUB HOUSE
PERTAMINA - UB MENGGUNAKAN PENDEKATAN *EVOLUTIONARY*
*PROTOTYPING***

SKRIPSI

KEMINATAN REKAYASA PERANGKAT LUNAK

Diajukan untuk memenuhi persyaratan
Memperoleh gelar Sarjana Komputer



Disusun oleh :

FAUZIAH PUSPITASARI

NIM : 115090601111005

Skripsi ini telah disetujui oleh dosen pembimbing pada

Tanggal

Dosen Pembimbing I

Dosen Pembimbing II

Satrio Agung W., S.Kom., M.Kom
NIP. 19860521 201212 1 001

Himawat Aryadita, ST., M.Sc
NIP. 19801018 200801 1 003

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM INFORMASI SPORT CLUB
HOUSE PERTAMINA – UB MENGGUNAKAN PENDEKATAN
EVOLUTIONARY PROTOTYPING**

SKRIPSI

KONSENTRASI REKAYASA PERANGKAT LUNAK

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

FAUZIAH PUSPITASARI

NIM : 115090601111005

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 12 Agustus 2015

Penguji I

Penguji II

Denny Sagita Rusdianto, S.Kom, M.kom

NIK. 851124 06 1 1 0250

Issa Arwani, S.Kom., M.Sc.

NIP. 19830922 201212 1 003

Penguji III

Aditya Rachmadi, S.ST.,MTI

NIK. 860421 16 1 1 0426

Mengetahui

Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

LEMBAR PERNYATAAN ORISINALITAS SKRIPSI

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

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

Malang, 28 Juni 2015

Mahasiswa

FAUZIAH PUSPITASARI

NIM 115090601111005

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, yang atas rahmat-Nya maka penulis dapat menyelesaikan penyusunan tugas akhir yang berjudul “Sistem Informasi Sport Club House Pertamina-UB Menggunakan Pendekatan *Evolutionary Prototyping*”. Shalawat serta salam atas junjungan besar Nabi kita Muhammad S.A.W. beserta keluarga dan para sahabat sekalian.

Tugas akhir ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Komputer di Program Studi Informatika /Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Dalam penulisan skripsi ini penulis merasa masih banyak kekurangan baik pada teknis penulisan maupun materi. Mengingat akan kemampuan yang dimiliki penulis, untuk itu penulis menyampaikan ucapan terimakasih dan penghargaan kepada:

1. Orang tua penulis dan seluruh keluarga besar tercinta yang senantiasa tiada henti hentinya memberikan do'a, dorongan semangat, dan segala pengorbanan baik materil maupun moril demi terselesainya skripsi ini.
2. Bapak Satrio Agung W., S.Kom., M.Kom selaku dosen pembimbing I dan Bapak Himawat Aryadita, ST., M.Sc selaku dosen pembimbing II yang telah sabar memberikan bimbingan, arahan, ilmu dan saran untuk laporan skripsi ini.
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, MT dan Bapak Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika serta segenap Bapak / Ibu Dosen, Staff Administrasi dan Perpustakaan Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Seluruh dosen Program Studi Informatika atas kesediaan membaginya ilmunya kepada penulis.

6. Ibu Siti, Mbak Rinna, Mbak Maria Ulfa, Mas Benny selaku coordinator dan karyawan Sport Club House Pertamina - UB yang memberikan kelancaran dalam skripsi ini.
7. Deny Kurniawan.,S.Kom yang memberikan dukungan dan bantuannya dalam proses pengerjaan skripsi ini.
8. Claudhia, Shofi, Atiqo, Tawang, Laily selaku sahabat yang tidak pernah lelah memberikan motivasi dan saran yang selalu membangun dalam penulisan skripsi ini serta Teman-teman seperjuangan ILKOM (Ilmu Komputer) angkatan 2011 yang tidak bisa disebutkan satu persatu atas dukungannya.
9. Teman-teman Asrama Griya Brawijaya khususnya Lobby A3 16-20 yaitu Mareta, Rofiqoh, Riska, Widdia, Pesona dan Shinta atas semangatnya.
10. Seluruh Keluarga Besar Program Teknologi Informasi dan Ilmu Komputer yang telah memberikan tempat untuk kuliah dan mencari pengalaman di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
11. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dan tidak terlepas dari kekurangan dan kesalahan karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Oleh karena itu, segala kritik dan saran yang bersifat membangun penulis harapkan untuk penyempurnaan skripsi ini. Penulis juga berharap mudah-mudahan skripsi ini dapat bermanfaat bagi semua pihak yang memerlukan

Malang, 28 Juni 2015

Penulis

ABSTRAK

Fauziah Puspitasari. 2015 : Rancang Bangun Sistem Informasi Sport Club House Pertamina-UB Menggunakan Pendekatan *Evolutionary Prototyping*. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing : Satrio Agung W., S.Kom., M.Kom dan Himawat Aryadita, ST., M.Sc

Sport Club House Pertamina - UB merupakan asset Universitas Brawijaya yang memiliki berbagai fasilitas yang dapat digunakan untuk melayani mahasiswa UB atau warga UB atau warga luar sebagai sarana dan prasarana olahraga. Berbagai macam olahraga yang disediakan yaitu tenis, fitness, aerobik, yoga, bulutangkis dan futsal. Sport Club House Pertamina - UB memiliki beberapa masalah, yaitu kurangnya informasi seputar Sport Club House Pertamina-UB sehingga menyebabkan pemesan tidak mengetahui informasi seperti cabang olahraga, harga olahraga, jadwal olahraga, dan berita terbaru serta tidak adanya penyalur kritik saran pengguna kepada pengelola Sport Club House Pertamina – UB, selain itu dalam melakukan pendaftaran member dan pemesanan lapangan olahraga dilakukan dengan datang langsung ketempat untuk melihat informasi jadwal yang telah terpesan, sehingga proses pemesanan tidak efisien. Berdasarkan masalah tersebut, solusinya adalah dengan membangun sistem informasi Sport Club House Pertamina - UB. Dalam pembuatan sistem ini menggunakan metode *evolutionary prototyping* karena daftar kebutuhan pada Sport Club House Pertamina - UB tidak spesifik. Dalam pembuatan *evolutionary prototyping* dilakukan iterasi sebanyak dua kali dengan menghasilkan sistem versi 2.0 atau versi terakhir, pada evaluasi oleh pengguna terdapat perubahan pada fitur sistem yang sesuai dengan kebutuhan pengguna. Berdasarkan hasil pengujian pada sistem informasi yang telah dilakukan diperoleh kesimpulan bahwa kompleksitas dari sistem ini adalah rendah sehingga kerumitan perbaikan coding juga rendah, kemudian secara fungsionalitas dari sistem ini berfungsi 100% valid dan secara keseluruhan berjalan dengan baik.

Kata Kunci: Sistem Informasi, Pendaftaran member, Pemesanan olahraga, *Evolutionary Prototyping*

ABSTACT

Fauziah Puspitasari. 2015 : Design of Information Systems Sport Club House Pertamina-UB Using Evolutionary Prototyping Approach. Program Information Technology and Computer Science, University of Brawijaya, Malang.

Advisory Lecturer : Satrio Agung W., S.Kom., M.Kom and Himawat Aryadita, ST., M.Sc

Sport club house pertamina - ub is asset Brawijaya University who has various facilities that can be used to serve students UB or residents of UB or residents of outer as sports facilities and infrastructure .Various kind of a sports that provided namely tennis, fitnes, aerobic, yoga, badminton and futsal . Sport Club House Pertamina - UB has some problems, such as lack of information about Sport Club House Pertamina-UB causing the buyer does not know the information such as sports, exercise price, exercise schedule, and the latest news as well as the absence of criticism advice to the management users Sport Club House Pertamina - UB., in addition in doing registration member and reserving the sports field done with come directly into place to see the information the booking schedule, so that the process of reserving not efficient .Based on the issue, the solution is to build system of information sport club house -- ub pertamina .In making this system uses a method of evolutionary prototyping because lists on sport needs club house pertamina - ub not specific. In the manufacture of evolutionary prototyping done iteration twice by producing version 2.0 or the latest version, on evaluation by user on features system there was a change to suit the needs users . Based on the results of testing on information systems that have been obtained the conclusion that the complexity of this system is low that the complexity of the coding improvement also low , then in functionality of this system serves 100 % valid and overall runs with good.

Keywords: *Information Systems, member registration, booking sports, Evolutionary Prototyping*

DAFTAR ISI

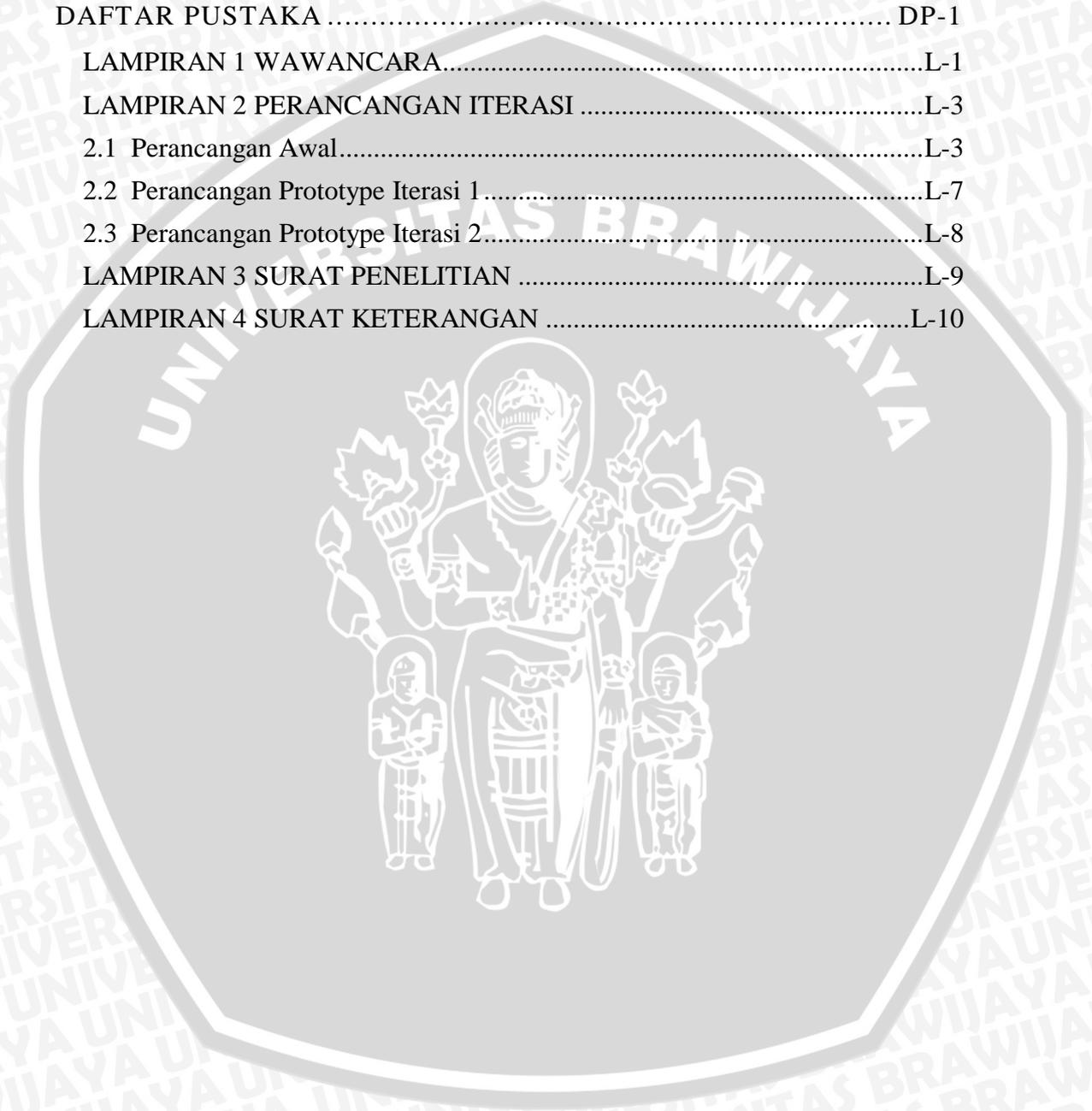
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN	iv
KATA PENGANTAR	v
ABSTRAK.....	vii
ABSTACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan.....	4
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI	6
2.1 Kajian Pustaka.....	6
2.2 Dasar Teori	7
2.2.1 Sport Club House Pertamina - UB.....	7
2.2.2 Konsep Sistem Informasi	8
2.2.3 Konsep Rekayasa Perangkat Lunak	9
2.2.3.1 Pengertian Rekayasa Perangkat Lunak.....	9
2.2.4 Konsep <i>Prototyping</i>	10
2.2.4.1 Pendekatan <i>Evolutionary prototyping</i>	11
2.2.4.2 Fase <i>Evolutionary prototyping</i>	12
2.2.4.3 Keuntungan dan kelebihan <i>Evolutionary prototyping</i>	13
2.2.4.4 Perbedaan <i>Evolutionary prototyping</i> dengan <i>prototyping</i>	13
2.2.5 Konsep <i>Unified Modelling Language (UML)</i>	14
2.2.5.1 Pengertian <i>Unified Modelling Language (UML)</i>	14



2.2.5.2	<i>Use Case Diagram</i>	15
2.2.5.3	<i>Class Diagram</i>	16
2.2.5.4	<i>Sequence Diagram</i>	18
2.2.5.5	<i>ActivityDiagram</i>	19
2.2.5.6	<i>Diagram hubungan entitas (ERD)</i>	20
2.2.6	Konsep Basis Data	21
2.2.6.1	<i>Relational Database Management System (RDBMS)</i>	22
2.2.7	Konsep Pengujian Perangkat Lunak	22
2.2.7.1	Pengujian Validasi dengan Metode Blackbox	23
2.2.7.2	Pengujian White Box	23
2.2.7.3	Pengujian Compatibility	24
BAB 3 METODOLOGI PENELITIAN		25
3.1	Tahap Studi Literatur.....	26
3.2	Fase Analisa Kebutuhan.....	26
3.3	Fase Perancangan	27
3.4	Fase Implementasi.....	27
3.4.1	Evaluasi Pengguna	28
3.5	Pengujian dan Analisis	28
3.6	Pengambilan Kesimpulan dan Saran	29
BAB 4 ANALISIS DAN PERANCANGAN		30
4.1	Fase Analisa Kebutuhan.....	31
4.1.1	Deskripsi Umum Sistem Informasi.....	31
4.1.2	Analisis Data	32
4.1.3	Kebutuhan Fungsional	33
4.1.4	Kebutuhan Non Fungsional.....	36
4.1.5	Karakteristik Pengguna	36
4.1.6	Model Diagram <i>Use Case</i> Iterasi -2.....	38
4.1.7	<i>Use Case Skenario</i>	39
4.2	Fase Perancangan	68
4.2.1	Perancangan Sequence Diagram	68
4.2.2	Perancangan Basis Data	76
4.2.2.1	Spesifikasi ERD.....	76
4.2.2.2	DIAGRAM ERD	78
4.2.2.3	ER Mapping.....	79
4.2.2.4	Physical Data Models	81

4.2.3	Perancangan Activity Diagram	82
4.2.4	Perancangan Class Diagram.....	89
4.2.5	Perancangan Prototype.....	93
BAB 5 IMPLEMENTASI		97
5.1	Fase Implementasi.....	97
5.1.1	Deskripsi Lingkungan Perangkat Keras.....	97
5.1.1.1	Deskripsi Lingkungan Perangkat Keras	98
5.1.1.2	Deskripsi Lingkungan Perangkat Lunak.....	98
5.1.2	Batasan Implementasi	98
5.1.3	Implementasi <i>Class</i> Pada File Program	99
5.1.4	Implementasi Basis Data.....	99
5.1.5	Implementasi Prototype Rilis.....	105
5.1.5.1	Implementasi Prototype Versi 0.0	105
5.1.5.2	Implementasi Prototype Versi 1.0	108
5.1.5.3	Implementasi Prototype Versi 2.0	109
5.2	Fase Evaluasi – Rilis Prototype.....	113
5.2.1	Evaluasi Prototype Versi 0.0.....	113
5.2.2	Evaluasi Prototype Versi 1.0.....	113
BAB 6 PENGUJIAN DAN ANALISIS		114
6.1	Pengujian Sistem	114
6.1.1	Pengujian White Box	115
6.1.1.1	Tujuan.....	115
6.1.1.2	Mekanisme.....	115
6.1.1.3	Hasil Pengujian.....	115
6.1.2	Pengujian Validasi dengan Metode <i>Black Box</i>	125
6.1.2.1	Tujuan.....	125
6.1.2.2	Mekanisme.....	125
6.1.2.3	Hasil Pengujian.....	125
6.1.3	Pengujian <i>Compatibility</i>	138
6.1.3.1	Tujuan.....	138
6.1.3.2	Mekanisme.....	138
6.1.3.3	Hasil Pengujian.....	138
6.2	Analisis Sistem	143
6.2.1	Analisis Pengujian Black Box.....	143
6.2.2	Analisis Pengujian White Box	143

6.2.3 Analisis Pengujian <i>Compatibility</i>	143
BAB 7 PENUTUP	144
6.1 Kesimpulan.....	144
6.2 Saran.....	144
DAFTAR PUSTAKA	DP-1
LAMPIRAN 1 WAWANCARA	L-1
LAMPIRAN 2 PERANCANGAN ITERASI	L-3
2.1 Perancangan Awal.....	L-3
2.2 Perancangan Prototype Iterasi 1.....	L-7
2.3 Perancangan Prototype Iterasi 2.....	L-8
LAMPIRAN 3 SURAT PENELITIAN	L-9
LAMPIRAN 4 SURAT KETERANGAN	L-10



DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Sport Club House Pertamina - UB 8

Gambar 2.3 Fase pada metode *Evolutionary prototyping* 12

Gambar 3.1 Diagram Alir Bab Metodologi Penelitian 25

Gambar 4.1 Diagram Blok Bab Analisis dan Perancangan 30

Gambar 4.2 Model Hirarki Aktor..... 38

Gambar 4.3 Model Use Case User iterasi ke 2 38

Gambar 4.4 Model Use Case Diagram iterasi 2..... 39

Gambar 4.5 Model Use Case Actor *User*..... 40

Gambar 4.6 Model Use Case Actor Member Account 42

Gambar 4.7 Model Use Case Actor Front Office..... 48

Gambar 4.8 Model Use Case Actor Kasir..... 50

Gambar 4.9 Model Use Case Actor Bagian Gedung 53

Gambar 4.10 Model Use Case Actor Keuangan..... 57

Gambar 4.11 Model Use Case Actor Admin 59

Gambar 4.12 *sequence diagram* membuat komplain 69

Gambar 4.13 *sequence diagram* validasi membership 69

Gambar 4.14 *sequence diagram* melihat tips..... 70

Gambar 4.15 *sequence diagram* mendaftar membership 71

Gambar 4.16 *sequence diagram* pemesanan tempat 71

Gambar 4.17 *sequence diagram* menambah pembayaran..... 72

Gambar 4.18 *sequence diagram* validasi pemesanan 73

Gambar 4.19 *sequence diagram* menambah membership 73

Gambar 4.20 *sequence diagram* menambah berita 74

Gambar 4.21 *sequence diagram* mengubah berita..... 75

Gambar 4.22 *sequence diagram* menghapus berita 75

Gambar 4.23 *sequence diagram* melihat laporan keuangan 76

Gambar 4.24 Spesifikasi ERD entitas pegawai dan entitas member .. 76

Gambar 4.25 Spesifikasi ERD entitas pegawai dan member..... 77

Gambar 4.26 Spesifikasi ERD entitas pegawai dan entitas berita 77

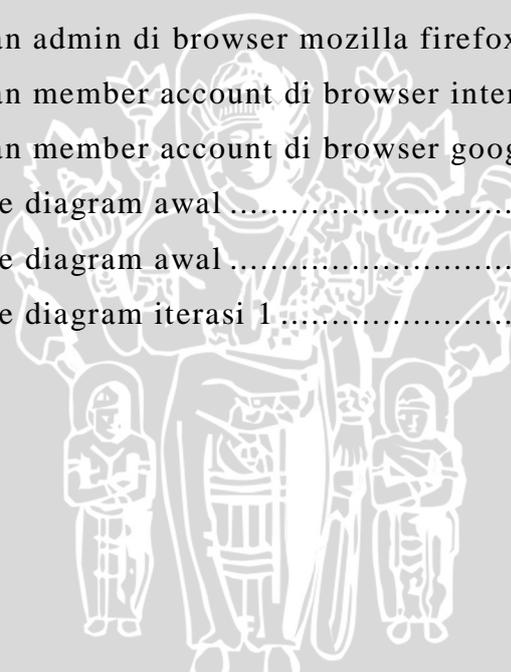
Gambar 4.27 Spesifikasi ERD entitas pegawai dan entitas tips 77

Gambar 4.28 Spesifikasi ERD entitas pegawai dan entitas fasilitas .. 78



Gambar 4.29 Spesifikasi ERD entitas member dan komplain	78
Gambar 4.30 Entity Relasional Diagram(ERD)	79
Gambar 4.32 activity diagram Login.....	82
Gambar 4.33 activity <i>diagram register account</i>	83
Gambar 4.34 activity <i>diagram</i> pemesanan tempat.....	84
Gambar 4.35 activity <i>diagram</i> validasi membership	85
Gambar 4.36 activity <i>diagram</i> membuat komplain.....	85
Gambar 4.37 activity <i>diagram</i> menambah berita.....	86
Gambar 4.38 activity <i>diagram</i> menghapus berita	87
Gambar 4.39 activity <i>diagram</i> mengubah berita	87
Gambar 4.40 activity <i>diagram</i> melihat laporan keuangan.....	88
Gambar 4.41 activity <i>diagram</i> menambah pembayaran	89
Gambar 4.42 Class Diagram pada model	90
Gambar 4.43 Class Diagram pada controller.....	91
Gambar 4.45 Rancangan prototype halaman berita.....	93
Gambar 4.46 Rancangan prototype halaman membuat complain.....	94
Gambar 4.47 Rancangan prototype halaman register account	94
Gambar 4.48 Rancangan prototype halaman mengelola berita	95
Gambar 4.49 Rancangan prototype halaman menambah tips olahraga.....	95
Gambar 4.50 Rancangan prototype halaman pemesanan tempat.....	96
Gambar 5.1 Diagram Blok Bab Implementasi	97
Gambar 5.2 Implementasi Beranda	105
Gambar 5.3 Implementasi Register Account	106
Gambar 5.5 Implementasi Login.....	107
Gambar 5.6 Implementasi Pemesanan Tempat	107
Gambar 5.7 Implementasi Menghapus Berita.....	108
Gambar 5.8 Implementasi Membuat Komplain	108
Gambar 5.9 Implementasi Validasi Membership	108
Gambar 5.10 Implementasi Mendaftar Membership Aerobik.....	109
Gambar 5.11 Implementasi Mencetak Struk Membership	109
Gambar 5.12 Implementasi Menambah Tips	109
Gambar 5.13 Implementasi Menghapus Tips	110

Gambar 5.14	Implementasi halaman home versi 2.0.....	110
Gambar 5.16	Implementasi halaman member versi 2.0.....	111
Gambar 5.17	Implementasi halaman front office versi 2.0.....	111
Gambar 5.18	Implementasi halaman kasir versi 2.0.....	112
Gambar 5.20	Implementasi halaman keuangan versi 2.0.....	112
Gambar 6.1	Diagram Blok Bab Pengujian dan Analisis.....	114
Gambar 6.2	Flowgraph Login.....	116
Gambar 6.3	Flowgraph daftar membership.....	120
Gambar 6.4	Flowgraph register account.....	123
Gambar 6.5	Halaman utama di linux Ubuntu.....	139
Gambar 6.6	Halaman utama di linux centOS.....	140
Gambar 6.7	Halaman admin di browser mozilla firefox.....	141
Gambar 6.8	Halaman member account di browser internet explorer.....	141
Gambar 6.9	Halaman member account di browser google chrome ..	142
Gambar 8.1	use case diagram awal.....	L-6
Gambar 8.2	use case diagram awal.....	L-7
Gambar 8.3	use case diagram iterasi 1.....	L-8



DAFTAR TABEL

Table 2.1 Perbedaan metode 14

Table 2.2 Keterangan Umum Pada Use Case Diagram..... 15

Table 2.3 Keterangan Hubungan Pada Use Case Diagram 16

Table 2.4 Keterangan Simbol Pada Class Diagram 17

Table 2.5 Keterangan Notasi Pada Sequence Diagram..... 19

Table 2.6 Keterangan Notasi Pada Activity Diagram 20

Tabel 4.1 Kebutuhan Fungsional iterasi ke 2..... 33

Tabel 4.2 Kebutuhan Non Fungsional..... 36

Tabel 4.3 Karakteristik Aktor 36

Tabel 4.4 Use Case Skenario login..... 39

Tabel 4.5 Use Case Skenario Melihat Berita 40

Tabel 4.6 Use Case Skenario Melihat Tips Olahraga..... 41

Tabel 4.7 Use Case Skenario Melihat Fasilitas..... 41

Tabel 4.8 Use Case Skenario *Register Account* 41

Tabel 4.9 Use Case Skenario Pemesanan Tempat 42

Tabel 4.10 Use Case Skenario Melihat FAQ 43

Tabel 4.11 Use Case Skenario cetak struk pemesanan..... 44

Tabel 4.12 Use Case Skenario Melihat Riwayat Pemesanan 44

Tabel 4.13 Use Case Skenario Cetak Struk Membership 45

Tabel 4.14 Use Case Skenario Mendaftar Membership..... 45

Tabel 4.15 Use Case Skenario Melihat Riwayat Membership 46

Tabel 4.16 Use Case Skenario Membuat Komplain..... 46

Tabel 4.17 Use Case Skenario Melihat Tagihan..... 47

Tabel 4.18 Use Case Skenario Menambah Membership..... 48

Tabel 4.19 Use Case Skenario Melihat data Membership 48

Tabel 4.20 Use Case Skenario Menghapus Membership 49

Tabel 4.21 Use Case Skenario Validasi Data Membership..... 50

Tabel 4.22 Use Case Skenario Menambah Pembayaran 51

Tabel 4.23 Use Case Skenario Menghapus Membership 51

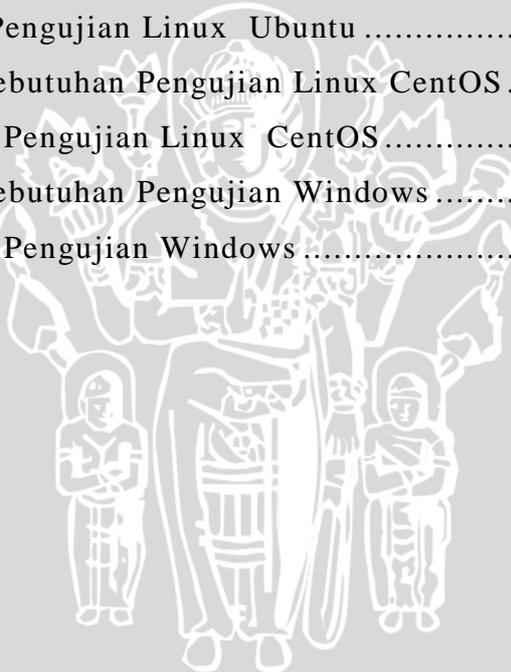
Tabel 4.24 Use Case Skenario Melihat Pemasukan Membership 52

Tabel 4.25 Use Case Skenario Validasi Pemesanan 53



Tabel 4.26	Use Case Skenario Menambah Pemesanan	54
Tabel 4.27	Use Case Skenario Menghapus Pemesanan	54
Tabel 4.28	Use Case Skenario Melihat Data Pemesan	55
Tabel 4.29	Use Case Skenario Menambah Pemasukan.....	55
Tabel 4.30	Use Case Skenario Menghapus Pemasukan.....	56
Tabel 4.31	Use Case Skenario Melihat Pemasukan Pemesanan	57
Tabel 4.32	Use Case Skenario Melihat laporan Keuangan	57
Tabel 4.33	Use Case Skenario Membalas Komplain.....	59
Tabel 4.34	Use Case Skenario Menambah Berita.....	60
Tabel 4.35	Use Case Skenario Menghapus Berita	60
Tabel 4.36	Use Case Skenario Mengubah Berita.....	61
Tabel 4.37	Use Case Skenario Menambah Pegawai.....	62
Tabel 4.38	Use Case Skenario Menghapus Pegawai	62
Tabel 4.39	Use Case Skenario Mengubah Pegawai	63
Tabel 4.40	Use Case Skenario Menambah Fasilitas	64
Tabel 4.41	Use Case Skenario Menghapus Fasilitas.....	64
Tabel 4.42	Use Case Skenario Mengubah Fasilitas	65
Tabel 4.43	Use Case Skenario Menambah Tips	66
Tabel 4.44	Use Case Skenario Menghapus Tips.....	66
Tabel 4.45	Use Case Skenario Mengubah Tips	67
Tabel 5.1	Spesifikasi perangkat keras komputer	98
Tabel 5.2	Spesifikasi perangkat keras komputer	98
Tabel 5.3	Struktur <i>Class</i> dan lokasi file.....	99
Tabel 5.4	Struktur DDL Tabel fakultas.....	100
Tabel 5.5	Struktur DDL Tabel unitkerja	100
Tabel 5.6	Struktur DDL Tabel tipeidentitas	100
Tabel 5.7	Struktur DDL Tabel member	100
Tabel 5.8	Struktur DDL Tabel pegawai	101
Tabel 5.9	Struktur DDL Tabel berita	102
Tabel 5.10	Struktur DDL Tabel fasilitas	102
Tabel 5.11	Struktur DDL Tabel tips	102
Tabel 5.12	Struktur DDL Tabel membership	103

Tabel 5.13 Struktur DDL Tabel olahraga	103
Tabel 5.14 Struktur DDL Tabel pemesanan.....	104
Tabel 5.15 Struktur DDL Tabel komplain.....	104
Tabel 6.1 Pseudo code Login	115
Tabel 6.2 Kasus Uji <i>Login</i>	117
Tabel 6.3 Pseudo code Mendaftar Membership.....	119
Tabel 6.4 Kasus Uji mendaftar membership.....	121
Tabel 6.5 Pseudo code register account	122
Tabel 6.6 Kasus Uji register account	124
Tabel 6.7 Pengujian Black Box Sistem Informasi UB Sport	125
Tabel 6.8 Daftar kebutuhan pengujian linux ubuntu.....	138
Tabel 6.9 Skenario Pengujian Linux Ubuntu	138
Tabel 6.10 Daftar Kebutuhan Pengujian Linux CentOS.....	139
Tabel 6.11 Skenario Pengujian Linux CentOS.....	139
Tabel 6.12 Daftar Kebutuhan Pengujian Windows	140
Tabel 6.13 Skenario Pengujian Windows	140



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Universitas Brawijaya merupakan Universitas ternama di Indonesia yang memiliki fasilitas yang sangat memadai, mulai bidang teknologi, kesehatan bahkan olahraga sekalipun. Dalam bidang olahraga, Universitas Brawijaya memiliki Sport Club House Pertamina UB atau yang biasa dikenal UB Sport Center merupakan aset Universitas Brawijaya yang memiliki berbagai fasilitas yang dapat digunakan untuk melayani mahasiswa atau warga Universitas Brawijaya atau warga di luar Universitas Brawijaya (umum) sebagai sarana dan prasarana olahraga. Berbagai macam olahraga yang disediakan oleh Sport Club House Pertamina - UB mulai dari cabang tenis, fitnes, aerobik, yoga, bulutangkis dan futsal. Semua kegiatan olahraga tersebut dikelola oleh pihak Sport Club House Pertamina - UB.

Pada Sport Club House Pertamina - UB memiliki beberapa masalah, yaitu kurangnya informasi seputar Sport Club House Pertamina-UB sehingga menyebabkan pemesan tidak mengetahui informasi seputar cabang olahraga, harga olahraga, jadwal olahraga, dan berita terbaru serta tidak adanya penyalur kritik saran pengguna kepada pengelola Sport Club House Pertamina - UB, kemudian untuk cabang olahraga yang menyewa tempat atau lapangan tidak ada sistem yang mengatur pemesanannya sehingga ketika calon pemesan yang ingin memesan tempat harus datang langsung untuk melihat informasi jadwal yang telah terpesan. Kemudian untuk cabang olahraga yang mendaftar member, pada pengelolaan data member kurang efisien akibat pengisian biodata yang berulang-ulang untuk pendaftaran olahraga yang berbeda atau pendaftaran ulang. Sehingga berdasarkan masalah tersebut perlu adanya sistem informasi yang dapat mengelola Sport Club House Pertamina - UB.

Oleh karena itu, untuk mengatasi permasalahan tersebut dibutuhkan sistem informasi yang mampu memberikan informasi yang *update* tentang Sport Club House Pertamina - UB seperti biaya pendaftaran member dan pemesanan lapangan, jadwal olahraga, fasilitas yang ada serta berita kegiatan terbaru pada

Sport Club House Pertamina - UB. Kemudian dalam proses pemesanan lapangan dan pendaftaran member dibutuhkan sistem yang dapat dipesan secara *online* melalui media internet, sehingga dapat mengurangi keterbatasan waktu dan meminimalisir biaya, karena calon pemesan akan dapat memesan lapangan atau ruangan tanpa harus datang ke Sport Club House Pertamina – UB.

Dalam pembuatan sistem ini, menggunakan pendekatan *evolutionary prototyping*, karena daftar kebutuhan akan sistem tidak spesifik secara detail dari segi masukan (*input*), proses, maupun keluaran (*output*) dan belum adanya standar operasional prosedur serta pemahaman tentang teknologi informasi yang berbeda-beda, maka pembuatan sistem informasi harus memenuhi kebutuhan pengguna dengan keterlibatan pengguna yang lebih awal untuk melihat fitur product dan dapat memberikan feedback sampai pengguna puas dengan sistem. Oleh karena itu menggunakan pendekatan *evolutionary prototyping* dikarenakan cara kerjanya yang memungkinkan interaksi antara pengguna dengan pengembang selama proses pembuatan sistem dengan melihat fitur produk, sehingga nantinya diharapkan dapat tercipta suatu sistem yang sesuai dengan harapan dan kebutuhan dari pengguna [PRE-12].

Berdasarkan permasalahan tersebut maka penggunaan metode tersebut dipilih sebagai metode dalam penelitian penulis. Sistem diharapkan dapat membantu pengguna dalam kebutuhan informasi di Sport Club House Pertamina - UB sehingga sesuai dengan kebutuhan dan keinginan pengguna. Berdasarkan latar belakang tersebut maka judul penelitian ini adalah **“Rancang Bangun Sistem Informasi Sport Club House Pertamina - UB Menggunakan Pendekatan *Evolutionary Prototyping*”**.

1.2 Rumusan Masalah

Berdasarkan paparan latar belakang tersebut, maka rumusan masalah adalah sebagai berikut:

1. Bagaimana menganalisis dan merancang serta mengimplementasi sistem informasi Sport Club House Pertamina – UB menggunakan pendekatan *evolutionary prototyping* ?

2. Bagaimana menguji sistem informasi Sport Club House Pertamina - UB?

1.3 Batasan Masalah

Batasan penelitian yang dilakukan pada penulisan skripsi ini mempunyai beberapa batasan masalah, yaitu:

1. Sistem di Sport Club House Pertamina - UB difokuskan pada sistem informasi olahraga, pendaftaran olahraga membership, pemesanan tempat lapangan dan laporan keuangan serta komplain.
2. Mekanisme pembayaran pada pendaftaran olahraga membership dilakukan di kasir Sport Club House Pertamina - UB dan pembayaran pada pemesanan tempat olahraga dilakukan di bagian gedung Sport Club House Pertamina - UB atau melalui rekening rektorat UB.
3. Mekanisme pemasukan keuangan hanya dari pendaftaran olahraga dan pemesanan tempat olahraga tanpa mekanisme pengeluaran keuangan.
4. Dalam fase menggunakan sistem pada tahapan *evolutionary prototyping* melakukan release sebanyak 2 iterasi dan tidak melakukan fase project release sistem yaitu instalasi aplikasi.

1.4 Tujuan

Tujuan yang ingin dicapai dari pengimplementasian sistem informasi ini adalah untuk merancang dan mengimplementasi sistem informasi Sport Club House Pertamina - UB dengan menggunakan pendekatan *evolutionary prototyping*.

1.5 Manfaat

Diharapkan dengan adanya perancangan sistem informasi ini dapat bermanfaat bagi :

a. Bagi Penulis

1. Dapat memahami dan bisa menerapkan pengembangan terhadap pembelajaran yang didapat selama masa perkuliahan.

2. Dapat menjadi pembelajaran dan dapat menambah pengalaman di bidang studi keilmuan yang terkait.

b. Bagi Pengguna

1. Sebagai media informasi bagi pengguna.
2. Dapat memberikan kemudahan dalam proses pendaftaran dan pemesanan olahraga.

c. Bagi Sport Club House Pertamina - UB

1. Dapat meningkatkan pelayanan terhadap *customer*.
2. Dapat membantu dalam mengelola proses bisnis di Sport Club House Pertamina - UB.
3. Dapat memberikan kemudahan dalam menyediakan sistem informasi.

1.6 Sistematika Penulisan

Penyusunan penelitian ini menggunakan kerangka pembahasan yang tersusun sebagai berikut :

BAB 1 Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB 2 Landasan Kepustakaan

Membahas kajian pustaka dan dasar teori penunjang yang berhubungan dengan sistem informasi, Sport Club House Pertamina - UB, rekayasa perangkat lunak, metode *Evolutionary prototyping* , *Unified Modelling Language*, basis data dan pengujian yg menjadi acuan dasar dalam pembuatan sistem ini.

BAB 3 Metodologi Penelitian

Membahas tentang metode yang digunakan dalam penulisan yang terdiri dari studi literatur, analisa kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, serta pengambilan kesimpulan dan saran.

BAB 4 Analisis dan Perancangan Sistem

Membahas tentang analisis dan perancangan dari sistem informasi sesuai dengan teori yang ada yang kemudian menjadi landasan implementasi.

BAB 5 Implementasi Sistem

Membahas implementasi perangkat lunak sesuai dengan perancangan sistem yang telah dibuat.

BAB 6 Pengujian dan Analisis

Membahas proses dan hasil pengujian terhadap sistem yang telah diimplementasi dan melakukan analisis terhadap hasil pengujian.

Bab 7 Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak berbasis website yang dikembangkan dalam penelitian ini serta saran-saran untuk pengembangan lebih lanjut.

BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab dua ini berisi kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka membahas penelitian yang dilakukan sebelumnya. Sedangkan dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini adalah membandingkan penelitian ini dengan penelitian sebelumnya. Penelitian ini membahas tentang “Rancang Bangun Sistem Informasi Sport Club House Pertamina - UB Menggunakan Pendekatan *Evolutionary prototyping*”. Penelitian serupa telah dilakukan oleh Xiaoshuan Zhang yang berjudul “*Applying evolutionary prototyping model for eliciting system requirement of meat traceability at agribusiness level*”. Pada penelitian ini membahas tentang sistem traceability berbasis IT pada tingkat agribisnis. Sistem ini menggunakan model *evolusioner prototyping* untuk memperoleh kebutuhan sistem karena persyaratan sistem dapat dikategorikan sebagai tidak spesifik. Tujuan dari penelitian ini adalah untuk membangun sistem traceability yang dapat memvalidasi persyaratan sistem pada tingkat agribisnis. Penelitian ini menghasilkan proses evolusi dari version 0.0, version 1.0 ke version 2.0 pada fase evolusioner [ZHA -10].

Penelitian selanjutnya dilakukan oleh Nian-Shing Chen dengan judul *Applying Evolutionary prototyping Model in Developing Stream-based Lecturing Systems*. Penelitian tersebut membahas tentang sistem pengembangan pembelajaran yang menggunakan model pengembangan prototipe evolusioner. Tujuan penelitian adalah untuk membangun sistem yang dapat digunakan pembelajaran siswa. Penelitian ini menghasilkan tiga version sistem pembelajaran dengan drawback dari pengguna untuk menyempurnakan sistem dengan solusi pengembangan full screen capture. [NIA-02].

Berdasarkan kedua Penelitian tersebut, penulis menjadikan kedua Penelitian tersebut sebagai salah satu acuan referensi. Pada penelitian pertama

yang menjadikan *evolutionary prototyping* model sebagai acuan development proses. penulis merferensi cara penamaan version pada implementasi sistem dan *feedback* pengguna, dimana prototype evolusioner terus menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem baru. Jadi satu prototype evolusioner akan menjadi sistem actual. Pada Penelitian yang kedua penulis mengadopsi kegiatan yang dilakukan pada setiap fase yang ada pada penelitian tersebut, yakni focus pada aspek kebutuhan pengguna kemudian menganalisis dan merancang versi pertama, setelah mengembangkan versi pertama dan mengimplementasikannya, pengguna memberikan *feedback* kelemahan sistem sehingga merevisi versi pertama dan terus berevolusi hingga tidak ada kelemahan pada sistem.

2.2 Dasar Teori

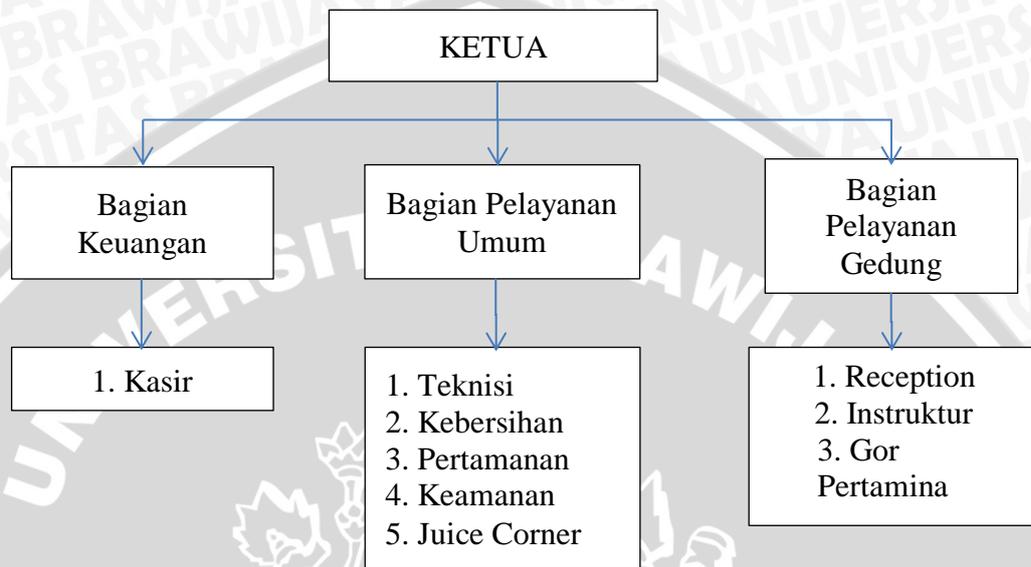
2.2.1 Sport Club House Pertamina - UB

Sport Club House Pertamina –UB atau yang biasa dikenal Sport Club House Pertamina - UB adalah gedung pusat olahraga Universitas Brawijaya bagi civitas akademika dan masyarakat umum yang bertujuan memberikan fasilitas olahraga secara lengkap dan nyaman khususnya bagi prestasi non-akademik mahasiswa/i. Gedung yang bertempat di Jl.Terusan Cobogo I, Malang ini didirikan pada tahun 2011 dan diresmikan pada tanggal 18 September 2013. Secara keseluruhan gedung Sport Club House Pertamina - UB yang tersusun tiga lantai ini memiliki fasilitas keolahragaan yang cukup bervariasi, diantaranya lapangan tenis indoor, lapangan bulu tangkis, lapangan futsal, fitness center, ruang yoga dan aerobic serta beberapa ruangan penunjang lainnya.

Prinsip kerja Sport Club House Pertamina - UB adalah memberikan layanan terbaik bagi seluruh civitas akademika UB maupun pihak lain dalam hal sarana dan prasarana keolahragaan.

