

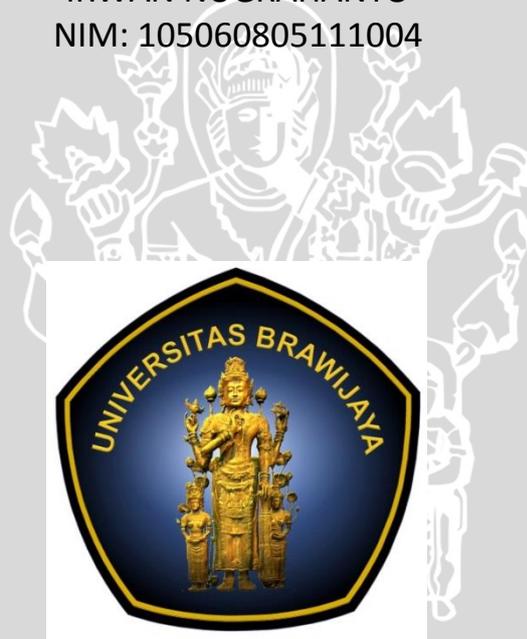
# **RANCANG BANGUN LABORATORIUM KIMIA VIRTUAL UNTUK SMA KELAS X**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

**IRWAN NUGRAHANTO**  
NIM: 105060805111004



**INFORMATIKA**

**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2016**

## PENGESAHAN

RANCANG BANGUN LABORATORIUM KIMIA  
VIRTUAL UNTUK SMA KELAS X

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Nama Irwan Nugrahanto

NIM: 105060805111004

Skripsi ini telah diuji dan dinyatakan lulus pada

19 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Erig Muhammad Adams J, ST., M.Kom.

NIP: 19850410 201212 1 001

Wijaya Kurniawan, S.T., M.T.

NIP: 19820125 201504 1 002

Mengetahui

Ketua Program Studi Informatika/Ilkom

Drs. Marji., M.T.

NIP: 19670801 199203 1 003

## PERNYATAAN ORISINALITAS

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

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

Malang, 27 Januari 2016



Irwan Nugrahanto

NIM: 10506080511004

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, karena dengan rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **“Rancang Bangun Laboratorium Kimia Virtual Untuk SMA Kelas X”** dengan baik.

Shalawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW, yang dengan semangat juangnya terus memberikan tauladan untuk tetap bersemangat dan tidak berputus asa dengan rahmat-Nya. Maka tanpa itu semua, niscaya Penulis tidak akan dapat menyelesaikan tugas akhir ini dengan baik.

Penulisan dan penyusunan laporan skripsi ini dapat terlaksana dengan baik atas bimbingan Eriq Muhammad Adams J, S.T, M.Kom., selaku dosen pembimbing I dan Wijaya Kurniawan, S.T, M.T., selaku dosen pembimbing II yang telah banyak memberikan ilmu dan saran dalam penulisan. Adapun pihak lain yang membantu dan mendukung dalam penulisan skripsi ini diantaranya:

1. Kedua orang tua, Bapak Sukamto dan Ibu Yuniar yang selalu memberikan dukungannya selama ini.
2. Mega yunisa putri yang telah menjadi sahabat serta partner yang telah sangat banyak memberikan motivasi dan dukungan dalam kelancaran pengerjaan skripsi.
3. Citra Krissandra, Dika Rizky Yuniarto, Vashty Armila yang telah menjadi sahabat dan kawan yang telah banyak memberikan masukan dan dukungan selama proses pengerjaan skripsi.
4. Seluruh teman-teman LPM DISPLAY, LAB GAME, dan Informatika/Ilkom angkatan 2010 terima kasih atas segala pengalamannya selama ini.
5. Segenap dosen dan karyawan PTIIK Universitas Brawijaya yang telah membantu pelaksanaan skripsi ini.
6. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat demi terselesaikannya tugas akhir ini.

Atas bantuan dan dukungan selama ini penulis mengucapkan banyak terima kasih dan semoga jasa serta amal baiknya mendapatkan balasan dari Allah SWT. Dengan segala kerendahan hati, penulis menyadari bahwa skripsi ini masih jauh dari sempurna karena keterbatasan materi dan pengetahuan yang dimiliki. Akhirnya semoga skripsi ini dapat bermanfaat bagi pembaca terutama mahasiswa PTIIK Universitas Brawijaya.

Malang, 27 Januari 2016

Penulis  
nugrahanto1205@gmail.com

## ABSTRAK

**Irwan Nugrahanto. 2016. Rancang Bangun Laboratorium Kimia Virtual Untuk SMA Kelas X.** Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Eriq Muhammad Adams J, S.T., M.Kom., dan Wijaya Kurniawan, S.T., M.T.

Pelajaran kimia merupakan salah satu pelajaran yang wajib dan didukung oleh pembelajaran secara teori dan praktik yang terkait dengan percobaan laboratorium, namun tanpa disadari laboratorium kimia dengan peralatan dan bahan kimia didalamnya menjadikan laboratorium itu sendiri menjadi tempat yang dapat berpotensi menimbulkan bahaya kepada para siswa siswi yang menggunakan. Dengan menggunakan laboratorium virtual diharapkan dapat mengurangi dampak yang dapat terjadi serta membantu siswa dan siswi dalam memahami materi yang diajarkan. Penelitian dilakukan dengan mengumpulkan sumber dan menentukan daftar kebutuhan yang terdapat dalam sistem sesuai dengan materi dan percobaan apa saja yang ada pada pelajaran siswa dan siswi sekolah menengah atas kelas 10. Kebutuhan sistem terkait kebutuhan fungsional dan non-fungsional sistem, selain itu dilakukan implementasi dan pengujian pada sistem dengan metode pengujian validasi dan pengujian unit serta pengujian pada algoritma sistem. Sistem dikembangkan dengan model pengembangan sekuensial linear atau yang biasa dikenal dengan model waterfall, dari hasil pengumpulan data diangkat 3 materi percobaan yang akan ada pada sistem antara lain percobaan ikatan kimia, larutan elektrolit dan non-elektrolit, serta pembuktian hukum Proust. Dari hasil pengujian didapat hasil valid terhadap sistem baik terkait dengan fungsi-fungsi yang ada dalam sistem ataupun unit dari sistem sehingga dapat ditarik kesimpulan bahwa sistem telah berjalan baik dan sesuai dengan kebutuhan yang telah dibuat.

**Kata Kunci:** Laboratorium Virtual, Laboratorium Kimia, SMA Kelas X, Pelajaran Kimia

## ABSTRACT

**Irwan Nugrahanto. 2016. Design of Virtual Chemistry Laboratory for Senior High School Grade X.** Information Technology and Computer Science Program, Brawijaya University, Malang. *Advisor:* Eriq Muhammad Adams J, S.T., M.Kom., dan Wijaya Kurniawan, S.T., M.T.

Chemistry is one of the lessons that must be supported by learning theory and practice related to laboratory experiments, however unwittingly with equipments and chemicals in chemical laboratory makes the laboratory itself into a place that could potentially cause harm to the students who use it. By using the virtual laboratory is expected to reduce the impact that may occur and help students understand the material being taught. Research was conducted by collecting the source and determine list of requirements system in accordance with the material and experiment chemistry lesson in senior high school grade 10. System requirements related to functional requirements and non-functional requirements system, after that carried out the implementation and testing of the system with validation method and unit testing method as well as testing the system algorithms. The system was developed with a linear or sequential development model that is commonly known as the waterfall model, from the data collecting getting 3 material experiment that will exist on the system include chemical bonding experiment, electrolytes and non-electrolytes experiment, and legal proof Poust. From the test results obtained valid results for systems related to both the functions that exist in the system or unit of the system, so it can be concluded that the system has been running well and in accordance with the requirements that have been made.

**Keywords:** Virtual Laboratory, Chemistry Laboratory, Senior High School Grade X, Chemistry

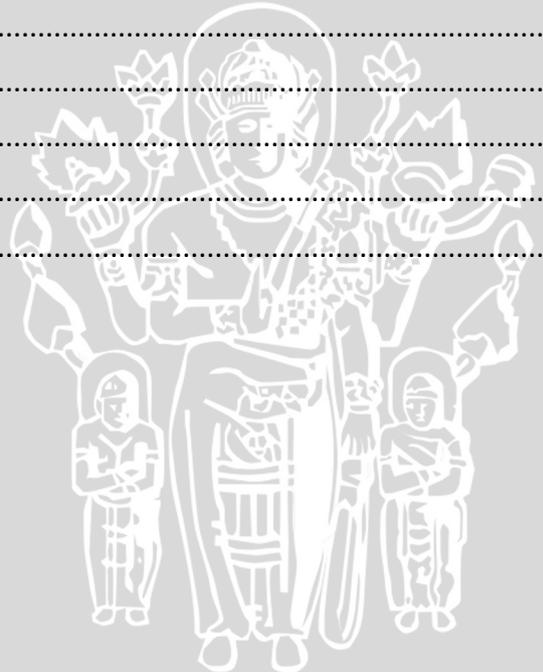
## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN .....	xiv
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	2
1.6 Sistematika pembahasan.....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>4</b>
2.1 Kajian Pustaka .....	4
2.2 Landasan Teori.....	4
2.2.1 Ilmu Kimia.....	4
2.2.2 Laboratorium Kimia.....	6
2.2.3 Virtual laboratory .....	8
2.2.4 Software Process Model .....	9
2.2.5 Unified Modeling Language (UML) .....	11
2.2.6 Unity Game .....	14
2.2.7 Pengujian Perangkat Lunak.....	17
<b>BAB 3 METODOLOGI .....</b>	<b>19</b>
3.1 Tipe Penelitian .....	19
3.2 Metode Penelitian .....	19
3.3 Metode Pengumpulan Data.....	20

3.4 Metode Pengujian dan Analisis Data .....	20
3.5 Perangkat Penelitian .....	21
3.6 Metode Pengembangan Sistem .....	21
<b>BAB 4 REKAYASA KEBUTUHAN SISTEM .....</b>	<b>23</b>
4.1 Analisis Kebutuhan Sistem .....	23
4.1.1 Gambaran Umum Sistem .....	23
4.1.2 Identifikasi Aktor .....	24
4.1.3 Daftar Aktor .....	24
4.1.4 Diagram Use Case .....	25
4.1.5 Skenario Use Case .....	25
4.1.6 Analisis Class .....	29
4.2 Perancangan Aplikasi .....	32
4.2.1 Perancangan Class Diagram .....	32
4.2.2 Diagram Activity .....	33
4.2.3 Diagram Sequence .....	38
4.2.4 Antarmuka .....	41
<b>BAB 5 IMPLEMENTASI .....</b>	<b>44</b>
5.1 Batas Implementasi .....	44
5.2 Implementasi Algoritma .....	44
5.2.1 Algoritma Main Button .....	44
5.2.2 Algoritma Menjalankan Halaman Percobaan .....	45
5.2.3 Algoritma Menampilkan Informasi Benda .....	46
5.2.4 Algoritma Menggerakkan Benda .....	47
5.3 Implementasi Antar Muka Pengguna .....	48
5.3.1 Halaman Utama .....	48
5.3.2 About Screen .....	48
5.3.3 Halaman Daftar Percobaan .....	49
5.3.4 Halaman Percobaan .....	49
5.3.5 Halaman Uji Percobaan 1 .....	50
5.3.6 Halaman Uji Percobaan 2 .....	50
5.3.7 Halaman Tutorial Penggunaan Lab Percobaan .....	51
<b>BAB 6 Pengujian Dan Analisis .....</b>	<b>52</b>



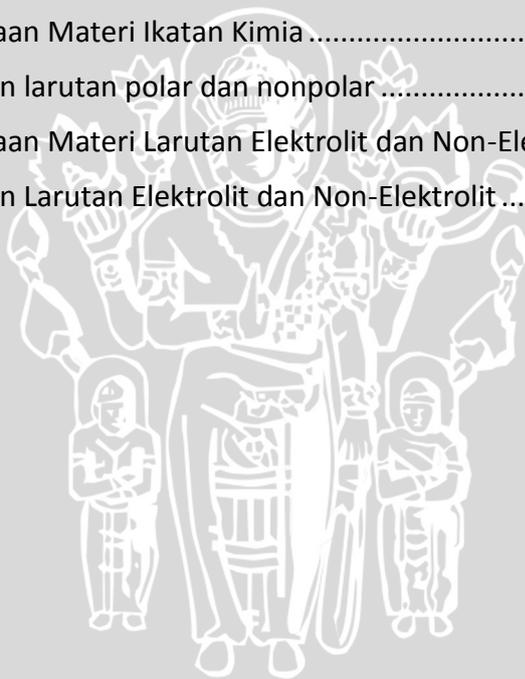
6.1 Pengujian Unit.....	52
6.1.1 Pengujian Unit Main Button.....	52
6.1.2 Pengujian Unit Menjalankan Halaman Percobaan .....	54
6.1.3 Pengujian Unit Menampilkan Informasi Benda .....	55
6.1.4 Pengujian Unit Menggerakkan Benda.....	57
6.2 Pengujian Validasi .....	60
6.2.1 Kasus Uji Pengujian Validasi.....	60
6.2.2 Hasil Pengujian Validasi.....	62
6.3 Analisis Hasil Pengujian.....	63
6.3.1 Analisis Hasil Pengujian Unit .....	63
6.3.2 Analisis Hasil Pengujian Validasi .....	63
BAB 7 Penutup .....	64
7.1 Kesimpulan.....	64
7.2 Saran .....	64
DAFTAR PUSTAKA.....	65
LAMPIRAN .....	67



## DAFTAR TABEL

Tabel 2.1 Keterangan Hubungan Pada Use Case Diagram.....	12
Tabel 4.1 Identifikasi Aktor .....	24
Tabel 4.2 Daftar Kebutuhan Fungsional Sistem .....	24
Tabel 4.3 Use Case Lihat Halaman About .....	25
Tabel 4.4 Use Case Lihat Daftar Percobaan .....	26
Tabel 4.5 Use Case Menjalankan Halaman Percobaan.....	26
Tabel 4.6 Use Case Lihat Halaman Uji.....	26
Tabel 4.6 Use Case Lihat Halaman Uji (Lanjutan) .....	27
Tabel 4.7 Use Case Lihat Halaman Tutorial.....	27
Tabel 4.8 Use Case Lihat Informasi Benda .....	28
Tabel 4.9 Use Case Menggerakkan Benda .....	28
Tabel 4.10 Use Case Menjalankan Animasi .....	29
Tabel 4.11 Analisis Class Lihat Halaman About.....	30
Tabel 4.12 Analisis Class Lihat Daftar Percobaan.....	30
Tabel 4.13 Analisis Class Menjalankan Halaman Percobaan .....	30
Tabel 4.14 Analisis Class Lihat Halaman Uji .....	31
Tabel 4.15 Analisis Class Lihat Halaman Tutorial .....	31
Tabel 4.16 Analisis Class Lihat Informasi Benda .....	31
Tabel 4.17 Analisis Class Menggerakkan Benda .....	32
Tabel 4.18 Analisis Class Menjalankan Animasi .....	32
Tabel 4.19 Desain dan Rancangan Antarmuka .....	41
Tabel 4.19 Desain dan Rancangan Antarmuka (Lanjutan).....	42
Tabel 4.19 Desain dan Rancangan Antarmuka (Lanjutan 2) .....	43
Tabel 6.1 Pengujian Unit Main Button.....	52
Tabel 6.2 Kasus Uji Jalur Independen Main Button .....	53
Tabel 6.3 Pengujian Unit Menjalankan Halaman Percobaan .....	54
Tabel 6.4 Kasus Uji Jalur Independen Menjalankan Halaman Percobaan.....	55
Tabel 6.5 Pengujian Unit Menampilkan Informasi Benda .....	55
Tabel 6.6 Kasus Uji Jalur Independen Menampilkan Informasi Benda.....	57
Tabel 6.7 Pengujian Unit Menggerakkan Benda.....	57

Tabel 6.8 Kasus Uji Jalur Independen Menggerakkan Benda .....	59
Tabel 6.9 Kasus Uji Lihat Halaman About .....	60
Tabel 6.10 Kasus Uji Lihat Daftar Percobaan .....	60
Tabel 6.11 Kasus Uji Menjalankan Halaman Percobaan.....	60
Tabel 6.12 Kasus Uji Lihat Halaman Uji.....	61
Tabel 6.13 Kasus Uji Lihat Halaman Tutorial.....	61
Tabel 6.14 Kasus Uji Lihat Informasi Benda .....	61
Tabel 6.15 Kasus Uji Menggerakkan Benda .....	61
Tabel 6.16 Kasus Uji Menjalankan Animasi .....	62
Tabel 6.17 Hasil Pengujian Validasi.....	62
Tabel 6.17 Hasil Pengujian Validasi (Lanjutan) .....	63
Tabel B.1 Hasil Percobaan Materi Ikatan Kimia .....	70
Tabel B.2 Penggolongan larutan polar dan nonpolar .....	70
Tabel B.3 Hasil Percobaan Materi Larutan Elektrolit dan Non-Elektrolit .....	71
Tabel B.4 Penggolongan Larutan Elektrolit dan Non-Elektrolit .....	71



## DAFTAR GAMBAR

Gambar 2.1 Simbol Laboratorium Kimia.....	6
Gambar 2.2 Model Pengembangan Perangkat Lunak Waterfall .....	10
Gambar 2.3 Learning The Interface of Unity .....	14
Gambar 2.4 Project Browser Interface .....	15
Gambar 2.5 Hierarchy Interface.....	15
Gambar 2.6 Toolbar Interface.....	16
Gambar 2.7 Inspector Interface .....	17
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	19
Gambar 3.2 Diagram Alir Pemodelan Waterfall .....	22
Gambar 4.1 Diagram Use Case.....	25
Gambar 4.2 Diagram Class Sistem .....	33
Gambar 4.3 Diagram Activity Lihat Halaman About .....	34
Gambar 4.4 Diagram Activity Lihat Daftar Percobaan .....	34
Gambar 4.5 Diagram Activity Menjalankan Halaman Percobaan .....	35
Gambar 4.6 Diagram Activity Lihat Halaman Uji.....	35
Gambar 4.7 Diagram Activity Lihat Halaman Tutorial .....	36
Gambar 4.8 Diagram Activity Lihat Informasi Benda.....	36
Gambar 4.9 Diagram Activity Menggerakkan Benda.....	37
Gambar 4.10 Diagram Activity Menjalankan Animasi .....	38
Gambar 4.11 Diagram Sequence Lihat Halaman About .....	38
Gambar 4.12 Diagram Sequence Lihat Daftar Percobaan .....	39
Gambar 4.13 Diagram Sequence Menjalankan Halaman Percobaan.....	39
Gambar 4.14 Diagram Sequence Lihat Halaman Uji.....	39
Gambar 4.15 Diagram Sequence Lihat Halaman Tutorial.....	40
Gambar 4.16 Diagram Sequence Lihat Informasi Benda .....	40
Gambar 4.17 Diagram Sequence Menggerakkan Benda .....	41
Gambar 4.18 Diagram Sequence Menjalankan Animasi .....	41
Gambar 5.1 Halaman Utama .....	48
Gambar 5.2 About Screen .....	48
Gambar 5.3 Halaman Daftar Percobaan .....	49



Gambar 5.4 Halaman Percobaan 1 .....	49
Gambar 5.5 Halaman Percobaan 2 .....	49
Gambar 5.6 Halaman Percobaan 3 .....	50
Gambar 5.7 Halaman Uji Percobaan 1 .....	50
Gambar 5.8 Halaman Uji Percobaan 2 .....	50
Gambar 5.9 Halaman Tutorial Penggunaan Lab Percobaan .....	51
Gambar 6.1 Flow Graph Main Button .....	53
Gambar 6.2 Flow Graph Menjalankan Halaman Percobaan .....	54
Gambar 6.3 Flow Graph Menampilkan Informasi Benda .....	56
Gambar 6.4 Flow Graph Menggerakkan Benda .....	58
Gambar A.1 Cara Kerja Percobaan Ikatan Kimia .....	67
Gambar A.2 Cara Kerja Percobaan Larutan Elektrolit dan Non-Elektrolit .....	68
Gambar A.3 Cara Kerja Percobaan Hukum Proust .....	69
Gambar B.1 Tabel Periodik .....	71



## DAFTAR LAMPIRAN

LAMPIRAN A CARA KERJA PERCOBAAN .....	67
LAMPIRAN B EVALUASI PROGRAM .....	70



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Ilmu kimia merupakan ilmu yang mempelajari materi dan perubahannya. Sebagian besar ilmu kimia merupakan ilmu percobaan dan pengetahuannya diperoleh dari penelitian di laboratorium (Chang, 2005). Di Indonesia sendiri pelajaran ini merupakan salah satu pelajaran yang wajib dan menjadi salah satu nilai ukur untuk sebuah kelulusan bagi siswa siswi baik di SMA maupun SMP.

Di dalam kurikulum 2013 yang ditetapkan oleh Kementerian Pendidikan dan Kebudayaan, pelajaran kimia didukung oleh pembelajaran secara teori dan praktik. Khususnya untuk pelajar kelas X SMA, banyak materi kimia yang harus dihadirkan bersamaan dengan pengamatan teori secara langsung melalui media praktik guna membantu siswa agar dapat lebih memahami konsep dari materi (Departemen Pendidikan dan Kebudayaan, 2013).

Di dalam silabus pembelajaran kimia kelas X, kegiatan pembelajaran siswa diharuskan diikuti dengan pengumpulan data melalui percobaan selain hanya melalui teori. Beberapa contoh percobaan tersebut antara lain melakukan percobaan terkait kepolaran beberapa senyawa, percobaan daya hantar listrik pada beberapa larutan, percobaan reaksi pembakaran dan serah terima elektron, serta percobaan untuk membuktikan hukum Lavoisier dan Proust (Departemen Pendidikan dan Kebudayaan, 2013).

Seperti banyak diketahui pada umumnya percobaan biasa dilakukan didalam laboratorium, laboratorium merupakan sarana untuk melaksanakan kegiatan penelitian ilmiah. Namun tanpa disadari laboratorium kimia dengan segala kelengkapan peralatan dan bahan kimia didalamnya menjadikan laboratorium itu sendiri menjadi tempat yang dapat berpotensi menimbulkan bahaya kepada para penggunanya (Suwahono, 2010). Bahan kimia yang mudah terbakar, bersifat racun, sangat reaktif, serta tabung reaksi dan gelas kimia yang mudah pecah menjadi faktor yang memungkinkan terjadinya kecelakaan dalam laboratorium (Suwahono, 2010). Tentunya ini akan membahayakan siswa dan membuat pembelajaran menjadi kurang efisien.

Salah satu solusi dari permasalahan diatas dapat ditemukan dalam adaptasi *virtual laboratory*, yang terbukti dapat menjadi *tools* edukasi yang penting yang dapat mengatasi kurangnya pengalaman praktikan dalam melakukan percobaan (Courtiat, 2005). *Virtual laboratory* sebagai suatu produk inovasi merupakan penerapan teknologi baru untuk metodologi pembelajaran aktif, dalam rangka meningkatkan pembelajaran baik belajar mandiri maupun pemahaman siswa (Tejedor, 2008).

*Virtual laboratory* kimia sendiri telah digunakan sebagai simulasi percobaan pada materi yang abstrak dan sulit dipahami untuk mengatasi kurangnya sarana, alat dan bahan di laboratorium, mahalnnya alat dan zat-zat

kimia dengan bantuan komputer dan telah terbukti kebermanfaatannya sebesar 82,81% (Sunendar, 2007).

Dengan menggunakan *virtual laboratory* ini diharapkan dapat mengatasi permasalahan diatas. Karena itu penulis mengangkat permasalahan tersebut untuk sebuah penelitian dengan tujuan dapat membantu siswa-siswi SMA kelas X dalam pembelajaran kimia dan menciptakan media pembelajaran efisien yang dapat menunjang pemahaman konsep dari materi yang diajarkan.

## 1.2 Rumusan Masalah

Dilihat dari uraian latar belakang maka dapat dirumuskan permasalahan yang ada ialah:

1. Bagaimana merancang dan mengimplementasikan laboratorium kimia virtual untuk SMA kelas X.
2. Bagaimana menguji laboratorium kimia virtual untuk SMA kelas X sesuai dengan perancangan dan pengimplementasian sistem.

## 1.3 Tujuan

Tujuan dari penelitian perancangan ini secara umum adalah untuk merancang, membangun, mengimplementasi serta menguji laboratorium virtual kimia sesuai dengan materi pembelajaran SMA kelas X.

