

PROYEK AKHIR
MATA KULIAH PEMODELAN BERORIENTASI OBJEK
SEMESTER GANJIL 2013-2014

PEMODELAN KERJA LIFT



Disusun oleh:

Kelompok B Kelas F

Aditya Sudarmadi (NIM 125150102111002)

Riki Hendra Laxsmana (NIM 125150101111011)

I Gde Yogi Mahatma (NIM 125150207111073)

Syafrudin Agustian P (NIM 125150206111008)

M Fajar Lazuardi (NIM 0910963116)

Dosen Pengajar: Wayan Firdaus Mahmudy, Ph.D.

PROGRAM STUDI INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG

DAFTAR ISI

DAFTAR ISI.....	2
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
BAB II DASAR TEORI	3
BAB III ANALISA DAN PERANCANGAN.....	9
BAB IV IMPLEMENTASI DAN PEMBAHASAN.....	12
BAB V KESIMPULAN DAN SARAN.....	15
DAFTAR PUSTAKA	16

BAB I

PENDAHULUAN

1.1. Latar Belakang

Lift atau elevator adalah suatu sarana transportasi yang terdapat di gedung-gedung tinggi yang berfungsi sebagai sarana pengangkut orang atau barang dari lantai satu ke lantai yang lainnya. Lift atau elevator bisa disebut juga sebagai alat pengganti tangga yang umum digunakan dalam kehidupan sehari-hari, umumnya manusia menggunakan tangga untuk menuju ke lantai lainnya, namun masalahnya jika pada suatu gedung terdapat banyak lantai maka hal tersebut akan menjadi masalah besar kalau harus menggunakan tangga. Manusia merasa malas untuk menuju lantai yang lebih tinggi jika harus menggunakan tangga waktu yang dibutuhkan lebih lama serta membutuhkan tenaga yang lebih banyak tentunya.

Untuk itulah dibutuhkan suatu sarana/alat yang lebih praktis yang dapat menanggulangi masalah tersebut. Sejarahnya pada tahun 1853 seorang ilmuwan bernama Elisha Graves Otis pertama kali menciptakan elevator yang awalnya berupa sebuah derek dari tali, melalui inovasi-inovasinya Otis berhasil menciptakan elevator untuk penumpang yang pertama kali dipasang di New York pada tahun 1857. Perjuangan Otis dilanjutkan oleh anak-anaknya dalam membangun suatu system lift. Keluarga Otis inilah yang sangat berjasa sebagai tonggak perkembangan lift atau elevator.

Cara kerja lift , Seorang penumpang di dalam atau di luar lift melakukan suatu interupsi ke system lift yang selanjutnya ditampung dan dijalankan sesuai perintah dari penupang tersebut. seseorang mungkin saja akan menekan tombol yang ada pada setiap lift, hal ini mengakibatkan semua lift mendapatkan permintaan dan akan menjalankan semua permintaan tersebut. Berdasarkan kasus diatas, maka kami ingin membuat sebuah program simulasi kontrol lift secara

paralel yang menerapkan dan membandingkan tiga buah algoritma penjadwalan disk dalam mengerjakan setiap permintaan secara optimal.

1.2. Rumusan Masalah

Bagaimana model penjadwalan lift yang efisien?

2. Bagaimana model kerja lift dengan Use Case, UML dan State Diagram?
3. Bagaimana implementasi program kerja lift?

BAB II

DASAR TEORI

2.1 Lift

Lift adalah salah satu fasilitas selain tangga yang biasanya disediakan di gedung-gedung bertingkat tinggi (lebih dari tiga tingkat) untuk memudahkan manusia mencapai tiap tingkat pada gedung tersebut. Apabila tidak terdapat lift dalam gedung bertingkat tinggi maka manusia akan lebih sulit dalam mencapai tingkat dari suatu gedung terutama dengan rentang yang panjang (misalnya dari lantai 1 ke lantai 15) apalagi dengan membawa barang.

Lift merupakan mesin yang di gunakan untuk kemudahan mengangkut penumpang atau peralatan untuk di bawa ke tempat yang tinggi (lantai bertingkat) dalam sebuah bangunan .Untuk memaksimalkan kinerja dari tiap lift digunakan penjadualan. Ada beberapa algoritma penjadualan yang bisa di gunakan dalam penjadwalan lift yaitu algoritma FCFS, C-SCAN dan C-LOOK.

FCFS (*First Come First Serve*) pada algoritma ini penjadwalan lift didasarkan oleh permintaan yang pertama masuk dalam antrian akan dilayani terlebih dahulu. Algoritma ini tidak efektif, jarak kerja menjadi lift akan sangat besar karena melayani penumpang berdasarkan waktu permintaan yang terlebih dahulu ada. Berikut contoh dari penjadualan FCFS

Posisi Permintaan	Tujuan	Waktu Diminta
Lantai 10 (keatas)	Lantai 49	09.00
Lantai 5 (keatas)	Lantai 20	09.20
Lantai 3 (keatas)	Lantai 20	09.21

Dari table tersebut lintasan yang dilalui oleh lift adalah pertama melayani permintaan dari lantai 10 ke lantai 49, dari lantai 49 lalu turun melayani permintaan dari lantai 5 ke lantai 20 setelah dari lantai 20 turun ke lantai 3

melayani permintaan dari lantai 3 ke lantai 20. Dengan menggunakan algoritma ini panjang lintasan lift menjadi 171 lantai.

