



**SISTEM PENGACAKAN LATIHAN SOAL UJIAN NASIONAL SMA  
MENGUNAKAN METODE *LINEAR CONGRUENT***

**SKRIPSI**

oleh :

**Feby Dinda Rizvi**

**112410101059**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2015**



**SISTEM PENGACAKAN SOAL UJIAN NASIONAL SMA  
MENGUNAKAN METODE *LINEAR CONGRUENT***

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi dan mencapai gelar Sarjana Komputer

oleh :

**Feby Dinda Rizvi**

**112410101059**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2015**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang masih memberikan saya kesehatan dan waktu luang juga nikmat – nikmat yang tak terhitung jumlahnya;
2. Kedua orangtua saya, Ayahanda H. Dimiyati Hamid dan Ibunda Hj. Iswinarti Rahmaningsih;
3. Nenek tercinta yang selalu menemani saya meskipun sering saya tinggal;
4. Saudaraku Andis Rizki Pahlevi S.TP yang telah bersedia memberikan tempat berteduh di Ibu kota selama pengerjaan skripsi ini;
5. Saudara seiman Kodak dan Ahmad B. yang telah memberikan keringatnya dan saudari W. A. N. Faizah;
6. Calon ibu dari anak – anak saya kelak.
7. Guru-guruku sejak sekolah dasar hingga perguruan tinggi yang formal maupun nonformal;
8. Barisan para mantan dan yang telah pergi tanpa sempat aku miliki
9. Seluruh teman-teman yang selalu memberikan bantuan dan dukungan;
10. Almamater Program Studi Sistem Informasi Universitas Jember.

**MOTO**

*“Sebaik-baik manusia adalah manusia yang bermanfaat bagi manusia lainnya”*



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Feby Dinda Rizvi

NIM : 112410101059

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pengacakan Soal Ujian Nasional SMA Menggunakan Metode *Linear Congruent*”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 22 Juni 2016

Yang menyatakan,

Feby Dinda Rizvi

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Sistem Pengacakan Soal Ujian Nasional SMA Menggunakan Metode *Linear Congruent*”, telah diuji dan disahkan pada :

hari, tanggal : 22 Juni 2016

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Prof. Drs. Slamir M.CompSc.,Ph.D  
NIP 196704201992011001

Windi Eka Yulia R, S.Kom., MT  
NIP 198403052010122002

**SKRIPSI**

**SISTEM PENGACAKAN SOAL UJIAN NASIONAL SMA  
MENGUNAKAN METODE *LINEAR CONGRUENT***

oleh :

Feby Dinda Rizvi

112410101059

Pembimbing :

Dosen Pembimbing Utama : Prof. Drs. Slamun M.CompSc.,Ph.D

Dosen Pembimbing Pendamping : Winda Eka Yulia R, S.Kom., MT

**PENGESAHAN**

Skripsi berjudul “Sistem Pengacakan Soal Ujian Nasional SMA Menggunakan Metode *Linear Congruent*”, telah diuji dan disahkan pada :

hari, tanggal : 22 Juni 2016

tempat : Program Studi Sistem Informasi Universitas Jember.

Penguji I,

Anang Andrianto ST., MT  
NIP 196906151997021002

Penguji II,

Yanuar Nurdiansyah ST., M.Cs  
NIP 198201012010121004

Mengesahkan  
Ketua Program Studi,

Prof. Drs. Slamini, M.Comp.Sc.,Ph.D  
NIP 196704201992011001



## PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Pengacakan Soal Ujian Nasional SMA Menggunakan Metode *Linear Congruent*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Utama dan Windi Eka Yulia R, S.Kom.,MT selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Ibu Nelly Oktavia A, S.Si., MT selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Teman-teman seperjuangan dan seangkatan.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 22 Juni 2016

Penulis

**DAFTAR ISI**

PERSEMBAHAN .....	ii
MOTO .....	iii
PERNYATAAN.....	iv
PRAKATA.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN .....	xiv
BAB 1. PENDAHULUAN .....	1
BAB 2. TINJAUAN PUSTAKA .....	5
2.1    Pengertian Sistem .....	5
2.2    Metode Latihan ( <i>Drill</i> ) .....	5
2.3    Distribusi Soal .....	5
2.4 <i>Linear Congruent Methode</i> (LCM) .....	6
2.5 <i>Hypertext PreProcessors</i> (PHP).....	7
2.6    Ujian Akhir Nasional (UAN) .....	7
2.7    Model <i>Waterfall</i> .....	7
BAB 3. METODOLOGI PENELITIAN.....	15
3.1    Tahapan Penelitian .....	15
3.2    Jenis Penelitian .....	16
3.3    Teknik Pengumpulan Data .....	16
3.3.1    Studi Literatur .....	16
3.3.2 <i>Interview</i> .....	16
3.4    Pembobotan soal.....	17
3.5    Tahap Analisis .....	18
3.6    Tahap Pengembangan Sistem.....	19
BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM.....	20
4.1    Pengumpulan Data .....	20

4.2	Desain Sistem .....	20
4.2.1	<i>Statement of Purpose</i> .....	21
4.2.2	Analisis Kebutuhan .....	21
4.2.3	<i>Bussiness Process</i> .....	22
4.2.4	<i>Usecase Diagram</i> .....	23
4.2.5	<i>Scenario</i> .....	25
1.	<i>Activity diagram</i> .....	31
2.	<i>Sequence diagram</i> .....	34
3.	<i>Class diagram</i> .....	40
4.	<i>Entity Relationship Diagram ( ERD )</i> .....	42
b.	Penulisan Kode Program .....	43
c.	Pengujian Sistem .....	43
1.	<i>White Box Testing</i> .....	43
2.	<i>Black Box Testing</i> .....	59
BAB 5. HASIL DAN PEMBAHASAN.....		63
5.1	Sistem Pengacakan Latihan Soal Ujian Nasional Dengan Menggunakan <i>Linear Congruent Methode</i> .....	63
5.2	Hasil Implementasi Sistem Pengacakan Soal SMA .....	71
BAB 6. PENUTUP .....		80
6.1	Kesimpulan.....	80
6.2	Saran .....	81
LAMPIRAN .....		133

DAFTAR TABEL

Tabel 2.1 Tabel pengujian <i>blackbox</i> .....	10
Tabel 2.2 Tabel pengujian data normal dan salah.....	11
Tabel 4.1 Definisi <i>usecase</i> .....	24
Tabel 4.2 Definisi Aktor .....	25
Tabel 4.3 Skenario manajemen data guru .....	25
Tabel 4.4 Skenario manajemen data soal.....	27
Tabel 4.5 Skenario sistem pengacakan soal.....	29
Tabel 4.6 <i>Test Case</i> fitur manajemen data guru.....	46
Tabel 4.7 <i>Test Case</i> fitur manajemen data soal .....	49
Tabel 4.8 <i>Test Case</i> pengacakan soal.....	56
Tabel 4.9 Pengujian <i>Black Box</i> fitur manajemen data guru .....	59
Tabel 4.10 Pengujian <i>Black Box</i> manajemen data soal.....	60
Tabel 4.11 Pengujian <i>Black Box</i> fitur pengacakan soal .....	61
Tabel 5.1 Perhitungan Manual Sistem .....	66
Tabel 5.2 Perhitungan pembangkitan bilangan acak.....	67
Tabel 6.1 Skenario manajemen data siswa .....	83
Tabel 6.2 Skenario manajemen data user.....	85
Tabel 6.3 Skenario manajemen data bobot soal.....	87
Tabel 6.4 Skenario manajemen data soal .....	89
Tabel 6.5 Skenario latihan soal .....	91
Tabel 6.6 Skenario manajemen jadwal latihan .....	92

**DAFTAR GAMBAR**

Gambar 2.1 Alur distribusi soal .....	6
Gambar 2.2 Ilustrasi Model <i>Waterfall</i> (Pressman R. , 2012).....	8
Gambar 2.3 Contoh <i>Listing Program</i> (Pressman, 2012) .....	12
Gambar 2.4 Contoh Diagram Alir (Pressman, 2012).....	13
Gambar 3.1 Diagram Alir Penelitian (Sumber:Hasil Analisis, 2015).....	15
Gambar 3.2 Pembobotan dasar ujian nasional .....	17
Gambar 3.3 Diagram alir sistem ( sumber:hasil analisis, 2015 ) .....	18
Gambar 4.1 <i>Business proces</i> sistem pengacakan soal sma .....	22
Gambar 4.2 <i>Usecase diagram</i> sistem pengacakan soal ujian nasional sma.....	23
Gambar 4.3 <i>Activity diagram</i> manajemen data guru.....	31
Gambar 4.4 <i>Activity diagram</i> manajemen data soal.....	32
Gambar 4.5 <i>Activity diagram</i> pengacakan soal.....	33
Gambar 4.6 <i>Sequence diagram</i> manajemen data guru.....	35
Gambar 4.7 <i>Sequence diagram</i> manajemen data soal.....	37
Gambar 4.8 <i>Sequence diagram</i> pengacakan soal .....	39
Gambar 4.9 <i>Class diagram system</i> pengacakan soal .....	41
Gambar 4.10 <i>Entity Relationship Diagram</i> ( ERD ) sistem pengacakan soal .....	42
Gambar 4.11 Kode Program Update Guru.....	44
Gambar 4.12 Kode Program Insert Guru .....	44
Gambar 4.13 Kode Program Hapus .....	44
Gambar 4.14 Diagram alir fitur manajemen data guru .....	45
Gambar 4.15 Kode Program Update Soal.....	47
Gambar 4.16 Kode Program Update Soal.....	47
Gambar 4.17 Kode Program Update Soal.....	47
Gambar 4.18 Diagram alir fitur manajemen data soal .....	48
Gambar 4.19 Kode Program function acak soal ( ).....	53
Gambar 4.20 Diagram alir function acak soal ( ).....	54
Gambar 5.1 <i>Database</i> bobot soal .....	63
Gambar 5.2 <i>Database</i> soal 1 - 200.....	64
Gambar 5.3 <i>Database</i> soal 200 - 300.....	64
Gambar 5.4 <i>Database</i> soal 300 - 350.....	64
Gambar 5.5 <i>Database</i> soal 351 - 400.....	65
Gambar 5.6 <i>Database</i> soal 401 - 439.....	65
Gambar 5.7 <i>Execute code</i> acak soal.....	69
Gambar 5.8 Tampilan Acak Soal .....	70
Gambar 5.9 Tampilan user guru mengsis <i>text field</i> .....	70
Gambar 5.10 Tampilan setelah klik acak soal .....	70
Gambar 5.11 Tampilan <i>login</i> .....	72

Gambar 5.12 Tampilan <i>Dashboard home</i> .....	72
Gambar 5.13 Tampilan Master Data User .....	73
Gambar 5.14 Tampilan Tabel Data Admin.....	73
Gambar 5.15 Tampilan Tabel Data Guru.....	74
Gambar 5.16 Tampilan Tabel Data Siswa .....	74
Gambar 5.17 Tampilan Tabel Data Mapel.....	75
Gambar 5.18 Tampilan <i>form</i> Data Bobot Soal.....	75
Gambar 5.19 Tampilan <i>form</i> Tabel Data Soal .....	76
Gambar 5.20 Tampilan <i>form</i> Data Pengacakan Soal .....	76
Gambar 5.21 Tampilan <i>form</i> Data Penjadwalan Latihan.....	77
Gambar 5.22 Tampilan Menu Dashboard Siswa .....	77
Gambar 5.23 Tampilan pengumuman.....	78
Gambar 5.24 Tampilan uji distribusi <i>empiris</i> .....	78
Gambar 5.25 Tampilan Mata Pelajaran <i>Try Out</i> .....	79

**DAFTAR LAMPIRAN**

**LAMPIRAN A. PERANCANGAN SISTEM ..... 83**



## BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan.

### 1.1 Latar Belakang

Ujian Nasional adalah upaya pemerintah untuk mengevaluasi tingkat pendidikan secara nasional dengan menetapkan standarisasi nasional pendidikan (Tilaar., 2006). Ujian Nasional (UN) adalah ujian yang dilakukan oleh negara kepada seluruh siswa dari sekolah dasar hingga sekolah menengah atas. Hal ini dilakukan berdasarkan Undang-Undang No. 20 tahun 2003 tentang Sistem Pendidikan Nasional (Sisdiknas) dan Keputusan Mendiknas No 153/U/2003 tentang Ujian Akhir Nasional (UAN) merupakan dasar penyelenggaraan Ujian Akhir Nasional yang bertujuan untuk meningkatkan mutu pendidikan nasional

Ada beberapa cara yang sering digunakan oleh sekolah maupun siswa dalam mempersiapkan diri dalam menghadapi ujian nasional. Diantaranya, mengadakan les disekolah, belajar di bimbingan belajar, mengadakan tryout baik dari sekolah maupun dari pihak luar. Namun hal tersebut bukanlah mudah, disamping memerlukan waktu dan biaya yang banyak, jenis-jenis soal nya juga sudah dapat diprediksi. Metode Latihan merupakan suatu cara mengajar dengan memberikan latihan latihan berstruktur dan sistematis terhadap apa yang telah dipelajari siswa sehingga memperoleh suatu keterampilan dengan memberikan latihan soal kepada siswa (Rijani, 2011). Metode latihan (*Driil*), merupakan suatu cara mengajar yang baik untuk menanamkan kebiasaan- kebiasaan tertentu. Latihan adalah suatu teknik mengajar yang mendorong siswa untuk melaksanakan kegiatan latihan agar memiliki ketangkasan / keterampilan yang lebih tinggi dari apa yang dipelajari. Fungsi dan manfaat metode latihan (*driil*) sesuai untuk keterampilan dapat dikuasai. Beberapa hal yang harus diperhatikan dalam penggunaan teknik latihan. Dan Untuk mengatasi masalah diatas, kami berkeinginan merancang sebuah perangkat lunak berbasis *web* yang dapat



mengacak soal-soal dan membentuknya menjadi paket- paket soal yang dapat digunakan untuk latihan ujian nasional. Perangkat lunak tersebut mengacak soal-soal dengan menggunakan *Linear Congruent Method* (LCM). Dengan mengacak soal-soal secara *random* maka akan dihasilkan soal yang baru.

Pembangkit bilangan acak atau *random number generator* adalah suatu algoritma yang digunakan untuk menghasilkan urutan-urutan atau *sequence* dari angka-angka sebagai hasil dari perhitungan dengan komputer yang diketahui distribusinya sehingga angka-angka tersebut muncul secara random dan digunakan terus-menerus (Nasution, 2013). *Linear Congruent Method* (LCM) merupakan salah satu metode pembangkit bilangan acak. Ciri khas dari LCM adalah terjadi perulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan. Penentuan konstanta LCM ( $a$ ,  $c$  dan  $m$ ) sangat menentukan baik tidaknya bilangan acak yang diperoleh dalam arti memperoleh bilangan acak yang seakan-akan tidak terjadi perulangan.

## 1.2 Rumusan Masalah

Berdasarkan uraian diatas maka dirumuskan permasalahan sebagai berikut:

1. Bagaimana menerapkan metode *linear congruent* pada sistem pengacakan latihan soal ujian akhir nasional SMA agar soal yang ditampilkan acak.
2. Bagaimana membangun sistem pengacakan latihan soal ujian nasional SMA menggunakan metode *linear congruent*

## 1.3 Tujuan

Tujuan dari penelitian adalah :

1. Menerapkan metode *linear congruent* pada sistem pembelajaran ujian akhir nasional SMA dalam menampilkan soal secara acak
2. Membangun sistem pengacakan latihan soal ujian nasional SMA menggunakan metode *linear congruent*

## 1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Mata pelajaran yang dipakai dalam aplikasi ini hanya berfokus pada fisika saja.
2. Pembuatan aplikasi pembelajaran ujian akhir Nasional SMA ini berbasis *web*
3. Penerapan metode *linear congruent* (LCM) hanya pada sebatas menampilkan soal secara *random*

### 1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

#### 1. Pendahuluan

Bab ini terdiri atas latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika penulisan.

#### 2. Tinjauan Pustaka

Bab ini berisi tentang kajian materi dan informasi apa saja yang digunakan dalam penelitian ini. Dimulai dari kajian pustaka mengenai penjelasan dari sistem pengacakan latihan soal ujian akhir nasional SMA

#### 3. Metodologi Penelitian

Bab ini menguraikan tentang metodologi penelitian apa yang dilakukan selama penelitian. Dimulai dari tahap pencarian permasalahan hingga pengujian sistem pengacakan latihan soal ujian akhir nasional SMA yang akan dibuat.

#### 4. Desain dan Pengembangan Aplikasi

Bab ini akan menguraikan tentang desain dan pengembangan aplikasi pembuatan aplikasi secara keseluruhan. Proses perancangan aplikasi dimulai dari analisis kebutuhan fungsional dan non-fungsional aplikasi, dilanjutkan dengan pembuatan *usecase diagram*, skenario *usecase*, *sequence diagram*, *activity diagram*, *class diagram* dan *entity relationship diagram* (ERD).

#### 5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang telah dilakukan dengan memaparkan hasil penelitian dan hasil percobaan pengimplementasian sistem.

#### 6. Penutup

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.



## BAB 2. TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan teori-teori dan pustaka yang digunakan dalam penelitian. Teori yang dibahas meliputi teori tentang, sistem, metode latihan (*Drill*), metode *linear congruent* (LCM) , *Hypertext PreProcessors* (PHP), Ujian Akhir Nasional (UAN), model *waterfall*.

### 2.1 Pengertian Sistem

Sistem merupakan komponen-komponen yang bekerja sama untuk mencapai tujuan tertentu. Sistem adalah suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu lingkungan kompleks (Marimin, 2009). Jadi dengan demikian dapat disimpulkan bahwa sistem adalah media yang didukung oleh komponen-komponen yang memiliki keterkaitan satu sama lain & dibatasi oleh suatu aturan tertentu untuk mencapai aturan tertentu.

### 2.2 Metode Latihan (*Drill*)

Metode *drill* adalah metode dalam pengajaran dengan melatih peserta didik terhadap bahan yang sudah diajarkan atau berikan agar memiliki ketangkasan atau ketrampilan dari apa yang telah dipelajari. Menurut Hernawo (2008) Metode *drill* (latihan) disebut juga metode training, merupakan suatu cara mengajar yang baik untuk menanamkan kebiasaan-kebiasaan tertentu. *Drill* merupakan suatu cara mengajar dengan memberikan latihan-latihan terhadap apa yang telah dipelajari siswa sehingga memperoleh suatu keterampilan tertentu.

### 2.3 Distribusi Soal

Distribusi soal merupakan suatu proses penyampaian soal ujian dari pembuat soal sampai proses *input* ke dalam *database*, tahapan yang terdapat dalam proses distribusi soal ujian digunakan untuk mengetahui darimana soal – soal tersebut berasal dan akan sampai mana soal akan berakhir. Alur distribusi soal dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur distribusi soal

#### 2.4 *Linear Congruent Methode (LCM)*

Nomor acak generator (RNG) adalah perangkat yang dirancang untuk menghasilkan urutan angka atau simbol yang tidak memiliki pola apapun. (Mary Evanchalin .S, 2014). Jenis PRNG yang dibahas disini adalah *Linear Congruential Generator (LCG)*. LCG merupakan jenis PRNG yang banyak digunakan dalam aplikasi komputer *modern* (Djati, 2007) LCG ditemukan oleh D.H Lehmer. LCM memanfaatkan model linier untuk membangkitkan bilangan acak yang didefinisikan pada persamaan [1].

$$Z_i = ((a Z_{i-1}) + c) \bmod m \dots\dots\dots [1]$$

Dimana :  $Z_i$  = bilangan acak ke  $-i$

$Z_{i-1}$  = bilangan acak sebelumnya

$a$  = faktor pengali

$c$  = *increment*

$m$  = modulus

Menurut Thomas J. Kikay dalam bukunya Pengantar Sistem Simulasi menyebutkan bahwa syarat-syarat untuk menentukan konstanta dalam LCM adalah sebagai berikut :

1. Konstanta  $a$  harus lebih besar dari  $\sqrt{m}$
2. Untuk koefisien  $c$ , harus berangka ganjil apabila  $m$  bernilai pangkat dua. Dan koefisien  $c$  tidak boleh nilai dari kelipatan  $m$
3. Untuk  $m$  harus bilangan prima
4. Untuk pertama  $Z_0$  harus merupakan angka integer dan juga ganjil cukup besar

## 2.5 *Hypertext PreProcessors (PHP)*

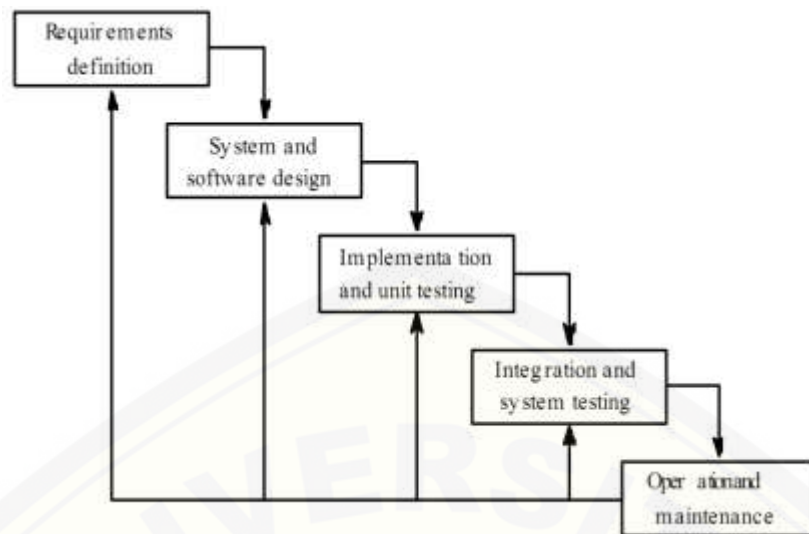
PHP adalah singkatan dari *Hypertext PreProcessors*, *PHP* saat ini adalah bahasa pemrograman interpreter yang paling banyak digunakan saat ini dikarenakan bersifat *open source* dan juga paling banyak didukung oleh banyak *web server*. PHP dapat digunakan oleh banyak sistem operasi dari Windows, Linux maupun BSD. PHP umumnya diintegrasikan dengan aplikasi *database* yang juga *open source* seperti *MySQL* maupun *PostgreSql*, tapi bisa juga diintegrasikan dengan *Microsoft SQL*, *Access* maupun *Oracle*.

## 2.6 **Ujian Akhir Nasional (UAN)**

Sedangkan menurut H. A. R. Tilaar, Ujian Nasional adalah upaya pemerintah untuk mengevaluasi tingkat pendidikan secara nasional dengan menetapkan standarisasi nasional pendidikan. Hasil dari Ujian Nasional yang diselenggarakan oleh Negara adalah upaya pemetaan masalah pendidikan dalam rangka menyusun kebijakan pendidikan nasional. Berdasarkan pendapat tersebut tentang Ujian Nasional maka dapat disimpulkan bahwa Ujian Nasional adalah sistem evaluasi atau penilaian standar pendidikan dasar dan menengah secara nasional dengan menetapkan standarisasi nasional pendidikan yang bertujuan sebagai pemetaan masalah pendidikan dalam rangka menyusun kebijakan pendidikan nasional.

## 2.7 **Model Waterfall**

Menurut Pressman (2012), model *waterfall* merupakan sebuah pendekatan pengembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Alur *life cycle* pengembangan perangkat lunak pada model *waterfall* diselesaikan pertahapan dan berurutan. Alur dari model *waterfall* dapat dilihat pada Gambar 2.2 Ilustrasi Model *Waterfall* .



