

# **Sim-G (*Simulation Generator*)**

## **TUGAS AKHIR**

Oleh :

<b>Abad Ribunanurin</b>	<b>3311001075</b>
<b>Anis Rahmi</b>	<b>3311001001</b>
<b>Nazilah Latuconsina</b>	<b>3311001069</b>

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BATAM  
BATAM  
2013**

## **LEMBAR PENGESAHAN**

Batam, 10 Februari 2013  
**Pembimbing,**

**Mir'atul Khusna Mufida, S.ST**  
**NIK. 109057**

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3311001075

Nama : Abad Ribunanurin

Adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa proyek akhir dengan judul:

*SIM-G (Simulation Generator)*

Disusun dengan:

1. Tidak melakukan plagiat terhadap naskah karya orang lain
2. Tidak melakukan pemalsuan data
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik. Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 10 Februari 2013

**Abad Ribunanurin**

331001075

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3311001001

Nama : Anis Rahmi

Adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa proyek akhir dengan judul:

*SIM-G (Simulation Generator)*

Disusun dengan:

1. Tidak melakukan plagiat terhadap naskah karya orang lain
2. Tidak melakukan pemalsuan data
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik. Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 10 Februari 2013

**Anis Rahmi**

331001001

## LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3311001069

Nama : Nazilah Latuconsina

Adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa proyek akhir dengan judul:

*SIM-G (Simulation Generator)*

Disusun dengan:

1. Tidak melakukan plagiat terhadap naskah karya orang lain
2. Tidak melakukan pemalsuan data
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik. Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 10 Februari 2013

**Nazilah Latuconsina**

331001069

## KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT, berkat rahmat dan hidayat-Nya, penyusun dapat menyelesaikan Tugas Akhir yang berjudul SIM-G (*Simulation Generator*). Aplikasi ini bertujuan untuk memudahkan pengguna dalam membuat dan mengatur objek simulasi yang diinginkan dengan menyediakan objek-objek simulasi.

Dalam kesempatan ini, penyusun mengucapkan terima kasih kepada :

1. Bapak Dr.Priyono Eko Sanyoto, selaku Direktur Politeknik Batam,
2. Ari Wibowo, MT selaku koordinator Tugas Akhir,
3. Mir'atul Khusna Mufida, S.ST selaku pembimbing Tugas Akhir,
4. Bapak/Ibu Dosen program studi Teknik Informatika atas bimbingannya,
5. Kedua orang tua tercinta yang memberikan dukungan moral dan doa,
6. Teman-teman seperjuangan Teknik Informatika angkatan 2010 yang memberikan semangat.

Dalam penulisan ini, penyusun mengakui bahwa masih terdapat kekurangan-kekurangan dalam penyusunannya. Oleh karena itu, penyusun sangat mengharapkan bantuan dari berbagai pihak berupa kritik ataupun saran guna penyempurnaan selanjutnya. Akhir kata penyusun mengucapkan terima kasih, semoga laporan ini dapat bermanfaat bagi pembaca yang ingin mengembangkan sebuah aplikasi yang serupa.

Batam, 10 Februari 2013

Penulis

## ABSTRAK

### **APLIKASI SIM-G (*Simulation Generator*)**

Penafsiran yang berbeda ketika pembaca memahami isi informasi secara lisan maupun tulisan melalui media cetak, online, brosur, spanduk, baliho membuktikan cara penyampaian informasi yang tidak tepat dan kurang menarik. Melalui simulasi penyampaian informasi menjadi lebih baik dan interaktif.

Dalam pembuatan simulasi, kebutuhan akan komponen serta objek-objek simulasi kurang tersedia, lamanya waktu yang diperlukan untuk membuat komponen dasar objek-objek yang terdapat pada suatu simulasi, serta sulitnya mengatur pergerakan objek yang terkait didalam simulasi tersebut, menyebabkan tertundanya pembuatan simulasi. Hal ini menjadikan perlunya pembuatan suatu aplikasi simulasi yang dapat membantu, menyediakan dan mengatur pergerakan komponen serta objek-objek informasi, sehingga pengguna dapat membuat simulasi secara mudah dan menarik.

Setelah dilakukan implementasi terhadap aplikasi SIM-G maka didapatkan aplikasi yang dapat membuat simulasi, dimana aplikasi tersebut menyediakan objek-objek simulasi dan dapat mengatur pergerakannya.

**Kata kunci : Informasi, Simulasi, SIM-G**

## **ABSTRACT**

### **SIMULATION GENERATOR APPLICATION**

Different interpretation in understanding the information whether oral and writing through printed media, online, brochures, banners, billboards, shows that the way to deliver information that is not appropriate and less attractive. Through simulation better information delivery and interactive.

In making simulation , component needed and simulation objects are less available, a long time needed to make a base components objects contained in a simulation, and the difficulty to regulate the movement related objects in the simulation, causing delays in manufacturing simulation. This makes the need for the development of a simulation application that can help, provide and regulate the movement of components and objects information, so that users can make the simulation easy and interesting.

After implementation of SIM-G application, the result is simulation application, which is providing objects simulation and can regulate their movements.

**Keywords : Information, Simulation, SIM-G**

## DAFTAR ISI

Halaman Judul.....	i
LEMBAR PENGESAHAN.....	ii
LEMBAR PERNYATAAN .....	iii
LEMBAR PERNYATAAN .....	iv
LEMBAR PERNYATAAN .....	v
KATA PENGANTAR .....	vi
ABSTRAK .....	vii
DAFTAR GAMBAR .....	xii
Bab I Pendahuluan .....	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	2
I.3 Batasan Masalah .....	2
I.4 Tujuan.....	2
I.5 Manfaat .....	2
Bab II Landasan Teori .....	3
II.1 Simulasi .....	3
II.1.1 Pengertian Simulasi .....	3
II.1.2 Contoh Penerapan Simulasi.....	4
II.1.3 Kelebihan dan Kekurangan Simulasi.....	4
II.1.4 Model Simulasi.....	5
II.2 Generator ( <i>Engine</i> ) .....	7
II.2.1 Search Engine .....	7
II.2.2 Database Engine .....	8
II.3 Bahasa Pemrograman Java .....	9
II.3.1 Kelebihan dan Kekurangan Bahasa Pemrograman Java.....	9
II.4 NetBeans IDE .....	10
II.5 JavaFX.....	11
II.6 Snagit.....	13
Bab III Analisis dan Perancangan .....	15
III.1 Batasan Sistem .....	15

III.2 Analisa contoh kasus pembuatan KTP.....	16
III.4 Gambaran Kerja Sistem .....	17
III.5 <i>Use Case Diagram</i> Aplikasi Sim-G.....	18
III.5.1 Skenario Use Case.....	19
III.6 Robustness Diagram.....	22
III.6.1 Robustness Diagram Membuka Lembar Kerja Simulasi .....	22
III.6.2 Robustness Diagram Menampilkan Pemilihan Objek.....	22
III.6.3 Robustness Diagram Menyimpan Hasil Inputan Label Objek.....	23
III.6.4 Robustness Diagram Menghubungkan satu objek dengan objek lainnya menggunakan konektor.....	23
III.6.5 Robustness Diagram Menampilkan Pergerakan Objek .....	23
III.6.6 Robustness Diagram Menyimpan Objek Simulasi dalam Bentuk File Video     24	
III.6.7 Robustness Diagram Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate dalam Format .SIG .....	24
III.7 Interaction Sequence Diagram .....	25
III.7.1 Membuka Lembar Kerja Simulasi.....	25
III.7.2 Menampilkan Pemilihan Objek.....	26
III.7.3 Menyimpan Hasil Inputan Label Objek .....	27
III.7.4 Menghubungkan satu objek dengan objek lainnya menggunakan konektor   27	
III.7.5 Menampilkan Pergerakan Objek .....	28
III.7.6 Menyimpan Objek Simulasi dalam Bentuk File Video.....	29
III.7.7 Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate Simulasi dalam Format Penyimpanan .SIG .....	30
III.8 Class Diagram .....	31
III.9 Rancangan Kelas Rinci .....	32
III.9.1 Kelas guiGambar .....	32
III.9.2 Kelas bukaLembarKerja .....	32
III.9.3 Kelas tampilPilihObjek .....	32
III.9.4 Kelas simpanLabelObjek.....	33
III.9.5 Kelas hubungkanObjekdgKonektor .....	33

III.9.6 Kelas tampilGerakObjek .....	33
III.9.7 Kelas simpanObjekSimulasi.....	33
III.9.8 Kelas Objek .....	33
III.9.9 Kelas Konektor.....	34
III.9.10 Kelas File Video .....	34
III.9.11 Kelas File SIG .....	34
III.10 Rancangan Database dan Skema Tabel.....	35
III.11 Algoritma .....	35
III.11.1 Algoritma Membuka Lembar Kerja Simulasi .....	35
III.11.2 Algoritma Menampilkan Pemilihan Objek .....	36
III.11.3 Algoritma Menyimpan Hasil Inputan Label Objek .....	36
III.11.4 Algoritma Menghubungkan satu objek dengan objek lainnya menggunakan konektor .....	37
III.11.5 Algoritma Menampilkan Pergerakan Objek.....	37
III.11.6 Algoritma Menyimpan Objek Simulasi dalam Bentuk File Video .	38
III.11.7 Algoritma Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate Simulasi dalam Format Penyimpanan .SIG .....	38
III.12 Perancangan Antarmuka .....	40
Bab IV Implementasi dan Pengujian.....	41
IV.1 Implementasi Kelas.....	41
IV.2 Implementasi Antarmuka.....	43
IV.3 Pengujian .....	44
IV.3.1 Skenario Pengujian .....	44
BAB V Kesimpulan dan Saran.....	47
V.1 Kesimpulan .....	47
V.2 Saran .....	47
DAFTAR PUSTAKA .....	48

## DAFTAR GAMBAR

Gambar 1. Bentuk Simulasi Pembuatan KTP .....	16
Gambar 2. Deskripsi Sistem Aplikasi Sim-G (Simulation Generator) .....	17
Gambar 3. <i>Use Case</i> Aplikasi Sim-G .....	19
Gambar 4. Robustness Diagram Use Case membuka lembar kerja simulasi.....	22
Gambar 5. Robustness Diagram Use Case menampilkan pemilihan objek .....	22
Gambar 6. Robustness Diagram Use Case menyimpan hasil inputan label objek.....	23
Gambar 7. Robustness Diagram Use Case menghubungkan satu objek dengan objek lainnya menggunakan konektor.....	23
Gambar 8. Robustness Diagram Use Case menampilkan pergerakan objek .....	23
Gambar 9. Robustness Diagram Use Case Menyimpan Objek Simulasi dalam Bentuk File Video .....	24
Gambar 10. Robustness Diagram Use Case menampilkan kembali simulasi yang disimpan dan mengupdate dalam format .SIG .....	24
Gambar 11. Interaction Sequence Diagram Membuka Lembar Kerja Simulasi ...	25
Gambar 12. Interaction Sequence Diagram Menampilkan Pemilihan Objek .....	26
Gambar 13. Interaction Sequence Diagram Menyimpan Hasil Inputan Label Objek .....	27
Gambar 14. Interaction Sequence Diagram Menghubungkan satu objek dengan objek lainnya menggunakan konektor.....	27
Gambar 15. Interaction Sequence Diagram Menampilkan Pergerakan Objek.....	28
Gambar 16. Interaction Sequence Diagram Menyimpan Objek Simulasi dalam Bentuk File Video .....	29
Gambar 17. Interaction Sequence Diagram Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate dalam Format SIG.....	30
Gambar 18. Class Diagram .....	31
Gambar 19. Rancangan Database.....	35
Gambar 20. Interface Aplikasi Sim-G.....	40
Gambar 21. Interface Aplikasi Sim-G.....	43
Gambar 22. Skenario Pengujian.....	44

## DAFTAR TABEL

Tabel 1. Spesifikasi Batasan Sistem pada PC .....	15
Tabel 2. Penggunaan objek pada <i>palette</i> aplikasi SIM-G .....	17
Tabel 3. Daftar Implementasi Kelas.....	41
Tabel 4. Daftar implementasi kelas antarmuka.....	43
Tabel 5. Hasil Rincian Pengujian.....	45

## **Bab I Pendahuluan**

### **I.1 Latar Belakang**

Penyampaian informasi melalui media cetak, online, brosur, spanduk, baliho dapat menimbulkan penafsiran yang berbeda ketika pembaca memahami isi informasi tersebut baik secara lisan maupun tulisan, hal tersebut membuktikan cara penyampaian informasi yang tidak tepat dan kurang menarik. Berbagai cara telah dilakukan untuk membuat informasi yang baik dan lebih interaktif, salah satunya melalui simulasi.