## 1.4 Manfaat

Berikut ini merupakan manfaat dari penelitian ini adalah:

1. Memudahkan siswa SMA kelas X melakukan percobaan sesuai materi pembelajarannya, tanpa harus melakukan secara langsung di laboratorium atau sekolah.
2. Meminimalisir kecelakaan di laboratorium dalam melakukan percobaan karena siswa melakukan percobaan dalam laboratorium virtual.
3. Meningkatkan efisiensi dalam melakukan percobaan yang terkendala alat dan bahan percobaan

## 1.5 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka perancangan pada penelitian ini dibatasi dalam hal:

1. Materi yang diangkat merupakan materi kimia SMA kelas X yang terdapat uji praktik atau percobaan dalam pengajaran materi tersebut.
2. Penggambaran objek dalam aplikasi sebatas tampilan objek asli sesuai dengan penglihatan mata manusia yang digambarkan kedalam tampilan 2D.
3. Aplikasi dikembangkan menggunakan *Game Engine* Unity dengan tampilan grafis 2D.

4. Materi yang diangkat sebanyak 3 pokok bahasan yang antara lain materi mengenai percobaan hukum kimia, elektrolit dan non-elektrolit, serta hukum proust.
5. Kebutuhan sistem diambil melalui studi literatur atau buku pelajaran kimia SMA Kelas X kurikulum 2013.

## 1.6 Sistematika Pembahasan

Penyusunan skripsi ini berdasarkan sistematika penulisan yang tersusun sebagai berikut:

### BAB 1 PENDAHULUAN

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat serta sistematika penulisan skripsi.

### BAB 2 LANDASAN KEPUTUSAN

Membahas kajian pustaka serta teori dasar dan teori penunjang yang berhubungan dengan pembahasan seputar percobaan kimia, *virtual laboratory*, pengujian dan teori lain yang digunakan dalam penelitian

### BAB 3 METODOLOGI

Membahas mengenai metode-metode yang digunakan selama penulisan.

### BAB 4 REKAYASA KEBUTUHAN SISTEM

Gambaran umum perancangan terkait analisa kebutuhan sistem dan diagram alur proses yang akan dbuat

### BAB 5 IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi dari perancangan sistem yang telah dijabarkan pada bab sebelumnya.

### BAB 6 PENGUJIAN DAN ANALISIS

Membahas mengenai proses pengujian sistem dan analisis berdasarkan hasil pengujian sistem.

### BAB 7 KESIMPULAN DAN SARAN

Pada bab ini memuat kesimpulan dari penelitian ini serta saran-saran untuk pengembangan penelitian selanjutnya.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Kajian pustaka merupakan penjelasan secara umum penelitian terdahulu yang berhubungan dengan topik yang dituliskan. Adapun penelitian terdahulu yang digunakan pada penelitian ini berjudul *An Online Virtual Laboratory of Electricity*, pada jurnal yang dituliskan oleh J.A. Gomez Tejedor, G. Molto Martinez, dan C. Barros Vidaurre ini dijelaskan mengenai *Java-based virtual laboratory* yang diakses melalui Internet dengan menggunakan *Web browser*. *Virtual laboratory* ini memungkinkan siswa membangun *direct* dan *alternating* sirkuit. Program mencakup antar muka grafis menyerupai papan koneksi, dan juga komponen kelistrikan dan alat-alat yang digunakan dalam laboratorium sesungguhnya untuk membangun sirkuit kelistrikan (Tejedor, 2008).

Laboratorium virtual ini dibangun menggunakan bahasa pemrograman java dan dibuat sebagai aplikasi yang dapat diakses melalui *web browser* dan berisikan simulasi sirkuit listrik. Dengan perangkat lunak simulasi ini siswa dimanjakan dengan alat yang berguna dan serbaguna untuk melakukan beberapa pelajaran praktis secara *online*, siswa dapat berlatih dalam lingkungan virtual sebelum melakukan percobaan pada laboratorium sesungguhnya dan bahkan dapat meningkatkan kemampuan mereka sebelum melakukan ujian (Tejedor, 2008).

Sedangkan perbedaan pada penelitian yang penulis buat terhadap penelitian sebelumnya terdapat pada topik yang diangkat yaitu materi pelajaran kimia, dimana topik difokuskan kepada materi pembelajaran kimia Sekolah Menengah Atas kelas X sedangkan pada penelitian *An Online Virtual Laboratory of Electricity* topik yang diangkat merupakan materi pada bidang kelistrikan atau fisika terkait pembuatan sirkuit kelistrikan. Selain itu penelitian kali ini menggunakan bahasa pemrograman C# dalam pengimplementasian sistem dan dibangun sebagai aplikasi yang diakses melalui *desktop*.

### 2.2 Landasan Teori

Landasan teori menuliskan berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian, landasan teori yang digunakan dalam penelitian ini antara lain terkait dengan:

#### 2.2.1 Ilmu Kimia

Ilmu kimia merupakan cabang ilmu pengetahuan alam (sains) yang berisi kajian tentang struktur, komposisi, sifat, dan perubahan materi serta energi yang mengikuti perubahan tersebut (Tim Guru Indonesia, 2011). Sejalan dengan pernyataan diatas kimia yang merupakan salah satu cabang dari IPA memberikan jawaban atas pertanyaan apa, mengapa, dan bagaimana fenomena alam yang berkaitan dengan komposisi, struktur, dan sifat, dinamika, dan energetik zat yang melibatkan keterampilan dan pelajaran (PR, 2007). Pada

penelitian ini diambil beberapa materi ilmu kimia yang terkait dengan materi siswa/siswi SMA kelas X seperti ikatan kimia, larutan elektrolit dan non-elektrolit, serta hukum proust. Adapun penjelasan dari materi-materi tersebut antara lain:

### **2.2.1.1 Ikatan Kimia**

Unsur-unsur biasanya ditemukan di alam dalam keadaan tidak stabil dan unsur-unsur tersebut cenderung untuk membentuk senyawa yang lebih stabil. Pembentukan senyawa ini terjadi melalui ikatan kimia. Ikatan kimia yang terdapat dalam senyawa dapat berupa ikatan ion atau ikatan kovalen. Garam dapur (NaCl) merupakan contoh dari senyawa yang dibentuk secara ikatan ion (Sunarya, 2009).

Dalam molekul diatom homointi, seperti  $H_2$ ,  $Cl_2$ ,  $N_2$ ,  $O_2$ , dan sejenisnya, kedua inti atom saling menarik pasangan elektron dengan ikatan sama besar sebab skala keelektronegatifan setiap atomnya sama. Jika dalam suatu ikatan kovalen terjadi pengkutuban muatan maka ikatan tersebut dinamakan ikatan kovalen polar. Molekul yang dibentuknya dinamakan *molekul polar*. Kepolaran molekul berkaitan dengan kemampuan suatu atom dalam molekul untuk menarik pasangan elektron ikatan ke arahnya. Kemampuan tersebut dinyatakan dengan skala keelektronegatifan. Selisih nilai keelektronegatifan dua buah atom yang berikatan kovalen memberikan informasi tentang ukuran kepolaran dari ikatan yang dibentuknya. Jika selisih keelektronegatifan nol atau sangat kecil, ikatan yang terbentuk cenderung kovalen murni. Jika selisihnya besar, ikatan yang terbentuk polar. Jika selisihnya sangat besar, berpeluang membentuk ikatan ion (Sunarya, 2009).

### **2.2.1.2 Larutan Elektrolit dan Non Elektrolit**

Larutan didefinisikan sebagai campuran dua atau lebih zat yang membentuk satu macam fasa (homogen) dan sifat kimia setiap zat yang membentuk larutan tidak berubah. Arti homogen menunjukkan tidak ada kecenderungan zat-zat dalam larutan terkonsentrasi pada bagian-bagian tertentu, melainkan menyebar secara merata di seluruh campuran. Bergantung pada sifat zat terlarut, ada larutan yang dapat menghantarkan listrik ada juga yang tidak dapat menghantarkan listrik. Larutan yang dapat menghantarkan arus listrik disebut larutan elektrolit, sedangkan larutan yang tidak dapat menghantarkan arus listrik disebut larutan nonelektrolit (Sunarya, 2009).

### **2.2.1.3 Hukum Proust**

Joseph Proust merupakan ilmuwan yang menemukan teori mengenai hukum proust, hukum proust sendiri mengatakan perbandingan massa unsur-unsur dalam senyawa adalah tetap. Lahirnya hukum proust sendiri didasari karena sifat senyawa yang terbentuk dari reaksi kimia unsur-unsur melalui reaksi pembentukan, sehingga senyawa mengandung unsur-unsur yang sama dengan unsur zat asal dengan perbandingan tertentu (Haris Watoni, 2013).

Perlu diingat bahwa massa suatu atom (atau massa atom rata-rata) dalam semua zat selalu tetap, karena massa atom merupakan sifat khas yang

membedakan satu atom dengan atom yang lain. Contoh, massa atom H = 1,0001 sma (dianggap = 1), massa atom N = 14, dan lain-lain. Yang berubah hanya komposisi zat ketika suatu zat mengalami perubahan (Haris Watoni, 2013). Contoh jika H<sub>2</sub> bereaksi sempurna dengan O<sub>2</sub> menjadi H<sub>2</sub>O menurut reaksi berikut:



Semua H dalam H<sub>2</sub>O berasal dari H<sub>2</sub> yang bereaksi dan semua O dalam H<sub>2</sub>O berasal dari O<sub>2</sub> yang bereaksi, sehingga

- Massa H (dalam H<sub>2</sub>O) = massa H (dalam H<sub>2</sub>)
- Massa O (dalam H<sub>2</sub>O) = massa O (dalam O<sub>2</sub>)

Dalam H<sub>2</sub>O :

$$\frac{\text{massa H}}{\text{massa O}} = \frac{i_{\text{H}} \times \text{massa atom H}}{i_{\text{O}} \times \text{massa atom O}} = \frac{2(1)}{1(16)} = 1 : 8$$

Oleh karena itu,

$$\frac{\text{massa H}_2 \text{ (yang bereaksi)}}{\text{massa O}_2 \text{ (yang bereaksi)}} = \frac{\text{massa H (dalam H}_2\text{O)}}{\text{massa O (dalam H}_2\text{O)}} = 1 : 8$$

Dengan  $i$  = angka indeks = jumlah atom unsur dalam molekul senyawa. Dalam H<sub>2</sub>O,  $i_{\text{H}} = 2$  dan  $i_{\text{O}} = 1$ .

Dari hasil diatas dapat ditarik pernyataan yang lebih jelas dari hukum perbandingan tetap (hukum proust), yaitu perbandingan massa unsur-unsur dalam senyawa = perbandingan massa atom-atom penyusun satu molekul senyawa (Haris Watoni, 2013).

### 2.2.2 Laboratorium Kimia

Di dalam laboratorium kimia, terdapat alat dan bahan eksperimen dimana alat dan bahan tersebut terkadang bersifat reaktif dan berbahaya, untuk itu dibutuhkan penalaran dan pengetahuan dini mengenai simbol dalam kemasan zat kimia guna menghindari kecelakaan dalam eksperimen (Tim Guru Indonesia, 2011). Beberapa simbol yang ada seperti terlihat pada Gambar 2.1



**Gambar 2.1 Simbol Laboratorium Kimia**

Sumber: (Tim Guru Indonesia, 2011 : 75)

Selain itu agar tidak terjadinya kecelakaan dalam melakukan eksperimen perlu diperhatikan beberapa hal (Tim Guru Indonesia, 2011), antara lain:

1. Mengetahui alat-alat praktikum kimia serta kegunaannya.
2. Mengetahui arti dari simbol-simbol yang terdapat pada laboratorium kimia.
3. Menggunakan alat pelindung, seperti jas laboratorium, sepatu, dan sarung tangan khusus karena bahan-bahan kimia dapat berbahaya.
4. Mematuhi aturan-aturan yang berlaku di laboratorium.
5. Melakukan langkah-langkah kerja secara beruntun.

Dalam laboratorium tentunya memiliki komponen-komponen khusus yang menggambarkan dan berguna untuk membantu dalam melakukan eksperimen. Komponen ini dapat berupa alat, material, ataupun senyawa yang berkaitan dan menunjang kinerja dalam melakukan eksperimen tersebut (Tim Guru Indonesia, 2011). Berikut adalah beberapa alat yang ada dan sering digunakan pada laboratorium kimia beserta fungsinya:

**1. Tabung reaksi**

Tabung reaksi berfungsi sebagai tempat untuk mereaksikan zat-zat kimia.

**2. Gelas ukur**

Gelas ukur berfungsi sebagai tempat untuk mengukur volume larutan.

**3. Rak tabung reaksi**

Rak tabung reaksi berfungsi sebagai tempat untuk meletakkan tabung reaksi.

**4. Gelas Beker**

Gelas beker berfungsi sebagai tempat mereaksikan dan menyimpan zat kimia.

**5. Erlenmeyer**

Erlenmeyer berfungsi sebagai penampung zat kimia yang dibuat dan juga untuk tempat mereaksikan zat-zat kimia.

**6. Labu alas bulat**

Labu alas bulat berfungsi sebagai tempat menyimpan zat-zat kimia.

**7. Penjepit**

Penjepit berfungsi sebagai penjepit tabung reaksi ketika melakukan reaksi kimia.

**8. Pembakar busen**

Pembakar busen berfungsi sebagai pemanas dalam reaksi kimia.

**9. Pipet**

Pipet berfungsi mengambil zat kimia berupa cairan dari wadahnya

### 2.2.3 Virtual Laboratory

Virtual laboratory seperti salinan aslinya, merupakan wadah bermain tempat untuk melakukan percobaan dan eksperimen. Virtual laboratory hadir dengan seperangkat objek yang berkaitan dengan materi ilmiah, peralatan untuk menjalankan objek, buku panduan, dan buku catatan. Buku panduan menjelaskan konsep melalui ilustrasi dan teks, dan menyajikan panduan untuk melakukan eksperimen (L. Mercer, 1990).

Daftar berikut menspesifikasikan fitur yang dianggap penting untuk pengoperasian lab (L. Mercer, 1990) :

1. *Consistent organization of the lab.* Dalam lingkungan laboratorium, eksperimen berjalan dengan menerapkan peralatan ke objek. Objek terdiri dari kumpulan beberapa file sehingga dapat dengan mudah diambil kembali. Format dari objek harus sesuai dengan standar untuk memungkinkan secara langsung mengimplementasikan operan standar seperti menyimpan dan menghapus objek.
2. *Inheritance of features.* Sering terjadi beberapa objek berbeda hanya pada detail. Mekanisme pewarisan harus digunakan untuk memanipulasi dan menyimpan objek secara efisien.
3. *Version control.* Interaksi dengan objek selama eksperimen dapat menghasilkan modifikasi secara permanen ataupun sementara. Dalam kasus terakhir, pengguna harus dapat menentukan apakah objek yang baru dibuat dapat menggantikan objek lama atau harus disimpan sebagai versi lain dari objek aslinya.
4. *Interactive manipulation of objects.* Laboratorium harus menyediakan seperangkat alat umum lab untuk memanipulasi parameter objek. Seperti, objek dapat dimodifikasi menggunakan *control panel* atau dengan mengubah wilayah tertentu pada deskripsi tekstual dari eksperimen.
5. *Flexibility in conducting experiments.* Pengguna harus dapat menerapkan alat-alat (program) ke objek (*data files*) dengan cara dinamis, ketika eksperimen sedang dilakukan. Ini dapat berlawanan dengan yang bersifat statis dimana objek secara langsung bergabung kedalam sistem.
6. *Guidance through the laboratory.* Laboratorium harus sudah memiliki sistem *hypertext* yang menggabungkan sekumpulan logika pada seperangkat objek, menyediakan deskripsi tekstual dari eksperimen, dan membuatnya dapat ditelusuri melalui eksperimen dalam banyak cara. Eksperimen yang lebih spesifik harus dapat dipanggil secara otomatis ketika teks yang sesuai dipilih, ini bertujuan untuk memfasilitasi peragaan dan membantu pengguna pemula.

## 2.2.4 Software Process Model

Sebuah model proses perangkat lunak adalah deskripsi yang disederhanakan dari proses perangkat lunak yang disajikan menjadi satu tampilan dari proses tersebut. Proses model dapat mencakup kegiatan yang merupakan bagian dari proses perangkat lunak, produk perangkat lunak, dan peran dari orang yang terlibat dalam rekayasa perangkat lunak (Sommerville, 2007). Berikut ini merupakan beberapa contoh jenis model proses perangkat lunak :

### 1. *Workflow Model*

Model ini menunjukkan urutan kegiatan dalam proses bersamaan dengan masukan, keluaran, dan ketergantungan. Kegiatan dalam model ini merepresentasikan tindakan manusia.

### 2. *Dataflow atau activity Model*

Model ini merepresentasikan proses sebagai serangkaian kegiatan, yang masing-masing melakukan beberapa transformasi data. Hal ini menunjukkan bagaimana input ke proses, seperti spesifikasi, ditransformasikan ke output, seperti desain. Kegiatan ini merepresentasikan transformasi yang dilakukan oleh orang-orang atau komputer.

### 3. *Role atau Action Model*

Model ini merepresentasikan peran dari orang-orang yang terlibat dalam proses perangkat lunak dan kegiatan yang menjadi tanggung jawabnya.

Kebanyakan model proses perangkat lunak didasarkan pada salah satu dari tiga model umum atau proses paradigma pengembangan perangkat lunak berikut (Sommerville, 2007) :

### 1. *Waterfall Model*

Pendekatan ini mengambil kegiatan proses dasar dari spesifikasi, pengembangan, validasi, dan evolusi perangkat lunak, serta mewakili proses tersebut sebagai tahap proses terpisah seperti spesifikasi kebutuhan, perancangan perangkat lunak, implementasi, pengujian dan seterusnya. Setelah setiap tahap didefinisikan itu adalah *sign-off* dan pembuatan perangkat lunak akan dilanjutkan ke tahap berikutnya.

### 2. *Evolutionary Development*

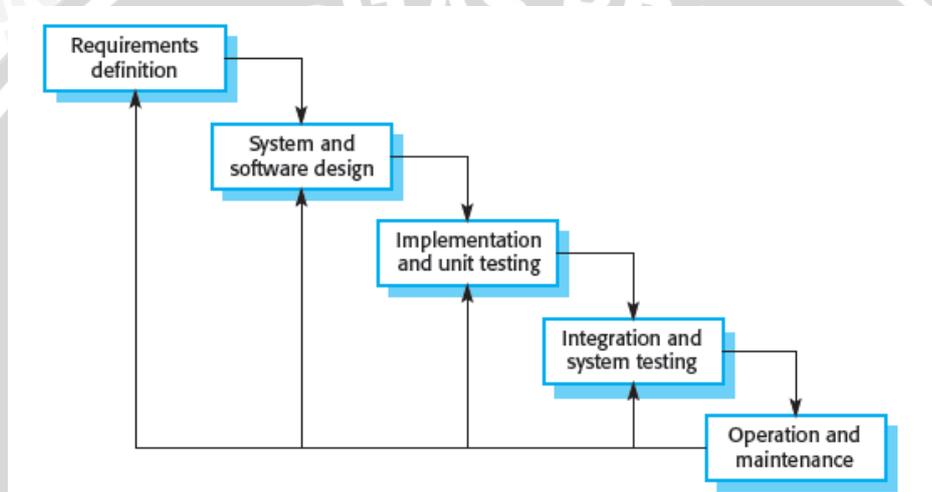
Pendekatan ini memisahkan kegiatan dari spesifikasi, pengembangan, dan validasi. Sistem awal dengan cepat dikembangkan dari spesifikasi abstrak. Hal ini kemudian disempurnakan dengan masukan pelanggan untuk menghasilkan sistem yang memenuhi kebutuhan pelanggan.

### 3. *Component-Based Software Engineering (CBSE)*

Pendekatan ini berdasarkan pada eksistensi dari sejumlah komponen yang dapat digunakan lagi. Proses pengembangan sistem berfokus pada mengintegrasikan komponen tersebut ke sistem daripada mengembangkan sistem dari awal

Ketiga model tersebut saat ini banyak digunakan dalam rekayasa perangkat lunak. Ketiga model tersebut tidak saling eksklusif dan sering digunakan bersama, terutama untuk pengembangan sistem yang besar. Sub-sistem dalam sistem yang lebih besar dapat dikembangkan dengan menggunakan pendekatan yang berbeda. Oleh karena itu, meskipun akan lebih mudah untuk membahas model ini secara terpisah, harus memahami bahwa dalam praktik, ketiga model tersebut sering digabungkan (Sommerville, 2007).

Dalam pembuatan perangkat lunak pada penelitian kali ini, pengembangan akan menggunakan model *waterfall*. Model *waterfall* adalah model yang diterbitkan pertama dari proses pengembangan perangkat lunak berasal dari proses rekayasa sistem yang lebih umum (Sommerville, 2007). Hal ini digambarkan pada Gambar 2.2



**Gambar 2.2. Model Pengembangan Perangkat Lunak Waterfall**  
Sumber: (Sommerville, 2007)

Berikut ini merupakan tahapan dari model yang digambarkan pada Gambar 2.3 (Sommerville, 2007) :

1. *Requirement analysis and definition*

Layanan, kendala dan tujuan sistem ditetapkan melalui konsultasi dengan pengguna sistem, kemudian didefinisikan secara rinci yang nantinya berfungsi sebagai spesifikasi sistem.

2. *System software and design*

Proses desain sistem membagi kebutuhan baik perangkat keras atau perangkat lunak sistem. Hal ini ditapkan pada keseluruhan arsitektur sistem. Desain perangkat lunak melibatkan identifikasi dan gambaran dasar abstraksi sistem perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

Selama tahap ini, desain perangkat lunak diwujudkan sebagai satu kumpulan program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

#### 4. *Integration and system testing*

Unit program atau program individu diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa kebutuhan perangkat lunak telah dipenuhi. Setelah pengujian, sistem perangkat lunak disampaikan kepada pelanggan.

#### 5. *Operation and maintenance*

Biasanya (meskipun tidak selalu dibutuhkan) merupakan tahap terpanjang dalam tahap siklus hidup perangkat lunak. Sistem ini dipasang dan diterapkan dalam penggunaan praktis. Pemeliharaan melibatkan pengecekan kesalahan yang tidak ditemukan pada tahap awal siklus hidup, meningkatkan implementasi unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan yang baru ditemukan.

Pada prinsipnya, hasil dari setiap tahap adalah satu atau lebih dokumen yang disetujui ('ditandatangani'). Tahap berikut ini tidak boleh dimulai sampai tahap sebelumnya selesai. Dalam prakteknya, tahap ini tumpang tindih dan memberikan informasi satu sama lain. Misalnya, selama proses desain, masalah dengan kebutuhan diidentifikasi (Sommerville, 2007).

Keuntungan dari model *waterfall* adalah bahwa dokumentasi diproduksi pada setiap tahap dan bahwa itu cocok dengan model proses rekayasa lainnya. Masalah utamanya adalah pembagian proyek menjadi tahap yang berbeda tidak fleksibel. Komitmen harus dibuat pada tahap awal dalam proses, yang membuatnya sulit untuk merespon perubahan kebutuhan pelanggan (Sommerville, 2007).

Oleh karena itu, model *waterfall* harus digunakan hanya ketika kebutuhan dipahami dengan baik dan tidak terjadi perubahan secara radikal selama pengembangan sistem. Namun, model *waterfall* mencerminkan jenis model proses yang digunakan dalam rekayasa perangkat lunak lainnya. Akibatnya, proses perangkat lunak berbasis pada pendekatan ini masih digunakan untuk mengembangkan perangkat lunak, terutama ketika proyek perangkat lunak merupakan bagian dari proyek rekayasa sistem yang lebih besar (Sommerville, 2007).

### 2.2.5 ***Unified Modeling Language (UML)***

UML adalah kependekan dari *Unified Modeling Language* yang merupakan suatu cara untuk menyelesaikan suatu masalah dengan mendeskripsikannya yang telah menjadi standar dalam dunia industri untuk memvisualisasikan, merancang dan mendokumentasikan sistem perangkat lunak. Dengan menggunakan UML kita dapat membangun model untuk segala bentuk dan jenis aplikasi perangkat lunak, yang mana aplikasi yang dibangun dapat berjalan pada perangkat lunak dengan sistem operasi dan jaringan apapun (Prayitno, 2005).

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 antara lain *Use Case Diagram*, *Class Diagram*, *State Diagram*, *Sequence Diagram*, *Collaboration Diagram*, *Activity Diagram*, *Component Diagram*, dan *Deployment Diagram* (Prayitno, 2005).