Gambar 2.2 Ilustrasi Model *Waterfall* (Pressman R. , 2012)

Keterangan dari skema di atas adalah :

1. Analisis Kebutuhan

Menganalisis kebutuhan yang akan digunakan dalam pembuatan aplikasi. Meliputi pengumpulan data kebutuhan fungsional dan non-fungsional dari aplikasi yang akan kita bangun. Setelah itu, menentukan fungsi dan fasilitas apa saja yang akan dibuat dalam aplikasi.

2. Desain Sistem

Jika proses analisis kebutuhan telah diketahui maka proses selanjutnya adalah pada tahapan desain sistem. Proses pendesainan sistem dari aplikasi yang akan kita bangun yaitu dengan menggunakan *Unified Modeling Language* (UML). Penggunaan UML karena sudah menggunakan konsep *Object Oriented Design* yang tentunya akan sangat memudahkan *developer* untuk membangun sebuah sistem. *UML* diagram yang akan dibuat antara lain:

a. *Business Process*

*Business Proses* digunakan untuk menggambarkan *input-an* data yang dibutuhkan sistem, output dari sistem serta tujuan dari pembuatan sistem.

b. *Use Case Diagram*

*Use case* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

c. *Scenario*

*Scenario* diagram digunakan untuk menjelaskan atau menceritakan fitur atau isi yang ada di *use case* diagram. *Scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.

d. *Sequence Diagram*

*Sequence diagram* (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/message.

e. *Activity Diagram*

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram mempunyai fungsi yang sama dengan *scenario* namun diimplementasikan dalam diagram alir.

f. *Class Diagram*

*Class* diagram digunakan untuk menggambarkan struktur statis class dalam sistem. *Class Diagram* dibuat untuk memudahkan dalam proses pengkodean. *Entity Relationship Diagram*

g. ERD

merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

3. *Coding* (Pengkodean)

Setelah proses desain sistem dikerjakan, proses selanjutnya adalah *coding* atau penulisan kode program. Bahasa pemrograman yang dipakai adalah *Java* dengan *tool / IDE* yang digunakan berupa NetBean. Proses pengkodean menggunakan konsep OOP ( *Object Oriented Programming* ) dengan tujuan programmer akan lebih mudah dalam melakukan *coding*, karena konsep OOP mengikuti model yang telah ada dalam kehidupan nyata. Semua bagian (*entity*) dari suatu permasalahan adalah objek adalah konsep dari OOP. Objek-objek ini kemudian juga dapat berupa gabungan dari beberapa objek yang lebih kecil. Objek besar dapat dibentuk dengan menggabungkan beberapa objek-objek dalam



bahasa pemrograman. Objek-objek tersebut berkomunikasi dengan saling interaksi kepada objek lain.

#### 4. Pengujian / Testing

Pengujian wajib dilakukan untuk menguji apakah sistem ini sudah sesuai dengan kebutuhan dari user atau belum. Dan apakah masih ada kesalahan maupun kelemahan terhadap sistem yang kami bangun tersebut. Diharapkan proses pengujian / testing dapat menyempurnakan sistem yang kami buat. Pengujian yang dilakukan melibatkan semua aspek sistem meliputi *hardware*, *software* aplikasi, *environment software*, penempatan aplikasi, dan *user* yang menggunakan aplikasi ini. Pengujian perangkat lunak menggunakan dua metode yakni :

##### 1. *Black Box Testing*

*Black Box Testing* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Wildan Agissa, 2013).

Pada pengujian *black box* ini, aplikasi yang dibangun pada penelitian ini akan diuji dengan mengujikan langsung *running aplikasi* dan melakukan kegiatan pengujian dengan menganalisis proses *input* dan *output* yang dihasilkan aplikasi.

Adapun tabel pengujian disusun seperti Tabel 2.1 berikut :

Tabel 2.1 Tabel pengujian *blackbox*

Kelas Uji	Butir Uji	Jenis Pengujian
-----------	-----------	-----------------

Keterangan Tabel :

- a. **Kelas Uji** : Merupakan fungsi aplikasi yang akan diujikan.
- b. **Butir Uji** : Rincian fitur yang diuji dari fungsi yang terdapat pada aplikasi.
- c. **Jenis Pengujian** : Metode pengujian yang dilakukan , yaitu *black box*.

Dalam metode *black box* juga dilakukan pengujian dengan cara memasukkan data normal dan data salah, dari masukan ini nantinya akan dilakukan analisis terhadap reaksi yang muncul pada aplikasi. Contoh tabel pengujian untuk *event* yang terjadi ketika ada data masukan dapat dilihat pada Tabel 2.2 dibawah ini.

Tabel 2.2 Tabel pengujian data normal dan salah

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username:admin Password : admin Klik tombol login	Aplikasi menampilkan semua menu yang dapat diakses oleh user, yaitu menu <i>input</i> data mapel, data guru, data murid, data admin	Dapat masuk ke tampilan utama dan menampilkan menu sesuai yg diharapkan	[x]diterima [ ]ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Username:admin Password : admin Klik tombol login	Tidak dapat login dan masuk ke tampilan utama serta menampilkan peringatan bahwa username dan password salah	Tidak dapat login dan masuk ke tampilan utama serta menampilkan peringatan bahwa username dan password salah	[x]diterima [ ]ditolak


## 2. *White Box Testing*

Merupakan cara pengujian dengan melihat modul untuk yang telah dibuat dengan program – program yang ada. Dan menganalisa apakah terjadi kesalahan atau tidak pada penulisan kode program. Pengujian ini dilakukan oleh

(*develeoper*) pembuat program. Jika ada modul yang menghasilkan output yang tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan di cek dan diperbaiki, kemudian di compile ulang (Agissa 2013). Menurut Pressman (2002) pengujian *white box* merupakan teknik pengujian jalur dasar yang digunakan untuk menentukan kompleksitas logis dengan menentukan rangkaian dasar jalur eksekusinya. Tahapan teknik pengujian jalur dasar meliputi dari mulai *listing* program, grafik alir, kompleksitas siklomatik, jalur program independen dan pengujian basis set.

a. *Listing* Program

Merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program. Contoh penerapan tahapan ini dapat dilihat pada Gambar 2.3



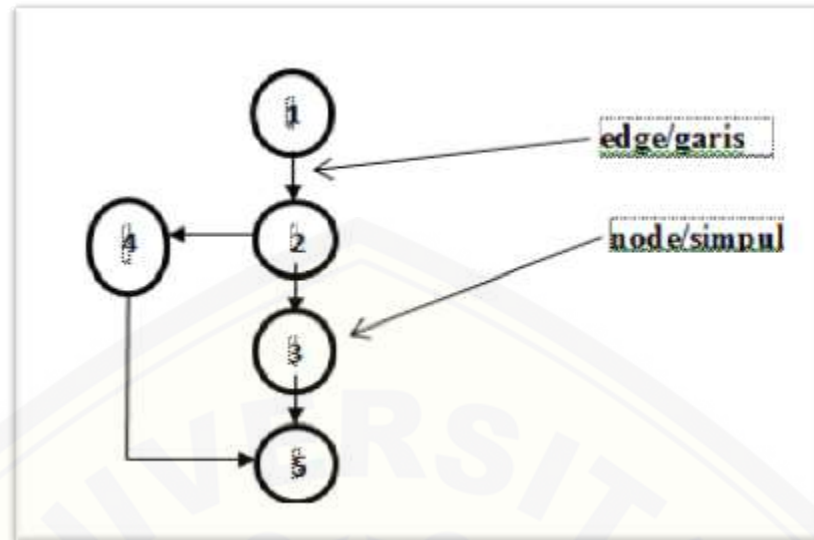
```
Spanjang = $_POST['p'];
$lebar   = $_POST['l'];
if($spanjang == $lebar)
{
    $jenisBangun = 'Persegi';
}else
    {
        $jenisBangun = 'Persegi Panjang';
    }

$luas = $spanjang * $lebar;
echo 'Luas bangun '.$jenisBangun.' adalah '.$luas;
```

Gambar 2.3 Contoh *Listing* Program (Pressman, 2012)

b. Grafik Alir

Menurut Pressman (2012) Grafik alir merupakan Sebuah notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari *listing* program. Grafik alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program. Contoh penggambaran diagram alir dapat dilihat pada Gambar 2.4.



Gambar 2.4 Contoh Diagram Alir (Pressman, 2012)

### c. Kompleksitas Siklomatik

Kompleksitas Siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program (Pressman, 2012). Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program (Pressman, 2012). Rumus yang digunakan untuk menghitung kompleksitas siklomatika yaitu:

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

Keterangan :

$V(G)$ : Kompleksitas Siklomatik

$E$  : Jumlah *Edge*

$N$  : Jumlah *Node*

Berdasarkan grafik alir yang ada pada tahapan kedua diketahui jumlah edge adalah 5 dan jumlah node adalah 5, sehingga dapat dihitung kompleksitas siklomatisk  $V(G) = E - N + 2 = 5 - 5 + 2 = 2$ . Jadi jumlah jalur independen adalah 2 jalur.

d. Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru (Pressman, 2012). Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi (Pressman, 2012). Dari perhitungan kompleksitas siklomatik *Basis set* yang dihasilkan dari jalur independent secara linier adalah 2 jalur, yaitu:

Jalur 1 : 1-2-3-5

Jalur 2 : 1-2-4-5

e. Pengujian Basis Set

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di *basis set*. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati *basis set* yang tersedia. Sistem telah memenuhi syarat kelayakan software jika salah satu jalur yang dieksekusi setidaknya satu kali. Dari tahap sebelumnya telah diketahui 2 *basis set* Jika kemudian diuji dengan memasukkan data panjang = 5 dan lebar 3, maka basis set jalur yang digunakan adalah 1-2-4-5. Dapat dilihat bahwa jalur telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem ini telah memenuhi syarat.

5. *Maintenance*

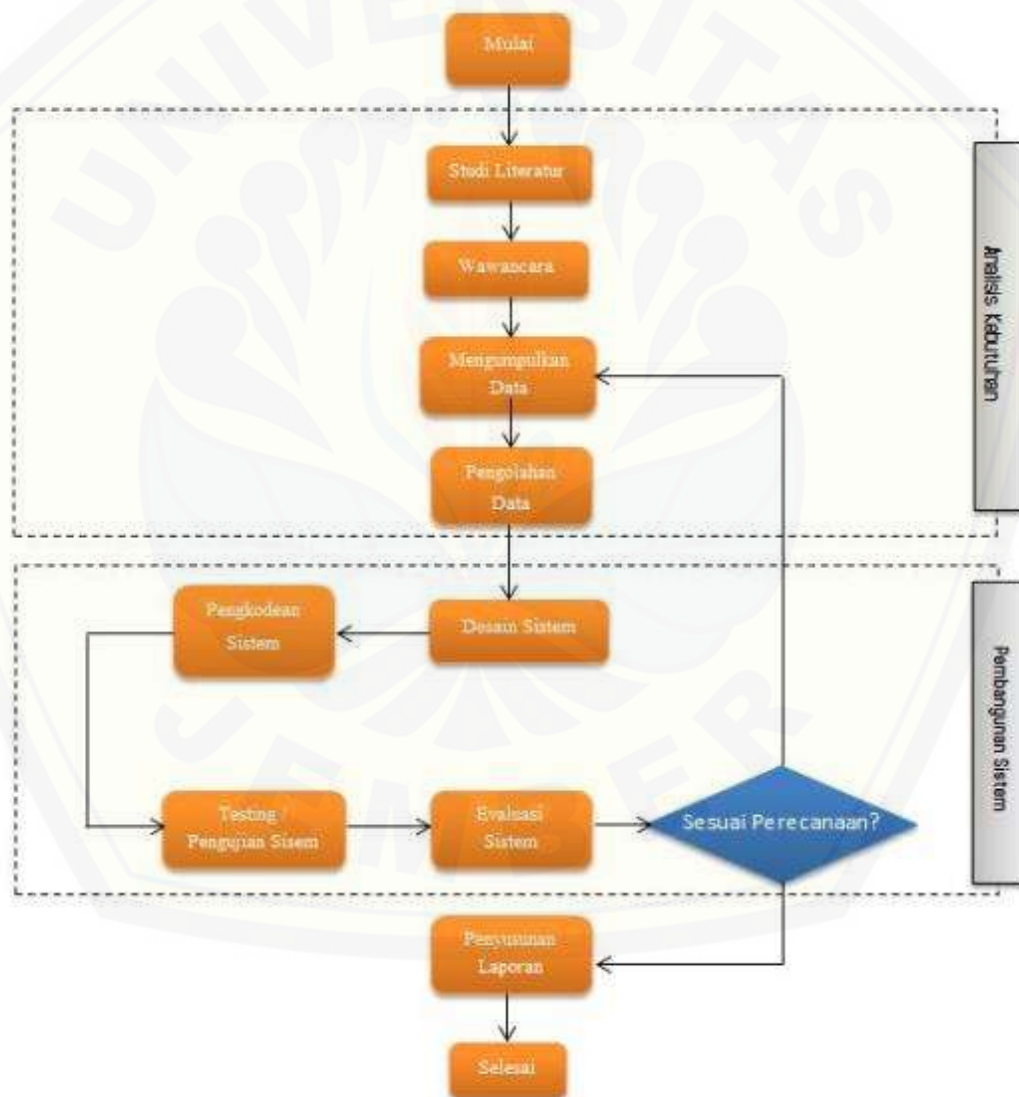
Perawatan diadakan untuk mengatasi masalah pada sistem dilain waktu ketika aplikasi sudah dapat digunakan oleh *user*. Selama *user* menemui *bug* pada aplikasi ini, maka *user* langsung dapat mengkonfirmasi kepada *developer* untuk segera ditangani oleh *developer*.

### BAB 3. METODOLOGI PENELITIAN

Pada bab ini dijelaskan tentang metode-metode yang digunakan selama penelitian dilakukan, seperti jenis penelitian, studi literatur, data dan sumber data penelitian, serta tahapan analisis hingga model perancangan sistem.

#### 3.1 Tahapan Penelitian

Penelitian akan dilaksanakan dalam beberapa tahap, tahapan alur penelitian untuk membuat sistem pengacakan soal dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian (Sumber:Hasil Analisis, 2015)

### 3.2 Jenis Penelitian

Pada penelitian ini digunakan jenis penelitian kualitatif. Jenis penelitian kualitatif digunakan karena penelitian ini menganalisa studi literatur dan melakukan *interview* untuk pengumpulan sampel data.

### 3.3 Teknik Pengumpulan Data

Proses untuk mendapatkan data yang dibutuhkan untuk membangun sistem pengacakan latihan soal ujian akhir nasional menggunakan dua cara, yaitu Studi Literatur dan melakukan *interview* untuk pengumpulan sampel data.

#### 3.3.1 Studi Literatur

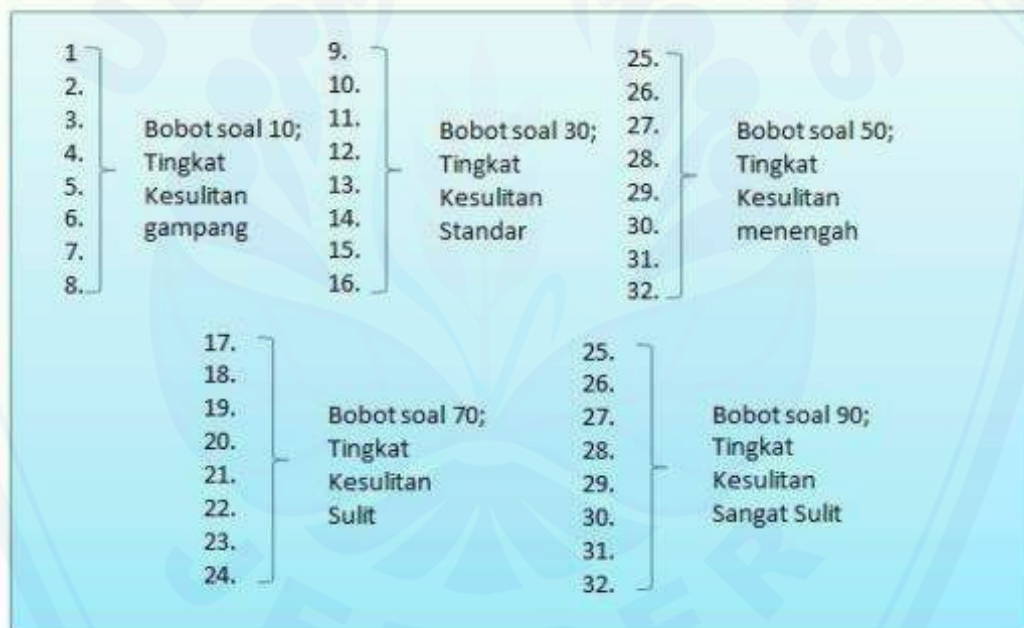
Pada tahap ini dilakukan pengumpulan data dan informasi yang diperlukan untuk proses perancangan sistem. Data dan informasi dapat diperoleh dari lokasi penelitian yaitu di SMADA Jember dengan guru guru fisika Bapak Hadiyanto S.Pd. Selain itu, studi literatur juga dapat diperoleh dari *paper*, jurnal ilmiah, serta buku-buku referensi yang berkaitan dengan penelitian.

#### 3.3.2 Interview

Teknik penyerapan pengetahuan terdiri atas dua bagian utama, yaitu identifikasi proyek dan penyerapan pengetahuan. Pelaksanaan penyerapan pengetahuan bisanya dilakukan dengan wawancara (*interview*). Metode wawancara yang digunakan adalah diskusi bebas (*talk through*), pembicaraan atas dasar kasus yang berkaitan dengan soal-soal terbaru dan cara penyelesaiannya. Penulis dalam penelitian ini melakukan wawancara secara langsung pada salah seorang guru yang berkompeten di bidang fisika yang dalam hal ini adalah Bapak Hadiyanto S.Pd yang sudah sangat berpengalaman dan banyak mendapat penghargaan di tingkat daerah maupun nasional dan mempunyai banyak sekali kumpulan soal soal UN dan soal – soal yang mendukung digunakan dalam latihan ujian.

### 3.4 Pembobotan soal

Dalam evaluasi pembelajaran diperlukan pedoman penskoran yang dapat digunakan sebagai petunjuk menilai pekerjaan siswa (Danielson, 1997). Soal Pedoman pembobotan adalah pedoman yang digunakan untuk menentukan skor hasil penyelesaian pekerjaan siswa. Skor ini kemudian ditafsirkan menjadi nilai. Kesulitan yang dihadapi adalah menetapkan skor dengan tepat terhadap penyelesaian pekerjaan siswa, baik tugas, ulangan, atau yang lain. Konsistensi penskoran sangat penting untuk pemerolehan hasil penilaian antar siswa yang tidak bisa dikarenakan penilaian guru yang tidak konsisten. Pembobotan yang terdapat dalam ujian soal ini terbagi menjadi 5, dengan tingkat kesulitan soal mudah, standar, menengah, sulit, dan sangat sulit. Dapat dilihat pada Gambar 3.2.

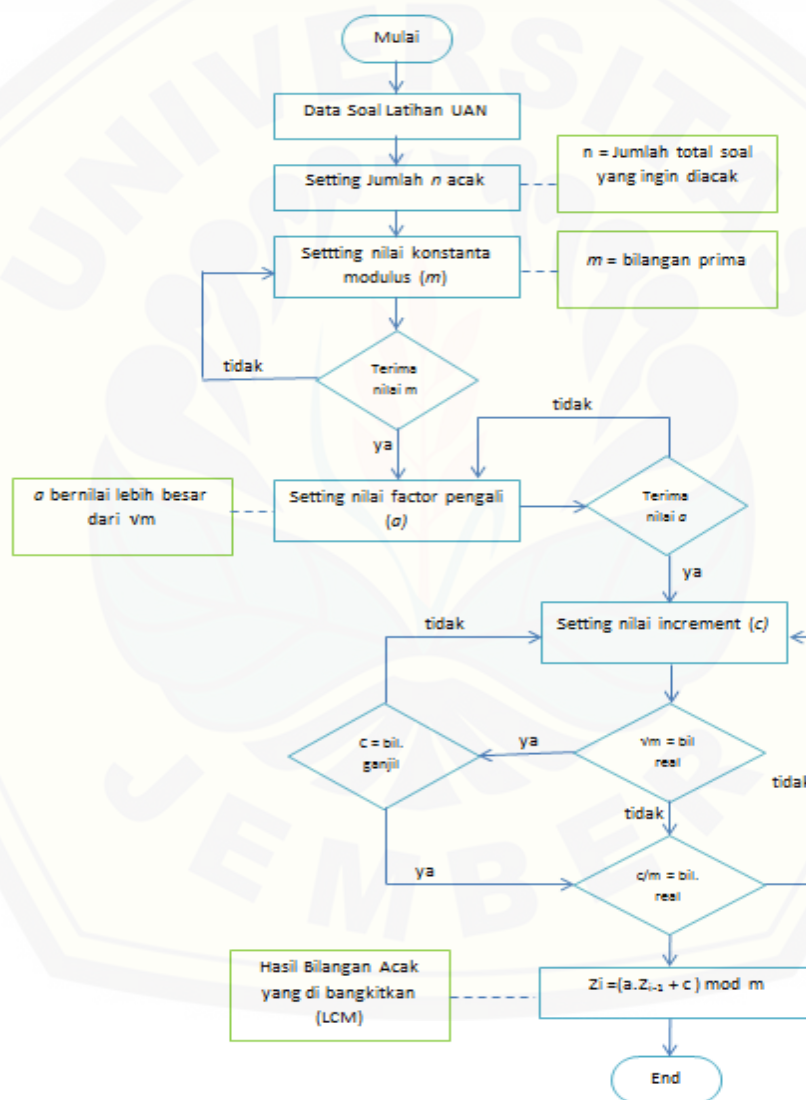


Gambar 3.2 Pembobotan dasar ujian nasional



### 3.5 Tahap Analisis

Tahap analisis dilakukan setelah melakukan pengumpulan data mengenai jenis-jenis soal yang paling banyak muncul dalam setiap ujian akhir. Data yang diperoleh selanjutnya akan dianalisa dan pengacakan soal untuk latihannya menerapkan metode *linear congruent*. Proses alur algoritma metode *linear congruent* pembuatan sistem pengacakan soal ujian nasional SMA dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram alir sistem ( sumber:hasil analisis, 2015 )

### 3.6 Tahap Pengembangan Sistem

Didalam pembuatan sistem pengacakan latihan soal ujian nasional SMA ini mengikuti tahapan *software development life cycle* (SDLC) *waterfall*. Penggunaan SDLC *waterfall* bertujuan untuk memudahkan alur pembuatan sistem. Tahapan SDLC dengan metode *waterfall* meliputi tahapan analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan.

Setelah tahap pengumpulan data selesai, selanjutnya akan dilanjutkan ke perancangan sistem dengan menggunakan konsep berbasis objek dengan pemodelan *Unified Modelling Language* (UML). Pemodelan UML yang digunakan pada penelitian ini antara lain, *Business Process*, *Usecase Diagram*, *Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class diagram* dan *Entity Relationship Diagram* (ERD). Setelah tahap perancangan selesai, dilanjutkan dengan tahap implementasi menggunakan bahasa pemrograman *Hypertext PreProcessors* (PHP) dan menggunakan basis data *MySQL*. Hasil perancangan dan implementasi kemudian akan diuji menggunakan metode pengujian *White box* dan *Black box*. Pada *White box testing* metode pengujian yang diuji berupa modul - modul dari program yang telah buat. Pengujian *White box testing* dilakukan oleh *developer* dimana yang bertindak sebagai *developer* atau pembuat program adalah penulis sendiri. Sedangkan pada *Black box testing* metode pengujiannya yaitu dengan cara memeriksa fungsionalitas dari program yang telah dibuat. Pengujian *Black box testing* pada program ini dilakukan oleh Ahmad Baihaqi S.Kom.

## BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM

Bab ini akan membahas tentang analisis dan pengembangan untuk membuat sistem pengacakan latihan soal ujian nasional SMA menggunakan metode *linear congruent*. Tahapan-tahapan perancangan dilaksanakan berdasarkan metode waterfall, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, dilanjutkan dengan pembuatan usecase diagram, skenario, activity diagram, sequence diagram, class diagram dan *entity relation diagram* (ERD).

### 4.1 Pengumpulan Data

Pengumpulan data dilakukan agar sistem yang dibangun sesuai dengan kebutuhan user. Pengumpulan data dilakukan dengan teknik wawancara langsung kepada guru fisika SMAN 2 Jember bapak Hadiyanto S.Pd dan literatur soal-soal ujian tahun sebelumnya.

Metode yang digunakan dalam sistem pengacakan latihan soal ujian nasional SMA merupakan metode *linear congruent*, metode ini digunakan untuk mengacak soal-soal latihan, dimana data dari soal di *database* yang dipanggil akan berbeda pada tiap pemanggilan. Soal yang akan ditampilkan pada sistem pengacakan soal ini sebanyak 439 soal. Dalam *database* terdapat berbagai macam soal yang diambil dari soal-soal UN tahun lalu dan berbagai literatur yang mendukung.

Proses awal dari penerapan metode *linear congruent method* (LCM) adalah menentukan jumlah bilangan yang akan diacak (439). Selanjutnya ditentukan pula nilai dari faktor pengali (200), *increment* (217) dan modulusnya (439). Setelah itu dihitung sesuai rumus perhitungan LCM.

### 4.2 Desain Sistem

Pengembangan sistem yang digunakan oleh penulis adalah pengembangan model *waterfall*. Pengembangan model *waterfall* terdiri dari beberapa tahapan

yaitu analisis kebutuhan, desain sistem, penulisan kode program, *testing*, penerapan program dan *maintenance*.

#### 4.2.1 Statement of Purpose

Aplikasi pengacakan latihan soal ujian nasional SMA membantu pihak sekolah dalam membuat pengacakan soal latihan UAN. Sistem ini akan membantu guru SMA dalam menentukan pengacakan soal latihan UAN. Aplikasi pengacakan latihan soal ujian nasional SMA ini memiliki beberapa fitur antara lain : fitur mengelola master data guru, master data mata pelajaran, master data soal latihan, fitur membangkitkan bilangan acak untuk pengacakan soal dan fitur report untuk menampilkan laporan data. Fitur-fitur tersebut diharapkan dapat memberikan pengacakan latihan soal yang optimal .

#### 4.2.2 Analisis Kebutuhan

Analisis kebutuhan perangkat lunak dalam penelitian ini yaitu dengan cara mengidentifikasi literature-literatur yang sesuai untuk kebutuhan pelajar pada tingkat sekolah menengah atas pada pelajaran fisika. Analisis kebutuhan yang dilakukan meliputi proses pengumpulan data kebutuhan fungsional dan non fungsional. Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Sedangkan kebutuhan nonfungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Berikut adalah kebutuhan fungsional dan non-fungsional sistem :

##### a. Kebutuhan fungsional dari sistem yang akan dibangun

1. Sistem dapat *input* data admin. Sistem dapat update data admin.
2. Sistem dapat delete data admin.
3. Sistem dapat mengelola *input* data guru.
4. Sistem dapat mengelola *update* data guru.
5. Sistem dapat mengelola *delete* data guru.
6. Sistem dapat mengelola *input* data siswa.
7. Sistem dapat mengelola *update* data siswa.

8. Sistem dapat mengelola *delete* data siswa.
9. Sistem dapat mengelola data soal.
10. Sistem dapat menampilkan laporan data.
11. Sistem dapat mengacak soal.

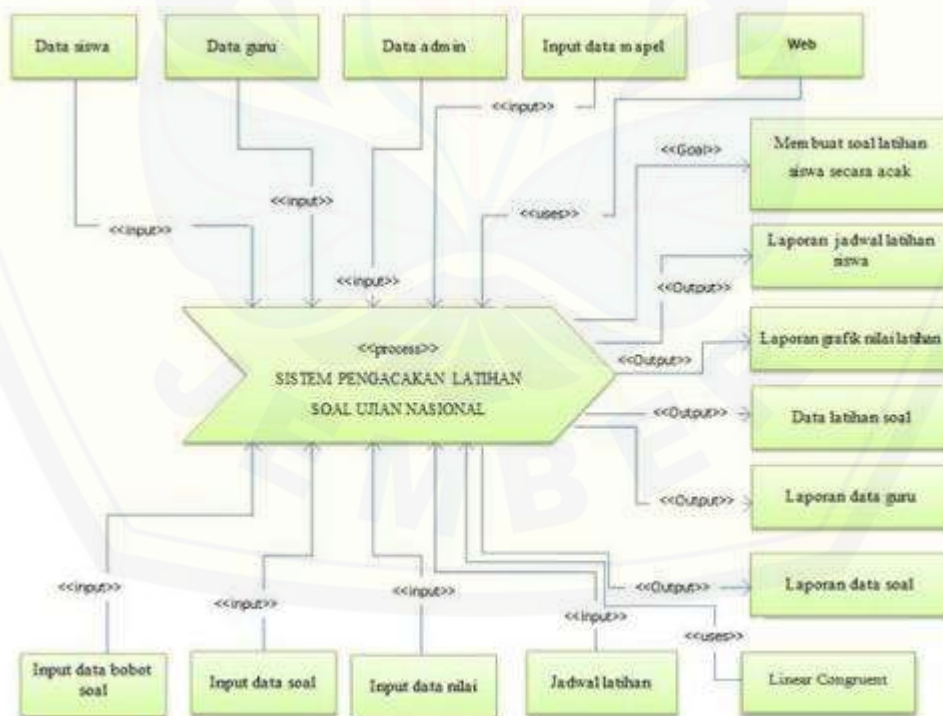
b. Kebutuhan non-fungsional dari sistem yang akan dibangun

1. Sistem berbentuk *web*
2. Sistem menggunakan *PHP Code Igniter* dengan konsep OOP ( *Object Oriented Programming* )

#### 4.2.3 Business Process

*Business Process* digunakan untuk menggambarkan *inputan* data yang dibutuhkan sistem, output dari sistem serta tujuan dari pembuatan sistem.

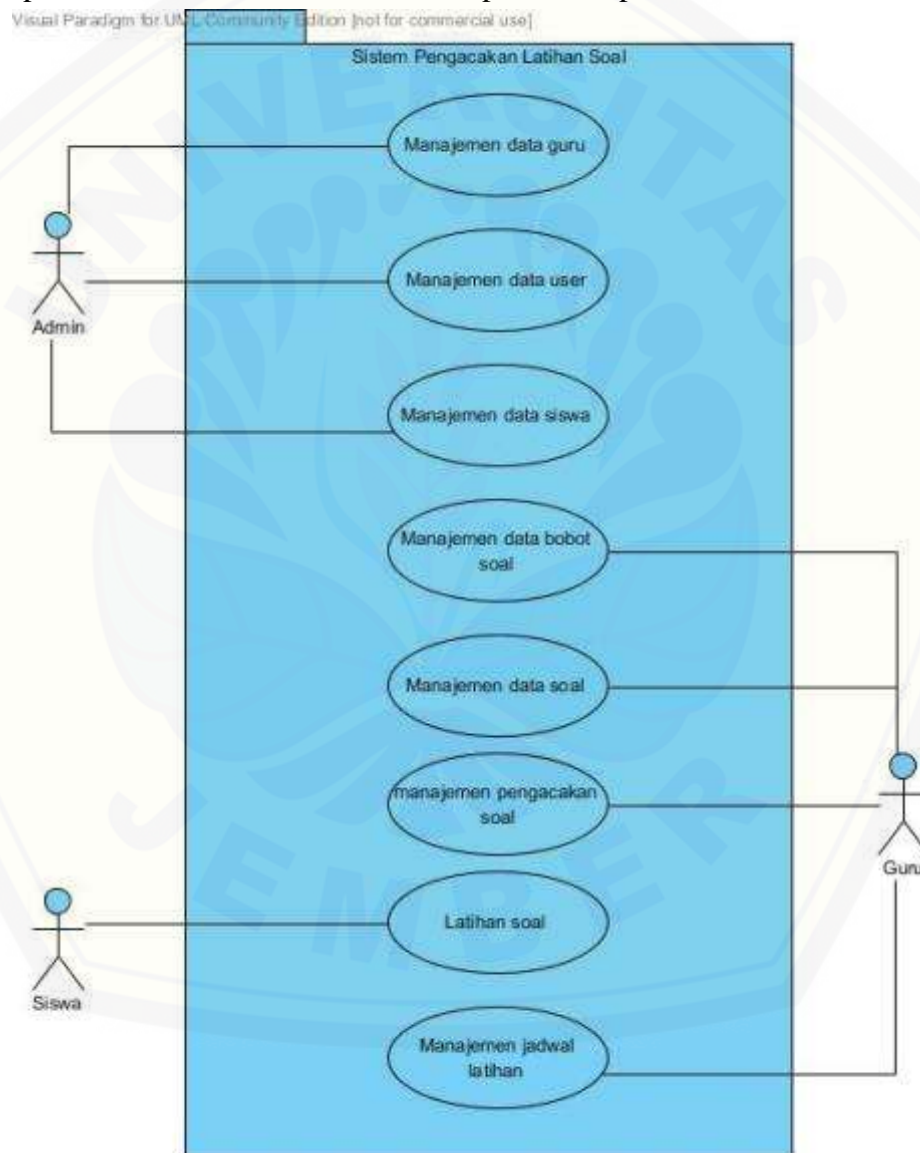
*Business Process* sistem dapat dilihat pada Gambar 4.1



Gambar 4.1 *Business proces* sistem pengacakan soal sma

#### 4.2.4 Usecase Diagram

*Usecase Diagram* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Usecase Diagram* berfungsi untuk menggambarkan fitur apa saja yang akan dijalankan pada *software* yang akan dibuat. *Usecase Diagram* sistem sistem pengacakan latihan soal dapat dilihat pada Gambar 4.2. Definisi *usecase* dapat dilihat pada Tabel 4.1 dan definisi aktor dapat dilihat pada Tabel 4.2.



Gambar 4.2 *Usecase diagram* sistem pengacakan soal ujian nasional sma

Tabel 4.1 Definisi *usecase*

No.	Usecase	Deskripsi
1.	Manajemen data guru	Proses manajemen data dimana user admin dapat menambahkan, mengupdate, dan menghapus data guru
2.	Manajemen data user	Proses manajemen data oleh admin untuk menambah atau mengedit data user admin baru.
3.	Manajemen data siswa	Proses manajemen data dimana user admin dapat menambahkan, mengupdate dan menghapus data siswa.
4.	Manajemen pengacakan soal	Proses manajemen data oleh guru yang mana guru membuat latihan soal secara acak dan soalnya diambil dari database soal.
5.	Manajemen data bobot soal	Proses manajemen data oleh guru dalam menentukan bobot soal pada setiap soal.
6.	Manajemen data soal	Proses manajemen data oleh guru dalam menambah, update atau menghapus data soal.
7.	Latihan soal	Siswa melihat soal untuk kemudian dikerjakan oleh siswa dimana data soal diambil dari databse soal.
8.	Manajemen jadwal latihan	Proses manajemen data oleh guru dalam menambah, update atau menghapus data jadwal latihan.

Tabel 4.2 Definisi Aktor

No.	Usecase	Deskripsi
1.	Admin	Admin merupakan aktor yang dapat melihat, menambah, mengubah dan menghapus data user, data guru dan data siswa
2.	Guru	Pengguna sistem yang mengelola data manajemen pengacakan soal, manajemen data soal, manajemen data bobot soal, report data soal, manajemen jadwal latihan, view report data nilai.
3.	Siswa	Pengguna melihat

#### 4.2.5 Scenario

*Scenario* berfungsi untuk menggambarkan alur sistem beserta alternatif alur yang akan dijalankan oleh user pada sistem pengacakan soal. Berikut adalah skenario dari sistem yang akan dibuat.

Tabel 4.3 Skenario manajemen data guru

ID	:	UCS-01
Name	:	Manajemen data guru
<i>Participating Aktor</i>	:	Admin
<i>Entry Condition</i>	:	Admin login ke sistem
<i>Exit Condition</i>	:	Admin sukses menambahkan data, mengubah data atau menghapus data guru.

#### SKENARIO NORMAL (LIHAT)

<i>Actor</i>	<i>System</i>
1. Klik menu data guru	
	2. Menampilkan halaman 'Data Guru' dengan tombol tambah, ubah, dan hapus.

#### SKENARIO NORMAL (TAMBAH)



<i>Actor</i>	<i>System</i>
3. Klik tombol tambah	
	4. Menampilkan 'Form Tambah Guru'
5. Mengisi 'Form Tambah Guru'	
6. Klik tombol 'simpan'	
	7. Cek <i>input-an</i> data
	8. Menyimpan data ke dalam database
	9. Menampilkan halaman 'Daftar Guru' yang telah diperbaharui.

#### SKENARIO ALTERNATIF

<i>Actor</i>	<i>System</i>
6. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input-annya</i> salah	
	7. Menampilkan warning 'masih ada data yang kosong/salah'

#### SKENARIO NORMAL (UBAH)

3. Klik tombol 'ubah'	
	4. Menampilkan halaman 'Form Ubah Guru'
5. Mengubah 'Form Ubah Guru'	
6. klik tombol 'simpan'	
	7. Cek <i>input-an</i> data
	8. Memperbaharui data di database
	9. menampilkan halaman 'Daftar Guru' yang telah diperbaharui

#### SKENARIO ALTERNATIF

<i>Actor</i>	<i>System</i>
6. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input-</i>	

annya salah	
	7. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (HAPUS)</b>	
<b>Actor</b>	<b>System</b>
9. klik tombol 'Hapus'	
	10. menampilkan alert 'Apakah Anda yakin akan menghapus data ini?'
11. klik tombol 'Ok'	
	12. menghapus data dalam database
	13. menampilkan halaman 'Daftar Guru' yang telah diperbaharui

Tabel 4.4 Skenario manajemen data soal

ID	: UCS-06
Name	: Manajemen data soal
<i>Participating Aktor</i>	: Guru
<i>Entry Condition</i>	: Guru login ke sistem
<i>Exit Condition</i>	: Guru sukses meng- <i>input</i> -kan soal

<b>SKENARIO NORMAL (LIHAT)</b>	
<b>Actor</b>	<b>System</b>
1. Klik menu 'Master Data Soal'	
2. klik submenu 'Data Soal'	
	3. menampilkan halaman 'Tabel Data Soal' dengan tombol 'Tambah Data', 'Edit', 'Delete'
<b>SKENARIO NORMAL (TAMBAH)</b>	
<b>Actor</b>	<b>System</b>
4. klik tombol 'Tambah'	
	5. Menampilkan 'Form Tambah Siswa'

6. Mengisi 'form data soal'	
7. Klik tombol 'simpan'	
	8. Cek <i>input</i> -an data
	9. Menyimpan data ke dalam database
	10. Menampilkan halaman 'Data soal' yang telah diperbaharui.

#### SKENARIO ALTERNATIF

<i>Actor</i>	<i>System</i>
7. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input</i> -annya salah	
	8. Menampilkan warning 'masih ada data yang kosong/salah'

#### SKENARIO NORMAL (UBAH)

4. Klik tombol 'ubah'	
	5. Menampilkan halaman 'form data soal'
6. Mengubah 'form data soal'	
7. klik tombol 'simpan'	
	8. Cek <i>input</i> -an data
	9. Memperbaharui data di database
	10. menampilkan halaman 'Data Soal' yang telah diperbaharui

#### SKENARIO ALTERNATIF

<i>Actor</i>	<i>System</i>
7. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input</i> -annya salah	
	8. Menampilkan warning 'masih ada data yang kosong/salah'

#### SKENARIO NORMAL (HAPUS)

<i>Actor</i>	<i>System</i>
4. klik tombol 'Hapus'	
	5. menampilkan alert 'Apakah Anda yakin akan menghapus data ini?'
6. klik tombol 'Ok'	
	7. menghapus data dalam database
	8. menampilkan halaman 'Data Soal' yang telah diperbaharui

Tabel 4.5 Skenario sistem pengacakan soal

ID	: UCS-04
Name	: Sistem pengacakan soal
<i>Participating Aktor</i>	: Guru
<i>Entry Condition</i>	: Guru login ke sistem
<i>Exit Condition</i>	: Guru sukses mengacak soal

**SKENARIO NORMAL (LIHAT)**

<i>Actor</i>	<i>System</i>
1. Klik menu "Manajemen Latihan"	
2. Klik 'submenu "Pengacakan soal'	
	3. Menampilkan 'Tabel Paket Soal' dengan tombol 'Acak Soal', 'View', 'Delete'.

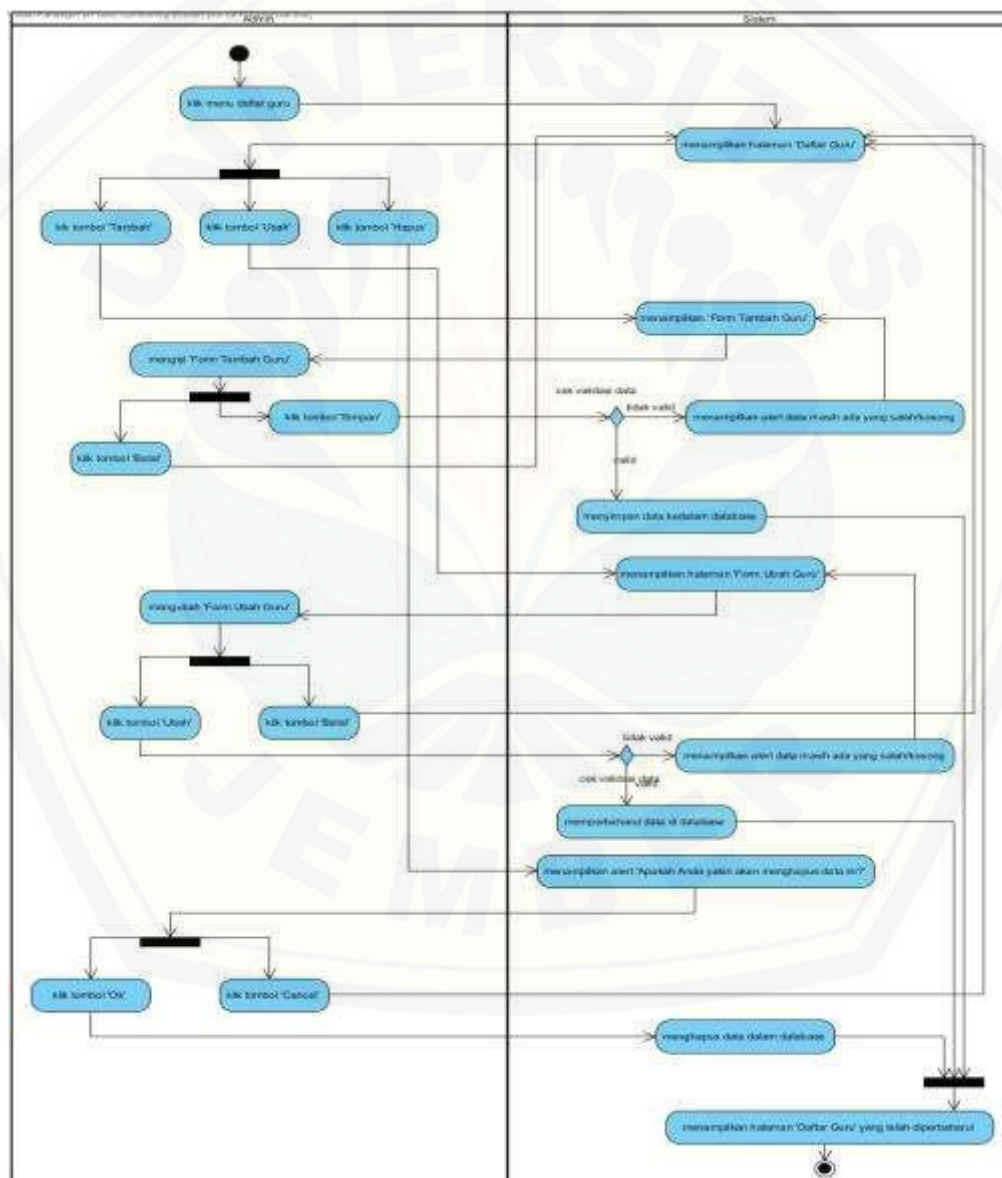
**SKENARIO NORMAL (ACAK SOAL)**

<i>Actor</i>	<i>System</i>
4. Kik tombol 'Acak Soal'	
	5. Menampilkan 'form pengacakan soal'
6. Mengisi 'form pengacakan soal'	
7. Klik tombol 'acak soal'	
	8. Simpan data acak dalam database sistem
	9. Menampilkan 'form hasil paket soal' yang telah diacak

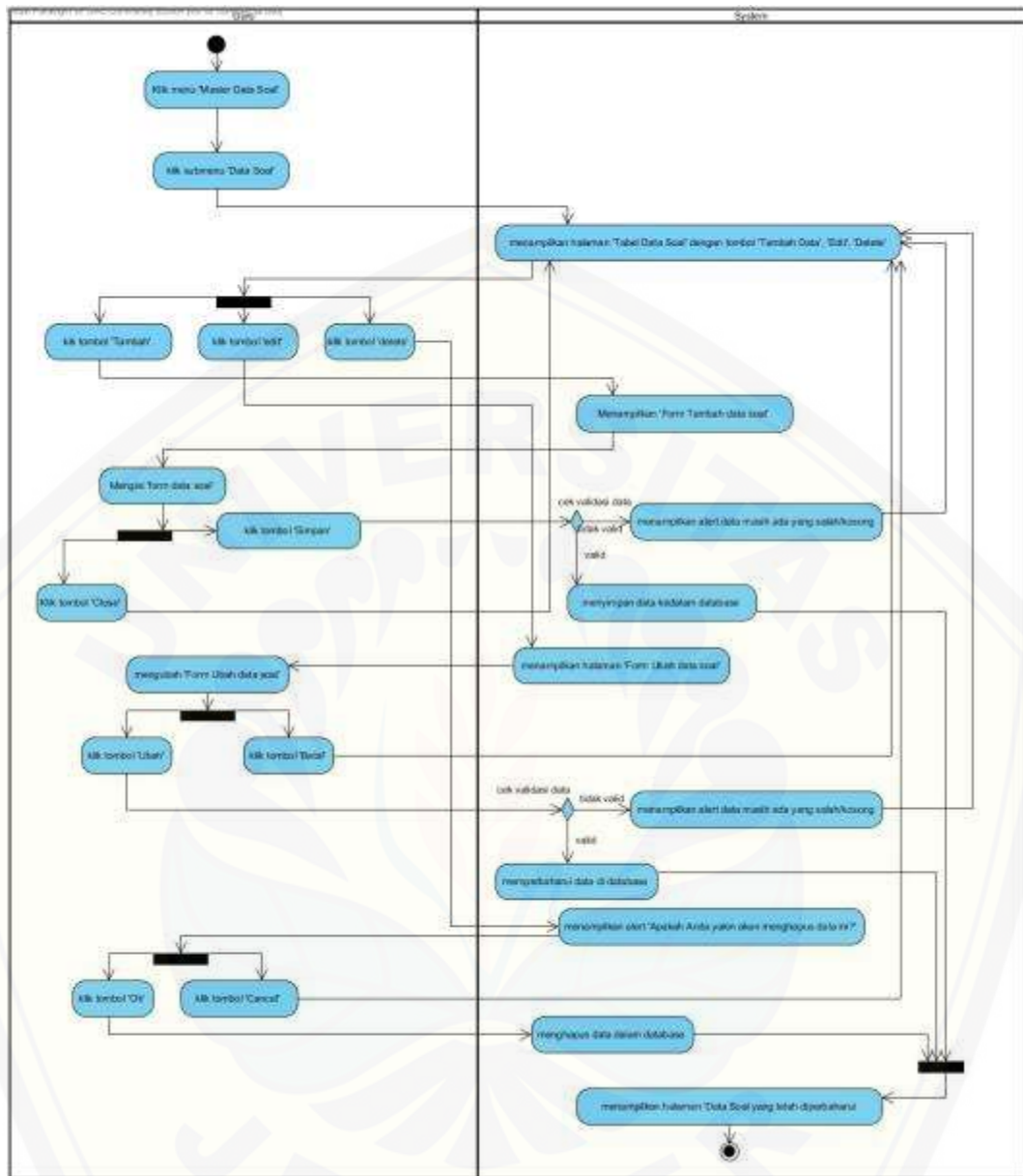
<b>SKENARIO ALTERNATIF</b>	
<i>Actor</i>	<i>System</i>
7. Ketika klik tombol 'acak soal' masih ada data yang kosong atau <i>input</i> -annya salah	
	8. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (VIEW)</b>	
4. Klik tombol 'View'	
	5. Menampilkan 'form hasil paket soal'
6. Klik tombol 'print'	
	7. Menyimpan data dalam bentuk PDF
<b>SKENARIO NORMAL (HAPUS)</b>	
<i>Actor</i>	<i>System</i>
3. Klik tombol 'Hapus'	
	4. Menampilkan alert 'Hapus data paket soal'
5. klik tombol 'Ok'	
	6. Menghapus data dalam database
	7. Menampilkan 'Tabel Paket Soal' dengan tombol 'Acak Soal', 'View', 'Delete'.