Simulasi merupakan cara penyajian dan penyampaian informasi dengan menggunakan situasi tiruan untuk memahami tentang konsep, prinsip, atau keterampilan tertentu melalui pergerakan objek-objek pada sebuah media tertentu. Dalam dunia IT, banyak software yang mendukung pembuatan simulasi, salah satunya Flash, namun dengan menggunakan Flash kebutuhan akan komponen serta objek-objek simulasi kurang tersedia, dan membutuhkan waktu yang lama untuk membuat komponen dasar objek-objek yang terdapat pada suatu simulasi, serta sulitnya mengatur pergerakan objek yang terkait didalam simulasi tersebut, menyebabkan tertundanya pembuatan simulasi. Hal ini menjadikan perlunya pembuatan suatu aplikasi simulasi yang dapat membantu dalam menyediakan dan mengatur pergerakan komponen serta objek-objek informasi, sehingga pengguna dapat membuat simulasi secara mudah dan menarik. Generator merupakan suatu alat atau aplikasi khusus yang dapat mengatur pergerakan objek-objek tertentu.

Oleh karena itu pembuatan aplikasi Sim-G (*Simulation Generator*) berfungsi untuk memudahkan pengguna dalam membuat dan mengatur pergerakan objek-objek pada suatu simulasi. Sehingga hal ini dapat membantu pengguna dalam merancang dan mengimplementasikan suatu simulasi yang diinginkan.

## **I.2 Rumusan Masalah**

Adapun rumusan masalah yang terbentuk dari latar belakang sebagai berikut:

1. Bagaimana cara menyampaikan informasi melalui simulasi ?
2. Bagaimana cara untuk mempermudah pengguna dalam membuat objek-objek simulasi?
3. Bagaimana membuat aplikasi khusus untuk simulasi, tidak untuk animasi ?

## **I.3 Batasan Masalah**

Adapun batasan masalah yang ada adalah

1. Objek simulasi yang digunakan untuk disimulasikan di lembar kerja pada aplikasi ini dibatasi sebanyak 6 objek utama.

## **I.4 Tujuan**

Tujuan dari penelitian ini adalah membangun serta merancang aplikasi simulasi yang dapat memberikan gambaran secara visual pergerakan objek-objek simulasi kepada pengguna.

## **I.5 Manfaat**

Manfaat dari penelitian ini adalah

1. Menyampaikan informasi melalui simulasi dengan membuat aplikasi simulasi.
2. Memudahkan pengguna dalam membuat dan mengatur objek simulasi dengan menyediakan objek-objek simulasi pada aplikasi simulasi.
3. Membuat aplikasi khusus simulasi.

## **Bab II Landasan Teori**

### **II.1 Simulasi**

#### **II.1.1 Pengertian Simulasi**

Simulasi adalah suatu proses peniruan dari sesuatu yang nyata beserta keadaan sekelilingnya. Aksi melakukan simulasi ini secara umum menggambarkan sifat-sifat karakteristik kunci dari kelakuan sistem fisik atau sistem yang abstrak tertentu. Ada beberapa definisi mengenai simulasi, yaitu :

- Simulasi ialah suatu metodologi untuk melaksanakan percobaan dengan menggunakan model dari satu sistem nyata.
- Simulasi merupakan suatu model pengambilan keputusan dengan mencontoh atau mempergunakan gambaran sebenarnya dari suatu sistem kehidupan dunia nyata tanpa harus mengalaminya pada keadaan yang sesungguhnya.
- Simulasi adalah suatu teknik yang dapat digunakan untuk memformulasikan dan memecahkan model-model dari golongan yang luas.

Simulasi mempelajari sistem secara numerik, dimana terdapat pengumpulan data untuk melakukan estimasi statistik untuk mendapatkan karakteristik asli dari sistem. Pendekatan simulasi memulai pembangunan model sistem nyata. Model tersebut harus dapat menunjukkan bagaimana berbagai komponen dalam sistem saling berinteraksi sehingga benar-benar menggambarkan perilaku sistem. Setelah pembuatan model lalu model tersebut melakukan proses transformasi data ke dalam program komputer sehingga terjadi simulasi.

### **II.1.2 Contoh Penerapan Simulasi**

Pada kasus antrian diambil dari kesibukan yang berhubungan dengan proses pengiriman pesanan di restoran McSatya. Permasalahan muncul pada proses pengiriman pesanan yang dijadwalkan dengan pasti untuk suatu area restoran. Dalam kasus ini diasumsikan bahwa restoran McSatya memiliki 7 outlet yang tersebar di wilayah Surabaya dan Sidoarjo. Contoh Penerapan Simulasi lainnya:

1. Simulasi terbang
2. Simulasi sistem ekonomi makro
3. Simulasi sistem perbankan
4. Simulasi antrian layanan bank
5. Simulasi game strategi pemasaran
6. Simulasi perang
7. Simulasi mobil
8. Simulasi tata kota

### **II.1.3 Kelebihan dan Kekurangan Simulasi**

Beberapa kelebihan model simulasi adalah sebagai berikut :

- a. Simulasi dapat memberi solusi kalau model analitik gagal melakukannya.
- b. Model simulasi lebih realistis terhadap sistem nyata karena memerlukan asumsi yang lebih sedikit.
- c. Perubahan konfigurasi dan struktur dapat dilaksanakan lebih mudah untuk menjawab pertanyaan : *what happen if*.

- d. Dalam banyak hal, simulasi lebih murah dari percobaannya sendiri.
- e. Simulasi dapat digunakan untuk maksud pendidikan.
- f. Untuk sejumlah proses dimensi, simulasi memberikan penyelidikan yang langsung dan terperinci dalam periode waktu khusus.

Namun, model simulasi juga memiliki beberapa kekurangan, yaitu :

- a. Simulasi bukanlah presisi dan juga bukan suatu proses optimisasi. Simulasi tidak menghasilkan solusi, tetapi ia menghasilkan cara untuk menilai solusi termasuk solusi optimal.
- b. Model simulasi yang baik dan efektif sangat mahal dan membutuhkan waktu yang lama dibandingkan dengan model analitik.
- c. Tidak semua situasi dapat dinilai melalui simulasi kecuali situasi yang memuat ketidakpastian.

#### **II.1.4 Model Simulasi**

Model-model simulasi yang ada dapat dikelompokkan ke dalam beberapa penggolongan, antara lain :

- a. Model Stochastic atau probabilistic

Model stokastik adalah model yang menjelaskan kelakuan sistem secara probabilistik, informasi yang masuk adalah secara acak. Model ini kadang-kadang juga disebut sebagai model simulasi Monte Carlo. Di dalam proses stochastic sifat-sifat keluaran (*output*) merupakan hasil dari konsep *random* (acak). Meskipun output yang diperoleh dapat dinyatakan dengan rata-rata, namun kadang-kadang ditunjukkan pula pola penyimpangannya. Model yang mendasarkan pada teknik peluang dan memperhitungkan ketidakpastian (*uncertainty*) disebut model probabilistik atau model stokastik.

b. Model Deterministik

Pada model ini tidak diperhatikan unsur random, sehingga pemecahan masalahnya menjadi lebih sederhana.

c. Model Dinamik

Model simulasi yang dinamik adalah model yang memperhatikan perubahan-perubahan nilai dari variabel-variabel yang ada kalau terjadi pada waktu yang berbeda.

d. Model Statik

Model statik adalah kebalikan dari model dinamik. Model statik tidak memperhatikan perubahan-perubahan nilai dari variabel-variabel yang ada kalau terjadi pada waktu yang berbeda.

e. Model Heuristik

Model heuristik adalah model yang dilakukan dengan cara percobaan, kalau dilandasi suatu teori masih bersifat ringan, langkah perubahannya dilakukan berulang-ulang, dan pemilihan langkahnya bebas, sampai diperoleh hasil yang lebih baik, tetapi belum tentu optimal.

f. Simulasi Analog

Simulasi Analog mempergunakan representasi fisik untuk menjelaskan karakteristik penting dari suatu masalah.

g. Simulasi Simbolik

Simulasi Simbolik yang pada dasarnya adalah model matematik yang pemecahannya dipermudah dengan menggunakan komputer. Disebut juga dengan Simulasi Komputer.

## **II.2 Generator (*Engine*)**

Merupakan suatu software yang didesain untuk menghasilkan aplikasi lain.

Contoh (search engine, database engine, application engine, framework engine).

### **II.2.1 Search Engine**

Pengertian Mesin pencari atau search engine adalah suatu program komputer yang dirancang untuk membantu, mempermudah, mempercepat seseorang menemukan informasi atau data yang diinginkan. Mesin pencari (*search engine*) semacam “penunjuk jalan” untuk mencari sesuatu yang dibutuhkan.

Mesin pencari akan berfungsi setelah mempunyai kriteria database yang dibuat sebelumnya dan akan menampilkan hasil sesuai dengan kriteria mesin pencari. Cara kerja mesin pencari adalah dengan memasukkan kata kunci pada kolom pencarian kemudian mesin pencari akan bekerja melakukan pencarian di database serta menampilkan hasil akurat yang memuat kata kunci tersebut dari database yang ada. Jika pada database tidak ada kecocokan dengan kata kunci maka hasil tidak ditampilkan.

Umumnya orang menyangka bahwa mesin pencari hanya digunakan pada internet, padahal mesin pencari (*search engine*) juga digunakan untuk dunia komputer lainnya termasuk pada software, operasi sistem, dan lain-lain. Sebagai contoh: Anda ingin mencari sebuah file yang tersimpan di hard disk komputer, maka jalan tercepat adalah membuka search pada operasi sistem yang dipakai lantas memasukkan kata atau frase yang diinginkan. Contoh lagi, apabila kita membuka program Photoshop kemudian ingin mengetahui kegunaan tools, maka dengan bantuan mesin pencari Photoshop (*menu help*), mungkin keinginan tersebut dapat terpenuhi.

Banyak situs mesin pencari dapat ditemukan di internet, berikut ini adalah beberapa mesin pencari terkenal:

- Google ( <http://www.google.com/> )
- Yahoo ( <http://www.yahoo.com/> )
- Bing (dulu MSN) ( <http://www.bing.com/> )
- Live ( <http://www.live.com> )
- Altavista ( <http://www.altavista.com> )
- AOL ( <http://www.aol.com> )
- Althweb ( <http://www.althweb.com> )
- Baidu ( <http://www.baidu.com> )
- Looksmart ( <http://www.looksmart.com> )
- Ask ( <http://www.ask.com> )
- Open Directory ( <http://www.dmoz.com/> )

## II.2.2 Database Engine

Merupakan Sebuah mesin *database* (mesin penyimpanan) adalah komponen perangkat lunak yang mendasari bahwa sistem manajemen database (*DBMS*) digunakan untuk membuat, membaca, memperbarui dan menghapus (*CRUD*) data dari database. Kebanyakan sistem database manajemen, termasuk aplikasi mereka sendiri antarmuka pemrograman (*API*) yang memungkinkan pengguna untuk berinteraksi dengan mesin yang mendasari mereka tanpa melalui antarmuka pengguna dari *DBMS*.

Istilah mesin database yang sering digunakan adalah server database atau sistem manajemen database.

Contoh-contoh dari aplikasi database engine misalnya seperti:

- SQL Server, dibuat oleh Microsoft.
- MS Access, dibuat oleh Microsoft.
- Oracle Database, dibuat oleh Oracle.

- MySQL, dibuat oleh MySQL AB.
- Firebird, dibuat oleh komunitas open source berdasarkan dari kode Interbase.
- PostgreSQL, dibuat oleh komunitas open source.
- DB2, dibuat oleh IBM.

## **II.3 Bahasa Pemrograman Java**

Java adalah sebuah bahasa pemrograman pada komputer sama seperti pendahulunya c++ . Bahasa pemrograman ini dikembangkan oleh Sun microsystems yang dikembangkan pada tahun 1995 . Java sendiri merupakan bahasa pemrograman yang berbasis objek, maksudnya adalah semua aspek yang terdapat pada java adalah objek sehingga sangat memudahkan untuk mendesain, membuat dan mengembangkan program java dengan cepat. Kelebihan ini membuat program java menjadi mudah untuk digunakan oleh banyak orang. Bahasa pemrograman ini juga dapat digunakan oleh banyak sistem operasi, seperti : Microsoft Windows, Linux, Mac OS, dan Sun Solaris (*Multiplatform*).