### 2.2.5.1 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” (Rosa A. S, 2013).

Syarat pemakaian *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian (Rosa A. S, 2013) :

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
- *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Beberapa simbol garis hubungan pada diagram *use case* yang sering digunakan adalah (Rosa A. S, 2013) :

**Table 2.1 Keterangan Hubungan Pada Use Case Diagram**

Gambar Garis Hubungan	Keterangan
	<i>Communicate</i> adalah garis yang menghubungkan antara aktor dengan <i>use case</i>
	<i>Extend</i> digunakan untuk menetapkan bahwa <i>use case</i> yang dituju memperpanjang perilaku dari sumber <i>use case</i> .
	<i>Include</i> digunakan 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 : (Rosa A. S, 2013)

### 2.2.5.2 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 (Prayitno, 2005) :

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



Attribut dan metode dalam *class diagram* dapat memiliki salah satu sifat seperti berikut di bawah ini (Prayitno, 2005) :

- *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 (Prayitno, 2005) :

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

#### 2.2.5.3 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* (Prayitno, 2005).

#### 2.2.5.4 Activity Diagram

*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 (Rosa A. S, 2013).

*Activity diagram* juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa A. S, 2013) :

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

## 2.2.6 Unity Game Engine

Unity merupakan mesin pengembangan untuk membangun permainan 2D, 3D, dan konten menarik lainnya. Dalam lingkungannya Unity didesain menjadi mesin *render* yang kuat dan terintegrasi dengan satu set perangkat lengkap dan *workflows* yang cepat untuk membangun konten 3D dan 2D yang interaktif, pengembangan permainan *multi platform* yang mudah, *assets* atau materi *game* yang siap digunakan pengguna (Unity, 2014).

Untuk pengembang dan studio *game* indie, Unity dapat menjadi satu pilihan yang tepat karena lebih dapat menghemat waktu dan biaya yang dibutuhkan dalam pembuatan *game* yang unik dan interaktif seperti halnya *game* profesional. Dengan Unity kita dapat membuat lingkungan atau suasana *game* sesuai dengan apa yang kita mau, serta menciptakan *game* yang menghubungkan dan menyenangkan bagi *player* pada segala *platform* (Unity, 2014).

Secara garis besar pada halaman antarmuka Unity terdiri dari beberapa komponen seperti *project browser*, *hierarchy*, *toolbar*, *scene view*, *game view*, dan *inspector* seperti yang terlihat pada Gambar 2.2. Komponen-komponen ini ada guna memudahkan pengguna dalam pembuatan *game* (Unity, 2014), berikut merupakan gambaran singkat serta penjelasan dari komponen tersebut.

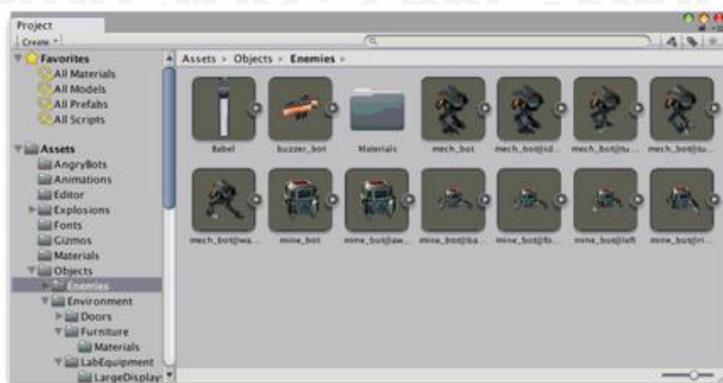


**Gambar 2.3 Learning the Interface of Unity**

Sumber: <http://docs.unity3d.com/Manual/LearningtheInterface.html>

### 1. Project Browser

Pada *project browser* pengguna dapat mengakses dan mengatur *assets* yang berkaitan pada *project*. Berikut tampilan dari *project browser* seperti pada Gambar 2.4



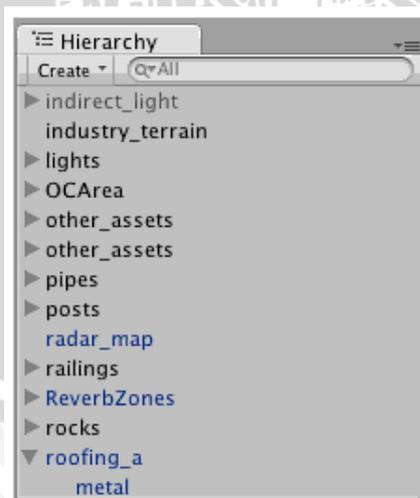
**Gambar 2.4 Project Browser Interface**

**Sumber:** <http://docs.unity3d.com/Manual/ProjectView.html>

Panel sebelah kiri pada *browser* menunjukkan struktur dari *project folder* dengan tampilan secara hirarki. Ketika *folder* dipilih oleh pengguna dari daftar yang ada, konten dari *folder* tersebut akan ditampilkan pada panel di sebelah kanan. *Assets* akan ditampilkan sebagai *icon* yang menggambarkan tipe *assets* (*script, material, sub-folder, dan lainnya*).

## 2. Hierarchy

*Hierarchy* berisi daftar dari setiap *GameObject* pada *scene* yang sedang kita buat saat ini. Beberapa dari *GameObject* merupakan *assets* langsung dari model 3D dan sebagian lainnya berupa *Prefabs*. Pada *hierarchy* kita dapat memilih secara langsung objek dan menggerakkan serta memindahkan ke objek lainnya untuk menjadikan objek lain tersebut sebagai *parent* dari objek pertama. Objek yang dimasukkan dan dihapus dari *scene view* secara langsung mempengaruhi daftar objek pada *hierarchy*. Tampilan *hierarchy* ditunjukkan pada Gambar 2.5



**Gambar 2.5 Hierarchy Interface**

**Sumber:** <http://docs.unity3d.com/Manual/Hierarchy.html>

### 3. *Toolbar*

*Toolbar* berisikan lima kontrol utama yang masing-masing kontrol berelasi dengan bagian dari Editor yang berbeda-beda. Tampilan *toolbar* ditunjukkan pada Gambar 2.6



**Gambar 2.6 *Toolbar Interface***

**Sumber:** <http://docs.unity3d.com/Manual/Toolbar.html>

### 4. *Scene View*

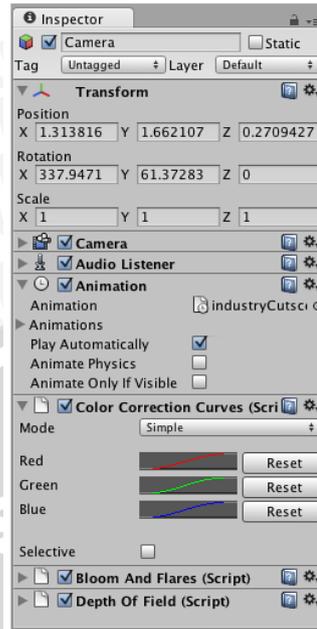
*Scene view* digunakan untuk memilih posisi lingkungan, pemain, kamera, musuh, dan *GameObjects* lainnya. Mengarahkan dan memanipulasi objek pada *scene view* merupakan salah satu fungsi penting dalam Unity. Unity juga menyediakan *keystrokes* untuk kegiatan-kegiatan paling umum guna memudahkan pengguna.

### 5. *Game View*

*Game view* merupakan hasil *render* dari kamera *game* yang dibuat. *Game view* memberikan gambaran akhir, *game* setelah diterbitkan. Untuk mengakses halaman *game view* dapat menggunakan tombol *Play Mode* pada *toolbar*, pada saat *play mode* diaktifkan semua perubahan yang dilakukan akan bersifat sementara dan akan dikembalikan keawal kembali, ketika pengguna keluar dari *play mode*.

### 6. *Inspector*

*Games* dalam Unity dibuat dari berbagai macam *GameObjects* yang terdiri dari *meshes*, *scripts*, *sounds*, atau elemen grafis lainnya seperti cahaya. *Inspector* menampilkan detail informasi mengenai *GameObject* yang sedang dipilih, informasi tersebut terkait semua *components* yang ada pada *GameObject* dan *properties* dari *GameObject* tersebut. Pada *inspector* pengguna dapat mengubah dan memodifikasi fungsi dari *GameObjects* yang terdapat pada scene. Semua properti yang ditampilkan dalam *inspector* dapat secara langsung dimodifikasi, bahkan variabel dari *script* yang dibuat dapat diubah tanpa harus memodifikasi langsung *script* tersebut. Tampilan dari *inspector* ditunjukkan pada Gambar 2.7



**Gambar 2.7 Inspector Interface**

Sumber: <http://docs.unity3d.com/Manual/Inspector.html>

## 2.2.7 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 ini menyediakan pendekatan sistematis untuk pengujian oleh developer. Terlebih lagi metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tertinggi untuk menemukan kesalahan-kesalahan dalam perangkat lunak. Teknik atau metode perancangan kasus uji yang digunakan adalah *black-box testing* dan *white-box testing* (Pressman, 2010).

### 2.2.7.1 White-Box Testing

*White box testing* adalah teknik pengujian yang berdasarkan jalur internal, struktur dan implementasi dari perangkat lunak yang sedang diuji. *White box testing* adalah teknik pengujian yang menggunakan struktur kontrol dari prosedur yang terdapat dalam perancangan untuk membuat kasus uji. Ada dua jenis pengujian yang termasuk *white box testing* yaitu *basis path testing* dan *control structure testing* (Pressman, 2010).

*Basis path testing* adalah teknik dalam metode *white box testing* untuk menyediakan desain *test case* yang dapat digunakan untuk menghitung kompleksitas desain prosedural untuk menghitung banyaknya jalur eksekusi. *Test case* tersebut digunakan untuk menjamin tiap *statement* telah dijalankan minimal satu kali tiap pengujian. Dalam konteks *basis path*, nilai yang dihitung di dalam *cyclomatic complexity* menjelaskan banyaknya jumlah jalur *independent* yang harus disediakan dalam pengujian *basis path* (Pressman, 2010).

### 2.2.7.2 Black-Box Testing

*Black box testing* adalah teknik pengujian yang menguji hanya berdasarkan kebutuhan dan spesifikasi. *Black box testing* juga disebut sebagai *behavioral testing* dan berfokus pada kebutuhan fungsi dari perangkat lunak (Pressman, 2010). Proses umum yang terjadi pada *black box testing* yaitu:

- a. Kebutuhan atau spesifikasi dianalisa terlebih dahulu.
- b. Penentuan input valid terpilih berdasarkan spesifikasi untuk menentukan perangkat lunak berjalan dengan benar. Input yang tidak valid juga harus dipilih untuk memverifikasi bahwa perangkat lunak dapat mendeteksinya dan menanganinya dengan baik.
- c. Penentuan output yang diharapkan sesuai dengan input yang telah dipilih.
- d. Pengujian dibuat dengan input yang telah dipilih.
- e. Pengujian dijalankan.
- f. Output yang sebenarnya dibandingkan dengan output yang diharapkan.
- g. Penentuan perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditentukan



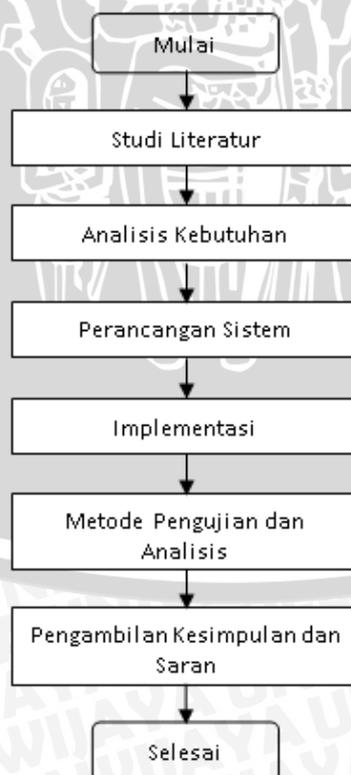
## BAB 3 METODOLOGI

### 3.1 Tipe Penelitian

Tipe penelitian pada penelitian ini merupakan tipe penelitian implementatif dimana penulis melakukan penelitian dengan merancang dan membangun sebuah sistem. Menurut Jogiyanto (2005:197), Rancang Bangun (desain) adalah tahapan setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan memiliki fungsi, serta termasuk mengkonfigurasi komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

### 3.2 Metode Penelitian

Metode penelitian yang dilakukan pada penelitian ini mengikuti metode penelitian pada penelitian rancang bangun sistem, ini dilakukan karena penelitian ini bertujuan untuk membangun sebuah aset teknologi informasi dimana metode yang digunakan diawali dengan pengumpulan data, analisa kebutuhan, perancangan sistem, implementasi sistem, pengujian serta analisis sistem, dan terakhir pengambilan kesimpulan dan saran. Adapun diagram alir dari metode penelitian ini antara lain sebagai berikut:



Gambar 3.1 Diagram Alir Metodologi Penelitian

### 3.3 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan penulis dalam penelitian ini adalah melakukan studi pustaka, dimana penulis mengumpulkan informasi dengan membaca dan mempelajari buku-buku, *e-book*, *e-journal*, serta melalui referensi terpercaya lain terkait teori-teori yang berkaitan dengan penulisan sebagai sumber acuan penulisan, perancangan, dan pengembangan sistem. Adapun teori yang diangkat dan dijadikan dasar teori dalam penulisan meliputi:

1. Pelajaran Kimia Kelas X Sekolah Menengan Atas, terkait:
  - a) Ikatan Kimia
  - b) Larutan Elektrolit dan Non-Elektrolit
  - c) Hukum Proust
2. Laboratorium Kimia dan *Virtual Laboratory*
3. Pemodelan Rekayasa Perangkat Lunak Secara *Waterfall*
4. UML (*Unified Modeling Language*)
  - a) *Use Case Diagram*
  - b) *Class Diagram*
  - c) *Sequence Diagram*
  - d) *Activity Diagram*
5. *Unity Game Engine*
6. Pengujian Sistem
  - a) *White-Box Testing*
  - b) *Black-Box Testing*

### 3.4 Metode Pengujian dan Analisis Data

Proses pengujian dilakukan melalui tahapan pengujian validasi. Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak atau belum. Pada pengujian validasi akan digunakan teknik pengujian *black box (Black Box Testing)*, pengujian validasi ini digunakan untuk menguji fungsional kerja dari sistem. Skenario dari pengujian ini dilakukan dengan menjalankan sistem sesuai dengan fungsi-fungsi internal yang dijabarkan pada perancangan, selain itu dilakukan pencatatan dalam bentuk tabel apakah hasil kebutuhan fungsional berhasil terpenuhi oleh sistem atau tidak.

Untuk analisis data pada penelitian dilakukan setelah proses pengujian sistem selesai dilakukan. Analisis data dilakukan dengan mengambil kesimpulan dari hasil pengujian, apakah semua fitur dapat berjalan sesuai fungsinya, apakah keluaran sesuai dengan harapan, dan apakah sistem dapat berjalan dengan semestinya ketika dijalankan pada perangkat bergerak yang berbeda.

### 3.5 Perangkat Penelitian

Dalam melakukan penelitian diperlukan adanya perangkat penelitian, perangkat ini digunakan sebagai sarana pendukung dalam melakukan penelitian. Perangkat yang digunakan dalam penelitian ini dibagi menjadi dua bagian, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras digunakan sebagai dasar persyaratan kebutuhan sistem yang dibangun, sedangkan untuk perangkat lunak merupakan kebutuhan dalam pembangunan dan perancangan sistem yang dibuat.

Adapun rincian perangkat penelitian untuk memenuhi persyaratan pembangunan dan perancangan sistem dalam penelitian ini antara lain:

1. Perangkat keras (*Hardware*)

*Notebook* Toshiba Satellite L645 dengan spesifikasi Intel(R) Core(TM) i3 2.27GHz 32 bit, RAM 2GB, VGA ATI Radeon 1GB.

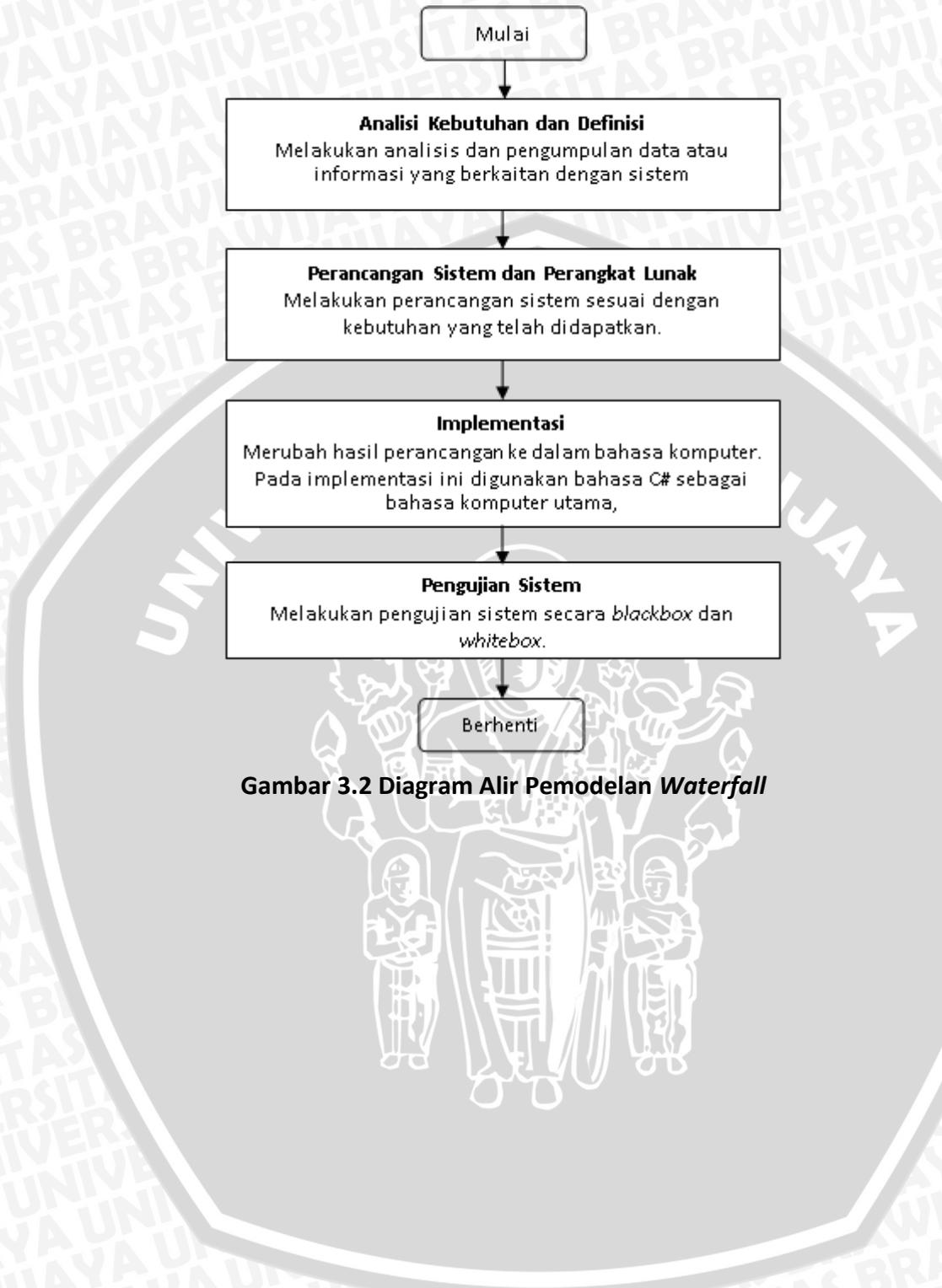
2. Perangkat lunak (*Software*)

perangkat lunak menggunakan *Unity Game Engine* untuk pembuatan sistem dengan menggunakan bahasa C#, serta *Adobe Photoshop CS5* untuk perancangan dan pembuatan objek dan animasi pada sistem.

### 3.6 Metode Pengembangan Sistem

Metode pengembangan sistem dilakukan untuk memudahkan proses pengembangan sistem dengan menggunakan pemodelan proses perangkat lunak. Pada penelitian ini, model yang akan digunakan adalah model sekuensial linier (*classic life cycle* atau *waterfall*), dimana proses pengembangan dengan model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial.

Proses pengembangan model ini dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan yang diawali dengan menganalisa kebutuhan, merancang, mengimplementasikan, dan menguji sistem. Selain itu, perancangan sistem juga dilakukan berdasarkan *Object oriented Analysis* dan *Object Oriented Design* dengan merancang antar muka pengguna, *use case*, *activity diagram*, dan *class diagram (Unified Modeleing Language)*. Adapun tahapan-tahapan dalam pembuatan model sekuensial linear dijelaskan pada Gambar 3.2.



Gambar 3.2 Diagram Alir Pemodelan *Waterfall*



## BAB 4 REKAYASA KEBUTUHAN SISTEM

Proses rekayasa kebutuhan sistem pada penelitian ini dibagi menjadi dua yaitu tahap analisis kebutuhan sistem dan tahap perancangan. Pada tahap analisis kebutuhan sistem dilakukan pembuatan kebutuhan fungsional dan non-fungsional sistem, serta pemodelan *use case* dan skenario *use case* berdasarkan pemodelan. Selanjutnya pada tahap perancangan dilakukan pembuatan perancangan *class diagram*, *activity diagram*, *sequence diagram*, dan antar muka pengguna.

### 4.1 Analisis Kebutuhan Sistem

Proses analisis kebutuhan diawali dengan penjabaran gambaran umum sistem, identifikasi aktor yang terlibat, penjabaran kebutuhan fungsional dan non-fungsional sistem dan kemudian memodelkannya ke dalam diagram *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

#### 4.1.1 Gambaran Umum Sistem

Laboratorium virtual kimia kelas X merupakan aplikasi virtual laboratorium berbasis *desktop* yang berisikan percobaan-percobaan kimia seperti halnya laboratorium sebenarnya yang diperuntukkan untuk siswa/siswi SMA kelas X. Sistem ini diharapkan dapat memberikan kemudahan serta membantu siswa/siswi dalam memahami materi yang diajarkan terkait bab/sub-bab yang terdapat percobaan didalamnya.

Materi percobaan dalam aplikasi ini dibuat mengikuti materi yg terdapat dalam kurikulum 2013 yang diperuntukkan bagi kalangan SMA/MA. Sedangkan materi yang diambil merupakan materi pelajaran yang hanya memiliki praktikum/percobaan didalamnya yang mampu diangkat untuk dijadikan kedalam bentuk virtual.

Materi percobaan yang terdapat dalam aplikasi sendiri terdiri dari tiga bagian yaitu percobaan terkait ikatan kimia, larutan elektrolit dan non-elektrolit, serta percobaan terkait pembuktian hukum proust.

##### a. Percobaan ikatan kimia

Dalam percobaan ikatan kimia pengguna akan melakukan percobaan yang mengharuskan menguji beberapa larutan yang telah disediakan merupakan larutan polar atau non-polar. Pengujian dilakukan dengan mendekatkan objek magnet yang ada kearah larutan yang mengalir.

##### b. Percobaan larutan elektrolit dan non-elektrolit

Dalam percobaan larutan elektrolit dan non-elektrolit pengguna akan menguji larutan yang telah disediakan merupakan larutan yang dapat menghantarkan arus listrik atau tidak. Pengujian dilakukan dengan

mengamati menyala atau tidaknya sirkuit lampu ketika penghantar dimasukkan kedalam larutan.

c. Percobaan hukum proust

Dalam percobaan hukum proust pengguna akan menguji benar atau tidaknya teori Proust melalui percobaan, dimana pengguna akan mengamati berat pita magnesium yang akan diuji sebelum dan sesudah dilakukan pembakaran.

#### 4.1.2 Identifikasi Aktor

Identifikasi aktor dilakukan untuk menjelaskan interaksi yang dilakukan aktor pada sistem. Tabel 4.1 memperlihatkan aktor yang terlibat beserta penjelasannya.

**Tabel 4.1 Identifikasi Aktor**

Aktor	Deskripsi
User	Pengguna yang menjalankan virtual lab dengan mengontrol objek dan mengerjakan aturan yang ada.