Berdasarkan stuktur organisasi yang terdapat pada gambar 2.1, Sport Club House Pertamina - UB atau Sport Club House Pertamina –UB dipimpin oleh ketua yang membawahi 3 bagian, tiap bagian tersebut yaitu, bagian keuangan, bagian pelayanan umum, bagian pelayanan gedung. Masing-masing bagian memiliki sub bagian. Bagian keuangan memiliki sub bagian kasir. Bagian

pelayanan umum memiliki sub bagian teknisi, kebersihan, pertamanan, keamanan, juice corner. Bagian pelayanan gedung memiliki sub bagian reception, instruktur dan gor pertamina. Berikut ini merupakan gambar struktur organisasi pada gambar 2.1 berikut:



Gambar 2.1 Struktur Organisasi Sport Club House Pertamina - UB
Sumber : Sport Club House Pertamina – UB

2.2.2 Konsep Sistem Informasi

Salah satu unsur penting dalam pengambilan keputusan dalam suatu organisasi adalah informasi. Sebelum mengambil keputusan perlu diketahui dari mana dan bagaimana informasi ini sendiri diperoleh. Pada dasarnya informasi tersebut dapat diperoleh dari sistem informasi dan adapun pengertian sistem informasi menurut Jogiyanto, H.M (1999:697) adalah sebagai berikut:

“Sistem informasi adalah suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, proses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdas”

Menurut Abdul Kadir(2003:10) mendefinisikan sistem informasi sebagai berikut:

“Sejumlah komponen (manusia, komputer, teknologi informasi dan procedure kerja) yang diproses (data menjadi informasi) dengan maksud untuk mendapat suatu sasaran atau tujuan”.

Berdasarkan teori-teori di atas, maka dapat disimpulkan bahwa sistem informasi adalah unsur-unsur yang terkait dalam sistem dan saling berhubungan untuk mengubah data(masukan) menjadi informasi (keluaran), guna mencapai sasaran-sasaran organisasi atau perusahaan.

2.2.3 Konsep Rekayasa Perangkat Lunak

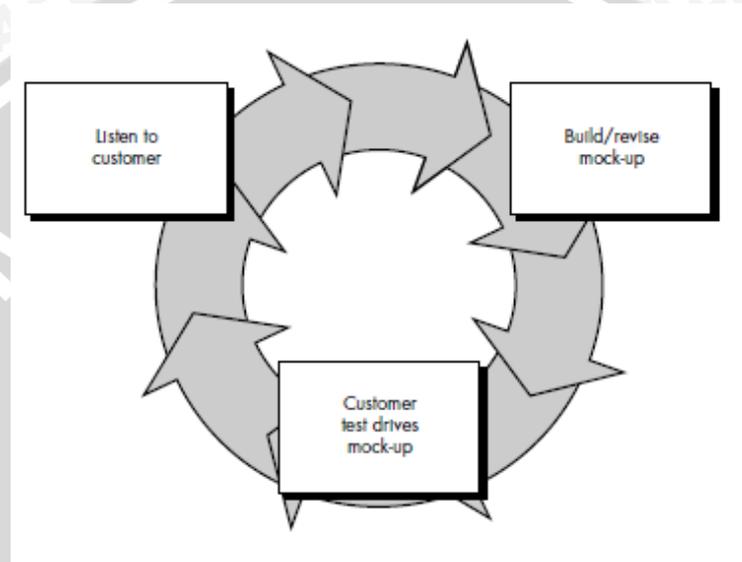
2.2.3.1 Pengertian Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan [SOM-03]. Perekayasa perangkat lunak memakai pendekatan yang sistematis dan terorganisir terhadap pekerjaan mereka karena cara ini seringkali paling efektif masalah pemilihan metode yang paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Rekayasa ini mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informasi terhadap pengembangan yang mungkin efektif pada beberapa keadaan [SOM-03].

Seorang software engineer harus menggabungkan strategi pengembangan perangkat lunak yang meliputi proses, metode, dan alat bantu yang digunakan dalam proses pengembangan. Strategi ini disebut sebagai model proses (process model) atau paradigma rekayasa perangkat lunak dipilih sesuai dengan sifat dari proyek dan aplikasi yang akan dibuat. Terdapat beberapa model proses untuk rekayasa perangkat lunak antara lain linear sequential model (waterfall), *Prototyping*, Rapid Application Development (RAD), incremental model dan spiral model [PRE-10]. Model proses yang digunakan dalam penelitian ini adalah *Prototyping* Model.

2.2.4 Konsep Prototyping

Prototyping merupakan salah satu metode dalam proses pengembangan perangkat lunak. Metode ini bekerja dengan cara interaksi dari pihak pengguna dengan pengembang/pembuat perangkat lunak. Hal ini bertujuan untuk membangun suatu aplikasi atau perangkat lunak yang sesuai dengan kebutuhan pengguna [PRE-12]. Proses pada prototyping dijelaskan pada gambar 2.2.



Gambar 2.2 Pendekatan Prototyping

Sumber : [PRE-12]

Tahap-tahap pengembangan Prototype model menurut Roger S. Pressman, Ph.D. adalah :

1. Mendengarkan pelanggan

Pada tahap ini dilakukan pengumpulan kebutuhan dari system dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu system yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana system yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

2. Membangun dan Memperbaiki Prototype

Pada tahap ini, dilakukan perancangan dan pembuatan prototype system. Prototype yang dibuat disesuaikan dengan kebutuhan system yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.

3. Uji coba

Pada tahap ini, Prototype dari system di uji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki Prototype yang ada.

Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. Setelah rancangan perangkat lunak disetujui oleh si pengguna, maka langkah selanjutnya dilakukan proses pengkodean sistem dan pengujian sistem. Selanjutnya langkah terakhir yang dilakukan setelah perangkat lunak selesai adalah penyerahan kepada pelanggan untuk siap digunakan. Ada beberapa jenis prototyping diantaranya *evolutionary prototyping*, *throw-away prototyping*, dan *user-interface prototyping*. Jenis prototyping model yang digunakan dalam penelitian ini adalah *Evolutionary prototyping*.

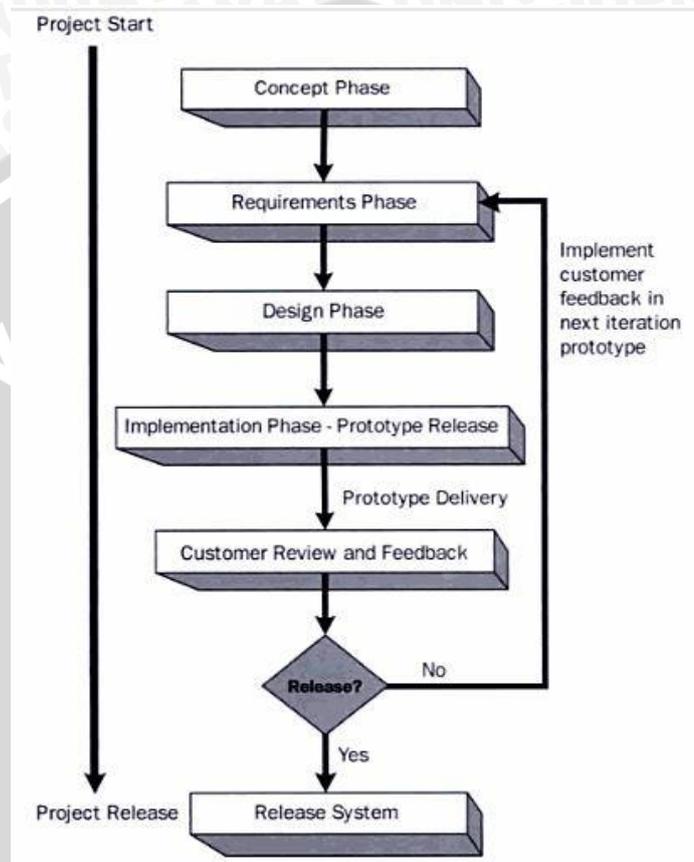
2.2.4.1 Pendekatan *Evolutionary prototyping*

Evolutionary prototyping adalah model lifecycle perangkat lunak di mana sistem dikembangkan secara bertahap sehingga mudah untuk dimodifikasi dalam menanggapi kebutuhan pengguna dan feedback dari pengguna. Pengembangan evolusioner prototyping dimulai dengan prototipe user interface dan kemudian mengembangkan sistem dari kebutuhan user.

Prototyping Evolusioner tidak membuang kode prototyping yang telah dibuat tetapi mengembangkan menjadi kode yang dibutuhkan user. Prototype evolusioner terus menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem yang baru. Prototype ini kemudian dilanjutkan dengan produksi. Jadi satu prototype evolusioner akan menjadi sistem actual. Prototype evolusioner digunakan untuk tujuan menyerahkan sistem kepada user untuk menjalankan semua prioritas utama.[MCC-96].

2.2.4.2 Fase *Evolutionary prototyping*

Berikut adalah fase-fase dalam metode *evolutionary prototyping* yang dilakukan ditunjukkan pada gambar 2.3.



Gambar 2.3 Fase pada metode *Evolutionary prototyping*
Sumber [LEN-03]

Berikut ini merupakan penjelasan dari fase pendekatan *evolutionary prototyping*:

1. Fase Analisa Kebutuhan

Pada tahap ini dilakukan analisa kebutuhan dari system dengan cara bertemu dengan pelanggan. Untuk membuat suatu system yang sesuai kebutuhan untuk prototype dalam iterasi yang sedang dibuat.

2. Fase Perancangan

Dari tahap kebutuhan di pecah-pecah menjadi beberapa bagian fungsi system, selanjutnya dilakukan perancangan sesuai dengan kebutuhan customer. Jika customer tidak puas maka dilakukan iterasi hingga customer setuju.

3. Implementasi – Fase Rilis Prototype

Pada tahap ini, implementasi Prototype dari hasil spesifikasi kebutuhan perancangan system. Selanjutnya diberikan kepada customer untuk di review.

4. Customer Review dan Feedback

Pada tahap ini dilakukan Review oleh customer dan customer memberikan masukan tentang sistem yang telah dibuat, serta memberikan perubahan dan fitur baru untuk dibuat pada iterasi selanjutnya, atau prototype disetujui oleh customer dan menjadikan prototype sebagai final release produk. [LEN-03]

2.2.4.3 Keuntungan dan kelebihan *Evolutionary prototyping*

Keuntungan *evolutionary prototyping* adalah:

- Penyerahan sistem yang dipercepat, sehingga dapat diantisipasi keterlambatan karena perubahan sistem.
- Keterlibatan user dengan sistem lebih awal dan lebih lama, sehingga menumbuhkan kepercayaan user.

Kelebihan *evolutionary prototyping* adalah :

- Fitur product dapat dilihat langsung oleh costumer pada tahap awal, dan costumer dapat langsung memberikan feedback
- Perubahan kebutuhan oleh costumer mudah untuk diakomodir oleh pengembang dan mudah untuk memperbaiki kesalahan dengan cepat.
- Kebutuhan menjadi lebih jelas untuk pembuatan software yang sesuai
- Mudah untuk dirancang dan pengembangan sistem disupport oleh model [SAB-09].

2.2.4.4 Perbedaan *Evolutionary prototyping* dengan *prototyping*

Perbedaan *Evolutionary prototyping* dengan *Prototyping* terletak pada fase –fase yang digunakan, berikut perbedaan *Evolutionary prototyping* dan *Prototyping* dijelaskan pada tabel 2.1.

Table 2.1 Perbedaan metode

Metode	Fase	Uraian
1. <i>Prototyping</i>	<ol style="list-style-type: none"> 1. Mendengarkan pelanggan 2. Membangun dan memperbaiki mock up 3. Pelanggan melihat /menguji mock up 	<ul style="list-style-type: none"> - Menghasilkan kebutuhan customer - Menghasilkan prototype - Menghasilkan evaluasi prototype
2. <i>Evolutionary prototyping</i>	<ol style="list-style-type: none"> 1. Fase konsep 2. Requirement 3. Perancangan 4. Prototype release 5. Customer review 6. Release System 	<ul style="list-style-type: none"> - Menghasilkan konsep awal sistem - Menghasilkan daftar kebutuhan, untuk digunakan pada iterasi - Menghasilkan prototype dan rancangan - Menghasilkan prototype untuk di rilis - Menghasilkan feedback dari customer - Dilakukan realease sistem

2.2.5 Konsep Unified Modelling Language (UML)

2.2.5.1 Pengertian Unified Modelling Language (UML)

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang yang berada di belahan negara manapun dapat mengerti pemodelan perangkat lunak yang telah dibuat. Banyak orang yang telah membuat bahasa pemodelan perangkat lunak dan yang berkembang saat itu bahasa pemrograman *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan bahasa terstruktur.

Pada perkembangan teknik pemrograman berorientasi objek, muncul pemodelan *Unified Modelling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak [ROS-11].

Dalam membangun suatu model perangkat lunak dengan UML, digunakan bentuk-bentuk diagram atau symbol untuk merepresentasikan

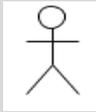
elemen-elemen dalam sistem. Bentuk diagram yang digunakan untuk merepresentasikannya adalah sebagai berikut [PRY-05] :

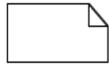
- *Use case Diagram*
- *Class Diagram*
- *State Diagram*
- *Sequence Diagram*
- *Collaboration Diagram*
- *Activity Diagram*

2.2.5.2 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk perlakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dalam *Use-case diagram* penekanannya adalah “apa” yang diperbuat oleh sistem, dan bukan “bagaimana”. Syarat pemakaian *use case* adalah nama didefinisikan sesimpel mungkin. Ada beberapa hal utama pada *use case* yang dijelaskan pada tabel 2.2.

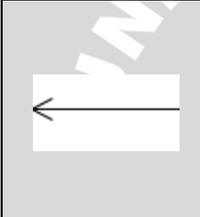
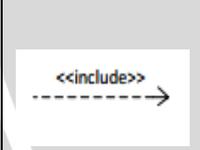
Table 2.2 Keterangan Umum Pada Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
3		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
4		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

5		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
---	---	------	---

Beberapa simbol garis hubungan pada diagram *use case* yang sering digunakan dijelaskan pada tabel 2.3 yaitu :

Table 2.3 Keterangan Hubungan Pada Use Case Diagram

Gambar Garis Hubungan	Nama	Keterangan
	<i>Communicate</i>	garis yang menghubungkan antara aktor dengan <i>use case</i>
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Extends</i>	untuk menetapkan bahwa <i>use case</i> yang dituju memperpanjang perilaku dari sumber <i>use case</i> .
	<i>Include</i>	untuk menetapkan bahwa <i>use case</i> awal secara eksplisit menggabungkan perilaku dari <i>use case</i> yang ditunjuk oleh <i>use case</i> awal.

Sumber : [ROS-13]

2.2.5.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Dalam setiap kelas memiliki tiga area pokok :

- Nama (dan *stereotype*)
- Atribut merupakan variabel yang dimiliki oleh suatu kelas.
- Metode/ operasi adalah fungsi yang dimiliki oleh suatu kelas.

Atribut dan metode dalam *class diagram* dapat memiliki salah satu sifat seperti berikut di bawah ini [PRY-05] :

- *Private* (-), hanya dapat diakses oleh *class* itu sendiri
- *Protected* (#), hanya dapat diakses oleh *class* itu sendiri dan turunan dari *class* tersebut
- *Public* (+), dapat diakses oleh *class* selain dari *class* yang bersangkutan

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Kelas – kelas yang ada pada struktur sistem harus dapat melakukan fungsi – fungsi sesuai dengan kebutuhan sistem sehingga *programmer* dapat membuat kelas – kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis – jenis kelas berikut [PRY-05]:

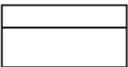
- Kelas main
- Kelas yang menangani tampilan (*view*)
- Kelas yang diambil dari pendefinisian *use case* (*controller*)
- Kelas yang diambil dari pendefinisian data (*model*)

Beberapa simbol pada diagram *class diagram* yang sering digunakan dijelaskan pada tabel 2.4 yaitu :

Table 2.4 Keterangan Simbol Pada Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Agregation</i>	Hubungan yang menyatakan bagian (“terdiri atas..”) atau biasa disebut relasi mempunyai sebuah.



3		<i>Class</i>	Himpunan dari objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan hasil yang terukur bagi suatu aktor
5		<i>Composition</i>	sebuah kelas tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.2.5.4 Sequence Diagram

Diagram sekuen menggambarkan kolaborasi yang dinamis antara obyek satu dengan yang lain. Kolaborasi ini ditunjukkan dengan adanya interaksi antar obyek di dalam dan di sekitar sistem yang berupa pesan atau instruksi yang berurutan. *Sequence diagram* umumnya digunakan untuk menggambarkan suatu skenario atau urutan langkah-langkah yang dilakukan baik oleh *actor* maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil atau *output* [PRY-05].

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri, jadi semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Beberapa simbol pada *sequence diagram* yang sering digunakan dijelaskan pada tabel 2.5 yaitu :

Table 2.5 Keterangan Notasi Pada Sequence Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menggambarkan orang yang berinteraksi dengan sistem
2		<i>Boundary</i>	Menggambarkan interaksi antara satu atau lebih actor di sistem
3		<i>Control</i>	perilaku mengatur, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem
4		<i>Entity</i>	Menggambarkan informasi yang harus disimpan oleh sistem.
5		<i>Lifeline</i>	Mengindikasikan keberadaan sebuah obyek dalam basis waktu
6		<i>Message</i>	pesan/hubungan antar objek, yang menunjukkan urutan kejadian yang terjadi

Sumber : [PRY-05]

2.2.5.5 ActivityDiagram

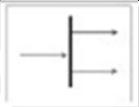
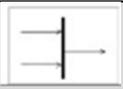
Activity diagram atau aktifitas diagram ini menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis yang digunakan untuk mendeskripsikan alur kegiatan yang lainnya seperti *use case* atau interaksi.

Activity diagram juga banyak digunakan untuk mendefinisikan hal-hal berikut [ROS-13] :

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan itu proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokkan tampilan dari sistem (*user interface*) dimana setiap aktivitas dianggap memiliki sebuah rancangan tampilan prototype.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan pengujian yang perlu didefinisikan kasus ujinya.
- Rancangan menu yang ditampilkan pada perangkat lunak.

Beberapa simbol pada *activity diagram* yang sering digunakan dijelaskan pada tabel 2.6 yaitu :

Table 2.6 Keterangan Notasi Pada Activity Diagram

No	Gambar	Keterangan
1		Start poin
2		End point
3		Activities
4		Fork (percabangan)
5		Join (penggabungan)
6		Decision
7	swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan actor atau pengelompokan dalam sebuah urutan yang sama

Sumber : [PRY-05]

2.2.5.6 Diagram hubungan entitas (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

ERD untuk memodelkan struktur data dan hubungan antara data, untuk menggambarannya digunakan beberapa notasi dan symbol. Pada dasarnya ada tiga symbol yang digunakan, yaitu [ROS-11]:

1. Entiti

Merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

2. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain.

3. Hubungan atau Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu [KAD-01]:

1) One to One (1 : 1)

Yaitu hubungan antara entity pertama dapat berhubungan dengan satu entity kedua dan entity kedua dapat berhubungan dengan entity pertama paling banyak satu entity.

2) One to Many (1 : M)

Entity pertama dapat berhubungan dengan sejumlah entity kedua, tetapi satu entity kedua hanya dapat berhubungan dengan satu entity pertama.

3) Many to Many (M : N)

Entitas pertama dapat berhubungan dengan banyak pada entity kedua, demikian pula sebaliknya.

2.2.6 Konsep Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks maupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi [ROS-13]

- Memasukkan, menyimpan, dan mengambil data.
- Membuat laporan berdasarkan data yang telah disimpan.

Tujuan dari dibuatnya tabel adalah untuk menyimpan data ke dalam tabel agar mudah diakses. Oleh karena itu dalam merancang tabel diperlukan pemikiran karena data yang akan disimpan nantinya dalam bentuk baris data.

2.2.6.1 Relational Database Management System (RDBMS)

RDBMS (*Relational Database Management System*) adalah perangkat lunak untuk membuat dan mengelola database, sering juga disebut sebagai database engine. Istilah RDBMS, database server-software, dan database engine mengacu ke hal yang sama, sedangkan RDBMS bukanlah database.

Relational Database Management System (RDBMS) adalah sebuah sistem yang secara otomatis menyatukan semua DBMS yang saling berhubungan. RDBMS biasanya menggunakan 4th Generation Language (4GL) dan sangat fleksibel sehingga data dapat dimodifikasi dengan mudah, demikian pula dengan struktur databasenya. Pengelolaannya sendiri biasanya ditangani oleh tenaga ahli yang spesialis menangani DBMS yang disebut sebagai DBA (*Database Administrator*).

Berikut ini adalah 4 macam DBMS versi komersial yang paling banyak digunakan dunia saat ini yaitu *Oracle*, *Microsoft SQL Server*, *IBM DB2* dan *Microsoft Access*. Sedangkan DBMS versi *open source* yang cukup berkembang dan paling banyak digunakan saat ini adalah *MySQL*, *PostgreSQL*, *Firebird*, dan *SQLite*.

2.2.7 Konsep Pengujian Perangkat Lunak

Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevolusi. Metode-metode ini menyediakan developer pendekatan sistematis untuk pengujian. Terlebih lagi metode-metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tertinggi untuk menemukan kesalahan-kesalahan dalam perangkat lunak[PRE-10].

Strategi untuk proses pengujian perangkat lunak yaitu mengintegrasikan metode desain kasus uji perangkat lunak pada serangkaian langkah yang disusun dengan baik dan menghasilkan konstruksi perangkat lunak yang berhasil. Strategi pada pengujian dapat dilakukan melalui pengujian tingkat rendah yakni pengujian yang dilakukan pada *source code* program untuk memberikan bukti segmen kode sumber yang kecil telah diimplementasikan dengan tepat. Selain itu juga ada pengujian tingkat tinggi yang memvalidasi fungsi-fungsi sistem mayor [STK-17].

2.2.7.1 Pengujian Validasi dengan Metode Blackbox

Pengujian sistem dilakukan dengan melakukan pengujian validasi dengan metode *blackbox*. Pengujian metode *blackbox* merupakan metode yang digunakan untuk mengetahui apakah sistem berfungsi dengan baik dan sesuai dengan harapan tanpa memperhatikan struktur logika internal perangkat lunak. Sedangkan pengujian validasi yaitu untuk mengetahui apakah sistem yang sudah dibangun telah benar sesuai dengan kebutuhan.

Software yang berintegrasi diuji berdasarkan pada kebutuhan untuk memastikan bahwa kita memiliki sistem yang benar. Fokus-nya adalah untuk meng-*uncover error* pada:

- *Input/output* sistem
- Informasi fungsi sistem dan data
- *Interface* sistem dengan bagian eksternal
- *User interface*
- Perilaku dan *performance* sistem

2.2.7.2 Pengujian White Box

Pengujian white box adalah pengujian yang didasarkan pada pengecekan terhadap detil perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Penentuan kasus uji disesuaikan dengan struktur system, pengetahuan mengenai program digunakan untuk mengidentifikasi kasus uji tambahan.

Tujuan penggunaan white box untuk menguji semua statement program.

Penggunaan metode pengujian white box dilakukan untuk :

1. memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali
2. menggunakan semua keputusan logis untuk semua kondisi true/false
3. mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.
4. menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

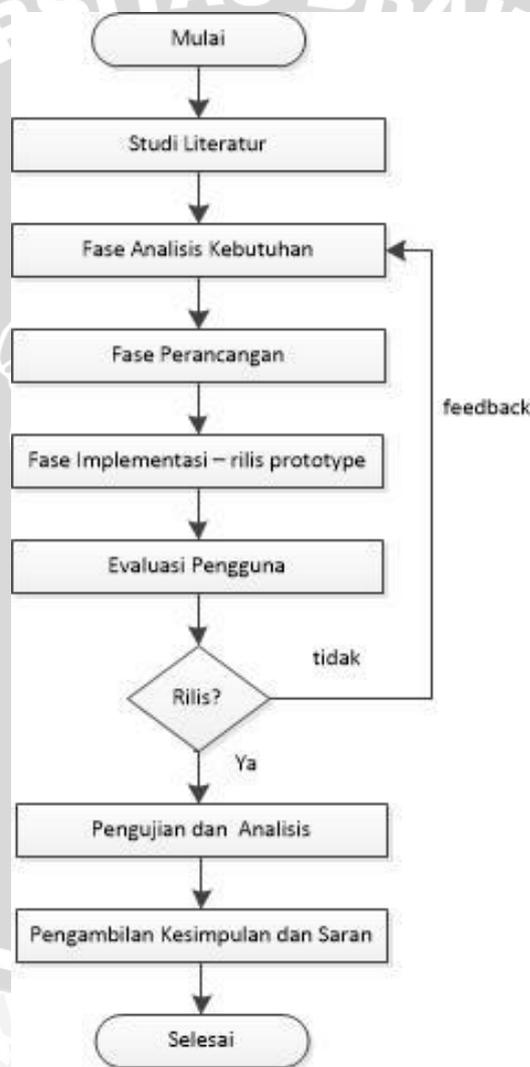
Dengan menggunakan metode pengujian *white-box*, perancang sistem dapat melakukan kasus uji yang memberikan jaminan bahwa semua jalur independen telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi *true* dan *false*, dan menggunakan struktur data *internal* untuk menjamin validitasnya. Jenis pengujian *white-box* antara lain adalah *Basis Path Testing* dan *Control Structure Testing* [SIM-10].

2.2.7.3 Pengujian Compatibility

Pengujian yang dilakukan untuk menguji bahwa fitur produk sistem bekerja konsisten dengan komponen infrastruktur yang berbeda. Pengujian berfungsi untuk menentukan lingkungan yang diharapkan dapat menjalankan aplikasi yang dikembangkan. Semakin aplikasi dapat berjalan di banyak jenis perangkat yang berbeda, maka semakin baik aspek compatibilitynya [DES-06].

BAB 3 METODOLOGI PENELITIAN

Dalam bab ini membahas metode dan tahapan pengerjaan yang dilakukan dalam penelitian. Penelitian ini menerapkan pendekatan *evolutionary prototyping* pada pengembangan perangkat lunak. Penelitian ini dimulai dari studi literature, fase analisa kebutuhan, fase perancangan, fase implementasi sistem, evaluasi pengguna dengan memberikan *feedback* dan pengujian serta pengambilan kesimpulan dan saran. Berikut adalah diagram alir yang dilakukan ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Alir Bab Metodologi Penelitian

3.1 Tahap Studi Literatur

Studi literature menjelaskan dasar teori yang digunakan sebagai sumber acuan untuk penunjang dan pendukung penulisan skripsi dengan pemahaman kepustakaan (*library research*). Sumber atau referensi yang digunakan antara lain buku, jurnal, laporan penelitian, dan bantuan mesin pencari (*search engine*) internet. Teori dan pustaka yang berkaitan dengan penelitian ini meliputi :

1. Sport Club House Pertamina - UB
2. Konsep Sistem Informasi
3. Konsep Rekayasa Perangkat Lunak
 - a. Pengertian Rekayasa Perangkat Lunak
4. Konsep *Prototyping* dan metode *Evolutionary prototyping*
5. Konsep UML (*Unified Modeling Language*)
 - a. Pengertian UML (*Unified Modeling Language*)
 - b. *Use Case Diagram*
 - c. *Class Diagram*
 - d. *Sequence Diagram*
 - e. *Activity Diagram*
 - f. *ERD*
6. Basis Data
 - a. Pengertian Basis Data
 - b. RDBMS
7. Pengujian Perangkat Lunak
 - a. Pengujian Blackbox
 - b. Pengujian White Box
 - c. Pengujian Compatibility

3.2 Fase Analisa Kebutuhan

Analisis kebutuhan atau tahapan mendengarkan pelanggan merupakan tahap yang paling penting dalam suatu pengembangan sebuah aplikasi, karena tahapan yang pertama kali harus dilakukan sebelum melakukan perancangan dan pembuatan perangkat lunak. Dimana pada tahap ini pengumpulan kebutuhan dilakukan oleh developer dan klien yang bertemu dengan melakukan wawancara

dengan staff dan Koordinator Sport Club House Pertamina - UB kemudian menentukan gambaran umum, identifikasi aktor-aktor yang terlibat di dalam sistem. Dari hasil tersebut tentunya ada beberapa yang perlu ditambahkan untuk nantinya digunakan sebagai pondasi dalam perancangan sistem. Metode analisis yang digunakan dengan *Object Oriented Analysis* menggunakan bahasa pemodelan UML (*Unified Modeling Language*). Diagram *Use Case* digunakan untuk mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas sistem dari perspektif *user*. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan sistem.

3.3 Fase Perancangan

Pada tahap ini dilakukan perancangan yang dibuat disesuaikan dengan yang telah didefinisikan sebelumnya dari kebutuhan pengguna. Pertama kali pada tahap ini akan menghasilkan perancangan awal kemudian perancangan iterasi 1 dst, dan akan terus diulang kembali sampai iterasi dimana pengguna merasa puas. Perancangan yang dibuat dapat dimanfaatkan kembali untuk membangun software lebih cepat. *Prototype* dibuat berdasarkan *Object Oriented Analysis* dan *Object Oriented Design* yaitu menggunakan pemodelan UML (*Unified Modeling Language*) seperti *use case diagram*, *class diagram*, *sequence diagram*, *activity diagram* dan perancangan interface yang akan dibuat sesuai dengan kebutuhan pengguna.

3.4 Fase Implementasi

Pada tahap ini dilakukan implementasi prototype atau pengkodean sistem dari perancangan sebelumnya. Implementasi yaitu melakukan kegiatan pengkodean atau yang biasa dikenal dengan proses *coding*, yaitu proses penterjemahan sistem ke dalam kode program dengan bahasa pemrograman. Implementasi yang akan dibuat berupa sistem informasi pada Sport Club House Pertamina - UB berbasis web. Implementasi perangkat lunak dilakukan dengan menggunakan implementasi *basis data* MySQL dengan *software* XAMPP 1.7.3 dan bahasa pemrograman PHP dengan framework Codeigniter. Dimana pada fase ini menghasilkan implementasi prototype versi 0.0 untuk perancangan pertama

kali, setelah melakukan iterasi maka menghasilkan implementasi prototype versi 1.0 hingga menghasilkan implementasi rilis system.

3.4.1 Evaluasi Pengguna

Pada tahap ini proses evaluasi dilakukan oleh pengguna. Pengguna dapat menilai apakah sistem awal yang dibangun telah sesuai dengan yang diharapkan. Jika tidak, maka dilakukan revisi dan *feedback* dengan menyesuaikan kebutuhan yang diinginkan pengguna. Maka dengan adanya proses ini diharapkan perangkat lunak nantinya akan sesuai dengan kebutuhan. Kemudian dilakukan review kekurangan-kekurangan dari kebutuhan pelanggan dengan analisis kebutuhan atau mendengarkan kembali keluhan dari pelanggan untuk memperbaiki sistem yang ada dan tidak dibuang. Setelah proses ini dilakukan maka akan dilakukan redesain perancangan sistem sesuai dengan hasil evaluasi *prototype*.

3.5 Pengujian dan Analisis

Pengujian sistem pada penelitian ini ditujukan untuk dapat menunjukkan bahwa sistem telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan sistem. Pengujian dilakukan dengan tiga metode yaitu pengujian black box, pengujian white box dan pengujian compatibility.

1. Pengujian *black box* dilakukan melalui tahapan pengujian validasi untuk menguji fungsional kerja dari sistem. Skenario pada pengujian ini dilakukan dengan menjalankan sistem sesuai dengan fungsi-fungsi internal yang akan dijabarkan pada bab analisis dan perancangan sub bab *use case scenario* dengan mencoba fitur pada sistem minimal 3 kali percobaan. jika dari minimal 3 kali percobaan sistem berjalan tanpa ada kesalahan maka pengujian dianggap valid. Sistem berjalan dengan baik sesuai dengan kebutuhan jika minimal 80% dari pengujian menunjukkan hasil yang valid. Setelah itu dilakukan pencatatan dalam bentuk tabel apakah hasil kebutuhan fungsional berhasil terpenuhi oleh sistem (valid) atau tidak (tidak valid).

2. Pengujian *white box* dilakukan dengan menggunakan teknik *basis path testing*. Teknik pengujian ini dilakukan dengan menggunakan rancangan atau kode program sebagai dasar untuk membuat *flow graph*. Setelah itu, berdasarkan *flow graph* yang telah dibuat akan ditentukan kompleksitas siklomatik serta himpunan basis dari jalur-jalur independen secara linier.

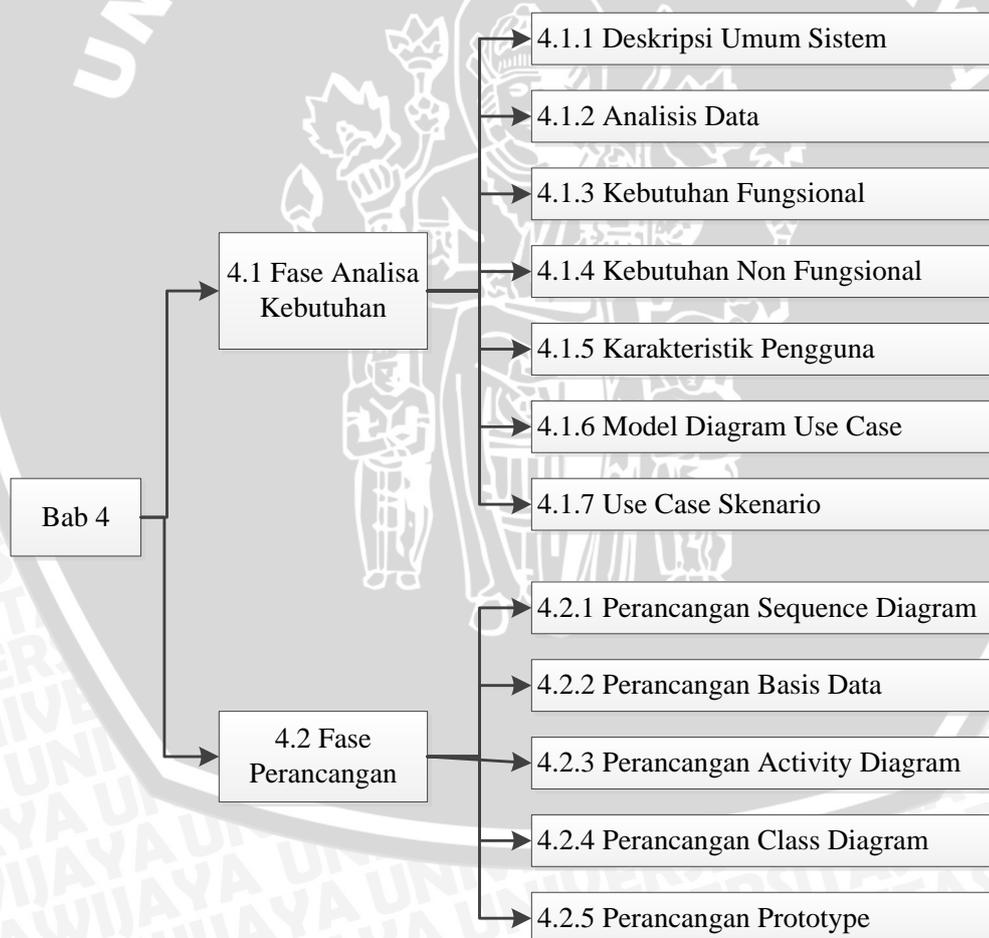
3. Pengujian *compatibility* akan dilakukan pengujian untuk mengetahui kinerja sistem pada OS dan browser yang berbeda. Pengujian ini dilakukan dengan menggunakan daftar kebutuhan dan scenario pengujian. Setelah itu dilakukan hasil pengujian apakah sistem berjalan normal di platform atau browser lainnya.

3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktik. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan ini adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan sistem informasi selanjutnya.

BAB 4 ANALISIS DAN PERANCANGAN

Pada bab ini membahas mengenai analisis dan perancangan sistem. Fase analisa kebutuhan sistem informasi terdiri atas tujuh langkah yaitu melakukan penjabaran tentang deskripsi umum sistem, melakukan proses analisis data yang diperlukan, melakukan analisa kebutuhan fungsional, non fungsional, karakteristik pengguna yang terlibat dalam aplikasi, membuat pemodelan diagram use case dan use case skenario. Fase perancangan aplikasi terdiri dari lima langkah yaitu pemodelan diagram sequence, pemodelan basis data, pemodelan diagram class, pemodelan activity diagram dan perancangan prototype. Berikut ini merupakan blok diagram bab analisis & perancangan yang dijelaskan pada gambar 4.1.



Gambar 4.1 Diagram Blok Bab Analisis dan Perancangan

4.1 Fase Analisa Kebutuhan

Tahapan awal adalah pengumpulan kebutuhan atau analisa kebutuhan ini diawali dengan melakukan teknik wawancara secara terstruktur dan nonterstruktur dengan pihak Sport Club House Pertamina - UB yang dijelaskan pada Lampiran 1 untuk wawancara terstruktur. Hasil dari teknik tersebut dijabarkan dengan gambaran umum aplikasi, identifikasi aktor yang terlibat, analisis data yang akan disimpan, penjabaran tentang daftar kebutuhan fungsional dan non fungsional, tahapan ini bertujuan untuk menggambarkan kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

Pada fase ini merupakan analisis hasil iterasi ke2 dari versi 1.0. Pada perancangan awal dan iterasi ke 1 dijelaskan dalam Lampiran 2 tentang perancangan iterasi.

4.1.1 Deskripsi Umum Sistem Informasi

Sistem informasi ini dapat digunakan untuk membantu pengguna dalam hal ini adalah masyarakat umum, mahasiswa UB, dan pegawai UB. Sistem informasi ini memiliki delapan bagian utama berdasarkan gambar 4.2 yaitu, subsistem untuk pengguna (*user*), subsistem untuk anggota(member account), subsistem untuk Admin, subsistem keuangan, subsistem bendahara, subsistem Bagian Gedung, subsistem Kasir, subsistem Front Office.

a. Sistem informasi bagi *User*

Sistem informasi untuk mahasiswa adalah sistem informasi yang berfungsi sebagai alat bantu pengguna dalam mengetahui informasi-informasi seputar Sport Club House Pertamina - UB serta melakukan proses register account untuk menjadi member account.

b. Sistem informasi bagi Member Account

Sistem informasi untuk member adalah sistem informasi yang berfungsi sebagai alat bantu dalam proses pendaftaran membership, pemesanan tempat olahraga dan dapat melakukan complain.

c. Sistem informasi bagi *Front office*

Sistem informasi untuk front office adalah sistem informasi yang digunakan untuk melakukan pendaftaran olahraga member seperti

fitness, aerobic, yoga dan dapat mengelola data member yang akan mengikuti olahraga tersebut.

d. Sistem informasi bagi Kasir

Sistem informasi untuk kasir adalah sistem informasi yang digunakan untuk mengelola data pembayaran membership.

e. Sistem informasi bagi Bagian Gedung

Sistem informasi untuk bagian gedung adalah sistem informasi yang digunakan untuk mengelola validasi pemesanan serta dalam pemesanan tempat sekaligus mengelola pemasukan keuangan.

f. Sistem Informasi bagi Admin

Sistem informasi untuk admin adalah sistem informasi yang digunakan untuk mengelola complain dari member, mengelola pegawai, berita, fasilitas dan tips olahraga.

g. Sistem Informasi bagi Keuangan

Sistem informasi bagi keuangan adalah sistem informasi yang digunakan untuk melihat keuangan yang berasal dari bagian gedung dan kasir.

h. Sistem Informasi bagi Bendahara

Sistem informasi bagi bendahara adalah sistem informasi yang digunakan untuk melihat laporan keuangan dari bagian keuangan.

4.1.2 Analisis Data

Analisis data bertujuan untuk mendapatkan struktur penyimpanan data yang dibutuhkan. Struktur penyimpanan data pada aplikasi disusun berdasarkan analisis data sebagai berikut :

1. Data member yang terdiri dari nama, TTL, fakultas, alamat asal, alamat Malang, noHp, no identitas.
2. Data pegawai yang terdiri dari nama atau *username* seluruh pengguna sistem, *password*, serta rule yang membedakan antara pengguna sistem informasi sebagai admin, bagian gedung, kasir, *front office*, bendahara, keuangan dan jabatan.
3. Data olahraga dan harga masing-masing olahraga

4. Data keuangan atau pemasukan yang terdiri dari pemasukan pencatatan kasir dan pemasukan pemesanan .

4.1.3 Kebutuhan Fungsional

Tahap ini merupakan kebutuhan fungsional yang didapat dari hasil wawancara dengan pengguna. Kebutuhan fungsional ini merupakan kebutuhan fungsional akhir (iterasi kedua) yang akan digunakan di dalam sistem, dimana sebelumnya telah mengalami perubahan atau penambahan di setiap iterasi sesuai yang dijelaskan pada Lampiran 2. Prioritas pada spesifikasi kebutuhan fungsional didapat dari hasil wawancara dengan mengetahui seberapa penting atau sering dari proses bisnis Sport Club House Pertamina-UB dan seberapa jarang proses ini digunakan. Prioritas kebutuhan fungsional digunakan untuk proses implementasi dalam sistem yang diprioritaskan dan sebagai acuan pengujian whitebox dengan basis path yang memiliki prioritas tinggi. Kebutuhan fungsional yang nantinya akan disediakan oleh sistem informasi ini yang dijelaskan pada tabel 4.1 berikut:

Tabel 4.1 Kebutuhan Fungsional iterasi ke 2

Kode Fungsi	Fungsi	Deskripsi	Prioritas
KF_01	Login	Sistem mampu menyediakan fungsi login dan memberi batasan hak akses.	high
KF_02	Melihat Berita	Sistem harus mampu menampilkan berita Sport Club House PertaminaUB	medium
KF_03	Melihat Tips Olahraga	Sistem harus mampu menampilkan tips olahraga	medium
KF_04	Melihat Fasilitas	Sistem harus mampu menampilkan fasilitas di Sport Club House Pertamina - UB	medium
KF_05	Register Account	Sistem harus mampu menyediakan fungsi Register Account(pendaftaran)	high
KF_06	Pemesanan	Sistem harus mampu melakukan	high

	Tempat	pemesanan tempat dengan melihat jadwal olahraga	
KF_07	Melihat FAQ	Sistem harus mampu menampilkan faq	medium
KF_08	Cetak struk pemesanan	Sistem harus mampu melakukan cetak data pemesanan	high
KF_09	Melihat Riwayat Pemesanan	Sistem harus mampu menampilkan riwayat pemesanan tempat	medium
KF_10	Cetak struk membership	Sistem harus mampu melakukan cetak data pendaftaran membership	high
KF_11	Mendaftar Membership	Sistem harus mampu melakukan pendaftaran membership	high
KF_12	Melihat Riwayat Membership	Sistem harus mampu menampilkan riwayat pendaftaran membership	medium
KF_13	Membuat Komplain	Sistem harus mampu melakukan komplain	high
KF_14	Melihat Tagihan	Sistem harus mampu menampilkan tagihan keuangan	medium
KF_15	Menambah Membership	Sistem harus mampu menambah membership	high
KF_16	Melihat data Membership	Sistem harus mampu menampilkan data membership	medium
KF_17	Menghapus Membership	Sistem harus mampu menghapus membership	low
KF_18	Validasi Membership	Sistem harus mampu memvalidasi membership	high
KF_19	Menambah Pembayaran	Sistem harus mampu menambah pembayaran	high
KF_20	Menghapus Pembayaran	Sistem harus mampu menghapus pembayaran	low
KF_21	Melihat Pemasukan	Sistem harus mampu menampilkan data pemasukan membership	medium

	Membership		
KF_22	Validasi Pemesanan	Sistem harus mampu memvalidasi pemesanan tempat	high
KF_23	Menambah Pemesanan	Sistem harus mampu menambah Pemesanan	high
KF_24	Menghapus Pemesanan	Sistem harus mampu menghapus Pemesanan	low
KF_25	Melihat Data Pemesan	Sistem harus mampu menampilkan data pemesan	medium
KF_26	Menambah Pemasukan	Sistem harus mampu meambah pemasukan pemesanan	high
KF_27	Menghapus Pemasukan	Sistem harus mampu menghapus pemasukan pemesanan	low
KF_28	Melihat Pemasukan pemesanan	Sistem harus mampu menampilkan total pemasukan pemesanan	medium
KF_29	Melihat Laporan Keuangan	Sistem harus mampu menampilkan laporan keuangan berdasarkan tanggal	high
KF_30	Membalas Komplain	Sistem harus mampu membalas komplain	high
KF_31	Menambah Berita	Sistem harus mampu menambah berita	high
KF_32	Menghapus Berita	Sistem harus mampu menghapus berita	low
KF_33	Mengubah Berita	Sistem harus mampu mengubah berita	medium
KF_34	Menambah Pegawai	Sistem harus mampu menambah pegawai	high
KF_35	Menghapus Pegawai	Sistem harus mampu menghapus pegawai	low
KF_36	Mengubah Pegawai	Sistem harus mampu mengubah pegawai	medium
KF_37	Menambah	Sistem harus mampu menambah	high

	Fasilitas	fasilitas	
KF_38	Menghapus Fasilitas	Sistem harus mampu menghapus fasilitas	low
KF_39	Mengubah Fasilitas	Sistem harus mampu mengubah fasilitas	medium
KF_40	Menambah Tips	Sistem harus mampu menambah tips	high
KF_41	Menghapus Tips	Sistem harus mampu menghapus tips	low
KF_42	Mengubah Tips	Sistem harus mampu mengubah tips	medium

4.1.4 Kebutuhan Non Fungsional

Tahap ini merupakan kebutuhan non fungsional yang nantinya akan disediakan oleh sistem informasi ini yang dijelaskan pada tabel 4.2 yaitu:

Tabel 4.2 Kebutuhan Non Fungsional

No	Parameter	Deskripsi Kebutuhan
1.	Compatibility	Sistem dapat diakses melalui browser dan sistem operasi yang berbeda.