### **II.3.1 Kelebihan dan Kekurangan Bahasa Pemrograman Java**

Beberapa kelebihan Bahasa Pemrograman Java adalah sebagai berikut :

- a. Berorientasi objek, memudahkan untuk mendesign dan mengembangkan program dengan cepat dan teliti, sehingga mudah digunakan. Salah satu bahasa pemrograman yang berorientasi objek secara murni. Mirip c++, Mempunyai sintaks yang mirip dengan bahasa pemrograman c++. Sehingga para pengguna c++ banyak yang hijrah menggunakan Java.
- b. Multiplatform, apat digunakan dibanyak sistem operasi .

- c. Perkembangan yang luas, dari game sampai sistem operasi handphone menggunakan program java . Misalnya Handphone Sonny Ericsson dan Opera Mini 3 yang bertipe .jar (*Java Archive*).
- d. Mempunyai pengumpulan sampah otomatis .

Namun, Bahasa Pemrograman Java juga memiliki beberapa kekurangan, yaitu :

- a. Proses Compile, mengharuskan pengguna mengcompile programnya sebelum dijalankan, berbeda dengan bahasa pemrograman python yang tidak perlu mengcompile terlebih dahulu.
- b. Penggunaan Memori yang besar, Berbeda dengan bahasa pemrograman lain yang hanya membutuhkan memori sedikit.

#### **II.4 NetBeans IDE**

NetBeans IDE merupakan lingkungan pengembangan, sebuah *tool, editor* untuk menulis bahasa pemrograman, mengkompilasi, mencari kesalahan dan membuat sebuah program. Netbeans IDE sendiri dikembangkan menggunakan bahasa Java. NetBeans menyediakan banyak fitur aplikatif yang akan membuat pengembangan sebuah aplikasi menjadi lebih mudah, bahkan sangat lebih mudah. Hampir seluruh pelaksanaan *editing source code* dan pengaturan file konfigurasi dilakukan secara otomatis. NetBeans merupakan sebuah proyek software *OpenSource*.

NetBeans IDE merupakan Integrated Development Environment (IDE) *open source* untuk *software developer*. IDE ini berjalan pada berbagai macam platform termasuk Windows, Linux, Solaris, dan MacOS. Selain itu, NetBeans IDE juga merupakan *software* yang mudah diinstal dan powerful. NetBeans IDE menyediakan kemampuan untuk *developer* dengan semua *tool* yang dibutuhkan untuk membuat aplikasi desktop, aplikasi enterprise, aplikasi web dan juga aplikasi *mobile* yang profesional serta lintas platform. Pengguna dapat menyediakan platform Java, C/C++, PHP, JavaScript ataupun Groovy untuk mengembangkan aplikasi yang mereka inginkan.

Adapun kelebihan Netbeans IDE antara lain :

- a. Mendukung berbagai bahasa pemrograman seperti Java, C/C++, PHP, JavaScript ataupun Groovy
- b. Berjalan pada multiplatform sistem operasi termasuk Windows, Linux, Mac OS, Solaris
- c. Berfungsi untuk pengembangan aplikasi mobile menggunakan bahasa Java
- d. Mendukung untuk pengembangan aplikasi web menggunakan PHP
- e. Mendukung permodelan perangkat lunak dengan UML (Unified Modeling Language)
- f. Terdapat banyak modul untuk mengembangkan lebih lanjut
- g. Merupakan produk *free* tanpa ada batasan penggunaannya
- h. Merupakan produk open source (baca: kode sumber terbuka)

## II.5 JavaFX

JavaFX adalah salah satu teknologi dari Java yang diperuntukan untuk membangun atau merancang aplikasi yang kaya dengan konten multimedia seperti: Gambar, Suara, dan Video yang diintegrasikan kedalam suatu jaringan internet maupun intranet.

JavaFX sering disebut dengan RIA (*Rich Internet Applications*), contoh aplikasi RIA adalah: framework, Curl, GWT, Adobe Flash, JavaFX, Mozilla XUL, dan Microsoft Silverlight. RIA adalah aplikasi semantik web (*Semantic Web Application*) yang sebagian besar karakteristik berasal dari aplikasi desktop, dengan kata lain kita dapat menghadirkan lingkungan desktop ke dalam Web browser, misalnya di desktop kita bisa melihat *thumbnail* koleksi dari photo kita, memutar musik, menonton video, dan masih banyak lagi; begitu juga memungkinkan berjalan di web browser kita. Aplikasi desktop ini bisa disampaikan dengan cara standar berbasis web browser atau secara independen melalui sandboxes atau Virtual Machine, dan hal ini adalah keunggulan yang

dimiliki oleh JavaFX dan sekaligus sebagai pembeda JavaFX dengan teknologi yang dimiliki oleh Java sebelumnya seperti JSE (*Java Standard Edition*), JEE (*Java Enterprise Edition*), JME (*Java Micro Edition*), dan memiliki kekurangan yaitu membutuhkan resource yang banyak atau aplikasi High End.

JavaFX diperkenalkan langsung oleh kepala Developer Sun Microsystem, James Gosling. Oleh James Gosling project ini diluncurkan untuk memberikan tingkat keamanan yang lebih tinggi dan baik yang tidak dimiliki oleh AJAX karena AJAX sendiri menggunakan Javascript yang memang tidak didesain dengan tingkat keamanan dan privasi yang tinggi dan baik. James Gosling yakin selain dengan tingkat keamanan yang tinggi dan baik, dan juga ditambah dengan kemudahan dalam mendesain sebuah program dan lebih deklaratif, JavaFX mampu menyaingi AJAX yang dianggap sangat rumit untuk dikerjakan oleh seorang Non Programmer.

Apabila pemrograman AJAX membutuhkan keahlian seorang Developer untuk menguasai teknik-teknik koding, satu hal yang tidak dimiliki oleh para pembuat situs web, dan merupakan sebuah hambatan bagi mereka dalam mengekspresikan website yang dituntut lebih ekspresif dan dinamis. Sun percaya bahwa JavaFX dapat mengatasi masalah ini dan menargetkan JavaFX dan JavaFX Script untuk dapat didaya gunakan oleh Developer yang bukan programmer.

JavaFX memiliki beberapa keunggulan diantaranya:

1. Fully cross platform.

Meskipun secara resmi versi solaris dan linux belum dirilis, tapi JavaFX telah dapat dijalankan didalam lingkungan unix-based.

2. Mengintegrasikan grafis dengan bantuan tool dari pihak ke-3.

JavaFX termasuk plugin satu set dengan Adobe Illustrator dan Photoshop yang memungkinkan untuk diintegrasikan langsung ke aplikasi JavaFX.

3. Draggable Applet/ Drag to Install

Salah satu yang paling revolusioner dari JDK ini, yaitu applet yang notabene hidup di browser.

## II.6 Snagit

Snagit adalah program *screenshot* yang beroperasi di bawah sistem operasi Windows. Snagit menggantikan fungsi asli Print Screen dengan fitur tambahan. Ini berisi fitur yang paling dibutuhkan oleh para pengguna.

Snagit dapat memotong dan menyesuaikan ukuran gambar hasil *capture*, Fungsi Image, Text, Video, dan Web capture, dukungan Add-ons untuk Microsoft Office, *All in one capture*, Fungsi transparan, memberi efek *blur* pada sebagian hasil capture, dan penambahan Watermark.

Snagit adalah adalah program penangkap image (image capture). Snagit memudahkan Anda menangkap image yang terjadi di layar monitor baik gambar pasif maupun gambar aktif (bergerak). Cara memakainya sangat mudah yaitu:

1. Aktifkan snagit
2. Atur inputnya dengan memilih alternatif yang disediakan sesuai kebutuhan. *Alternatif region* untuk menangkap image dengan ukuran yang berubah-ubah. Untuk menangkap image hanya dapat dilakukan oleh window yang aktif oleh karena itu pilih active window. Bentuk penangkapan juga dapat diatur dengan memilih alternatif yang ada, seperti segitiga, elips, polygon, dan lainnya.
3. Atur output dengan memilih properties pada menu output. Pilih format image yang akan disimpan dalam file. Sebaiknya pilih format JPEG agar jarak yang dibutuhkan tidak terlalu besar. Berilah tanda centang pada preview window agar sebelum disimpan sebagai file, image dapat ditampilkan terlebih dahulu.
4. Tekan Ctrl-Shift-P untuk memulai penangkapan.
5. Ketika mouse pointer berubah bentuk, itu berarti sudah dapat mulai mengambil image pada screen.

6. Edit image yang telah tertangkap menggunakan Snagit editor. Editor Snagit adalah program editing gambar dasar yang datang dengan Snagit. Hal ini dapat digunakan untuk membuat perubahan sederhana untuk screenshot, termasuk menambahkan panah, mengaburkan bagian, atau cropping. The Snagit Editor (foto) menggunakan Microsoft Fluent Ribbon program .

### Bab III Analisis dan Perancangan

Pada bab analisis dan perancangan ini akan menjelaskan tentang batasan sistem, analisa contoh kasus dan rancangan bentuk simulasi pada pembuatan KTP, Generalisasi rancangan objek, Gambaran Kerja Sistem, *Use Case Diagram*, Analisis Kelas, *Interaction Sequence Diagram*, *Class Diagram*, Rancangan Kelas Rinci, Rancangan Database dan Skema Tabel serta Algoritma.

#### III.1 Batasan Sistem

Dalam pembuatan tugas akhir ini memiliki batasan sistem. Adapun spesifikasi pada sistem ini, yaitu :

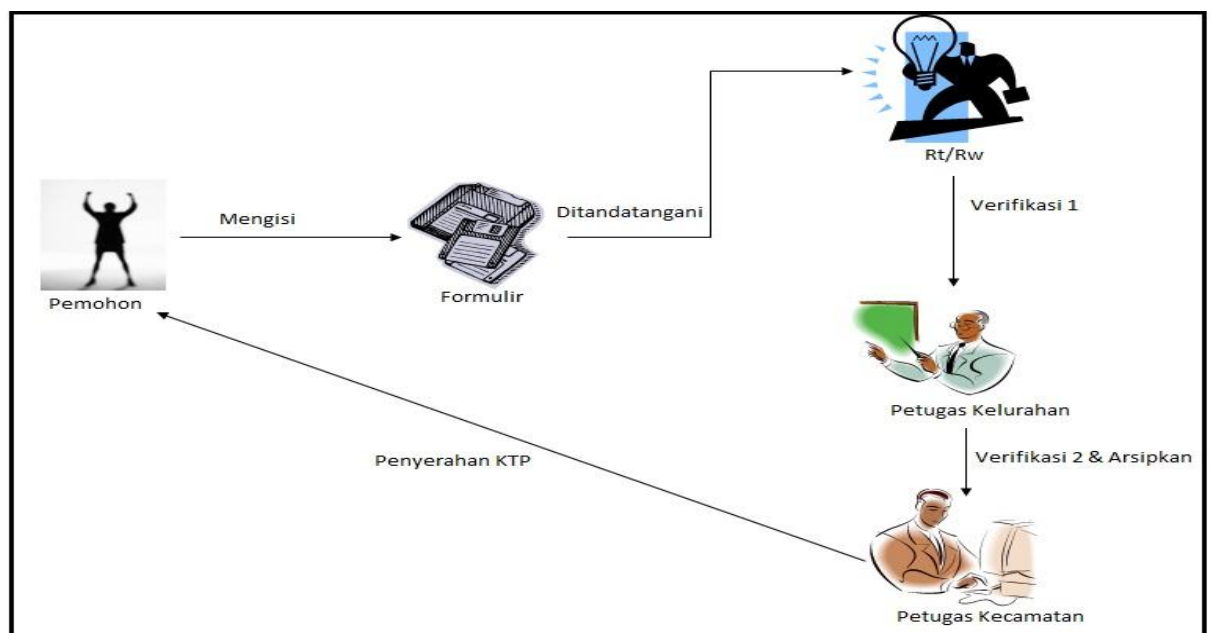
Tabel 1. Spesifikasi Batasan Sistem pada PC

Deskripsi	Spesifikasi	Keterangan
Sistem Operasi	Windows 7	
RAM	512 MB	
IDE	Netbeans	Versi 7.2.1
Software	Java EE	jre7
	JavaFX 2.2	jdk 1.7.0
Library	Default Javafx Platform	Library ini digunakan untuk membuat tampilan grafik suatu aplikasi yang menarik disertai efek animasi

### III.2 Analisa contoh kasus pembuatan KTP

- a. Pemohon mengisi Formulir Master KTP yang berisi data-data pemohon yang ditandatangani oleh RT dan RW
- b. Membawa Formulir Master KTP ke Kelurahan dengan melampirkan persyaratan
- c. Kelurahan menerima dan meneliti berkas permohonan beserta persyaratannya, memberikan nomor register dan memberikan resi tanda terima serta mengirimkan berkas beserta persyaratannya ke Kecamatan
- d. Kecamatan menerima dan meneliti berkas permohonan beserta persyaratannya, kemudian melakukan proses input data berdasarkan data-data yang ada dalam Formulir Master KTP, setelah itu KTP diproses, serta diteliti kembali dan apabila sesuai dengan data-data yang ada, maka Camat menanda tangani KTP Dan mengarsipkan berkas permohonan KTP.
- e. KTP diberikan kepada pemohon.