#### 4.1.3 Daftar Kebutuhan

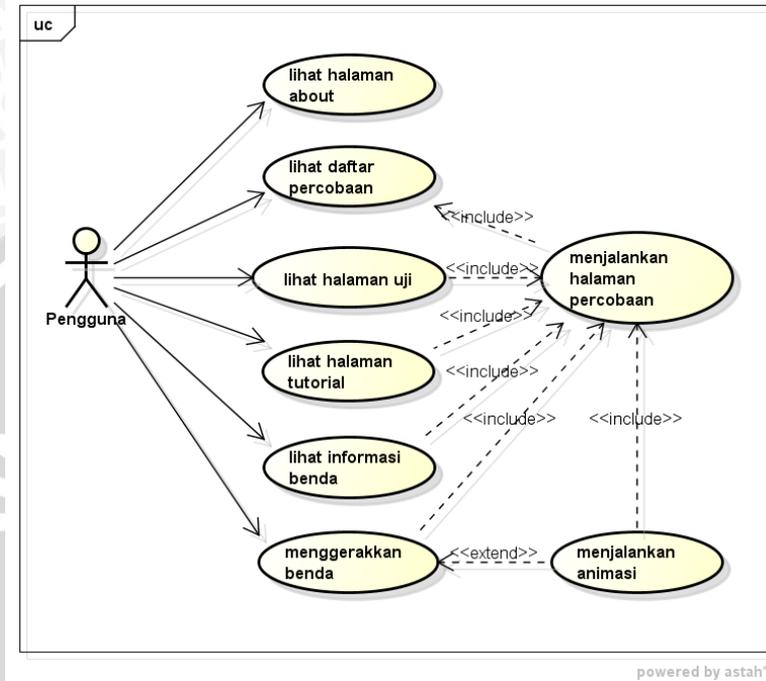
Daftar kebutuhan merupakan penjelasan dari kebutuhan fungsional yang ada pada aplikasi. Daftar kebutuhan fungsional sistem dispesifikasikan dengan kebutuhan fungsional *user* atau pengguna, daftar kebutuhan fungsional sistem ditunjukkan pada Tabel 4.2

**Tabel 4.2 Daftar Kebutuhan Fungsional Sistem**

ID	Kebutuhan	Use Case
SRS_01	Sistem harus mampu menyediakan pilihan about dan dapat menampilkan halaman about	Lihat Halaman About
SRS_02	Sistem harus mampu menyediakan pilihan menu mulai dan menampilkan halaman dari setiap percobaan yang tersedia dalam sistem.	Lihat Daftar Percobaan
SRS_03	Sistem harus dapat memulai percobaan dengan membuka halaman percobaan saat pemain memilih materi, sesuai dengan materi yang dipilih.	Menjalankan Halaman Percobaan
SRS_04	Sistem harus dapat menampilkan halaman uji saat berada pada halaman percobaan.	Lihat Halaman Uji
SRS_05	Sistem harus menyediakan fasilitas cara penggunaan laboratorium terkait percobaan yang dipilih pengguna.	Lihat Halaman <i>Tutorial</i>
SRS_06	Sistem harus dapat menampilkan benda sesuai dengan percobaan yang dijalankan dan menampilkan informasi dari sebagian benda.	Lihat Informasi Benda
SRS_07	Sistem harus dapat menggerakkan sebagian benda yang ditunjuk dalam halaman percobaan.	Menggerakkan Benda
SRS_08	Sistem harus dapat menjalankan animasi ketika terjadi <i>collision</i> atau ketika persyaratan terpenuhi.	Menjalankan Animasi

#### 4.1.4 Diagram Use Case

Diagram *use case* merupakan gambaran fungsionalitas dari sistem. Diagram ini dibuat berdasarkan identifikasi aktor dan identifikasi kebutuhan yang telah dilakukan sebelumnya dalam *use case* yang melibatkan pengguna sebagai aktor. Diagram *use case* sistem ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram Use Case

#### 4.1.5 Skenario Use Case

Skenario *use case* digunakan untuk menjelaskan secara detail setiap kebutuhan fungsional yang terdapat pada *use case* diagram. Detail dari setiap kebutuhan akan dijelaskan kedalam beberapa tabel antara lain:

Tabel 4.3 Use Case Lihat Halaman About

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_01
Nama	Lihat Halaman About
Tujuan	Menampilkan halaman <i>about</i> atau tentang perangkat
Deskripsi	Use Case ini menjelaskan bagaimana proses menampilkan halaman tentang perangkat ketika pengguna berada di halaman utama.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem telah berjalan dan menampilkan halaman utama aplikasi.
Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol <i>about</i> pada sistem	2. Sistem menampilkan halaman tentang perangkat atau <i>about</i>
Kondisi Akhir	Halaman tentang perangkat atau <i>about</i> sistem berhasil ditampilkan.

**Tabel 4.4 Use Case Lihat Daftar Percobaan**

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_02
Nama	Lihat Daftar Percobaan
Tujuan	Menampilkan halaman daftar seluruh percobaan yang terdapat pada sistem
Deskripsi	Use case ini menjelaskan bagaimana proses menampilkan daftar percobaan yang ada dalam aplikasi ketika pengguna berada pada halaman utama.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem telah berjalan dan menampilkan halaman utama aplikasi
Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol mulai yang ada pada halaman utama pada sistem.	2. Sistem menampilkan halaman daftar percobaan 3. Sistem menampilkan seluruh daftar percobaan yang ada pada sistem terkait judul beserta deskripsi singkat mengenai percobaan yang ditampilkan.
Kondisi Akhir	Halaman daftar percobaan berhasil ditampilkan dan dijalankan.

**Tabel 4.5 Use Case Menjalankan Halaman Percobaan**

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_03
Nama	Menjalankan Halaman Percobaan
Tujuan	Menjalankan dan membuka halaman percobaan yang dipilih oleh pengguna
Deskripsi	Use case ini menjelaskan bagaimana proses sistem dalam menjalankan halaman percobaan yang dipilih pengguna.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem telah berjalan dan menampilkan halaman daftar percobaan.
Aksi Aktor	Reaksi Sistem
1. Pengguna memilih percobaan yang ada pada daftar percobaan	2. Sistem menampilkan judul beserta deskripsi singkat mengenai percobaan yang dipilih.
3. Pengguna menekan tombol masuk ke halaman percobaan	4. Sistem menjalankan halaman percobaan sesuai yang dipilih pengguna
Kondisi Akhir	Halaman percobaan yang dipilih pengguna berhasil ditampilkan dan dijalankan.

**Tabel 4.6 Use Case Lihat Halaman Uji**

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_04
Nama	Lihat Halaman Uji
Tujuan	Menampilkan halaman uji percobaan 1 dan percobaan 2

**Tabel 4.6 Use Case Lihat Halaman Uji (Lanjutan)**

Skenario Kasus Pada Sistem	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana proses menampilkan halaman uji percobaan 1 dan percobaan 2 ketika pengguna akan memulai pengujian.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem telah berjalan dan menampilkan halaman percobaan 1 atau percobaan 2, serta variabel benda uji pada tiap percobaan tidak bernilai null atau kosong.
Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol kiri <i>mouse</i> pada buret di percobaan 1 atau menekan tombol kanan <i>mouse</i> saat terjadi <i>collision</i> pada gelas kimia dan rangkaian pada percobaan 2	2. Sistem mengecek variabel buret pada percobaan 1 atau gelas kimia pada percobaan 2 3. Sistem menampilkan halaman uji jika variabel buret pada percobaan 1 atau gelas kimia pada percobaan 2 tidak null atau kosong
<b>Alur Alternatif 1 : Jika variabel <i>collision</i> gelas kimia dan rangkaian bernilai salah</b>	
	4. Sistem menggerakkan rangkaian mengikuti arah pergerakan <i>mouse</i> yang dilakukan pengguna.
<b>Alur Alternatif 2 : Jika variabel benda bernilai null atau kosong</b>	
	5. Sistem tidak menampilkan halaman uji
Kondisi Akhir	Sistem berhasil menampilkan halaman uji

**Tabel 4.7 Use Case Lihat Halaman Tutorial**

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_05
Nama	Lihat Halaman <i>Tutorial</i>
Tujuan	Menampilkan halaman <i>tutorial</i> dari percobaan yang dijalankan
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana proses menampilkan halaman <i>tutorial</i> saat pengguna ingin menampilkan halaman bantuan percobaan sesuai dengan halaman percobaan yang sedang dijalankan.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem telah berjalan dan menampilkan halaman percobaan
Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol bantuan pada halaman percobaan	2. Sistem menampilkan halaman bantuan atau <i>tutorial</i> 3. Sistem menampilkan alur kerja cara menjalankan percobaan
4. Pengguna mendekati <i>cursor mouse</i> ke salah satu langkah kerja percobaan	5. Sistem menampilkan langkah kerja berupa gambar sesuai dengan langkah kerja yang dipilih pengguna.
Kondisi Akhir	Sistem berhasil menampilkan halaman <i>tutorial</i> percobaan.

**Tabel 4.8 Use Case Lihat Informasi Benda**

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_06
Nama	Lihat Informasi Benda
Tujuan	Menampilkan informasi singkat terkait benda pada halaman percobaan
Deskripsi	Use case ini menjelaskan bagaimana proses menampilkan informasi dari sebagian benda yang dilakukan oleh pengguna pada halaman percobaan.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem telah berjalan dan menampilkan halaman percobaan
Aksi Aktor	Reaksi Sistem
1. Pengguna mendekati <i>cursor mouse</i> kearah benda dan meletakkan <i>cursor</i> diatas benda.	2. Sistem mengecek letak posisi <i>cursor</i> 3. Sistem mengecek benda memiliki informasi atau tidak 4. Sistem menampilkan informasi singkat terkait benda pada halaman percobaan jika benda yang dipilih memiliki informasi untuk ditampilkan
Alur Alternatif 1 : Jika benda tidak memiliki informasi benda	
	6. Sistem tidak menampilkan informasi apapun.
Kondisi Akhir	Sistem berhasil menampilkan informasi terkait benda

**Tabel 4.9 Use Case Menggerakkan Benda**

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_07
Nama	Menggerakkan Benda
Tujuan	Menggerakkan dan menampilkan pergerakan benda pada halaman percobaan
Deskripsi	Use case ini menjelaskan bagaimana proses sistem dalam menggerakkan benda ketika pengguna memilih menggerakkan benda.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem telah berjalan dan menampilkan halaman percobaan
Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol kiri <i>mouse</i> pada benda	2. Sistem mengecek letak penekanan yang dilakukan pengguna 3. Sistem memposisikan benda yang dipilih sesuai letak <i>cursor</i>
4. Pengguna menggerakkan <i>mouse</i> dengan tetap menekan benda	5. Sistem menggerakkan benda sesuai dengan pergerakan <i>mouse</i>
6. Pengguna melepas penekanan tombol <i>mouse</i> pada benda	7. Sistem mengembalikan benda ke posisi awal benda berada.
Kondisi Akhir	Sistem berhasil menggerakkan benda pada halaman percobaan sesuai pergerakan <i>mouse</i> yang diarahkan pengguna.

**Tabel 4.10 Use Case Menjalankan Animasi**

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_08
Nama	Menjalankan Animasi
Tujuan	Menjalankan animasi benda ketika persyaratan terpenuhi
Deskripsi	Use case ini menjelaskan bagaimana proses sistem dalam menjalankan animasi benda ketika persyaratan menjalankan animasi terpenuhi.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Sistem telah berjalan dan pengguna menggerakkan benda pada halaman percobaan
Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol kiri <i>mouse</i> pada benda dan menggerakkan benda	2. Sistem menggerakkan benda sesuai dengan pergerakan <i>mouse</i>
3. Pengguna menggerakkan benda ke arah benda lain yang termasuk persyaratan menjalankan animasi.	4. Sistem mengecek variabel dari benda 5. Sistem mengecek <i>collision</i> yang terjadi pada benda dan benda tujuan 6. Sistem menjalankan animasi yang terdapat pada benda ketika variabel benda dan <i>collision</i> bernilai benar
Alur Alternatif 1 : Jika variabel benda dan <i>collision</i> tidak bernilai benar	
	7. Sistem menggerakkan benda pada sesuai dengan pergerakan <i>mouse</i> yang dilakukan pengguna
Kondisi Akhir	Sistem berhasil menjalankan animasi benda

#### 4.1.6 Analisis Class

Analisis *class* dilakukan untuk menganalisa *class-class* yang terdapat dalam *use case* yang telah dibuat sebelumnya, serta menentukan *class-class* yang termasuk kedalam *boundary*, *controller*, dan *entity* untuk selanjutnya dilakukan pemodelan aliran jalannya proses interaksi antar objek dan *class* yang disusun berdasarkan urutan waktu. Analisis *class* dari setiap *use case* ditunjukkan dalam beberapa tabel antara lain:

**Tabel 4.11 Analisis Class Lihat Halaman About**

Nomor Use Case	SRS_01
Nama	Lihat Halaman About
<b>Nama Boundary</b>	<b>Penjelasan</b>
 MainPage	Halaman utama dari sistem
<b>Nama Control</b>	<b>Penjelasan</b>
 MainButtonScript	Controller data antara boundary dan entity
<b>Nama Entity</b>	<b>Penjelasan</b>
 GameObject	Daftar Id objek yang ada pada sistem

**Tabel 4.12 Analisis Class Lihat Daftar Percobaan**

Nomor Use Case	SRS_02
Nama	Lihat Daftar Percobaan
<b>Nama Boundary</b>	<b>Penjelasan</b>
 MainPage	Halaman Utama dari sistem
<b>Nama Control</b>	<b>Penjelasan</b>
 MainButtonScript	Controller data antara boundary dan entity
<b>Nama Entity</b>	<b>Penjelasan</b>
 GameObject	Daftar Id objek yang ada pada sistem

**Tabel 4.13 Analisis Class Menjalankan Halaman Percobaan**

Nomor Use Case	SRS_03
Nama	Menjalankan Halaman Percobaan
<b>Nama Boundary</b>	<b>Penjelasan</b>
 GameObject	Daftar Id objek yang ada pada sistem
<b>Nama Control</b>	<b>Penjelasan</b>
 StageScript	Control untuk menjalankan scene yang ditunjuk pada halaman daftar percobaan
<b>Nama Entity</b>	<b>Penjelasan</b>
 Application	Class untuk membuka scene percobaan

**Tabel 4.14 Analisis Class Lihat Halaman Uji**

Nomor <i>Use Case</i>	SRS_04
Nama	Lihat Halaman Uji
<b>Nama Boundary</b>	<b>Penjelasan</b>
 SceneManager	Daftar scene yang ada pada sistem
<b>Nama Control</b>	<b>Penjelasan</b>
 BgWindowComponent	Control untuk tampilan halaman uji
<b>Nama Entity</b>	<b>Penjelasan</b>
 GameObject	Daftar Id objek yang ada pada sistem

**Tabel 4.15 Analisis Class Lihat Halaman Tutorial**

Nomor <i>Use Case</i>	SRS_05
Nama	Lihat Halaman Tutorial
<b>Nama Boundary</b>	<b>Penjelasan</b>
 SceneManager	Daftar scene yang ada pada sistem
<b>Nama Control</b>	<b>Penjelasan</b>
 icMenuScript	Control untuk menjalankan <i>menu tutorial</i>
<b>Nama Entity</b>	<b>Penjelasan</b>
 GameObject	Daftar Id objek yang ada pada sistem

**Tabel 4.16 Analisis Class Lihat Informasi Benda**

Nomor <i>Use Case</i>	SRS_06
Nama	Lihat Informasi Benda
<b>Nama Entity</b>	<b>Penjelasan</b>
 GameObject	Daftar Id objek yang ada pada sistem
<b>Nama Entity</b>	<b>Penjelasan</b>
 AppTipsDetector	Control class untuk mendeteksi persyaratan menampilkan benda
<b>Nama Entity</b>	<b>Penjelasan</b>
 AppTips	Daftar data dari informasi yang dimiliki benda

**Tabel 4.17 Analisis Class Menggerakkan Benda**

Nomor Use Case	SRS_07
Nama	Menggerakkan Benda
<b>Nama Entity</b>	<b>Penjelasan</b>
 GameObject	Daftar Id objek yang ada pada sistem
<b>Nama Entity</b>	<b>Penjelasan</b>
 Move	Control class untuk menggerakkan benda yang ditunjuk
<b>Nama Entity</b>	<b>Penjelasan</b>
 OnMouseDown	Entity berupa fungsi untuk menggerakkan benda

**Tabel 4.18 Analisis Class Menjalankan Animasi**

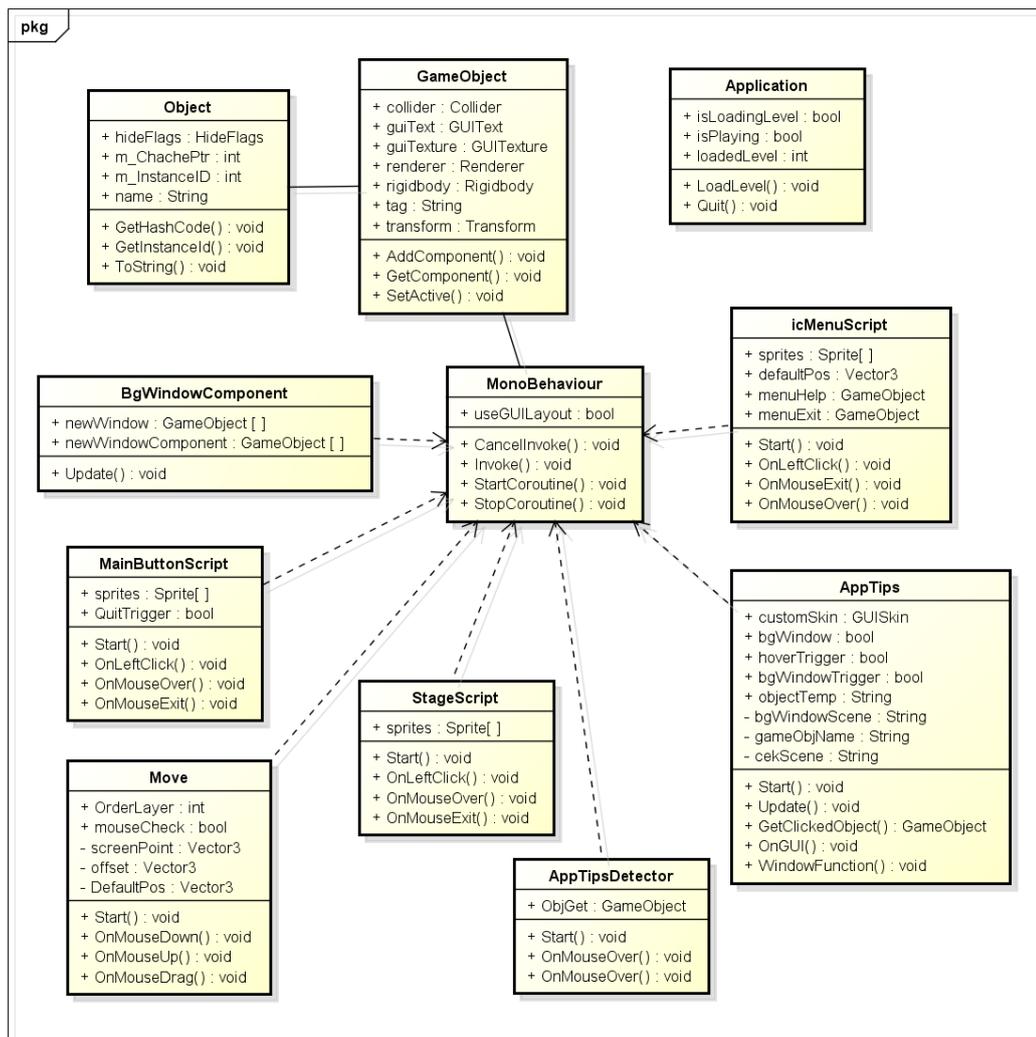
Nomor Use Case	SRS_08
Nama	Menjalankan Animasi
<b>Nama Boundary</b>	<b>Penjelasan</b>
 GameObject	Daftar Id objek yang ada pada sistem
<b>Nama Control</b>	<b>Penjelasan</b>
 AnimatorController	Controller animasi untuk menjalankan animasi benda
<b>Nama Entity</b>	<b>Penjelasan</b>
 Animator	Daftar id dari animasi benda yang ada pada sistem

## 4.2 Perancangan Aplikasi

Proses perancangan aplikasi dilakukan berdasarkan analisis kebutuhan yang telah dilakukan. Perancangan dilakukan dalam empat tahap, yaitu pemodelan *class diagram*, perancangan *activity diagram*, perancangan *sequence diagram*, dan terakhir perancangan antarmuka.

### 4.2.1 Perancangan Class Diagram

Diagram *class* memberikan gambaran pemodelan elemen-elemen *class* yang membentuk sebuah sistem. Selain itu, diagram *class* juga menggambarkan relasi-relasi *class* pada sistem. *Class* bisa didapat dengan menganalisis secara detail terhadap *use case* yang dimodelkan. Berikut ini merupakan rancangan diagram *class* sistem yang akan dijelaskan pada Gambar 4.5



powered by astah

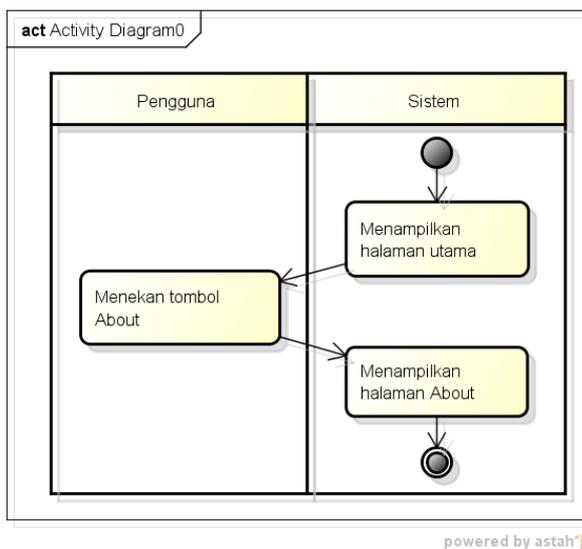
Gambar 4.2 Diagram Class Sistem

## 4.2.2 Diagram Activity

Diagram *activity* digunakan untuk menjelaskan urutan dari aktivitas dengan menggambarkan langkah demi langkah. Diagram *activity* menggambarkan alur kerja mulai dari *starting point* hingga *finish point* dengan memberikan detail alur percabangan yang terdapat pada suatu *event* dalam aktivitas.