4.1.5 Karakteristik Pengguna

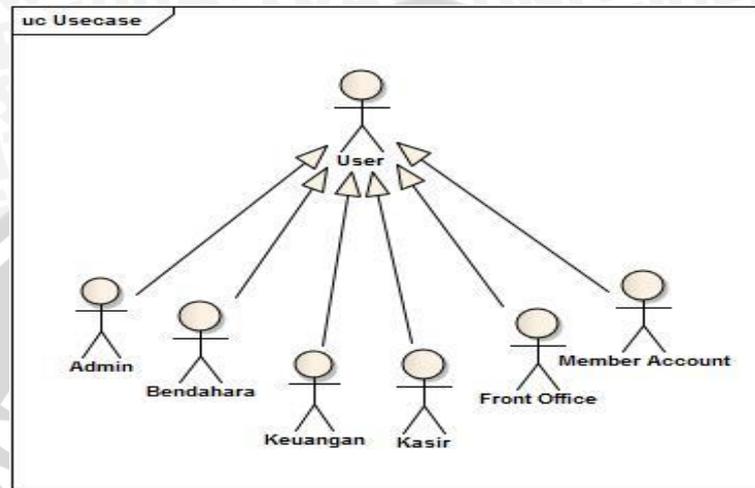
Pada tahap ini adalah untuk melakukan karakteristik terhadap pengguna yang akan berinteraksi dengan aplikasi. Tabel 4.3 memperlihatkan pengguna yang terlibat beserta penjelasannya.

Tabel 4.3 Karakteristik Aktor

Aktor	Tugas	Hak Akses ke Aplikasi
<i>User</i>	<i>User</i> adalah pengguna yang memiliki hak untuk melihat berita, fasilitas dan tips olahraga serta dapat melakukan register account.	KF_02 KF_03, KF_04 KF_05
<i>Member Account</i>	<i>Member account</i> adalah anggota yang telah terdaftar di sistem yang memiliki hak untuk melakukan pemesanan tempat, mendaftar membership, melihat riwayat pemesanan, riwayat membership	KF_01 KF_06 KF_07 KF_08 KF_09 KF_10 KF_11 KF_12 KF_13 KF_14

Front Office	Front office adalah pengguna sistem informasi yang memiliki hak untuk memvalidasi data membership, mendaftarkan olahraga membership serta menghapusnya	KF_01 KF_15 KF_16 KF_17
Kasir	Kasir adalah pengguna sistem informasi yang memiliki hak untuk mengelola data pembayaran membership	KF_01 KF_18 KF_19 KF_20 KF_21
Bendahara	Bagian bendahara adalah pengguna sistem informasi yang memiliki hak untuk melihat laporan keuangan	KF_01 KF_29
Bagian Gedung	Bagian Gedung adalah pengguna sistem informasi yang memiliki hak untuk memvalidasi pemesanan yang dilakukan member account, mengelola pemesanan, dan mengelola pemasukan pemesanan	KF_01 KF_22 KF_23 KF_24 KF_25 KF_26 KF_27 KF_28
Admin	Admin adalah pengguna sistem informasi yang memiliki hak untuk mengelola berita, mengelola tips olahraga, mengelola komplain dan mengelola pegawai, mengelola fasilitas dalam artian bisa melihat, menambah, mengubah dan menghapus.	KF_01 KF_30 KF_31 KF_32 KF_33 KF_34 KF_35 KF_36 KF_37 KF_38 KF_39 KF_40 KF_41 KF_42
Keuangan	Keuangan adalah pengguna sistem informasi keuangan yang memiliki hak untuk melihat pencacatan kasir dan transaksi pemesanan serta laporan keuangan	KF_01 KF_29 KF_21 KF_28

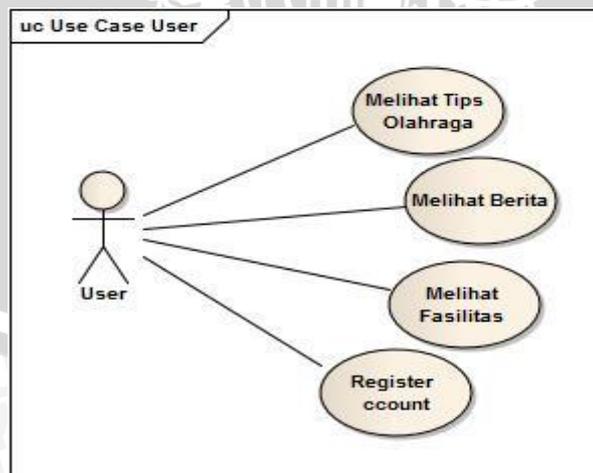
Berdasarkan karakteristik actor pada tabel 4.3 maka model hirarki actor pada gambar 4.2 memperlihatkan hubungan generalisasi terhadap user dan specification terhadap actor selain user.



Gambar 4.2 Model Hirarki Aktor

4.1.6 Model Diagram Use Case Iterasi -2

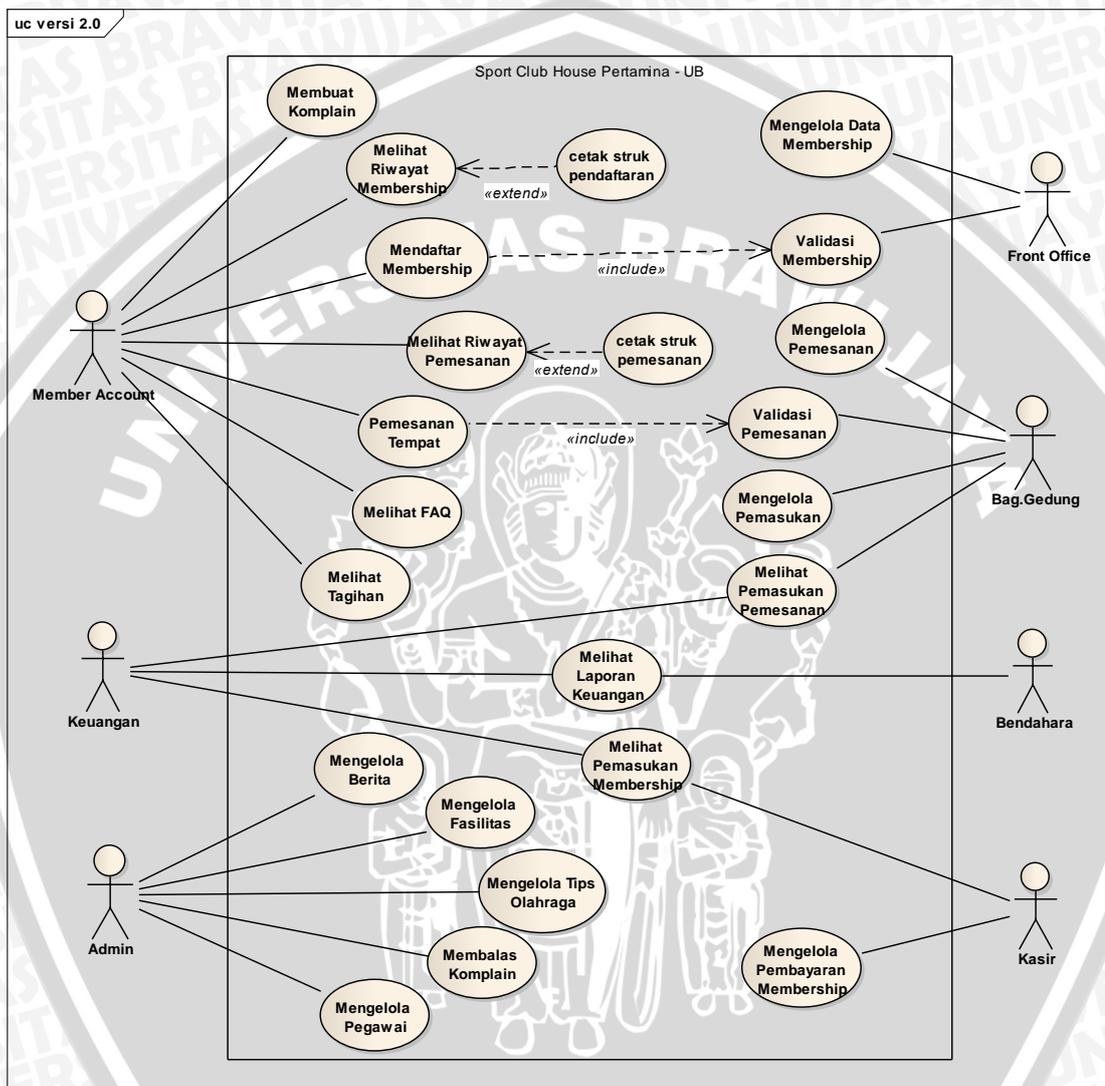
Use case iterasi ke 2 dari actor user. Usecase ini menjelaskan bahwa actor user tanpa melakukan Login ke sistem informasi Sport Club House Pertamina - UB. Dijelaskan dengan gambar 4.3 sebagai berikut:



Gambar 4.3 Model Use Case User iterasi ke 2



Use case global iterasi ke 2 ini menjelaskan bahwa actor member account, bag gedung, kasir, front office, admin, keuangan dan bendahara adalah spesifikasi dari actor User dan bisa melakukan Login ke sistem informasi Sport Club House Pertamina - UB. Dijelaskan dengan gambar 4.4 sebagai berikut:



Gambar 4.4 Model Use Case Diagram iterasi 2

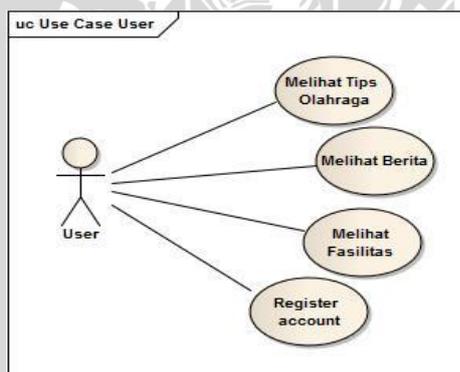
4.1.7 Use Case Skenario

Tabel 4.4 Use Case Skenario login

ID	KF_01
Nama Use-case	Login
Aktor	Member Account, admin, frontoffice, kasir, gedung,



	keuangan, dan bendahara
Deskripsi	Use case ini menjelaskan bagaimana melakukan proses login ke sistem
Pre-condition	User telah masuk ke halaman login
Main Flow	<ol style="list-style-type: none"> 1. User mengisi form login dan menekan tombol login 2. Sistem menampilkan halaman utama masing-masing user
Alternative Flows	<ol style="list-style-type: none"> 1. Jika <i>user</i> tidak valid, maka sistem akan menampilkan notifikasi “username atau <i>Password</i> anda salah” 2. Jika <i>user</i> tidak mengisi salah satu atau beberapa atau semua kolom input, maka sistem akan menampilkan notifikasi “please fill out this field”
Post-condition	user masuk ke halaman utama sistem



Gambar 4.5 Model Use Case Actor *User*

Tabel 4.5 Use Case Skenario Melihat Berita

ID	KF_02
Nama Use-case	Melihat Berita
Aktor	<i>User</i>
Deskripsi	Use case ini menjelaskan bagaimana <i>user</i> melakukan proses untuk melihat berita.
Pre-condition	<i>User</i> sudah masuk ke halaman utama sistem
Main Flow	<ol style="list-style-type: none"> 1. <i>User</i> memilih menu “News” pada sistem. 2. Sistem menampilkan halaman berita
Alternative	<ol style="list-style-type: none"> 1. Jika berita tidak tersedia, maka sistem akan

Flows	menampilkan notifikasi “berita tidak tersedia untuk sementara ini”
Post-condition	Berita yang tersedia telah ditampilkan

Tabel 4.6 Use Case Skenario Melihat Tips Olahraga

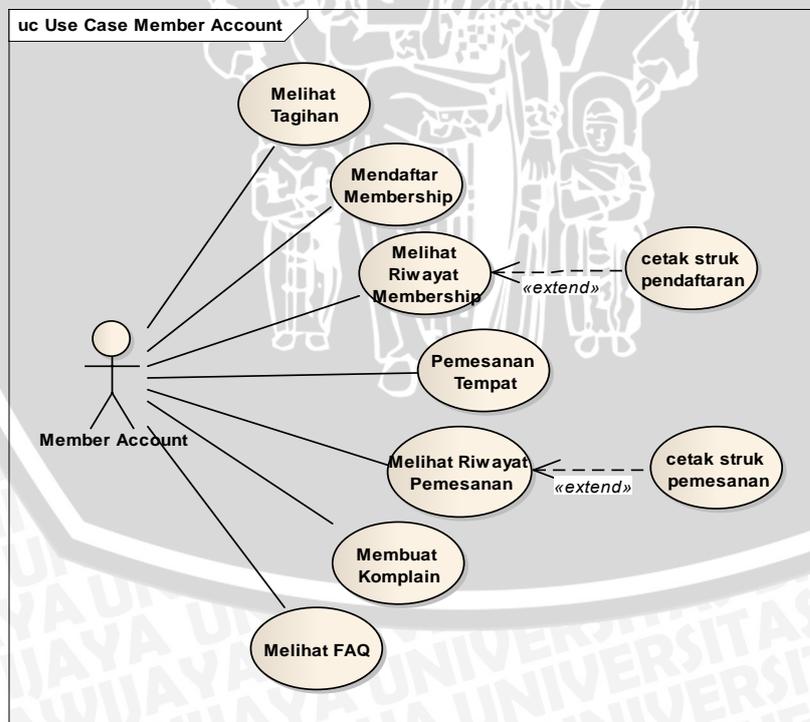
ID	KF_03
Nama Use-case	Melihat Tips Olahraga
Aktor	<i>User</i>
Deskripsi	Use case ini menjelaskan bagaimana <i>user</i> melakukan proses untuk melihat tips olahraga.
Pre-condition	<i>User</i> sudah masuk ke halaman utama sistem
Main Flow	<ol style="list-style-type: none"> 1. <i>User</i> memilih menu “Tips Sport” pada sistem. 2. Sistem menampilkan halaman tips olahraga
Alternative Flows	<ol style="list-style-type: none"> 1. Jika tips olahraga tidak tersedia, maka sistem akan menampilkan notifikasi “tips tidak tersedia untuk sementara ini”
Post-condition	Tips olahraga yang tersedia telah ditampilkan

Tabel 4.7 Use Case Skenario Melihat Fasilitas

ID	KF_04
Nama Use-case	Melihat Fasilitas
Aktor	<i>User</i>
Deskripsi	Use case ini menjelaskan bagaimana <i>user</i> melakukan proses untuk melihat fasilitas
Pre-condition	<i>User</i> sudah masuk ke halaman utama sistem
Main Flow	<ol style="list-style-type: none"> 1. <i>User</i> memilih menu “Facilities” pada sistem. 2. Sistem menampilkan halaman fasilitas
Alternative Flows	<ol style="list-style-type: none"> 1. Jika fasilitas tidak tersedia, maka sistem akan menampilkan notifikasi “fasilitas tidak tersedia untuk sementara ini”
Post-condition	fasilitas yang tersedia telah ditampilkan

Tabel 4.8 Use Case Skenario Register Account

ID	KF_05
Nama Use-case	<i>Register Account</i>
Aktor	<i>User</i>
Deskripsi	Use case ini menjelaskan bagaimana <i>user</i> melakukan proses untuk mendaftar sebagai <i>member account</i> dalam sistem
Pre-condition	Member account sudah masuk ke halaman utama sistem
Main Flow	<ol style="list-style-type: none"> 1. <i>user</i> memilih menu “Register” 2. Sistem menampilkan halaman register account dengan pilihan sebagai mahasiswa UB, pegawai UB atau masyarakat umum. Pilih salah satu 3. mengisi form register dan menekan tombol <i>register</i> 4. sistem menyimpan data user
Alternative Flows	<ol style="list-style-type: none"> 1. Jika <i>user</i> tidak mengisi salah satu atau beberapa atau semua kolom input, maka sistem akan menampilkan notifikasi “please fill out this field”
Post-condition	Sistem berhasil menambahkan data user ke database.



Gambar 4.6 Model Use Case Actor Member Account

Tabel 4.9 Use Case Skenario Pemesanan Tempat

ID	KF_06
Nama Use-case	Pemesanan Tempat
Aktor	<i>Member Account</i>
Deskripsi	Use case ini menjelaskan bagaimana member account melakukan proses untuk memesan tempat olahraga diantaranya futsal dan badminton
Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama member account
Main Flow	<ol style="list-style-type: none"> 1. Member account memilih menu “pemesanan tempat” dan submenu “lihat jadwal” 2. Sistem menampilkan halaman olahraga yang tersedia 3. Member account memilih dan menekan salah satu olahraga 4. Sistem menampilkan halaman jadwal olahraga tersebut 5. Pilih tanggal untuk melihat jadwal terpesan atau terpakai. 6. Mengisi data pemesanan dan menekan tombol “Daftar”. 7. Sistem menambah data pemesanan dan kembali ke halaman utama member account
Alternative Flows	<ol style="list-style-type: none"> 1. Jika input tanggal tidak valid, maka sistem akan menampilkan notifikasi “Pendaftaran gagal disimpan, tanggal kadaluarsa” 2. Jika input jam tidak valid, maka sistem akan menampilkan notifikasi “Pendaftaran gagal disimpan, jam akhir kurang dari jam awal”
Post-condition	Pemesanan tempat telah berhasil ditambahkan

Tabel 4.10 Use Case Skenario Melihat FAQ

ID	KF_07
Nama Use-case	Melihat FAQ
Aktor	<i>Member Account</i>
Deskripsi	Use case ini menjelaskan bagaimana member account melakukan proses untuk melihat faq

Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama <i>member</i> account
Main Flow	<ol style="list-style-type: none"> 1. <i>Member</i> account memilih menu “FAQ” 2. Sistem menampilkan halaman faq
Alternative Flows	<ol style="list-style-type: none"> 1. <i>Member</i> account tidak memilih menu “FAQ” 2. Sistem tidak menampilkan halaman faq
Post-condition	FAQ yang tersedia telah ditampilkan

Tabel 4.11 Use Case Skenario cetak struk pemesanan

ID	KF_08
Nama Use-case	Cetak struk pemesanan
Aktor	<i>MemberAccount</i>
Deskripsi	Use case menjelaskan bagaimana <i>member</i> account melakukan proses untuk cetak struk pemesanan
Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama <i>member</i> account
Main Flow	<ol style="list-style-type: none"> 1. Memilih menu “Pesan Tempat” dan pilih submenu “Lihat Riwayat Pemesanan” 2. Sistem akan menampilkan halaman riwayat pemesanan. 3. <i>Member</i> account memilih icon print pada salah satu kolom dengan status belum divalidasi.
Alternative Flows	<ol style="list-style-type: none"> 1. Jika pilih icon print dengan status selain ‘belum divalidasi’, maka sistem akan menampilkan notifikasi “cetak struk gagal, status sudah divalidasi”
Post-condition	struk pemesanan telah ditampilkan

Tabel 4.12 Use Case Skenario Melihat Riwayat Pemesanan

ID	KF_09
Nama Use-case	Melihat Riwayat Pemesanan
Aktor	<i>Member Account</i>
Deskripsi	Use case ini menjelaskan bagaimana <i>member</i> account melakukan proses untuk melihat riwayat pemesanan tempat

Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama member account
Main Flow	<ol style="list-style-type: none"> 1. Member account memilih menu “Pesan Tempat” dan pilih submenu “Lihat Riwayat Pemesanan” 2. Sistem menampilkan riwayat pemesanan.
Alternative Flows	<ol style="list-style-type: none"> 1. Jika riwayat pemesanan tidak tersedia, maka sistem akan menampilkan notifikasi “riwayat pemesanan tidak tersedia untuk sementara ini”
Post-condition	halaman riwayat pemesanan yang tersedia telah ditampilkan

Tabel 4.13 Use Case Skenario Cetak Struk Membership

ID	KF_10
Nama Use-case	Cetak Struk Membership
Aktor	<i>MemberAccount</i>
Deskripsi	Use case menjelaskan bagaimana member account melakukan proses untuk cetak struk membership
Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama member account
Main Flow	<ol style="list-style-type: none"> 1. Memilih menu “Daftar Membership” dan pilih submenu “Lihat Riwayat Membership” 2. Sistem akan menampilkan halaman riwayat membership. 3. Member account memilih icon print pada salah satu kolom dengan status belum divalidasi.
Alternative Flows	<ol style="list-style-type: none"> 1. Jika pilih icon print dengan status selain ‘belum divalidasi’, maka sistem akan menampilkan notifikasi “cetak struk gagal, status sudah divalidasi”
Post-condition	struk membership telah ditampilkan

Tabel 4.14 Use Case Skenario Mendaftar Membership

ID	KF_11
Nama Use-case	Mendaftar Membership
Aktor	<i>Member Account</i>
Deskripsi	Use case ini menjelaskan bagaimana member account

	melakukan proses untuk mendaftar olahraga membership diantaranya fitness, yoga dan aerobik.
Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama member account
Main Flow	<ol style="list-style-type: none"> 1. Member account memilih menu “daftar membership” 2. Sistem menampilkan halaman olahraga yang tersedia 3. Member account memilih dan menekan salah satu olahraga 4. Sistem menampilkan halaman olahraga tersebut 5. Member account mengisi data membership dan menekan tombol “Daftar”. 6. Sistem menambah data membership dan kembali ke halaman utama member account
Alternative Flows	<ol style="list-style-type: none"> 1. Jika input tanggal tidak valid, maka sistem akan menampilkan notifikasi “Pendaftaran gagal disimpan, tanggal kadaluarsa”
Post-condition	Daftar membership telah berhasil ditambahkan

Tabel 4.15 Use Case Skenario Melihat Riwayat Membership

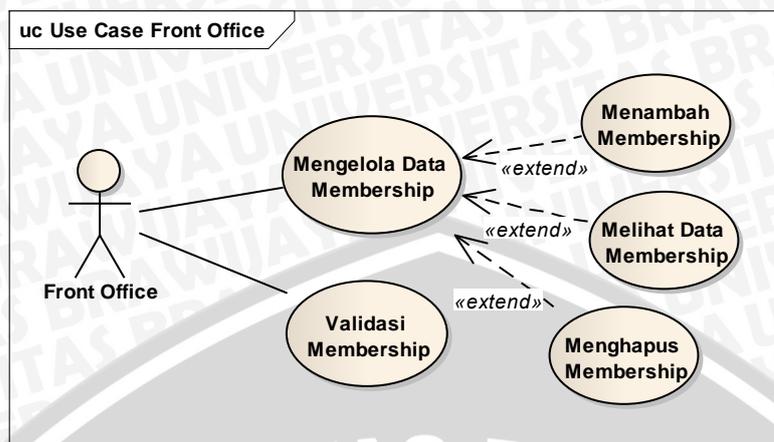
ID	KF_12
Nama Use-case	Melihat Riwayat Membership
Aktor	<i>Member Account</i>
Deskripsi	Use case ini menjelaskan bagaimana member account melakukan proses untuk melihat riwayat membership
Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama member account
Main Flow	<ol style="list-style-type: none"> 1. Member account memilih menu “Daftar Membership” dan pilih submenu “Lihat Riwayat Membership” 2. Sistem menampilkan riwayat membership.
Alternative Flows	<ol style="list-style-type: none"> 2. Jika riwayat membership tidak tersedia, maka sistem akan menampilkan notifikasi “riwayat membership tidak tersedia untuk sementara ini”
Post-condition	halaman riwayat membership yang tersedia telah ditampilkan

Tabel 4.16 Use Case Skenario Membuat Komplain

ID	KF_13
Nama Use-case	Membuat Komplain
Aktor	<i>Member Account</i>
Deskripsi	Use case menjelaskan bagaimana member account melakukan proses melakukan komplain.
Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama member account
Main Flow	<ol style="list-style-type: none"> 1. Member account memilih menu “Membuat Komplain” 2. Sistem menampilkan halaman list komplain. 3. Member account memilih tombol ‘Keluhan’ dan isi data keluhan 4. Pilih ‘send complain’
Alternative Flows	<ol style="list-style-type: none"> 1. Jika <i>user</i> tidak mengisi salah satu atau beberapa atau semua kolom input, maka sistem akan menampilkan notifikasi “please fill out this field”
Post-condition	Komplain dari member account terkirim ke halaman admin

Tabel 4.17 Use Case Skenario Melihat Tagihan

ID	KF_14
Nama Use-case	Melihat Tagihan
Aktor	<i>Member Account</i>
Deskripsi	Use case ini menjelaskan bagaimana member account melakukan proses untuk melihat tagihan
Pre-condition	<i>member</i> account telah melakukan login dan masuk ke halaman utama member account
Main Flow	<ol style="list-style-type: none"> 1. Member account memilih menu “Melihat Tagihan” 2. Sistem menampilkan tagihan pemesanan dan membership
Alternative Flows	<ol style="list-style-type: none"> 1. Jika riwayat membership tidak tersedia, maka sistem akan menampilkan notifikasi “riwayat membership tidak tersedia untuk sementara ini”
Post-condition	halaman riwayat membership yang tersedia telah ditampilkan



Gambar 4.7 Model Use Case Actor Front Office

Tabel 4.18 Use Case Skenario Menambah Membership

ID	KF_15
Nama Use-case	Menambah Membership
Aktor	<i>Front Office</i>
Deskripsi	Use case ini menjelaskan bagaimana front office melakukan proses untuk menambah membership
Pre-condition	<i>Front office</i> telah melakukan login dan masuk ke halaman utama <i>Front office</i>
Main Flow	<ol style="list-style-type: none"> 1. <i>Front Office</i> memilih menu “Data Membership” Pilih submenu “Add Membership” 2. Sistem menampilkan halaman olahraga yang tersedia dan pilih salah satu 3. Front office mengisi data membership dan menekan tombol “Daftar”. 4. Sistem menambah data membership dan kembali ke halaman utama front office
Alternative Flows	<ol style="list-style-type: none"> 1. Jika input tanggal tidak valid, maka sistem akan menampilkan notifikasi “Pendaftaran gagal disimpan, tanggal kadaluarsa”
Post-condition	Daftar membership telah berhasil ditambahkan

Tabel 4.19 Use Case Skenario Melihat data Membership

ID	KF_16
-----------	-------

Nama Use-case	Melihat data Membership
Aktor	<i>Front Office</i>
Deskripsi	Use case ini menjelaskan bagaimana front office melakukan proses untuk melihat data membership
Pre-condition	<i>Front office</i> telah melakukan login dan masuk ke halaman utama <i>Front office</i>
Main Flow	<ol style="list-style-type: none"> 1. <i>Front Office</i> memilih menu “Data Membership” Pilih submenu “List Membership” 2. Sistem menampilkan halaman list membership.
Alternative Flows	<ol style="list-style-type: none"> 1. Jika list membership tidak tersedia, maka sistem akan menampilkan notifikasi “list membership tidak tersedia untuk sementara ini”
Post-condition	halaman list membership yang tersedia telah ditampilkan

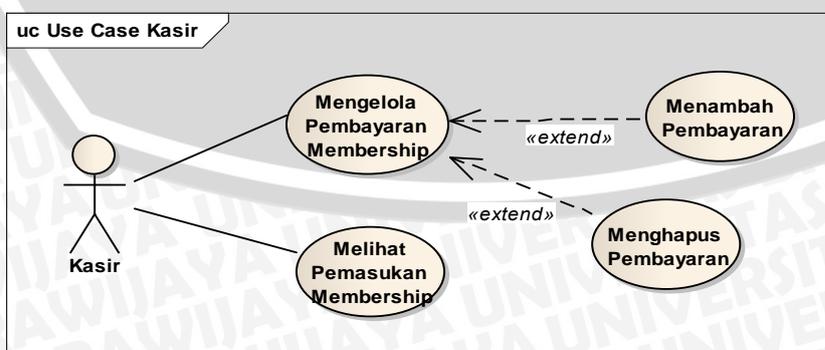
Tabel 4.20 Use Case Skenario Menghapus Membership

ID	KF_17
Nama Use-case	Menghapus Membership
Aktor	<i>Front Office</i>
Deskripsi	Use case ini menjelaskan bagaimana front office melakukan proses untuk menghapus membership
Pre-condition	<i>Front office</i> telah melakukan login dan masuk ke halaman utama <i>Front office</i>
Main Flow	<ol style="list-style-type: none"> 1. <i>Front Office</i> memilih menu “Data Membership” Pilih submenu “List Membership” 2. Sistem menampilkan halaman list membership. 3. Front office memilih membership yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik OK 5. Sistem menghapus data
Alternative Flows	<ol style="list-style-type: none"> 1. <i>Front Office</i> memilih menu “Data Membership” Pilih submenu “List Membership” 2. Sistem menampilkan halaman list membership. 3. Front office memilih membership yang akan dihapus. Klik icon hapus

	<ol style="list-style-type: none"> 4. Sistem menampilkan peringatan, dan klik cancel 5. Sistem tidak melakukan aksi apapun
Post-condition	Sistem berhasil menghapus data membership dari database.

Tabel 4.21 Use Case Skenario Validasi Data Membership

ID	KF_18
Nama Use-case	Validasi data Membership
Aktor	<i>Front Office</i>
Deskripsi	Use case ini menjelaskan bagaimana front office melakukan proses untuk memvalidasi data membership
Pre-condition	<i>Front office</i> telah melakukan login dan masuk ke halaman utama <i>Front office</i>
Main Flow	<ol style="list-style-type: none"> 1. <i>Front Office</i> memilih menu “Validasi Membership” 2. Sistem menampilkan halaman daftar membership 3. Front office memilih salah satu membership dengan tombol ‘validasi’ 4. Sistem menampilkan peringatan dan klik OK 5. Maka status membership menjadi “tervalidasi”
Alternative Flows	<ol style="list-style-type: none"> 1. <i>Front Office</i> memilih menu “Validasi Membership” 2. Sistem menampilkan halaman daftar membership 3. Front office memilih salah satu membership dengan tombol ‘validasi’ 4. Sistem menampilkan peringatan dan klik Cancel 5. Maka status membership tetap “belum divalidasi”
Post-condition	Sistem berhasil mengubah status membership



Gambar 4.8 Model Use Case Actor Kasir

Tabel 4.22 Use Case Skenario Menambah Pembayaran

ID	KF_19
Nama Use-case	Menambah Pembayaran
Aktor	Kasir
Deskripsi	Use case ini menjelaskan bagaimana kasir melakukan proses untuk menambah pembayaran
Pre-condition	Kasir telah melakukan login dan masuk ke halaman utama Kasir
Main Flow	<ol style="list-style-type: none"> 1. Kasir memilih menu “Pembayaran Membership” Pilih submenu “Add Pembayaran” 2. Sistem menampilkan halaman form add Pembayaran 3. Kasir mengisi data no membership, lalu pilih tombol “submit” 4. Sistem menampilkan data tersebut 5. Lalu pilih tombol ‘bayar’
Alternative Flows	<ol style="list-style-type: none"> 1. Jika data membership tidak valid ketika memilih submit maka sistem akan menampilkan notifikasi “no membership tidak ditemukan”
Post-condition	Sistem menambahkan data pembayaran membership

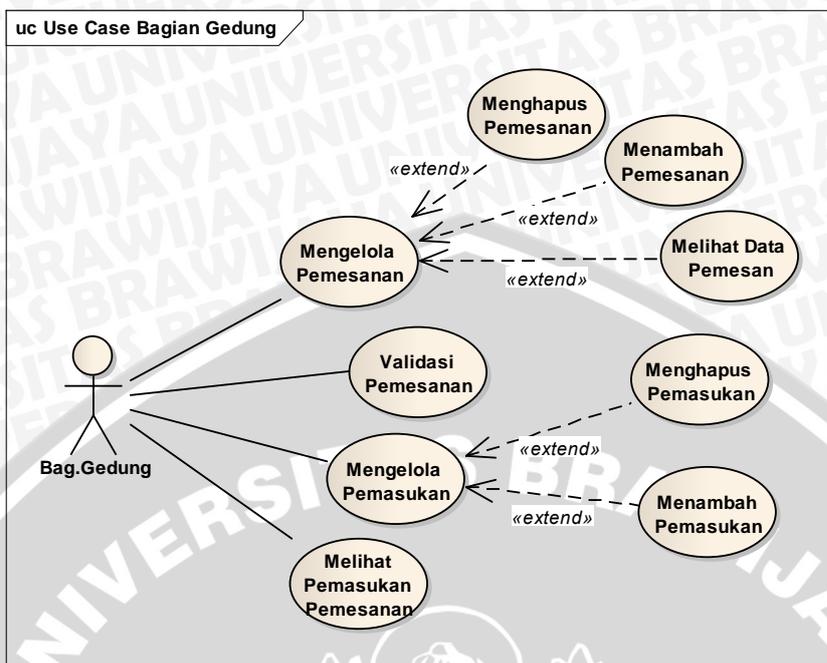
Tabel 4.23 Use Case Skenario Menghapus Membership

ID	KF_20
Nama Use-case	Menghapus Pembayaran
Aktor	Kasir
Deskripsi	Use case ini menjelaskan bagaimana kasir melakukan proses untuk menghapus pembayaran
Pre-condition	Kasir telah melakukan login dan masuk ke halaman utama Kasir
Main Flow	<ol style="list-style-type: none"> 1. Kasir memilih menu “Pembayaran Membership” Pilih submenu “List Pembayaran” 2. Sistem menampilkan halaman list pembayaran. 3. Kasir memilih pembayaran yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik OK

	5. Sistem menghapus data
Alternative Flows	<ol style="list-style-type: none"> 1. Kasir memilih menu “Pembayaran Membership” Pilih submenu “List Pembayaran” 2. Sistem menampilkan halaman list pembayaran. 3. Kasir memilih pembayaran yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik cancel 5. Sistem tidak melakukan aksi apapun
Post-condition	Sistem menghapus data pembayaran dari database.

Tabel 4.24 Use Case Skenario Melihat Pemasukan Membership

ID	KF_21
Nama Use-case	Melihat Pemasukan membership
Aktor	Kasir dan Bagian Keuangan
Deskripsi	Use case ini menjelaskan bagaimana kasir melakukan proses untuk Melihat Pemasukan membership
Pre-condition	Kasir telah melakukan login dan masuk ke halaman utama Kasir
Main Flow	<ol style="list-style-type: none"> 1. Kasir memilih menu “Laporan Pemasukan” 2. Sistem menampilkan halaman pemasukan membership 3. Kasir memilih cek berdasarkan tanggal 4. Sistem menampilkan halaman pemasukan sesuai tanggal
Alternative Flows	<ol style="list-style-type: none"> 1. Jika pemasukan tidak tersedia, maka sistem akan menampilkan notifikasi “pemasukan tidak tersedia untuk sementara ini”
Post-condition	Sistem menampilkan laporan pemasukan dari database.



Gambar 4.9 Model Use Case Actor Bagian Gedung

Tabel 4.25 Use Case Skenario Validasi Pemesanan

ID	KF_22
Nama Use-case	Validasi Pemesanan
Aktor	Bagian Gedung
Deskripsi	Use case menjelaskan bagaimana bagian gedung melakukan proses untuk memvalidasi pemesanan
Pre-condition	Bagian gedung telah melakukan login dan masuk ke halaman utama bagian gedung
Main Flow	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “Validasi Pemesanan” 2. Sistem menampilkan halaman daftar pesan 3. Bagian gedung memilih salah satu pesan dengan tombol ‘validasi’ 4. Sistem menampilkan peringatan dan klik OK 5. Maka status pesan menjadi “sudah divalidasi”
Alternative Flows	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “Validasi Pemesanan” 2. Sistem menampilkan halaman daftar pesan 3. Bagian gedung memilih salah satu pesan dengan tombol ‘validasi’ 4. Sistem menampilkan peringatan dan klik Cancel 5. Maka status pesan tetap “belum divalidasi”

Post-condition	Sistem mengubah status membership ke database.
-----------------------	--

Tabel 4.26 Use Case Skenario Menambah Pemesanan

ID	KF_23
Nama Use-case	Menambah Pemesanan
Aktor	Bagian Gedung
Deskripsi	Use case ini menjelaskan bagaimana Bagian gedung melakukan proses untuk menambah pemesanana
Pre-condition	Bagian gedung telah melakukan login dan masuk ke halaman utama bagian gedung
Main Flow	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “pemesanan” Pilih submenu “Add Pemesanan” 2. Sistem akan menampilkan halaman olahraga. 3. Bagian gedung memilih salah satu olahraga 4. Sistem menampilkan form pemesanan 5. Bagian gedung mengisi data pemesanan, lalu pilih tombol “daftar” 6. Sistem menambahkan data pemesanan dan menampilkan halaman utama bagian gedung
Alternative Flows	<ol style="list-style-type: none"> 1. Jika input tanggal tidak valid, maka sistem akan menampilkan notifikasi “Pendaftaran gagal disimpan, tanggal kadaluarsa” 2. Jika input jam tidak valid, maka sistem akan menampilkan notifikasi “Pendaftaran gagal disimpan, jam akhir kurang dari jam awal”
Post-condition	Pemesanan tempat telah berhasil ditambahkan

Tabel 4.27 Use Case Skenario Menghapus Pemesanan

ID	KF_24
Nama Use-case	Menghapus Pemesanan
Aktor	Bagian Gedung
Deskripsi	Use case ini menjelaskan bagaimana Bagian gedung melakukan proses untuk menghapus pemesanan
Pre-condition	Bagian gedung telah melakukan login dan masuk ke halaman

	utama bagian gedung
Main Flow	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “pemesanan” Pilih submenu “List Pemesan” 2. Sistem menampilkan halaman list pemesanan. 3. Bagian gedung memilih list yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik OK 5. Sistem menghapus data
Alternative Flows	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “pemesanan” Pilih submenu “List Pemesan” 2. Sistem menampilkan halaman list pemesanan. 3. Bagian gedung memilih list yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik Cancel 5. Sistem tidak melakukan aksi apapun
Post-condition	Sistem menghapus data pemesanan dari database.

Tabel 4.28 Use Case Skenario Melihat Data Pemesan

ID	KF_25
Nama Use-case	Melihat data Membership
Aktor	Bagian Gedung
Deskripsi	Use case ini menjelaskan bagaimana bagian gedung melakukan proses untuk melihat data pemesan
Pre-condition	Bagian gedung telah melakukan login dan masuk ke halaman utama bagian gedung
Main Flow	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “Pemesanan”, pilih submenu “List Pemesan” 2. Sistem menampilkan halaman list pemesan.
Alternative Flows	<ol style="list-style-type: none"> 1. Jika list pemesan tidak tersedia, maka sistem akan menampilkan notifikasi “list pemesanan tidak tersedia untuk sementara ini”
Post-condition	halaman list pemesanan yang tersedia telah ditampilkan

Tabel 4.29 Use Case Skenario Menambah Pemasukan

ID	KF_26
-----------	-------

Nama Use-case	Menambah Pemasukan
Aktor	Bagian gedung
Deskripsi	Use case ini menjelaskan bagaimana Bagian gedung melakukan proses untuk menambah pemasukan
Pre-condition	Bagian gedung telah melakukan login dan masuk ke halaman utama bagian gedung
Main Flow	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “pemasukan” pilih submenu “Add pemasukan” 2. Sistem akan menampilkan halaman form add Pembayaran 3. Bagian gedung mengisi data no pemesan, lalu pilih tombol “submit” 4. Sistem menampilkan data tersebut 5. Lalu pilih tombol ‘bayar’
Alternative Flows	<ol style="list-style-type: none"> 1. Jika data pemesan tidak valid ketika memilih submit maka sistem akan menampilkan notifikasi “no pemesan tidak ditemukan”
Post-condition	Sistem menambahkan data pemasukan pemesanan

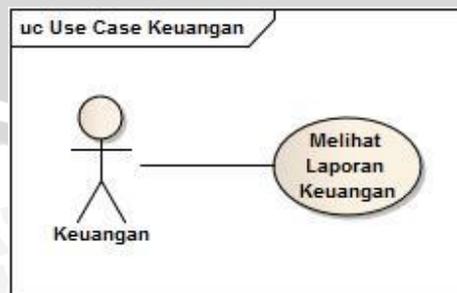
Tabel 4.30 Use Case Skenario Menghapus Pemasukan

ID	KF_27
Nama Use-case	Menghapus Pemasukan
Aktor	Bagian gedung
Deskripsi	Use case ini menjelaskan bagaimana Bagian gedung melakukan proses untuk menghapus pemasukan
Pre-condition	Bagian gedung telah melakukan login dan masuk ke halaman utama bagian gedung
Main Flow	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “pemasukan” pilih submenu “List pemasukan” 2. Sistem menampilkan halaman list 3. Bagian gedung memilih data yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik OK 5. Sistem menghapus data

Alternative Flows	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “pemasukan” pilih submenu “List pemasukan” 2. Sistem menampilkan halaman list 3. Bagian gedung memilih data yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik Cancel 5. Sistem tidak melakukan aksi apapun
Post-condition	Sistem menghapus data dari database.