Dari hasil analisa diatas, menghasilkan informasi yang dikonversi dalam bentuk simulasi pembuatan KTP



Gambar 1. Bentuk Simulasi Pembuatan KTP

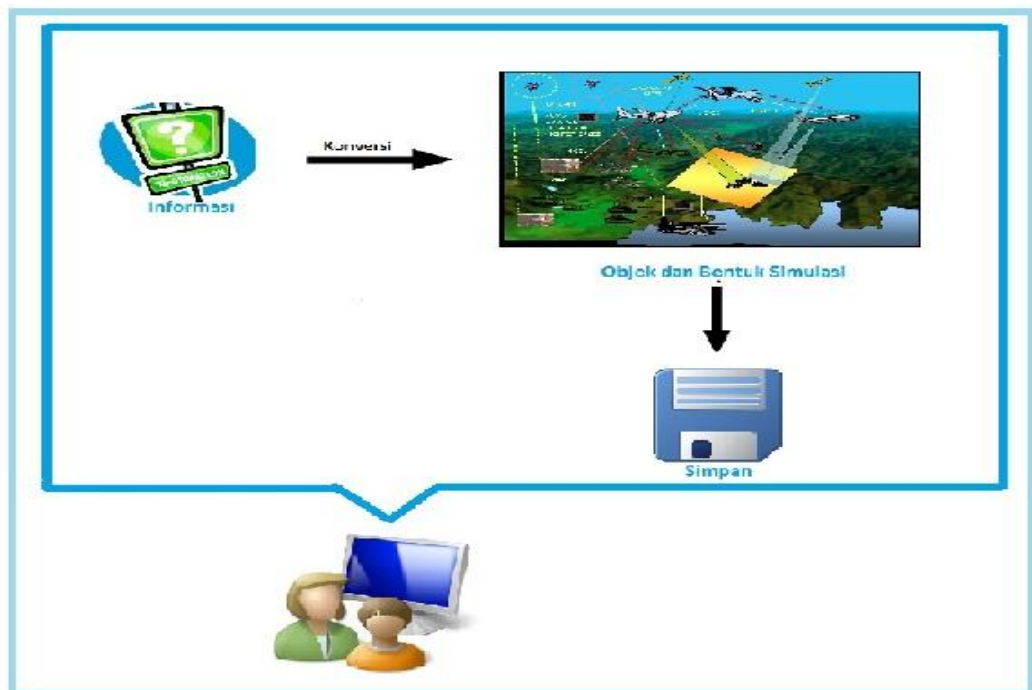
### III.3 Generalisasi rancangan objek

Berdasarkan analisa studi kasus simulasi pembuatan KTP, penggunaan objek yang akan disimulasikan yang terdapat pada *palette* aplikasi SIM-G terdiri dari 5 jenis objek utama, meliputi

Tabel 2. Penggunaan objek pada *palette* aplikasi SIM-G

Orang
Dokumen
Validasi
Bangunan
Lainnya

### III.4 Gambaran Kerja Sistem



Gambar 2. Deskripsi Sistem Aplikasi Sim-G (Simulation Generator)

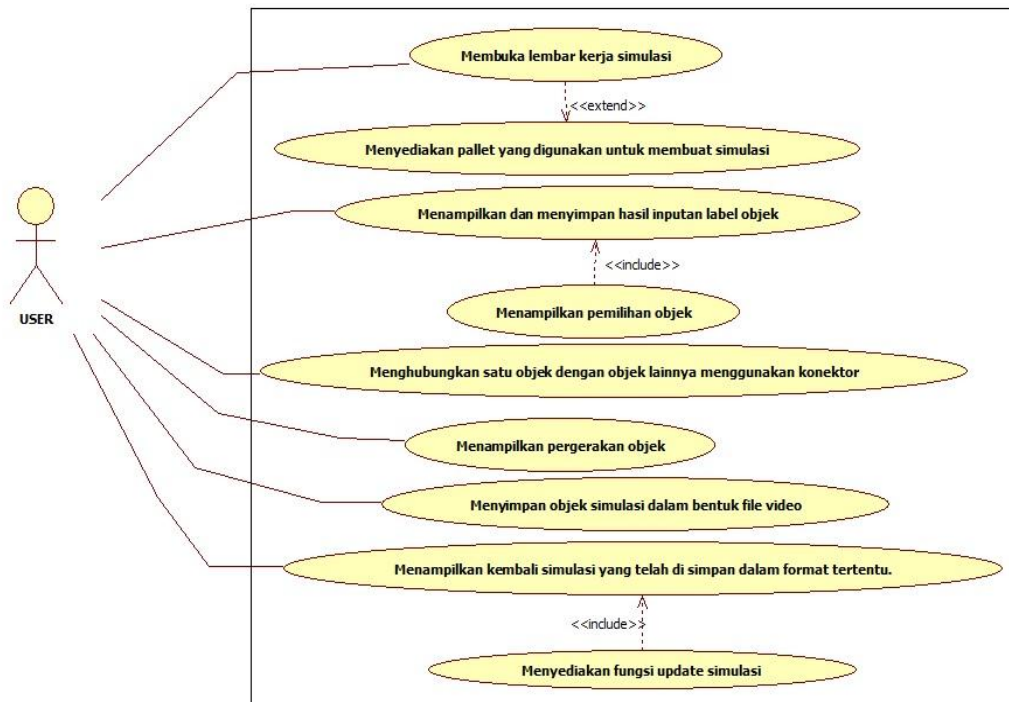
Pada Gambar 2 Deskripsi Sistem Aplikasi Sim-G (Simulation Generator) menjelaskan tentang proses kerja pada Aplikasi Sim-G .

Berikut merupakan spesifikasi sistem aplikasi Sim-G :

1. Sistem mampu membuka lembar kerja simulasi
2. Sistem menyediakan pallet yang digunakan untuk membuat simulasi
3. Sistem menampilkan dan menyimpan hasil inputan label objek dari user
4. Sistem menampilkan pemilihan objek dari user
5. Sistem menghubungkan satu objek dengan objek lainnya dengan konektor
6. Sistem menampilkan pergerakan objek untuk menggambarkan objek yang akan disimulasikan.
7. Sistem dapat menyimpan objek simulasi dalam bentuk file video
8. Sistem menampilkan kembali simulasi yang telah di simpan dalam format file tertentu.
9. Sistem menyediakan fungsi update simulasi yang telah disimpan user.

### **III.5 Use Case Diagram Aplikasi Sim-G**

*Use Case Diagram* aplikasi Sim-G ini digunakan untuk menggambarkan hubungan sejumlah *external actor* dengan *use case* yang terdapat dalam sistem aplikasi Sim-G. *Use Case Diagram* ini hanya menggambarkan keadaan lingkungan sistem yang dapat dilihat dari luar oleh aktor.



Gambar 3. Use Case Aplikasi Sim-G

### III.5.1 Skenario Use Case

Dari gambar 3 dapat dijelaskan bahwa user dapat membuka lembar kerja simulasi, lalu sistem akan menyediakan pallet yang digunakan untuk membuat simulasi. User dapat menampilkan dan menyimpan hasil inputan label objek. User dapat menampilkan pemilihan dan pergerakan objek dengan meng-click tombol. User dapat menyimpan objek simulasi dalam bentuk file video. Serta user dapat menampilkan kembali simulasi yang telah disimpan dalam format tertentu.

#### III.5.1.1 Use Case Membuka Lembar Kerja Simulasi

Kondisi awal : Tersedia *window* atau tampilan layar *GuiMenggambar* beserta pallette

Skenario : User memilih *button* new project

Kondisi akhir : lembar kerja simulasi terbuka

### **III.5.1.2 Use Case Menampilkan Pemilihan Objek dan Menyimpan Hasil Inputan Label Objek**

Kondisi awal : Tersedia *pallette* yang berisi 6 jenis objek

Skenario : User memilih objek simulasi dengan melakukan drag and drop objek dari *pallette* ke lembar kerja pada Gui Menggambar, lalu memberi label pada objek tersebut, serta menghubungkan setiap objek yang akan disimulasikan dengan konektor untuk menjalankan simulasi.

Kondisi akhir : objek simulasi ditampilkan dan disimpan sesuai hasil inputan user

### **III.5.1.3 Use Case Menghubungkan satu objek dengan objek lainnya menggunakan konektor**

Kondisi awal : Tersedia satu objek dengan objek lainnya yang telah diatur, dipilih dan dilabeli user

Skenario : User menghubungkan konektor dari objek pertama ke objek kedua, dan seterusnya sampai dihubungkan ke objek terakhir.

Kondisi akhir : Satu objek dengan objek lainnya terhubung dengan konektor

### **III.5.1.4 Use Case Menampilkan Pergerakan Objek**

Kondisi awal : Tersedia objek yang belum bergerak yang telah dipilih dan dilabeli users

Skenario : User menekan tombol (*Button*) start simulation yang terdapat pada area simulasi yang ingin ditampilkan, kemudian sistem akan menampilkan simulasi.

Kondisi akhir : Simulasi ditampilkan pergerakannya sesuai yang diinginkan user

### **III.5.1.5 Use Case Menyimpan Objek Simulasi dalam Bentuk File Video**

Kondisi awal : Tersedia pergerakan objek yang akan disimulasikan

Skenario : User memilih menu penyimpanan untuk menyimpan simulasi yang berformat file video

Kondisi akhir : objek simulasi tersimpan dalam file video

### **III.5.1.6 Use Case Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate dalam Format Penyimpanan.SIG**

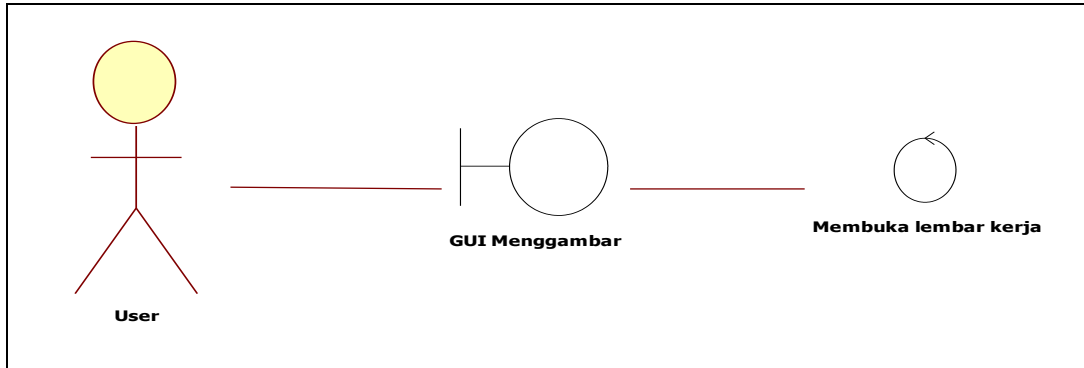
Kondisi awal : Tersedia file simulasi yang belum diubah oleh user

Skenario : User memilih menu open untuk membuka *file* simulasi yang berformat *file* ekstensi .SIG untuk diupdate lagi

Kondisi akhir : file simulasi terupdate sesuai yang diinginkan user.