### 4.2.2.1 Diagram Activity Lihat Halaman About

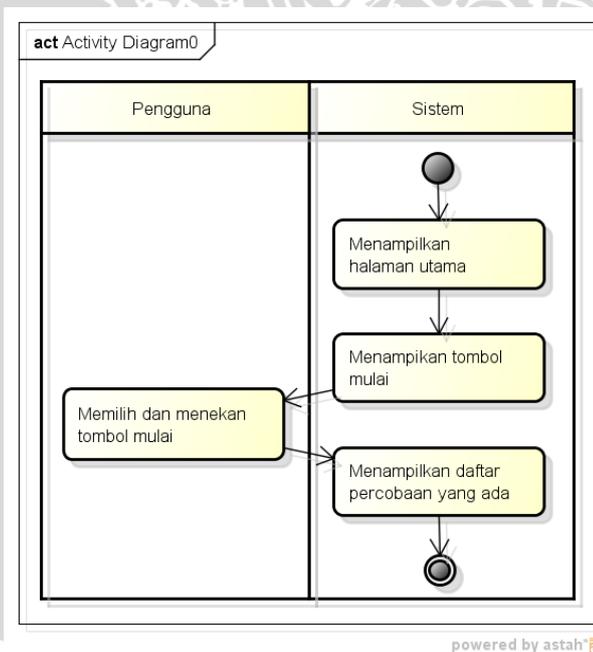
Diagram *activity* lihat halaman *about* menggambarkan alur proses saat pengguna akan menampilkan atau membuka halaman *about*. Diagram *activity* lihat halaman *about* ditampilkan pada Gambar 4.3



Gambar 4.3 Diagram Activity Lihat Halaman About

#### 4.2.2.2 Diagram Activity Lihat Daftar Percobaan

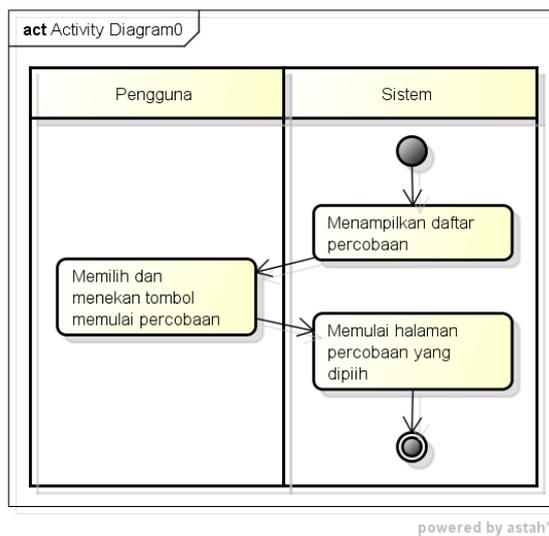
Diagram *activity* lihat daftar percobaan menggambarkan alur proses saat pengguna akan menampilkan atau membuka halaman daftar percobaan. Diagram *activity* lihat daftar percobaan ditampilkan pada Gambar 4.4



Gambar 4.4 Diagram Activity Lihat Daftar Percobaan

#### 4.2.2.3 Diagram Activity Menjalankan Halaman Percobaan

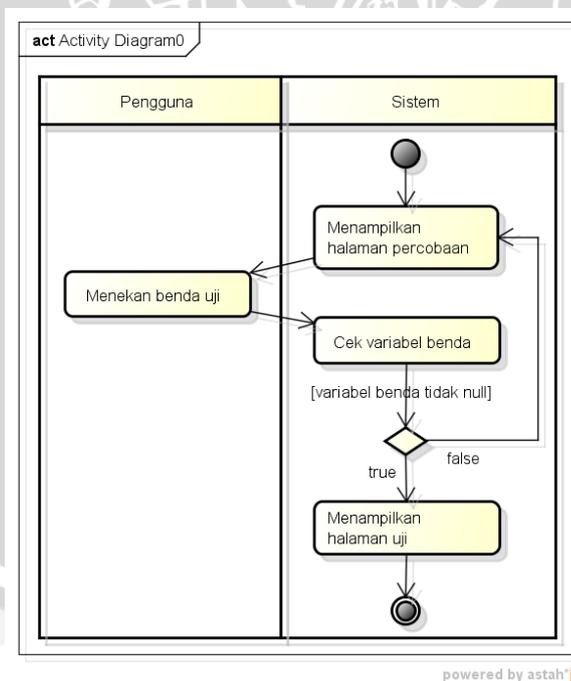
Diagram *activity* menjalankan halaman percobaan menggambarkan alur proses saat pengguna akan menampilkan atau memulai halaman percobaan. Diagram *activity* menjalankan halaman percobaan ditampilkan pada Gambar 4.5



Gambar 4.5 Diagram Activity Menjalankan Halaman Percobaan

#### 4.2.2.4 Diagram Activity Lihat Halaman Uji

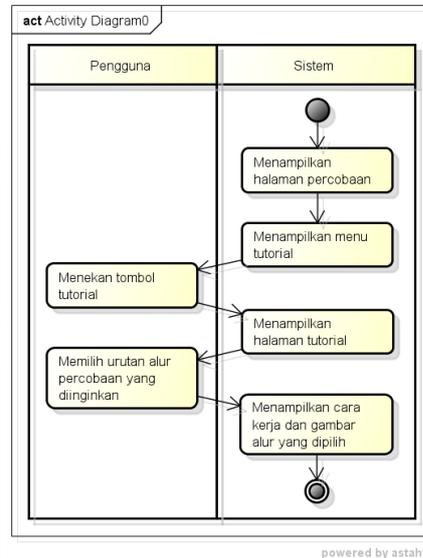
Diagram *activity* lihat daftar percobaan menggambarkan alur proses saat pengguna akan menampilkan atau membuka halaman uji pada saat berada pada halaman percobaan. Diagram *activity* lihat halaman uji ditampilkan pada Gambar 4.6



Gambar 4.6 Diagram Activity Lihat Halaman Uji

#### 4.2.2.5 Diagram Activity Lihat Halaman Tutorial

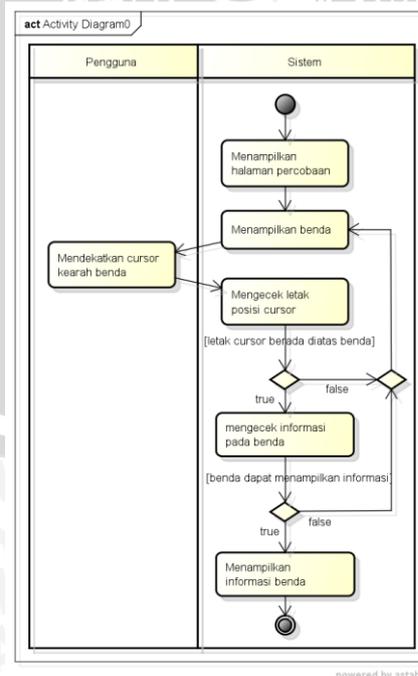
Diagram *activity* lihat halaman *tutorial* menggambarkan alur proses saat pengguna akan menampilkan atau membuka halaman *tutorial* atau cara kerja dari percobaan yang sedang dijalankan pada saat berada pada halaman percobaan. Diagram *activity* lihat halaman *tutorial* ditampilkan pada Gambar 4.7



Gambar 4.7 Diagram Activity Lihat Halaman Tutorial

#### 4.2.2.6 Diagram Activity Lihat Informasi Benda

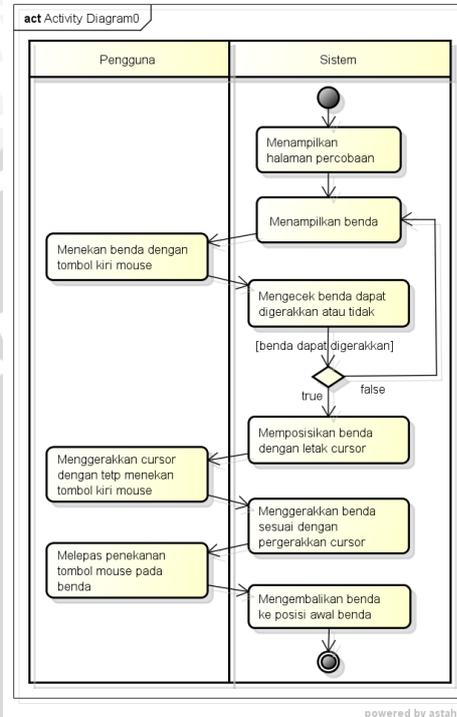
Diagram *activity* lihat informasi benda menggambarkan alur proses saat pengguna akan menampilkan informasi dari benda laboratorium pada saat berada pada halaman percobaan. Diagram *activity* lihat informasi benda ditampilkan pada Gambar 4.8



Gambar 4.8 Diagram Activity Lihat Informasi Benda

#### 4.2.2.7 Diagram Activity Menggerakkan Benda

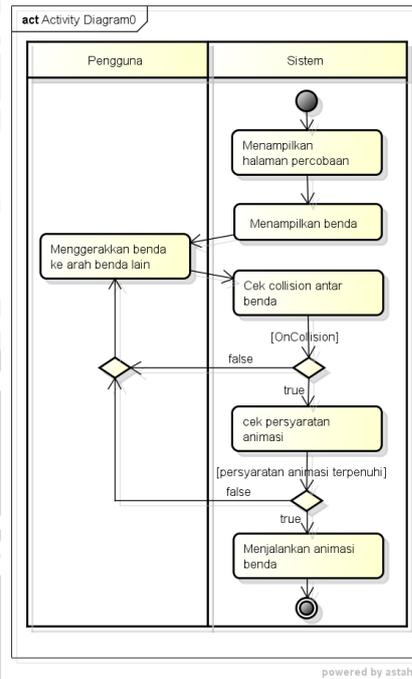
Diagram *activity* menggerakkan benda menggambarkan alur proses saat pengguna akan menggerakkan benda yang ditunjuk pada saat berada pada halaman percobaan. Diagram *activity* menggerakkan benda ditampilkan pada Gambar 4.9



Gambar 4.9 Diagram Activity Menggerakkan Benda

#### 4.2.2.8 Diagram Activity Menjalankan Animasi

Diagram *activity* menjalankan animasi menggambarkan alur proses saat pengguna melakukan aksi pada benda yang menjalankan animasi benda pada saat berada pada halaman percobaan. Diagram *activity* lihat halaman uji ditampilkan pada Gambar 4.10



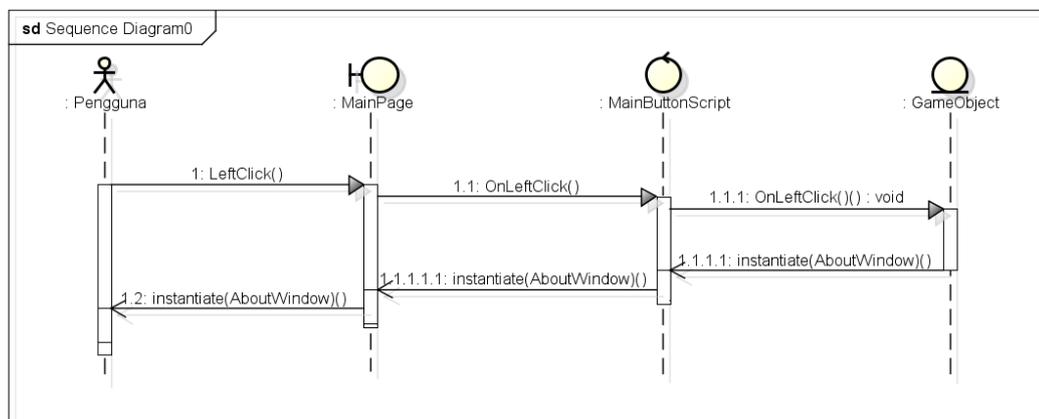
Gambar 4.10 Diagram Activity Menjalankan Animasi

### 4.2.3 Diagram Sequence

Diagram *sequence* menunjukkan pemodelan aliran jalannya proses interaksi antar objek dan *class* yang disusun berdasarkan urutan waktu. Diagram *sequence* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai tanggapan dari *event* untuk menghasilkan *output* tertentu.

#### 4.2.3.1 Diagram Sequence Lihat Halaman About

Diagram *sequence* lihat halaman *about* merupakan diagram *sequence* ketika pengguna membuka halaman *about*. Perancangan diagram *sequence* lihat halaman *about* dapat dilihat pada Gambar 4.11

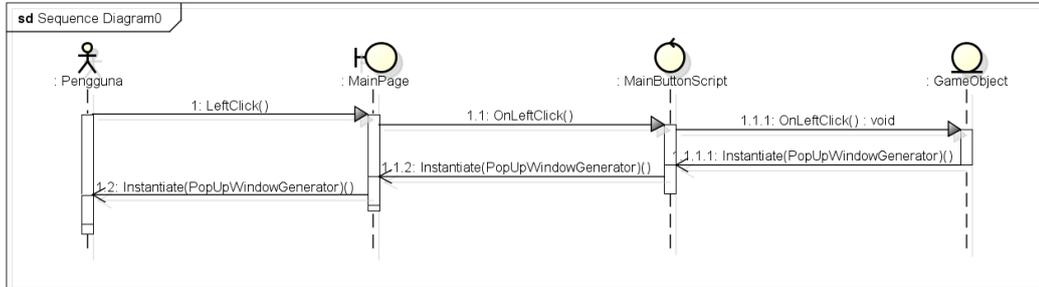


Gambar 4.11 Diagram Sequence Lihat Halaman About



#### 4.2.3.2 Diagram Sequence Lihat Daftar Percobaan

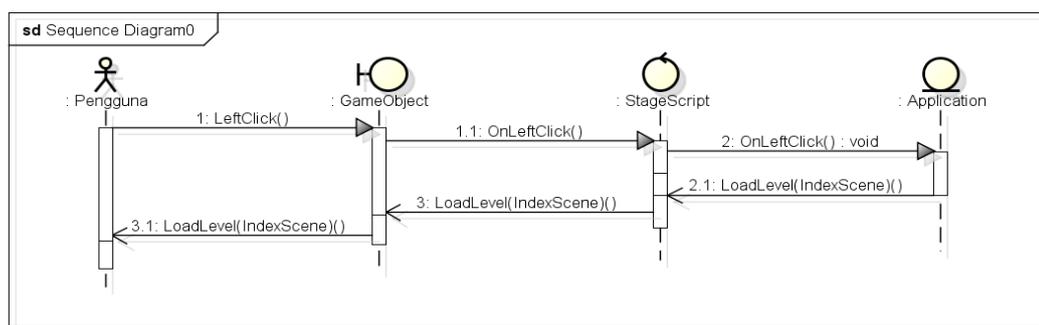
Diagram *sequence* lihat halaman daftar percobaan merupakan diagram *sequence* ketika pengguna membuka halaman daftar percobaan. Perancangan diagram *sequence* lihat daftar percobaan dapat dilihat pada Gambar 4.12



Gambar 4.12 Diagram Sequence Lihat Daftar Percobaan

#### 4.2.3.3 Diagram Sequence Menjalankan Halaman Percobaan

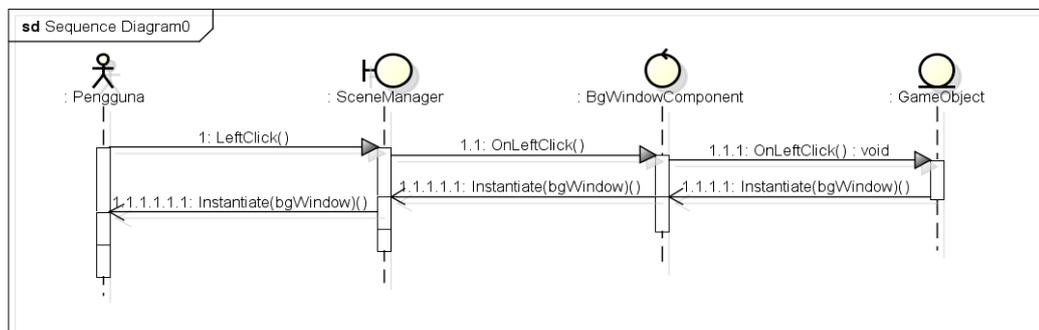
Diagram *sequence* menjalankan halaman percobaan merupakan diagram *sequence* ketika pengguna menjalankan halaman percobaan. Perancangan diagram *sequence* menjalankan halaman percobaan dapat dilihat pada Gambar 4.13



Gambar 4.13 Diagram Sequence Menjalankan Halaman Percobaan

#### 4.2.3.4 Diagram Sequence Lihat Halaman Uji

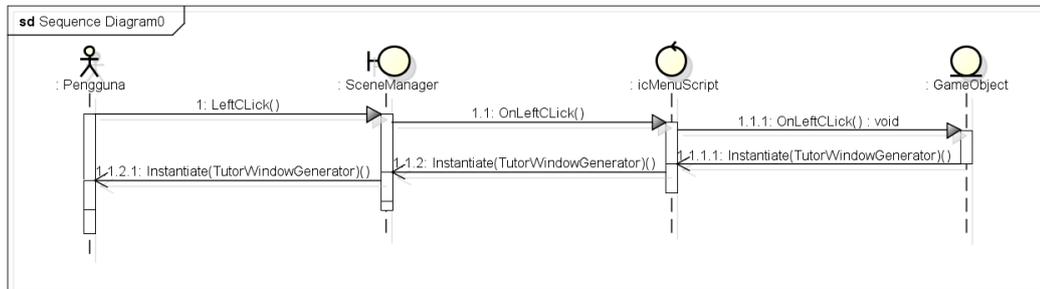
Diagram *sequence* lihat halaman uji merupakan diagram *sequence* ketika pengguna membuka halaman uji pada halaman percobaan. Perancangan diagram *sequence* lihat halaman uji dapat dilihat pada Gambar 4.14



Gambar 4.14 Diagram Sequence Lihat Halaman Uji

#### 4.2.3.5 Diagram Sequence Lihat Halaman Tutorial

Diagram *sequence* lihat halaman *tutorial* merupakan diagram *sequence* ketika pengguna membuka halaman *tutorial* pada halaman percobaan. Perancangan diagram *sequence* lihat halaman *tutorial* dapat dilihat pada Gambar 4.15

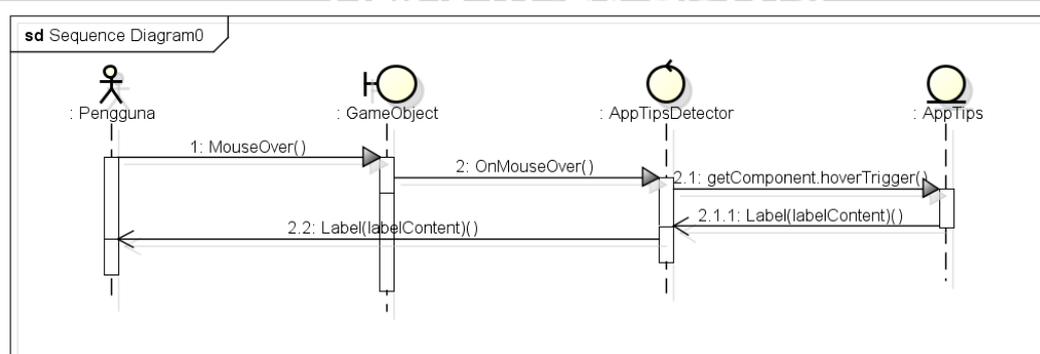


powered by astah

Gambar 4.15 Diagram Sequence Lihat Halaman Tutorial

#### 4.2.3.6 Diagram Sequence Lihat Informasi Benda

Diagram *sequence* lihat informasi benda merupakan diagram *sequence* ketika pengguna menampilkan informasi benda pada halaman percobaan. Perancangan diagram *sequence* lihat informasi benda dapat dilihat pada Gambar 4.16

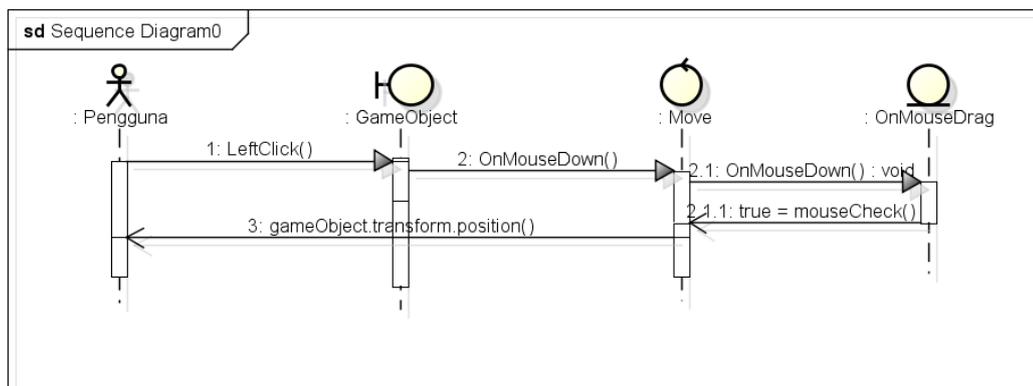


powered by astah

Gambar 4.16 Diagram Sequence Lihat Informasi Benda

#### 4.2.3.7 Diagram Sequence Menggerakkan Benda

Diagram *sequence* menggerakkan benda merupakan diagram *sequence* ketika pengguna menggerakkan benda pada halaman percobaan. Perancangan diagram *sequence* menggerakkan benda dapat dilihat pada Gambar 4.17

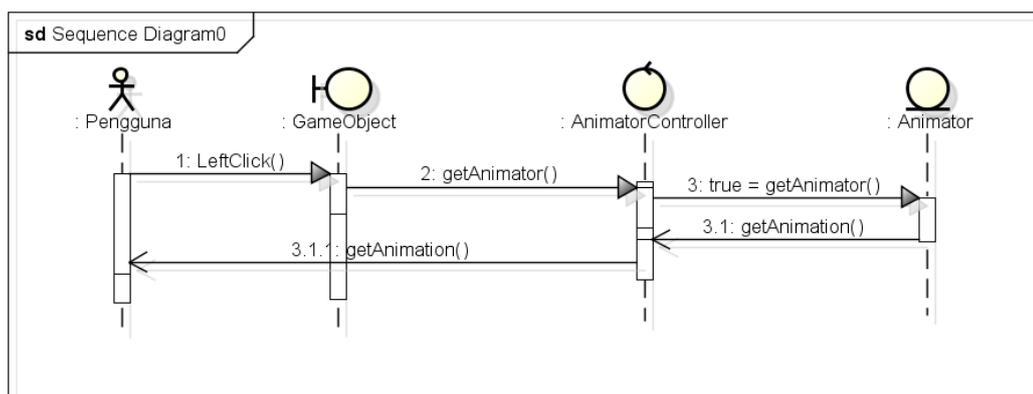


powered by astah

Gambar 4.17 Diagram Sequence Menggerakkan Benda

#### 4.2.3.8 Diagram Sequence Menjalankan Animasi

Diagram *sequence* menjalankan animasi merupakan diagram *sequence* ketika animasi benda terjadi pada halaman percobaan. Perancangan diagram *sequence* menjalankan animasi dapat dilihat pada Gambar 4.18



powered by astah

Gambar 4.18 Diagram Sequence Menjalankan Animasi

#### 4.2.4 Antarmuka

Desain dan perancangan tampilan antarmuka setiap *screen* yang ada dalam sistem, ditunjukkan pada Tabel 4.19

Tabel 4.19 Desain dan Rancangan Antarmuka

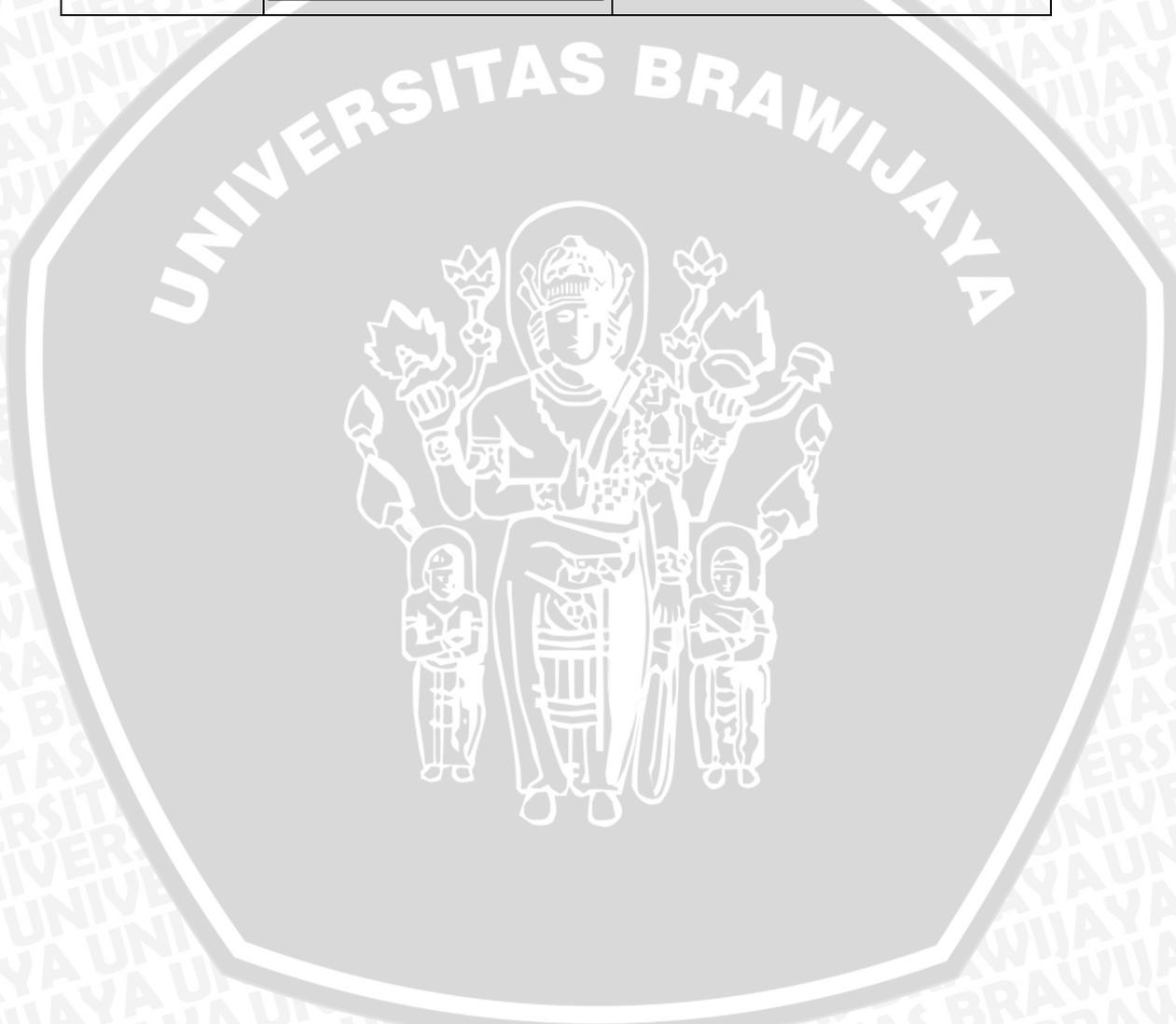
Nama Layer	Desain Antarmuka	Keterangan
Halaman Utama		<ol style="list-style-type: none"> <li>1. Tombol mulai, melihat daftar percobaan.</li> <li>2. Tombol <i>acout</i>, menampilkan halaman tentang perangkat.</li> <li>3. Tombol keluar dari aplikasi.</li> <li>4. Judul aplikasi.</li> <li>5. Gambar dan animasi bergerak</li> </ol>

Tabel 4.19 Desain dan Rancangan Antarmuka (Lanjutan)

Nama Layer	Desain Antarmuka	Keterangan
About Screen		<ol style="list-style-type: none"> <li>1. <i>About</i>, menampilkan tulisan mengenai deskripsi singkat permainan dan pembuat.</li> <li>2. Tombol <i>back</i>, tombol untuk kembali ke halaman utama.</li> </ol>
Halaman daftar percobaan		<ol style="list-style-type: none"> <li>1. Judul ruangan dari percobaan.</li> <li>2. Tombol masuk kedalam ruangan percobaan sesuai percobaan yang dipilih.</li> <li>3. Tombol melihat daftar percobaan sebelumnya</li> <li>4. Tombol melihat daftar percobaan sesudahnya.</li> <li>5. Deskripsi mengenai percobaan yang ditampilkan</li> </ol>
Halaman percobaan		<ol style="list-style-type: none"> <li>1. <i>Box</i> pemberitahuan</li> <li>2. Letak benda tiap-tiap percobaan.</li> <li>3. Tombol keluar, kembali ke halaman utama.</li> <li>4. Tombol <i>tutorial</i>, melihat cara kerja dan alur proses percobaan.</li> </ol>
Halaman Uji Percobaan 1		<ol style="list-style-type: none"> <li>1. Cara kerja atau melakukan percobaan 1</li> <li>2. <i>Water generator</i> tempat keluarnya air untuk percobaan</li> <li>3. Magnet, alat dan bahan untuk melakukan percobaan 1</li> <li>4. Tombol untuk menutup halaman uji kembali ke halaman percobaan 1</li> </ol>
Halaman Uji Percobaan 2		<ol style="list-style-type: none"> <li>1. Cara kerja atau melakukan percobaan 2</li> <li>2. Sel volta, alat dan bahan untuk melakukan percobaan 2</li> <li>3. Rangkaian lampu, alat dan bahan untuk melakukan percobaan 2</li> <li>4. Tombol untuk menutup halaman uji kembali ke halaman percobaan 2</li> </ol>

Tabel 4.19 Desain dan Rancangan Antarmuka (Lanjutan)

Nama Layer	Desain Antarmuka	Keterangan
Halaman <i>Tutorial</i> penggunaan lab percobaan		<ol style="list-style-type: none"> <li>1. Daftar list alur cara kerja percobaan</li> <li>2. Gambar cara kerja percobaan sesuai urutan yang dipilih</li> <li>3. Tombol untuk menutup halaman <i>tutorial</i> kembali ke halaman percobaan.</li> </ol>



## BAB 5 IMPLEMENTASI

Proses pengimplementasian sistem dilakukan berdasarkan perancangan sistem dan analisa kebutuhan yang telah dijelaskan pada bab sebelumnya. Pengimplementasian sistem akan menjelaskan tentang spesifikasi lingkungan pengembangan sistem, batasan implementasi, implementasi algoritma, dan implementasi antar muka pengguna.