Tabel 4.31 Use Case Skenario Melihat Pemasukan Pemesanan

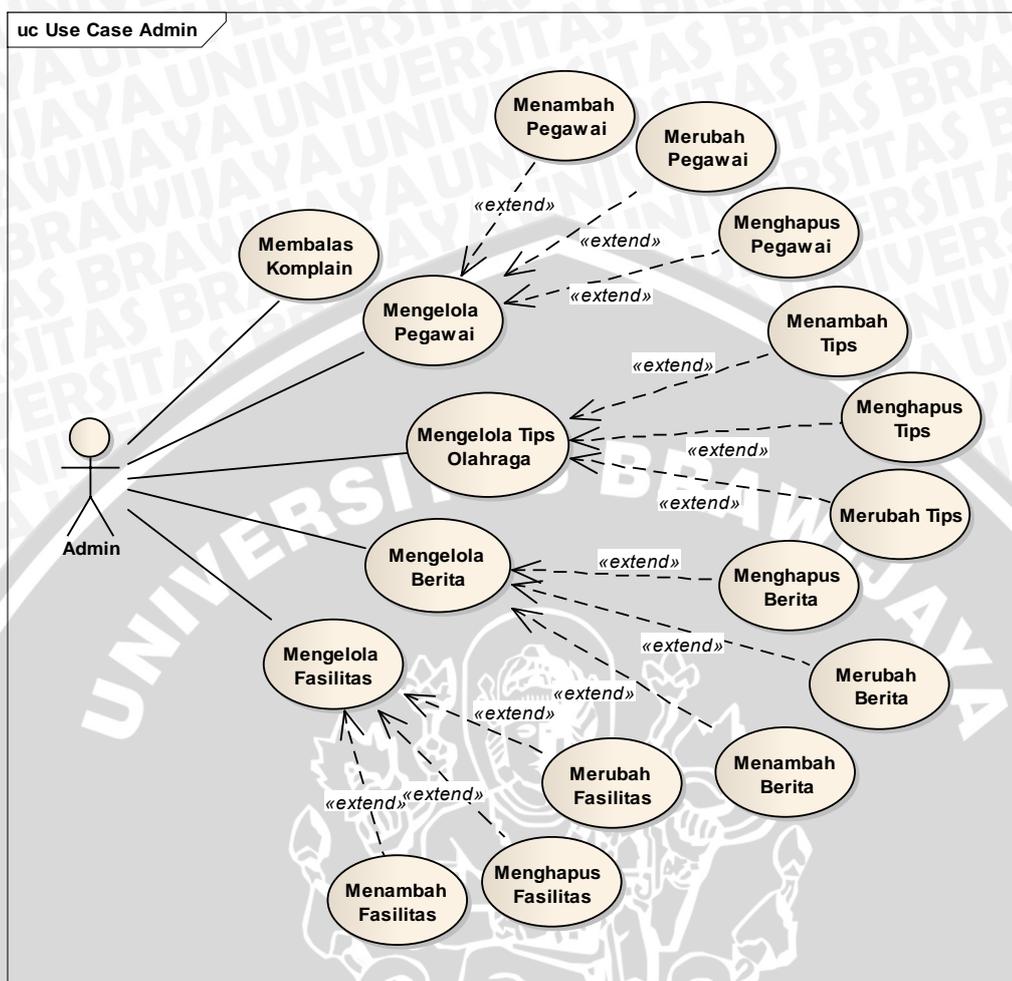
ID	KF_28
Nama Use-case	Melihat Pemasukan pemesanan
Aktor	Bagian Gedung dan Keuangan
Deskripsi	Use case ini menjelaskan bagaimana bagian gedung melakukan proses untuk Melihat Pemasukan pemesanan
Pre-condition	Bagian gedung telah melakukan login dan masuk ke halaman utama bagian gedung
Main Flow	<ol style="list-style-type: none"> 1. Bagian gedung memilih menu “Laporan Pemesanan” 2. Sistem menampilkan halaman pemasukan pemesanan 3. Cek berdasarkan tanggal yang diinginkan 4. Sistem menampilkan pemasukan sesuai tanggal yang dipilih
Alternative Flows	<ol style="list-style-type: none"> 1. Jika pemasukan tidak tersedia, maka sistem akan menampilkan notifikasi “pemasukan tidak tersedia untuk sementara ini”
Post-condition	Sistem menampilkan pemasukan pemesanan dari database.



Gambar 4.10 Model Use Case Actor Keuangan

Tabel 4.32 Use Case Skenario Melihat laporan Keuangan

ID	KF_29
Nama Use-case	Melihat Laporan Keuangan
Aktor	Bagian Keuangan dan bendahara
Deskripsi	Use case ini menjelaskan bagaimana keuangan dan bagian bendahara untuk melihat laporan keuangan
Pre-condition	keuangan dan bendahara telah melakukan login dan masuk ke halaman utama masing-masing
Main Flow	<ol style="list-style-type: none">1. Keuangan memilih menu “Laporan keuangan”2. Sistem menampilkan halaman laporan keuangan3. Cek berdasarkan tanggal yang diinginkan4. Sistem menampilkan laporan keuangan sesuai tanggal yang dipilih5. Bendahara memilih menu “Laporan keuangan”6. Sistem menampilkan halaman laporan keuangan7. Cek berdasarkan tanggal yang diinginkan8. Sistem menampilkan laporan keuangan sesuai tanggal yang dipilih
Alternative Flows	<ol style="list-style-type: none">1. Jika laporan tidak tersedia, maka sistem akan menampilkan notifikasi “laporan tidak tersedia untuk sementara ini”
Post-condition	Sistem menampilkan laporan keuangan



Gambar 4.11 Model Use Case Actor Admin

Tabel 4.33 Use Case Skenario Membalas Komplain

ID	KF_30
Nama Use-case	Membalas Komplain
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana admin melakukan proses untuk membalas komplain.
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Komplain” 2. Sistem menampilkan halaman list komplain. 3. Admin memilih komplain yang akan dibalas. Isi field balas dan tekan tombol “send complain” 4. Sistem menampilkan pesan complain terkirim

Alternative Flows	1. Jika admin tidak mengisi salah satu atau semua kolom input, maka sistem akan menampilkan notifikasi “please fill out this field”
Post-condition	Membalas complain telah berhasil ditambah

Tabel 4.34 Use Case Skenario Menambah Berita

ID	KF_31
Nama Use-case	Menambah Berita
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk menambah berita
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Berita” memilih submenu “Tulis Berita” 2. Sistem menampilkan halaman form tulis berita 3. Admin mengisi data berita, lalu pilih tombol “submit” 4. Sistem berhasil menambah berita dan kembali ke halaman utama admin
Alternative Flows	Jika admin tidak mengisi salah satu atau semua kolom input, maka sistem akan menampilkan notifikasi “please fill out this field”
Post-condition	Sistem menambahkan data berita ke database

Tabel 4.35 Use Case Skenario Menghapus Berita

ID	KF_32
Nama Use-case	Menghapus Berita
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk menghapus berita
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin

Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Berita” memilih submenu “List Berita” 2. Sistem menampilkan halaman list berita 3. Admin memilih berita yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik OK 5. Sistem menghapus data
Alternative Flows	<ol style="list-style-type: none"> 1. Admin memilih menu “Berita” memilih submenu “List Berita” 2. Sistem menampilkan halaman list berita 3. Admin memilih berita yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik Cancel 5. Sistem tidak melakukan aksi apapun
Post-condition	Sistem menghapus data berita dari database.

Tabel 4.36 Use Case Skenario Mengubah Berita

ID	KF_33
Nama Use-case	Mengubah Berita
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana admin melakukan proses untuk mengubah berita
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Berita” memilih submenu “List Berita” 2. Sistem menampilkan halaman daftar berita. 3. Admin memilih berita yang akan di edit. Klik icon edit 4. Sistem menampilkan form edit berita tersebut. 5. Admin mengisi update data dan tekan tombol “submit” 6. Sistem menyimpan perubahan
Alternative Flows	<ol style="list-style-type: none"> 1. Admin memilih menu “Berita” memilih submenu “List Berita” 2. Sistem menampilkan halaman daftar berita.

	<ol style="list-style-type: none"> 3. Admin memilih berita yang akan di edit. Klik icon edit 4. Sistem menampilkan form edit berita tersebut. 5. Admin mengisi update data dan tidak menekan tombol “submit” 6. Sistem tidak menyimpan perubahan
Post-condition	Sistem mengubah data berita ke database.

Tabel 4.37 Use Case Skenario Menambah Pegawai

ID	KF_34
Nama Use-case	Menambah Pegawai
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk menambah pegawai
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Pegawai” memilih submenu “Tambah pegawai” 2. Sistem menampilkan halaman form tambah pegawai 3. Admin mengisi data pegawai, lalu pilih tombol “submit” 4. Sistem berhasil menambah pegawai dan kembali ke halaman utama admin
Alternative Flows	<ol style="list-style-type: none"> 1. Jika admin tidak mengisi salah satu atau semua kolom input, maka sistem akan menampilkan notifikasi “please fill out this field”
Post-condition	Sistem berhasil menambahkan data pegawai ke database

Tabel 4.38 Use Case Skenario Menghapus Pegawai

ID	KF_35
Nama Use-case	Menghapus Pegawai
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk menghapus pegawai

Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Pegawai” memilih submenu “List Pegawai” 2. Sistem menampilkan halaman list pegawai 3. Admin memilih pegawai yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik OK 5. Sistem menghapus data
Alternative Flows	<ol style="list-style-type: none"> 1. Admin memilih menu “Pegawai” memilih submenu “List Pegawai” 2. Sistem menampilkan halaman list pegawai 3. Admin memilih pegawai yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik Cancel 5. Sistem tidak melakukan aksi apapun
Post-condition	Sistem berhasil menghapus data pegawai dari database.

Tabel 4.39 Use Case Skenario Mengubah Pegawai

ID	KF_36
Nama Use-case	Mengubah Pegawai
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk mengubah pegawai
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Pegawai” memilih submenu “List Pegawai” 2. Sistem menampilkan halaman daftar pegawai. 3. Admin memilih pegawai yang akan di edit. Klik icon edit 4. Sistem menampilkan form edit pegawai tersebut. 5. Admin mengisi update data dan tekan tombol “submit” 6. Sistem menyimpan perubahan
Alternative	<ol style="list-style-type: none"> 1. Admin memilih menu “Pegawai” memilih submenu

Flows	<p>“List Pegawai”</p> <ol style="list-style-type: none"> 2. Sistem menampilkan halaman daftar pegawai. 3. Admin memilih pegawai yang akan di edit. Klik icon edit 4. Sistem menampilkan form edit pegawai tersebut. 5. Admin mengisi update data dan tidak menekan tombol “submit” 6. Sistem tidak menyimpan perubahan
Post-condition	Sistem berhasil mengubah data pegawai ke database.

Tabel 4.40 Use Case Skenario Menambah Fasilitas

ID	KF_37
Nama Use-case	Menambah Fasilitas
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk menambah fasilitas
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Fasilitas” memilih submenu “Tulis Fasilitas” 2. Sistem menampilkan halaman form tulis fasilitas 3. Admin mengisi data faslitas, lalu pilih tombol “submit” 4. Sistem berhasil menambah fasilitas dan kembali ke halaman utama admin
Alternative Flows	<ol style="list-style-type: none"> 1. Jika admin tidak mengisi salah satu atau semua kolom input, maka sistem akan menampilkan notifikasi “please fill out this field”
Post-condition	Sistem berhasil menambahkan data fasilitas ke database

Tabel 4.41 Use Case Skenario Menghapus Fasilitas

ID	KF_38
Nama Use-case	Menghapus Fasilitas
Aktor	Admin

Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk menghapus fasilitas
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Fasilitas” memilih submenu “List Fasilitas” 2. Sistem menampilkan halaman list fasilitas 3. Admin memilih fasilitas yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik OK 5. Sistem menghapus data
Alternative Flows	<ol style="list-style-type: none"> 1. Admin memilih menu “Fasilitas” memilih submenu “List Fasilitas” 2. Sistem menampilkan halaman list fasilitas 3. Admin memilih fasilitas yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik Cancel 5. Sistem tidak melakukan aksi apapun
Post-condition	Sistem berhasil menghapus data fasilitas dari database.

Tabel 4.42 Use Case Skenario Mengubah Fasilitas

ID	KF_39
Nama Use-case	Mengubah Fasilitas
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk mengubah fasilitas
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Fasilitas” memilih submenu “List Fasilitas” 2. Sistem menampilkan halaman daftar fasilitas. 3. Admin memilih fasilitas yang akan di edit. Klik icon edit 4. Sistem menampilkan form edit fasilitas tersebut. 5. Admin mengisi update data dan tekan tombol “submit”

	6. Sistem menyimpan perubahan
Alternative Flows	<ol style="list-style-type: none"> 1. Admin memilih menu “Fasilitas” memilih submenu “List Fasilitas” 2. Sistem menampilkan halaman daftar fasilitas. 3. Admin memilih fasilitas yang akan di edit. Klik icon edit 4. Sistem menampilkan form edit fasilitas tersebut. 5. Admin mengisi update data dan tidak menekan tombol “submit” 6. Sistem tidak menyimpan perubahan
Post-condition	Sistem berhasil mengubah data fasilitas ke database.

Tabel 4.43 Use Case Skenario Menambah Tips

ID	KF_40
Nama Use-case	Menambah Tips
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk menambah tips
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Tips Olahraga” memilih submenu “Tulis Tips” 2. Sistem menampilkan halaman form tulis tips 3. Admin mengisi data tips, lalu pilih tombol “submit” 4. Sistem berhasil menambah tips olahraga dan kembali ke halaman utama admin
Alternative Flows	<ol style="list-style-type: none"> 1. Jika admin tidak mengisi salah satu atau semua kolom input, maka sistem akan menampilkan notifikasi “please fill out this field”
Post-condition	Sistem berhasil menambahkan data tips ke database

Tabel 4.44 Use Case Skenario Menghapus Tips

ID	KF_41
Nama Use-case	Menghapus Tips

Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk menghapus tips
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Tips Olahraga” memilih submenu “List Tips” 2. Sistem menampilkan halaman list tips 3. Admin memilih tips olahraga yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik OK 5. Sistem menghapus data
Alternative Flows	<ol style="list-style-type: none"> 1. Admin memilih menu “Tips Olahraga” memilih submenu “List Tips” 2. Sistem menampilkan halaman list tips 3. Admin memilih tips olahraga yang akan dihapus. Klik icon hapus 4. Sistem menampilkan peringatan, dan klik Cancel 5. Sistem tidak melakukan aksi apapun
Post-condition	Sistem berhasil menghapus data Tips dari database.

Tabel 4.45 Use Case Skenario Mengubah Tips

ID	KF_42
Nama Use-case	Mengubah Tips
Aktor	Admin
Deskripsi	Use case ini menjelaskan bagaimana Admin melakukan proses untuk mengubah tips
Pre-condition	Admin telah melakukan login dan masuk ke halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu “Tips Olahraga” memilih submenu “List Tips” 2. Sistem menampilkan halaman daftar tips. 3. Admin memilih tips yang akan di edit. Klik icon edit 4. Sistem menampilkan form edit tips tersebut. 5. Admin mengisi update data dan tekan tombol

	<p>“submit”</p> <p>6. Sistem menyimpan perubahan</p>
Alternative Flows	<ol style="list-style-type: none"> 1. Admin memilih menu “Tips Olahraga” memilih submenu “List Tips” 2. Sistem menampilkan halaman daftar tips. 3. Admin memilih tips yang akan di edit. Klik icon edit 4. Sistem menampilkan form edit tips tersebut. 5. Admin mengisi update data dan tidak menekan tombol “submit” 6. Sistem tidak menyimpan perubahan
Post-condition	Sistem berhasil mengubah data tips ke database.

4.2 Fase Perancangan

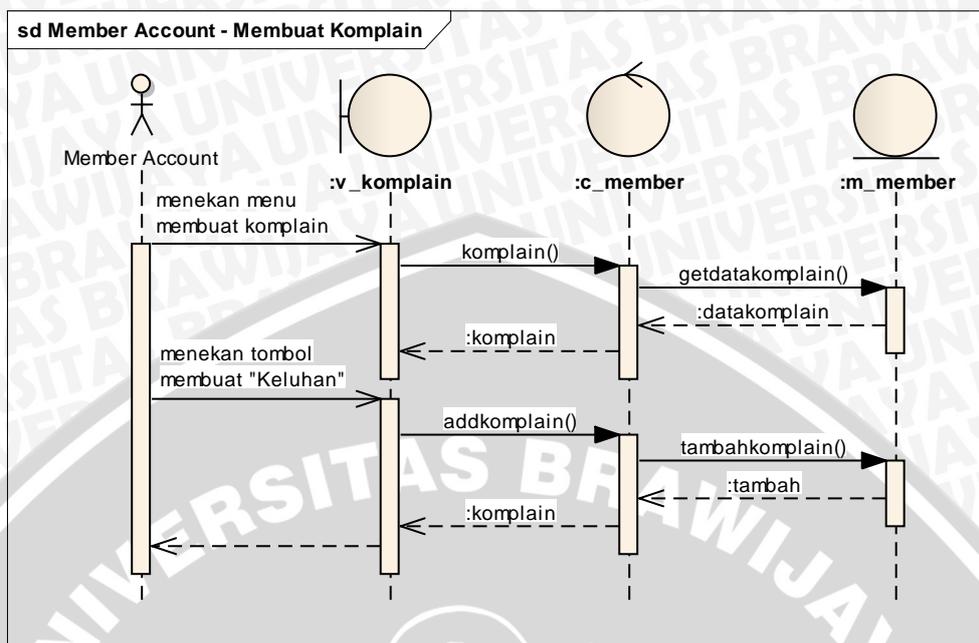
Pada tahapan ini perancangan pada penelitian ini menggunakan pendekatan desain berorientasi objek yang direpresentasikan menggunakan UML (*Unified Modelling Language*).

4.2.1 Perancangan Sequence Diagram

Sequence diagram menggambarkan interaksi antar masing-masing objek pada setiap usecase dalam urutan waktu. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu.

a. Sequence Diagram Membuat Komplain

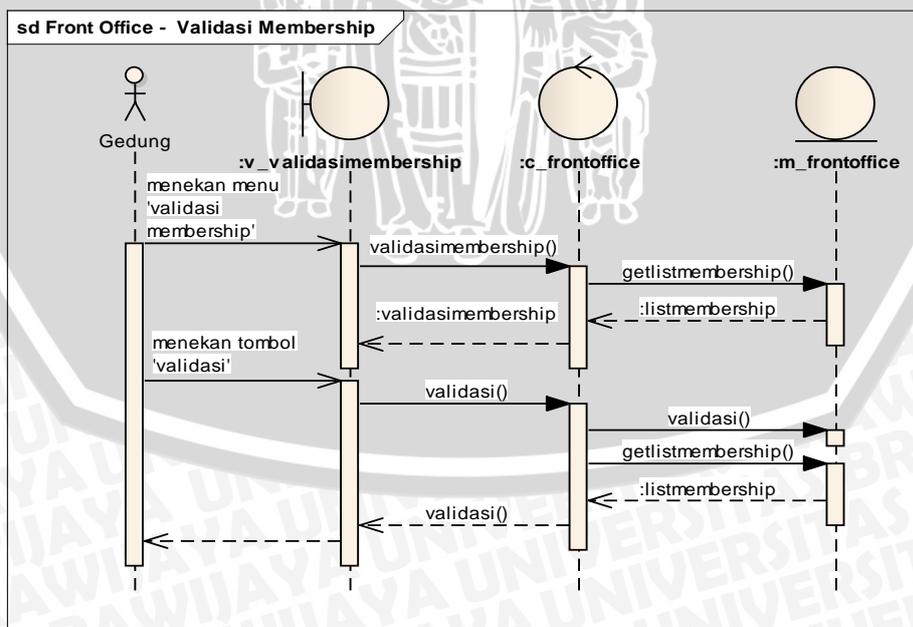
Sequence diagram membuat komplain adalah event ketika *member account* ingin melakukan complain kepada pihak Sport Club House Pertamina - UB. Perancangan *sequence diagram* membuat komplain dapat dilihat pada Gambar 4.12.



Gambar 4.12 sequence diagram membuat komplain

b. Sequence Diagram Validasi Membership

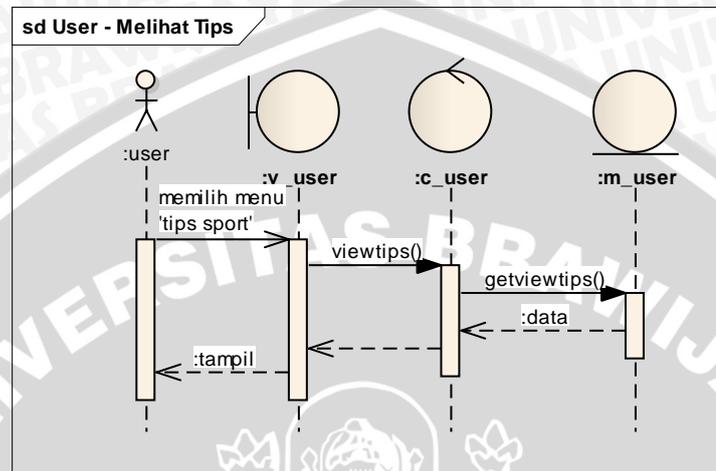
Sequence diagram validasi membership adalah event ketika front office ingin melakukan validasi membership dari member account . Perancangan sequence diagram validasi membership dapat dilihat pada Gambar 4.13.



Gambar 4.13 sequence diagram validasi membership

c. Sequence Diagram Melihat tips

Sequence diagram melihat tips adalah event ketika user ingin melihat tips olahraga melalui sistem informasi. Perancangan sequence diagram melihat tips dapat dilihat pada Gambar 4.14



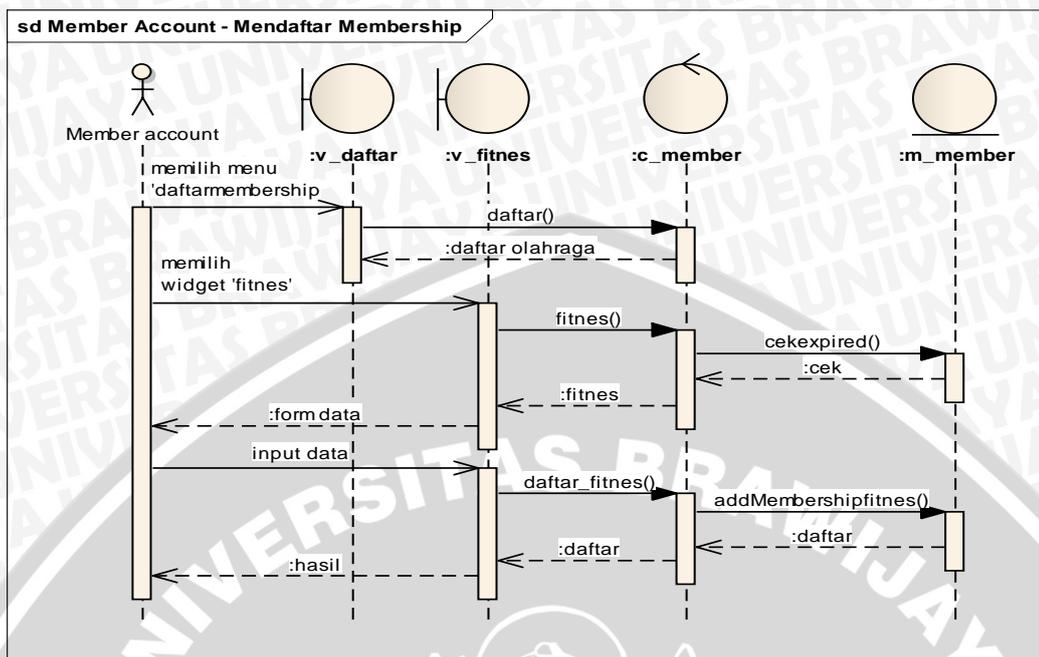
Gambar 4.14 sequence diagram melihat tips

d. Sequence Diagram Mendaftar Membership

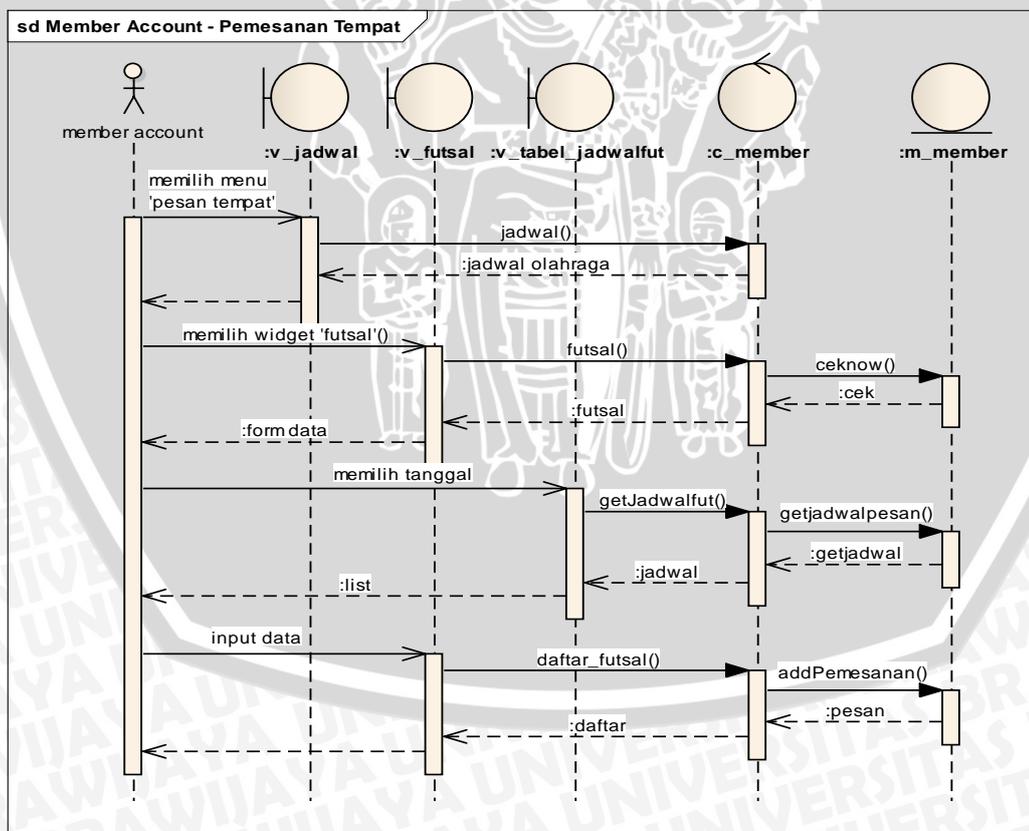
Sequence diagram mendaftar membership adalah event ketika member account ingin mendaftar membership olahraga melalui sistem informasi. Perancangan sequence diagram mendaftar membership dapat dilihat pada Gambar 4.15.

e. Sequence Diagram Pemesanan Tempat

Sequence diagram pemesanan tempat adalah event ketika member account ingin pemesanan olahraga. Perancangan sequence diagram pemesanan tempat dapat dilihat pada Gambar 4.16.



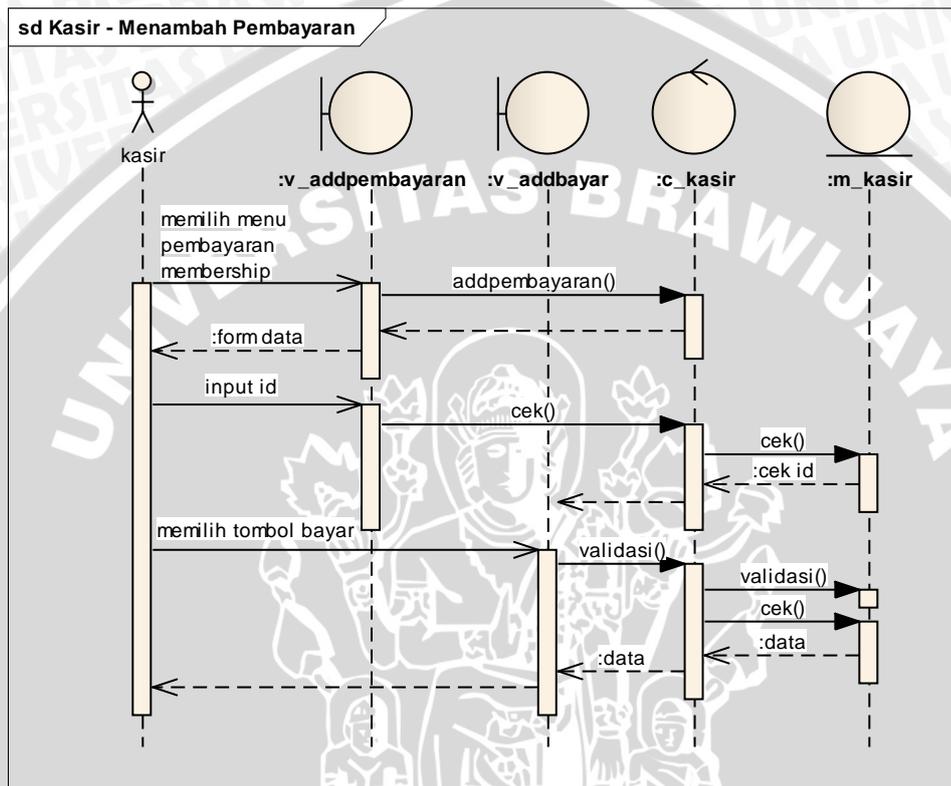
Gambar 4.15 sequence diagram mendaftar membership



Gambar 4.16 sequence diagram pemesanan tempat

f. Sequence Diagram Menambah Pembayaran

Sequence diagram menambah pembayaran adalah event ketika kasir ingin menambah pembayaran olahraga membership. Perancangan sequence diagram menambah pembayaran dapat dilihat pada Gambar 4.17.



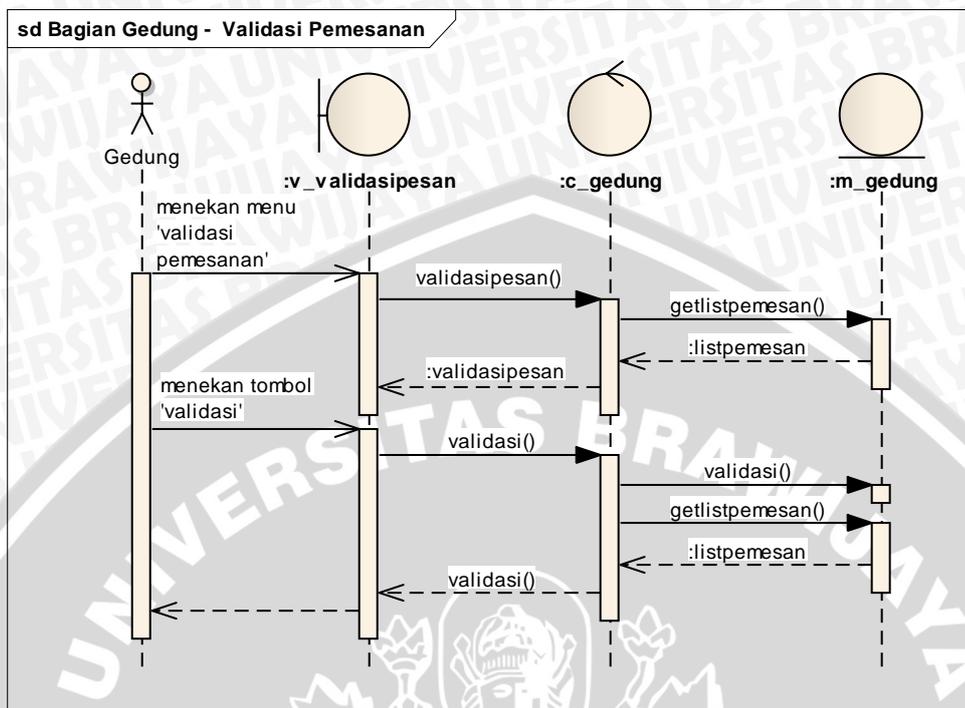
Gambar 4.17 sequence diagram menambah pembayaran

g. Sequence Diagram Validasi Pemesanan

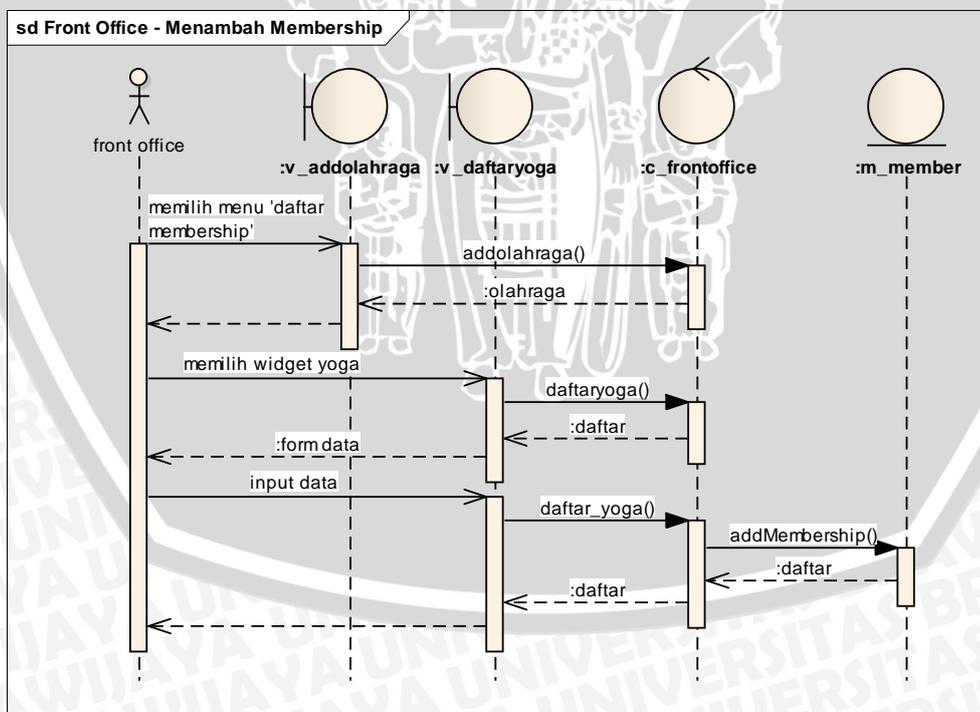
Sequence diagram validasi pemesanan adalah event ketika bagian gedung ingin memvalidasi pemesanan olahraga pemesan. Perancangan sequence diagram validasi pemesanan dapat dilihat pada Gambar 4.18.

h. Sequence Diagram Menambah Membership

Sequence diagram menambah membership adalah event ketika bagian front office ingin menambah membership olahraga melalui sistem informasi. Perancangan sequence diagram menambah membership dapat dilihat pada Gambar 4.19.



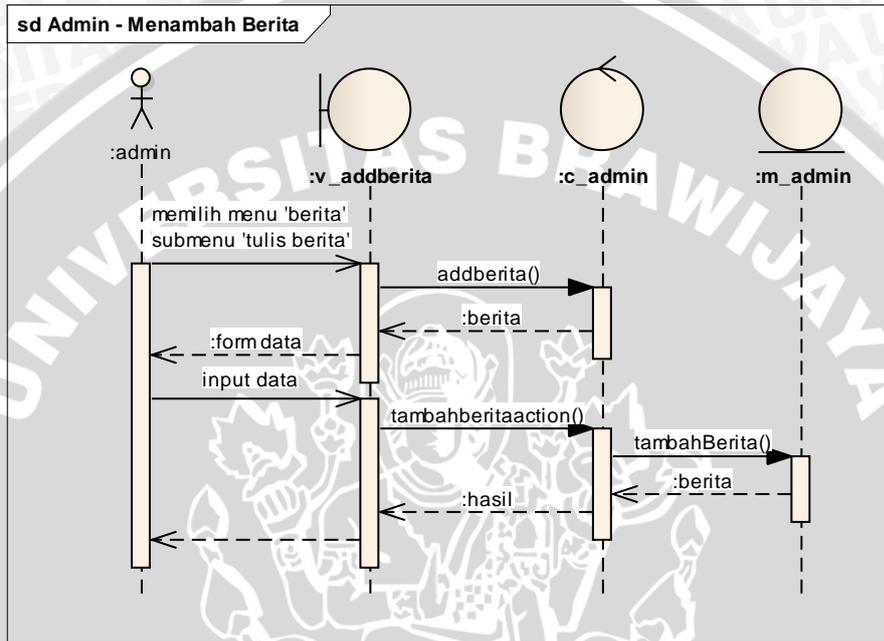
Gambar 4.18 sequence diagram validasi pemesanan



Gambar 4.19 sequence diagram menambah membership

i. Sequence Diagram Menambah Berita

Sequence diagram menambah berita adalah event ketika bagian admin ingin menambah berita seputar informasi UB Sport Club House Perancangan sequence diagram menambah berita dapat dilihat pada Gambar 4.20.



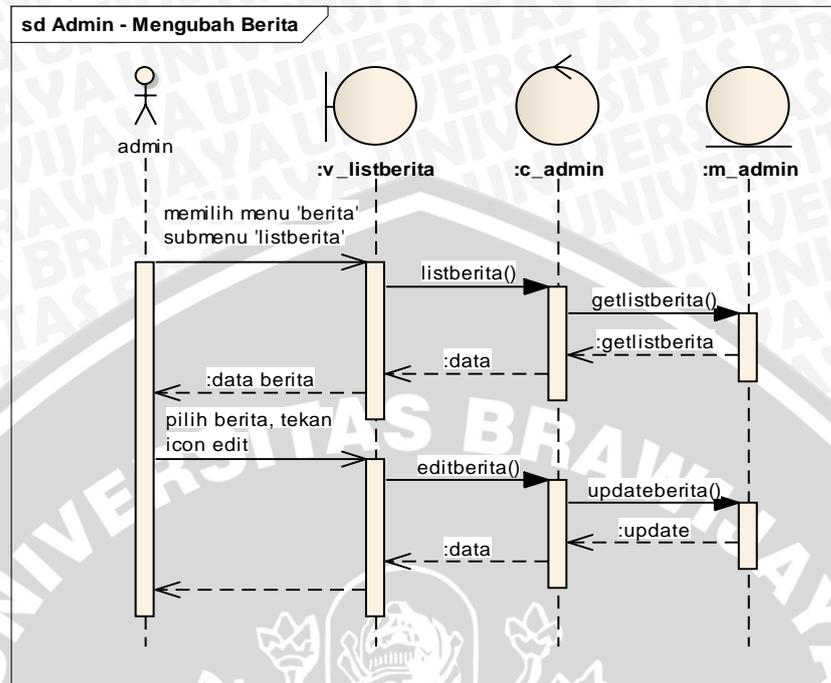
Gambar 4.20 sequence diagram menambah berita

j. Sequence Diagram Mengubah Berita

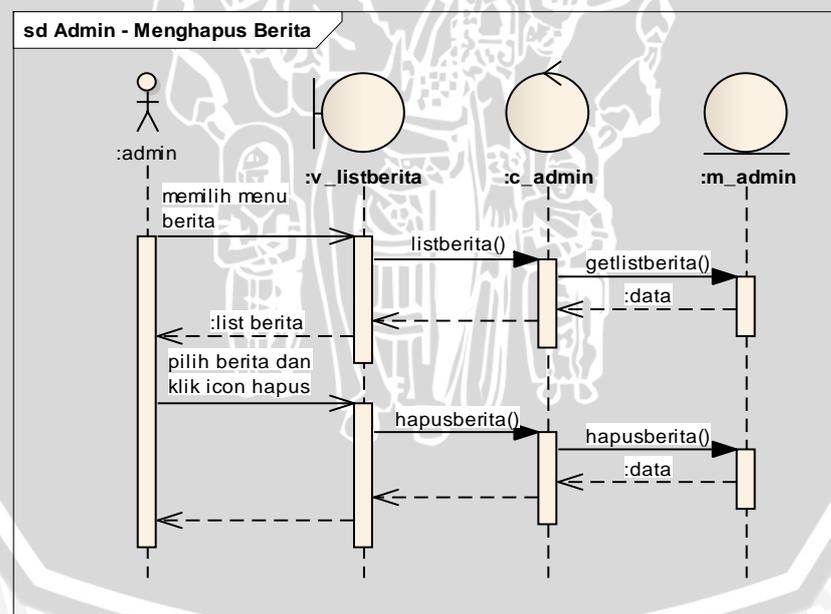
Sequence diagram mengubah berita adalah event ketika bagian admin ingin mengubah berita seputar informasi UB Sport Club House Perancangan sequence diagram mengubah berita dapat dilihat pada Gambar 4.21.

k. Sequence Diagram Menghapus Berita

Sequence diagram menghapus berita adalah event ketika bagian admin ingin menghapus berita seputar informasi UB Sport Club House Perancangan sequence diagram menghapus berita dapat dilihat pada Gambar 4.22.



Gambar 4.21 sequence diagram mengubah berita

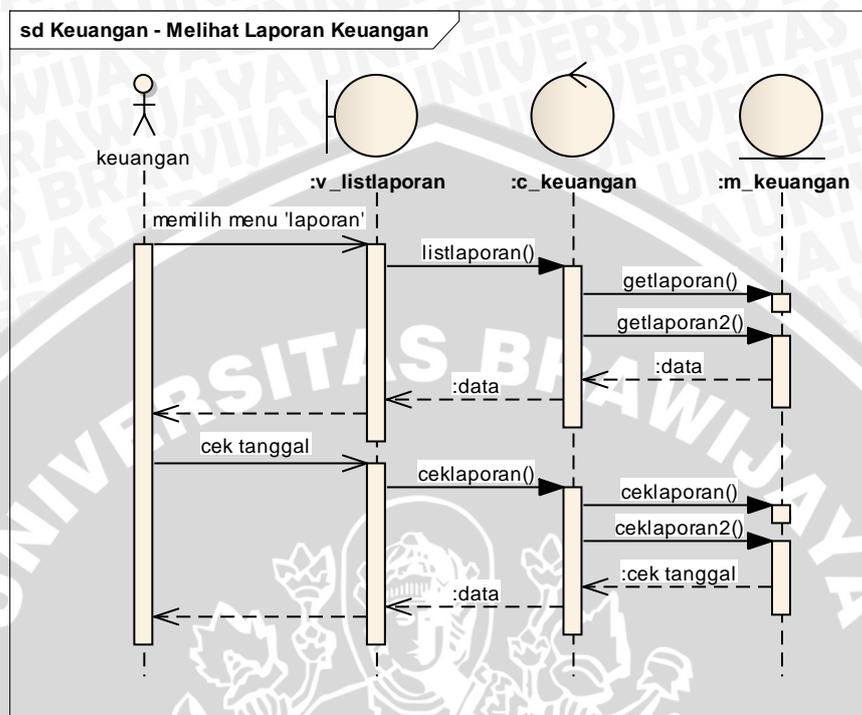


Gambar 4.22 sequence diagram menghapus berita

1. Sequence Diagram Melihat Laporan Keuangan

Sequence diagram melihat laporan keuangan adalah event ketika bagian keuangan ingin melihat laporan keuangan dari pemesanan dan

pendaftaran membership. Perancangan *sequence diagram* melihat laporan keuangan dapat dilihat pada Gambar 4.23.



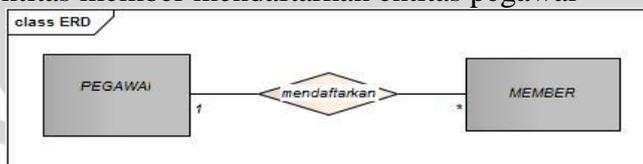
Gambar 4.23 *sequence diagram* melihat laporan keuangan

4.2.2 Perancangan Basis Data

4.2.2.1 Spesifikasi ERD

Untuk mendapatkan struktur tabel yang baik dan sesuai dengan kebutuhan Sistem Informasi, maka basis data dirancang dengan menggunakan ERD yang sebelumnya membuat spesifikasi ERD agar lebih mudah dan jelas dalam merancang ERD.

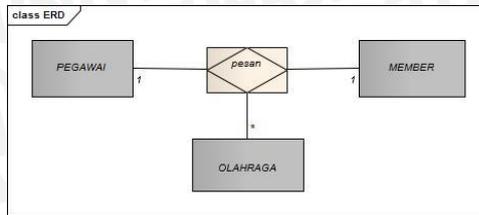
1. Entitas member mendaftarkan entitas pegawai



Gambar 4.24 Spesifikasi ERD entitas pegawai dan entitas member

Berdasarkan gambar 4.24 dalam entitas pegawai memiliki relasi *one to many* dengan entitas member. Dimana satu pegawai bisa mendaftarkan banyak member

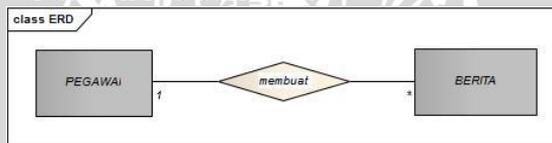
2. Entitas pegawai dan member memesan entitas olahraga



Gambar 4.25 Spesifikasi ERD entitas pegawai dan member

Berdasarkan gambar 4.25 dalam entitas pegawai memiliki relasi *one to many* dengan entitas olahraga dan entitas member memiliki relasi *one to many* dengan entitas olahraga. Dimana satu pegawai atau satu member dapat melakukan pemesanan banyak tempat olahraga.