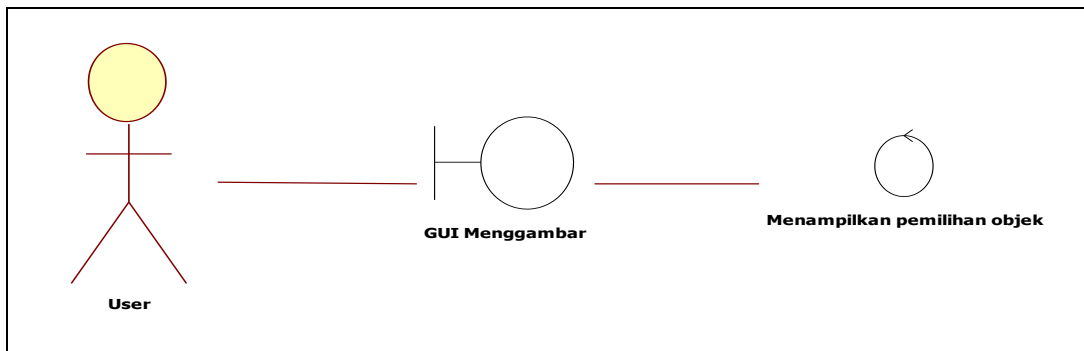
### III.6 Robustness Diagram

#### III.6.1 Robustness Diagram Membuka Lembar Kerja Simulasi



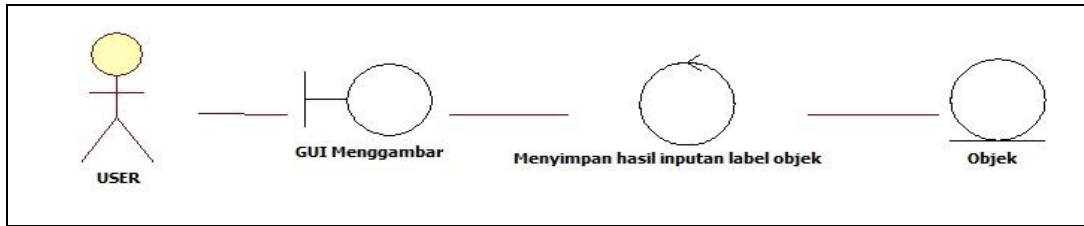
Gambar 4. Robustness Diagram Use Case membuka lembar kerja simulasi

#### III.6.2 Robustness Diagram Menampilkan Pemilihan Objek



Gambar 5. Robustness Diagram Use Case menampilkan pemilihan objek

### III.6.3 Robustness Diagram Menyimpan Hasil Inputan Label Objek



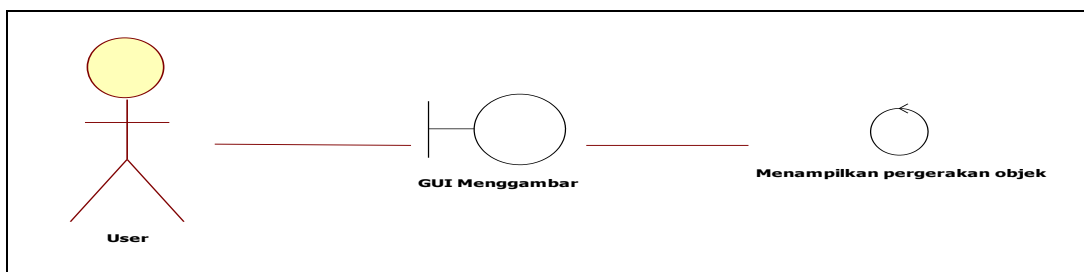
Gambar 6. Robustness Diagram Use Case menyimpan hasil inputan label objek

### III.6.4 Robustness Diagram Menghubungkan satu objek dengan objek lainnya menggunakan konektor



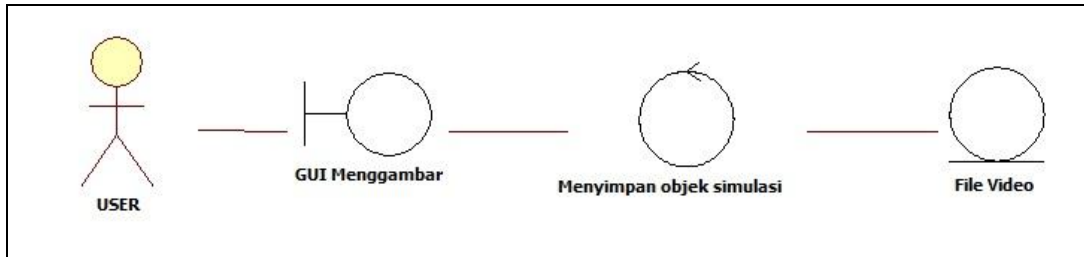
Gambar 7. Robustness Diagram Use Case menghubungkan satu objek dengan objek lainnya menggunakan konektor

### III.6.5 Robustness Diagram Menampilkan Pergerakan Objek



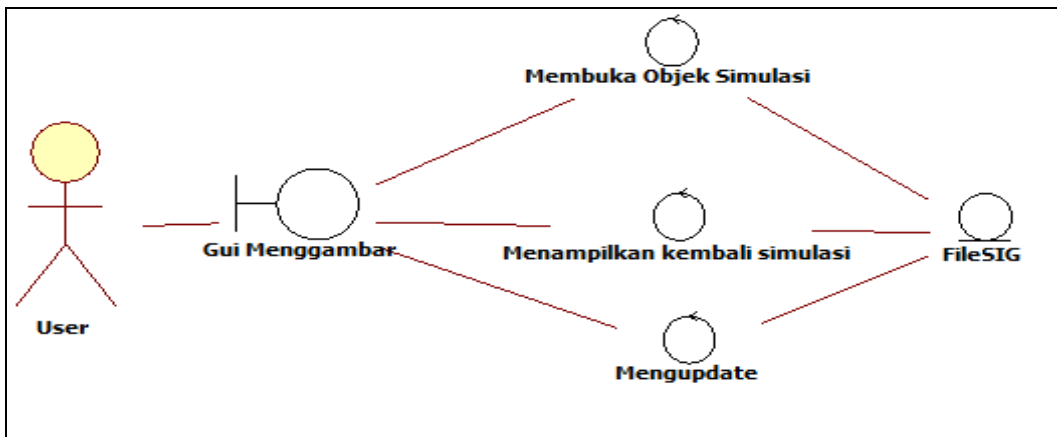
Gambar 8. Robustness Diagram Use Case menampilkan pergerakan objek

### III.6.6 Robustness Diagram Menyimpan Objek Simulasi dalam Bentuk File Video



Gambar 9. Robustness Diagram Use Case Menyimpan Objek Simulasi dalam Bentuk File Video

### III.6.7 Robustness Diagram Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate dalam Format .SIG

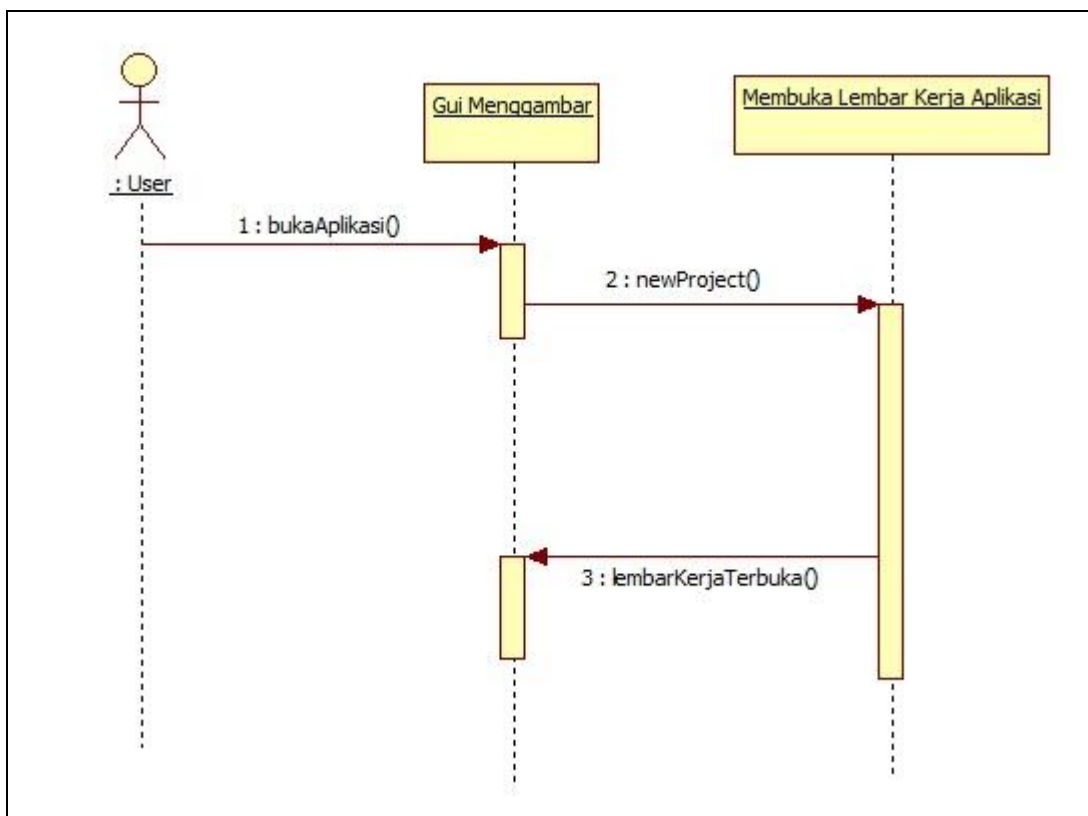


Gambar 10. Robustness Diagram Use Case menampilkan kembali simulasi yang disimpan dan mengupdate dalam format .SIG

### III.7 Interaction Sequence Diagram

Diagram ini menggambarkan urutan proses yang akan terjadi dalam sistem ini. Diagram ini juga menggambarkan method yang dijalankan oleh masing-masing kelas setiap proses yang terjadi pada sistem.

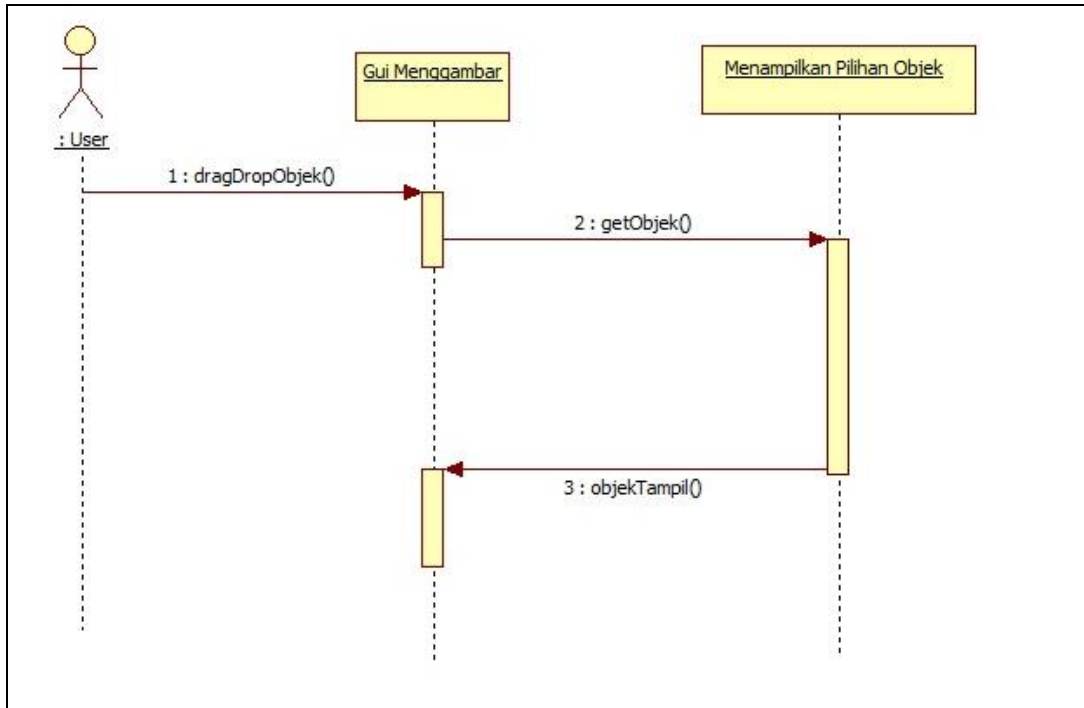
#### III.7.1 Membuka Lembar Kerja Simulasi



Gambar 11. Interaction Sequence Diagram Membuka Lembar Kerja Simulasi

Pada gambar 11 *user* akan membuka aplikasi Sim-G yang terdapat *window* atau tampilan layar *GuiMenggambar* beserta *pallette*, setelah itu *user* melakukan klik pada *button* new project, kemudian *system* akan menampilkan lembar kerja Aplikasi Sim-G.