1. *Activity diagram*

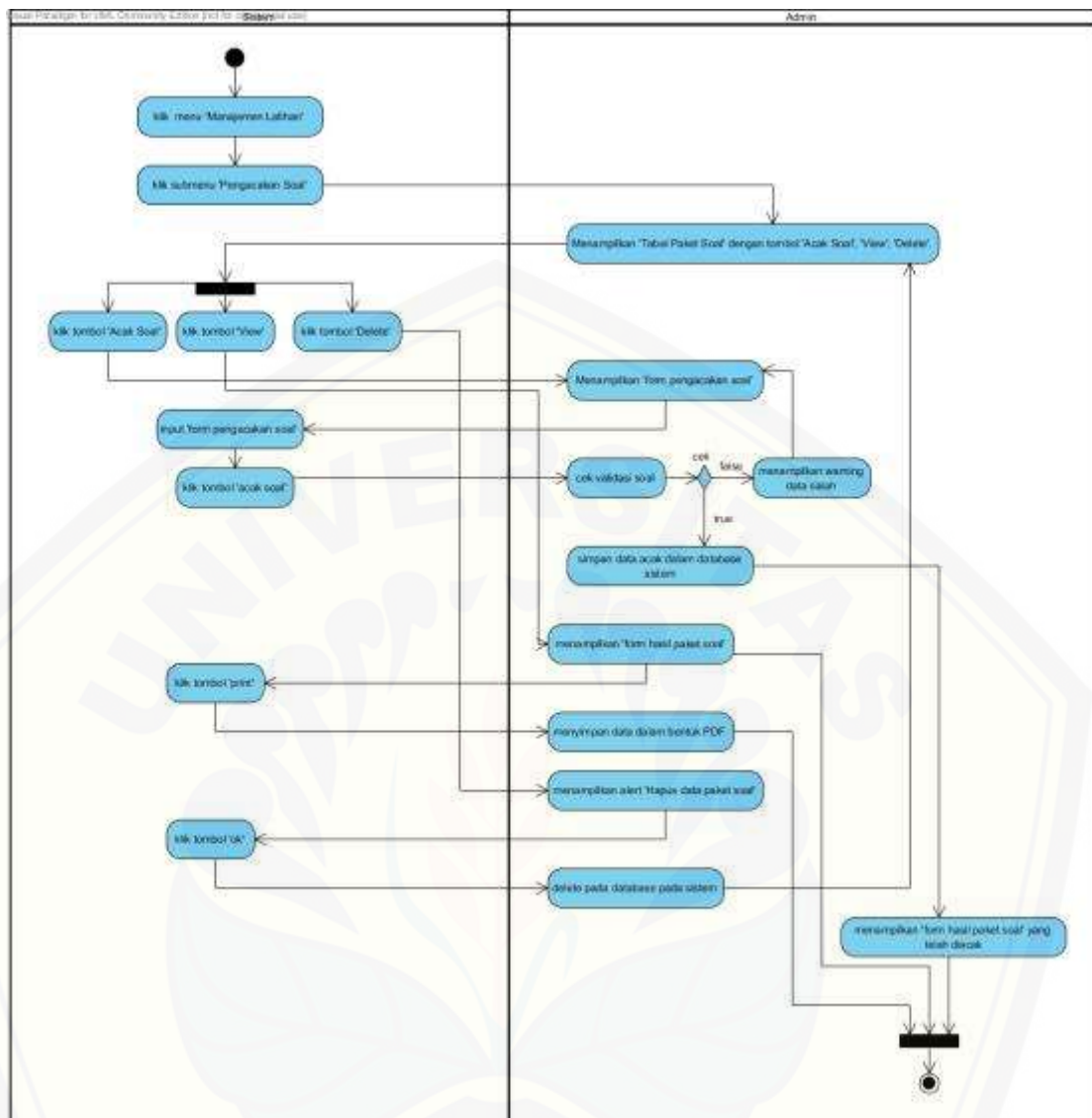
*Activity diagram* sistem pengacakan soal ujian nasional sma ini berfungsi untuk menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* manajemen data guru dapat dilihat pada Gambar 4.3, *Activity diagram* manajemen data soal pada Gambar 4.4, dan *Activity diagram* pengacakan soal pada Gambar 4.5:



Gambar 4.3 *Activity diagram* manajemen data guru



Gambar 4.4 Activity diagram manajemen data soal

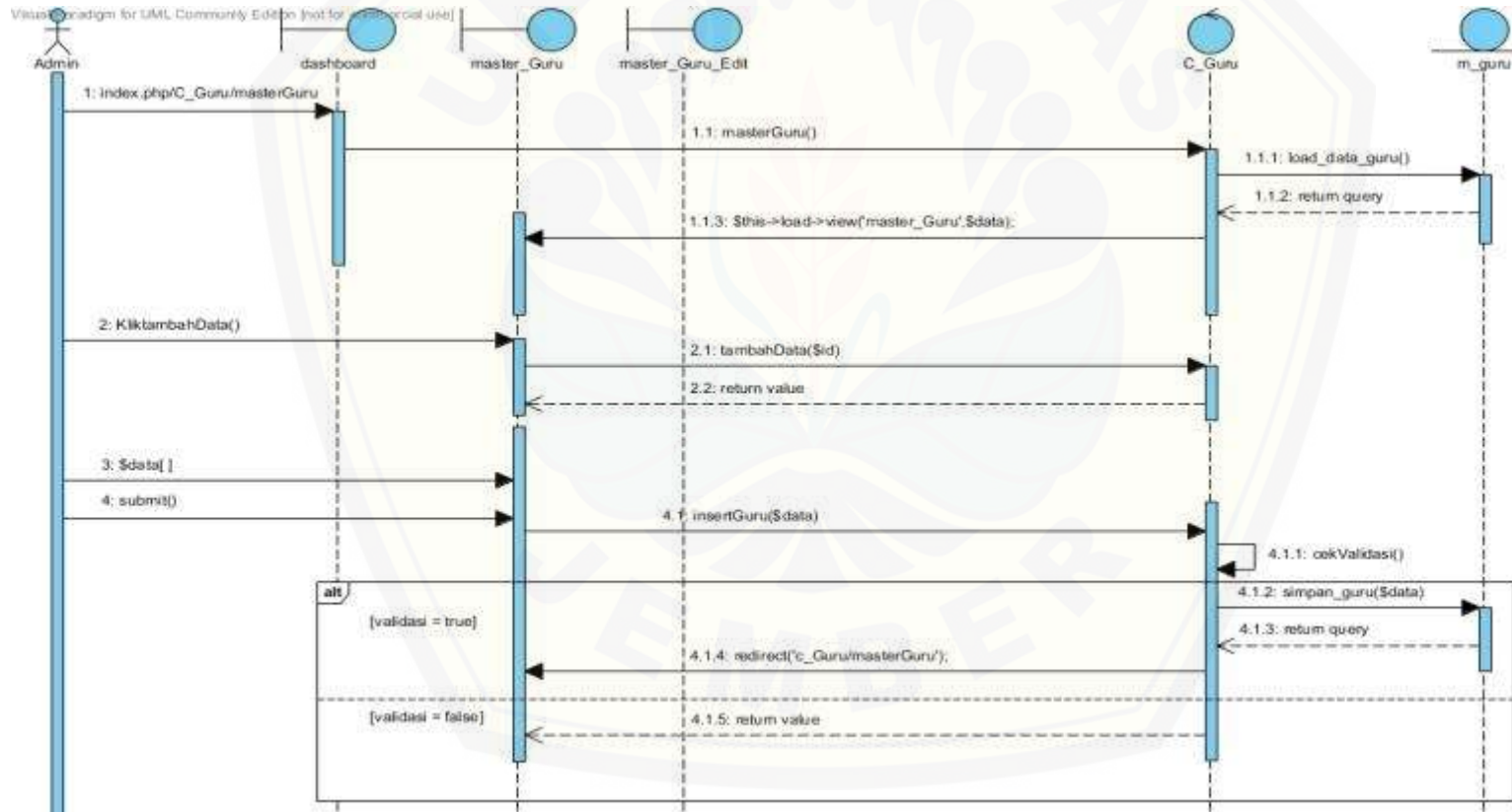


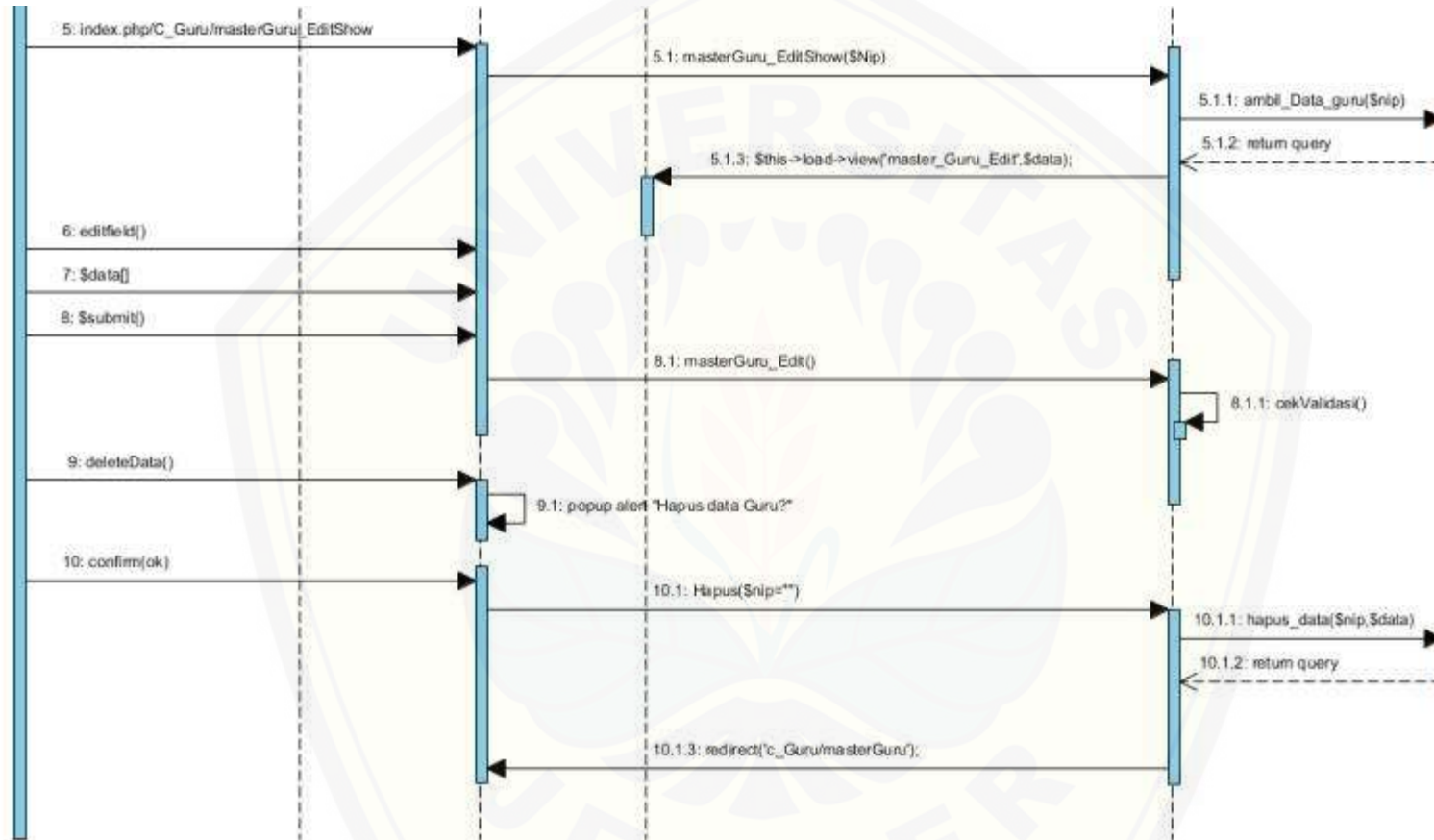
Gambar 4.5 Activity diagram pengacakan soal



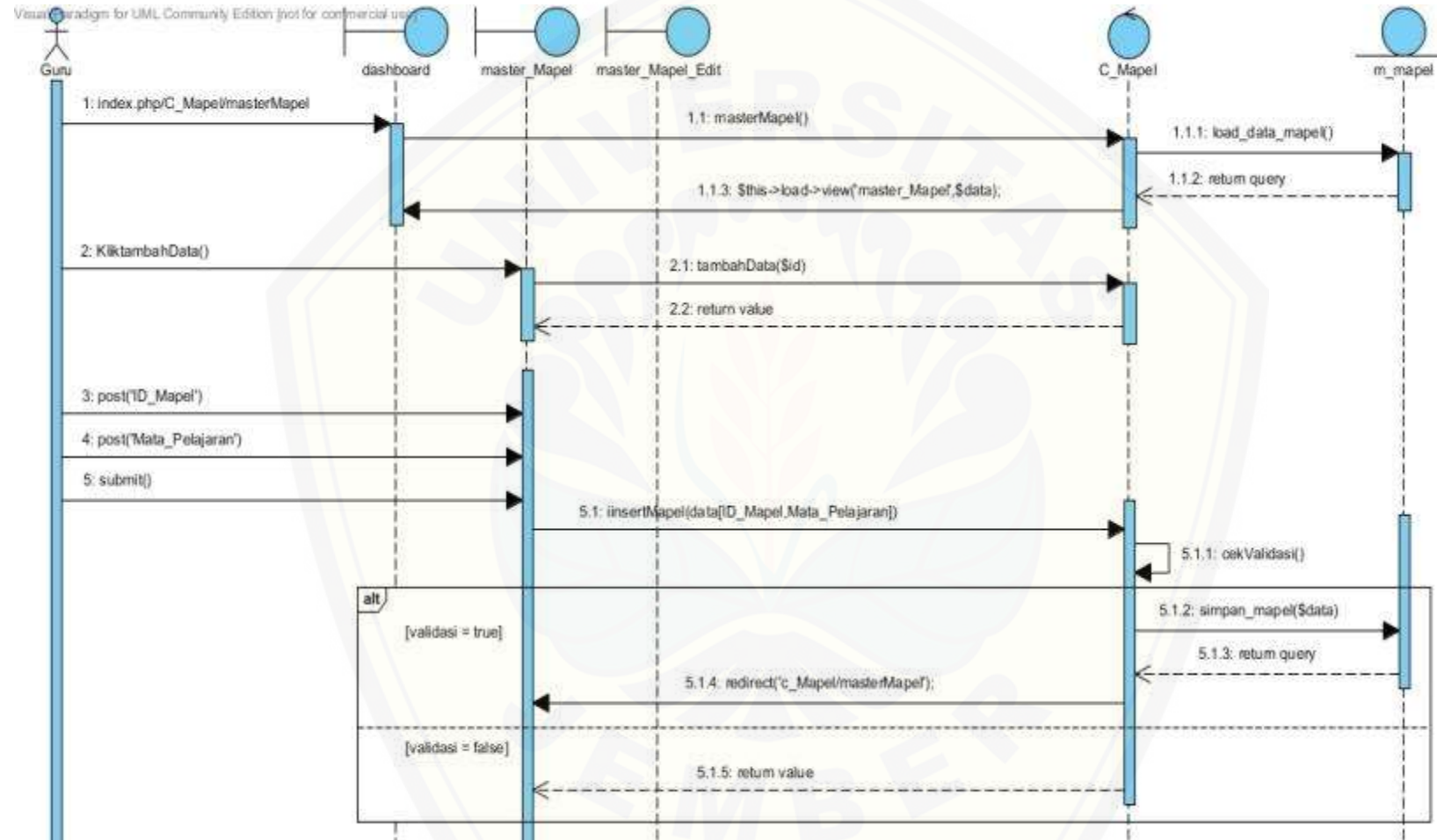
2. Sequence diagram

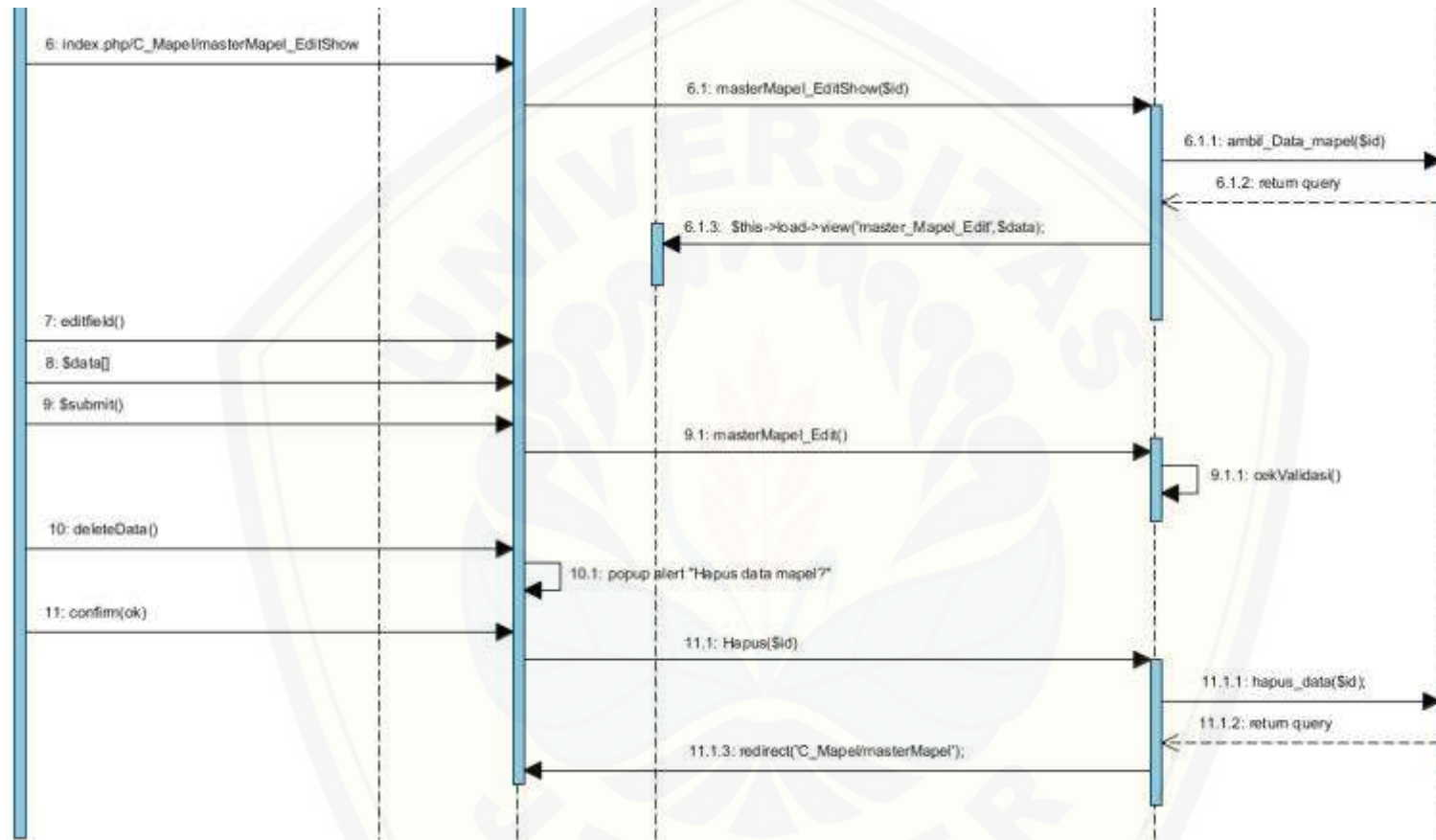
Sequence diagram pada sistem pengacakan soal ujian nasional sma ini digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan *output* tertentu. Sequence Diagram diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Sequence Diagram dari sistem seperti pada Gambar 4.6 Sequence diagram manajemen data guru, Sequence diagram manajemen data soal pada Gambar 4.7, dan Sequence diagram pengacakan soal pada Gambar 4.8.