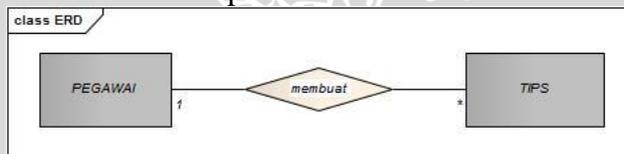
3. Entitas pegawai dan entitas berita



Gambar 4.26 Spesifikasi ERD entitas pegawai dan entitas berita

Berdasarkan gambar 4.26 dalam entitas pegawai memiliki relasi *one to many* dengan entitas berita. Dimana satu pegawai dapat membuat banyak berita.

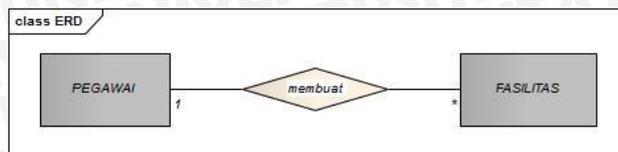
4. Entitas pegawai dan entitas tips



Gambar 4.27 Spesifikasi ERD entitas pegawai dan entitas tips

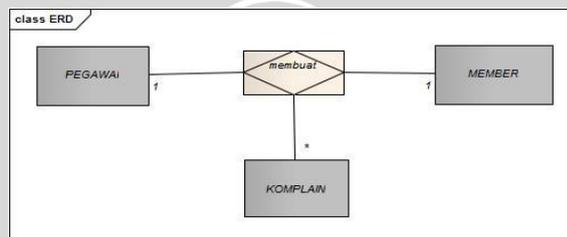
Berdasarkan gambar 4.27 dalam entitas pegawai memiliki relasi *one to many* dengan entitas tips. Dimana satu pegawai dapat membuat banyak tips.

5. Entitas pegawai dan entitas fasilitas



Gambar 4.28 Spesifikasi ERD entitas pegawai dan entitas fasilitas
Berdasarkan gambar 4.28 dalam entitas pegawai memiliki relasi *one to many* dengan entitas fasilitas. Dimana satu pegawai dapat membuat banyak fasilitas.

6. Entitas member dan pegawai membuat entitas komplain



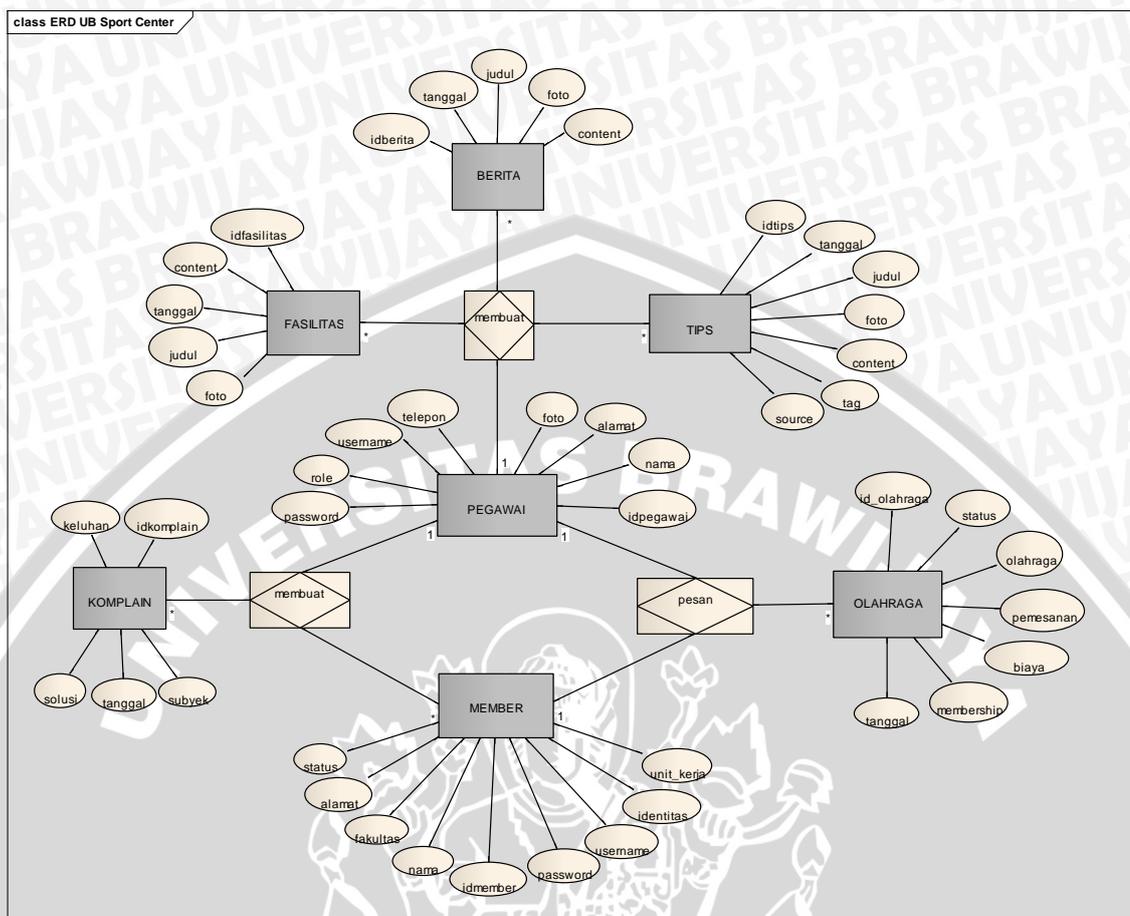
Gambar 4.29 Spesifikasi ERD entitas member dan komplain

Berdasarkan gambar 4.29 dalam entitas member memiliki relasi *one to many* dengan entitas complain. dan entitas pegawai memiliki relasi *one to many* dengan entitas komplain. dimana satu pegawai atau satu member dapat membuat banyak komplain.

4.2.2.2 DIAGRAM ERD

ERD digunakan untuk menggambarkan entitas-entitas secara umum yang terdapat pada Sistem Informasi Sport Club House Pertamina - UB. ERD akan menggambarkan hubungan entitas satu dengan lainnya.

Secara umum ERD dari struktur basis data pada Sistem Informasi Sport Club House Pertamina-UB ditunjukkan pada gambar 4.30.



Gambar 4.30 Entity Relasional Diagram(ERD)

4.2.2.3 ER Mapping

Dari *Entity Relationship Diagram* pada Gambar 4.30 dapat dibuat pemetaan (*ER-Mapping*) ke dalam tabel database yang ditunjukkan dengan uraian dibawah ini.

Tabel PEGAWAI

<u>ID_PEGAWAI</u>	USERNAME	NAMA	ALAMAT
EMAIL	PASSWORD	FOTO	ROLE
TELEPON	DATE		

Tabel KOMPLAIN

<u>IDKOMPLAIN</u>	PENERIMA	TANGGAL	PENGIRIM
SUBJEK	KELUHAN	SOLUSI	

Tabel FASILITAS

<u>IDFASILITAS</u>	JUDUL	CONTENT	FOTO	TANGGAL
--------------------	-------	---------	------	---------



Tabel BERITA

<u>IDBERITA</u>	JUDUL	CONTENT	FOTO	TANGGAL
-----------------	-------	---------	------	---------

Tabel TIPS

<u>IDTIPS</u>	JUDUL	CONTENT	FOTO	TANGGAL	TAG
---------------	-------	---------	------	---------	-----

Tabel MEMBER

<u>ID_MEMBER</u>	USERNAME	PASSWORD	TANGGAL
NAMA	ALAMATMALANG	FOTO	<u>IDENTITAS</u>
EMAIL	ALAMATASAL	NOTELEPON	TTL

Tabel OLAHRAGA

<u>IDOLAHRAGA</u>	<u>PEMESANAN</u>	<u>PENDAFTARAN</u>	PEMASUKAN
<u>OLAHRAGA</u>	STATUS	JADWAL	TANGGAL
TGL_PESAN	JAM AWAL	JAM AKHIR	BIAYA

Struktur tabel diatas merupakan tabel secara umum yang dapat dinormalkan. Data tersebut dapat ditunjukkan dalam uraian dibawah ini.

Tabel KOMPLAIN

<u>IDKOMPLAIN</u>	PENERIMA	TANGGAL	PENGIRIM
SOLUSI	KELUHAN	SUBJEK	

Tabel FASILITAS

<u>IDFASILITAS</u>	JUDUL	CONTENT	FOTO	TANGGAL
--------------------	-------	---------	------	---------

Tabel OLAHRAGA

<u>IDOLAHRAGA</u>	OLAHRAGA
-------------------	----------

Tabel BERITA

<u>IDBERITA</u>	JUDUL	CONTENT	FOTO	TANGGAL
-----------------	-------	---------	------	---------

Tabel TIPS

<u>IDTIPS</u>	JUDUL	CONTENT	FOTO	TANGGAL	TAG
---------------	-------	---------	------	---------	-----

Tabel MEMBER

<u>ID_MEMBER</u>	USERNAME	PASSWORD	TANGGAL
NAMA	ALAMATMALANG	FOTO	IDENTITAS
EMAIL	ALAMATASAL	NOTELEPON	TTL

Tabel MEMBERSHIP

<u>IDMEMBERSHIP</u>	TGL_AWAL	TGL_AKHIR	TGL_PESAN	TGL_BAYAR
TGL_EXPIRED	JAM_AWAL	STATUS	JAM_AKHIR	TEMPAT
BIAYA	JENIS			

Tabel PEGAWAI

<u>ID_PEGAWAI</u>	USERNAME	NAMA	ALAMAT
TELEPON	PASSWORD	FOTO	ROLE
IDENTITAS	DATE		

Tabel FAKULTAS

<u>IDFAKULTAS</u>	FAKULTAS
-------------------	----------

Tabel TIPEIDENTITAS

<u>IDTIPE</u>	TIPEIDENTITAS
---------------	---------------

Tabel UNIT_KERJA

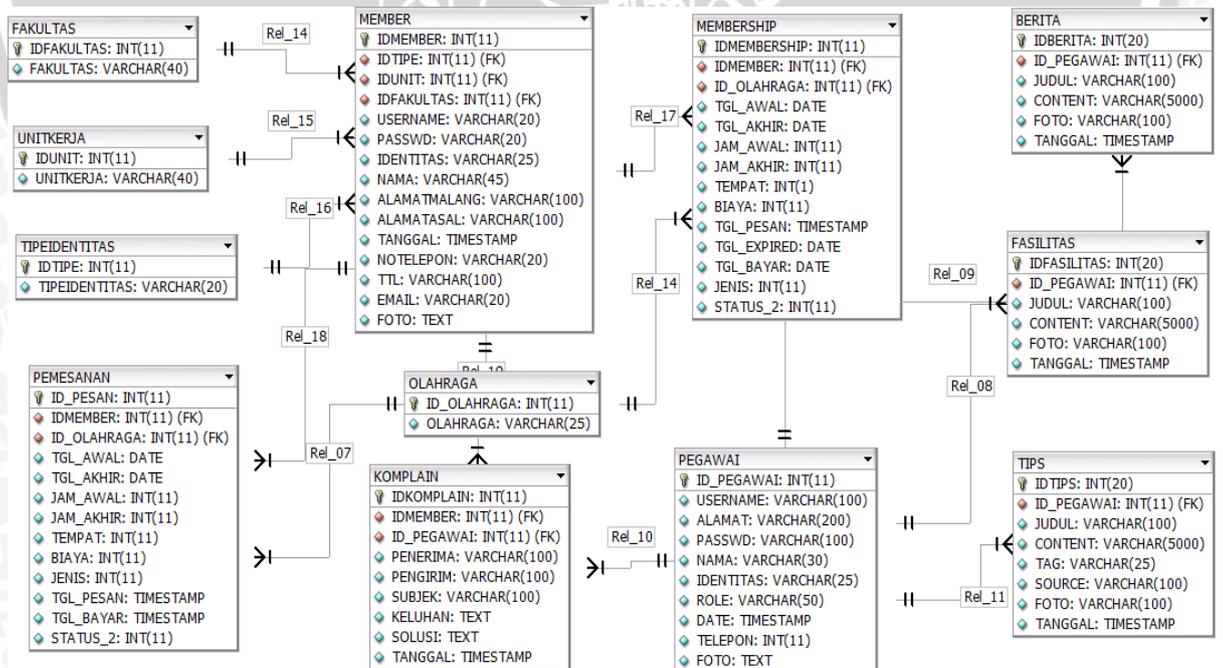
<u>IDUNIT</u>	UNITKERJA
---------------	-----------

Tabel PEMESANAN

<u>ID_PESAN</u>	TGL_AWAL	TGL_AKHIR	TGL_PESAN	TGL_BAYAR
JENIS	JAM_AWAL	STATUS	JAM_AKHIR	TEMPAT
BIAYA				

4.2.2.4 Physical Data Models

Perancangan basis data dapat dilakukan dengan menggunakan diagram physical data model seperti gambar 4.31 berikut ini.



Gambar 4.31 Physical Data Model

4.2.3 Perancangan Activity Diagram

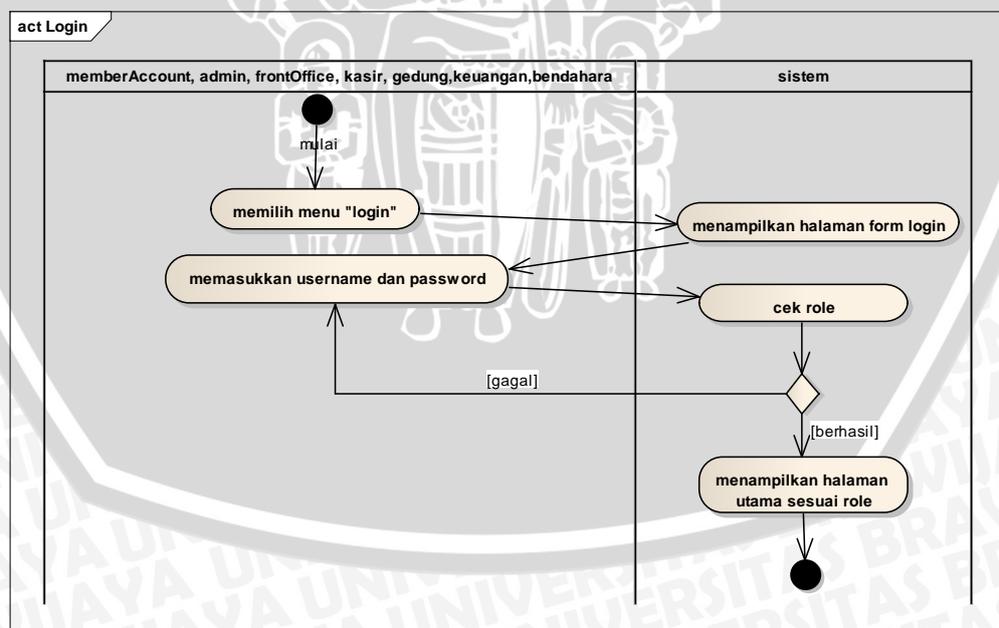
Diagram activity menggambarkan pemodelan proses sebuah interaksi antara *user* dengan sistem. Pembuatan *activity diagram* ini bertujuan untuk menggambarkan urutan aktivitas dari proses dari setiap *use case* yang ada. Berikut merupakan gambar dari masing-masing *activity diagram*.

a. Activity Diagram login

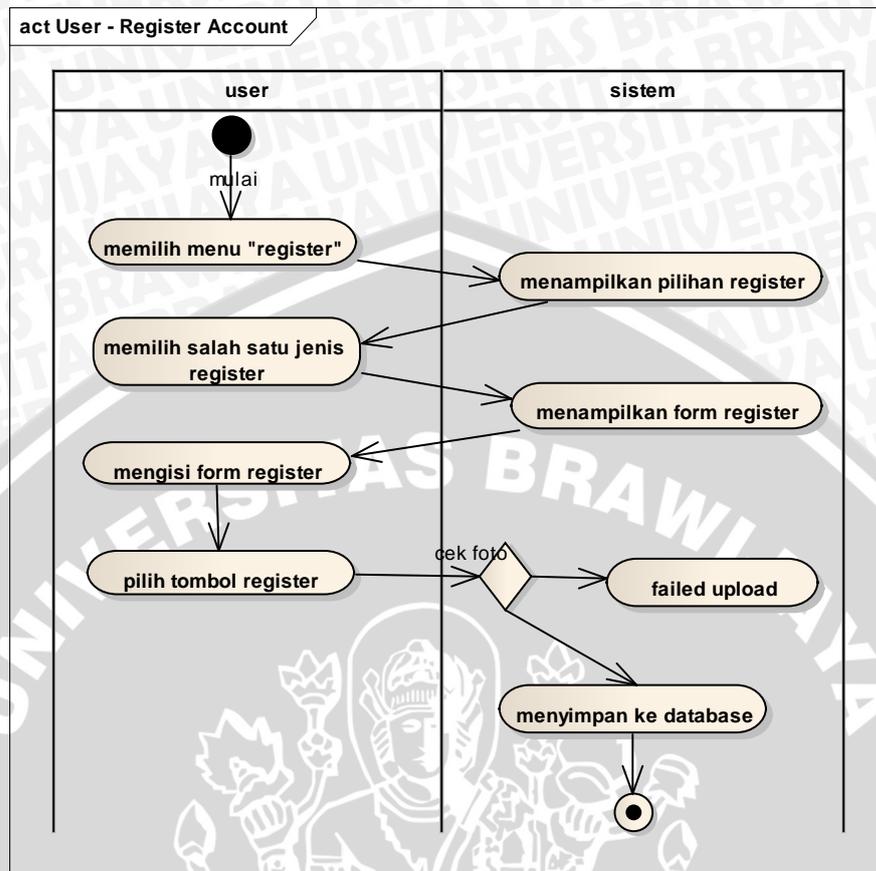
Activity diagram login adalah *event* ketika *user* ingin login ke dalam sistem Sport Club House Pertamina - UB. Perancangan *activity diagram login* dapat dilihat pada Gambar 4.32.

b. Activity Diagram Register Account

Activity diagram register account adalah *event* ketika *user* ingin mendaftarkan sebagai *member account* untuk melakukan pemesanan tempat olahraga yang disediakan Sport Club House Pertamina - UB. Perancangan *activity diagram register account* dapat dilihat pada Gambar 4.33.



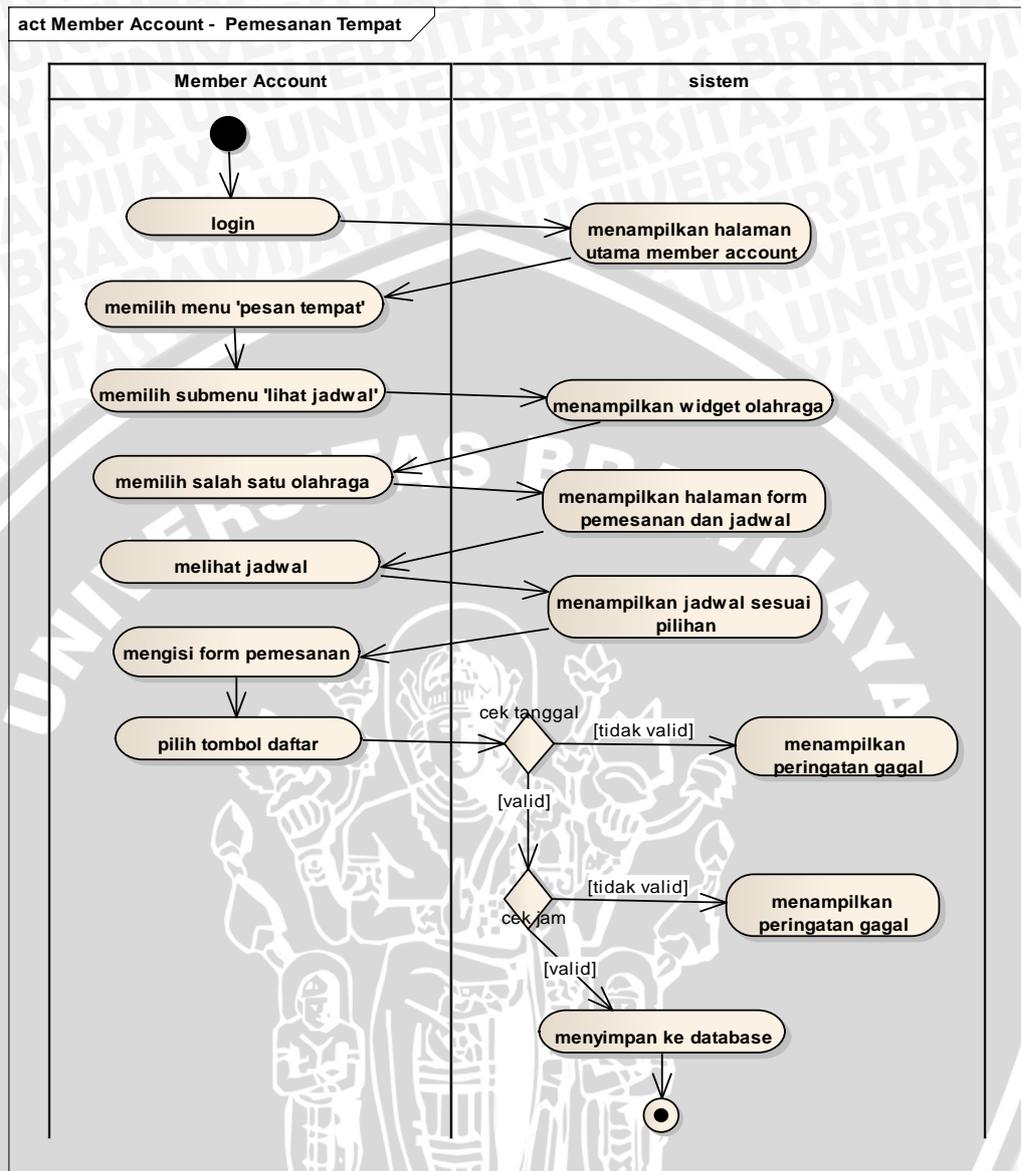
Gambar 4.32 activity diagram Login



Gambar 4.33 activity diagram register account

c. Activity Diagram Pemesanan Tempat

Activity diagram pemesanan tempat adalah event ketika member account ingin memesan tempat olahraga melalui sistem informasi dengan menampilkan seluruh jadwal pada masing-masing olahraga. Perancangan activity diagram pemesanan tempat dapat dilihat pada Gambar 4.34.



Gambar 4.34 activity diagram pemesanan tempat

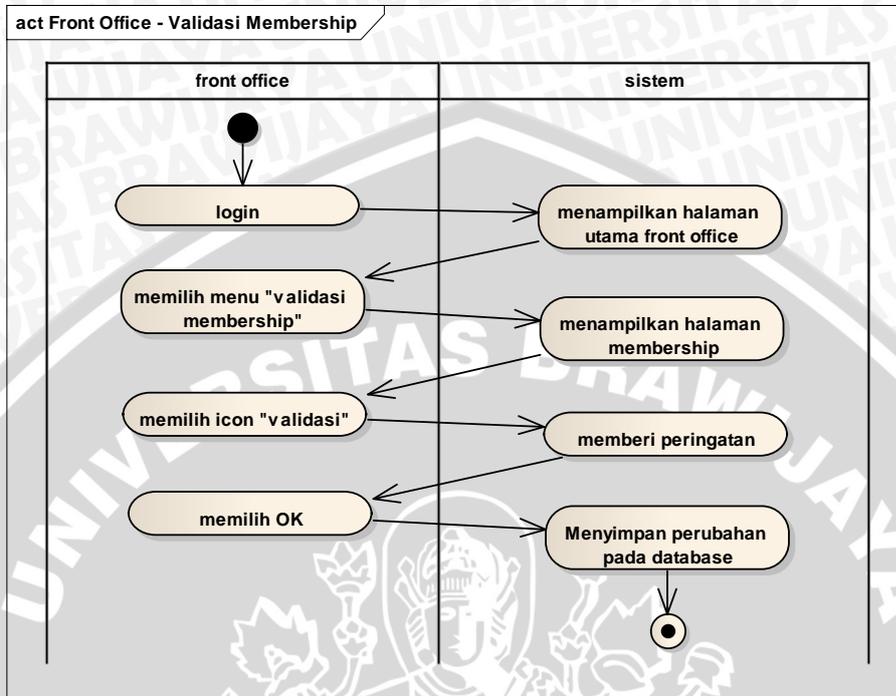
d. Activity Diagram validasi membership

Activity diagram validasi membership adalah event ketika front office ingin memvalidasia data member olahraga. Perancangan activity diagram validasi membership dapat dilihat pada Gambar 4.35.

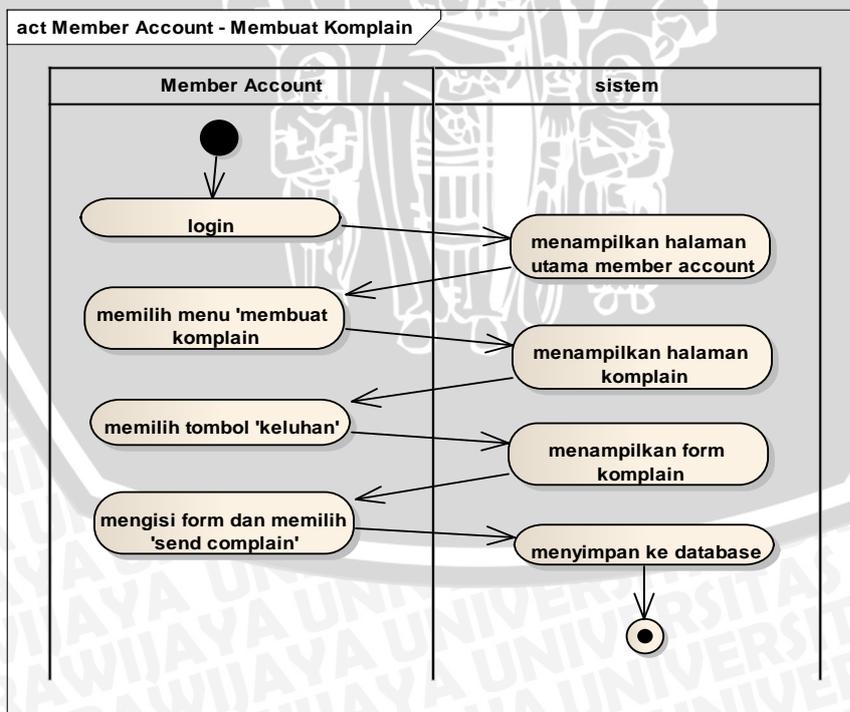
e. Activity Diagram Membuat Komplain

Activity diagram membuat komplain adalah event ketika member account ingin membuat complain kepada pihak Admin UB Sport Club

House. Perancangan *activity diagram* membuat komplain dapat dilihat pada Gambar 4.36.



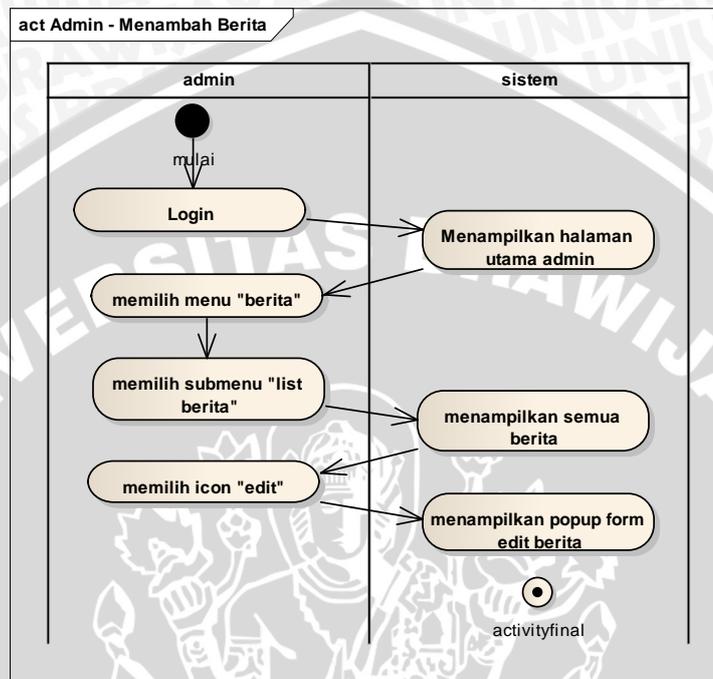
Gambar 4.35 *activity diagram* validasi membership



Gambar 4.36 *activity diagram* membuat komplain

f. Activity Diagram Menambah Berita

Activity diagram menambah berita adalah *event* ketika admin ingin menambah berita seputar UB Sport Club House. Perancangan *activity diagram* menambah berita dapat dilihat pada Gambar 4.37.



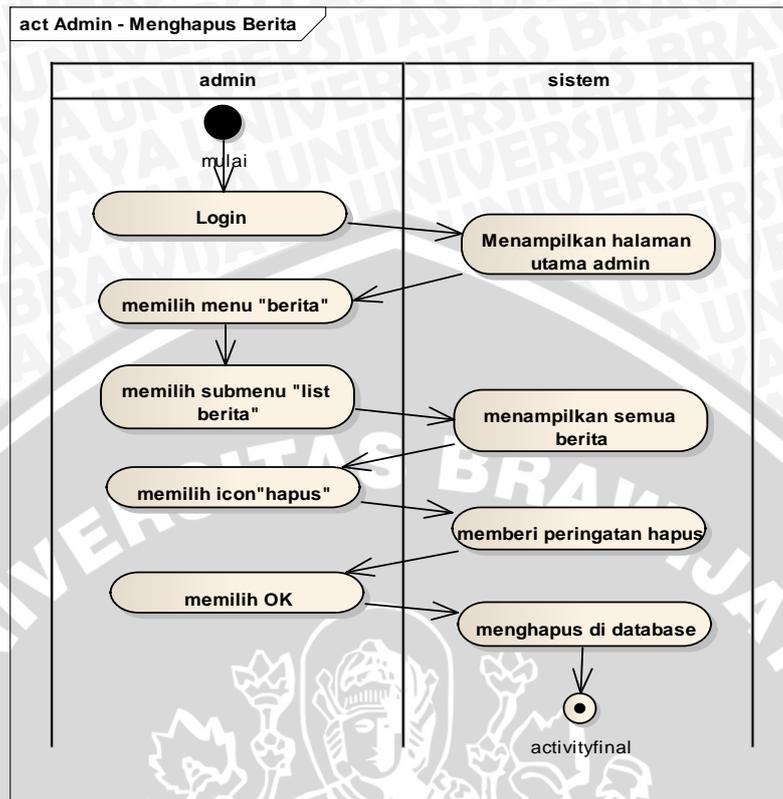
Gambar 4.37 *activity diagram* menambah berita

g. Activity Diagram Menghapus Berita

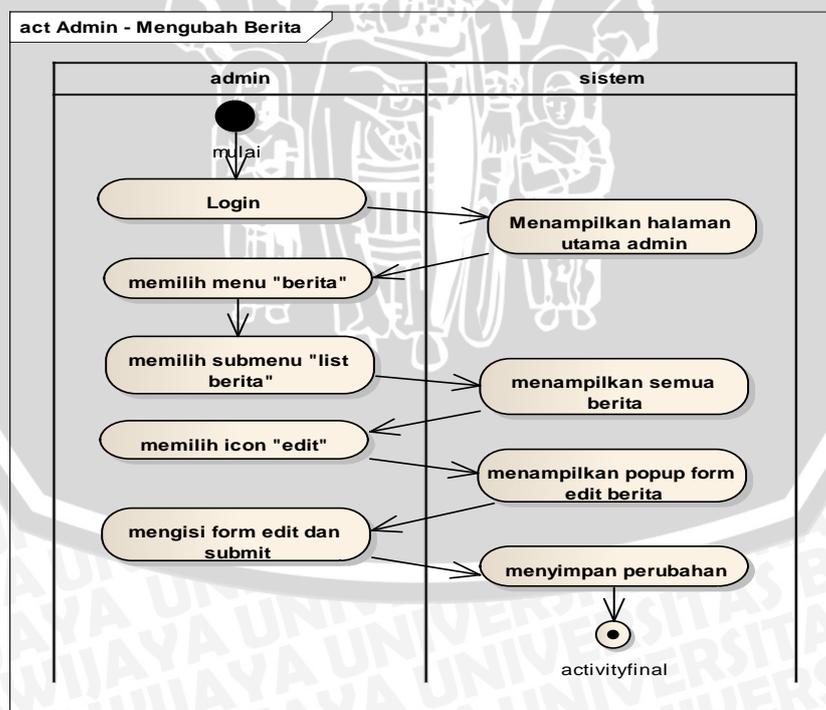
Activity diagram menghapus berita adalah *event* ketika admin ingin menghapus berita seputar UB Sport Club House. Perancangan *activity diagram* menghapus berita dapat dilihat pada Gambar 4.38.

h. Activity Diagram Mengubah Berita

Activity diagram mengubah berita adalah *event* ketika admin ingin mengubah berita seputar UB Sport Club House. Perancangan *activity diagram* mengubah berita dapat dilihat pada Gambar 4.39.



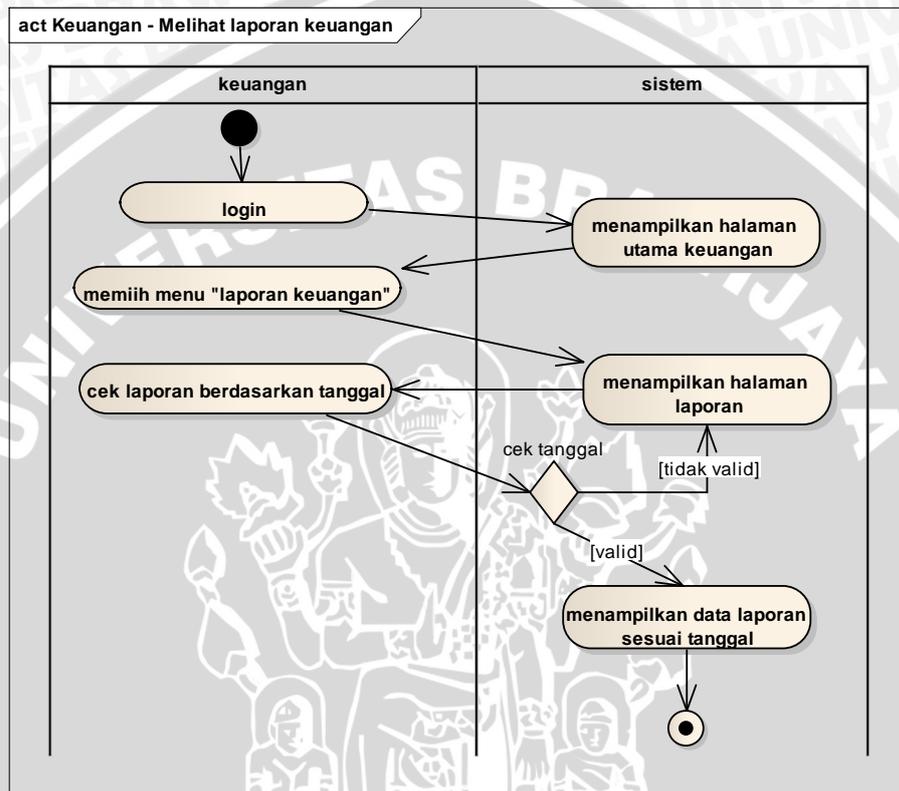
Gambar 4.38 activity diagram menghapus berita



Gambar 4.39 activity diagram mengubah berita

i. Activity Diagram Melihat Laporan Keuangan

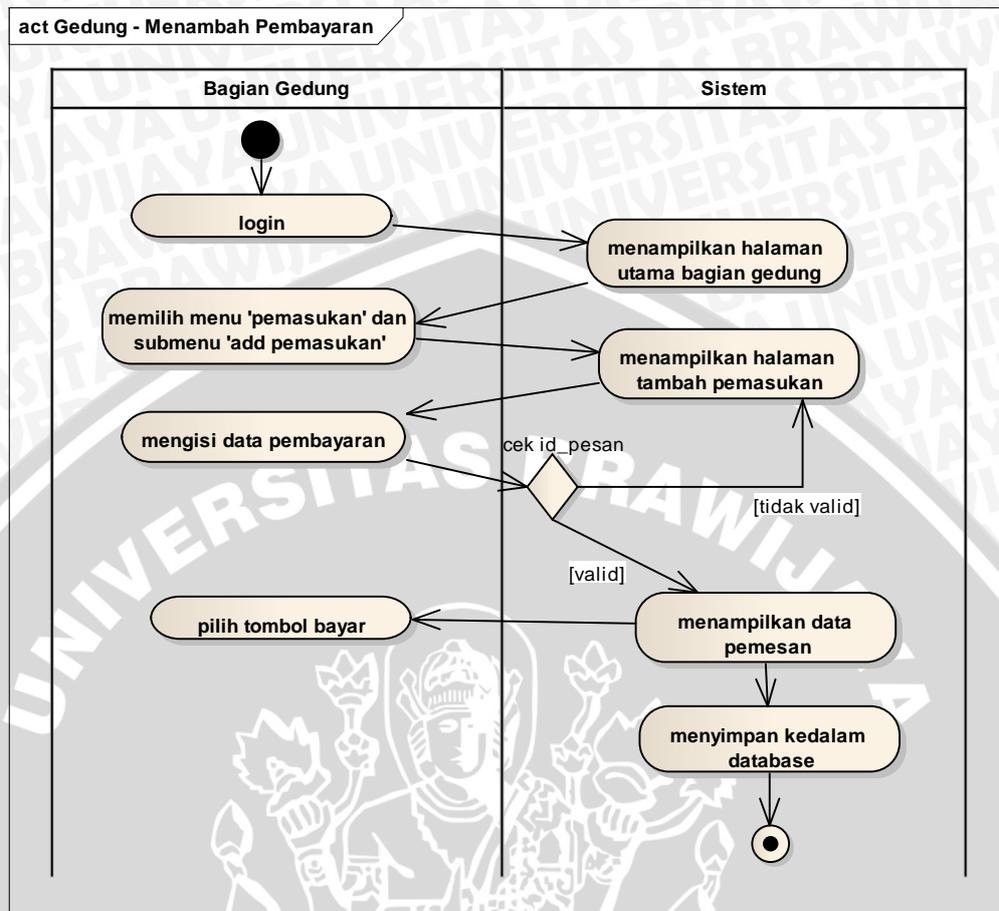
Activity diagram melihat laporan keuangan adalah event ketika bagian keuangan ingin melihat laporan keuangan pemasukan. Perancangan activity diagram melihat laporan keuangan dapat dilihat pda Gambar 4.40.



Gambar 4.40 activity diagram melihat laporan keuangan

j. Activity Diagram Menambah Pembayaran

Activity diagram menambah pembayaran adalah event ketika admin ingin menambah pembayaran olahraga. Perancangan activity diagram menambah pembayaran dapat dilihat pada Gambar 4.41



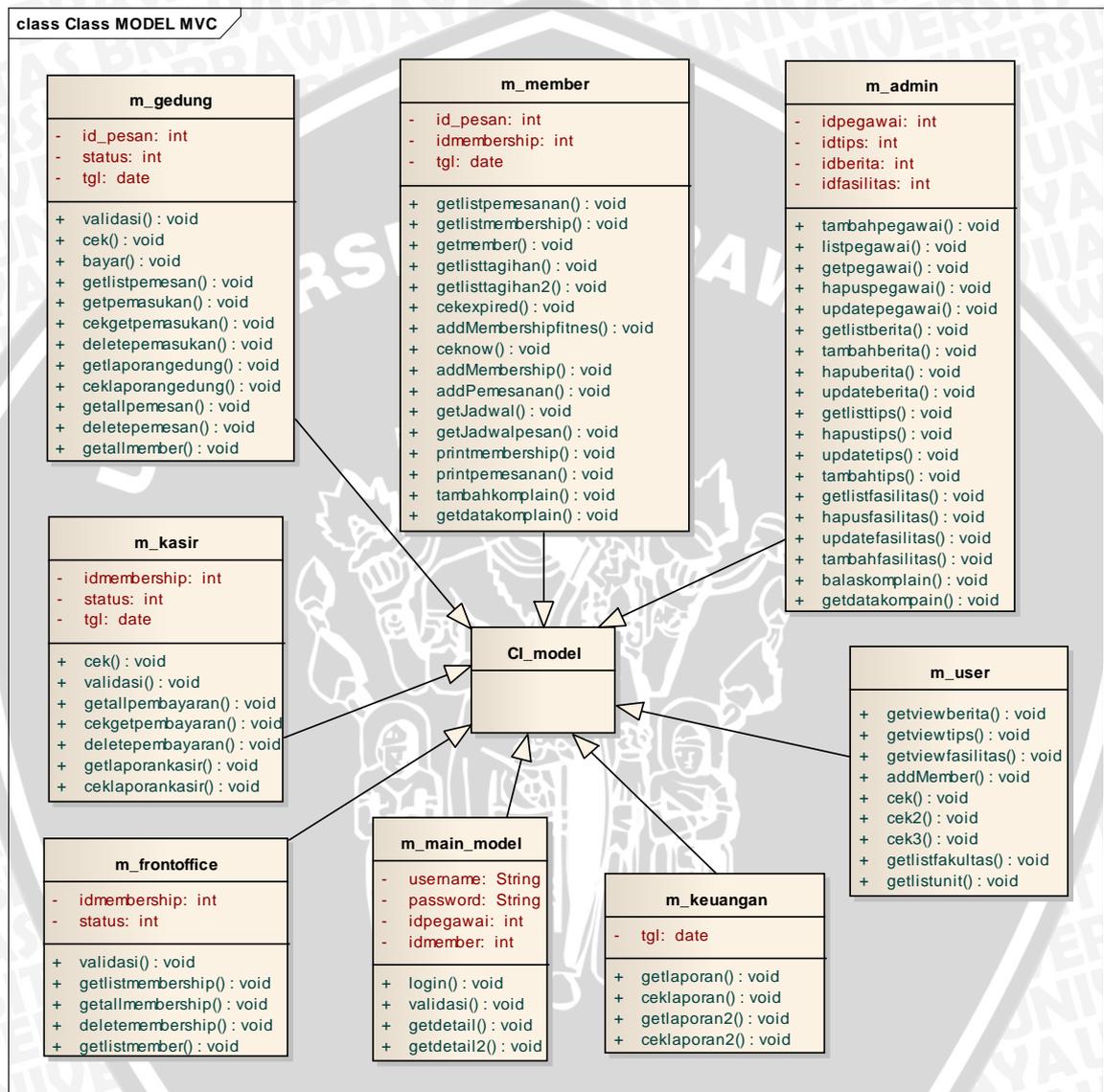
Gambar 4.41 activity diagram menambah pembayaran

4.2.4 Perancangan Class Diagram

Diagram *class* memberikan gambaran pemodelan elemen-elemen *class* yang membentuk sebuah sistem. *Class* bisa didapatkan dengan menganalisis secara detail terhadap *use case* yang dimodelkan.

Pada class diagram ini ada 3 jenis perancangan kelas yakni *entity*, *control*, dan *view*, dimana termasuk uraian class analysis diagram. Kelas-kelas ini nantinya akan diimplementasikan dalam bentuk MVC.

Berikut merupakan diagram kelas dari Sistem Sport Club House Pertamina – UB yang akan dijelaskan rancangan kelas-kelas model aplikasi pada Gambar 4.42.