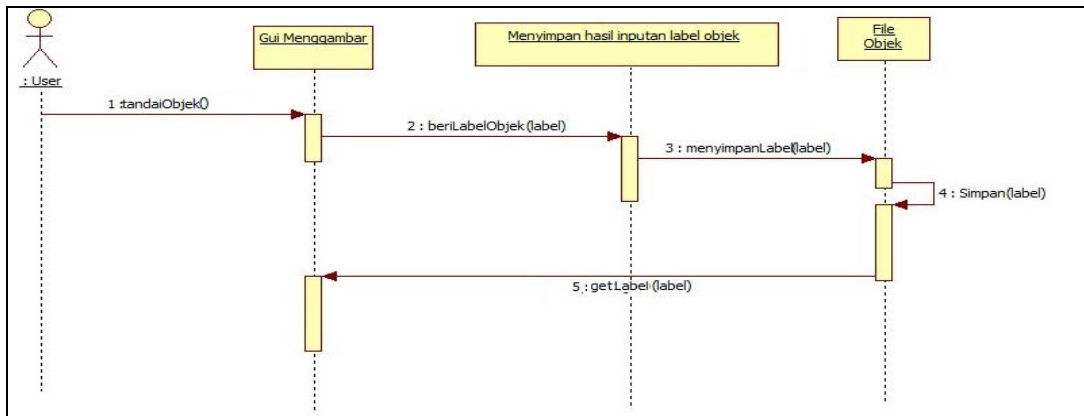
### III.7.2 Menampilkan Pemilihan Objek



**Gambar 12. Interaction Sequence Diagram Menampilkan Pemilihan Objek**

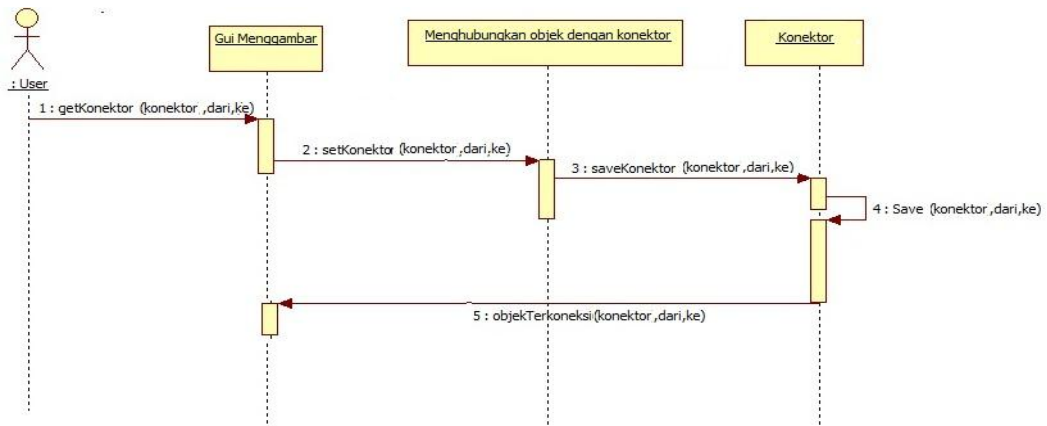
Pada gambar 12 *User* memilih objek simulasi dengan melakukan drag and drop objek dari pallete ke lembar kerja pada Gui Menggambar, lalu getObject dari Objek Simulasi maka sistem akan menampilkan objek yang dipilih pada Gui Menggambar.

### III.7.3 Menyimpan Hasil Inputan Label Objek



Gambar 13. Interaction Sequence Diagram Menyimpan Hasil Inputan Label Objek

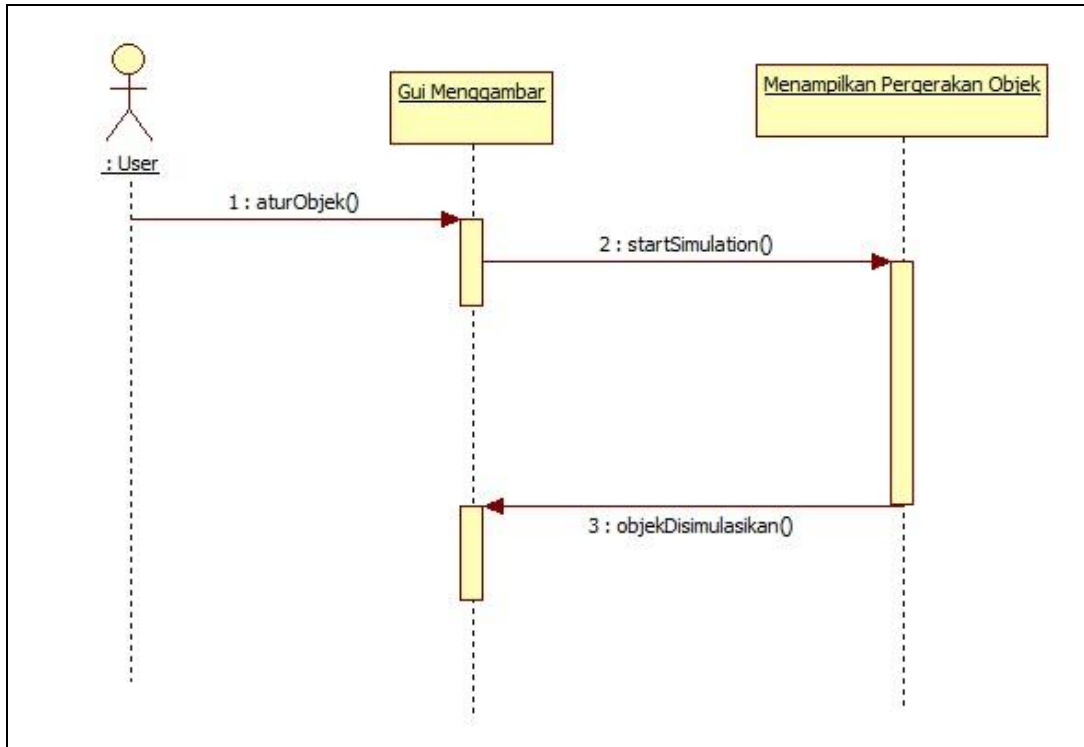
### III.7.4 Menghubungkan satu objek dengan objek lainnya menggunakan konektor



Gambar 14. Interaction Sequence Diagram Menghubungkan satu objek dengan objek lainnya menggunakan konektor

Pada gambar 13 *user* memberikan konektor pada satu objek dengan objek lainnya yang belum bergerak yang telah dipilih dan diberi label , setelah itu sistem akan membuat masing-masing objek terkoneksi.

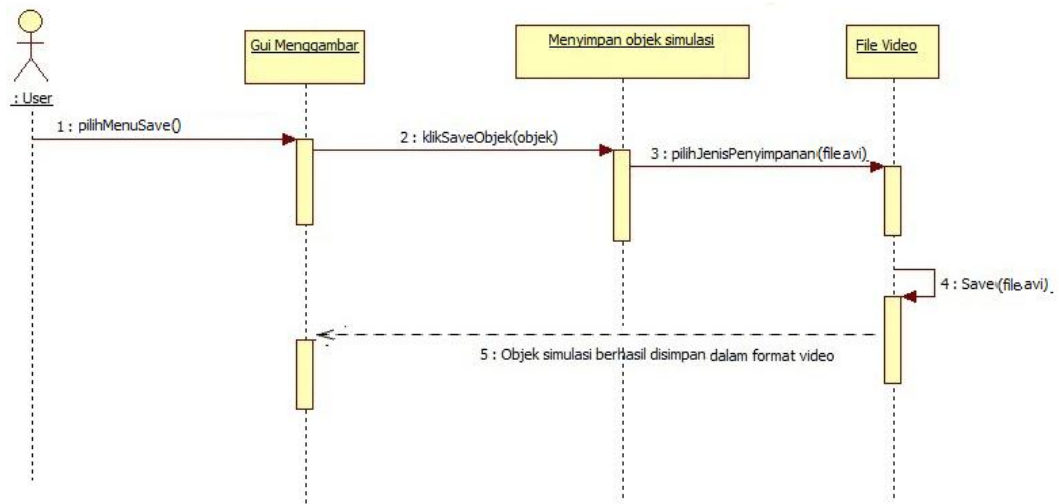
### III.7.5 Menampilkan Pergerakan Objek



Gambar 15. Interaction Sequence Diagram Menampilkan Pergerakan Objek

Pada gambar 15 *user* mengatur objek yang belum bergerak yang telah dipilih dan dilabeli , setelah itu *user* mengklik tombol (*Button*) start simulation yang terdapat pada area simulasi yang ingin ditampilkan, kemudian sistem akan menampilkan pergerakan simulasi pada objek tersebut.

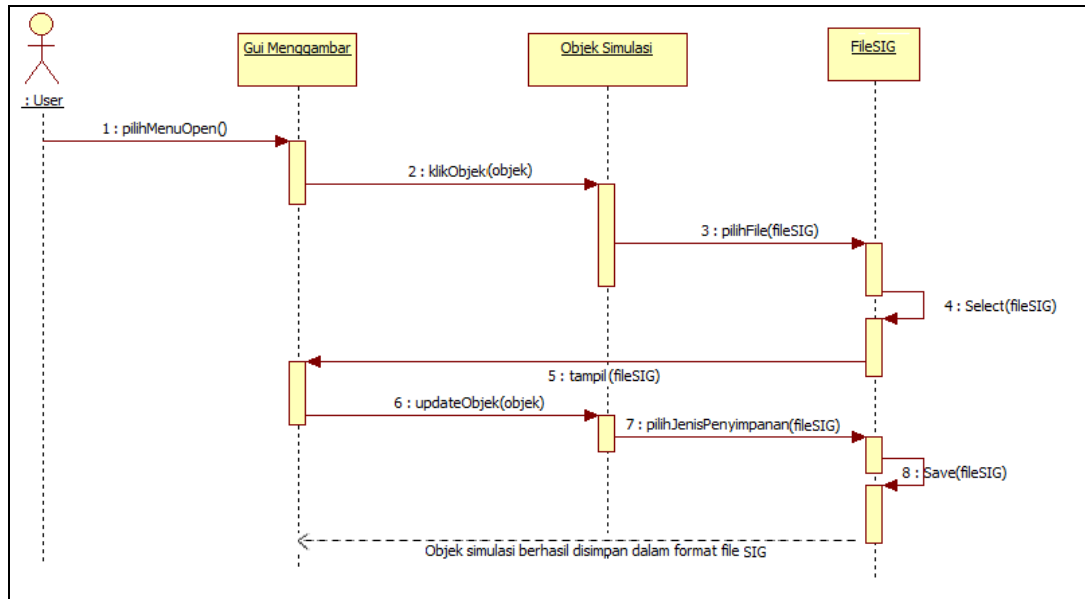
### III.7.6 Menyimpan Objek Simulasi dalam Bentuk File Video



**Gambar 16. Interaction Sequence Diagram Menyimpan Objek Simulasi dalam Bentuk File Video**

Pada gambar 16 Tersedia pergerakan objek yang akan disimulasikan, kemudian *user* memilih menu penyimpanan untuk menyimpan simulasi yang berformat file video, lalu *system* akan melakukan penyimpanan *file* pada media penyimpanan dalam format video.

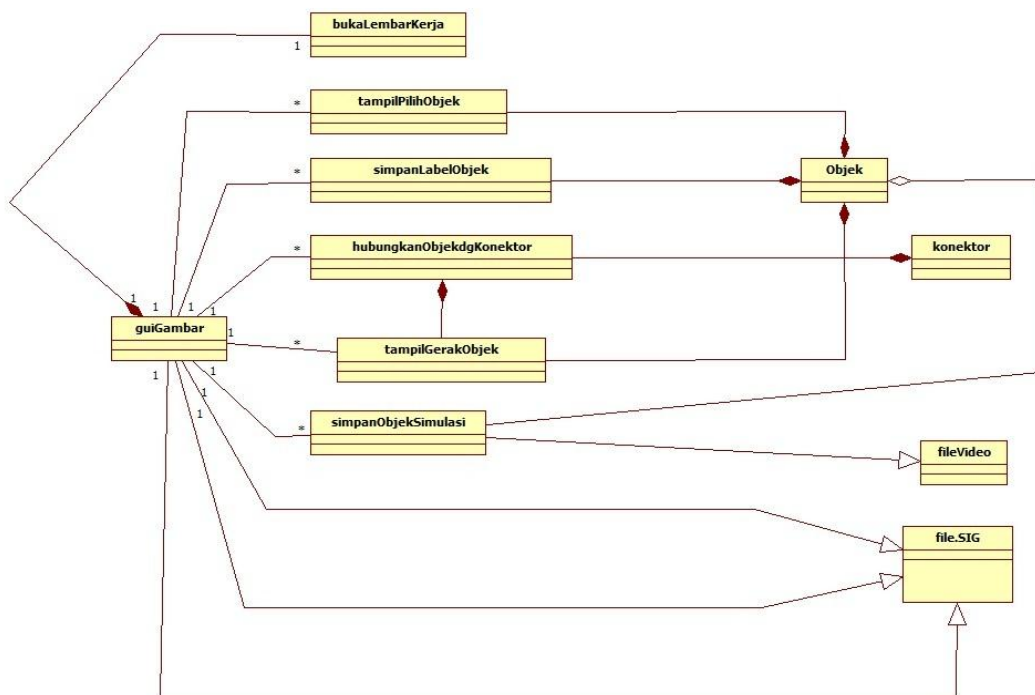
### III.7.7 Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate Simulasi dalam Format Penyimpanan .SIG



**Gambar 17. Interaction Sequence Diagram Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate dalam Format SIG**

Pada gambar 17 Tersedia *file* simulasi yang belum diubah oleh *user* yang sudah tersimpan sebelumnya dalam format yang memiliki ekstensi file.SIG, lalu diatur lagi objek simulasinya, kemudian sistem akan mengganti *file* simulasi sesuai yang di *update* user.