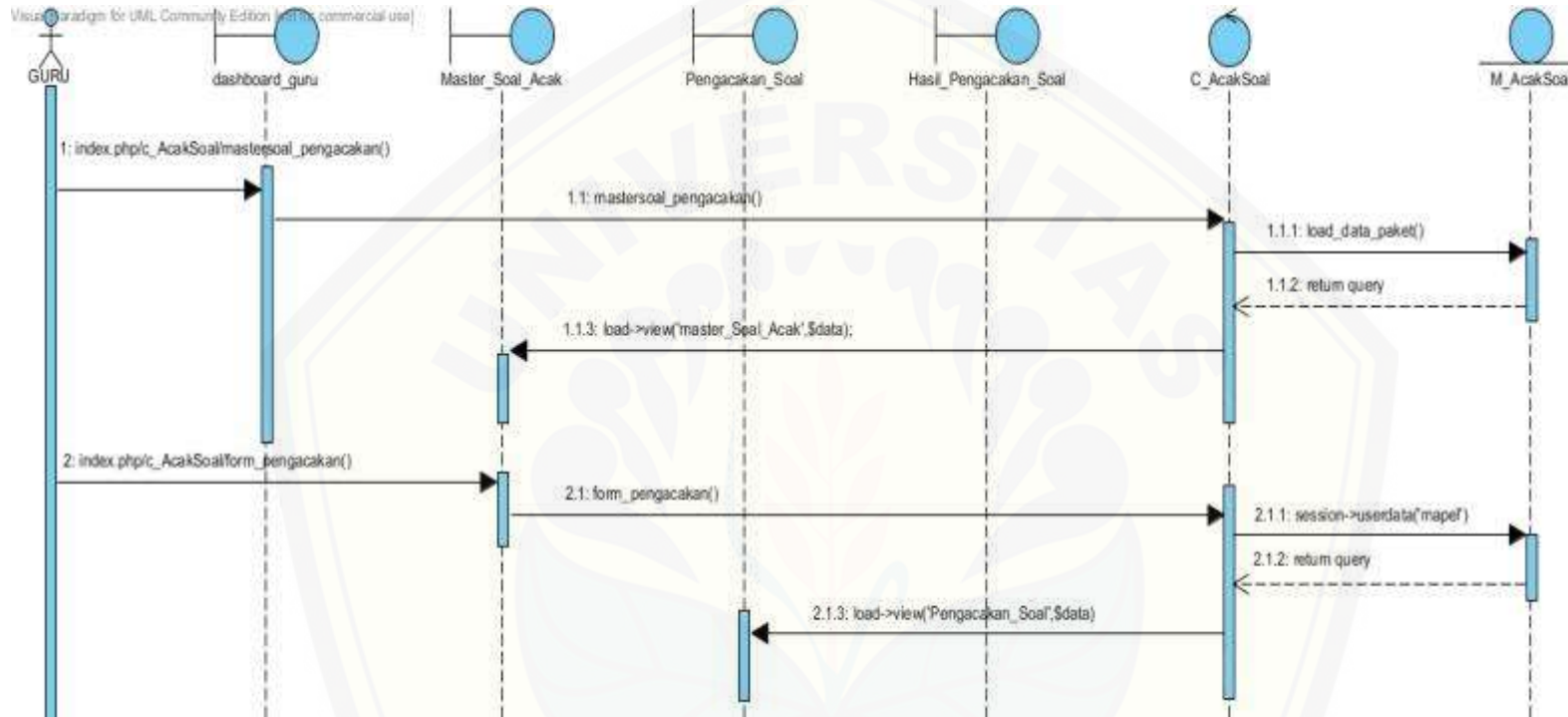


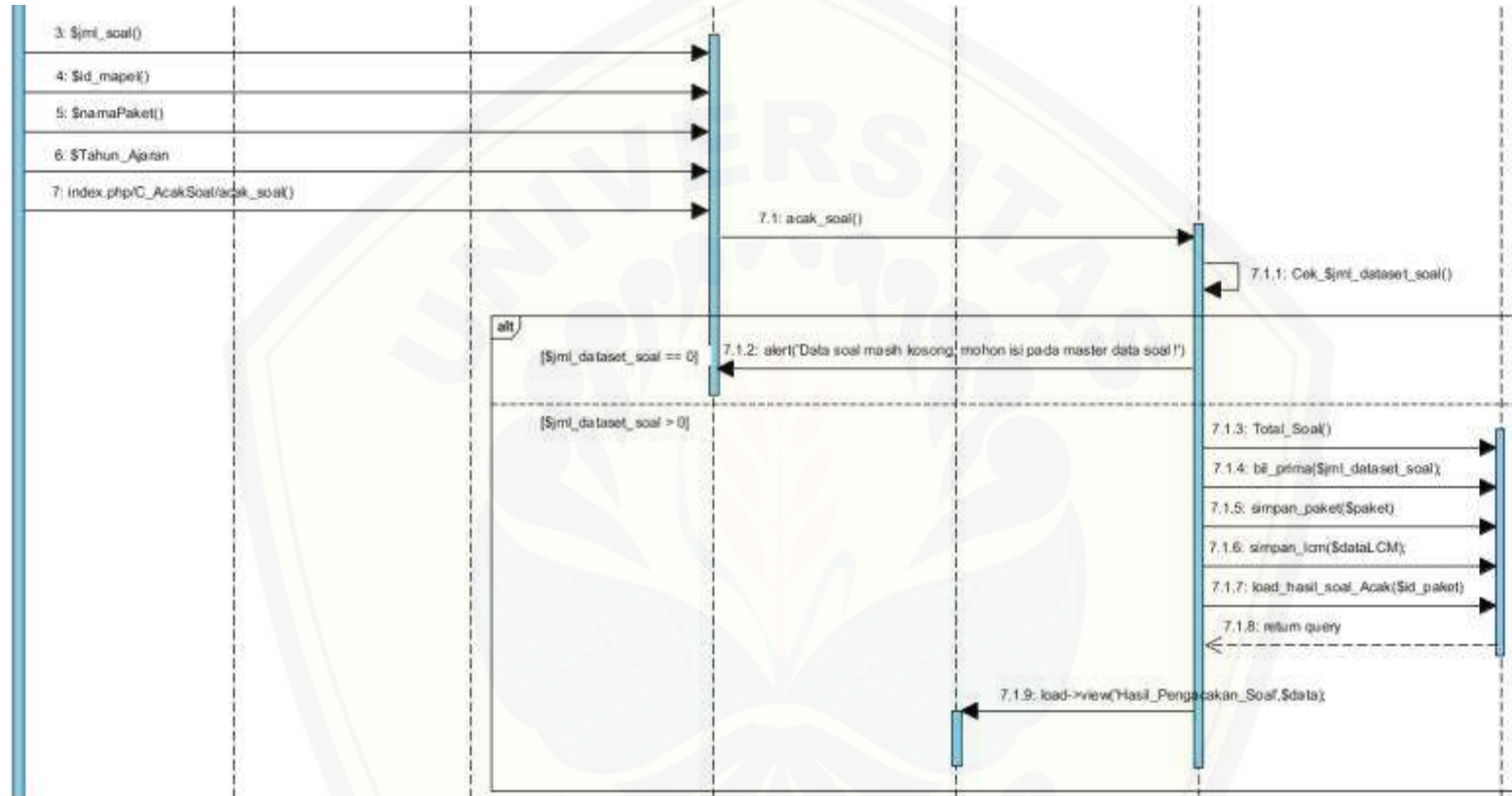
Gambar 4.6 Sequence diagram manajemen data guru





Gambar 4.7 Sequence diagram manajemen data soal



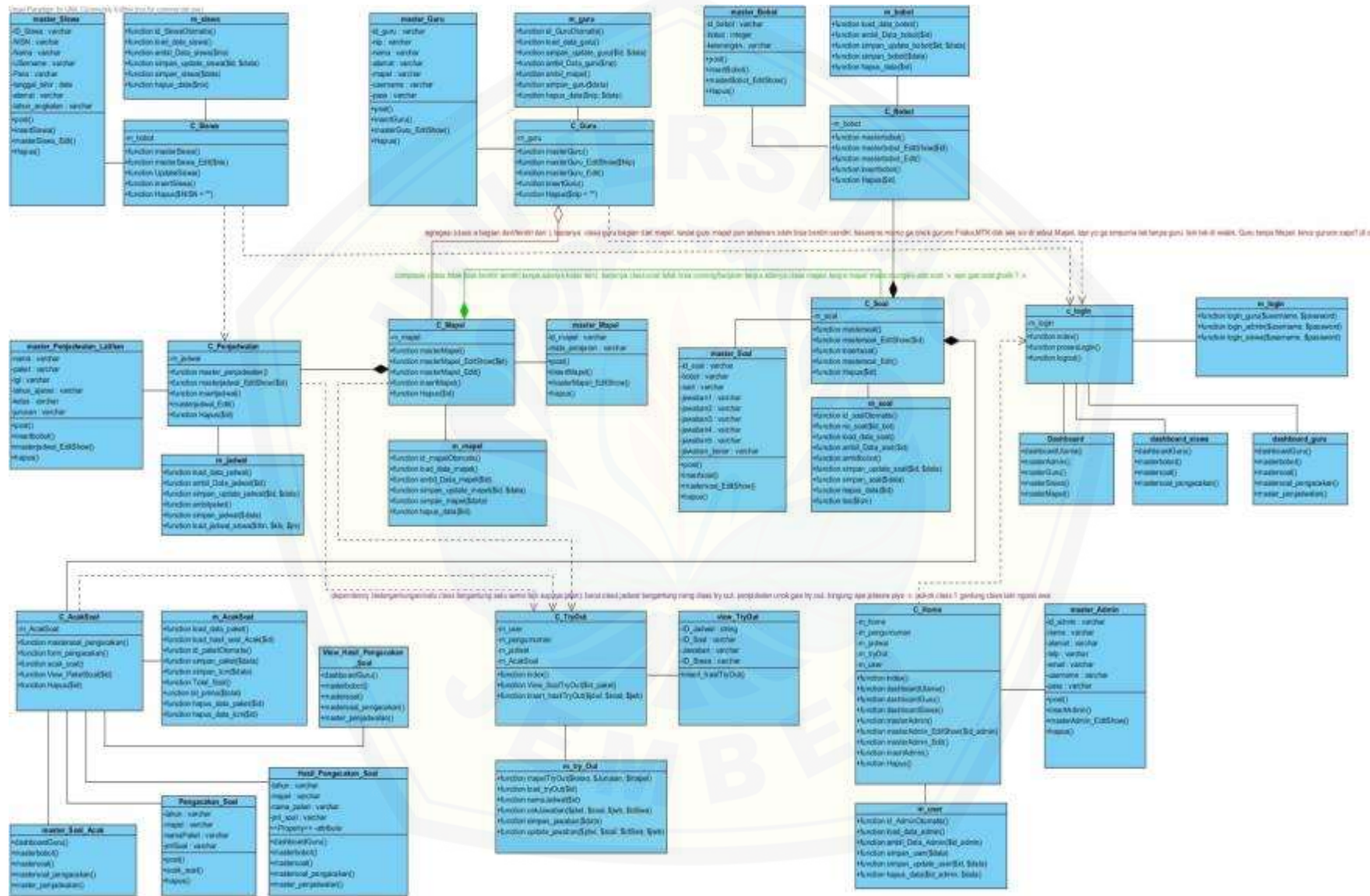


Gambar 4.8 Sequence diagram pengacakan soal

### 3. *Class diagram*

*Class diagram* menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti *dependency*, *asosiasi*, *agregasi*, dan lain-lain. Selain itu class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dirancang sehingga bagaimana caranya setiap class saling berkolaborasi untuk mencapai sebuah tujuan. *Class diagram* sistem dapat dilihat pada Gambar 4.9.



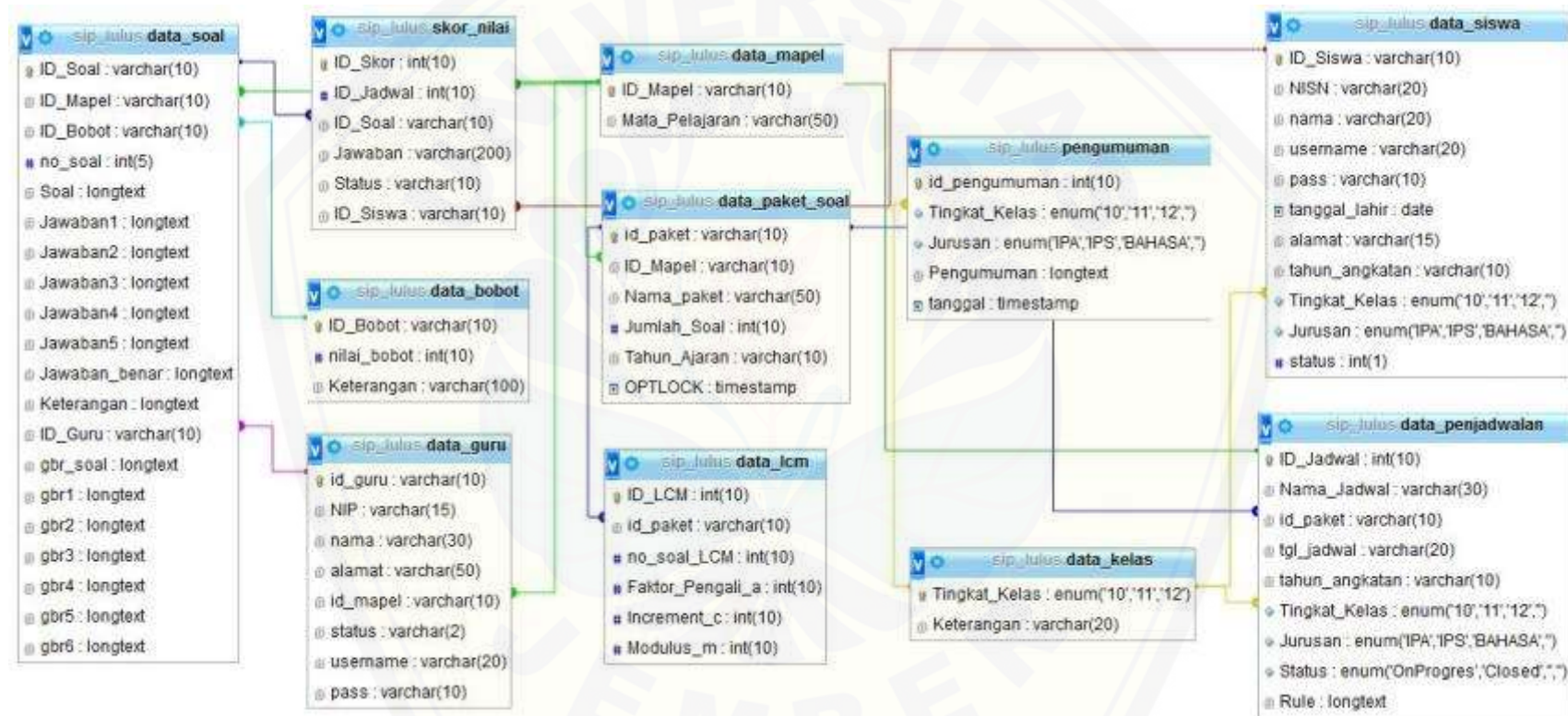


Gambar 4.9 Class diagram system pengacakan soal



#### 4. Entity Relationship Diagram ( ERD )

Entity Relationship Diagram ( ERD ) pada sistem pengacakan soal ujian nasional menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD aplikasi ditunjukkan pada Gambar 4.10.



Gambar 4.10 Entity Relationship Diagram ( ERD ) sistem pengacakan soal

**b. Penulisan Kode Program**

Setelah tahap desain perancangan selesai, tahap selanjutnya dalam penelitian ini yaitu tahap pengimplementasian desain perancangan ke dalam bahasa pemrograman. Desain sistem dari semua fitur sistem yang telah dibuat menggunakan pemodelan UML akan diimplementasikan kedalam kode program. Penulisan kode program menggunakan bahasa pemrograman *Java programming* menerapkan konsep teknik pemrograman OOP (*Object Oriented programming*) MVC dengan menggunakan *database My-SQL*.

**c. Pengujian Sistem**

Pada penelitian ini penulis menggunakan dua metode pengujian sistem yaitu *Black Box Testing* dan *White Box Testing*. Berikut adalah hasil pengujian sistem :

**1. White Box Testing**

Pengujian *white box testing* terdiri dari *listing* program, diagram alir, *cyclomatic complexity*, jalur program independen dan *test case*. Pada tahap ini fitur yang diuji adalah sebagai berikut:

**1. Pengujian White Box Testing CRUD manajemen data guru**

Pengujian *White Box Testing* CRUD data guru meliputi fitur tambah data, edit data, dan hapus data. Pengujian *White Box Testing* CRUD manajemen data guru adalah sebagai berikut :

## a. Listing program fitur manajemen guru

```
41     function masterGuru_Edit() {
42         $id['id_guru']     = $this->input->post('id_guru');
43         $data['NIP']       = $this->input->post('nip');
44         $data['nama']      = $this->input->post('nama');
45         $data['alamat']    = $this->input->post('alamat');
46         $data['id_mapel']  = $this->input->post('mapel');
47         $data['username']  = $this->input->post('username');
48         $data['pass']      = $this->input->post('pass');
49
50
51         $this->m_guru->simpan_update_guru($id,$data);
52         redirect('c_Guru/masterGuru');
53     }
```

Gambar 4.11 Kode Program Update Guru

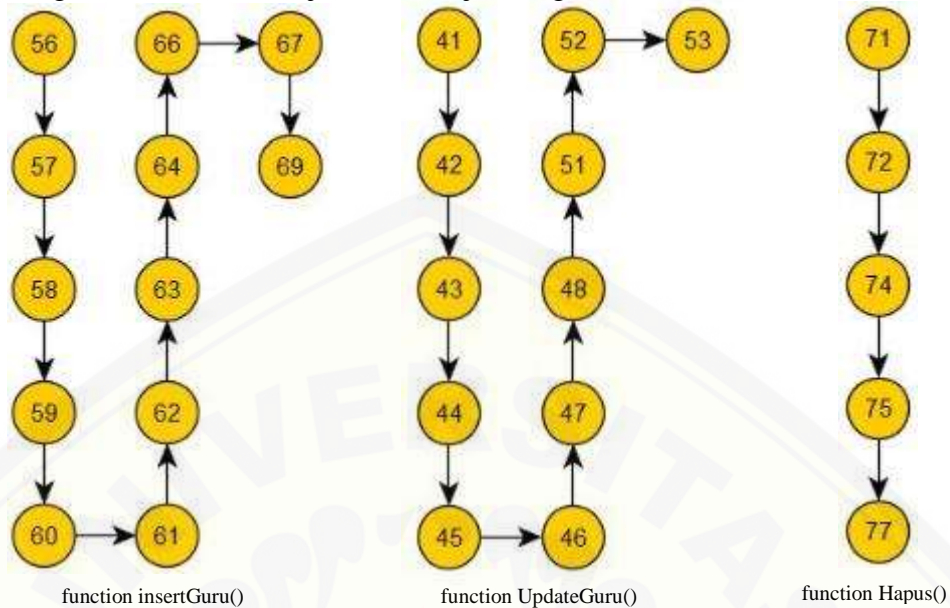
```
56     public function insertGuru(){
57         $data['id_guru'] = $this->input->post('id');
58         $data['NIP']     = $this->input->post('nip');
59         $data['Nama']    = $this->input->post('nama');
60         $data['Alamat']  = $this->input->post('alamat');
61         $data['ID_Mapel'] = $this->input->post('mapel');
62         $data['Username'] = $this->input->post('username');
63         $data['Pass']    = $this->input->post('pass');
64         $data['status']  = '0';
65
66         $this->m_guru->simpan_guru($data);
67         redirect('c_Guru/masterGuru');
68
69     }
```

Gambar 4.12 Kode Program Insert Guru

```
71     public function Hapus($nip=""){
72         $data['status'] = '1';
73
74         $this->m_guru->hapus_data($nip,$data);
75         redirect('C_Guru/masterGuru');
76
77     }
```

Gambar 4.13 Kode Program Hapus

## b. Diagram alir fitur manajemen manajemen guru



Gambar 4.14 Diagram alir fitur manajemen data guru

c. Perhitungan *Cyclomatic Complexity* fitur manajemen data guru

Perhitungan diagram alir pada manajemen data manajemen guru menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$\text{method function insertGuru() : } V(G) = E - N + 2 = 11 - 12 + 2 = 1$$

$$\text{method function UpdateGuru() : } V(G) = E - N + 2 = 10 - 11 + 2 = 1$$

$$\text{method function Hapus() : } V(G) = E - N + 2 = 4 - 5 + 2 = 1$$

## d. Pengujian jalur program fitur manajemen data guru

Pengujian jalur program fitur manajemen data guru berdasarkan diagram alir fitur manajemen kriteria :

$$\text{method function insertGuru() : jalur 1 : } 56 - 57 - 58 - 59 - 60 - 61 - 62 - 63 - 64 - 66 - 67 - 69$$

$$\text{method function UpdateGuru() : jalur 1 : } 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 51 - 52 - 53$$

$$\text{method function Hapus() : jalur 1 : } 71 - 72 - 74 - 75 - 77$$

e. *Test Case* fitur manajemen data guruTabel 4.6 *Test Case* fitur manajemen data guru

<i>Test Case method</i> function Insert Guru ()	
Jalur 1	
<i>Test Case</i>	Jika Tambah Guru berhasil
Target yang diharapkan	Menyimpan data Guru ke <i>database</i>
Hasil pengujian	Benar
Path/Jalur	56 – 57 – 58 – 59 – 60 – 61 – 62 – 63 – 64 – 66 – 67 - 69
<i>Test Case method</i> function Update Guru ()	
Jalur 1	
<i>Test Case</i>	Jika ubah update guru berhasil
Target yang diharapkan	Mengubah data guru di <i>database</i>
Hasil pengujian	Benar
Path/Jalur	41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 51 – 52 – 53
<i>Test Case method</i> function Hapus ()	
Jalur 1	
<i>Test Case</i>	Jika hapus data guru berhasil
Target yang diharapkan	Menghapus data guru dari <i>database</i>
Hasil pengujian	Benar
Path/Jalur	71 – 72 – 74 – 75 – 77

## 2. Pengujian *White Box Testing* CRUD manajemen data Soal

### a. *Listing* program fitur manajemen soal

```

40     public function insertsoal(){
41
42
43         $data['ID_Soal']       = $this->input->post('id');
44         $data['ID_Mapel']     = $this->input->post('id_mapel');
45         $data['ID_Bobot']    = $this->input->post('bobot');
46         $data['no_soal']     = $this->m_soal->no_soal($data['ID_Mapel']);
47         $data['Soal']        = $this->input->post('Soal');
48         $data['Jawaban1']    = $this->input->post('Jawaban1');
49         $data['Jawaban2']    = $this->input->post('Jawaban2');
50         $data['Jawaban3']    = $this->input->post('Jawaban3');
51         $data['Jawaban4']    = $this->input->post('Jawaban4');
52         $data['Jawaban5']    = $this->input->post('Jawaban5');
53         $data['Jawaban_benar'] = $this->input->post('Jawaban_benar');
54         $data['Keterangan']  = $this->input->post('Keterangan');
55         $data['ID_Guru']     = $this->session->userdata('nip');
56
57
58
59         $this->m_soal->simpan_soal($data);
60         redirect('c_soal/mastersoal');
61
62     }

```

Gambar 4.15 Kode Program Update Soal

```

64     function mastersoal_Edit() {
65         $id = $this->input->post('id');
66         $data['ID_Mapel'] = $this->input->post('id_mapel');
67         $data['ID_Bobot'] = $this->input->post('bobot');
68         $data['no_soal'] = $this->m_soal->no_soal($data['ID_Mapel']);
69         $data['Soal'] = $this->input->post('Soal');
70         $data['Jawaban1'] = $this->input->post('Jawaban1');
71         $data['Jawaban2'] = $this->input->post('Jawaban2');
72         $data['Jawaban3'] = $this->input->post('Jawaban3');
73         $data['Jawaban4'] = $this->input->post('Jawaban4');
74         $data['Jawaban5'] = $this->input->post('Jawaban5');
75         $data['Jawaban_benar'] = $this->input->post('Jawaban_benar');
76         $data['Keterangan'] = $this->input->post('Keterangan');
77         $data['ID_Guru'] = $this->session->userdata('nip');
78
79         $this->m_soal->simpan_update_soal($id,$data);
80         redirect('c_soal/mastersoal');
81     }