### 5.1 Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem *virtual laboratory* ini antara lain sebagai berikut:

1. Pengimplementasian program menggunakan *Game Engine Unity* sebagai pengimplementasian keseluruhan sistem sedangkan untuk bahasa pemrograman menggunakan bahasa pemrograman C# sebagai pembuatan kode program yang juga telah disediakan pada *Unity*. Untuk pengimplementasian objek laboratorium dan antar muka pengguna dilakukan penggambaran 2D yang dibuat menggunakan perangkat lunak Adobe Photoshop CS5.
2. Data yang digunakan terkait percobaan dan cara menjalankan percobaan dalam sistem diambil dari cara kerja melakukan percobaan yang diambil dari referensi buku pelajaran kimia siswa/siswi sekolah menengah atas kelas X (lampiran A cara kerja percobaan).
3. Hasil percobaan yang ada pada program mengikuti teori yang diambil mengikuti dasar teori. Hasil percobaan pada program dan pembuktian sesuai dengan teori dijelaskan lebih detail pada lembar lampiran B evaluasi program.
4. Implementasi yang dilakukan tidak menjelaskan implementasi keseluruhan dari sistem melainkan implementasi sesuai dengan daftar kebutuhan dan perancangan yang telah dibuat sebelumnya.

### 5.2 Implementasi Algoritma

Implementasi algoritma dilakukan berdasarkan pada perancangan sistem yang telah dijelaskan pada bab sebelumnya.

#### 5.2.1 Algoritma Main Button

0.	Inisialisasi:
1.	Sprite[] <- index sprite
2.	
3.	Start:
4.	<b>If</b> nama <b>GameObject</b> = nama <b>GameObject</b> <b>Then</b>
5.	<b>GameObject</b> <b>sprite</b> <- <b>sprites</b> []
6.	<b>else</b>
7.	<b>GameObject</b> <b>Destroy</b>
8.	<b>End</b>

```

9.
10. function LeftClick():
11.     If nama GameObject = Start Then
12.         findGameObjectTag Start
13.         If Tag = 0 Then
14.             Load Window Start
15.         End
16.     Else If nama GameObject = About Then
17.         findGameObjectTag About
18.         If Tag = 0 Then
19.             Load Window About
20.         End
21.     Else If nama GameObject = Quit Then
22.         findGameObjectTag Quit
23.         If Tag = 0 Then
24.             Load Window Quit
25.         End
26.     End
27. End Function

```

Algoritma main button menjelaskan implementasi terkait daftar kebutuhan fungsional sistem yang antara lain kebutuhan fungsional lihat halaman *about* dan lihat halaman percobaan, dimana pada pseudocode diatas dijelaskan untuk menjalankan kebutuhan fungsional lihat halaman *about* dan lihat halaman percobaan dibutuhkan penekanan tombol kiri *mouse* pada GUI berupa tombol yang ada pada halama utama sistem.

### 5.2.2 Algoritma Menjalankan Halaman Percobaan

```

0. Inisialisasi:
1. Sprite[] <- index sprite
2.
3. Start:
4. If nama GameObject = NoteButton Then
5.     GameObject sprite <- sprites[]
6. End
7.
8. function LeftClick():
9.     Find GameObject WindowGenerator
10.    Int Index = WindowGenerator Stage Index
11.    If Index == 1 Then
12.        Load Window Percobaan 1
13.    Else If Index == 2 Then
14.        Load Window Percobaan 2
15.    Else If Index == 3 Then
16.        Load Window Percobaan 3
17.    End
18. End Function

```

Algoritma menjalankan halaman percobaan menjelaskan implementasi terkait daftar kebutuhan fungsional menjalankan halaman percobaan, pseudocode diatas menjelaskan algoritma sistem dalam membuka halaman percobaan pada sistem dengan penekanan GUI tombol pada sistem menggunakan tombol kiri *mouse*.

### 5.2.3 Algoritma Menampilkan Informasi Benda

```

0.  Inialisasi:
1.  InputHandler = false
2.  HoverTrigger = false
3.  Objek klik = null
4.  Objek klik return = false
5.
6.  Start:
7.  If cek Scene atau Halaman Percobaan != null Then
8.      Scene = index Scene
9.  End
10.
11. Function Update():
12.     If Klik kiri mouse = true Then
13.         InputHandler = true
14.     If Objek klik = null Then
15.         Objek klik = GetObjek()
16.         If Objek klik != null Then
17.             Objek klik return = true
18.         End
19.     End
20.     Else If Klik kiri mouse = false Then
21.         If Objek klik != null Then
22.             Objek klik = null
23.         If Objek klik = null Then
24.             Objek klik return = false
25.         End
26.     End
27. End
28. EndFunction
29.
30. Function GetObjek():
31.     Camera Ray = Input mouse ubah ScreenPoint ke Ray
32.     RaycastHit = Hit
33.     If cast ray get objek Then
34.         Return objek hit
35.     Else Return null
36.     End
37. EndFunction
38.
39. Function GUI():
40.     GUI.Skin = make Window
41.     If Objek klik return != null Then
42.         hoverTrigger = false
43.         Objek klik return get nama objek
44.     End
45.
46.     If HoverTrigger = true Then
47.         GameObjectNama = nama GameObject
48.         If GameObjectNama != null Then

```

```

49. Window posisi kanan atas layar
50.     End
51.     End
52. EndFunction
53.
54. Function WindowFunction():
55.     String LabelContent = null
56.     If HoverTrigger = true Then
57.         If GameObjectNama = nama GameObject Then
58.             Label = Isi informasi benda
59.         End
60.     End
61.     GUILabel (return LabelContent)
62. EndFunction

```

Algoritma menampilkan informasi benda menjelaskan implementasi terkait daftar kebutuhan fungsional lihat informasi benda, pseudocode diatas menjelaskan algoritma sistem dalam menampilkan informasi benda dalam sistem dengan metode *hover*.

#### 5.2.4 Algoritma Menggerakkan Benda

```

0. Inisialisasi:
1. OrderLayer
2. ScreenPoint <- Vector3
3. Offset <- Vector3
4. DefaultPos <- Vector3
5. Mousecheck bool
6.
7. Start:
8. Mousecheck <- false
9. OrderLayer = Objek order layer
10.
11. function MouseDown():
12.     screenPoint = camera posisi vector3 world to screenpoint
13.     offset = (posisi gameobject)-(posisi mouse pada
14. screenpoint)
15. End Function
16.
17. Function MouseUp():
18.     Mousechek <- false
19.     Objek order layer = OrderLayer
20. EndFunction
21.
22. Function MouseDrag():
23.     mouseCheck <- true
24.     Objek order layer = 5
25.     curScreenPoint = posisi vector3 mouse
26.     currentPosition = curScreenPoint Vector3 world to
27. screenpoint
28.
29.     If GameObject.y >= screen atas &&
30.     currentPosition.y >= screen atas Then
31.         GameObject posisi newVector3 batas screen atas
32.     Else If GameObject.y <= sceen bawah &&
33.     currentPosition.y <= screen bawah Then
34.         GameObject posisi newVector3 batas screen bawah
35.     Else If GameObject.x <= screen kiri &&

```

```

36.     currentPosition.x <= screen kiri Then
37.     GameObject posisi newVector3 batas screen kiri
38.     Else If GameObject.x >= screen kanan &&
39.     currentPosition.x >= screen kanan Then
40.     GameObject posisi newVector3 batas screen kanan
41.     Else GameObject posisi = currentPosition
42.     End
43. EndFunction

```

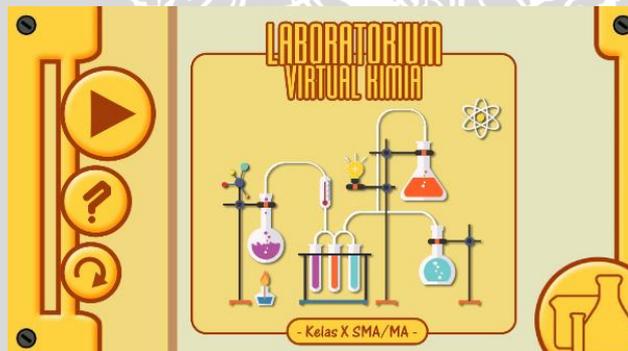
Algoritma menggerakkan benda menjelaskan implementasi terkait daftar kebutuhan fungsional menggerakkan benda, pseudocode diatas menjelaskan algoritma sistem dalam menggerakkan benda membutuhkan penarikan benda dan memperlihatkan wilayah pergerakan benda pada sistem.

### 5.3 Implementasi Antar Muka Pengguna

Implementasi antarmuka dilakukan dengan mengimplementasikan rancangan antarmuka sistem yang telah dibuat pada bab sebelumnya.

#### 5.3.1 Halaman Utama

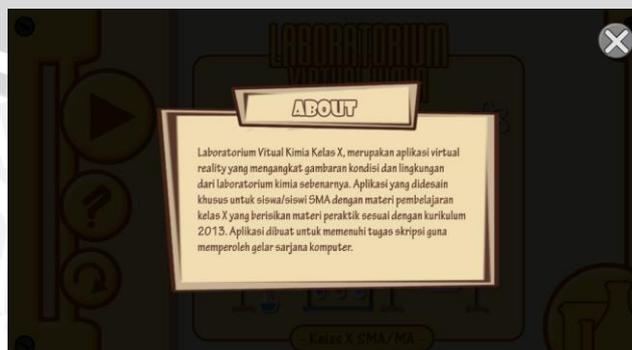
Berikut ini merupakan implementasi antarmuka halaman utama yang ditunjukkan pada Gambar 5.1.



Gambar 5.1. Halaman Utama

#### 5.3.2 About Screen

Berikut ini merupakan implementasi antarmuka halaman *about* yang ditunjukkan pada Gambar 5.2.



Gambar 5.2. About Screen

### 5.3.3 Halaman daftar percobaan

Berikut ini merupakan implementasi antarmuka halaman daftar percobaan yang ditunjukkan pada Gambar 5.3.



Gambar 5.3. Halaman Daftar Percobaan

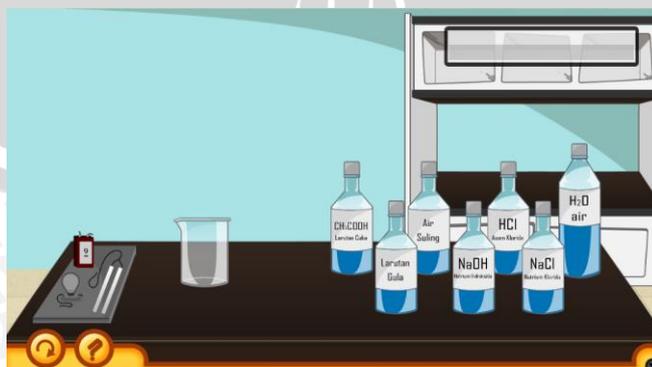
### 5.3.4 Halaman percobaan

Berikut ini merupakan implementasi antarmuka halaman percobaan 1 yang ditunjukkan pada Gambar 5.4.



Gambar 5.4. Halaman Percobaan 1

Berikut ini merupakan implementasi antarmuka halaman percobaan 2 yang ditunjukkan pada Gambar 5.5.



Gambar 5.5. Halaman Percobaan 2

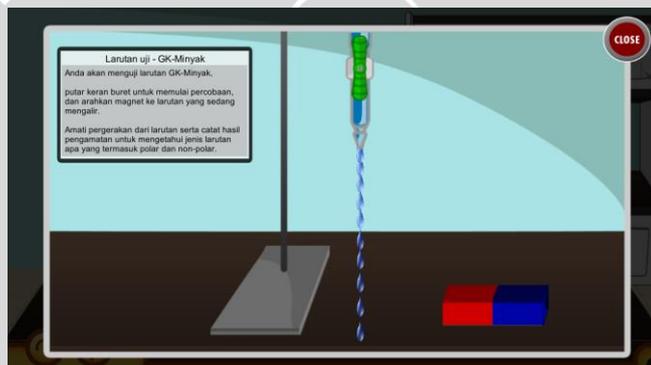
Berikut ini merupakan implementasi antarmuka halaman percobaan 3 yang ditunjukkan pada Gambar 5.6.



Gambar 5.6. Halaman Percobaan 3

### 5.3.5 Halaman Uji Percobaan 1

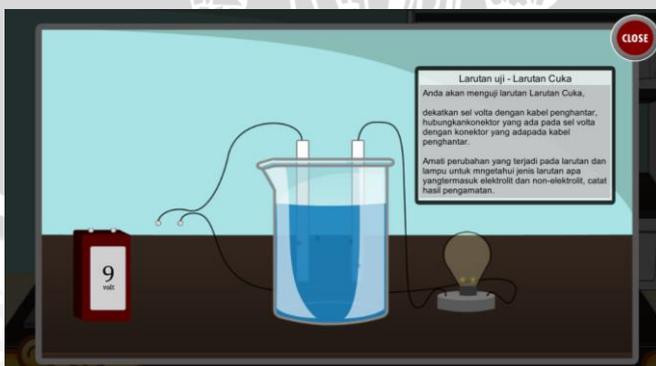
Berikut ini merupakan implementasi antarmuka halaman uji percobaan 1 yang ditunjukkan pada Gambar 5.7.



Gambar 5.7. Uji Percobaan 1

### 5.3.6 Halaman Uji Percobaan 2

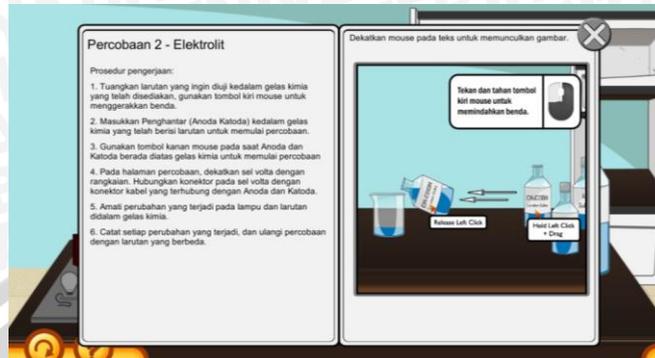
Berikut ini merupakan implementasi antarmuka halaman uji percobaan 2 yang ditunjukkan pada Gambar 5.8.



Gambar 5.8. Uji Percobaan 2

### 5.3.7 Halaman *Tutorial* penggunaan lab percobaan

Berikut ini merupakan implementasi antarmuka halaman *tutorial* penggunaan lab percobaan yang ditunjukkan pada Gambar 5.9.



Gambar 5.9. Halaman *Tutorial* Penggunaan Lab Percobaan



## BAB 6 PENGUJIAN DAN ANALISIS

Pada tahap ini akan dibahas mengenai pengujian sistem (*testing*), pengujian sistem dilakukan dengan melakukan dua metode yaitu metode *white-box* dan *black-box testing*. Untuk pengujian *white-box* akan digunakan metode pengujian unit, sedangkan untuk pengujian *black-box* akan digunakan metode pengujian validasi dimana pengujian dilakukan untuk mengetahui apakah sistem yang sudah dibangun telah benar sesuai dengan kebutuhan atau tidak. Pengujian dilakukan berdasarkan kebutuhan fungsional dan implementasi sistem yang dibuat pada bab sebelumnya.

### 6.1 Pengujian Unit

Pengujian unit dilakukan dengan metode *basis path testing*, dimana proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah *cyclomatic complexity*, menentukan jalur independen, dan menentukan kasus uji berdasarkan jalur independen yang telah ditentukan. Pengujian unit dilakukan terhadap algoritma sistem yang telah di implementasikan pada bab implementasi sebelumnya, adapun algoritma sistem yang diuji antara lain algoritma *main button*, algoritma menjalankan halaman percobaan, algoritma menggerakkan benda, dan algoritma menampilkan informasi benda.

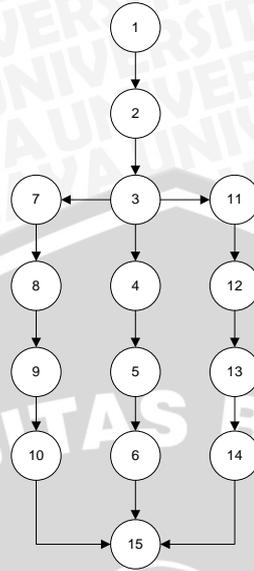
#### 6.1.1 Pengujian Unit *Main Button*

Berikut ini merupakan algoritma untuk melakukan proses *main button* yang akan dijelaskan pada Tabel 6.1.

**Tabel 6.1 Pengujian Unit *Main Button***

Nama Algoritma: <i>Main Button</i>	
1.	Jika nama GameObject sama dengan nama salah satu tombol ( <i>menu</i> )
2.	Maka sprite GameObject diisikan sprite dari tombol ( <i>menu</i> )
3.	Endif, Jika nama GameObject sama dengan <i>start</i>
4.	Maka mengecek GameObject Tag dengan nama Tag <i>Start</i>
5.	Jika Tag tidak ditemukan
6.	Maka membuka halaman mulai daftar percobaan
7.	Endif, Namun jika nama GameObject sama dengan <i>about</i>
8.	Maka mengecek GameObject Tag dengan nama Tag <i>About</i>
9.	Jika Tag tidak ditemukan
10.	Maka membuka halaman <i>about</i>
11.	Endif, Namun jika nama GameObject sama dengan <i>quit</i>
12.	Maka mengecek GameObject Tag dengan nama Tag <i>Quit</i>
13.	Jika Tag tidak ditemukan
14.	Maka menutup aplikasi
15.	Endif
Keluaran: <i>Button</i> berfungsi sesuai kebutuhan	

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 6.1



**Gambar 6.1 Flow Graph Main Button**

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  adalah jumlah kompleksitas siklomatis,  $E$  merupakan sisi atau *edge* (garis penghubung antar *node*) dan  $N$  merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis *main button*:

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 16 - 15 + 2 \\
 &= 3
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan maka akan didapat 3 jalur independen yang ditentukan sebagai berikut:

- Jalur 1 : 1-2-3-4-5-6-15
- Jalur 2 : 1-2-3-7-8-9-10-15
- Jalur 3 : 1-2-3-11-12-13-14-15

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada tabel 6.2

**Tabel 6.2 Kasus Uji Jalur Independen Main Button**

Jalur	Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan
1	Menekan tombol 1 untuk membuka halaman mulai atau daftar percobaan pada sistem	Sistem akan menampilkan halaman daftar percobaan	Sistem menampilkan halaman daftar percobaan
2	Menekan tombol 2 untuk membuka halaman <i>about</i> pada sistem	Sistem akan menampilkan halaman <i>about</i>	Sistem menampilkan halaman <i>about</i>
3	Menekan tombol 3 untuk menutup dan keluar dari sistem	Sistem akan menutup aplikasi dan keluar dari sistem	Sistem menutup aplikasi dan keluar dari sistem

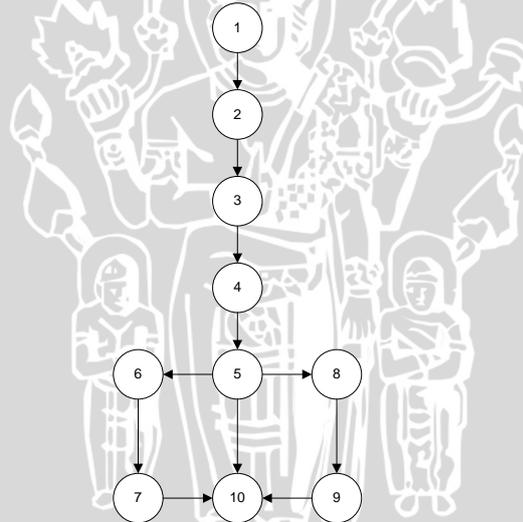
### 6.1.2 Pengujian Unit Menjalankan Halaman Percobaan

Berikut ini merupakan algoritma untuk melakukan proses menjalankan halaman percobaan yang akan dijelaskan pada Tabel 6.3.