### III.8 Class Diagram



Gambar 18. Class Diagram

Pada gambar 18 menjelaskan hubungan antar kelas dalam suatu diagram kelas, yaitu :

- 1 kelas guiGambar bisa menangani 1 kelas bukaLembarKerja.
- 1 guiGambar bisa menangani banyak kelas tampilPilihObjek.
- 1 guiGambar bisa menangani banyak kelas simpanLabelObjek.
- 1 guiGambar bisa menangani banyak kelas hubungkanObjekdgKonektor.
- 1 guiGambar bisa menangani banyak kelas tampilGerakObjek.
- 1 guiGambar bisa menangani banyak kelas simpanObjekSimulasi.
- 1 guiGambar bisa menangani 1 kelas fileSIG.
- Kelas tampilPilihObjek, simpanLabelObjek, tampilGerakObjek sangat bergantung dengan kelas Objek atau memiliki hubungan *Composition*.
- Kelas simpanObjekSimulasi memiliki hubungan agregasi dengan kelas Objek, karena Kelas simpanObjekSimulasi merupakan bagian dari kelas Objek
- Kelas simpanObjekSimulasi berhubungan statis dengan kelas fileVideo

### III.9 Rancangan Kelas Rinci

#### III.9.1 Kelas guiGambar

guiGambar
-label -objek -konektor -dari -ke
+bukaAplikasi() +dragDropObjek() +objekTampil() +tandaObjek() +getLabel(label) +getKonektor(konektor,dari,ke) +objekTerkoneksi(konektor,dari,ke) +aturObjek() +objekDisimulasikan() +pilihSaveObjek() +pilihMenuOpen() +tampilFileSIG()

#### III.9.2 Kelas bukaLembarKerja

BukaLembarKerja
+ newProject()

#### III.9.3 Kelas tampilPilihObjek

TampilPilihObjek
+ getObject()

### III.9.4 Kelas simpanLabelObjek

SimpanLabelObjek
- label
+ beriLabelObjek(label)

### III.9.5 Kelas hubungkanObjekdgKonektor

hubungkanObjekdgKonektor
-konektor -dari -ke
+setKonektor(konektor,dari,ke)

### III.9.6 Kelas tampilGerakObjek

tampilGerakObjek
+ startSimulation()

### III.9.7 Kelas simpanObjekSimulasi

simpanObjekSimulasi
- objek
+ klikSaveObjek(Objek)

### III.9.8 Kelas Objek

Objek
-objek
klikObjek(objek) updateObjek(objek)

### III.9.9 Kelas Konektor

Konektor
-konektor -dari -ke
Save(konektor,dari,ke)

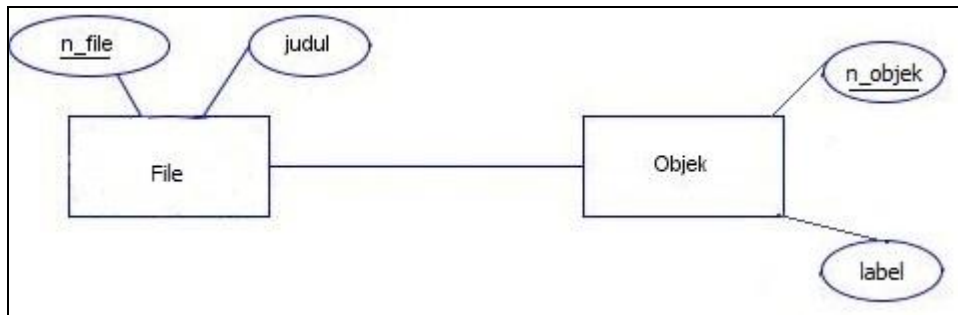
### III.9.10 Kelas File Video

File Video1
- fileavi
+ pilihJenisPenyimpanan(fileavi)
+ Save(fileavi)

### III.9.11 Kelas File SIG

File SIG
- fileSIG
+ pilihFile(fileSIG)
+ Select(fileSIG)
+pilihJenisPenyimpanan(fileSIG)
+Save(fileSIG)

### III.10 Rancangan Database dan Skema Tabel



Gambar 19. Rancangan Database

Pada gambar 19 menjelaskan hubungan *one to many* antara entitas File dan entitas Objek, dimana satu File bisa berhubungan dengan banyak Objek. Entitas File memiliki atribut *n\_file* sebagai *primary key* dan atribut *judul*, sedangkan entitas Objek memiliki atribut *n\_objek* sebagai *primary key* dan atribut *label*.

#### File

<u>n_file</u>	Judu
	1

#### Objek

<u>n_objek</u>	Label
----------------	-------

### III.11 Algoritma

#### III.11.1 Algoritma Membuka Lembar Kerja Simulasi

{ Membuka Lembar Kerja Simulasi }

Inisial state : Tersedia *window* atau tampilan halaman awal SIM-G

Final state : lembar kerja simulasi terbuka

Algoritma:

```
Mulai
Show layar GuiMenggambar
IF pilih menu “ new project” pada GuiMenggambar THEN
Show lembar kerja Simulasi
ENDIF
```

### III.11.2 Algoritma Menampilkan Pemilihan Objek

```
{ Menampilkan Pemilihan Objek }
Initial State : Tersedia objek yang akan dipilih pada palette
Final State : Objek ditampilkan sesuai pilihan

Algoritma:
IF drag drop “ objek” pada palette THEN
    Show objek simulasi
ENDIF
```

### III.11.3 Algoritma Menyimpan Hasil Inputan Label Objek

```
{ Menyimpan Hasil Inputan Label Objek }
Initial State : Objek diberi label
Final State : Label objek tersimpan

Algoritma:
Read label
//melakukan insert objek
Insert into objek values (label)
IF pemrosesan berhasil THEN
write “ label”
ELSE
error message
ENDIF
```

### III.11.4 Algoritma Menghubungkan satu objek dengan objek lainnya menggunakan konektor

```
{ Menghubungkan satu objek dengan objek lainnya menggunakan konektor
}
Initial State : adanya objek yang akan dihubungkan
Final State : satu objek dengan objek lainnya terhubung dengan konektor

Algoritma:
Read dari, ke
//Menggambar konektor
pilih “ konektor”
show konektor
IF konektor dihubungkan pada objek THEN
insert into konektor values (dari,ke)
ELSE
error message
ENDIF
```

### III.11.5 Algoritma Menampilkan Pergerakan Objek

```
{ Menampilkan Pergerakan Objek }
Initial State : terdapat objek-objek yang akan disimulasikan
Final State : Menampilkan pergerakan objek simulasi

Algoritma:
//memulai pergerakan objek yang disimulasikan
tekan tombol ” start”
IF tombol start di klik THEN
Show animasi
Else
Animasi tidak bergerak
ENDIF
```

### III.11.6 Algoritma Menyimpan Objek Simulasi dalam Bentuk File Video

{ Menyimpan Objek Simulasi dalam Bentuk File Video }

Initial State : terdapat objek-objek yang akan disimpan

Final State : Objek tersimpan dalam bentuk File Video

Algoritma:

Read n\_file, judul

//pilih menu penyimpanan

pilih menu “ Save as” pada *GuiMenggambar* THEN

show tampilan penyimpanan

IF pilih jenis penyimpanan “ fileavi” THEN

Insert into file values (n\_file,judul)

ELSE

error message

ENDIF

### III.11.7 Algoritma Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate Simulasi dalam Format Penyimpanan .SIG

{ Menampilkan Kembali Simulasi yang Disimpan dan Mengupdate Simulasi dalam Format Penyimpanan .SIG }

Initial State : tersedia lembar kerja yang digunakan untuk menampilkan kembali file simulasi

Final State : Update simulasi dalam format penyimpanan .SIG

Algoritma:

Read n\_file, judul, n\_objek, deskripsi, label, category, dari, ke

//membuka file simulasi yang telah disimpan

pilih Menu Open

show tampilan penyimpanan

```
IF pilih format file penyimpanan “.SIG” THEN
Select * from file Where n_file = “ SIG”
ELSE
tidak dapat dibuka
ENDIF

//memilih file simulasi yang berformat SIG
Select * from file Where n_file = “ SIG”
Update objek simulasi
IF objek simulasi di update THEN
Update objek,konektor
set deskripsi= “ deskripsi” , label= “ label” ,category= “ category” ,
dari = “ dari” , ke= “ ke” Where n_objek=n_objek
ELSE
tidak terupdate
ENDIF
```

### III.12 Perancangan Antarmuka

Aplikasi Sim-G ini hanya memiliki 1 gui, yaitu gui menggambar.



Gambar 20. Interface Aplikasi Sim-G

## Bab IV Implementasi dan Pengujian

Untuk mengetahui keberhasilan dari program yang telah dirancang, maka perlu dilakukan pengujian terhadap aplikasi ini. Dalam bab ini akan dibahas mengenai proses pengujian yang dilakukan untuk mengetahui keakuratan, keefektifitas, efisiensi dan lain-lain dari aplikasi ini.

### IV.1 Implementasi Kelas

Implementasi kelas dalam Aplikasi SIM-G (*Simulation Generator*) dijelaskan dalam tabel 4.1 berikut:

Tabel 3. Daftar Implementasi Kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable	Keterangan
1	MainPage	MainPage.java	MainPage.class	Menjalankan fungsi : 1. Create lembar kerja 2. Insert dan delete objek simulasi pada lembar kerja 3. Save nama dan label objek simulasi ke database 4. Insert Konektor pada lembar kerja 5. Pergerakan objek simulasi
2	buildLeft	buildLeft.java	buildLeft.class	Menyediakan objek-objek pada <i>palette</i>
3	Koneksi	koneksi.java	koneksi.class	Menghubungkan data ke database mysql
4	n_fileDialog	n_fileDialog.java	n_fileDialog.class	Save nama file

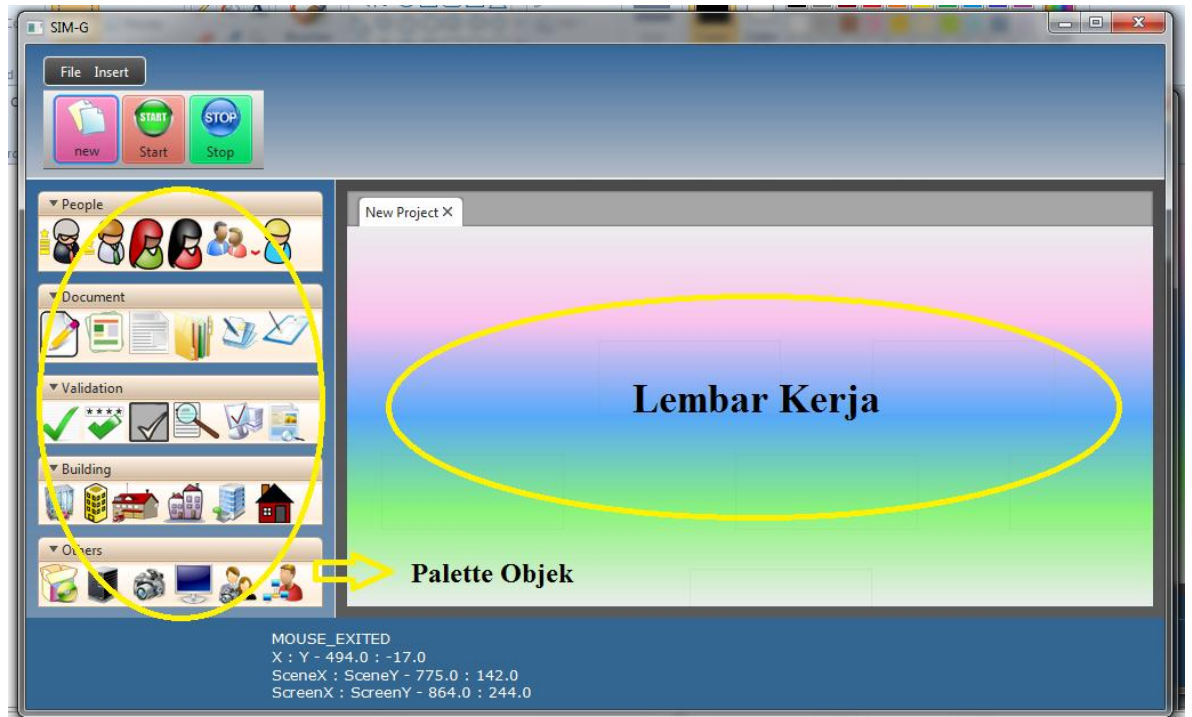
				ke database
5	Demo1	Demo1.css	Demo1.class	Pendukung yang memperindah tampilan pada interface SIM-G