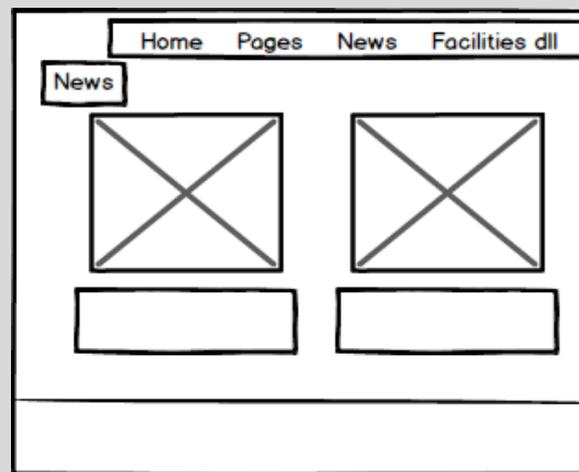
Gambar 4.42 Class Diagram pada model

4.2.5 Perancangan Prototype

Pada bagian ini akan dijelaskan tentang perancangan prototype sistem informasi Sport Club House Pertamina - UB. Berikut merupakan rancangan prototype Sistem Informasi Sport Club House Pertamina - UB.

1. Rancangan Prototype Halaman Melihat Berita

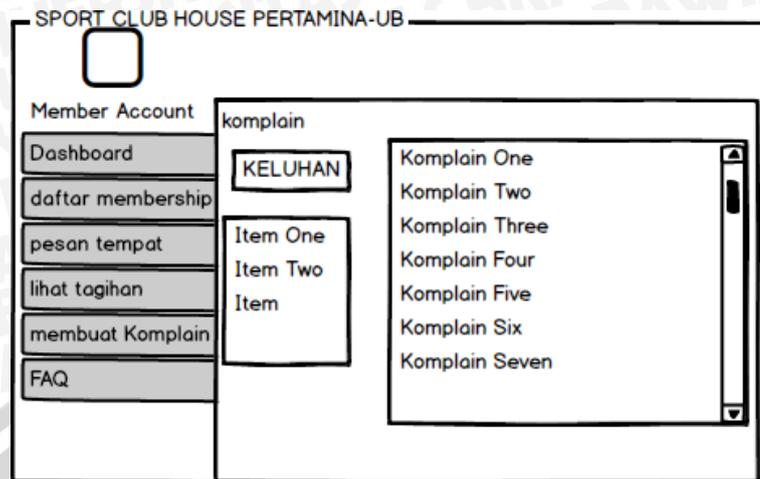
Halaman melihat berita merupakan halaman *user*. Halaman ini menampilkan informasi berita tentang Sport Club House Pertamina - UB. Berikut ini merupakan rancangan prototype halaman berita pada gambar 4.45.



Gambar 4.45 Rancangan prototype halaman berita

2. Rancangan Prototype Halaman Membuat Komplain

Halaman membuat komplain merupakan halaman dari member account. Halaman ini menampilkan informasi data complain dan menambah komplain. Berikut ini merupakan rancangan prototype halaman membuat komplain pada gambar 4.46.



Gambar 4.46 Rancangan prototype halaman membuat complain

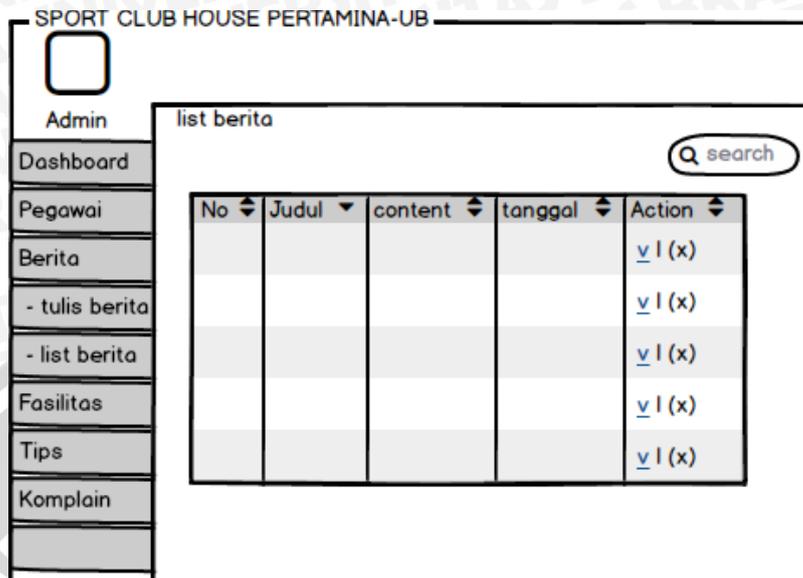
3. Rancangan Prototype Halaman RegisterAccount

Halaman register account merupakan halaman dari *user*. Halaman ini menampilkan form register untuk menjadi member account. Berikut ini merupakan rancangan prototype halaman register account pada gambar 4.47.

Gambar 4.47 Rancangan prototype halaman register account

4. Rancangan Prototype Halaman Mengelola Berita

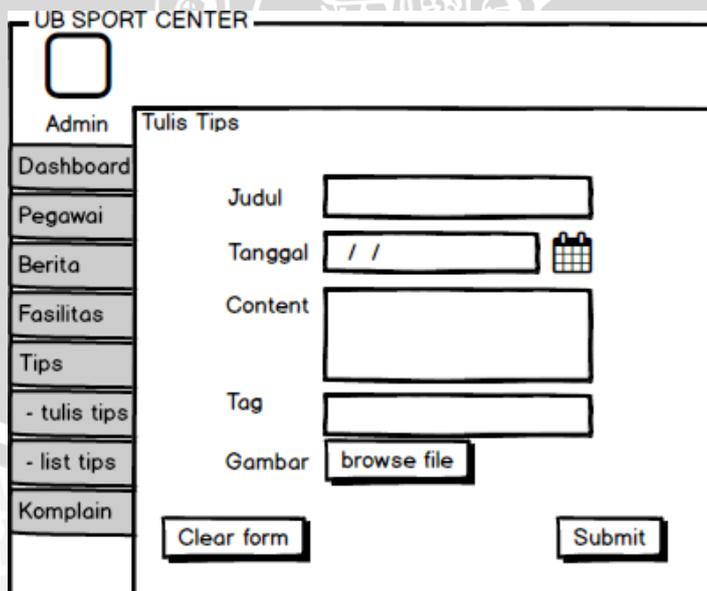
Halaman mengelola berita merupakan halaman dari admin. Halaman ini menampilkan seluruh data berita. Berikut ini merupakan rancangan prototype halaman mengelola berita pada gambar 4.48.



Gambar 4.48 Rancangan prototype halaman mengelola berita

5. Rancangan Prototype Halaman menambah tips olahraga

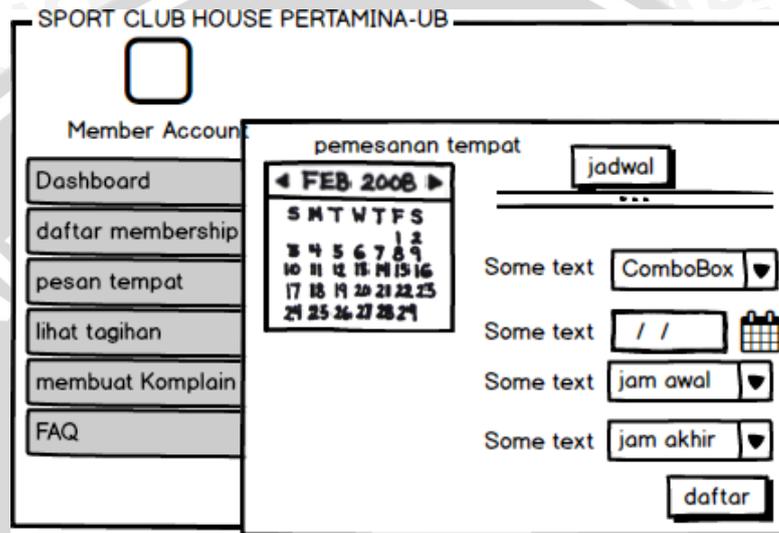
Halaman menambah tips merupakan dari actor admin. Halaman ini digunakan untuk menambah tips olahraga. Berikut ini merupakan rancangan prototype halaman menambah tips olahraga pada gambar 4.49.



Gambar 4.49 Rancangan prototype halaman menambah tips olahraga

6. Rancangan Prototype Halaman Pemesanan Tempat

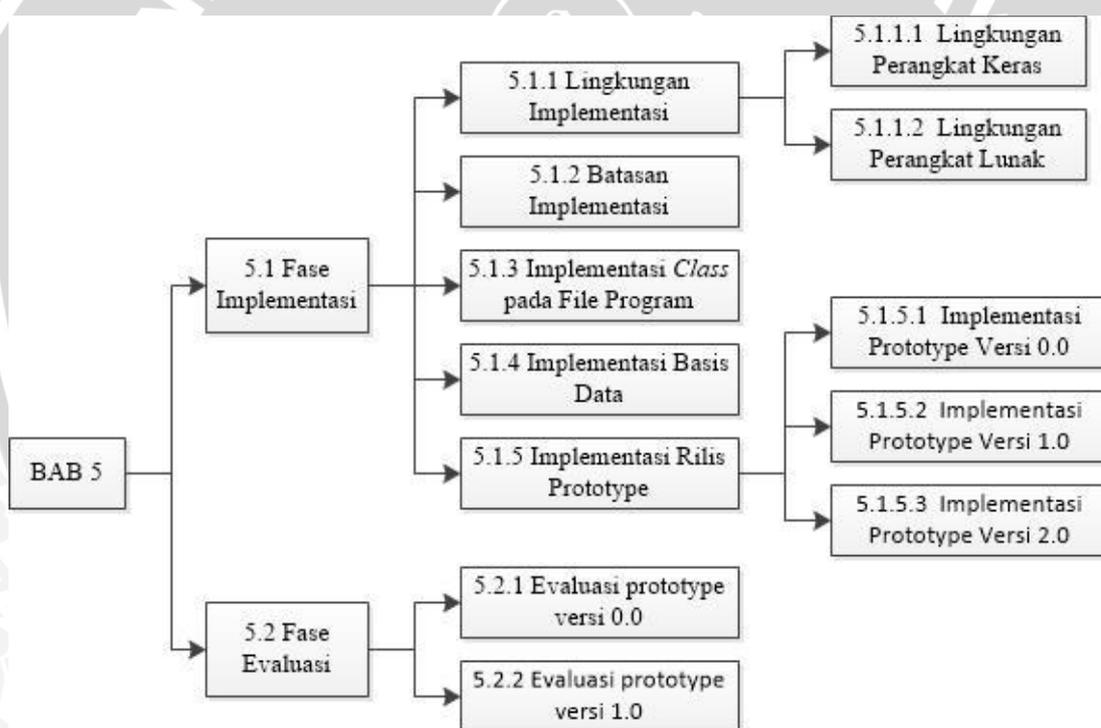
Halaman pemesanan tempat merupakan dari actor member account. Halaman ini digunakan untuk melakukan pemesanan tempat olahraga member. Berikut ini merupakan rancangan prototype halaman pemesanan tempat pada gambar 4.50.



Gambar 4.50 Rancangan prototype halaman pemesanan tempat

BAB 5 IMPLEMENTASI

Bab ini membahas mengenai implementasi sistem informasi Sport Club House Pertamina-UB dan hasil evaluasi pengguna. Fase Implementasi didasarkan pada hasil yang didapat dari proses analisis kebutuhan dan perancangan aplikasi. Pembahasan terdiri atas penjelasan tentang lingkungan implementasi, batasan-batasan dalam implementasi, implementasi tiap *class* pada *file program*, implementasi basis data, dan implementasi rilis prototype. Fase evaluasi diperoleh dari hasil implementasi prototype yang dilihat pengguna kemudian pengguna memberi *feedback* yang menghasilkan evaluasi prototype. Struktur bab implementasi ditunjukkan oleh Gambar 5.1.



Gambar 5.1 Diagram Blok Bab Implementasi

5.1 Fase Implementasi

5.1.1 Deskripsi Lingkungan Perangkat Keras

Sistem informasi ini dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak yang menggunakan bahasa pemrograman tertentu.

5.1.1.1 Deskripsi Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang dipakai dalam proses pengembangan dijelaskan pada tabel 5.1 :

Tabel 5.1 Spesifikasi perangkat keras komputer

PC- DELL inspiron N4050	
<i>Processor</i>	Intel® Core™ i3-2310M CPU @2.10GHz
<i>Memory (RAM)</i>	4096MB RAM
<i>Harddisk</i>	500GB
<i>Monitor</i>	Generic PnP Monitor

5.1.1.2 Deskripsi Lingkungan Perangkat Lunak

Spesifikasi perangkat lunak yang dipakai dalam proses pengembangan sistem informasi dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat keras komputer

PC Dell-Inspiron N4050	
<i>Operating System</i>	Microsoft Windows 8 Pro 64-bit
<i>Programming Language</i>	PHP, HTML, CSS 3, java script
<i>Editor</i>	SublimeText, phpDesigner8, Enterprise Architect 8, DBDesigner4
<i>Database Management System</i>	XAMPP 1.7.3

5.1.2 Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem informasi ini adalah sebagai berikut :

1. Sistem yang dibangun khususnya menangani di bagian sistem informasi olahraga, pendaftaran olahraga membership, pemesanan tempat, dan complain.
2. *Platform* pembuatan yang digunakan adalah PHP dengan *framework Code Igniter*.
3. *Database Management System* yang digunakan adalah MySQL yang terletak di *localhost*.
4. Penerapan *database* belum memakai metode keamanan data yang sempurna, seperti penggunaan enkripsi untuk *password*.

5.1.3 Implementasi Class Pada File Program

Implementasi Sistem Informasi Sport Club House Pertamina - UB ini menggunakan *framework CodeIgniter* yang menerapkan pola MVC. Untuk memudahkan pengembangan sistem ini akan dijelaskan masing-masing file *Class Controllers* dan *models* yang terdapat dalam folder “ubsportcenter”. Berikut ini merupakan hubungan antar *class* dengan file pada tabel 5.3

Tabel 5.3 Struktur *Class* dan lokasi file

No	Package	Nama Class	Lokasi File	Nama File Program
1.	<i>Controllers</i>	CI_Controllers	/system/core/	Controllers.php
2.	<i>Controllers</i>	Main_Controller	/application/controller/	main_controller.php
3.	<i>Controllers</i>	C_member	/application/controller/	c_member.php
4.	<i>Controllers</i>	C_admin	/application/controller/	c_admin.php
5.	<i>Controllers</i>	C_frontoffice	/application/controller/	c_frontoffice.php
6.	<i>Controllers</i>	C_gedung	/application/controller/	c_gedung.php
7.	<i>Controllers</i>	C_kasir	/application/controller/	c_kasir.php
8.	<i>Controllers</i>	C_keuangan	/application/controller/	c_keuangan.php
9.	<i>Controllers</i>	C_bendahara	/application/controller/	c_bendahara.php
10.	<i>models</i>	CI_Model	/application/models/	Model.php
11.	<i>models</i>	M_main_model	/application/models/	m_main_model.php
12.	<i>models</i>	M_member	/application/models/	m_member.php
13.	<i>models</i>	M_admin	/application/models/	m_admin.php
14.	<i>models</i>	M_gedung	/application/models/	m_gedung.php
15.	<i>models</i>	M_frontoffice	/application/models/	m_frontoffice.php
16.	<i>models</i>	M_kasir	/application/models/	m_kasir.php
17.	<i>models</i>	M_keuangan	/application/models/	m_keuangan.php

5.1.4 Implementasi Basis Data

Implementasi database dilakukan berdasarkan perancangan pada bab sebelumnya pada subbab 4.2.2 tentang perancangan basis data menggunakan MySQL. Struktur tabel yang digunakan pada implementasi sistem ini sama dengan perancangan yang telah dibuat sebelumnya. Pernyataan yang digunakan untuk menciptakan suatu tabel dalam basis data

menggunakan DDL (Data Definition Language) yang dijelaskan pada tabel 5.4 hingga tabel 5.15.

Tabel 5.4 Struktur DDL Tabel fakultas

```
create table if not exists `fakultas` (
  `idfakultas` int(11) not null auto_increment primary key,
  `fakultas` varchar(40) default null
)
```

Pada tabel 5.4 dijelaskan bahwa tabel fakultas akan digunakan untuk menyimpan data fakultas. Pada tabel ini terdapat beberapa atribut, diantaranya idfakultas sebagai *primary key*, dan fakultas digunakan untuk menyimpan nama fakultas.

Tabel 5.5 Struktur DDL Tabel unitkerja

```
create table if not exists `unitkerja` (
  `idunit` int(11) not null auto_increment primary key,
  `unitkerja` varchar(40) default null
)
```

Pada tabel 5.5 dijelaskan bahwa tabel unitkerja akan digunakan untuk menyimpan data unit kerja. Pada tabel ini terdapat beberapa atribut, diantaranya idunit sebagai *primary key*, dan unitkerja digunakan untuk menyimpan nama unit kerja.

Tabel 5.6 Struktur DDL Tabel tipeidentitas

```
create table if not exists `tipeidentitas` (
  `idtype` int(11) not null auto_increment primary key,
  `tipeidentitas` varchar(20) default null
)
```

Pada tabel 5.6 dijelaskan bahwa tabel tipeidentitas akan digunakan untuk menyimpan data tipe identitas. Pada tabel ini terdapat beberapa atribut, diantaranya idtype sebagai *primary key*, dan tipeidentitas digunakan untuk menyimpan nama tipe identitas.

Tabel 5.7 Struktur DDL Tabel member

```
create table if not exists `member` (
  `idmember` int(11) not null auto_increment primary key,
  `idfakultas` int(11) default null,
  `idunit` int(11) default null,
  `idtype` int(11) default null,
  `username` varchar(20) default null,
  `password` varchar(20) default null,
```

<code>`identitas` varchar(25) default null,</code>
<code>`nama` varchar(45) default null,</code>
<code>`alamatmalang` varchar(100) default null,</code>
<code>`alamatasal` varchar(100) default null,</code>
<code>`tanggal` timestamp,</code>
<code>`notelepon` varchar(20) default null,</code>
<code>`ttl` varchar(100) default null,</code>
<code>`foto` text, `email` varchar(20) default null,</code>
<code>foreign key `idfakultas` references fakultas(`idfakultas`),</code>
<code>foreign key `idfakultas` references fakultas(`idfakultas`),</code>
<code>foreign key `idfakultas` references fakultas(`idfakultas`)</code>

Pada tabel 5.7 dijelaskan bahwa tabel member akan digunakan untuk menyimpan data member. Pada tabel ini terdapat beberapa atribut, diantaranya idmember sebagai *primary key*, idfakultas sebagai foreign key yang me-reference ke tabel fakultas, idtipe sebagai *foreign key* yang me-reference ke tabel tipeidentitas, idunit sebagai *foreign key* yang me-reference ke tabel unitkerja, username untuk menyimpan username member, password untuk menyimpan password member, nama untuk menyimpan nama member, alamatmalang dan alamatasal untuk menyimpan alamat member, notelepon untuk menyimpan no telepon, email untuk menyimpan email member.

Tabel 5.8 Struktur DDL Tabel pegawai

<code>create table if not exists `pegawai` (</code>
<code>`idpegawai` int(11) not null auto_increment primary key,</code>
<code>`username` varchar(100) default null,</code>
<code>`password` varchar(100) default null,</code>
<code>`identitas` varchar(25) default null,</code>
<code>`nama` varchar(45) default null,</code>
<code>`alamat` varchar(200) default null,</code>
<code>`telepon` varchar(11) default null,</code>
<code>`role` varchar(50) default null,</code>
<code>`foto` text, `date` timestamp)</code>

Pada tabel 5.8 dijelaskan bahwa tabel pegawai akan digunakan untuk menyimpan data pegawai. Pada tabel ini terdapat beberapa atribut, diantaranya idpegawai sebagai *primary key*, username untuk menyimpan username pegawai, password untuk menyimpan password pegawai, nama untuk menyimpan nama

pegawai, alamat untuk menyimpan alamat, notelepon untuk menyimpan no telepon, role untuk menyimpan role member dan foto untuk menyimpan foto.

Tabel 5.9 Struktur DDL Tabel berita

create table if not exists `berita` (
`idberita` int(20) not null auto_increment primary key,
`idpegawai` int (11) default null,
`judul` varchar(100) default null,
`content` varchar(5000) default null,
`foto` varchar(100) default null,
`tanggal` timestamp,
foreign key `idpegawai` references pegawai(`idpegawai`))

Pada tabel 5.9 dijelaskan bahwa tabel berita akan digunakan untuk menyimpan berita. Pada tabel ini terdapat beberapa atribut, diantaranya idberita sebagai *primary key*, idpegawai sebagai foreign key yang me-reference ke tabel pegawai, judul untuk menyimpan judul berita, content untuk menyimpan isi berita, foto untuk menyimpan foto berita.

Tabel 5.10 Struktur DDL Tabel fasilitas

create table if not exists `fasilitas` (
`idfasilitas` int(20) not null auto_increment primary key,
`idpegawai` int (11) default null,
`judul` varchar(100) default null,
`content` varchar(5000) default null,
`foto` varchar(100) default null,
`tanggal` timestamp,
foreign key `idpegawai` references pegawai(`idpegawai`))

Pada tabel 5.10 dijelaskan bahwa tabel fasilitas akan digunakan untuk menyimpan fasilitas. Pada tabel ini terdapat beberapa atribut, diantaranya idfasilitas sebagai *primary key*, idpegawai sebagai foreign key yang me-reference ke tabel pegawai, judul untuk menyimpan judul, content untuk menyimpan isi dan foto untuk menyimpan foto fasilitas.

Tabel 5.11 Struktur DDL Tabel tips

create table if not exists `tips` (
`idtips` int(20) not null auto_increment primary key,
`idpegawai` int (11) default null,
`judul` varchar(100) default null,
`content` varchar(5000) default null,
`foto` varchar(100) default null,

<code>`tag` varchar(25) default null,</code>
<code>`source` varchar(100) default null,</code>
<code>`tanggal` timestamp,</code>
<code>foreign key `idpegawai` references pegawai(`idpegawai`))</code>

Pada tabel 5.11 dijelaskan bahwa tabel tips akan digunakan untuk menyimpan tips olahraga. Pada tabel ini terdapat beberapa atribut, diantaranya idtips sebagai *primary key*, idpegawai sebagai foreign key yang me-reference ke tabel pegawai, judul untuk menyimpan judul, content untuk menyimpan isi, source untuk menyimpan sumber dan foto untuk menyimpan foto fasilitas.

Tabel 5.12 Struktur DDL Tabel membership

<code>create table if not exists `membership` (</code>
<code>`idmembership` int(11) not null auto_increment primary key,</code>
<code>`idmember` int(11) default null,</code>
<code>`id_olahraga` int(11) default null,</code>
<code>`tgl_awal` date, `tgl_akhir` date,</code>
<code>`jam_awal` int(11), `jam_akhir` int(11),</code>
<code>`tempat` int(11), `biaya` int(11),</code>
<code>`tgl_pesan` timestamp, `tgl_expired` date,</code>
<code>`jenis` int(11), `status` int(11), `tgl_bayar` date,</code>
<code>foreign key `idmember` references member(`idmember`),</code>
<code>foreign key `id_olahraga` references</code>
<code>olahraga(`id_olahraga`))</code>

Pada tabel 5.12 dijelaskan bahwa tabel membership akan digunakan untuk menyimpan data membership. Pada tabel ini terdapat beberapa atribut, diantaranya idmembership sebagai *primary key*, idmember sebagai foreign key yang me-reference ke tabel member, id_olahraga sebagai *foreign key* yang me-reference ke tabel olahraga, tgl_awal untuk menyimpan tanggal awal daftar, jam_awal untuk menyimpan jam awal daftar, tgl_akhir untuk menyimpan tanggal akhir daftar, jam_akhir untuk menyimpan jam akhir daftar, biaya untuk menyimpan biaya pendaftaran, status untuk menyimpan status member.

Tabel 5.13 Struktur DDL Tabel olahraga

<code>create table if not exists `olahraga` (</code>
<code>`id_olahraga` int(11) not null auto_increment primary key,</code>
<code>`olahraga` varchar(25) default null</code>
<code>)</code>

Pada tabel 5.13 dijelaskan bahwa tabel olahraga akan digunakan untuk menyimpan data olahraga. Pada tabel ini terdapat beberapa atribut, diantaranya

id_olahraga sebagai *primary key*, dan olahraga digunakan untuk menyimpan nama olahraga.

Tabel 5.14 Struktur DDL Tabel pemesanan

create table if not exists `pemesanan` (
`id_pesan` int(11) not null auto_increment primary key,
`idmember` int(11) default null,
`id_olahraga` int(11) default null,
`tgl_awal` date, `tgl_akhir` date,
`jam_awal` int(11), `jam_akhir` int(11),
`tempat` int(11), `biaya` int(11),
`tgl_pesan` timestamp, `tgl_bayar` date,
`jenis` int(11), `status` int(11),
foreign key `idmember` references member(`idmember`),
foreign key `id_olahraga` references
olahraga(`id_olahraga`)

Pada tabel 5.14 dijelaskan bahwa tabel pemesanan akan digunakan untuk menyimpan data pemesan. Pada tabel ini terdapat beberapa atribut, diantaranya id_pesan sebagai *primary key*, idmember sebagai foreign key yang me-reference ke tabel member, id_olahraga sebagai sebagai *foreign key* yang me-reference ke tabel olahraga, tgl_awal untuk menyimpan tanggal awal pesan, jam_awal untuk menyimpan jam awal pesan, tgl_akhir untuk menyimpan tanggal akhir daftar, jam_akhir untuk menyimpan jam akhir daftar, biaya untuk menyimpan biaya pemesanan, status untuk menyimpan status pemesan.

Tabel 5.15 Struktur DDL Tabel komplain

create table if not exists `komplain` (
`idkomplain` int(20) not null auto_increment primary key,
`idpegawai` int (11) default null,
`idmember` int (11) default null,
`subjek` varchar(100) default null,
`pengirim` varchar(100) default null,
`penerima` varchar(100) default null,
`keluhan` text default null,
`solusi` text default null,
`tanggal` timestamp,
foreign key `idpegawai` references pegawai(`idpegawai`),
foreign key `idmember` references member(`idmember`))

Pada tabel 5.15 dijelaskan bahwa tabel komplain akan digunakan untuk menyimpan komplain. Pada tabel ini terdapat beberapa atribut, diantaranya idkomplain sebagai *primary key*, idpegawai sebagai foreign key yang me-

reference ke tabel pegawai, idmember sebagai foreign key yang me-reference ke tabel member, subjek untuk menyimpan judul, keluhan untuk menyimpan isi, solusi untuk menyimpan solusi.

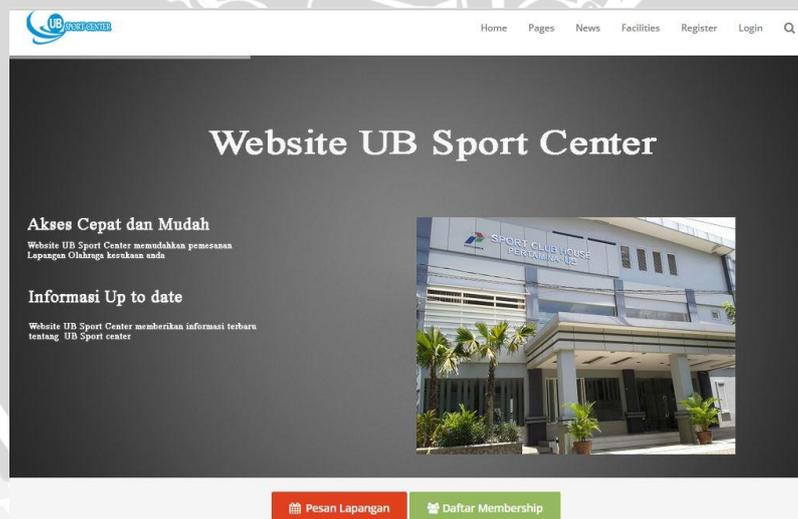
5.1.5 Implementasi Prototype Rilis

Implementasi sistem informasi Sport Club House Pertamina-UB digunakan oleh pengguna sistem yang dibagi menjadi 8 bagian yaitu user, admin, member account, front office, kasir, bagian gedung, keuangan dan bendahara. Dimana implementasi dibagi menjadi 3 versi rilis berdasarkan hasil 2 kali iterasi sistem rilis prototype kepada pengguna pihak Sport Club House Pertamina-UB. Hasil akhir dari rilis version adalah versi 2.0 pada Gambar 5.13 hingga Gambar 5.20.

5.1.5.1 Implementasi Prototype Versi 0.0

1. Halaman Beranda

Gambar 5.2 menggambarkan hasil implementasi dari halaman utama Sport Club House Pertamina-UB.



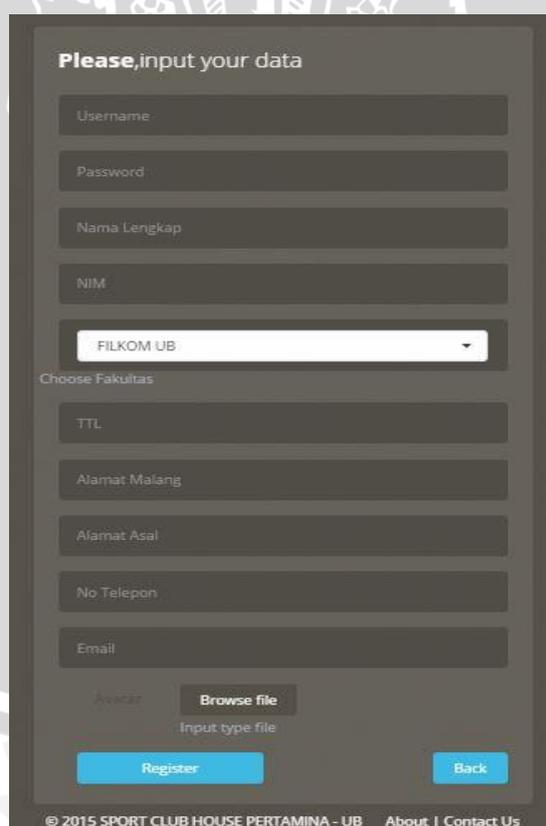
Gambar 5.2 Implementasi Beranda

2. Halaman Register Account

Gambar 5.3 dan Gambar 5.4 menggambarkan hasil implementasi dari halaman register. Halaman ini digunakan untuk melakukan pilihan register terhadap pengguna yang ingin mendaftar member account.



Gambar 5.3 Implementasi Register Account



Gambar 5.4 Implementasi Register Account Mahasiswa

3. Halaman Login

Gambar 5.5 menggambarkan hasil implementasi dari halaman login. Halaman ini digunakan untuk melakukan login ke sistem.

Gambar 5.5 Implementasi Login

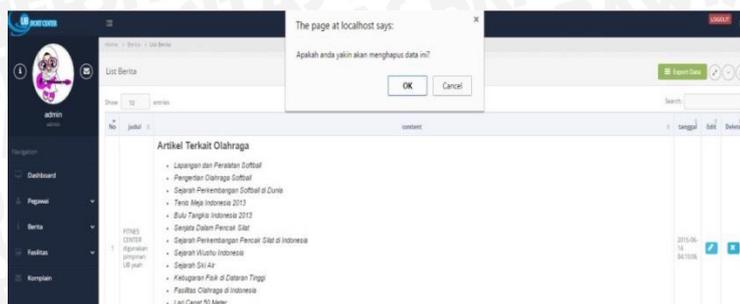
4. Halaman Pemesanan Tempat

Gambar 5.6 menggambarkan hasil implementasi dari halaman futsal. Halaman ini digunakan untuk melakukan pemesanan tempat olahraga futsal ke sistem.

Gambar 5.6 Implementasi Pemesanan Tempat

5. Halaman Berita Tampilan Menghapus Berita

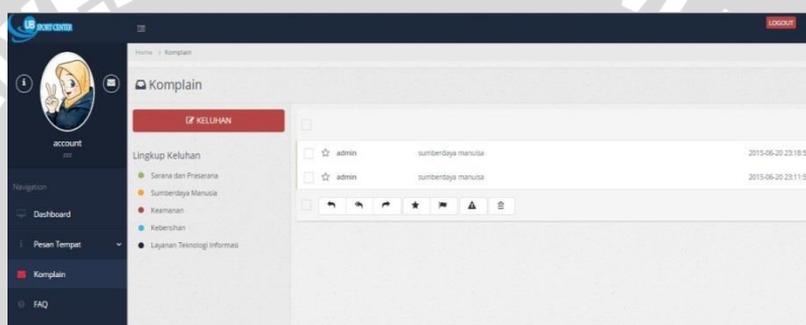
Gambar 5.7 menggambarkan hasil implementasi dari halaman listberita. Halaman ini digunakan untuk menghapus berita.



Gambar 5.7 Implementasi Menghapus Berita

6. Halaman Membuat Komplain

Gambar 5.8 menggambarkan hasil implementasi dari halaman komplain. Halaman ini digunakan untuk melakukan complain.

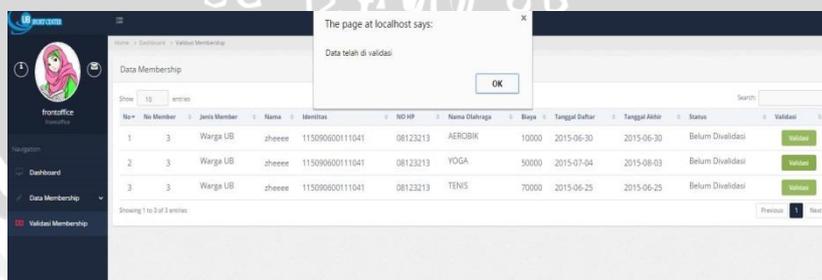


Gambar 5.8 Implementasi Membuat Komplain

5.1.5.2 Implementasi Prototype Versi 1.0

1. Halaman Validasi Membership

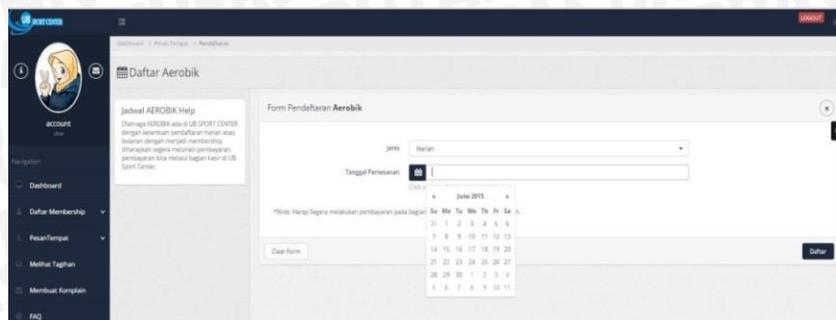
Gambar 5.9 menggambarkan hasil implementasi dari halaman validasi membership. Halaman ini merupakan hasil dari iterasi pertama dan digunakan untuk memvalidasi membership pada actor front office.



Gambar 5.9 Implementasi Validasi Membership

2. Halaman Mendaftar Membership

Gambar 5.10. menggambarkan pendaftaran membership berdasarkan olahraga dan form pendaftaran yang dilakukan member account.



Gambar 5.10 Implementasi Mendaftar Membership Aerobik

3. Halaman Mencetak Struk Membership

Gambar 5.11. menggambarkan halaman cetak struk membership.

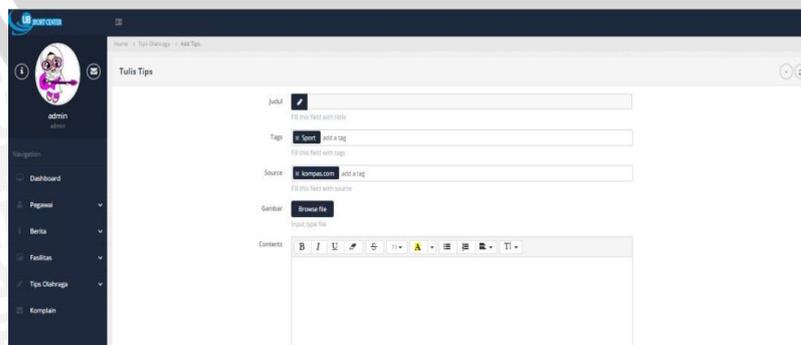
 Sport Club House Pertamina-UB LOCATION BASED ON: Jl Cibogo No 1 Malang		ID MEMBER : 3 ID MEMBERSHIP : 26
DATE OF REGISTER : 2015-06-30		DATE : 2015-06-30
NAME : Fauziah zhie		
No Identitas : 115090600111041		Tipe Identitas : Warga UB
DESCRIPTION OF MEMBER : Telepon= 08123213 Alamat Malang= Jl Veteran 6B Malang Alamat Asal= Probolinggo Kota		
SPORT	PRICE OF SPORT	TTD FO
FITNESS	Rp. 15.000,00	

Gambar 5.11 Implementasi Mencetak Struk Membership

5.1.5.3 Implementasi Prototype Versi 2.0

1. Halaman Menambah Tips

Gambar 5.12 menggambarkan hasil implementasi halaman tambah tips. Halaman ini merupakan hasil dari iterasi kedua. Halaman ini digunakan untuk menambah tips olahraga.

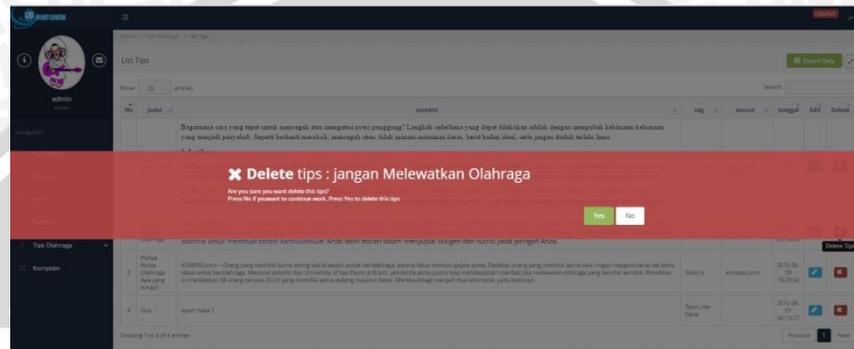


Gambar 5.12 Implementasi Menambah Tips

2. Halaman Tips Tampilan Menghapus Tips

Gambar 5.13 menggambarkan hasil implementasi halaman list tips.

Halaman ini merupakan hasil dari iterasi kedua dimana semua proses untuk menghapus data dengan menampilkan popup dan audio ‘fail’ beserta nama data yang akan dihapus.



Gambar 5.13 Implementasi Menghapus Tips

3. Halaman Home versi 2.0

Gambar 5.14 menggambarkan hasil implementasi halaman index.

Halaman ini merupakan hasil dari iterasi kedua dimana terdapat penambahan menu tips olahraga.

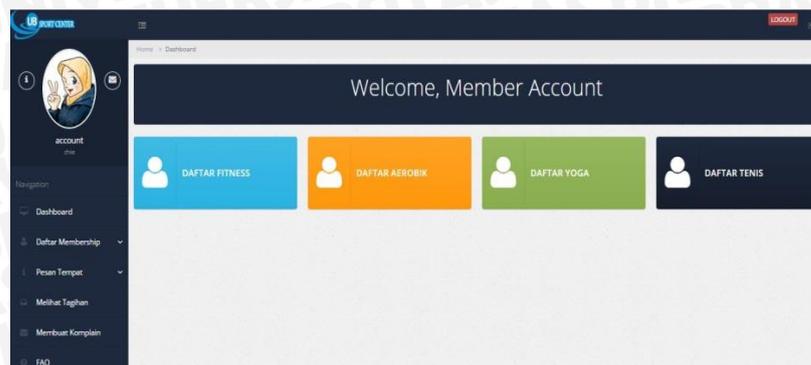


Gambar 5.14 Implementasi halaman home versi 2.0

4. Halaman Utama Member Account versi 2.0

Gambar 5.16 merupakan hasil dari iterasi kedua. Pada iterasi-iterasi tersebut terdapat penambahan menu mendaftar membership, melihat riwayat membership dan mencetak struk membership.





Gambar 5.16 Implementasi halaman member versi 2.0

5. Halaman Utama Front Office versi 2.0

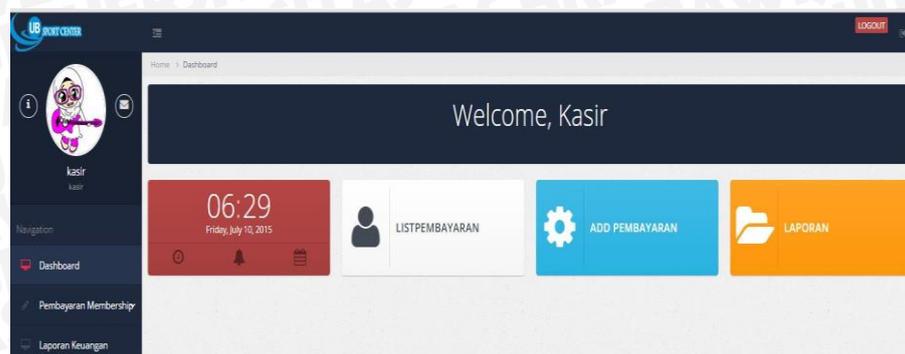
Gambar 5.17 merupakan hasil dari iterasi kedua. Pada iterasi-iterasi tersebut terdapat penambahan menu yaitu validasi membership.

No	No Member	No Membership	Jenis Member	Nama	Identitas	NO HP	Nama Olahraga	Biaya	Tanggal Daftar	Tanggal Akhir	Status	Validasi
1	3	45	Warga UB	Fauziah zhe	115090600111041	08123213	FITNES	15000	2015-07-31	2015-07-31	Belum Divalidasi	Validasi
2	3	44	Warga UB	Fauziah zhe	115090600111041	08123213	FITNES	0	2015-07-01	2015-07-01	Belum Divalidasi	Validasi
3	3	43	Warga UB	Fauziah zhe	115090600111041	08123213	AEROBIK	60000	2015-07-01	2015-07-31	Belum Divalidasi	Validasi
4	3	36	Warga UB	Fauziah zhe	115090600111041	08123213	FITNES	90000	2015-07-29	2015-06-28	Belum Divalidasi	Validasi
5	3	35	Warga UB	Fauziah zhe	115090600111041	08123213	FITNES	80000	2015-07-02	2015-08-01	Belum Divalidasi	Validasi
6	3	9	Warga UB	Fauziah zhe	115090600111041	08123213	AEROBIK	10000	2015-06-30	2015-06-30	Belum Divalidasi	Validasi

Gambar 5.17 Implementasi halaman front office versi 2.0

6. Halaman Utama Kasir versi 2.0

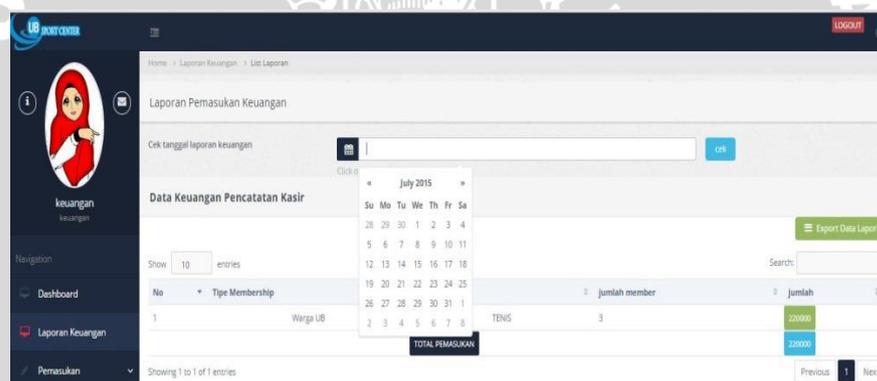
5.18 merupakan hasil dari iterasi kedua. Pada iterasi-iterasi tersebut terdapat perubahan yakni pada laporan kasir yang dicek berdasarkan tanggal.



Gambar 5.18 Implementasi halaman kasir versi 2.0

7. Halaman Utama Keuangan versi 2.0

Gambar 5.20 merupakan hasil dari iterasi kedua. Pada iterasi-iterasi tersebut terdapat perubahan yakni pada laporan keuangan yang dicek berdasarkan tanggal.