**Tabel 6.3 Pengujian Unit Menjalankan Halaman Percobaan**

Nama Algoritma: Menjalankan Halaman Percobaan	
1.	Jika nama GameObject sama dengan salah satu tombol (menu)
2.	Maka sprite GameObject diisikan sprite dari tombol (menu)
3.	Endif, temukan GameObject WindGenerator dan masukkan index dari WindGenerator Stage ke Index Integer
4.	Jika Index sama dengan 1
5.	Maka membuka halaman percobaan 1
6.	Namun Jika Index sama dengan 2
7.	Maka membuka halaman percobaan 2
8.	Namun Jika Index sama dengan 3
9.	Maka membuka halaman percobaan 3
10.	Endif
Keluaran : halaman percobaan	

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 6.2



**Gambar 6.2 Flow Graph Menjalankan Halaman Percobaan**

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  adalah jumlah kompleksitas siklomatis,  $E$  merupakan sisi atau *edge* (garis penghubung antar *node*) dan  $N$  merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis menjalankan halaman percobaan:

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 11 - 10 + 2 \\ &= 3 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan maka akan didapat 3 jalur independen yang ditentukan sebagai berikut:

- Jalur 1 : 1-2-3-4-5-10
- Jalur 2 : 1-2-3-4-5-6-7-10
- Jalur 3 : 1-2-3-4-5-8-9-10

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada tabel 6.4

**Tabel 6.4. Kasus Uji Jalur Independen Menjalankan Halaman Percobaan**

Jalur	Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan
1	Sistem membuka halaman percobaan 1 pada halaman daftar percobaan	Sistem akan menampilkan halaman percobaan 1	Sistem menampilkan halaman percobaan 1
2	Sistem membuka halaman percobaan 2 pada halaman daftar percobaan	Sistem akan menampilkan halaman percobaan 2	Sistem menampilkan halaman percobaan 2
3	Sistem membuka halaman percobaan 3 pada halaman daftar percobaan	Sistem akan menampilkan halaman percobaan 3	Sistem menampilkan halaman percobaan 3

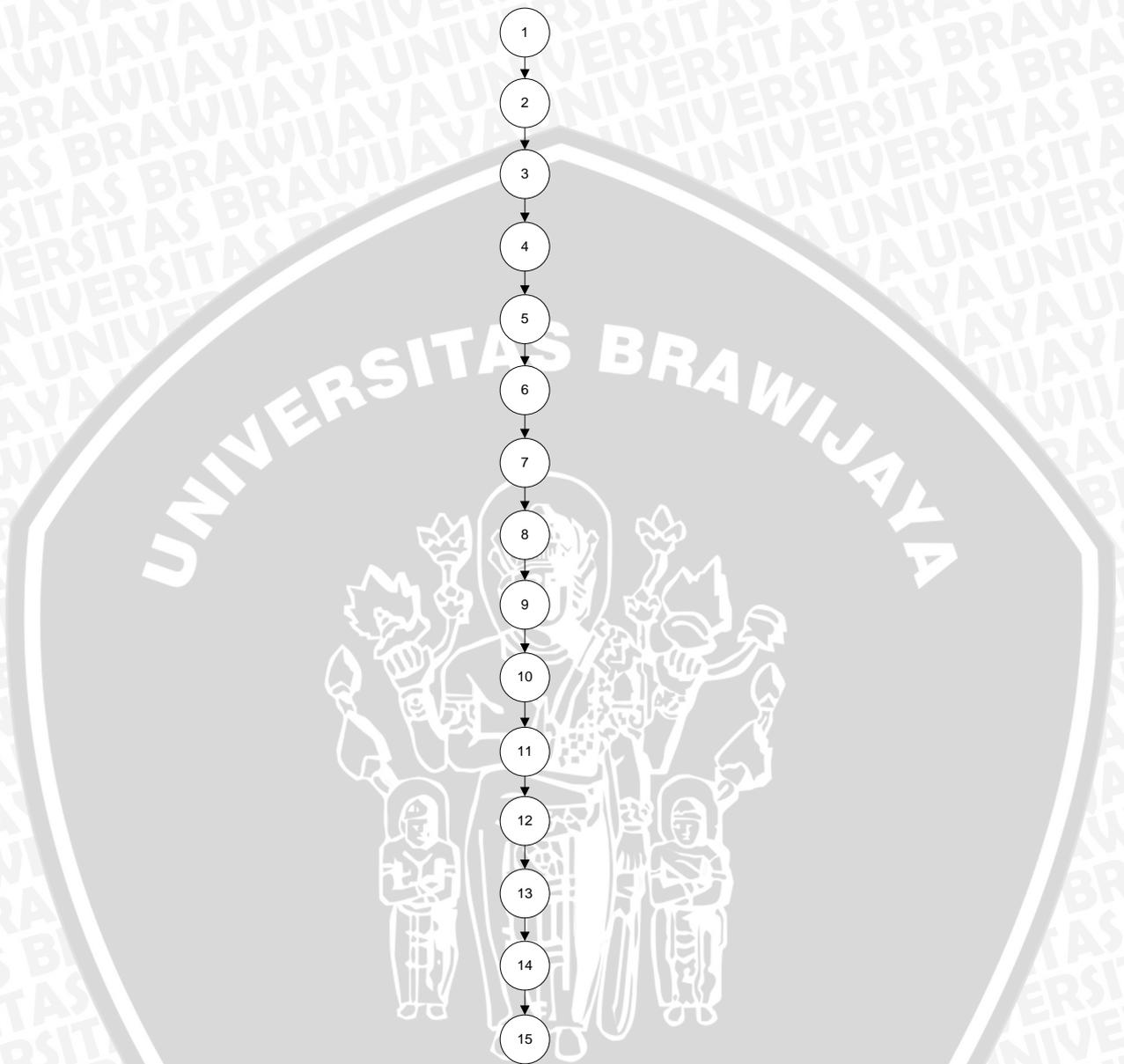
### 6.1.3 Pengujian Unit Menampilkan Informasi Benda

Berikut ini merupakan algoritma untuk melakukan proses menggerakkan benda yang akan dijelaskan pada tabel 6.5

**Tabel 6.5 Pengujian Unit Menampilkan Informasi Benda**

Nama Algoritma: Menampilkan informasi benda Benda	
1.	Variabel camera ray diisi lokasi input mouse yang diconvert dari lokasi screenpoint ke ray. Variabel raycasthit diisi hit
2.	Jika cast ray mengenai objek
3.	Maka mengembalikan objek yang dikenai
4.	Kalau tidak mengembalikan null
5.	Endif, gui skin diisi variabel window
6.	Jika objek yang disentuh tidak null
7.	Maka trigger hover bernilai salah serta objek klik diisi nama objek yang didapat
8.	Endif, jika trigger hover bernilai benar
9.	Maka variabel nama gameobject diisi dengan nama gameobject
10.	Jika variabel nama gameobject tidak bernilai null
11.	Maka box informasi benda diletakkan pada posisi kanan atas
12.	Endif, Isi Label diisi null
13.	Jika variabel trigger hover bernilai benar
14.	Maka, jika variabel nama gameobject sama dengan nama gameobject
15.	Maka label diisi dengan informasi dari benda
16.	Endif, guiLabel mengembalikan isi label
Keluaran : Menampilkan informasi benda	

Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 6.3



**Gambar 6.3 Flow Graph Menampilkan Informasi Benda**

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  adalah jumlah kompleksitas siklomatis,  $E$  merupakan sisi atau *edge* (garis penghubung antar *node*) dan  $N$  merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis menampilkan informasi benda:

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 14 - 15 + 2 \\ &= 1 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan maka akan didapat 1 jalur independen yang ditentukan sebagai berikut:

Jalur 1 : 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada tabel 6.6.

**Tabel 6.6. Kasus Uji Jalur Independen Menampilkan Informasi Benda**

Jalur	Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan
1	Sistem menampilkan informasi benda pada saat pengguna mendekati mouse ke benda	Sistem akan menampilkan informasi benda yang ditunjuk	Sistem menampilkan informasi benda yang ditunjuk

### 6.1.4 Pengujian Unit Menggerakkan Benda

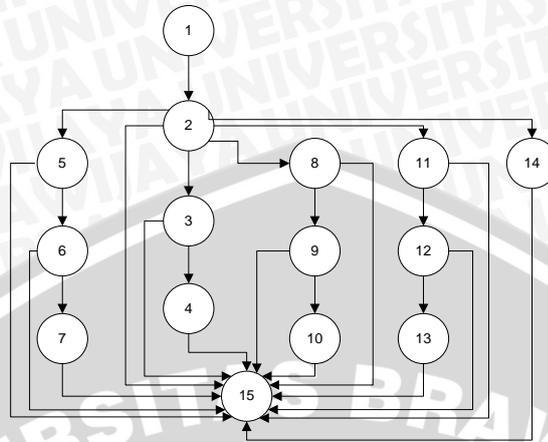
Berikut ini merupakan algoritma untuk melakukan proses menggerakkan benda yang akan dijelaskan pada Tabel 6.7.

**Tabel 6.7 Pengujian Unit Menggerakkan Benda**

Nama Algoritma: Menggerakkan Benda	
1.	Isi variabel mousecheck dan Orderlayer dengan true dan batas atas layer, Isi variabel screenpoint dengan vector 3 camera, dan nilai offset dengan nilai dari pengurangan posisi objek dan posisi mouse. Isi variabel posisi screenpoint sekarang dengan posisi <i>mouse</i> , dan posisi benda sekarang dengan posisi screenpoint saat ini
2.	Jika posisi y benda lebih besar sama dengan batas layar atas
3.	Dan posisi saat ini lebih besar sama dengan batas layar atas
4.	Maka GameObject diposisikan sebatas batas layar atas
5.	Namun jika posisi y lebih kecil sama dengan batas layar bawah
6.	Dan posisi saat ini lebih kecil sama dengan batas layar bawah
7.	Maka GameObject diposisikan sebatas batas layar bawah
8.	Namun jika posisi x lebih kecil sama dengan batas layar kiri
9.	Dan posisi saat ini lebih kecil sama dengan batas layar kiri
10.	Maka GameObject diposisikan sebatas batas layar kiri
11.	Namun jika posisi x lebih besar sama dengan batas layar kanan
12.	Dan posisi saat ini lebih besar sama dengan batas layar kanan
13.	Maka GameObject diposisikan sebatas batas layar kanan
14.	Selain itu GameObject diposisikan sama dengan posisi saat ini
15.	Endif
Keluaran : Benda berhasil bergerak	



Berikut ini merupakan *flow graph* dari algoritma diatas yang akan dijelaskan pada Gambar 6.4



**Gambar 6.4 Flow Graph Menggerakkan Benda**

Berdasarkan *flow graph* diatas akan diperoleh jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  adalah jumlah kompleksitas siklomatis,  $E$  merupakan sisi atau *edge* (garis penghubung antar *node*) dan  $N$  merupakan jumlah simpul (*node*). Berikut ini merupakan perhitungan dari jumlah kompleksitas siklomatis menjalankan halaman percobaan:

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 26 - 15 + 2 \\
 &= 13
 \end{aligned}$$

Berdasarkan jumlah kompleksitas siklomatis yang telah didapatkan maka akan didapat 13 jalur independen yang ditentukan sebagai berikut:

- Jalur 1 : 1-2-3-4-15
- Jalur 2 : 1-2-15
- Jalur 3 : 1-2-3-15
- Jalur 4 : 1-2-5-6-7-15
- Jalur 5 : 1-2-5-15
- Jalur 6 : 1-2-5-6-15
- Jalur 7 : 1-2-8-9-10-15
- Jalur 8 : 1-2-8-15
- Jalur 9 : 1-2-8-9-15
- Jalur 10 : 1-2-11-12-13-15
- Jalur 11 : 1-2-11-15
- Jalur 12 : 1-2-11-12-15
- Jalur 13 : 1-2-14-15

Berdasarkan jalur independen yang telah ditentukan, maka dapat diperoleh kasus uji yang akan dijelaskan pada tabel 6.8

**Tabel 6.8 Kasus Uji Jalur Independen Menggerakkan Benda**

Jalur	Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan
1	Sistem menggerakkan benda hingga posisi benda dan posisi mouse lebih dari sama dengan batas atas layar	Sistem menggerakkan benda maksimal hingga batas atas layar	Benda dapat digerakkan tidak lebih dari batas atas layar
2	Sistem menggerakkan benda hingga posisi benda kurang dari batas atas layar	Sistem menggerakkan benda dalam batas atas layar	Benda dapat digerakkan dalam batas atas layar
3	Sistem menggerakkan benda hingga posisi mouse kurang dari batas atas layar	Sistem menggerakkan benda dalam batas atas layar	Benda dapat digerakkan dalam batas atas layar
4	Sistem menggerakkan benda hingga posisi benda dan posisi mouse kurang dari sama dengan batas bawah layar	Sistem menggerakkan benda maksimal hingga batas bawah layar	Benda dapat digerakkan tidak lebih dari batas bawah layar
5	Sistem menggerakkan benda hingga posisi benda lebih dari batas bawah layar	Sistem menggerakkan benda dalam batas bawah layar	Benda dapat digerakkan dalam batas bawah layar
6	Sistem menggerakkan benda hingga posisi mouse lebih dari batas bawah layar	Sistem menggerakkan benda dalam batas bawah layar	Benda dapat digerakkan dalam batas bawah layar
7	Sistem menggerakkan benda hingga posisi benda dan posisi mouse kurang dari sama dengan batas kiri layar	Sistem menggerakkan benda maksimal hingga batas kiri layar	Benda dapat digerakkan tidak lebih dari batas kiri layar
8	Sistem menggerakkan benda hingga posisi benda lebih dari batas kiri layar	Sistem menggerakkan benda dalam batas kiri layar	Benda dapat digerakkan dalam batas kiri layar
9	Sistem menggerakkan benda hingga posisi mouse lebih dari batas kiri layar	Sistem menggerakkan benda dalam batas kiri layar	Benda dapat digerakkan dalam batas kiri layar
10	Sistem menggerakkan benda hingga posisi benda dan posisi mouse kurang dari sama dengan batas kanan layar	Sistem menggerakkan benda maksimal hingga batas kanan layar	Benda dapat digerakkan tidak lebih dari batas kanan layar
11	Sistem menggerakkan benda hingga posisi benda lebih dari batas kanan layar	Sistem menggerakkan benda dalam batas kanan layar	Benda dapat digerakkan dalam batas kanan layar
12	Sistem menggerakkan benda hingga posisi mouse lebih dari batas kanan layar	Sistem menggerakkan benda dalam batas kanan layar	Benda dapat digerakkan dalam batas kanan layar
13	Sistem menggerakkan benda dalam wilayah layar	Sistem menggerakkan benda dalam batas layar	Benda dapat digerakkan dalam wilayah batas layar

## 6.2 Pengujian Validasi

Pengujian validasi dilakukan untuk menguji seluruh skenario *use case* yang telah dibuat berdasar dari analisis kebutuhan fungsional sistem yang telah dibuat sebelumnya. Pengujian ini dilakukan untuk mengetahui fungsi-fungsi khusus yang dirancang untuk mengetahui seberapa jauh sistem berjalan dan seberapa banyak kesalahan yang ada pada sistem.

### 6.2.1 Kasus Uji Pengujian Validasi

Berikut ini merupakan kasus uji untuk melakukan pengujian validasi dari setiap kebutuhan fungsional yang telah dibuat.

**Tabel 6.9. Kasus Uji Lihat Halaman *About***

<b>Nama Kasus Uji</b>	Kasus Uji Lihat Halaman <i>About</i>
<b>Objek Uji</b>	SRS_01
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menjalankan dan menampilkan halaman <i>about</i>
<b>Prosedur Uji</b>	1. Membuka aplikasi 2. memilih <i>menu about</i> pada halaman utama 3. memilih <i>menu</i> kembali ke halaman utama
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan halaman <i>about</i> ketika pengguna selesai menekan <i>menu about</i> dan menutup jendela <i>about</i> dan membuka halaman utama ketika pengguna menekan <i>menu</i> kembali

**Tabel 6.10. Kasus Uji Lihat Daftar Percobaan**

<b>Nama Kasus Uji</b>	Kasus Uji Lihat Daftar Percobaan
<b>Objek Uji</b>	SRS_02
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menjalankan dan menampilkan halaman daftar dari setiap percobaan
<b>Prosedur Uji</b>	4. Membuka aplikasi 5. memilih <i>menu</i> mulai pada halaman utama 6. memilih <i>menu</i> selanjutnya dan sebelumnya untuk melihat percobaan 7. memilih <i>menu</i> kembali ke halaman utama
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan jendela daftar percobaan ketika pengguna selesai menekan <i>menu</i> mulai dan menutup jendela daftar percobaan dan membuka halaman utama ketika pengguna menekan <i>menu</i> kembali

**Tabel 6.11. Kasus Uji Menjalankan Halaman Percobaan**

<b>Nama Kasus Uji</b>	Kasus Uji Menjalankan Halaman Percobaan
<b>Objek Uji</b>	SRS_03
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menjalankan dan memulai percobaan yang dipilih pengguna
<b>Prosedur Uji</b>	8. Membuka aplikasi 9. memilih <i>menu</i> mulai pada halaman utama 10. memilih percobaan 11. memilih <i>menu</i> kembali ke halaman utama
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan halaman percobaan sesuai dengan percobaan yang dipilih ketika pengguna selesai menekan salah satu percobaan dan menutup halaman percobaan dan membuka halaman utama ketika pengguna menekan <i>menu</i> kembali

**Tabel 6.12. Kasus Uji Lihat Halaman Uji**

<b>Nama Kasus Uji</b>	Kasus Uji Lihat Halaman Uji
<b>Objek Uji</b>	SRS_04
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menjalankan dan menampilkan halaman uji
<b>Prosedur Uji</b>	12. Membuka aplikasi dan menjalankan halaman percobaan 13. Memenuhi persyaratan uji pada percobaan yang dipilih 14. Membuka halaman uji 15. Memilih <i>menu</i> kembali ke halaman percobaan
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan jendela uji ketika pengguna membuka jendela uji setelah persyaratan terpenuhi dan menutup jendela uji ketika pengguna menekan <i>menu</i> kembali

**Tabel 6.13. Kasus Uji Lihat Halaman Tutorial**

<b>Nama Kasus Uji</b>	Kasus Uji Lihat Halaman <i>Tutorial</i>
<b>Objek Uji</b>	SRS_05
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menjalankan dan menampilkan halaman <i>tutorial</i>
<b>Prosedur Uji</b>	16. Membuka aplikasi dan menjalankan halaman percobaan 17. memilih <i>menu tutorial</i> pada halaman percobaan 18. memilih <i>menu</i> kembali ke halaman percobaan
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan jendela <i>tutorial</i> ketika pengguna selesai menekan <i>menu tutorial</i> dan akan menutup jendela <i>tutorial</i> ketika pengguna menekan <i>menu</i> kembali

**Tabel 6.14. Kasus Uji Lihat Informasi Benda**

<b>Nama Kasus Uji</b>	Kasus Uji Lihat Informasi Benda
<b>Objek Uji</b>	SRS_06
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menjalankan dan menampilkan informasi dari setiap benda yang dipilih pengguna
<b>Prosedur Uji</b>	19. Membuka aplikasi dan menjalankan halaman percobaan 20. Mendekatkan <i>cursor mouse</i> ke objek pada halaman percobaan 21. Menjauhkan <i>cursor mouse</i> dari objek pada halaman percobaan
<b>Hasil Yang Diharapkan</b>	Sistem akan menampilkan informasi benda ketika pengguna mendekatkan <i>cursor mouse</i> ke benda dan menutup kotak informasi ketika pengguna menjauhkan <i>cursor mouse</i> dari benda

**Tabel 6.15. Kasus Uji Menggerakkan Benda**

<b>Nama Kasus Uji</b>	Kasus Uji Menggerakkan Benda
<b>Objek Uji</b>	SRS_07
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menjalankan fungsi menggerakkan benda yang dipilih pengguna
<b>Prosedur Uji</b>	22. Membuka aplikasi dan menjalankan halaman percobaan 23. Menekan tombol kiri <i>mouse</i> pada benda yang dapat digerakkan 24. Menggerakkan <i>mouse</i> dengan tetap menekan tombol kiri <i>mouse</i> pada benda 25. Melepas penekanan tombol kiri <i>mouse</i> pada benda
<b>Hasil Yang Diharapkan</b>	Sistem menggerakkan benda yang dipilih pengguna sesuai dengan arah gerak <i>mouse</i> dan mengembalikan benda ke posisi awal ketika pengguna melepas penekanan tombol <i>mouse</i>

**Tabel 6.16. Kasus Uji Menjalankan Animasi**

<b>Nama Kasus Uji</b>	Kasus Uji Menjalankan Animasi
<b>Objek Uji</b>	SRS_08
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menjalankan animasi dari benda
<b>Prosedur Uji</b>	26. Membuka aplikasi dan menjalankan halaman percobaan 27. Menggerakkan benda 28. Memenuhi persyaratan animasi benda
<b>Hasil Yang Diharapkan</b>	Sistem menampilkan dan menjalankan animasi pada benda ketika persyaratan animasi terpenuhi

### 6.2.2 Hasil Pengujian Validasi

Berikut ini merupakan hasil pengujian validasi yang dilakukan sesuai dengan prosedur uji yang ada pada kasus uji validasi.

**Tabel 6.17. Hasil Pengujian Validasi**

No	Nama Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Validitas
1	Kasus Uji Lihat Halaman <i>About</i>	Sistem dapat menampilkan halaman <i>about</i> dan menutup jendela <i>about</i> kembali ke halaman utama	Sistem menampilkan halaman <i>about</i> dan dapat menutup jendela <i>about</i> kembali ke halaman utama	Valid
2	Kasus Uji Lihat Daftar Percobaan	Sistem dapat menampilkan jendela daftar percobaan dan menutup jendela daftar percobaan	Sistem menampilkan jendela daftar percobaan dan dapat menutup jendela daftar percobaan	Valid
3	Kasus Uji Menjalankan Halaman Percobaan	Sistem dapat menampilkan halaman percobaan sesuai dengan percobaan yang dipilih dan menutup halaman percobaan	Sistem menampilkan halaman percobaan sesuai dengan pilihan dan dapat menutup halaman percobaan untuk kembali ke halaman utama	Valid
4	Kasus Uji Lihat Halaman Uji	Sistem dapat menampilkan jendela uji dan menutup jendela uji	Sistem menampilkan dan dapat menutup jendela uji	Valid
5	Kasus Uji Lihat Halaman <i>Tutorial</i>	Sistem dapat menampilkan jendela <i>tutorial</i> dan dapat menutup jendela <i>tutorial</i>	Sistem menampilkan jendela <i>tutorial</i> dan menutup jendela <i>tutorial</i>	Valid
6	Kasus Uji Lihat Informasi Benda	Sistem dapat menampilkan informasi benda dan menutup kotak informasi	Sistem menampilkan informasi benda dan dapat menutup kotak indormasi	Valid

Tabel 6.17. Hasil Pengujian Validasi (Lanjutan)

No	Nama Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Validitas
7	Kasus Uji Menggerakkan Benda	Sistem dapat menggerakkan benda dan mengembalikan posisi awal benda ketika penekanan pada benda dilepas	Sistem menggerakkan benda dan dapat mengembalikan posisi benda ke posisi awal ketika penekanan pada benda dilepas	Valid
8	Kasus Uji Menjalankan Animasi	Sistem menampilkan animasi pada benda tertentu	Sistem menampilkan animasi pada benda	Valid

### 6.3 Analisis Hasil Pengujian

Analisis hasil pengujian dilakukan untuk mendapatkan kesimpulan dari hasil pengujian sistem yang telah dilakukan. Analisis dilakukan kepada setiap pengujian yang telah dilakukan, yaitu analisis hasil pengujian unit dan validasi.

#### 6.3.1 Analisis Hasil Pengujian Unit

Proses analisis hasil pengujian unit dilakukan dengan melihat kesesuaian fungsi dari implementasi sistem dengan hasil pengujian unit yang telah dilakukan. Berdasarkan hasil pengujian unit, implementasi sistem sudah sesuai dengan hasil yang diharapkan. Sehingga, dapat diambil kesimpulan bahwa laboratorium virtual yang dibangun telah memenuhi setiap kebutuhan atau fungsionalitas yang telah dirancang pada bab sebelumnya.

#### 6.3.2 Analisis Hasil Pengujian Validasi

Proses analisis hasil pengujian validasi dilakukan dengan melihat kesesuaian kinerja sistem dengan daftar kebutuhan fungsional sistem. Berdasarkan hasil pengujian validasi, kinerja sistem yang dibangun sudah sesuai dengan setiap kebutuhan fungsional sistem. Sehingga, dapat diambil kesimpulan, bahwa laboratorium virtual yang dibangun ini telah memenuhi setiap kebutuhan fungsional sistem yang telah dirancang pada bab sebelumnya.

## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, implementasi, dan pengujian sistem yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Proses analisis kebutuhan dalam membangun sistem dilakukan dengan menganalisa faktor dan hal-hal apa saja yang dibutuhkan pada laboratorium melalui studi literatur dan pencarian dasar teori melalui sumber buku, jurnal penelitian dan studi literatur lain.
2. Proses perancangan sistem dilakukan dengan membuat beberapa perancangan kebutuhan sistem antara lain perancangan *class diagram*, perancangan *activity diagram*, perancangan *sequence diagram*, dan perancangan antarmuka sistem.
3. Proses implementasi dilakukan berdasarkan hasil kebutuhan sistem yang telah dibuat dan kemudian di implementasikan kedalam *class-class* sebagai dasar pengembangan sistem.

Berdasarkan hasil pengujian unit dan validasi, sistem laboratorium virtual kimia kelas X dapat berjalan dengan baik sesuai dengan kebutuhan sistem dikarenakan hasil pengujian terhadap kebutuhan sistem menghasilkan pernyataan valid

### 7.2 Saran

Untuk melakukan pengembangan lebih lanjut terhadap sistem dapat dilakukan beberapa hal sebagai berikut:

1. Adanya penambahan fitur dalam laboratorium dan penambahan percobaan lain selain percobaan yang telah ada.
2. Penggabungan dengan modul lain seperti materi pelajaran kelas XI dan XII Sekolah Menengah Atas.
3. Pengembangan sistem terkait percobaan yang ada dalam laboratorium dilakukan pembaharuan mengikuti materi terkini menurut kurikulum yang berlaku setiap tahunnya.
4. Pengembangan sistem terkait kebutuhan sistem ditambahkan kebutuhan berdasarkan pakar yang ahli pada bidangnya.