Setelah melakukan implementasi kelas, ternyata dibutuhkan kelas sebanyak 5 kelas, yaitu

1. Sebagai antarmuka dibutuhkan kelas MainPage dan buildLeft, kelas MainPage berfungsi sebagai *form* antarmuka, *toolbar*, *menubar*, pembuatan area lembar kerja simulasi, menyimpan informasi objek simulasi ke dalam database, mengatur pergerakan objek simulasi, serta statusbar yang digunakan untuk mengetahui *event-event* yang terjadi di lembar kerja simulasi pada aplikasi.
2. Kelas buildLeft berfungsi untuk membuat dan menyediakan objek-objek simulasi yang terdapat pada area kiri antarmuka aplikasi.
3. Kelas Koneksi berfungsi sebagai kelas koneksi data ke database.
4. Kelas n\_fileDialog berfungsi untuk membuat tampilan *pop up dialog* yang digunakan user untuk memasukkan nama file lembar kerja kedalam database.
5. Kelas Demo1 berfungsi sebagai kelas yang mengatur dan mempercantik tampilan antarmuka aplikasi.

## IV.2 Implementasi Antarmuka

Implementasi antarmuka dalam Aplikasi SIM-G (*Simulation Generator*) dijelaskan dalam tabel 4 berikut:



Gambar 21. Interface Aplikasi Sim-G

Tabel 4. Daftar implementasi kelas antarmuka

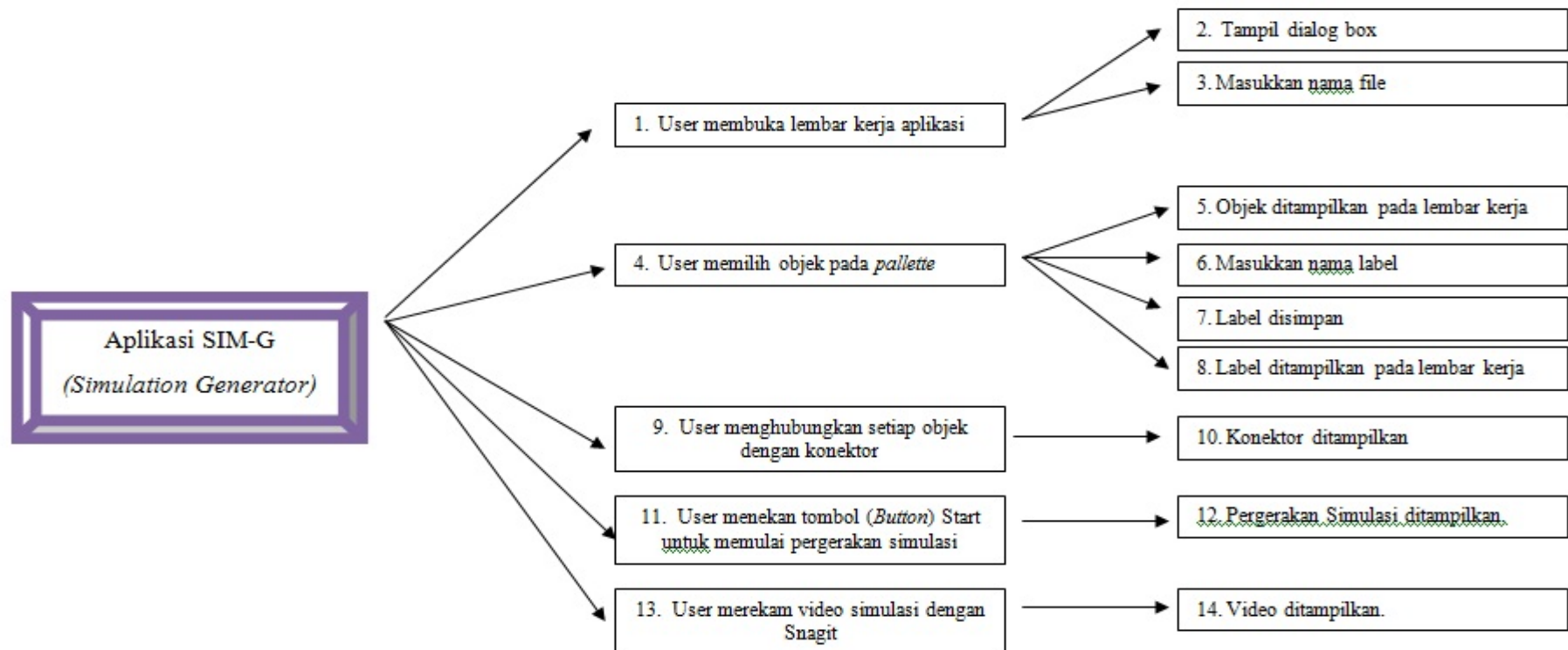
No	Nama Kelas	Nama File Fisik	Nama File Executable
1	guiGambar	MainPage.java	MainPage.class
		buildLeft.java	buildLeft.class

Berdasarkan perancangan dan implementasi yang dilakukan, aplikasi hanya membutuhkan satu antarmuka, karena kegiatan menggunakan objek simulasi serta mengatur pergerakan objek-objek simulasi tersebut dapat dilakukan dalam satu antarmuka saja.

## IV.3 Pengujian

### IV.3.1 Skenario Pengujian

Skenario pengujian implementasi aplikasi Pemodelan Basisdata dapat dilihat pada gambar 22



Gambar 22. Skenario Pengujian

### IV.3.2 Hasil Rincian Pengujian

**Tabel 5. Hasil Rincian Pengujian**

No	Kelas	Use Case	Skenario	Data Uji	Target	Verifikasi
1	guiGambar	Membuka Lembar Kerja Simulasi	User membuka lembar kerja aplikasi	Memberi nama file dengan nama "New Project"	Tampil nama file dengan nama "New Project"	
2	bukaLembarKerja					
3	tampilPilihObjek	Menampilkan Pemilihan Objek dan Menyimpankan Hasil Inputan Label Objek	User memilih objek pada <i>pallette</i>	Memilih objek People pada <i>pallette</i>	Tampil objek Human yang di pilih pada <i>pallette</i>	
				Memilih objek Document pada <i>pallette</i>	Tampil objek Document yang di pilih pada <i>pallette</i>	
				Memilih objek Validation pada <i>pallette</i>	Tampil objek Validation yang di pilih pada <i>pallette</i>	
				Memilih objek Building pada <i>pallette</i>	Tampil objek Building yang di pilih pada <i>pallette</i>	
				Memilih objek Others pada <i>pallette</i>	Tampil objek Others yang di pilih pada <i>pallette</i>	
4	Objek	Menampilkan Pemilihan Objek dan Menyimpankan Hasil Inputan Label Objek	User menyimpan objek simulasi	Simpan objek simulasi dengan n_objek = 1	Objek tersimpan dengan n_objek = 1	
5	simpanObjekSimulasi					
6	simpanLabelObjek					
7	hubungkanObjekdgKonektor	Menghubungkan satu objek dengan objek lainnya menggunakan konektor	User menghubungkan satu objek dengan objek lainnya menggunakan konektor	Menghubungkan satu objek dengan objek lainnya menggunakan konektor	Tampil konektor yang dihubungkan dari objek pertama sampai dengan objek terakhir	
8	Konektor					

9	tampilGerakObjek	Menampilkan Pergerakan Objek	User menekan tombol (Button) Start untuk memulai pergerakan simulasi	Menekan tombol (Button) Start untuk memulai pergerakan simulasi	Tampil pergerakan simulasi	
10	File Video	Menyimpan Objek Simulasi dalam Bentuk File Video	User merekam video simulasi dengan Snagit	Merekam Simulasi dalam Bentuk File Video dengan Snagit	Tampil Simulasi dalam Bentuk File Video	

## BAB V Kesimpulan dan Saran

### V.1 Kesimpulan

1. Aplikasi dapat membuat simulasi dengan cara menyediakan objek-objek simulasi pada *palette* dan menjalankannya dengan konektor, serta menyimpan dalam bentuk file video dengan Snagit.
2. Berdasarkan implementasi yang dilakukan, aplikasi SIM-G menyediakan objek-objek pada *palette* sehingga dapat mempermudah pengguna dalam membuat dan mengatur objek-objek yang diinginkan pada suatu simulasi.
3. Aplikasi SIM-G merupakan aplikasi khusus untuk simulasi yang lebih praktis dari segi kompleksitas fungsi dan target simulasi.
4. Aplikasi SIM-G tidak bisa membuka kembali, mengubah serta menyimpan file sendiri harus dengan bantuan aplikasi lain yaitu Snagit.

### V.2 Saran

Sebagai langkah pengembangan aplikasi ini pada masa yang akan datang, maka saran yang dapat diberikan adalah:

1. Aplikasi SIM-G ini dapat menyimpan dan membuka kembali file yang berisi objek-objek simulasi.
2. Aplikasi SIM-G ini dapat menghasilkan file video dari aplikasi tersebut tanpa bantuan dari aplikasi lain.
3. Semua objek simulasi dapat bergerak tidak hanya gambar dokumen pada konektor.

## DAFTAR PUSTAKA

### Buku

Aldino, S. dan Zanuwarwan, B. 2011. “*Aplikasi Pemodelan Basis Data*“. Politeknik Negeri Batam.

### Internet

Agarwal, A. 2007. *Download SnagIt for Free – Full Licensed Version*. Tersedia di <http://www.labnol.org/software/download/free-download-snagit-with-serial-number/1841/>, diakses pada tanggal 4 September 2012

Figaro, Julian Razif. 2010. *Sekilas Tentang javafx*. Tersedia di [http://opensource.telkomspeedy.com/wiki/index.php/Sekilas\\_Tentang\\_javafx](http://opensource.telkomspeedy.com/wiki/index.php/Sekilas_Tentang_javafx), diakses pada tanggal 2 September 2012

Mubaroki. 2009. *Contoh Engine dan Definisinya*. Tersedia di <http://mubaroki.com/tag/contoh-engine-dan-definisinya>, diakses pada tanggal 2 September 2012

Rachmasita, K. 2009. *Tugas 1 Teknik Simulasi*. Tersedia di [http://karlinarachmasita.blogspot.com/2009/08/tugas-1-teknik-simulasi\\_29.html](http://karlinarachmasita.blogspot.com/2009/08/tugas-1-teknik-simulasi_29.html), diakses pada tanggal 2 September 2012

Savitri. 2010. *Cara membuat KTP, KK, Kipem, KIK, SIMS, KIDS dan SBPDK*. Tersedia di <http://anisavitri.wordpress.com/2010/08/07/cara-membuat-ktp-kk-kipem-kik-sims-kids-%C2%A0sbpdk/>, diakses pada tanggal 4 September 2012

Wikipedia. 2010. *Simulasi*. Tersedia di <http://id.wikipedia.org/wiki/Simulasi>, diakses pada tanggal 2 September 2012

Wiranata, Riko Bayu. 2010. *Sejarah Fungsi dan Pengertian NetBeans*. Tersedia di <http://www.hi-techmall.org/geek/blog/sejarah-fungsi-pengertian-netbeans>, diakses pada tanggal 2 September 2012