Gambar 5.20 Implementasi halaman keuangan versi 2.0

5.2 Fase Evaluasi – Rilis Prototype

Pada tahap proses evaluasi prototype sistem ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah rilis sistem akan diambil. Jika tidak, maka *prototyping* direvisi dan memberi *feedback* dengan mengulang dari fase analisa kebutuhan pengguna. Dalam kasus ini dibatasi hanya dilakukan sebanyak 2 kali, sesuai dengan batasan yang telah ditetapkan.

5.2.1 Evaluasi Prototype Versi 0.0

Prototype versi 1.0 dilakukan setelah dilakukan evaluasi dari protitipe versi 0.0 atau iterasi 1. Hasil dari evaluasi prototype versi 1.0 adalah sebagai berikut.

1. Menambah Aktor bendahara
2. Menambah fitur melihat laporan keuangan pada aktor bendahara
3. Fitur Harga tidak perlu dibuat pada Aktor Admin
4. Aktor Front Office dapat melakukan validasi membership
5. Aktor Member account dapat mendaftar membership dan melihat riwayat membership serta dapat mencetak struk membership.
6. Penambahan cek tanggal pada laporan pemasukan, laporan pembayaran dan laporan keuangan.

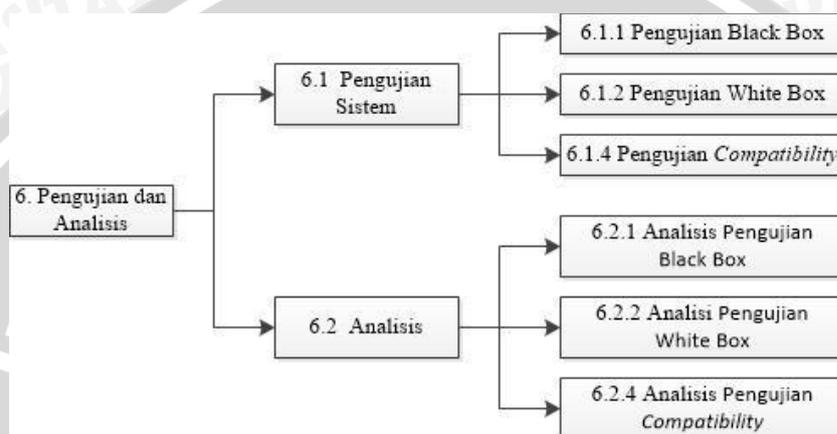
5.2.2 Evaluasi Prototype Versi 1.0

Prototipe versi 2.0 dilakukan setelah dilakukan evaluasi dari protitipe versi 1.0 atau iterasi 2. Hasil dari evaluasi tersebut adalah sebagai berikut.

1. Aktor Admin dapat mengelola tips olahraga
2. Actor user dapat melihat tips olahraga
3. Merubah tampilan popup pada semua proses hapus

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas tahapan Pengujian dan Analisis Sistem Informasi Sport Club House Pertamina - UB. Pembahasan terdiri atas penjelasan tentang metode pengujian yang digunakan beserta hasil pengujiannya dan analisa dari hasil pengujian.



Gambar 6.1 Diagram Blok Bab Pengujian dan Analisis

6.1 Pengujian Sistem

Pengujian dilakukan dengan tiga tahapan yaitu pengujian black box, pengujian white box dan *Compatibility*. Seluruh proses pengujian ini dilakukan dengan menggunakan spesifikasi perangkat keras dan perangkat lunak sesuai dengan sub bab 5.1 yang telah dijelaskan sebelumnya. Pengujian Black box dilakukan dengan menguji satu persatu fitur yang ada sesuai dengan kebutuhan fungsional, pengujian white box dilakukan dengan melihat ke dalam modul untuk meneliti kode-kode program dan menganalisis ada kesalahan atau tidak dan pengujian *compatibility* dilakukan dengan menguji sistem pada sistem operasi yang berbeda.

6.1.1 Pengujian White Box

Pengujian *white box* dilakukan dengan pengujian *Basis Path*. Pengujian *Basis Path* ditunjukkan pada gambar 6.2 hingga gambar 6.4.

6.1.1.1 Tujuan

Tujuan dilakukannya pengujian *white box* adalah untuk menguji semua *statement* program pada beberapa algoritma yang memiliki prioritas *high* yang telah diimplementasikan sesuai dengan yang diharapkan.

6.1.1.2 Mekanisme

Pengujian *white box* dilakukan dengan membuat *flowgraph* dari *pseudo code* dilanjutkan dengan menghitung nilai *cyclomatic complexity* dan menentukan jalur independen *path* serta memaparkan kasus ujinya. Pengujian *white box* ini dilakukan pada tiga fitur yaitu login, mendaftar membership dan register account.

6.1.1.3 Hasil Pengujian

a. Kasus Uji Login

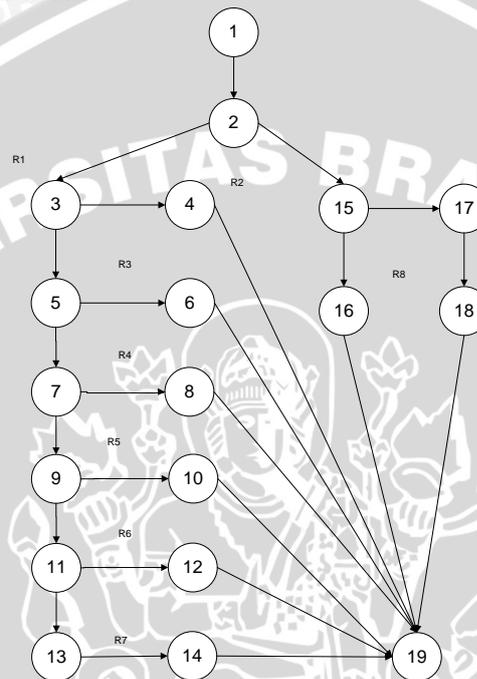
Berikut adalah *pseudo code* login ini yang nantinya digunakan untuk mencari *flowgraph* dan melanjutkan pada tahap berikutnya. ditunjukkan pada Tabel 6.1 berikut :

Tabel 6.1 Pseudo code Login

1.	Inputan username & password
2.	If (\$login) == 1
3.	If role==admin
4.	Admin login
5.	Else if role==kasir
6.	kasir login
7.	Else if role==frontoffice
8.	frontoffice login
9.	Else if role==gedung
10.	gedung login
11.	Else if role==bendahara
12.	bendahara login
13.	Else if role==keuangan
14.	keuangan login
15.	Else if(\$login)==2

16.	Member login
17.	Else
18.	invalid username atau password
19.	END

Flowgraph dari pseudo code kasus uji kasus login akan ditunjukkan pada gambar 6.2 berikut.



Gambar 6.2 Flowgraph Login

Pada Gambar 6.2 digambarkan *flowgraph* dari login yang didapatkan dari *pseudo code* yang telah dijabarkan. Berikut adalah penjelasan dari setiap node pada *flowgraph* tersebut :

1. Node 1 adalah menampilkan form inputan username dan password
2. Node 2 adalah node kondisi “IF “ login sama dengan satu maka ke node 3 dan jika tidak ke ke node 15
3. Node 3 adalah kondisi case jika role==admin maka ke node 4
4. Node 5 adalah kondisi jika role==kasir maka ke node 6
5. Node 7 adalah kondisi jika role==frontoffice ke node 8
6. Node 9 adalah kondisi jika role== gedung maka ke node 10
7. Node 11 adalah kondisi jika role==bendahara maka ke node 12
8. Node 13 adalah kondisi jika role == keuangan maka ke node 14

9. Node 15 adalah node kondisi “IF “ login sama dengan dua maka ke node 16 dan jika tidak ke ke node 17
10. Node 18 menampilkan informasi password atau username salah
11. Node 19 adalah end

Dari *flowgraph* login yang sudah dijelaskan pada Gambar 6.2 dapat dihitung nilai *cyclomatic complexity* sebagai berikut :

$$\begin{aligned}
 V(G) &= 8 \text{ regions} \\
 V(G) &= E - N + 2 \\
 &= 25 - 19 + 2 \\
 &= 8 \\
 V(G) &= P + 1 \\
 &= 7 + 1 \\
 &= 8
 \end{aligned}$$

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 8 jalur independen yaitu:

1. Jalur 1 : 1 – 2 – 3 – 4 – 19
2. Jalur 2 : 1 – 2 – 3 – 5 – 6 – 19
3. Jalur 3 : 1 – 2 – 3 – 5 – 7 – 8 – 19
4. Jalur 4 : 1 – 2 – 3 – 5 – 7 – 9 – 10 – 19
5. Jalur 5 : 1 – 2 – 3 – 5 – 7 – 9 – 11 – 12 – 19
6. Jalur 6 : 1 – 2 – 3 – 5 – 7 – 9 – 11 – 13 – 14 – 19
7. Jalur 7 : 1 – 2 – 15 – 16 – 19
8. Jalur 8 : 1 – 2 – 15 – 17 – 18 – 19

Berdasarkan 8 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.2 memaparkan kasus uji dari *login*

Tabel 6.2 Kasus Uji *Login*

No	Jalur	Kelas	Method	Data Input	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	1 – 2 – 3 – 4 – 19	Main_controller	Dologin()	Username, password	Sistem menampilkan halaman	Sistem menampilkan halaman admin	Valid

					admin jika role =admin	jika role =admin	
2	1 – 2 – 3 – 5 – 6 – 19	Main_controller	Dologin()	Username, password	Sistem menampilkan halaman kasir jika role =kasir	Sistem menampilkan halaman kasir jika role =kasir	Valid
3	1 – 2 – 3 – 5 – 7 – 8 – 19	Main_controller	Dologin()	Username, password	Sistem menampilkan halaman frontoffice jika role =frontoffice	Sistem menampilkan halaman frontoffice jika role =frontoffice	Valid
4	1 – 2 – 3 – 5 – 7 – 9 – 10 – 19	Main_controller	Dologin()	Username, password	Sistem menampilkan halaman gedung jika role =gedung	Sistem menampilkan halaman gedung jika role =gedung	Valid
5	1. – 1 – 2 – 3 – 5 – 7 – 9 – 11 – 12 – 19	Main_controller	Dologin()	Username, password	Sistem menampilkan halaman bendahara jika role =bendahara	Sistem menampilkan halaman bendahara jika role =bendahara	Valid
6	1 – 2 – 3 – 5 – 7 – 9 – 11 – 13 – 14 – 19	Main_controller	Dologin()	Username, password	Sistem menampilkan halaman keuangan jika role =keuangan	Sistem menampilkan halaman keuangan jika role =keuangan	Valid
7	1 – 2 – 15 – 16 – 19	Main_controller	Dologin()	Username, password	Sistem menampilkan halaman member jika login=2	Sistem menampilkan halaman member jika login=2	Valid
8	1 – 2 – 15 – 17 – 18 – 19	Main_controller	Dologin()	Username, password	Sistem memberi peringatan data input tidak sesuai	Sistem memberi peringatan data input tidak sesuai	Valid

b. Kasus Uji Mendaftar Membership

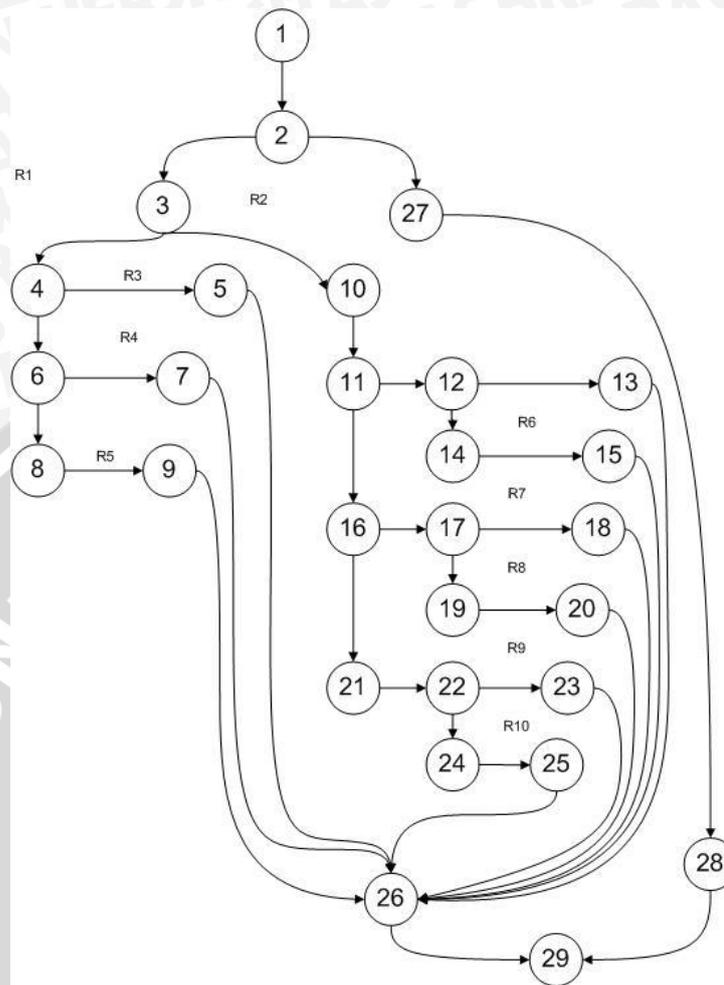
Berikut adalah pseudo code pada kasus uji Mendaftar Membership

Fitness ini yang nantinya digunakan untuk mencari flowgraph dan melanjutkan pada tahap berikutnya dijelaskan pada tabel 6.3.

Tabel 6.3 Pseudo code Mendaftar Membership

1.	Input \$tanggal, \$idtype, \$akhir, \$expired
2.	if(\$tanggal >= \$now)
3.	if(\$jenis==1)
4.	if(\$idtype==1)
5.	\$biaya = 10000;
6.	else if(\$idtype==2)
7.	\$biaya = 15000;
8.	else
9.	\$biaya = 20000
10.	else
11.	if(\$idtype==1)
12.	if(\$tanggal >= \$akhir && \$tanggal <= \$expired)
13.	\$biaya = 65000;
14.	else
15.	\$biaya = 75000;
16.	else if(\$idtype==2)
17.	if(\$tanggal >= \$akhir && \$tanggal <= \$expired)
18.	\$biaya = 80000;
19.	else
20.	\$biaya = 90000;
21.	else
22.	if(\$tanggal >= \$akhir && \$tanggal <= \$expired)
23.	\$biaya = 150000;
24.	else
25.	\$biaya = 170000;
26.	Print "Pendaftaran berhasil, harap segera melakukan pembayaran"
27.	else
28.	print "Pendaftaran gagal disimpan"
29.	END

Flowgraph dari pseudo code kasus uji kasus daftar membership akan ditunjukkan pada gambar 6.3



Gambar 6.3 Flowgraph daftar membership

Dari flowgraph yang sudah dijelaskan pada Gambar 6.3 juga dapat dihitung nilai cyclomatic complexity sebagai berikut :

$$\begin{aligned}
 V(G) &= 10 \text{ regions} \\
 V(G) &= E - N + 2 \\
 &= 37 - 29 + 2 \\
 &= 10 \\
 V(G) &= P + 1 \\
 &= 9 + 1 \\
 &= 10
 \end{aligned}$$

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 10 jalur independen yaitu:

1. Jalur 1 : 1 - 2 - 3 - 4 - 5 - 26 - 29

2. Jalur 2 : 1 – 2 – 3 – 4 – 6 – 7 – 26 – 29
3. Jalur 3 : 1 – 2 – 3 – 4 – 6 – 8 – 9 – 26 – 29
4. Jalur 4 : 1 – 2 – 3 – 10 – 11 – 12 – 13 – 26 – 29
5. Jalur 5 : 1 – 2 – 3 – 10 – 11 – 12 – 14 – 15 – 26 – 29
6. Jalur 6 : 1 – 2 – 3 – 10 – 11 – 16 – 17 – 18 – 26 – 29
7. Jalur 7 : 1 – 2 – 3 – 10 – 11 – 16 – 17 – 19 – 20 – 26 – 29
8. Jalur 8 : 1 – 2 – 3 – 10 – 11 – 16 – 21 – 22 – 23 – 26 – 29
9. Jalur 9 : 1 – 2 – 3 – 10 – 11 – 16 – 21 – 22 – 24 – 25 – 26 – 29
10. Jalur 10 : 1 – 2 – 27 – 28 – 29

Berdasarkan 10 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.4 memaparkan kasus uji dari mendaftar membership.

Tabel 6.4 Kasus Uji mendaftar membership

no	Jalur	Kelas	Method	Data Input	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	1 – 2 – 3 – 4 – 5 – 26 – 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	Sistem memproses biaya jika jenis=1 dan tipe=1	Sistem memproses biaya jika jenis=1 dan tipe=1	Valid
2	1 – 2 – 3 – 4 – 6 – 7 – 26 – 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	Sistem memproses biaya jika jenis=1 dan tipe=2	Sistem memproses biaya jika jenis=1 dan tipe=2	Valid
3	1. 1 – 2 – 3 – 4 – 6 – 8 – 9 – 26 – 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	Sistem memproses biaya jika jenis=1 dan tipe=3	Sistem memproses biaya jika jenis=1 dan tipe=3	Valid
4	1 – 2 – 3 – 10 – 11 – 12 – 13 – 26 – 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	Sistem memproses biaya jika jenis=2, tipe=1 dan tanggal sesuai	Sistem memproses biaya jika jenis=2, tipe=1 dan tanggal sesuai	Valid
5	1 – 2 – 3 – 10 – 11 – 12 – 14 – 15 – 26 – 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now,	Sistem memproses biaya jika jenis=2, tipe=1 dan tanggal	Sistem memproses biaya jika jenis=2, tipe=1 dan tanggal	Valid

6	1 - 2 - 3 - 10 - 11 - 16 - 17 - 18 - 26 - 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	tidak sesuai Sistem memproses biaya jika jenis=2, tipe=2 dan tanggal sesuai	tidak sesuai Sistem memproses biaya jika jenis=2, tipe=2 dan tanggal sesuai	Valid
7	1 - 2 - 3 - 10 - 11 - 16 - 17 - 19 - 20 - 26 - 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	Sistem memproses biaya jika jenis=2, tipe=2 dan tanggal tidak sesuai	Sistem memproses biaya jika jenis=2, tipe=2 dan tanggal tidak sesuai	Valid
8	1 - 2 - 3 - 10 - 11 - 16 - 21 - 22 - 23 - 26 - 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	Sistem memproses biaya jika jenis=2, tipe=3 dan tanggal sesuai	Sistem memproses biaya jika jenis=2, tipe=3 dan tanggal sesuai	Valid
9	1. 1 - 2 - 3 - 10 - 11 - 16 - 21 - 22 - 24 - 25 - 26 - 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	Sistem memproses biaya jika jenis=2, tipe=3 dan tanggal tidak sesuai	Sistem memproses biaya jika jenis=2, tipe=3 dan tanggal tidak sesuai	Valid
10	1 - 2 - 27 - 28 - 29	c_member	daftar_fitnes ()	Idmember, idtype, expired, tanggal, jenis, now, akhir	Sistem menampilkan pesan kesalahan karena gagal disimpan	Sistem menampilkan pesan kesalahan karena gagal disimpan	Valid

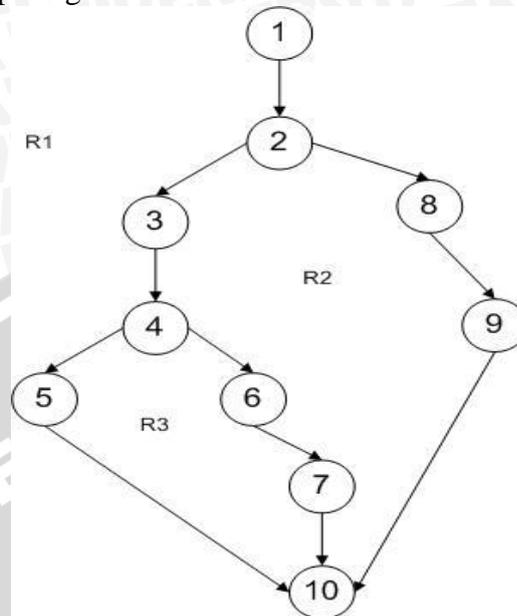
a. Kasus Uji Register Account

Berikut adalah *pseudo code* pada kasus uji register account ini yang nantinya digunakan untuk mencari flowgraph dan melanjutkan pada tahap berikutnya dijelaskan pada tabel 6.5 berikut.

Tabel 6.5 Pseudo code register account

1.	Inputan registrasi
2.	if(\$fileSize > 0 \$fileError == 0)
3.	move upload file
4.	if(move)
5.	data has been upload
6.	else
7.	data failed to upload
8.	else
9.	failed to upload + nama file
10.	End

Flowgraph dari pseudo code kasus uji kasus register account akan ditunjukkan pada gambar 6.4



Gambar 6.4 Flowgraph register account

Pada Gambar 6.4 digambarkan *flowgraph* yang didapatkan dari *pseudo code* yang telah dijabarkan. Berikut adalah penjelasan dari setiap node pada *flowgraph* tersebut :

1. Node 1 adalah menampilkan form inputan registrasi
2. Node 2 adalah node kondisi “IF “ filesize sama dengan nol maka ke node 3 dan jika tidak ke ke node 7
3. Node 3 adalah menampilkan bahwa file dipindah
4. Node 4 adalah kondisi jika dipindah
5. Node 5 adalah menampilkan data berhasil di upload
6. Node 6 adalah kondisi else
7. Node 7 kondisi menampilkan pesan data failed
8. Node 8 adalah kondisi else
9. Node 9 menampilkan pesan failed to upload + nama file
10. Node 10 end

Dari *flowgraph* yang sudah dijelaskan pada Gambar 6.4 juga dapat dihitung nilai *cyclomatic clomplexity* sebagai berikut :

$$V(G) = 3 \text{ regions}$$

$$V(G) = E - N + 2$$

$$= 11 - 10 + 2$$

$$= 3$$

$$V(G) = P + 1$$

$$= 2 + 1$$

$$= 3$$

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 3 jalur independen yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 10

Jalur 2 : 1 – 2 – 3 – 4 – 6 – 7 – 10

Jalur 3 : 1 – 2 – 8 – 9 – 10

Berdasarkan 3 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.6 memaparkan kasus uji dari register account

Tabel 6.6 Kasus Uji register account

No	Jalur	Kelas	Method	Data Input	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	1 – 2 – 3 – 4 – 5 – 10	c_user	addmemberaction()	foto, username, password, identitas, nama, alamatmalang, alamatasal, idfakultas, idunit, notelepon, ttl, email, idtipe.	Foto berhasil di upload jika filesize <= nol dan fileError =nol	Foto berhasil di upload jika filesize <= nol dan fileError =nol	Valid
2	1 – 2 – 3 – 4 – 6 – 7 – 10	C_user	addmemberaction()	foto, username, password, identitas, nama, alamatmalang, alamatasal, idfakultas, idunit, notelepon, ttl, email, idtipe.	Foto gagal di upload jika tidak move	Foto gagal di upload jika tidak move	Valid
3	1 – 2 – 8 – 9 – 10	c_user	addmemberaction()	foto, username, password, identitas, nama, alamatmalang, alamatasal, idfakultas, idunit, notelepon, ttl, email, idtipe.	Foto gagal di upload jika filesize = nol dan fileError !=nol	Foto gagal di upload jika filesize = nol dan fileError !=nol	Valid

6.1.2 Pengujian Validasi dengan Metode *Black Box*

6.1.2.1 Tujuan

Tujuan dilakukannya pengujian validasi adalah untuk memastikan setiap spesifikasi kebutuhan fungsional yang didefinisikan telah sesuai dengan yang diharapkan dengan menguji seluruh skenario *use case* yang telah dibuat pada bab dari perancangan sebelumnya.

6.1.2.2 Mekanisme

Setiap scenario *use case* yang didefinisikan pada tahap analisis kebutuhan akan diujikan pada pengujian ini dengan cara mendefinisikan kasus uji dan hasil yang didapat terhadap setiap kebutuhan tersebut.

6.1.2.3 Hasil Pengujian

Pengujian *black box* secara validasi dijelaskan pada Tabel 6.7 berikut.

Tabel 6.7 Pengujian Black Box Sistem Informasi UB Sport

KODE	FUNGSI	KASUS UJI	YANG DIHARAPKAN	HASIL YANG DIDAPAT	STATUS
KF_01	Login	<ul style="list-style-type: none">- membuka sistem dan pilih menu 'login'- Input Username dan password- menekan tombol 'login'	<ul style="list-style-type: none">- Sistem dapat mengecek kebenaran input data user di database sehingga user dapat masuk ke dalam sistem.	Sistem mengecek kebenaran input data user di database sehingga user dapat masuk ke dalam sistem.	Valid
		<p>Alternative 1</p> <ul style="list-style-type: none">- membuka sistem dan pilih menu 'login'- tidak mengisi salah satu kolom input- menekan tombol 'login'	<ul style="list-style-type: none">- sistem dapat menampilkan pesan 'fill this field'.- sistem menampilkan pesan	<ul style="list-style-type: none">- sistem menampilkan pesan 'fill this field'.- sistem menampilkan pesan invalid username dan password.	

		Alternative 2 - membuka sistem dan pilih menu 'login' - input username dan password yang salah - menekan tombol 'login'	invalid username dan password.		
KF_02	Melihat Berita	-membuka sistem -memilih menu "News" pada sistem	Menampilkan data berita yang ada	Menampilkan data berita yang ada	Valid
KF_03	Melihat Tips Olahraga	-membuka sistem -memilih menu "Tips Sport" pada sistem	Menampilkan data tips olahraga yang ada	Menampilkan data tips olahraga yang ada	Valid
KF_04	Melihat Fasilitas	-membuka sistem -memilih menu "Facilities" pada sistem	Menampilkan data fasilitas yang ada	Menampilkan data fasilitas yang ada	Valid
KF_05	Register Account	-membuka sistem -memilih menu "Register" pada sistem -menampilkan button pilihan identitas member dan pilih salah Satu -menampilkan halaman form register -isi data register dan pilih button 'register'	-Sistem dapat menampilkan pesan "pendaftaran berhasil" dan data masuk ke dalam database - sistem dapat menampilkan pesan 'fill this field'.	-Sistem menampilkan pesan "pendaftaran berhasil" dan data masuk ke dalam database. - sistem dapat menampilkan pesan 'fill this field'.	Valid
KF_06	Pemesanan Tempat	Alternative 1 -membuka sistem -memilih menu "Register" pada sistem -menampilkan button pilihan identitas member dan pilih salah Satu -menampilkan halaman form register -tidak mengisi salah satu kolom pilih button 'register'	-Sistem dapat menampilkan pesan "pemesanan berhasil" dan data masuk ke dalam database	-Sistem menampilkan pesan "pemesanan berhasil" dan data masuk ke dalam database	Valid

		<ul style="list-style-type: none"> -menampilkan widget olahraga -memilih salah satu olahraga maka muncul jadwal olahraga dan pilih salah satu jadwal -isi data dan pilih button 'pesan' <p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai member account -memilih menu "Daftar Membership" -memilih submenu "Pesan Tempat" -menampilkan widget olahraga -memilih salah satu olahraga maka muncul jadwal olahraga dan pilih salah satu jadwal -salah input data dan pilih button 'pesan' 	database		
				- sistem menampilkan pesan gagal karena salah input tanggal atau jam	
				- sistem dapat menampilkan pesan gagal karena salah input tanggal atau jam	
KF_07	Melihat FAQ	<ul style="list-style-type: none"> -login ke sistem sebagai member account -memilih menu "FAQ" 	-Sistem dapat menampilkan halaman FAQ	-Sistem dapat menampilkan halaman FAQ	Valid
KF_08	Cetak struk pemesanan	<ul style="list-style-type: none"> -login ke sistem sebagai member account -memilih menu "Lihat Riwayat Pemesanan" -memilih data dengan status 'belum divalidasi' yang akan di cetak dan pilih icon "print" -login ke sistem sebagai member account -memilih menu "Lihat Riwayat Pemesanan" -memilih data dengan status bukan 'belum divalidasi' dan pilih icon "print" 	<ul style="list-style-type: none"> -Sistem dapat menampilkan halaman cetak struk pemesanan --Sistem dapat menampilkan halaman cetak kosong 	<ul style="list-style-type: none"> -Sistem dapat menampilkan halaman cetak struk -Sistem dapat menampilkan halaman cetak kosong 	Valid
KF_09	Melihat Riwayat Pemesanan	<ul style="list-style-type: none"> -login ke sistem sebagai member account -memilih menu "Pesan Tempat" pada sistem -memilih submenu "Lihat Riwayat Pemesanan" 	Sistem dapat menampilkan halaman riwayat pemesanan	Sistem menampilkan halaman riwayat pemesanan	Valid

<p>KF_10</p>	<p>Cetak struk Membership</p>	<p>-login ke sistem sebagai member account -memilih menu “Lihat Riwayat Membership” -memilih data dengan status ‘belum divalidasi’ yang akan di cetak dan pilih icon “print”</p> <p>-login ke sistem sebagai member account -memilih menu “Lihat Riwayat Membership” -memilih data dengan status bukan ‘belum divalidasi’ dan pilih icon “print”</p>	<p>-Sistem dapat menampilkan halaman cetak struk membership --Sistem dapat menampilkan halaman cetak kosong</p>	<p>-Sistem dapat menampilkan halaman cetak struk membership -Sistem dapat menampilkan halaman cetak kosong</p>	<p>Valid</p>
<p>KF_11</p>	<p>Mendaftar Membership</p>	<p>-login ke sistem sebagai member account -memilih menu “Daftar Membership” -memilih submenu “Daftar Membership” -menampilkan widget olahraga -memilih salah satu olahraga maka muncul form tambah . mengisi data dan pilih “daftar”</p> <p>Alternative 1 -login ke sistem sebagai member account -memilih menu “Daftar Membership” -memilih submenu “Daftar Membership” -menampilkan widget olahraga -memilih salah satu olahraga maka muncul form tambah . - salah mengisi data tanggal dan pilih “daftar”</p>	<p>-sistem dapat menampilkan pesan ‘daftar berhasil ditambahkan’ dan Dapat menambah data ke dalam database</p> <p>- sistem dapat menampilkan pesan gagal disimpan</p>	<p>-sistem dapat menampilkan pesan ‘daftar berhasil ditambahkan’ dan Dapat menambah data ke dalam database</p> <p>- sistem menampilkan pesan gagal disimpan.</p>	<p>Valid</p>
<p>KF_12</p>	<p>Melihat Riwayat Membership</p>	<p>-login ke sistem sebagai member account -memilih menu “Daftar Membership” -memilih submenu “Lihat Riwayat Membership”</p>	<p>Sistem dapat menampilkan halaman riwayat membership</p>	<p>Sistem menampilkan halaman riwayat membership</p>	<p>Valid</p>
<p>KF_13</p>	<p>Membuat Komplain</p>	<p>-login ke sistem sebagai member account -memilih menu “Membuat Komplain” pada sistem</p>	<p>-sistem dapat menampilkan pesan ‘komplain berhasil</p>	<p>-sistem menampilkan pesan ‘komplain berhasil</p>	<p>Valid</p>

		<p>-memilih button “Keluhan” menampilkan form komplain -mengisi data form -memilih tombol “Send Message”</p> <p>Alternative 1 -login ke sistem sebagai member account -memilih menu “Komplain” pada sistem -memilih button “Keluhan” menampilkan form tambah komplain -tidak mengisi salah satu data form -memilih tombol “Send Message”</p>	<p>ditambahkan’ dan Dapat menambah data ke dalam database</p> <p>- sistem dapat menampilkan pesan ‘fill this field’.</p>	<p>ditambahkan’ dan Dapat menambah data ke dalam database.</p> <p>- sistem menampilkan pesan ‘fill this field’.</p>	
KF_14	Melihat Tagihan	<p>-login ke sistem sebagai member account -memilih menu “Melihat Tagihan”</p>	<p>-sistem dapat menampilkan halaman lihat tagihan member</p>	<p>-sistem menampilkan halaman lihat tagihan member</p>	Valid
KF_15	Menambah Membership	<p>-login ke sistem sebagai front office -memilih menu “Data Membership” -memilih submenu “Add Membership” menampilkan form tambah lalu isi data -memilih tombol “Daftar”</p> <p>Alternative 1 -login ke sistem sebagai front office -memilih menu “Data Membership” -memilih submenu “Add Membership” menampilkan form tambah -lalu tidak mengisi salah satu field -memilih tombol “Daftar”</p>	<p>-sistem dapat menampilkan pesan ‘membership berhasil ditambahkan’ dan Dapat menambah data ke dalam database</p> <p>- sistem dapat menampilkan pesan ‘fill this field’.</p>	<p>-sistem menampilkan pesan ‘membership berhasil ditambahkan’ dan Dapat menambah data ke dalam database.</p> <p>- sistem menampilkan pesan ‘fill this field’.</p>	Valid
KF_16	Melihat data Membership	<p>-login ke sistem sebagai front office -memilih menu “Data Membership”</p>	<p>-sistem dapat menampilkan halaman membership</p>	<p>-sistem menampilkan halaman membership</p>	Valid

		-memilih submenu “List Membership”			
KF_17	Menghapus Membership	-login ke sistem sebagai front office -memilih menu “Data Membership” -memilih submenu “List Membership” memilih data yang ingin dihapus lalu memilih icon “delete”	-Sistem dapat data membership yang diinginkan	-sistem mengubah data membership yang diinginkan	Valid
KF_18	Validasi Membership	-login ke sistem sebagai front office -memilih menu “Validasi Membership” -menampilkan halaman form validasi -memilih data yang ingin divalidasi -pilih icon “validasi”	-sistem dapat menampilkan pesan ‘data berhasil divalidasi’ dan status data berhasil dirubah di database	-sistem dapat menampilkan pesan ‘data berhasil divalidasi’ dan status data berhasil dirubah di database	Valid
KF_19	Menambah Pembayaran	-login ke sistem sebagai kasir -memilih menu “Pembayaran Membership” -memilih submenu “Add Pembayaran” -input data id membership -memilih tombol “Submit” maka akan muncul data member tersebut Alternative 1 -login ke sistem sebagai kasir -memilih menu “Pembayaran Membership” -memilih submenu “Add Pembayaran” -input data id membership yang salah -memilih tombol “Submit”	-sistem dapat menampilkan pesan ‘pembayaran berhasil ditambahkan’ dan Dapat menambah data ke dalam database - sistem dapat menampilkan pesan ‘fill this field’.	-sistem menampilkan pesan ‘pembayaran berhasil ditambahkan’ dan Dapat menambah data ke dalam database. - sistem menampilkan pesan ‘fill this field’.	Valid
KF_20	Menghapus Pembayaran	-login ke sistem sebagai kasir -memilih menu “Pembayaran Membership” -memilih submenu “List Pembayaran” memilih data yang ingin dihapus lalu memilih icon “delete”	-sistem dapat menampilkan pesan ‘pembayaran berhasil dihapus’ dan data dihapus dari database	-sistem menampilkan pesan ‘pembayaran berhasil dihapus’ dan data dihapus dari database	Valid

KF_21	Melihat Pemasukan Membership	-login ke sistem sebagai kasir -memilih menu “Laporan Keuangan” -mengecek tanggal sesuai dengan keinginan	-Sistem dapat menampilkan halaman laporan	-sistem dapat menampilkan halaman laporan	Valid
KF_22	Validasi Pemesanan	-login ke sistem sebagai bag gedung -memilih menu “Validasi” pada sistem -menampilkan halaman form validasi -memilih data yang ingin divalidasi -pilih icon “validasi”	-sistem dapat menampilkan pesan ‘data berhasil divalidasi’ dan status data berhasil dirubah di database	-sistem dapat menampilkan pesan ‘data berhasil divalidasi’ dan status data berhasil dirubah di database	Valid
KF_23	Menambah Pemesanan	-login ke sistem sebagai gedung -memilih menu “Pemesanan” pada sistem -memilih submenu “Add Pemesanan” menampilkan widget olahraga, pilih dan menampilkan form tambah lalu isi data -memilih tombol “Submit”	-sistem dapat menampilkan halaman data pesan	-sistem dapat menampilkan halaman data pesan	Valid
		Alternative 1 -login ke sistem sebagai gedung -memilih menu “Pemesanan” pada sistem -memilih submenu “Add Pemesanan” menampilkan widget olahraga, pilih dan menampilkan form tambah lalu salah isi data -memilih tombol “Submit”	-sistem dapat menampilkan pesan gagal disimpan	-sistem menampilkan pesan gagal disimpan	
KF_24	Menghapus Pemesanan	-login ke sistem sebagai gedung -memilih menu “Pemesanan” pada sistem -memilih submenu “List Pemesanan” memilih data yang ingin dihapus lalu memilih icon “delete”	-sistem dapat menampilkan pesan ‘pemasukan berhasil dihapus’ dan data dihapus dari database	-sistem menampilkan pesan ‘pemasukan berhasil dihapus’ dan data dihapus dari database	Valid
KF_25	Melihat Data Pemesan	-login ke sistem sebagai gedung -memilih menu “Pemesanan” pada sistem -memilih submenu “List Pemesan”	-Sistem dapat menampilkan halaman pemesanan	-sistem menampilkan halaman pemesanan	Valid

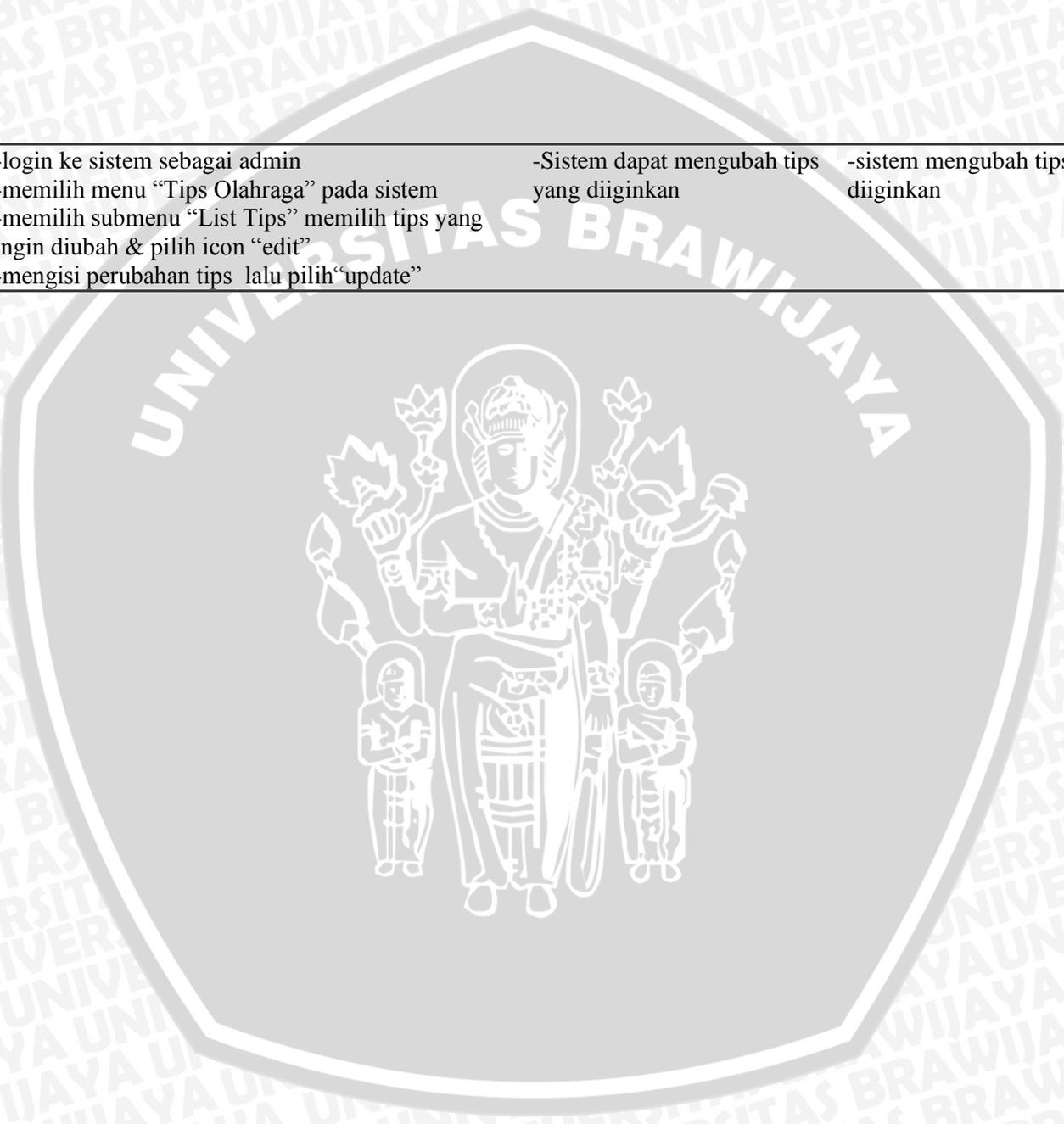
KF_26	Menambah Pemasukan	-login ke sistem sebagai gedung -memilih menu “Pemasukan” pada sistem -memilih submenu “add Pemasukan” -input no pesan dan pilih submit maka menampilkan data pemesanan. -pilih tombol bayar.	-sistem dapat menampilkan pesan ‘data berhasil ditambahkan’ dan Dapat menambah laporan ke dalam database	-sistem menampilkan pesan ‘data berhasil ditambahkan’ dan Dapat menambah laporan ke dalam database.	Valid
		Alternative 1 -login ke sistem sebagai gedung -memilih menu “Pemasukan” pada sistem -memilih submenu “add Pemasukan” -input no pesan yang salah dan pilih submit	-sistem dapat menampilkan pesan gagal karena nomer salah	-sistem menampilkan pesan gagal disimpan karena nomer salah	
KF_27	Menghapus Pemasukan	-login ke sistem sebagai gedung -memilih menu “Pemasukan” pada sistem -memilih submenu “List Pemasukan” memilih data yang ingin dihapus lalu memilih icon “delete”	-sistem dapat menampilkan pesan ‘data berhasil dihapus’ dan laporan dihapus dari database	-sistem menampilkan pesan ‘data berhasil dihapus’ dan laporan dihapus dari database	Valid
KF_28	Melihat Pemasukan Pemesanan	-login ke sistem sebagai gedung -memilih menu “Pemasukan” pada sistem -memilih submenu “List Pemasukan” -mengecek tanggal sesuai dengan keinginan	- Sistem dapat menampilkan halaman pemasukan	Sistem dapat menampilkan halaman pemasukan	Valid
KF_29	Melihat Laporan Keuangan	-login ke sistem sebagai bendahara -memilih menu “Laporan Keuangan” pada sistem -mengecek tanggal sesuai dengan keinginan	Menampilkan Laporan Keuangan	Menampilkan Laporan Keuangan	Valid
KF_30	Membalas Komplain	-login ke sistem sebagai admin -memilih menu “Komplain” pada sistem	-sistem dapat menampilkan pesan ‘berita berhasil ditambahkan’	-sistem dapat menampilkan pesan ‘berita berhasil ditambahkan’	Valid
KF_31	Menambah Berita	-login ke sistem sebagai admin -memilih menu “Berita” pada sistem -memilih submenu “Tulis Berita”	-sistem dapat menampilkan pesan ‘berita berhasil ditambahkan’ dan Dapat	-sistem menampilkan pesan ‘berita berhasil ditambahkan’ dan Dapat menambah berita	Valid