```

Gambar 4.16 Kode Program Update Soal

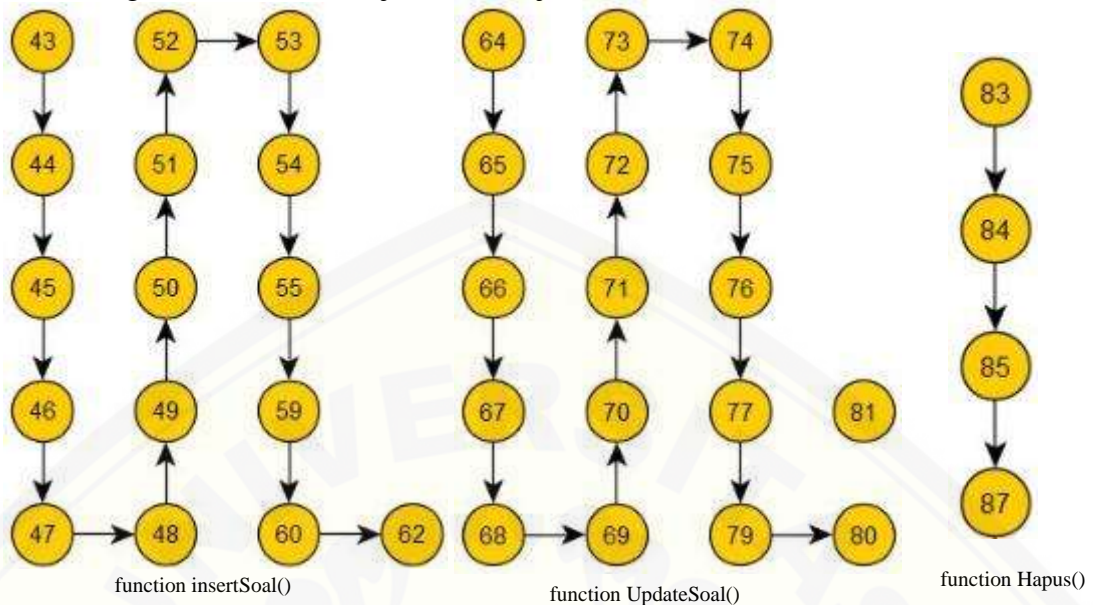
```

83     public function Hapus($id){
84         $this->m_soal->hapus_data($id);
85         redirect('C_soal/mastersoal');
86
87     }

```

Gambar 4.17 Kode Program Update Soal

## a. Diagram alir fitur manajemen manajemen soal



Gambar 4.18 Diagram alir fitur manajemen data soal

b. Perhitungan *Cyclomatic Complexity* fitur manajemen data soal

Perhitungan diagram alir pada manajemen data manajemen soal menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$\text{method function insertSoal() : } V(G) = E - N + 2 = 15 - 16 + 2 = 1$$

$$\text{method function UpdateSoal() : } V(G) = E - N + 2 = 16 - 17 + 2 = 1$$

$$\text{method function Hapus() : } V(G) = E - N + 2 = 3 - 4 + 2 = 1$$

## c. Pengujian jalur program fitur manajemen data soal

Pengujian jalur program fitur manajemen data soal berdasarkan diagram alir fitur manajemen kriteria :

$$\text{method function insertSoal() : jalur 1 : 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 55 - 59 - 60 - 62}$$

$$\text{method function UpdateSoal() : jalur 1 : 64 - 65 - 66 - 67 - 68 - 69 - 70 - 71 - 72 - 73 - 74 - 75 - 76 - 77 - 79 - 80 - 81}$$

$$\text{method function Hapus() : jalur 1 : 83 - 84 - 85 - 87}$$

d. *Test Case* fitur manajemen data soal

Tabel 4.7 *Test Case* fitur manajemen data soal

<i>Test Case method</i> function Insert Soal ()	
Jalur 1	
<i>Test Case</i>	Jika Tambah Soal berhasil
Target yang diharapkan	Menyimpan data soal ke <i>database</i>
Hasil pengujian	Benar
Path/Jalur	43 - 44 - 45 - 46 - 47 - 48- 49 - 50 -51 - 52 - 53 - 54 - 55 - 59 - 60 - 62
<i>Test Case method</i> function Update Soal ()	
Jalur 1	
<i>Test Case</i>	Jika ubah update soal berhasil
Target yang diharapkan	Mengubah data soal di <i>database</i>
Hasil pengujian	Benar
Path/Jalur	64 - 65 - 66 - 67 - 68 - 69 - 70 - 71 - 72 - 73 - 74 – 75 - 76- 77 - 79 - 80 - 81
<i>Test Case method</i> function Hapus ()	
Jalur 1	
<i>Test Case</i>	Jika hapus data soal berhasil
Target yang diharapkan	Menghapus data soal dari <i>database</i>
Hasil pengujian	Benar
Path/Jalur	83 - 84 - 85 - 87



### 3. Pengujian *White Box Testing* Acak Soal

#### a. Listing program function acak soal ()

```
41 function acak_soal(){
42
43     $id_paket      = $this->m_AcakSoal->id_paketOtomatis();
44     $jml_soal     = $this->input->post('jmlSoal');
45
46     $paket['id_paket'] = $id_paket;
47     $paket['ID_Mapel'] = $this->session->userdata('id_mapel');
48     $paket['Nama_paket'] = $this->input->post('namaPaket');
49     $paket['Jumlah_Soal'] = $jml_soal;
50     $paket['Tahun_Ajaran'] = $this->input->post('tahun');
51
52     //simpan ke data paket
53     $this->m_AcakSoal->simpan_paket($paket);
54
55     //Bitung hasil acak dengan LCM lalu simpan di database:
56     $jml_dataset_soal = $this->m_AcakSoal->Total_Soal();
57     // cari total soal pada dataset
58
59     if($jml_dataset_soal == 0)
60     // jika belum ada dataset soal / kosong
61     {
62         echo "<script>alert('Data soal masih kosong, mohon isi pada master data soal !');history.go(-1);</script>";
63     }
64     else{
65
66         $jml_dataset_soal = (int)$jml_dataset_soal;
67
68         $jml_soal_perpaket = $jml_soal;
69         $candidate_prima = $this->m_AcakSoal->bil_prima($jml_dataset_soal);
70         // di cari bilangan prima yang mendekati jumlah dataset
71         $m = $candidate_prima;
72         // m = bil.prima
73         $akar_m = sqrt($m);
74         // nilai akar m untuk salah satu syarat perhitungan LCM
75         $round_akar_m = ceil($akar_m);
76         // nilai akar m di bulatkan ke atas
77
78         $a = rand($round_akar_m-1,$m);
79         // faktor pengali (nilai a > akar m)
```

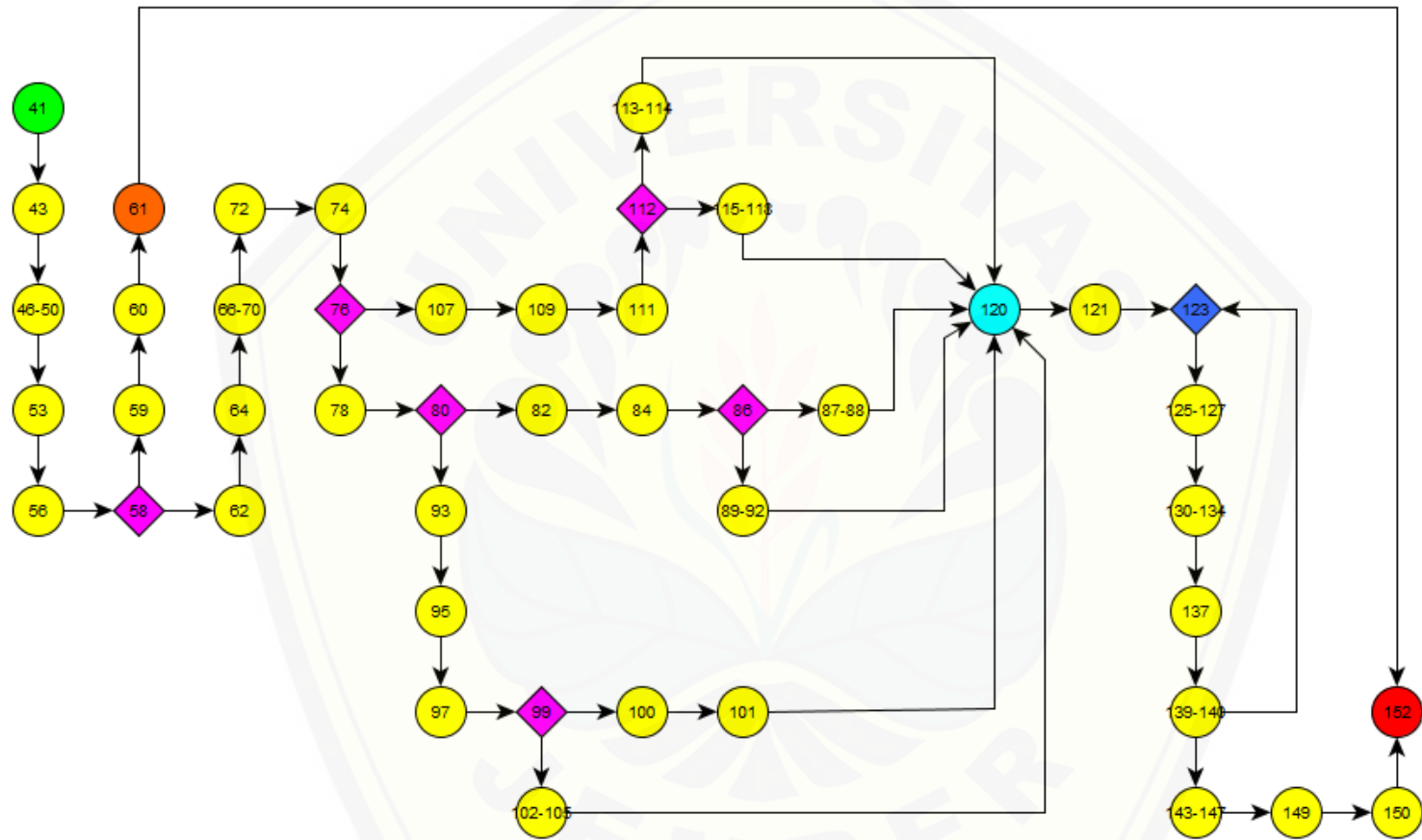
```
75 $round_akar_m = ceil($akar_m);
76 // nilai akar m di bulatkan ke atas
77
78 $a = rand($round_akar_m+1,$m);
79 // faktor pengali (nilai a > akar m)
80 $c = 0;
81 // nilai increment (c = ganjil, apabila m = bernilai pangkat 2,
82 // c tidak boleh nilai dari kelipatan m), untuk temp awal c = 0
83
84 if($akar_m = $round_akar_m){
85 // untuk mengecek apakah m bernilai pangkat dua,
86 // yaitu dengan apabila hasil akar m = bil.bulat
87
88 $c = rand(3,$m);
89 // nilai c di random mulai 3, (syarat c harus ganjil jika m bernilai pangkat 2)
90
91 if($c % 2 == 0){
92 // jika c mod 2 hasil sisa baginya = 0
93 $c-$c+1;
94 // c berarti genap, maka + 1, ( bil genap + 1 = bil. ganjil )
95
96 $cek_kelipatanC = fmod($m,$c);
97 // cek apakah c = kelipatan m, dengan mencari hasil sisa baginya m/c
98
99 if($cek_kelipatanC >= 1){
100 // cek jika sisa baginya != 0, maka c bukan kelipatan m, nilai c tetap
101 $c = $c;
102 }
103 else{
104 // jika sisa baginya di bawah 1 atau otomatis = 0
105 $c = $c+2;
106 // nilai c + 2, yang otomatis fmod(m,(c + 2)) = (>0)
107 }
108 }
109 else{
110 // sama seperti atas nilai c tetap karena sudah ganjil
111
112 $c-$c;
113
114 $cek_kelipatanC = fmod($m,$c);
```

```
115
116         if($cek_kelipatanC >= 1){
117             $c = $c;
118         }
119         else{
120             $c = $c+2;
121         }
122     }
123
124 }else{
125     // m tidak bernilai pangkat 2
126
127     $c = rand(2,$m);
128
129     $cek_kelipatanC = fmod($m,$c);
130     // cek kelipatan seperti sebelumnya
131     if($cek_kelipatanC >= 1){
132         $c = $c;
133     }
134     else{
135         $c = $c+1;
136     }
137 }
138
139 $z0 = 1;
140 //setting z0 awal = 1
141 $z = 0;
142 //setting z acak temp = 0
143
144 for($i = 1;$i <= $jml_soal_perpaket;$i++){
145     // hitung LCM sesuai Persamaan rumus LCM , dengan menggunakan
146     // algoritma perulangan hasil lcm dapat di hitung
147
148     $z = fmod(($a*$z0+$c),$m);
149     $zi = $z;
150     $z0 = $z;
```

```
151
152 //insert ke database
153 $dataLCM['id_paket'] = $id_paket;
154 $dataLCM['Faktor_Pengali_a'] = $a;
155 $dataLCM['Increment_c'] = $c;
156 $dataLCM['Modulus_m'] = $m;
157 $dataLCM['no_soal_LCM'] = $zi;
158
159 //simpan ke data_lcm
160 $this->m_AcakSoal->simpan_lcm($dataLCM);
161
162 }
163 }
164 // end hitung LCM
165
166 $data['tahun'] = $this->input->post('tahun');
167 $data['Jumlah_Soal'] = $jml_soal;
168 $data['Nama_paket'] = $this->input->post('namaPaket');
169 $data['mapel'] = $this->session->userdata('id_mapel');
170 $data['mapel'] = $this->session->userdata('mapel');
171
172 $data['Soal'] = $this->m_AcakSoal->load_hasil_soal_Acak($id_paket);
173 $this->load->view('Hasil_Pengacakan_Soal',$data);
174
175 }
```

Gambar 4.19 Kode Program function acak soal ()

b. Diagram alir acak soal



Gambar 4.20 Diagram alir function acak soal ()

c. Perhitungan *Cyclomatic Complexity* proses pengacakan soal

Perhitungan diagram alir pada proses pengacakan soal menggunakan *Cyclomatic Complexity* berdasarkan diagram alir method function acak soal () pada Gambar 4.20 adalah sebagai berikut:

*Method* function acak\_soal() :

$$V(G) = E - N + 2 = 52 - 46 + 2 = 8$$

d. Pengujian jalur program fitur pengacakan soal

Pengujian jalur program pengacakan berdasarkan diagram alir pengacakan pada Gambar 4.20 adalah sebagai berikut :

*Method* function acak\_soal() :

- Jalur 1 : 41 - 43 - [46-50] - 53 - 56 - [58-61] - 152
- Jalur 2 : 41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 107 - 109 - [111-14] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
- Jalur 3 : 41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 78 - 80 - 82 - 84 - 86 - [87-88] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
- jalur 4 : 41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 78 - 80 - 82 - 84 - 86 - [89-92] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
- jalur 5 : 41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 78 - 80 - 93 - 95 - 97 - [99-101] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152

- jalur 6 : 41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 78 - 80 - 93 - 95 - 97 - 99 - [102-105] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
- jalur 7 : 41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 107 - 109 - 111 - 112 - [115-118] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
- jalur 8 : 41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 107 - 109 - [111-114] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152

e. *Test Case* proses acak soal

Tabel 4.8 *Test Case* pengacakan soal

Test Case method function acak_soal()	
Jalur 1	
<i>Test Case</i>	Acak soal gagal ( <i>dataset</i> soal = 0 )
Target yang diharapkan	Menampilkan <i>alert</i> data masih kosong
Hasil pengujian	Benar
Path/Jalur	41 - 43 - [46-50] - 53 - 56 - [58-61] - 152
Jalur 2	
<i>Test Case</i>	Acak soal dengan nilai “m” tidak bernilai pangkat dua.
Target yang diharapkan	Soal akan diacak sesuai metode LCM dengan m $\neq$ pangkat dua
Hasil pengujian	Benar
Path/Jalur	41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 107 - 109 - [111-114] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
Jalur 3	
<i>Test Case</i>	Acak soal dengan nilai “m” tidak bernilai pangkat

	dua.
Target yang diharapkan	Soal akan diacak sesuai metode LCM dengan $m =$ pangkat dua
Hasil pengujian	Benar
Path/Jalur	41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 78 - 80 - 82 - 84 - 86 - [87-88] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
Jalur 4	
<i>Test Case</i>	Acak soal dengan nilai "c" bernilai kelipatan dari nilai "m".
Target yang diharapkan	Soal akan diacak sesuai metode LCM $c =$ kelipatan m
Hasil pengujian	Benar
Path/Jalur	41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 78 - 80 - 82 - 84 - 86 - [89-92] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
Jalur 5	
<i>Test Case</i>	Acak soal dengan nilai "c" = bilangan ganjil
Target yang diharapkan	Soal akan diacak sesuai metode LCM $c =$ bil ganjil
Hasil pengujian	Benar
Path/Jalur	41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 78 - 80 - 93 - 95 - 97 - [99-101] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
Jalur 6	
<i>Test Case</i>	Acak soal dengan nilai "c" = bilangan ganjil dan nilai "c" bukan kelipatan nilai "m"
Target yang diharapkan	Soal akan diacak sesuai metode LCM $c =$ bil ganjil dan $c \neq$ kelipatan m
Hasil pengujian	Benar
Path/Jalur	41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 78 - 80 - 93 - 95 - 97 - 99 - [102-105] - 120 - 121 - 123 - [125-127] - [130-134] -



	137 - [139-140] - [143-147] - 149 - 150 - 152
<b>Jalur 7</b>	
<i>Test Case</i>	Acak soal dengan nilai “c” = bilangan ganjil dan nilai “c” kelipatan nilai “m”
Target yang diharapkan	Soal akan diacak sesuai metode LCM c = bil ganjil dan c = kelipatan m
Hasil pengujian	Benar
Path/Jalur	41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 107 - 109 - 111 - 112 - [115-118] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152
<b>Jalur 8</b>	
<i>Test Case</i>	Acak soal dengan dataset soal bernilai lebih besar dari 0
Target yang diharapkan	Soal akan diacak
Hasil pengujian	Benar
Path/Jalur	41 - 43 - [46-50] - 53 - 56 - 58 - 62 - 64 - [66-70] - 72 - 74 - 76 - 107 - 109 - [111-114] - 120 - 121 - 123 - [125-127] - [130-134] - 137 - [139-140] - [125-127] - [130-134] - 137 - [139-140] - [143-147] - 149 - 150 - 152

## 2. *Black Box Testing*

Pengujian *black box* menitik beratkan pada fungsionalitas sistem. Pengujian ini tidak melihat kinerja internal dari sistem, jadi hanya berfokus pada kinerja sistem sesuai dengan spesifikasi dan kebutuhan yang dianalisis pada bab perancangan. Hasil pengujian *black box* dapat dilihat pada Tabel 4.9.

Tabel 4.9 Pengujian *Black Box* fitur manajemen data guru

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah Guru	<ul style="list-style-type: none"> <li>Ketika klik tombol tambah dan semua <i>field</i> sudah terisi dengan benar</li> <li>Ketika klik tombol tambah dan belum mengisi secara lengkap / <i>field</i> masih ada yang kosong</li> <li>Ketika klik tombol tambah dan mengisi <i>field</i> tidak sesuai <i>format</i> isian dari <i>field</i></li> </ul>	<ul style="list-style-type: none"> <li>Menambah data guru sesuai yang di-<i>input</i>-kan</li> <li>Menampilkan peringatan <i>error message</i></li> <li>Menampilkan peringatan <i>error message</i></li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>
2.	Ubah Data Guru	<ul style="list-style-type: none"> <li>Ketika klik tombol edit dan semua <i>field</i> sudah terisi dengan benar</li> <li>Ketika klik tombol edit dan belum mengisi secara lengkap / <i>field</i> masih ada yang kosong</li> <li>Ketika klik tombol edit dan mengisi <i>field</i> tidak sesuai <i>format</i> isian dari <i>field</i></li> <li>Ketika klik tombol edit dan belum memilih row data yang akan diedit</li> </ul>	<ul style="list-style-type: none"> <li><i>Update</i> data data guru waktu sesuai yang di-<i>input</i>-kan</li> <li>Menampilkan peringatan <i>error message</i></li> <li>Menampilkan peringatan <i>error message</i></li> <li>Menampilkan peringatan <i>error message</i></li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>

3.	Hapus Data Guru	<ul style="list-style-type: none"> <li>• Ketika klik tombol delete dan sudah memilih row data yang akan didelete</li> <li>• Ketika klik tombol delete dan belum memilih row data yang akan didelete</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Delete</i> data Guru sesuai yang dipilih [ <input checked="" type="checkbox"/> ] Berhasil</li> <li>• Menampilkan peringatan <i>error message</i> [ <input type="checkbox"/> ] Gagal</li> </ul>
----	-----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabel 4.9 merupakan hasil pengujian *Black Box Testing* pada fitur manajemen data soal. Hasil pengujian menunjukkan bahwa hasil yang sesuai dengan yang diharapkan oleh pengguna.

Tabel 4.10 Pengujian *Black Box* manajemen data soal

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah Data Soal	<ul style="list-style-type: none"> <li>• Ketika klik tombol tambah dan semua <i>field</i> sudah terisi dengan benar</li> <li>• Ketika klik tombol tambah dan belum mengisi secara lengkap / <i>field</i> masih ada yang kosong</li> <li>• Ketika klik tombol tambah dan mengisi <i>field</i> tidak sesuai <i>format</i> isian dari <i>field</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menambah data soal sesuai yang di-<i>input</i>-kan</li> <li>• Menampilkan peringatan <i>error message</i></li> <li>• Menampilkan peringatan <i>error message</i></li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>
2.	Ubah Data Soal	<ul style="list-style-type: none"> <li>• Ketika klik tombol edit dan semua <i>field</i> sudah terisi dengan benar</li> <li>• Ketika klik tombol edit dan belum mengisi secara lengkap / <i>field</i> masih ada yang kosong</li> <li>• Ketika klik tombol edit</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Update</i> data data soal waktu sesuai yang di-<i>input</i>-kan</li> <li>• Menampilkan peringatan <i>error message</i></li> <li>• Menampilkan peringatan <i>error</i></li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>

		dan mengisi <i>field</i> tidak sesuai <i>format</i> isian dari <i>field</i>	• Menampilkan peringatan <i>error message</i>
		• Ketika klik tombol edit dan belum memilih row data yang akan diedit	• <i>Delete</i> data Soal sesuai yang dipilih [ <input checked="" type="checkbox"/> ] Berhasil
3.	Hapus Data Soal	• Ketika klik tombol delete dan sudah memilih row data yang akan didelete	• Menampilkan peringatan <i>error message</i>
		• Ketika klik tombol delete dan belum memilih row data yang akan didelete	[ <input type="checkbox"/> ] Gagal

Tabel 4.10 merupakan hasil pengujian *Black Box Testing* pada fitur pengacakan soal. Hasil pengujian menunjukkan bahwa hasil yang sesuai dengan yang diharapkan oleh pengguna.

Tabel 4.11 Pengujian *Black Box* fitur pengacakan soal

No	Fitur	Kasus	Hasil	Keterangan
1.	Membuat soal acak	<ul style="list-style-type: none"> <li>• Ketika klik tombol acak soal</li> <li>• Ketika mengisi <i>form</i> pengacakan soal lalu klik tombol soal acak</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>form</i> pengacakan soal</li> <li>• Menampilkan <i>form</i> hasil paket soal yang telah di-acak</li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>
2.	View soal acak	<ul style="list-style-type: none"> <li>• Ketika klik tombol 'View'</li> <li>• Ketika klik tombol 'print'</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan 'form hasil paket soal</li> <li>• Menyimpan data dalam bentuk PDF</li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>
3.	Hapus	• Ketika klik tombol delete dan sudah	• <i>Delete</i> data sesuai yang dipilih	

---

memilih row data yang akan di delete

- Ketika klik tombol delete dan belum memilih row data yang akan di delete

- Menampilkan peringatan *error message* [  ] Berhasil [  ] Gagal



## BAB 6. PENUTUP

Pada bab ini merupakan bagian akhir di dalam penulisan skripsi, berisi tentang kesimpulan dan saran. Kesimpulan yang ditulis merupakan kesimpulan dari hasil penelitian yang telah dilakukan dan saran lanjutan untuk dilakukan pada penelitian selanjutnya.