## DAFTAR PUSTAKA

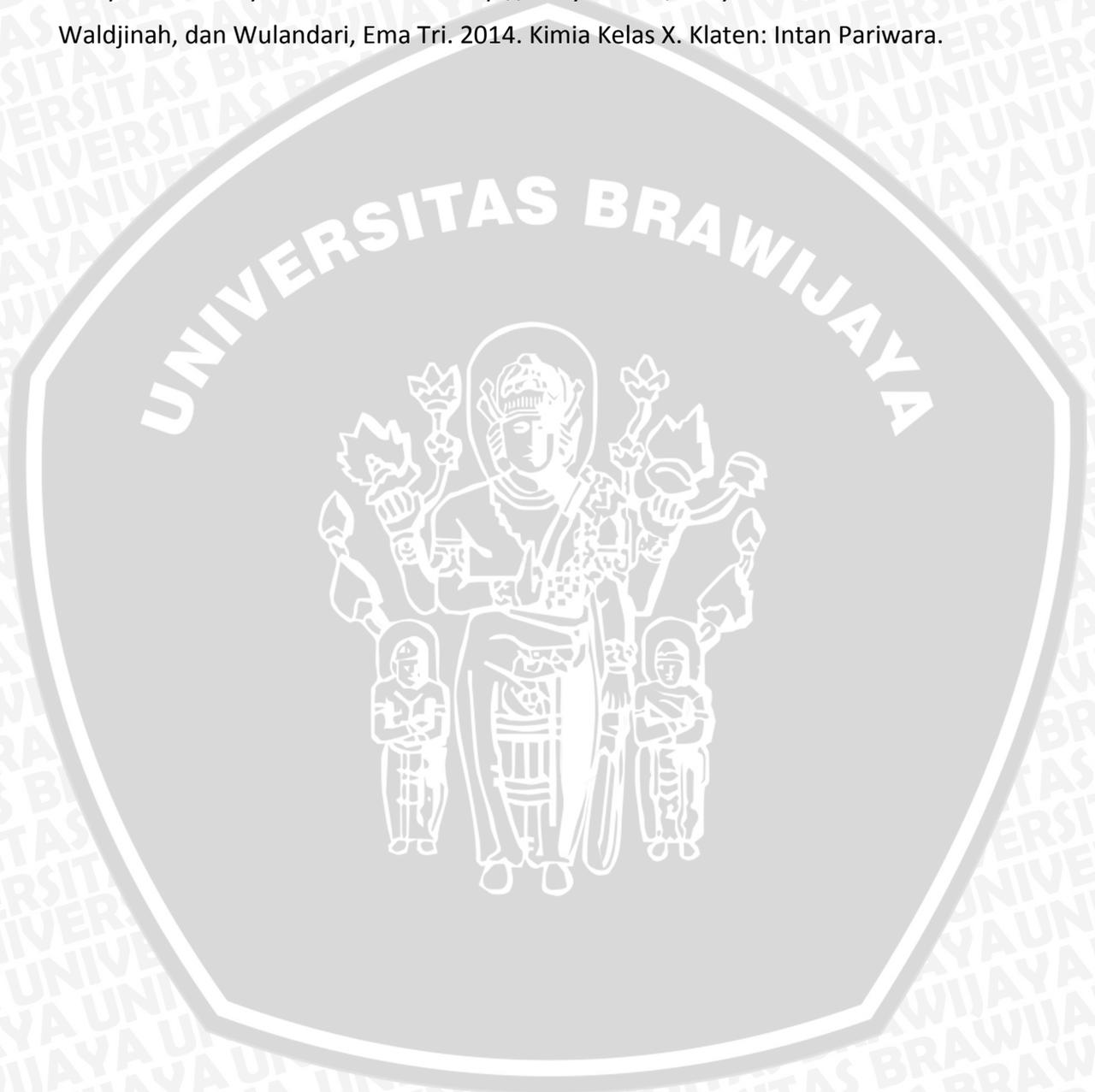
- Chang, Raymond. 2005. Kimia Dasar: Konsep-Konsep Inti Edisi Ketiga Jilid 1. Jakarta: Erlangga.
- Courtiat, J.P, Davarakis, Costas, dan Villemur, Thierry. 2005. *Technology Enhanced Learning*. New York: International Federation for Information Processing.
- Departemen Pendidikan dan Kebudayaan. (2013). *Silabus Mata Pelajaran Kimia (Permintaan Bidang MIPA)*. Jakarta: Depdikbud.
- Departemen Pendidikan dan Kebudayaan. (2013). *Kompetensi Dasar Sekolah Menengah Atas (SMA)/Madrasah Aliyah (MA)*. Jakarta: Depdikbud.
- Haris Watoni, A. 2013. *Kimia Untuk SMA/MA Kelas X*. Bandung : Yrama Widya.
- Jogiyanto. 2005. *Analisis dan Desain*. Yogyakarta : Andi.
- L. Mercer, P. Prusinkiewicz, J. Hanan. The concept and design of a Virtual Laboratory. In *Graphics Interface '90 Conference proceedings*, pages 149-155. Canadian Information Processing Society, 1990.
- Pressman, Roger S. 2010. *Software Engineering : A Practitione's Approach*, Sevent Edition. McGraw Hill.
- PR, Kuncoro, Purwadi, Suyatno Aris, dan Widiyanto, Henang. 2007. *Kimia SMA/MA Kls XII*. Jakarta: PT Grasindo.
- Prayitno, Wendhie; S, Kom. 2005. *Desain Model Sistem Perangkat Lunak dengan UML*.
- Rosa A. S, M.Salahuddin. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika : Bandung.
- Sommerville, Ian. 2007. "Software Engineering". Penerbit Addison-Wesley. United Kingdom
- Sunendar, T. 2007. *Pemanfaatan Laboratorium Kimia Virtual*. Penelitian Tindakan Kelas. Bandung: SMAN Kota Bandung.
- Suwahono. 2010. Keselamatan Kerja Laboratorium. [http://www.chem-is-try.org/materi\\_kimia/kimia-aplikasi/manajemen-laboratorium-kimia/keselamatan-kerja-laboratorium](http://www.chem-is-try.org/materi_kimia/kimia-aplikasi/manajemen-laboratorium-kimia/keselamatan-kerja-laboratorium). 15 Januari 2014.
- Suwahono. 2010. Sumber Bahaya yang Berpotensi Menimbulkan Kecelakaan Kerja. [http://www.chem-is-try.org/materi\\_kimia/kimia-aplikasi/manajemen-laboratorium-kimia/sumber-bahaya-yang-berpotensi-menimbulkan-kecelakaan-kerja](http://www.chem-is-try.org/materi_kimia/kimia-aplikasi/manajemen-laboratorium-kimia/sumber-bahaya-yang-berpotensi-menimbulkan-kecelakaan-kerja). 15 Januari 2014.
- Sunarya, Yayan, dan Setiabudi, Agus. 2009. *Mudah dan Aktif Belajar Kimia 1: Untuk Kelas X Sekolah Menengah Atas/Madrasah Aliyah*. Jakarta: Depdiknas.

Tejedor, J.A.G, Martinez, G.M, dan Vidaurre, C.B . 2008. An Online Virtual Laboratory of Electricity. International Journals of Distance Education Technologies. Volume 6 , No 2.

Tim Guru Indonesia. 2011. Target Nilai Rapor 10 Kupas Habis Semua Pelajaran SMP/MTs Kelas VII. Jakarta: Wahyumedia.

Unity. 2014. Unity Documentation. <http://unity3d.com/unity>. 26 Maret 2014.

Waldjinah, dan Wulandari, Ema Tri. 2014. Kimia Kelas X. Klaten: Intan Pariwara.



## LAMPIRAN A CARA KERJA PERCOBAAN

### A.1 Percobaan Ikatan Kimia



#### Praktikum

#### Kepolaran Senyawa

##### A. Tujuan

Mengamati kepolaran berbagai larutan.

##### B. Alat dan Bahan

1. Statif
2. Buret

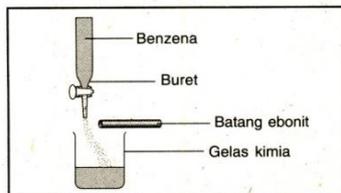
##### 3. Corong

4. Batang ebonit
5. Gelas kimia
6. Larutan benzena
7. Larutan nitro benzena

8. Larutan trikloro metana
9. Larutan karbon tetraklorida
10. Akuades
11. Larutan natrium klorida

##### C. Cara Kerja

1. Pasang buret pada statif. Pastikan keran sudah tertutup rapat.
2. Isi buret dengan benzena.
3. Alirkan benzena dari buret ke dalam gelas kimia secara perlahan-lahan dan dekatkan ebonit pada aliran benzena tersebut. Perhatikan gambar berikut ini! Amati apa yang terjadi.



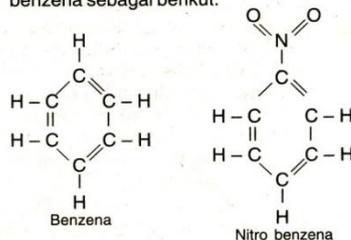
4. Ulangi langkah 1–3, tetapi benzena diganti dengan nitro benzena, trikloro metana, karbon tetraklorida, dan natrium klorida.
5. Lakukan dengan *jujur, disiplin, teliti*, dan *proaktif* dalam kerja sama kelompok.

##### D. Hasil Pengamatan

No.	Bahan	Aliran Zat Cair	
		Diblokkan	Tidak Diblokkan
1.	Benzena	...	...
2.	Nitro benzena	...	...
3.	Trikloro metana	...	...
4.	Karbon tetraklorida	...	...
5.	Akuades	...	...
6.	Natrium klorida	...	...

##### E. Pertanyaan

1. Cairan manakah yang dipengaruhi oleh ebonit?
2. Bagaimana keadaan cairan yang terpengaruh ebonit?
3. Bentuk molekul benzena dan nitro benzena sebagai berikut.



Bentuk molekul manakah yang simetris dan bentuk molekul manakah yang asimetris?

4. Pada molekul nitro benzena ( $C_6H_5NO_2$ ), keelektronegatifan  $C_6H_5$  lebih besar daripada keelektronegatifan  $NO_2$ . Dengan demikian, lebih tertarik ke manakah pasangan elektron ikatan antara  $C_6H_5$  dengan  $NO_2$ ?
5. Berdasarkan percobaan, bahan apa sajakah yang bersifat polar dan bahan apa sajakah yang bersifat nonpolar?
6. Samakah kepolaran benzena dengan nitro benzena? Jelaskan alasannya!

##### F. Unjuk Kreativitas

Berdasarkan prosedur pada praktikum di atas, gantilah bahan-bahan tersebut dengan minyak, larutan garam, dan larutan gula! Bagaimanakah kepolaran bahan-bahan tersebut?

Di laboratorium, terdapat banyak larutan yang bersifat polar dan nonpolar. Untuk mengetahui kepolaran larutan-larutan tersebut, kembangkanlah *kreativitas* dan *rasa ingin tahu* Anda dengan menguji kepolaran berbagai larutan encer yang tersedia di laboratorium seperti prosedur dalam praktikum di atas. Dengan mengetahui arah aliran larutan, Anda dapat menentukan kepolaran suatu larutan. Pengetahuan tentang kepolaran larutan dapat Anda gunakan untuk menentukan jenis larutan yang cocok digunakan sebagai pelarut sehingga pemborosan penggunaan bahan kimia dapat dihindari.

### 3. Ikatan Logam

Ikatan logam adalah ikatan antaratom dalam suatu unsur logam dengan menggunakan interaksi antarelektrovalensi. Unsur logam mempunyai kecenderungan untuk menjadi ion positif karena energi potensial ionisasi yang rendah dan mempunyai elektron valensi kecil. Ketika atom-atom logam yang bermuatan ini saling berdekatan, kemudian elektron valensinya akan terdelokalisasi membentuk "lautan elektron" di sekitar ion-ion positif. Lautan elektron ini akan bertindak sebagai perekat atom-atom logam.

Gambar A.1. Cara Kerja Percobaan Ikatan Kimia



## A.2 Percobaan Larutan Elektrolit dan Non-Elektrolit

### Evaluasi 3

Jawablah pertanyaan-pertanyaan berikut ini dengan tepat!

1. Bandingkan derajat disosiasi larutan elektrolit lemah, elektrolit kuat, dan nonelektrolit!
2. Diberikan senyawa-senyawa  $H_2SO_4$ ,  $Mg(OH)_2$ , dan  $C_2H_5OH$ .
  - a. Partikel apa saja yang terdapat dalam larutan ketika masing-masing senyawa tersebut dilarutkan dalam air?
  - b. Tuliskan reaksi yang terjadi ketika senyawa-senyawa tersebut dilarutkan dalam air!
3. Bagaimana Anda dapat membedakan larutan elektrolit lemah dan elektrolit kuat? Jelaskan!
4. Kelompokkan senyawa-senyawa berikut ke dalam larutan elektrolit lemah, elektrolit kuat, dan nonelektrolit. Jelaskan pula alasan Anda.  
Barium sulfat, vitamin C, asam nitrat, asam fluorida, aluminium hidroksida, barium hidroksida, etanol, kalsium klorida, glukosa.
5. Gambarkan dan jelaskan bagaimana proses yang terjadi pada larutan elektrolit ketika dihubungkan dengan sumber listrik dan bola lampu melalui rangkaian percobaan elektrolisis. Tunjukkan pada gambar tersebut arah aliran elektron dan perpindahan ion-ionnya.
6. Dengan menggunakan larutan HCl sebagai larutan uji elektrolisis, tunjukkan reaksi-reaksi elektrode yang menghasilkan gas di kedua elektrode.

### Kegiatan Siswa 9: Praktikum

#### Pengujian Daya Hantar Listrik Larutan

##### I. Tujuan Percobaan

Mengamati gejala hantaran listrik berbagai larutan dengan pelarut air.

##### II. Peralatan dan Bahan-bahan

###### Peralatan:

- Baterai 1,5 – 6 V
- Kabel secukupnya
- Gelas
- Bola lampu 5 W
- Elektrode karbon (pensil) 2 buah
- Kertas tisu

###### Bahan-bahan:

- Larutan HCl 1 mol/L
- Air suling
- Larutan NaCl 1 mol/L
- Larutan NaOH 1 mol/L
- Larutan gula
- Larutan garam dapur
- Larutan cuka 1 mol/L
- Air laut

##### III. Prosedur

1. Isilah gelas dengan larutan HCl 1 mol/L sebanyak 100 mL.
2. Masukkan dua buah elektrode pensil ke dalam gelas berisi larutan HCl.
3. Pasanglah dua buah kabel pada bola lampu.
4. Hubungkan satu kabel dari bola lampu dengan salah satu elektrode dan satu kabel lainnya dari bola lampu tersebut dengan baterai melalui bagian yang bermuatan positif.
5. Hubungkan elektrode yang satunya lagi dengan baterai melalui ujung yang bermuatan negatif.
6. Amati intensitas nyala lampu dan gelembung gas yang terjadi pada kedua elektrode dalam larutan.
7. Lakukan hal yang sama terhadap larutan-larutan yang lain.

###### Perhatian!

Sebelum dan sesudah percobaan untuk setiap larutan, bilaslah setiap elektrode dengan air suling dan keringkan dengan kertas tisu.

Gambar A.2. Cara Kerja Percobaan Larutan Elektrolit dan Non-Elektrolit

## A.3 Percobaan Hukum Proust

### Kegiatan Siswa 7: Praktikum

- I. **Tujuan:** Menentukan perbandingan massa magnesium dan oksigen dalam magnesium oksida
- II. **Peralatan dan Bahan-bahan:**

#### Peralatan:

- Lumpang porselin kecil dengan tutupnya untuk peleburan (1)
- Kawat segitiga dengan porselin (1)
- Tungku berkaki tiga (tripod) (1)

#### Bahan-bahan, Kimia:

- Pembakar (1)
- Timbangan (1)
- Pita magnesium, 15 – 30 cm
- Penggaris

370

KIMIA untuk SMA/MA Kelas

### III. Dasar Teori

Atom-atom dalam senyawa memiliki perbandingan massa yang tetap. Contoh, air tersusun dari hidrogen dan oksigen dan selalu mengandung 2 g hidrogen untuk setiap 16 gram oksigen. Oleh karena itu, rasio massa hidrogen terhadap oksigen adalah 1 : 8. Dengan hal yang sama, rasio massa oksigen terhadap tembaga dalam CuO adalah 1 : 4 dan rasio massa belerang terhadap besi dalam FeS adalah 4 : 7. Proust menyatakan bahwa perbandingan massa unsur-unsur dalam senyawa adalah tetap.

Dari rasio ini kita dapat merumuskan bahwa komposisi rumus dan senyawa selalu tetap. Contoh, pembakaran karbon (C) dengan oksigen di udara menghasilkan karbon dioksida ( $\text{CO}_2$ ) dan pembakaran minyak bumi dengan oksigen di udara juga menghasilkan karbon dioksida ( $\text{CO}_2$ ). Jadi, karbon dioksida yang dihasilkan dari dua reaksi yang berbeda memiliki massa molar dan rumus yang sama ( $\text{CO}_2$ ).

### IV. Prosedur

#### Penyiapan

1. Potonglah pita magnesium sepanjang 5 cm dan 10 cm. **Perhatian:** Pastikan bahwa permukaan pita bebas dari oksida.
2. Timbanglah lumpang peleburan beserta tutupnya yang bersih dan kering. Catatlah nilai ini dalam "Tabel Pengamatan dan Data".
3. Ambillah pita magnesium yang 5 cm dan tempatkan dalam lumpang peleburan, kemudian timbanglah, termasuk dengan penutupnya. Catatlah nilai ini dalam "Tabel Pengamatan dan Data".

#### Prosedur:

1. Panaskan lumpang peleburan dengan menggunakan pembakar hingga magnesium terbakar. **Catatan:** Mula-mula nyala api harus rendah, kemudian panas dinaikkan perlahan-lahan hingga magnesium mulai terbakar. Nyala api tidak boleh terlalu besar untuk menghindari hilangnya magnesium oksida.
2. Ambillah pembakar dan angkatlah tutup lumpangnya beberapa saat agar udara masuk. **Catatan:** Pastikan bahwa asap yang dapat keluar dari lumpang sesedikit mungkin.
3. Ulangi pemanasan, angkat dan turunkan tutupnya untuk melihat dan menjamin bahwa semua magnesium telah terbakar.
4. Segera setelah magnesium tampak terbakar semua, hentikan pembakaran.
5. Hentikan pemanasan dan biarkan lumpang peleburan menjadi dingin. Selanjutnya, timbanglah lumpang peleburan beserta isi dan tutupnya dan catatlah dalam "Tabel Pengamatan dan Data".
6. Ulangilah pengerjaan 1 – 5 untuk pita magnesium 10 cm dan catatlah hasilnya dalam "Tabel Pengamatan dan Data". **Catatan:** lebih banyak pita magnesium yang dicoba akan memberikan hasil percobaan yang lebih sensitif.

### V. Pengamatan dan Data

1. Pengamatan

.....  
 .....

2. Tuliskan data pengukuran Anda dalam tabel berikut:

	Pita magnesium 5 cm	Pita magnesium 10 cm
Berat lumpang dan penutup ( $W_1$ )	.....g	.....g
Berat lumpang, penutup, dan magnesium ( $W_2$ )	.....g	.....g
Berat lumpang, penutup, dan magnesium oksida ( $W_3$ )	.....g	.....g

Gambar A.3. Cara Kerja Percobaan Hukum Proust

## LAMPIRAN B EVALUASI PROGRAM

Evaluasi program merupakan lampiran hasil dari setiap percobaan pada program yang menunjukkan bahwa hasil yang didapat dari setiap objek yang diuji sesuai dengan teori-teori yang berlaku yang dijadikan dasar teori pembuatan sistem. Adapun evaluasi program dibagi kedalam 3 bahasan sesuai dengan materi disetiap percobaannya, 3 bahasan tersebut antara lain:

### B.1 Percobaan Ikatan Kimia

Pada percobaan ikatan kimia dalam program didapatkan hasil dari setiap larutan yang diuji sebagai berikut, ditunjukkan pada tabel B.1.

**Tabel B.1. Hasil Percobaan Materi Ikatan Kimia**

No	Nama Larutan	Jenis Larutan
1	H <sub>2</sub> O / Air	Polar
2	Minyak Tanah	Non- Polar
3	C <sub>2</sub> H <sub>6</sub> OH / Alkohol	Polar
4	CCl <sub>4</sub> / Tetraklorometana	Non-Polar
5	CHCl <sub>3</sub> / Triklorometana	Polar

Hasil diatas sesuai dengan teori yang terdapat dalam buku pelajaran Kimia Kelas X yang menyatakan bahwa dalam ikatan kovalen dapat terjadi kutub listrik yang disebabkan adanya polarisasi ikatan. Besarnya kekuatan gaya tarik elektron dari suatu atom dinyatakan sebagai harga keelektronegatifan, semakin besar perbedaan harga keelektronegatifan akan menyebabkan semakin polar ikatannya. Sehingga ikatan yang terjadi pada atom-atom yang memiliki perbedaan keelektronegatifan disebut dengan ikatan kovalen polar, sebaliknya ikatan pada atom-atom yang tidak memiliki perbedaan keelektronegatifan akan membentuk ikatan kovalen nonpolar (Waldjinah, 2014 : 94-95).

Kepolaran suatu senyawa dapat ditentukan oleh beberapa hal antara lain, jumlah momen dipol, bentuk molekul, dan jenis unsur pembentuk senyawa. Dari keterangan ini didapatkan hasil menurut teori yang antara lain dijelaskan dalam tabel B.2 (Waldjinah, 2014 : 94-97).

**Tabel B.2. Penggolongan larutan polar dan nonpolar**

Jenis Larutan	Contoh
Polar	H <sub>2</sub> O, CHCl <sub>3</sub> , C <sub>2</sub> H <sub>6</sub> OH
Nonpolar	CCl <sub>4</sub> , Minyak Tanah

### B.2 Percobaan Larutan Elektrolit dan Non-Elektrolit

Pada percobaan larutan elektrolit dan non-elektrolit dalam program didapatkan hasil dari setiap larutan yang diuji sebagai berikut, ditunjukkan pada tabel B.3.

**Tabel B.3. Hasil Percobaan Materi Larutan Elektrolit dan Non-Elektrolit**

No	Nama Larutan	Jenis Larutan
1	H <sub>2</sub> O / air	Non-Elektrolit
2	HCl / Asam Klorida	Elektrolit Kuat
3	Air Suling	Non-Elektrolit
4	CH <sub>3</sub> COOH / Larutan Cuka	Elektrolit Lemah
5	Larutan Gula	Non-Elektrolit
6	NaOH / Natrium Hidroksida	Elektrolit Kuat
7	NaCl / Natrium Klorida	Elektrolit Kuat

Hasil diatas sesuai dengan teori yang terdapat dalam buku pelajaran Kimia untuk SMA/MA kelas X yang menyatakan bahwa, larutan elektrolit kuat diperoleh dengan melarutkan zat-zat terlarut ionik atau kovalen polar yang dapat terdisosiasi atau terionisasi sempurna dalam air. Sehingga didapat zat-zat ini memiliki derajat disosiasi = 1, sedangkan untuk elektrolit lemah derajat elektrolit yang didapat < 1, dan untuk larutan non-elektrolit derajat disosiasi = 0. (Haris Watoni, A., 2013 : 258-259)

Adapun hasil dari tiap-tiap zat terlarut yang merupakan elektrolit kuat, lemah, atau non-elektrolit yang dijelaskan dalam teori antara lain dijelaskan dalam tabel b.4 (Haris Watoni, A., 2013 : 258-265):

**Tabel B.4. Penggolongan larutan elektrolit dan non-elektrolit**

Derajat Disosiasi/ionik	Jenis Elektrolit	Contoh
= 1	Kuat	HCl, NaCl, NaOH
< 1	Lemah	Larutan Cuka (CH <sub>3</sub> COOH)
0	Nonelektrolit	Glukosa (C <sub>6</sub> H <sub>12</sub> O <sub>6</sub> ), air (H <sub>2</sub> O)

### B.3 Percobaan Hukum Proust

Sedangkan pada percobaan hukum proust hasil yang didapatkan dalam program yaitu perbandingan massa unsur H dan O dalam senyawa H<sub>2</sub>O sama dengan perbandingan massa atom penyusun yakni 1:8 baik sebelum dan sesudah reaksi dengan panjang pita magnesium yang berbeda-beda. Hal ini sesuai dengan perbandingan dari massa atom 2H dan O yang bernilai 1:8 yang dapat dilihat pada tabel periodik. Tabel periodik ditunjukkan pada gambar B.1.

**Gambar B.1. Tabel Periodik**

Sumber: <http://www.bacajuga.com/wp-content/uploads/2013/05/tabel-periodik.jpg>

Hal ini membuktikan bahwa hasil percobaan pada program telah sesuai dengan bunyi hukum proust yaitu perbandingan massa unsur-unsur dalam senyawa adalah tetap, dan dengan rumus dari hukum proust didapati hasil yang sesuai dengan teori dari Hukum Proust itu sendiri, rumus hukum proust sendiri yaitu (Haris Watoni, A., 2013 : 365-366) :

$$\frac{\text{massa unsur}}{\text{massa senyawa}} = \frac{i_{\text{unsur}} \times (A_r \text{ unsur})}{M_r \text{ senyawa}}$$

UNIVERSITAS BRAWIJAYA