		<p>menampilkan form tambah berita -mengisi data form -memilih tombol “Submit”</p> <p>Alternative 1 -login ke sistem sebagai admin -memilih menu “Berita” pada sistem -memilih submenu “Tulis Berita” menampilkan form tambah berita -tidak mengisi salah satu data form -memilih tombol “Submit”</p>	<p>menambah berita ke dalam database</p> <p>- sistem dapat menampilkan pesan ‘fill this field’.</p>	<p>ke dalam database.</p> <p>- sistem menampilkan pesan ‘fill this field’.</p>	
KF_32	Menghapus Berita	<p>-login ke sistem sebagai admin -memilih menu “Berita” pada sistem -memilih submenu “List Berita” memilih berita yang ingin dihapus lalu memilih icon “delete”</p>	<p>-sistem dapat menampilkan pesan ‘berita berhasil dihapus’ dan berita dihapus dari database</p>	<p>-sistem menampilkan pesan ‘berita berhasil dihapus’ dan berita dihapus dari database</p>	Valid
KF_33	Mengubah Berita	<p>-login ke sistem sebagai admin -memilih menu “Berita” pada sistem -memilih submenu “List Berita” memilih berita yang ingin diubah lalu memilih Icon “edit” -mengisi perubahan berita lalu pilih “update”</p>	<p>-Sistem dapat mengubah berita yang diinginkan</p>	<p>-sistem mengubah berita yang diinginkan</p>	Valid
KF_34	Menambah Pegawai	<p>-login ke sistem sebagai admin -memilih menu “Pegawai” pada sistem -memilih submenu “Tambah Pegawai” menampilkan form tambah pegawai -mengisi data form -memilih tombol “Submit”</p> <p>Alternative 1 -login ke sistem sebagai admin</p>	<p>-sistem dapat menampilkan pesan ‘pegawai berhasil ditambahkan’ dan Dapat menambah pegawai ke dalam database</p> <p>- sistem dapat menampilkan pesan ‘fill this field’.</p>	<p>-sistem menampilkan pesan ‘pegawai berhasil ditambahkan’ dan Dapat menambah pegawai ke dalam database.</p> <p>- sistem menampilkan pesan ‘fill this field’.</p>	Valid

		<ul style="list-style-type: none"> -memilih menu “Pegawai” pada sistem -memilih submenu “Tambah pegawai” menampilkan form tambah pegawai -tidak mengisi salah satu data form -memilih tombol “Submit” 			
KF_35	Menghapus Pegawai	<ul style="list-style-type: none"> -login ke sistem sebagai admin -memilih menu “Pegawai” pada sistem -memilih submenu “List Pegawai” memilih pegawai yang ingin dihapus lalu memilih icon “delete” 	<ul style="list-style-type: none"> -sistem dapat menampilkan pesan ‘pegawai berhasil dihapus’ dan pegawai dihapus dari database 	<ul style="list-style-type: none"> -sistem menampilkan pesan ‘pegawai berhasil dihapus’ dan pegawai dihapus dari database 	Valid
KF_36	Mengubah Pegawai	<ul style="list-style-type: none"> -login ke sistem sebagai admin -memilih menu “Berita” pada sistem -memilih submenu “List Berita” memilih berita yang ingin diubah lalu memilih Icon “edit” -mengisi perubahan berita lalu pilih “update” 	<ul style="list-style-type: none"> -Sistem dapat mengubah pegawai yang diinginkan 	<ul style="list-style-type: none"> -sistem mengubah pegawai yang diinginkan 	Valid
KF_37	Menambah Fasilitas	<ul style="list-style-type: none"> -login ke sistem sebagai admin -memilih menu “Fasilitas” pada sistem -memilih submenu “Tulis Fasilitas” menampilkan form tambah Fasilitas -mengisi data form -memilih tombol “Submit” <p>Alternative 1</p> <ul style="list-style-type: none"> -login ke sistem sebagai admin -memilih menu “Fasilitas” pada sistem -memilih submenu “Tulis fasilitas” menampilkan form tambah Fasilitas -tidak mengisi salah satu data form -memilih tombol “Submit” 	<ul style="list-style-type: none"> -sistem dapat menampilkan pesan ‘Fasilitas berhasil ditambahkan’ dan Dapat menambah Fasilitas ke dalam database - sistem dapat menampilkan pesan ‘fill this field’. 	<ul style="list-style-type: none"> -sistem menampilkan pesan ‘Fasilitas berhasil ditambahkan’ dan Dapat menambah Fasilitas ke dalam database. - sistem menampilkan pesan ‘fill this field’. 	Valid

<p>KF_38</p>	<p>Menghapus Fasilitas</p>	<p>-login ke sistem sebagai admin -memilih menu “Fasilitas” pada sistem -memilih submenu “List Fasilitas i” memilih Fasilitas yang ingin dihapus lalu memilih icon “delete”</p>	<p>-sistem dapat menampilkan pesan ‘fasilitas berhasil dihapus’ dan fasilitas dihapus dari database</p>	<p>-sistem menampilkan pesan ‘fasilitas berhasil dihapus’ dan fasilitas dihapus dari database</p>	<p>Valid</p>
<p>KF_39</p>	<p>Mengubah Fasilitas</p>	<p>-login ke sistem sebagai admin -memilih menu “Fasilitas” pada sistem -memilih submenu “List Fasilitas” memilih fasilitas yang ingin diubah & pilih icon “edit” -mengisi perubahan fasilitas lalu pilih “update”</p>	<p>-Sistem dapat mengubah Fasilitas yang diinginkan</p>	<p>-sistem mengubah Fasilitas yang diinginkan</p>	<p>Valid</p>
<p>KF_40</p>	<p>Menambah Tips</p>	<p>-login ke sistem sebagai admin -memilih menu “Tips Olahraga” pada sistem -memilih submenu “Tambah tips olahraga” menampilkan form tambah tips olahraga -mengisi data form -memilih tombol “Submit” Alternative 1 -login ke sistem sebagai admin -memilih menu “Tips Olahraga” pada sistem -memilih submenu “Tulis Tips” menampilkan form tambah Tips Olahraga -tidak mengisi salah satu data form -memilih tombol “Submit”</p>	<p>-sistem dapat menampilkan pesan ‘Tips berhasil ditambahkan’ dan Dapat menambah tips olahraga ke dalam database - sistem dapat menampilkan pesan ‘fill this field’.</p>	<p>-sistem menampilkan pesan ‘Tips berhasil ditambahkan’ dan Dapat menambah tips olahraga ke dalam database. - sistem menampilkan pesan ‘fill this field’.</p>	<p>Valid</p>
<p>KF_41</p>	<p>Menghapus Tips</p>	<p>-login ke sistem sebagai admin -memilih menu “Tips Olahraga” pada sistem -memilih submenu “List Tips” memilih tips yang ingin dihapus lalu memilih icon “delete”</p>	<p>-sistem dapat menampilkan pesan ‘pegawai berhasil dihapus’ dan pegawai dihapus dari database</p>	<p>-sistem menampilkan pesan ‘pegawai berhasil dihapus’ dan pegawai dihapus dari database</p>	<p>Valid</p>

<p>KF_42</p>	<p>Mengubah Tips</p>	<ul style="list-style-type: none"> -login ke sistem sebagai admin -memilih menu “Tips Olahraga” pada sistem -memilih submenu “List Tips” memilih tips yang ingin diubah & pilih icon “edit” -mengisi perubahan tips lalu pilih“update” 	<ul style="list-style-type: none"> -Sistem dapat mengubah tips yang diinginkan 	<ul style="list-style-type: none"> -sistem mengubah tips yang diinginkan 	<p>Valid</p>
---------------------	----------------------	--	---	---	--------------



6.1.3 Pengujian *Compatibility*

6.1.3.1 Tujuan

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

6.1.3.2 Mekanisme

Mekanisme pengujian dengan melakukan pendefinisian daftar kebutuhan yang diperlukan untuk menguji sistem dan membuat scenario pengujian serta hasil yang diperoleh terhadap setiap pengujian tersebut. Pengujian dilakukan dengan 3 sistem operasi dan 5 browser yang berbeda.

6.1.3.3 Hasil Pengujian

1. Daftar kebutuhan pengujian pada sistem operasi Linux di Ubuntu dijelaskan pada tabel 6.8:

Tabel 6.8 Daftar kebutuhan pengujian linux ubuntu

Daftar kebutuhan pengujian compatibility	
1	Linux Ubuntu 12.04
2	Xampp version: 5.3
3	Mozilla firefox 15

Setelah didapatkan perangkat lunak yang diperlukan untuk menguji sistem, langkah selanjutnya adalah membuat skenario pengujian *compatibility* sistem dijelaskan pada tabel 6.9.

Tabel 6.9 Skenario Pengujian Linux Ubuntu

Scenario pengujian di Ubuntu	Hasil yang diperoleh
1. Installasi Linux Ubuntu 12.04	Berdasarkan pengujian ini maka diperoleh suatu hasil yakni sistem dapat berjalan dengan lancar pada sistem operasi Linux Ubuntu 12.04
2. Installasi Xampp version: 5.3	
3. Salin file projek ke dalam folder htdocs	
4. Import basis data ke phpmyadmin	
5. Buka localhost/ubsportcenter	

Berdasarkan scenario pengujian maka hasil pengujian ditunjukkan pada Gambar 6.5



Gambar 6.5 Halaman utama di linux Ubuntu

2. Daftar kebutuhan pengujian compatibility pada sistem operasi Linux CentOS dijelaskan pada tabel 6.10:

Tabel 6.10 Daftar Kebutuhan Pengujian Linux CentOS

Daftar kebutuhan pengujian compatibility	
1	CentOS 6.5
2	Xampp version: 5.3
3	Mozzila firefox 15

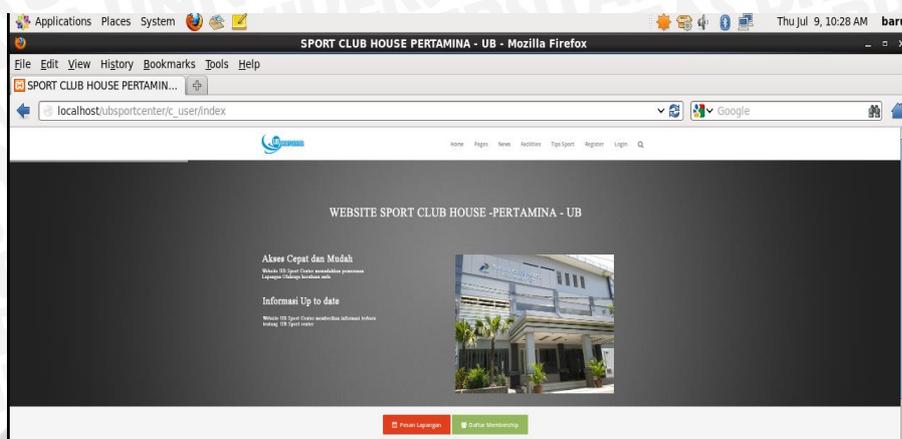
Setelah didapatkan perangkat lunak yang diperlukan untuk menguji sistem langkah selanjutnya adalah membuat skenario pengujian compatibility sistem dijelaskan pada tabel 6.11.

Tabel 6.11 Skenario Pengujian Linux CentOS

Scenario pengujian di Ubuntu	Hasil yang diperoleh
1. Installasi Linux CentOS 6.5	Berdasarkan pengujian ini maka diperoleh suatu hasil yakni sistem dapat berjalan dengan lancar pada sistem operasi Linux CentOS 6.5
2. Installasi Xampp version: 5.3	
3. Salin file projek ke dalam folder htdocs	
4. Import basis data ke phpmyadmin	
5. Buka localhost/ubspportcenter	



Berdasarkan scenario pengujian maka hasil pengujian ditunjukkan pada Gambar 6.6.



Gambar 6.6 Halaman utama di linux centOS

3. Daftar kebutuhan pengujian compatibility pada sistem operasi windows dijelaskan pada tabel 6.12:

Tabel 6.12 Daftar Kebutuhan Pengujian Windows

Daftar kebutuhan pengujian compatibility	
1	Windows 8 64bit
2	Xampp 1.7.3
3	Mozilla firefox 15
4	Google Chrome Version 43.0.2357.130
5	Internet Explore 10

Setelah didapatkan perangkat lunak yang diperlukan untuk menguji sistem langkah selanjutnya adalah membuat skenario pengujian compatibility sistem dijelaskan pada tabel 6.13.

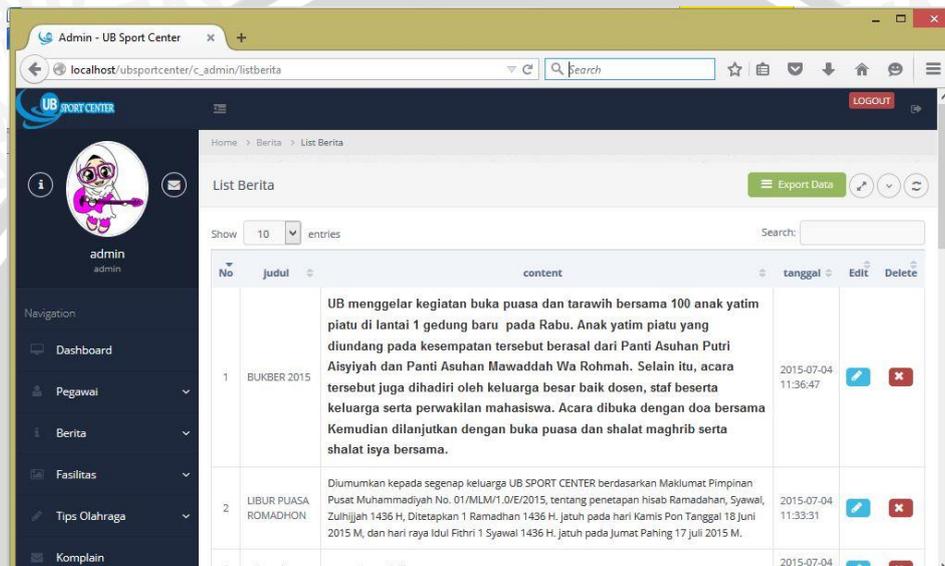
Tabel 6.13 Skenario Pengujian Windows

Scenario pengujian di Ubuntu	Hasil yang diperoleh
4. Installasi Windows 8 64bit	Berdasarkan pengujian ini maka diperoleh suatu hasil yakni sistem dapat berjalan dengan lancar pada sistem operasi Windows
2. Installasi Xampp 1.7.3	
3. Salin file projek ke dalam folder htdocs	
4. Import basis data ke phpmyadmin	
5. Buka localhost/ubsportcenter di browser google chrome	

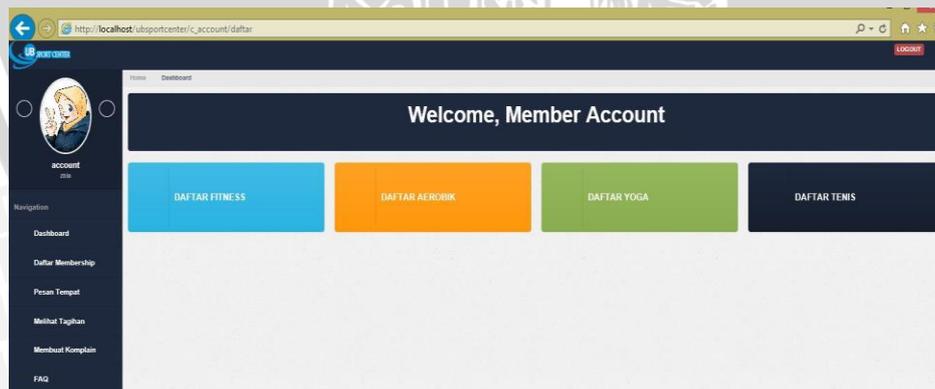


6. Buka localhost/ubsportcenter di browser internet explore 10
7. Buka localhost/ubsportcenter di browser Mozilla firefox

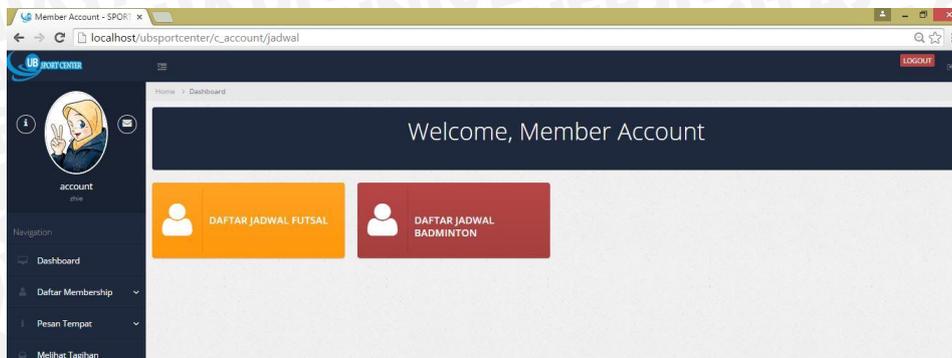
Berdasarkan scenario pengujian maka hasil pengujian ditunjukkan pada Gambar 6.7 hingga Gambar 6.9.



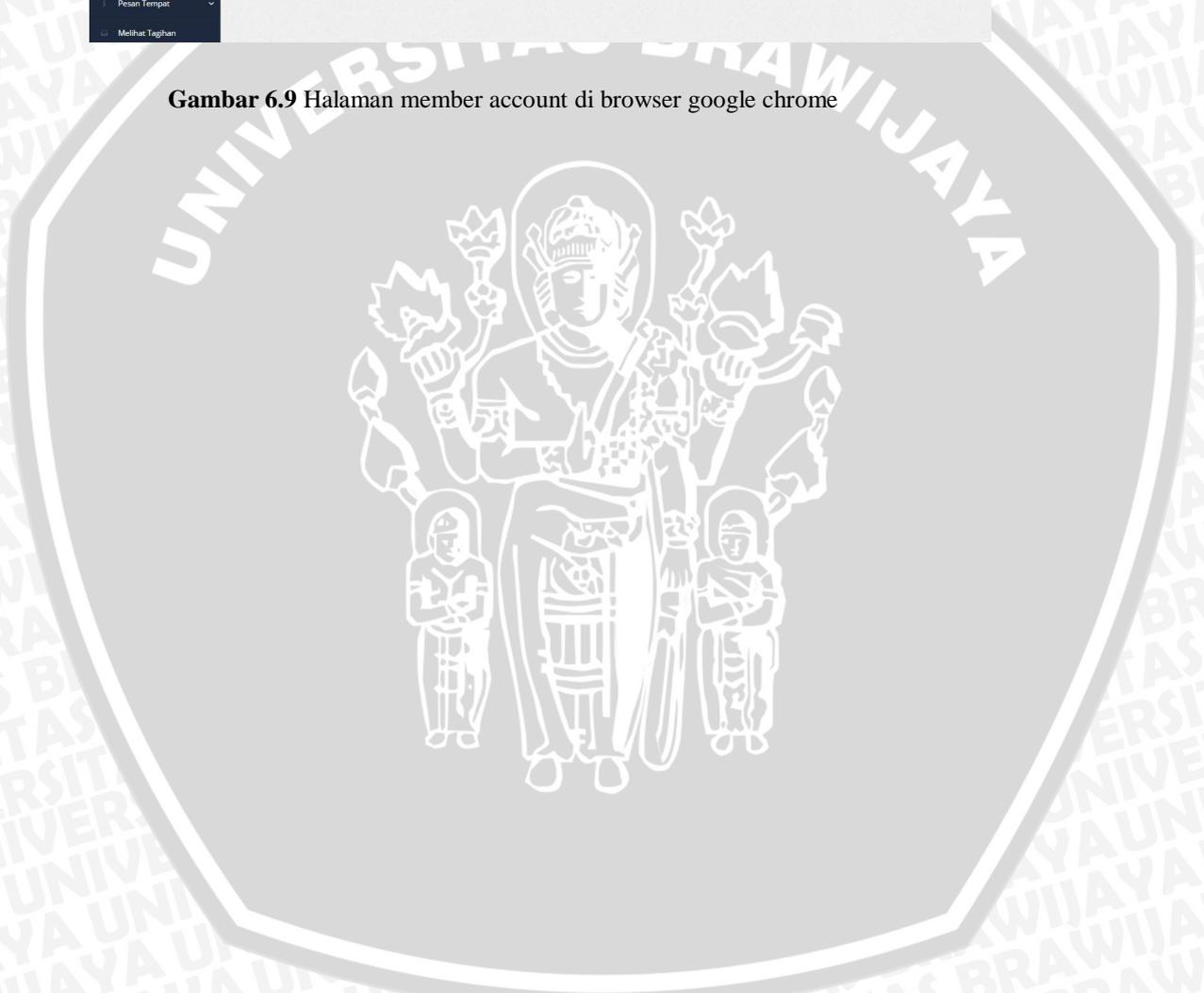
Gambar 6.7 Halaman admin di browser mozilla firefox



Gambar 6.8 Halaman member account di browser internet explorer



Gambar 6.9 Halaman member account di browser google chrome



6.2 Analisis Sistem

Pada bagian ini akan dilakukan seluruh analisa hasil dari semua pengujian yang telah dilakukan, analisis bertujuan untuk menganalisa data hasil pengujian hingga menghasilkan kesimpulan-kesimpulan. Analisis ini dilakukan berdasarkan hasil implementasi hingga pengujian yang telah dilakukan terhadap sistem dan mengacu pada dasar teori. Proses analisis yang dilakukan antara lain adalah analisis hasil pengujian *white box*, analisa hasil dari pengujian *black box* dan analisa hasil pengujian *compatibility*.

6.2.1 Analisis Pengujian Black Box

Berdasarkan hasil pengujian *black box* secara validasi, implementasi perancangan sistem dapat dikatakan layak dan berhasil serta sesuai dengan kebutuhan fungsional yang ada, hasil pengujian menunjukkan dari 42 pengujian yang diuji minimal 2 kali menunjukkan bahwa 100% kebutuhan fungsional dari sistem telah valid ketika diuji dengan kasus tertentu. Sehingga dapat disimpulkan bahwa Sistem Informasi Sport Club House Pertamina - UB yang dibuat telah berjalan dengan baik sesuai dengan yang diharapkan.

6.2.2 Analisis Pengujian White Box

Berdasarkan hasil pengujian *white box* yang dilakukan, menunjukkan bahwa semua jalur algoritma telah diuji, dengan didapatkan hasil berupa *cyclomatic complexity* yang berbeda dari setiap data uji. Dimana untuk fitur login didapatkan *cyclomatic complexity* sebanyak 8, fitur mendaftar membership fitnes didapatkan *cyclomatic complexity* sebanyak 10, dan register member account *cyclomatic complexity* sebanyak 3.

6.2.3 Analisis Pengujian Compatibility

Berdasarkan hasil pengujian yang dilakukan dengan melihat hasil kinerja sistem yang dilakukan terhadap 3 sistem operasi dan 5 browser. Berdasarkan hasil pengujian *compatibility* dapat disimpulkan bahwa sistem informasi Sport Club House Pertamina-UB dapat berjalan secara normal pada 3 sistem operasi yang berbeda dan 5 browser tersebut.

BAB 7 PENUTUP

6.1 Kesimpulan

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

1. Dalam perancangan dan implementasi sistem informasi pada Sport Club House Pertamina - UB menggunakan pendekatan *evolutionary prototyping* telah sesuai dengan kebutuhan pengguna.
2. Berdasarkan hasil pengujian pada sistem informasi yang telah dilakukan diperoleh kesimpulan bahwa kompleksitas dari sistem ini adalah rendah sehingga kerumitan perbaikan coding juga rendah, kemudian secara fungsionalitas dari 42 pengujian kebutuhan fungsional sistem ini bahwa berfungsi 100% valid dan secara keseluruhan berjalan dengan baik dan sistem berjalan secara normal pada 3 sistem operasi dan 5 browser yang berbeda.

6.2 Saran

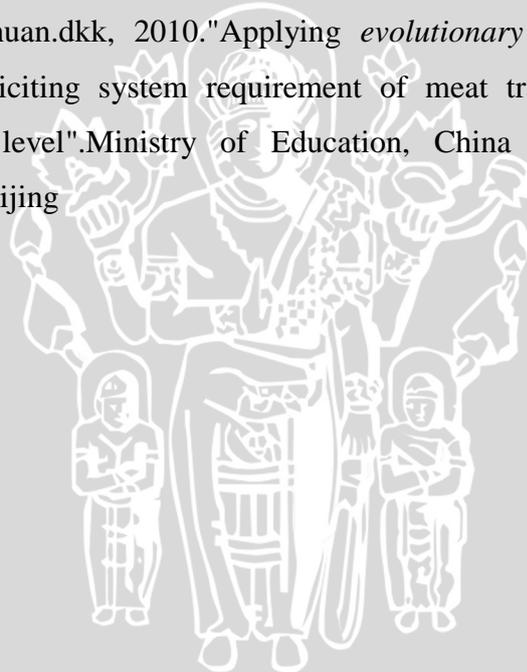
Saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah:

1. Untuk penelitian lebih lanjut, diharapkan sistem ini dapat mencakup semua aspek manajemen di dalam Sport Club House Pertamina - UB seperti mengelola pemasukan juice corner, mengelola pengecekan member setiap hari, mengelola loker, mengelola pengeluaran keuangan.
2. Pengimplementasian sistem informasi Sport Club House Pertamina - UB dapat dilakukan dengan menambahkan integrasi *web service* pada kebutuhan fungsi *login*.
3. Pengimplementasian sistem informasi Sport Club House Pertamina - UB dapat dilakukan dengan menambahkan integrasi *web service* pada fitur keuangan dengan sistem rektorat UB yaitu aplikasi SPM.
4. Pengembangan sistem dapat dilakukan dengan menambahkan fitur sms *gateway* untuk penjadwalan dan pendaftaran member.

DAFTAR PUSTAKA

- [ALF-07] Al Fatta, Hanif. 2007. Analisis & Perancangan Sistem Informasi. Yoyakarta : Penerbit Andi
- [LEN-03] Lenz Gunther, Thomas Moeller . 2003. “.Net- A Complete Development Cycle”. Boston : Pearson Education, Inc
- [LET-02] Lethbridge, Timothy C., Robert Laganieri. 2002. *Object-Oriented Software Engineering: Practical Software Development using UML and Java*. England : McGraw-Hill International (UK) Limited,.
- [LUC-11] Luca,Mario.dkk.2011. *A Model-Driven Approach for the Fast Prototyping of Web Applications*. IEEE. 978-1-4577-0700-1/11/\$26.00.
- [MCC-96] McConnell, Steve. 1996. “Rapid Development”. Microsoft Press.Canada
- [SIM-10] Simarmata, Janner.2010. *Rekayasa Web.CV* Andi Offset: Yogyakarta.
- [SUT-11] Sutanta, Edhy. 2011. *Basis Data dalam Tinjauan Konseptual*. Yogyakarta : Penerbit Andi.
- [SAB-09] Sabharwai , Sangeeta .2009. “Software Engineering” New Age International Pvt Ltd Publishers
- [RAN-13] Ranjan,Narayan.dkk. 2013.*Requirement Elicitation: A Live Prototyping Approach. International Journal of Computer Applications.Volume 72– No.13, May 2013*
- [ROS-11] Rosa, A.S. & Shalahuddin, M. 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Modula.
- [ROS-13] Rosa A. S, M.Salahuddin. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika : Bandung.

- [PRE-12] Pressman, Roger. 2012. *Rekayasa Perangkat Lunak*. Yogyakarta.
- [PRE-10] Pressman, Roger. 2010. *Software Engineering: A Practitioner's Approach*. Mc Graw-Hill.
- [PRY-05] Prayitno, Wendhie; S, Kom. 2005. *Desain Model Sistem Perangkat Lunak dengan UML*.
- [TAV-12] Taveter, Kuldar.dkk.2012. *Method for Rapid Prototyping of Societal Information Systems*. IEEE. ISBN 978-83-60810-48-4
- [TAL-01] Tale, G. (2001). *Prototyping: Helping to Build the right software*. Sciece Direct, 237-238.
- [WHI-04] Whitten et. al. 2004. *Systems Analysis and Design Methods, 6th edition*. New York : McGraw-Hill Companies, Inc.
- [ZHA-10] Zhang xiaoshuan.dkk, 2010. "Applying *evolutionary prototyping* model for eliciting system requirement of meat traceability at agribusiness level". Ministry of Education, China Agricultural University:Beijing



LAMPIRAN 1 WAWANCARA

Lampiran ini berisi pertanyaan-pertanyaan yang diajukan ketika melakukan wawancara secara terstruktur. Wawancara dilakukan dengan Kepala dan pegawai Sport Club House Pertamina-UB. Wawancara dilakukan guna mengetahui informasi secara umum seputar tugas mereka dan permasalahan yang dihadapi terkait tugas – tugas yang dilakukan. Berikut adalah beberapa pertanyaan yang penulis utarakan kepada narasumber.

1. Bagaimana struktur organisasi di Sport Club House Pertamina-UB? Bagian-bagian mana yang saling terkait satu sama lain?
2. Bagaimana Kendala saat ini di Sport Club House Pertamina – UB?
3. Apa penanganan yang dilakukan terhadap kendala tersebut?
4. Bagaimana pemahaman pegawai tentang Teknologi Informasi?
5. Bagaimana promosi kepada warga UB atau umum? Seperti apa contohnya?
6. Apa saja Olahraga beserta harganya?
7. Bagaimana prosedur dalam menjalankan proses bisnis dan apa menggunakan standar operasional prosedur?
8. Bagaimana proses pendaftaran member dan pembayaran?
9. Bagaimana proses pemesanan tempat lapangan dan pembayaran?
10. Setelah dilakukan analisa, menurut Anda proses mana yang penting atau sering digunakan dalam menjalankan proses bisnis?
11. Menurut Anda proses mana yang sedang dalam menjalankan proses bisnis?

Jawaban dari pertanyaan-pertanyaan wawancara secara jelas dan singkat.

1. Struktur organisasi masih menggunakan yang lama (bisa dilihat pada bab 2 dasar teori) dan bagian-bagian terpenting yang saling terkait satu sama lain seperti bagian gedung dengan keuangan, bagian front office dengan kasir, bagian kasir dengan keuangan dan bagian keuangan dengan bendahara.

2. Beberapa kendala saat ini mengenai kebutuhan akan informasi yang kurang seperti media promosi dan jadwal pemesanan yang bisa dilihat di masing-masing lapangan membuat mereka harus datang langsung untuk melihat dan memesan serta data registrasi pendaftaran member yang kurang terorganisir dengan baik jika melakukan pendaftaran ulang.
3. Penanganan saat ini dengan adanya kotak saran di dekat kantor dan data menggunakan excel untuk data member.
4. Tidak adanya pegawai sebagai admin sistem membuat jarang nya penggunaan teknologi informasi.
5. Selama ini menggunakan facebook dan email sebagai media promosi tetapi belum bisa diorganisir dengan baik dan uptodate.
6. Olahraga yang tersedia adalah futsal, badminton, tenis, fitness, aerobic dan yoga dan untuk masing-masing harga bisa dilihat pada SK Rektor 2013.
7. Menggunakan prosedur dari rektorat dan belum turun SOPnya.
8. Proses pendaftaran member dilakukan pada bagian frontoffice untuk mendaftar dan bagian kasir untuk membayar
9. Proses pemesanan lapangan melalui kantor di bagian gedung dan jadwal yang tertera di lapangan masing-masing serta pembayaran bisa melalui bagian gedung atau rekening rektorat UB.
10. Proses untuk register, mendaftar member dan pemesanan lapangan, validasi, pembayaran atau pemasukan keuangan dan laporan keuangan serta complain dan menambah beberapa proses.
11. Proses untuk melihat data-data

LAMPIRAN 2 PERANCANGAN ITERASI

2.1 Perancangan Awal

Dalam tahap perancangan analisis ini dilakukan review terhadap dokumen-dokumen analisis beserta wawancara terhadap pihak terkait di Sport Club House Pertamina - UB. Tujuan dari tahap ini adalah untuk mengumpulkan informasi tentang entitas yang terlibat dalam sistem dan kebutuhan fungsional sistem yang menggambarkan bagaimana sistem Sport Club House Pertamina - UB didefinisikan.

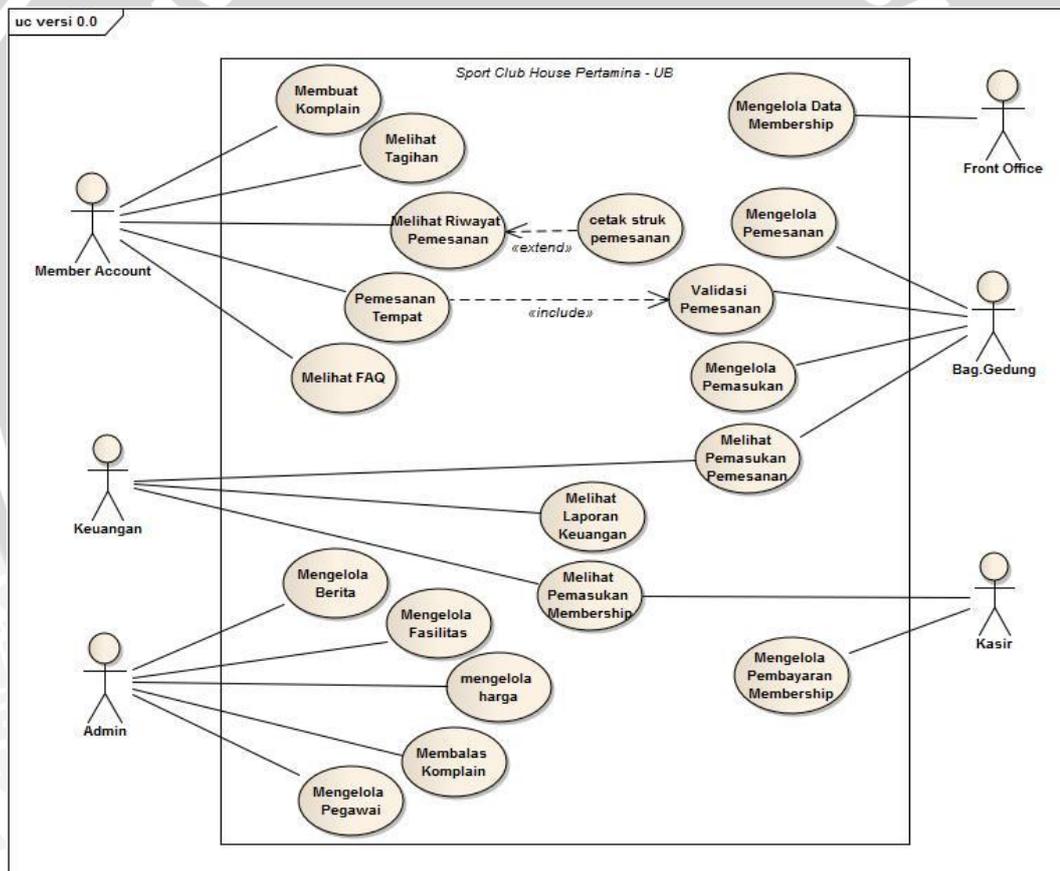
Adapun daftar kebutuhan fungsional awal dalam sistem Sport Club House Pertamina - UB diperlihatkan pada tabel berikut.

Kode Fungsi	Fungsi	Deskripsi	Prioritas
KF_01	Login	Sistem harus mampu menyediakan fungsi login dan memberikan batasan hak akses.	high
KF_02	Melihat Berita	Sistem harus mampu menampilkan berita Sport Club House Pertamina - UB	high
KF_03	Melihat Fasilitas	Sistem harus mampu menampilkan fasilitas di Sport Club House Pertamina - UB	high
KF_04	Register Account	Sistem harus mampu menyediakan fungsi Register Account (pendaftaran) ke sistem	high
KF_05	Pemesanan Tempat	Sistem harus mampu melakukan pemesanan tempat dengan melihat jadwal olahraga	high
KF_06	Melihat FAQ	Sistem harus mampu menampilkan faq	medium
KF_07	Cetak struk pemesanan	Sistem harus mampu melakukan cetak data pemesanan	high
KF_08	Melihat Riwayat Pemesanan	Sistem harus mampu menampilkan riwayat pemesanan tempat	medium
KF_09	Membuat Komplain	Sistem harus mampu melakukan komplain	medium
KF_10	Melihat Tagihan	Sistem harus mampu menampilkan tagihan keuangan	medium

KF_11	Menambah Membership	Sistem harus mampu menambah membership	high
KF_12	Melihat data Membership	Sistem harus mampu menampilkan data membership	medium
KF_13	Menghapus Membership	Sistem harus mampu menghapus membership	low
KF_14	Menambah Pembayaran	Sistem harus mampu menambah pembayaran	high
KF_15	Menghapus Pembayaran	Sistem harus mampu menghapus pembayaran	low
KF_16	Melihat Pemasukan Membership	Sistem harus mampu menampilkan data pemasukan membership	medium
KF_17	Validasi Pemesanan	Sistem harus mampu memvalidasi pemesanan tempat	high
KF_18	Menambah Pemesanan	Sistem harus mampu menambah Pemesanan	high
KF_19	Menghapus Pemesanan	Sistem harus mampu menghapus Pemesanan	low
KF_20	Melihat data pemesanan	Sistem harus mampu menampilkan data pemesanan	medium
KF_21	Menambah Pemasukan	Sistem harus mampu meambah pemasukan pemesanan	high
KF_22	Menghapus Pemasukan	Sistem harus mampu menghapus pemasukan pemesanan	low
KF_23	Melihat Pemasukan pemesanan	Sistem harus mampu menampilkan total pemasukan pemesanan	medium
KF_24	Melihat Laporan Keuangan	Sistem harus mampu menampilkan laporan keuangan	high
KF_25	Membalas Komplain	Sistem harus mampu membalas komplain	high
KF_26	Menambah Berita	Sistem harus mampu menambah berita	high
KF_27	Menghapus Berita	Sistem harus mampu menghapus berita	low
KF_28	Mengubah Berita	Sistem harus mampu mengubah berita	medium
KF_29	Menambah Pegawai	Sistem harus mampu menambah pegawai	high
KF_30	Menghapus Pegawai	Sistem harus mampu menghapus pegawai	low

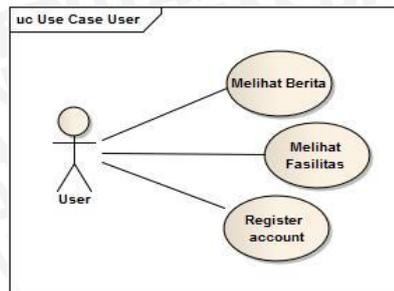
KF_31	Mengubah Pegawai	Sistem harus mampu mengubah pegawai	medium
KF_32	Menambah Fasilitas	Sistem harus mampu menambah fasilitas	high
KF_33	Menghapus Fasilitas	Sistem harus mampu menghapus fasilitas	low
KF_34	Mengubah Fasilitas	Sistem harus mampu mengubah fasilitas	medium
KF_35	Menambah Harga	Sistem harus mampu menambah harga	high
KF_36	Menghapus Harga	Sistem harus mampu menghapus harga	low
KF_37	Mengubah Harga	Sistem harus mampu mengubah harga	medium

Model Diagram Use Case dengan login.



Gambar 8.1 use case diagram awal

Model Diagram Use Case Iterasi 0 di actor User tanpa login pada gambar 8.2.



Gambar 8.2 use case diagram awal

2.2 Perancangan Prototype Iterasi 1

Evaluasi ini dilakukan setelah dilakukan evaluasi pada implementasi versi 0.0. Hasil dari evaluasi tersebut adalah sebagai berikut.

1. Menambah Aktor bendahara
2. Menambah fitur melihat laporan keuangan pada aktor bendahara
3. Fitur Harga tidak perlu dibuat pada Aktor Admin
4. Aktor Front Office dapat melakukan validasi membership
5. Aktor Member account dapat mendaftar membership dan melihat riwayat membership serta dapat mencetak struk membership
6. Penambahan cek tanggal pada laporan pemasukan, laporan pembayaran dan laporan keuangan.

Evaluasi ini mengakibatkan perlu dilakukannya perubahan pada kebutuhan sistem. Berikut pengurangan fitur harga pada kebutuhan fungsionalitas sistem setelah dilakukan evaluasi tahap awal.

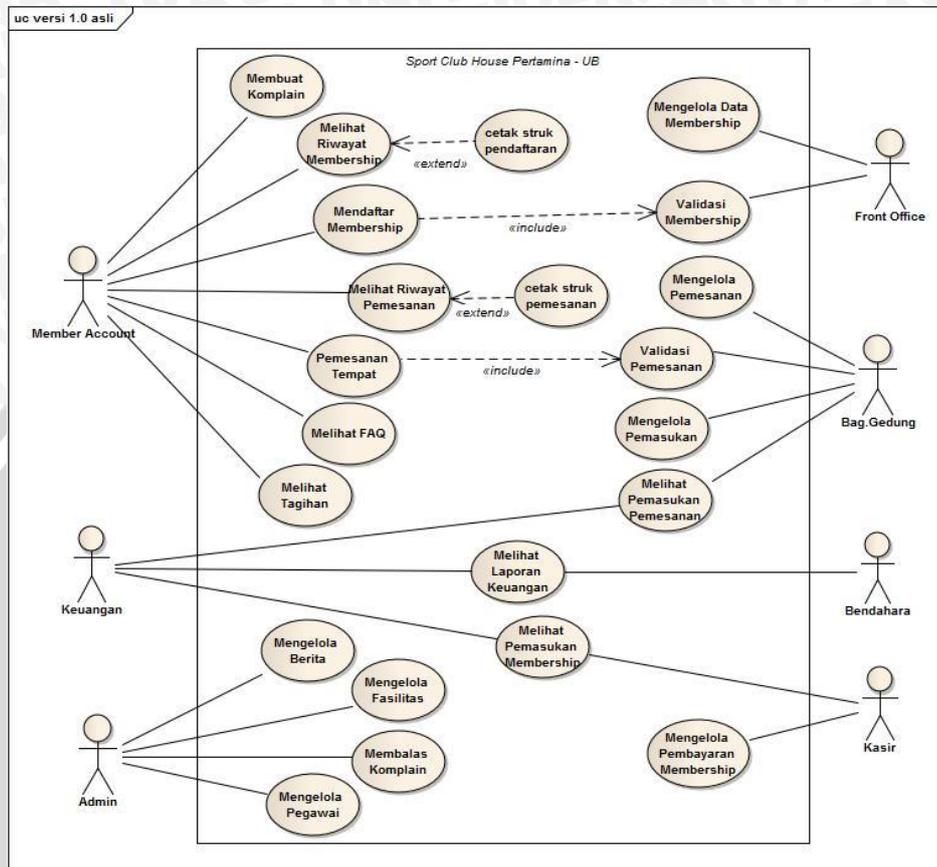
Fungsi	Deskripsi	Prioritas
Menambah Harga	Sistem harus mampu menambah harga olahraga	high
Menghapus Harga	Sistem harus mampu menghapus harga olahraga	low
Mengubah Harga	Sistem harus mampu mengubah harga olahraga	medium

Berikut tambahan kebutuhan fungsionalitas sistem setelah dilakukan evaluasi tahap awal.

Fungsi	Deskripsi	Prioritas
Mendaftar Membership	Sistem harus mampu menampilkan pendaftaran membership	high
Cetak struk membership	Sistem harus mampu melakukan cetak data pendaftaran membership	high
Melihat Riwayat Membership	Sistem harus mampu menampilkan riwayat pendaftaran membership	medium
Validasi data	Sistem harus mampu memvalidasi data	high

Membership	membership
------------	------------

Berikut model use case iterasi 1 setelah dilakukan evaluasi tahap 1.



Gambar 8.3 use case diagram iterasi 1

2.3 Perancangan Prototype Iterasi 2

Prototype ini dilakukan setelah dilakukan evaluasi implementasi versi 1.0 Hasil dari evaluasi tersebut adalah sebagai berikut.

- 1 Aktor Admin dapat mengelola tips olahraga
- 2 Actor user dapat melihat tips olahraga
- 3 Merubah tampilan popup pada semua proses hapus

Evaluasi ini mengakibatkan perlu dilakukannya penambahan pada analisa kebutuhan.

Pemodelan Use Case Tahap 3 atau Iterasi 3 dapat dilihat pada bab sebelumnya pada gambar 4.4 dan gambar 4.5 mengenai model use case diagram iterasi 3.

UNIVERSITAS BRAWIJAYA

LAMPIRAN 3 SURAT PENELITIAN

**SURAT
KETERANGAN PENELITIAN**



UNIVERSITAS BRAWIJAYA

LAMPIRAN 4 SURAT KETERANGAN

**SURAT KETERANGAN
PENERIMAAN APLIKASI**