### 6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Sistem pengacakan soal ujian nasional SMA dengan menggunakan metode *linear congruent* dibuat dengan menggunakan 3 hak akses yaitu user admin untuk memajemen data user, user guru dan user siswa. Dan user Guru memiliki beberapa fitur yaitu manajemen data soal, data bobot soal, data pengacakan soal dan manajemen data jadwal latihan. Terakhir adalah user siswa yang memiliki fitur view data soal yang nantinya digunakan untuk mengerjakan soal latihan.
2. Sistem pengacakan soal ujian nasional SMA dibangun dengan menerapkan metode *linear congruet*. Penerapan metode ini terdapat pada pengacakan soal yang memungkinkan untuk tidak adanya data yang terpanggil kembali sebelum data dalam database dihabiskan dulu. Pada percobaan dalam penelitian, dilakukan simulasi pada data sampel data tanggal 06-10-2015 pada data soal fisika di SMA Negeri 2 Jember pada rentang tahun 2010 sampai dengan 2015. Modulus ( $m$ ) bernilai bilangan prima, pada contoh kasus nilai ( $m$ ) = 439, maka  $m$  memenuhi syarat dan selanjutnya menentukan faktor pengalinya, Faktor pengali ( $a$ ) bernilai lebih besar dari  $\sqrt{m}$ , pada contoh kasus nilai ( $a$ ) = 250,  $\sqrt{m} = 439 = 20.9$ , maka  $m$  memenuhi syarat dan selanjutnya menentukan nilai increment, Incrment ( $c$ ) harus berangka ganjil apabila  $m$  bernilai pangkat dua, pada contoh kasus nilai ( $c$ ) = 43, dan ( $m$ ) = 439, maka ( $c$ ) memenuhi syarat. Nilai ( $c$ ) tidak boleh kelipatan dari ( $m$ ), ( $c$ ) = 43 dan ( $m$ ) = 439, maka nilai ( $c$ ) memenuhi syarat.

3. Pengacakan soal akan lebih efisien apabila soal dalam *database* diperbanyak, dan semakin mempekecil kemungkinan soal yang akan terulang kembali pada paket soal yang lain.

## 6.2 Saran

Beberapa saran dan masukan berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu :

1. Melakukan penelitian simulasi dengan soal yang lebih variatif.
2. Melakukan penelitian simulasi dengan menambah jumlah mata pelajaran yang di ujiakan sesuai dengan standar ujian nasional.
3. *Input*-an jumlah soal agar lebih banyak agar kemungkinan soal terulang kembali akan semakin kecil bahkan tidak ada. Serta proporsi dari pembobotan soal akan lebih merata.

**DAFTAR PUSTAKA**

- Agissa, W. (2013). White Box and Black Box Testing. <http://bangwildan.web.id/berita-176-white-box-testing--black-box-testing.html>.
- Andriasnyah. (2014, Maret ). Perancangan Aplikasi Game Edukasi Menggunakan Metode Linear Congruent Methode (LCM). VI.
- Danielson, C. (1997). A Collection of Performance Task and Rubrics.
- Mary Evanchalin .S, A. .. (2014, March). Linear Congruential Generator for LUT-SR Architecture. *International Journal of Scientific Engineering and Research (IJSER)*.
- Nasution, S. D. (2013, Agustus). PENERAPAN METODE LINIER KONGRUEN DAN ALGORITMA VIGENÈRE CHIPER PADA APLIKASI SISTEM UJIAN BERBASIS LAN. 94.
- Ni Made Suriadi, N. D. (2013). Penerapan Metode Drill Untuk Meningkatkan Aktivitas. *e-Journal Program Pascasarjana Universitas Pendidikan Ganesha*, 3.
- Pressman, R. (2012). *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: ANDI OFFSET.
- Rijani, E. W. (2011). Implementasi Metode Latihan Berjenjang. *E-Jurnal Dinas Pendidikan Kota Surabaya*, 1, 2.
- Supardi, Y. (2010). *Semua bisa menjadi Programmer java basic programming*. Jakarta: Elex Media Komputindo.
- Suryani, E. (2006). *Pemodelan dan Simulasi*. Yogyakarta: Graha Ilmu.
- Tilaar., H. A. (2006). Standarisasi Pendidikan Nasional: Suatu Tinjauan Kritis. 109-110.



## LAMPIRAN A. PERANCANGAN SISTEM

A.1 Perancangan *Scenario*1. *Scenario* manajemen data siswa

Tabel 6.1 Skenario manajemen data siswa

ID	:	UCS-03
Name	:	Manajemen data siswa
<i>Participating Aktor</i>	:	Admin
<i>Entry Condition</i>	:	Admin login ke sistem
<i>Exit Condition</i>	:	Admin sukses menambahkan data, mengubah data atau menghapus data siswa.

**SKENARIO NORMAL (LIHAT)**

<i>Actor</i>	<i>System</i>
1. Klik menu data siswa	
	2. Menampilkan halaman 'Data Siswa' dengan tombol tambah, ubah, dan hapus.

**SKENARIO NORMAL (TAMBAH)**

<i>Actor</i>	<i>System</i>
3. Klik tombol tambah	
	4. Menampilkan 'Form Tambah Siswa'
5. Mengisi 'Form Tambah Siswa'	
6. Klik tombol 'simpan'	
	7. Cek <i>input-an</i> data
	8. Menyimpan data ke dalam database
	9. Menampilkan halaman 'Daftar Siswa' yang telah diperbaharui.

**SKENARIO ALTERNATIF**

<i>Actor</i>	<i>System</i>
6. Ketika klik tombol 'simpan' masih	

ada data yang kosong atau <i>input</i> -annya salah	
	7. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (UBAH)</b>	
10. Klik tombol 'ubah'	
	11. Menampilkan halaman 'Form Ubah Siswa'
12. Mengubah 'Form Ubah Siswa'	
13. klik tombol 'simpan'	
	14. Cek <i>input</i> -an data
	15. Memperbaharui data di database
	16. menampilkan halaman 'Daftar Siswa' yang telah diperbaharui
<b>SKENARIO ALTERNATIF</b>	
<b>Actor</b>	<b>System</b>
13. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input</i> -annya salah	
	14. Cek <i>input</i> -an data
	15. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (HAPUS)</b>	
<b>Actor</b>	<b>System</b>
17. klik tombol 'Hapus'	
	18. menampilkan alert 'Apakah Anda yakin akan menghapus data ini?'
19. klik tombol 'Ok'	
	20. menghapus data dalam database
	21. menampilkan halaman 'Daftar Siswa'

	yang telah diperbaharui
2. <i>Scenario</i> manajemen data user	

Tabel 6.2 Skenario manajemen data user

ID	:	UCS-02
Name	:	Manajemen data user
<i>Participating Aktor</i>	:	Admin
<i>Entry Condition</i>	:	Admin login ke sistem
<i>Exit Condition</i>	:	Admin sukses menambahkan data, mengubah data atau menghapus data user.

**SKENARIO NORMAL (LIHAT)**

<i>Actor</i>	<i>System</i>
1. Klik menu data user	
	2. Menampilkan halaman 'Data User' dengan tombol tambah, ubah, dan hapus.

**SKENARIO NORMAL ( TAMBAH)**

<i>Actor</i>	<i>System</i>
3. Klik tombol tambah	
	4. Menampilkan 'Form Tambah User'
5. Mengisi 'Form Tambah User'	
6. Klik tombol 'simpan'	
	7. Cek <i>input-an</i> data
	8. Menyimpan data ke dalam database
	9. Menampilkan halaman 'Data User' yang telah diperbaharui.

**SKENARIO ALTERNATIF**

<i>Actor</i>	<i>System</i>
6. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input-annya</i> salah	

	7. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (UBAH)</b>	
10. Klik tombol 'ubah'	
	11. Menampilkan halaman 'Form Ubah User'
12. Mengubah 'Form Ubah User'	
13. klik tombol 'simpan'	
	14. Cek <i>input</i> -an data
	15. Memperbaharui data di database
	16. menampilkan halaman 'Data User' yang telah diperbaharui
<b>SKENARIO ALTERNATIF</b>	
<i>Actor</i>	<i>System</i>
13. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input</i> -annya salah	
	14. Cek <i>input</i> -an data
	15. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (HAPUS)</b>	
<i>Actor</i>	<i>System</i>
17. klik tombol 'Hapus'	
	18. menampilkan alert 'Apakah Anda yakin akan menghapus data ini?'
19. klik tombol 'Ok'	
	20. menghapus data dalam database
	21. menampilkan halaman 'Data User' yang telah diperbaharui

3. *Scenario* manajemen data bobot soal

Tabel 6.3 Skenario manajemen data bobot soal

ID	:	UCS-05
Name	:	Manajemen data bobot soal
<i>Participating Aktor</i>	:	Guru
<i>Entry Condition</i>	:	Guru login ke sistem
<i>Exit Condition</i>	:	Guru sukses menambahkan data, mengubah data atau menghapus data bobot soal.
<b>SKENARIO NORMAL (LIHAT)</b>		
<i>Actor</i>		<i>System</i>
1. Klik menu data bobot soal		
		2. Menampilkan halaman 'Data Bobot Soal' dengan tombol tambah, ubah, dan hapus.
<b>SKENARIO NORMAL (TAMBAH)</b>		
<i>Actor</i>		<i>System</i>
3. Klik tombol tambah		
		4. Menampilkan 'Form Tambah Bobot Soal'
5. Mengisi 'Form Tambah Bobot Soal'		
6. Klik tombol 'simpan'		
		7. Cek <i>input-an</i> data
		8. Menyimpan data ke dalam database
		9. Menampilkan halaman 'Data Bobot Soal' yang telah diperbaharui.
<b>SKENARIO ALTERNATIF</b>		
<i>Actor</i>		<i>System</i>
6. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input-</i>		

annya salah	
	7. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (UBAH)</b>	
10. Klik tombol 'ubah'	
	11. Menampilkan halaman 'Form Ubah Bobot Soal'
12. Mengubah 'Form Ubah Bobot Soal'	
13. klik tombol 'simpan'	
	14. Cek <i>input</i> -an data
	15. Memperbaharui data di database
	16. Menampilkan halaman 'Data Bobot Soal' yang telah diperbaharui
<b>SKENARIO ALTERNATIF</b>	
<b>Actor</b>	<b>System</b>
13. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input</i> -annya salah	
	14. Cek <i>input</i> -an data
	15. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (HAPUS)</b>	
<b>Actor</b>	<b>System</b>
17. klik tombol 'Hapus'	
	18. menampilkan alert 'Apakah Anda yakin akan menghapus data ini?'
19. klik tombol 'Ok'	
	20. menghapus data dalam database
	21. menampilkan halaman 'Data Bobot Soal' yang telah diperbaharui

4. *Scenario* manajemen data soal

Tabel 6.4 Skenario manajemen data soal

ID	:	UCS-05
Name	:	Manajemen data soal
<i>Participating Aktor</i>	:	Guru
<i>Entry Condition</i>	:	Guru login ke sistem
<i>Exit Condition</i>	:	Guru sukses menambahkan data, mengubah data atau menghapus data soal.

**SKENARIO NORMAL (LIHAT)**

<i>Actor</i>	<i>System</i>
1. Klik menu data soal	
	2. Menampilkan halaman 'Data Bobot Soal' dengan tombol tambah, ubah, dan hapus.

**SKENARIO NORMAL (TAMBAH)**

<i>Actor</i>	<i>System</i>
3. Klik tombol tambah	
	4. Menampilkan 'Form Tambah Soal'
5. Mengisi 'Form Tambah Soal'	
6. Klik tombol 'simpan'	
	7. Cek <i>input-an</i> data
	8. Menyimpan data ke dalam database
	9. Menampilkan halaman 'Data Soal' yang telah diperbaharui.

**SKENARIO ALTERNATIF**

<i>Actor</i>	<i>System</i>
6. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input-annya</i> salah	

	7. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (UBAH)</b>	
10. Klik tombol 'ubah'	
	11. Menampilkan halaman 'Form Ubah Soal'
12. Mengubah 'Form Ubah Soal'	
13. klik tombol 'simpan'	
	14. Cek <i>input</i> -an data
	15. Memperbaharui data di database
	16. Menampilkan halaman 'Data Soal' yang telah diperbaharui
<b>SKENARIO ALTERNATIF</b>	
<b>Actor</b>	<b>System</b>
13. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input</i> -annya salah	
	14. Cek <i>input</i> -an data
	15. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (HAPUS)</b>	
<b>Actor</b>	<b>System</b>
17. klik tombol 'Hapus'	
	18. menampilkan alert 'Apakah Anda yakin akan menghapus data ini?'
19. klik tombol 'Ok'	
	20. menghapus data dalam database
	21. menampilkan halaman 'Data Soal' yang telah diperbaharui



## 5. Scenario Latihan Soal

Tabel 6.5 Skenario latihan soal

ID	:	UCS-05
Name	:	Latihan soal
<i>Participating Aktor</i>	:	Siswa
<i>Entry Condition</i>	:	Siswa login ke sistem
<i>Exit Condition</i>	:	Siswa mengerjakan dan sukses mengerjakan latihan soal.

**SKENARIO NORMAL (LIHAT)**

<i>Actor</i>	<i>System</i>
1. Klik menu ujian try out	
	2. Menampilkan <i>modals</i> 'Mata pelajaran try out'

**SKENARIO NORMAL**

<i>Actor</i>	<i>System</i>
3. Klik tombol 'Fisika Paket-1'	
	4. Menampilkan modals mata pelajaran try out
	5. Menampilkan halaman 'Fisika Paket-1'
6. Mengisi halaman 'Fisika Paket-1'	
7. Klik tombol 'selesai'	
	8. Cek <i>input-an</i> data
	9. Menyimpan data ke dalam database
	10. Menampilkan halaman 'Hasil Try Out'.

**SKENARIO ALTERNATIF**

<i>Actor</i>	<i>System</i>
6. Ketika klik tombol 'selesai' masih ada data yang belum diisi	
	7. Menampilkan warning 'soal nomor '

	belum terisi’.
--	----------------

6. *Scenario* manajemen data jadwal latihan

Tabel 6.6 Skenario manajemen jadwal latihan

ID	:	UCS-10
Name	:	Manajemen jadwal latihan
<i>Participating Aktor</i>	:	Guru
<i>Entry Condition</i>	:	Guru login ke sistem
<i>Exit Condition</i>	:	Guru sukses menambahkan data, mengubah data atau menghapus jadwal latihan.

**SKENARIO NORMAL (LIHAT)**

<i>Actor</i>	<i>System</i>
1. Klik menu data jadwal latihan	
	2. Menampilkan halaman ‘Data Jadwal Latihan’ dengan tombol tambah, ubah, dan hapus.

**SKENARIO NORMAL (TAMBAH)**

<i>Actor</i>	<i>System</i>
3. Klik tombol tambah	
	4. Menampilkan ‘Form Tambah Jadwal Latihan’
5. Mengisi ‘Form Tambah Jadwal Latihan’	
6. Klik tombol ‘simpan’	
	7. Cek <i>input-an</i> data
	8. Menyimpan data ke dalam database
	9. Menampilkan halaman ‘Data Jadwal Latihan’ yang telah diperbaharui.

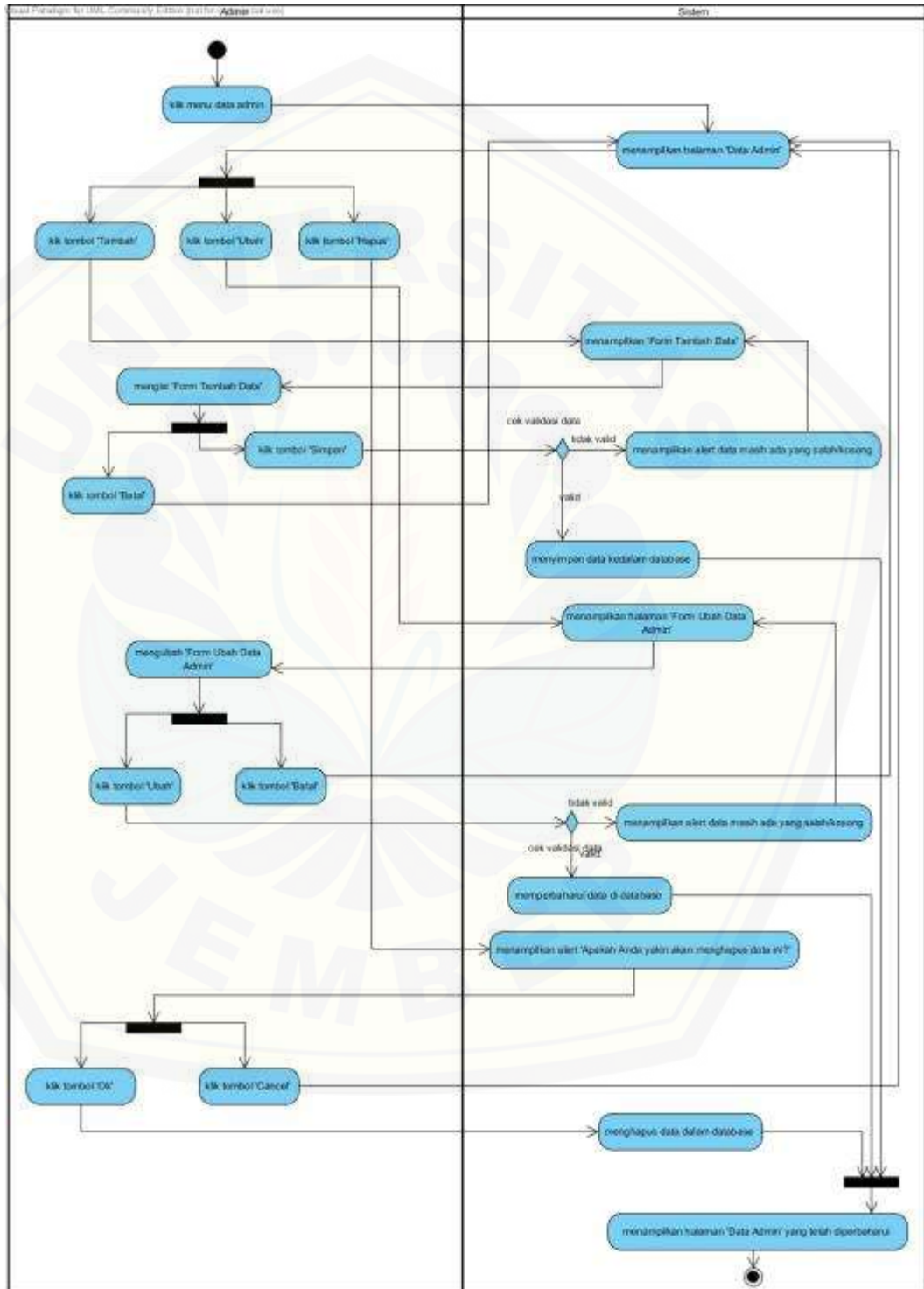
**SKENARIO ALTERNATIF**

<i>Actor</i>	<i>System</i>
6. Ketika klik tombol ‘simpan’ masih	

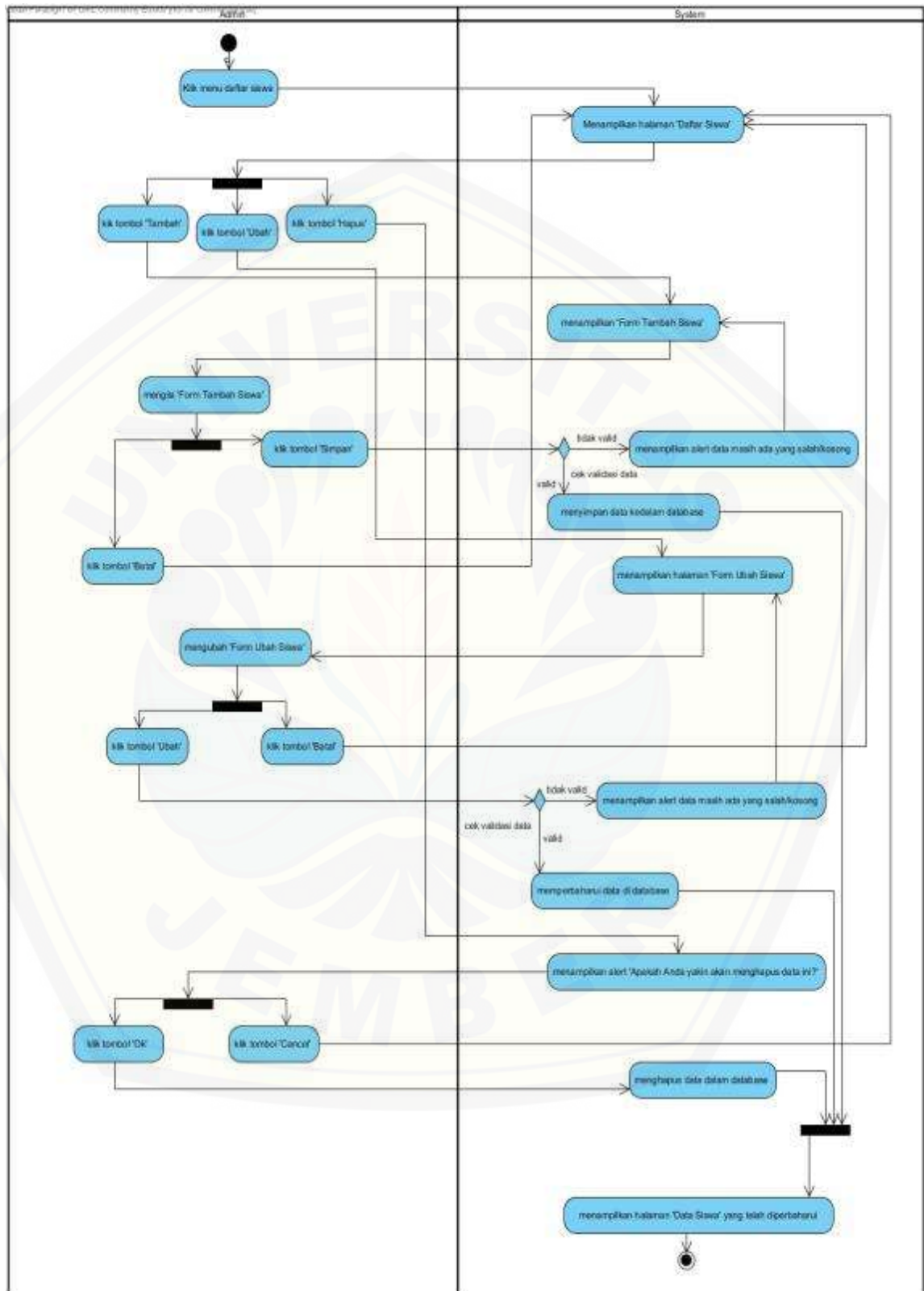
ada data yang kosong atau <i>input</i> -annya salah	
	7. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (UBAH)</b>	
10. Klik tombol 'ubah'	
	11. Menampilkan halaman 'Form Ubah Jadwal Latihan'
12. Mengubah 'Form Ubah Jadwal Latihan'	
13. klik tombol 'simpan'	
	14. Cek <i>input</i> -an data
	15. Memperbaharui data di database
	16. Menampilkan halaman 'Data Jadwal Latihan' yang telah diperbaharui
<b>SKENARIO ALTERNATIF</b>	
<b>Actor</b>	<b>System</b>
13. Ketika klik tombol 'simpan' masih ada data yang kosong atau <i>input</i> -annya salah	
	14. Menampilkan warning 'masih ada data yang kosong/salah'
<b>SKENARIO NORMAL (HAPUS)</b>	
<b>Actor</b>	<b>System</b>
17. klik tombol 'Hapus'	
	18. menampilkan alert 'Apakah Anda yakin akan menghapus data ini?'
19. klik tombol 'Ok'	
	20. menghapus data dalam database
	21. menampilkan halaman 'Data Jadwal

**A.2 Perancangan Activity Diagram**

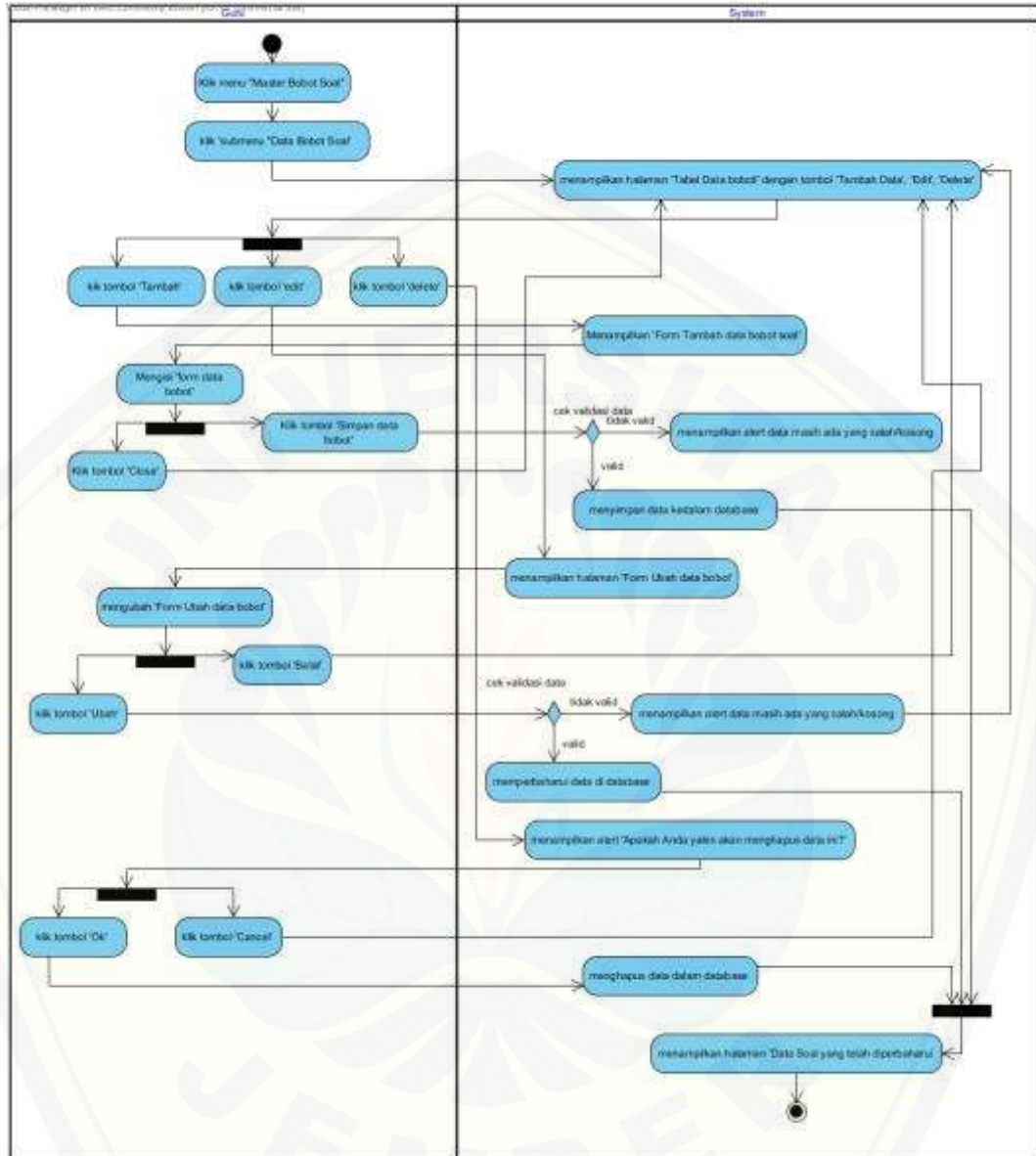
1. Activity diagram manajemen data user



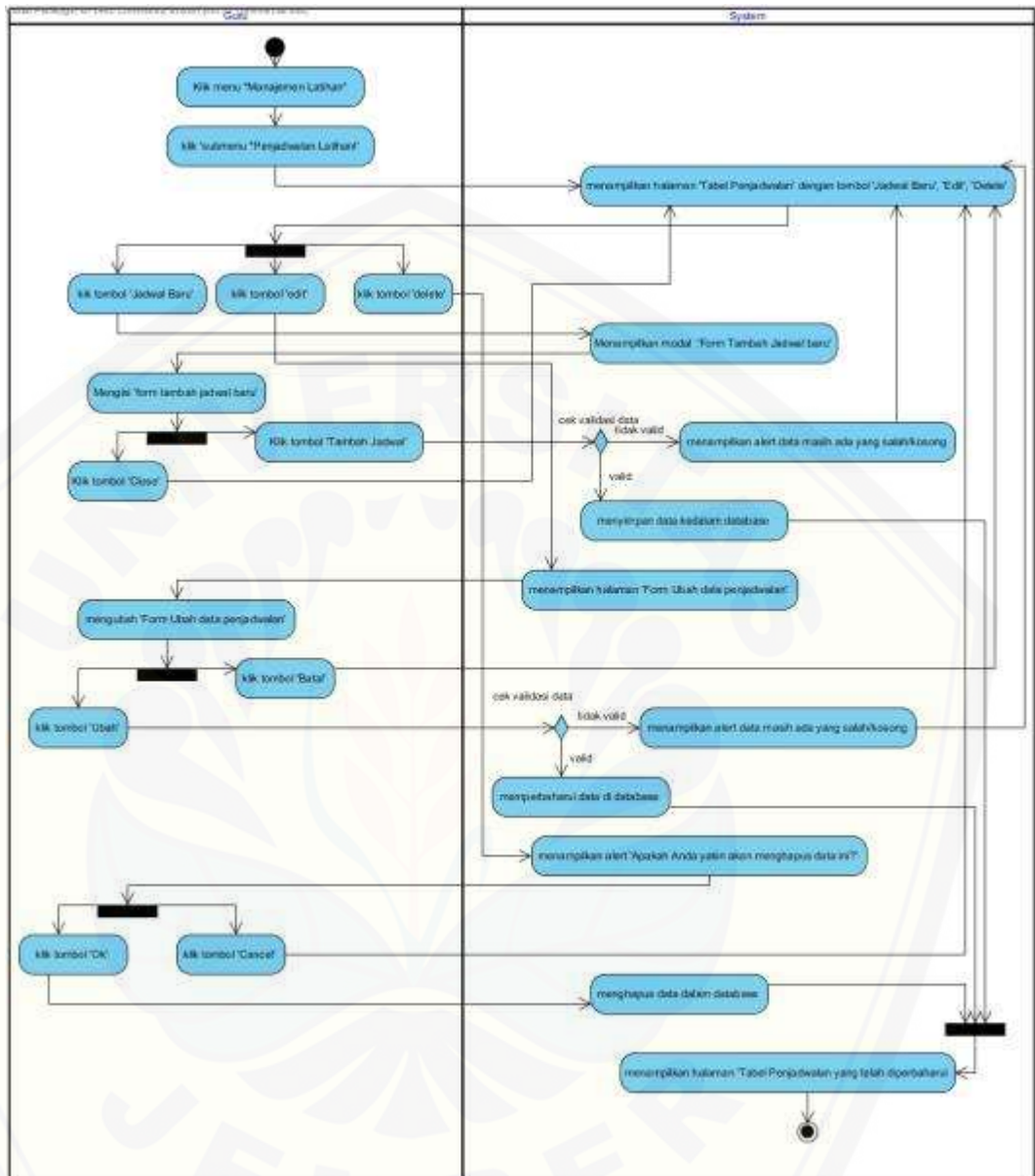
2. Activity diagram manajemen data siswa



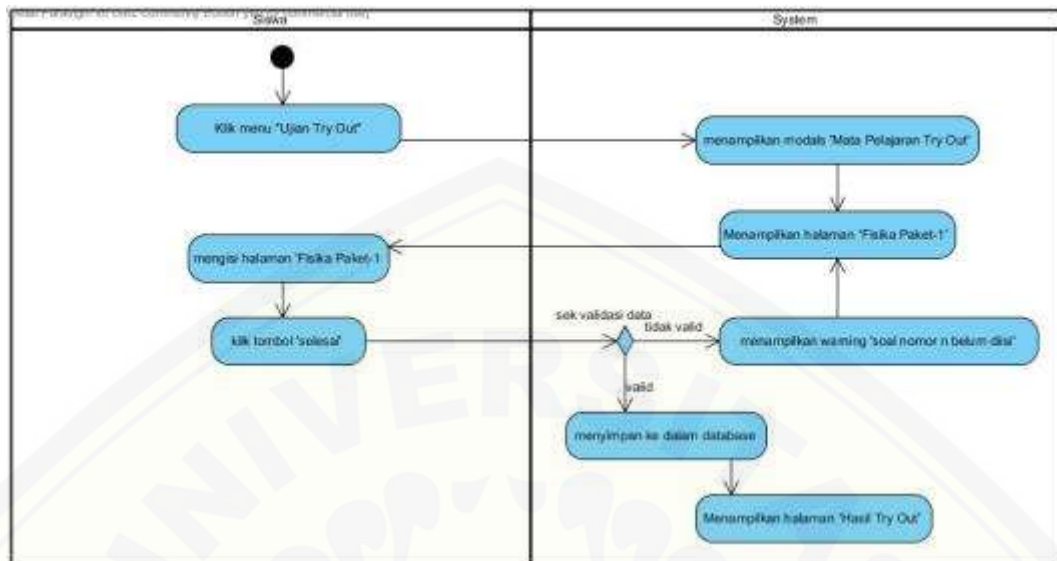
3. Activity diagram manajemen data bobot soal



4. Activity diagram manajemen jadwal latihan



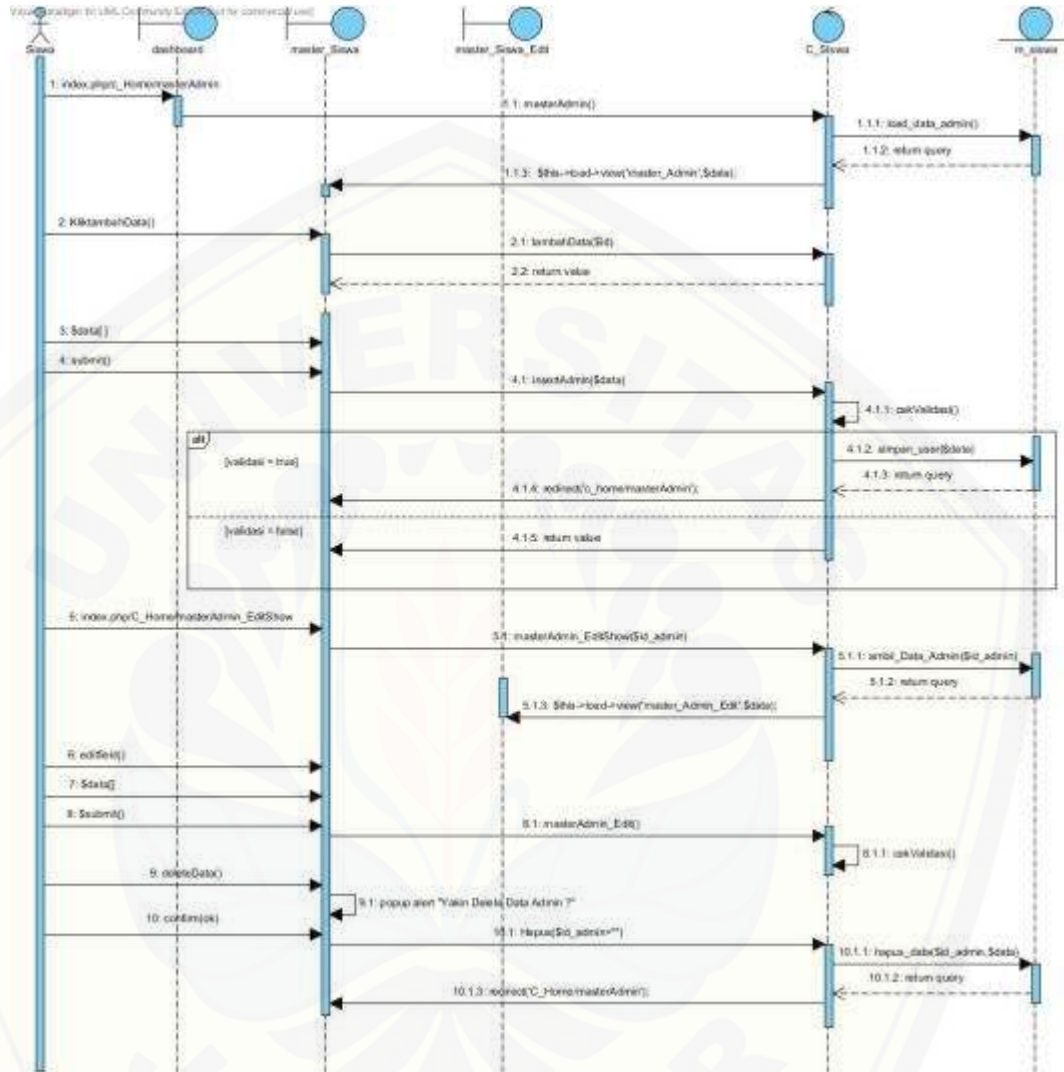
## 5. Activity diagram latihan soal



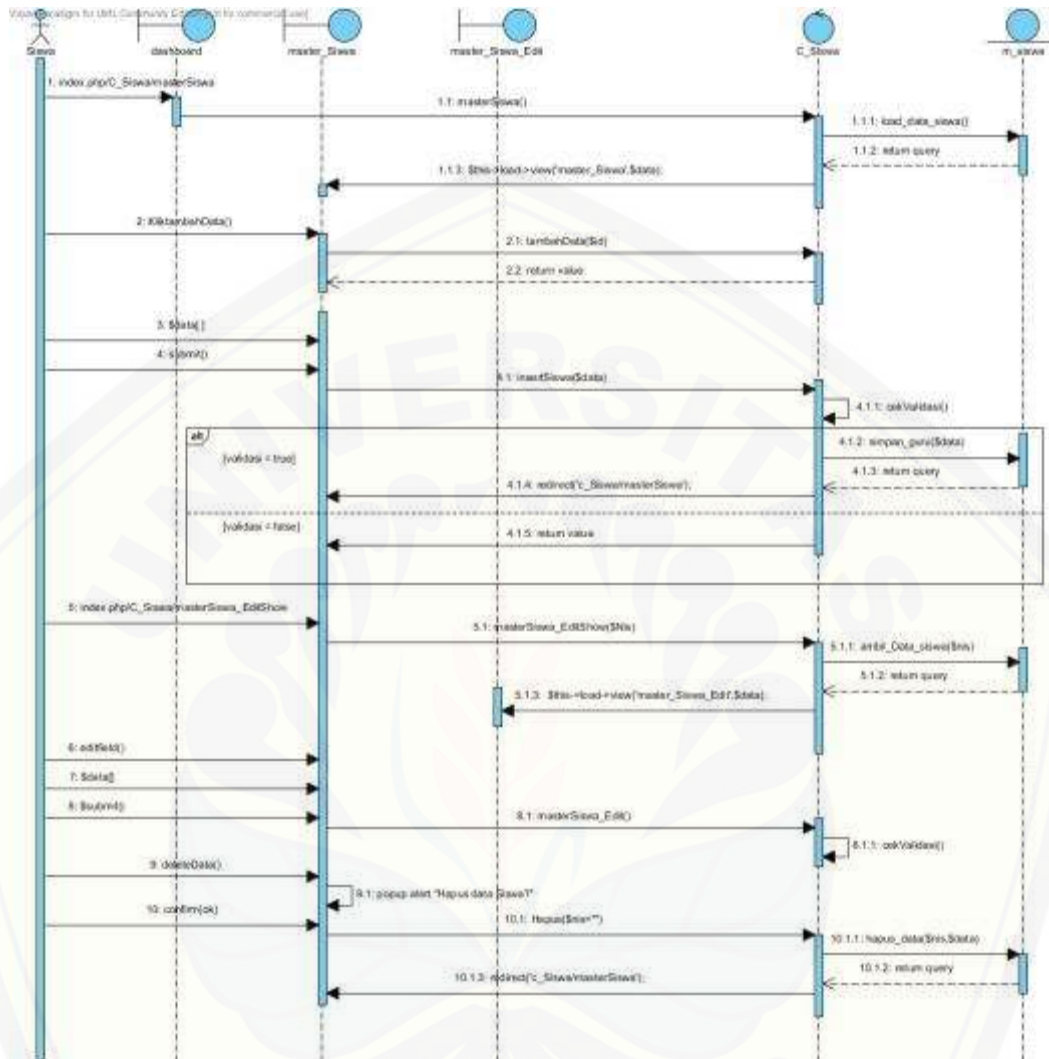


### A.3 Perancangan Sequence Diagram

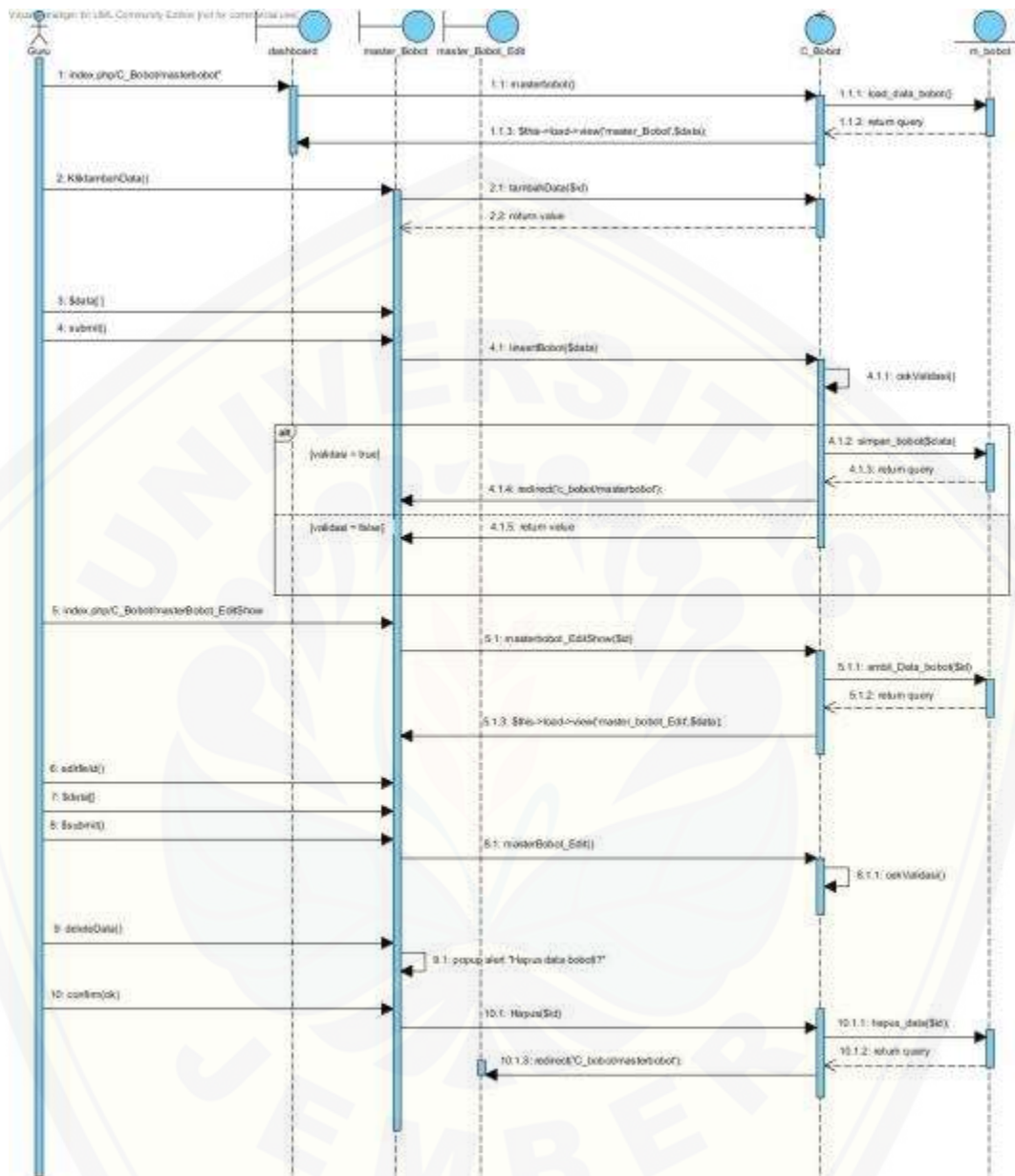
#### 1. Sequence diagram manajemen data user



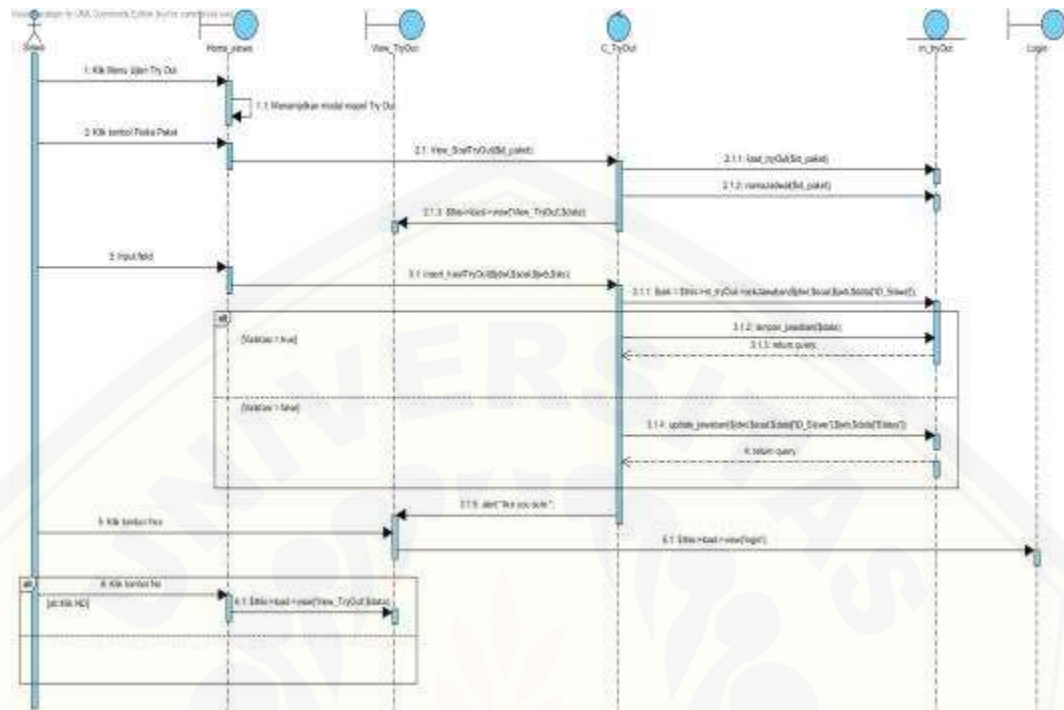
2. *Sequence diagram* manajemen data siswa



3. *Sequence diagram* manajemen data bobot soal



4. *Sequence diagram* manajemen latihan soal



5. *Sequence diagram* manajemen jadwal latihan