C-SCAN adalah salah satu algoritma penjadwalan dimana lift akan melewati lintasan yang ada. Lift akan melayani permintaan dari lantai terbawah lalu naik ke lantai teratas (naik terus sampai lantai teratas) lalu dari lantai teratas turun ke lantai terbawah (turun terus sampai lantai terbawah). Algoritma ini kurang efektif karena lift akan menghampiri lantai teratas atau terbawah walaupun tidak ada permintaan di lantai tersebut. Berikut contoh dari penjadualan C-SCAN (dalam kasus ini dimisalkan tingkat tertinggi adalah lantai 50 dan waktu akses per-lantai 0.5 menit)

Posisi Permintaan	Tujuan	Waktu Diminta
Lantai 10 (keatas)	Lantai 49	09.00
Lantai 5 (keatas)	Lantai 20	09.20
Lantai 3 (keatas)	Lantai 20	09.21

Dari table diatas dapat kita tentukan lintasan yang dilalui oleh lift. Pertama lift akan melayani permintaan dari lantai 10 ke lantai 49 lalu naik lagi sampai lantai tertinggi yaitu lantai 50. Dari lantai 50 lift akan turun ke lantai 1(lantai terbawah di waktu 09.40) lalu melayani permintaan naik. Lift akan pergi ke lantai 3 lalu naik ke lantai 5 dri lantai 5 naik ke lantai 20. Dengan menggunakan algoritman ini panjang lintasan lift menjadi 108 lantai tetapi algoritma ini masih kurang efisien.

C-LOOK adalah algoritma penjadwalan lift yang mirip dengan C-SCAN. Lift akan melayani permintaan satu arah seperti pada C-SCAN tetapi pada algoritma ini pada saat melayani permintaan naik lift akan naik sampai lantai tertatas dimana ada permintaan. Berikut contoh dari penjadualan C-SCAN (dalam kasus ini dimisalkan tingkat tertinggi adalah lantai 50 dan waktu akses per-lantai 0.5 menit)

Posisi Permintaan	Tujuan	Waktu Diminta
Lantai 10 (keatas)	Lantai 49	09.00

Lantai 5 (keatas)	Lantai 20	09.20
Lantai 3 (keatas)	Lantai 20	09.21

Dari table diatas lintasan yang dilalui lift menjadi dari lantai 10 ke lantai 49 lalu turun ke lantai dengan permintaan terbawah ke lantai 3 lalu ke lantai 5 setelah itu ke lantai 20. Dengan menggunakan algoritma ini panjang lintasan lift menjadi 105.

2.2 Kelebihan dan Kekurangan Lift

Kelebihan lift :

- Hemat Daya (bisa dibbilang ramah lingkungan)
- Hemat Tempat (khususnya ruang mesin)
- Efisien terhadap waktu yang digunakan sehingga mempercepat untuk mengarahkan orang-orang pada tempat yang dituju
- Karena menggunakan Gearless motor, maka lebih nyaman.

Kekurangan lift :

- Biaya pembuatan yang cukup mahal
- Membuat orang malas untuk berolahraga dengan menggunakan tangga
- Jika listrik padam maka lift dan escalator tidak dapat digunakan sehingga menghambat kegiatan
- Jika tidak hati-hati bisa menelan korban, seperti : terjepit lift

2.3 Jenis-jenis Lift

Secara umum jenis-jenis dapat di golongan menjadi beberapa kelompok yaitu :

Lift Penumpang (Passenger Elevator)

Lift Barang (Freight elevator)

Lift Observasi (Panorama)

Lift kendaraan

Lift pasien (Bed Elevator)

Dumbwaiter, lift berbentuk kotak kecil yang sering digunakan di restoran atau perpustakaan untuk mengangkut barang-barang dalam suatu gedung

2.4 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifacts* dari sistem *software*, untuk memodelkan bisnis, dan sistem *nonsoftware* lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks (Gunadi dan Suhendar, 2002, h. 26).

2.4.1 Artifact UML

Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut:

1. *use-case diagram*
2. *class diagram*
3. *behavior diagram*:
 - a. *statechart diagram*
 - b. *activity diagram*
 - c. *interaction diagram*:
 - i. *sequence diagram*
 - ii. *collaboration diagram*
4. *implementation diagram*:
 - a. *component diagram*
 - b. *deployment diagram*

Diagram-diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisis rekayasa. Dari berbagai diagram tersebut, kami hanya menjelaskan empat diagram saja yang pada umumnya yaitu

use case, *class*, dan *activity*. Ketiga diagram tersebut akan dijelaskan sebagai berikut.

2.4.2 Use Case Diagram

Use case diagram menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada di luar sistem (*aktor*). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar.

Use case diagram dapat digunakan selama proses analisis untuk menangkap *requirements* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case diagram* menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use case diagram*.

2.4.3 Class Diagram

Class diagram membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antarkelas dan penjelasan detail tiap-tiap kelas didalam model desain (dalam *logical view*) dari suatu sistem.

Selama proses analisis, *class diagram* memperlihatkan aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, *class diagram* berperan dalam menangkap stuktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

2.4.4 Activity diagram

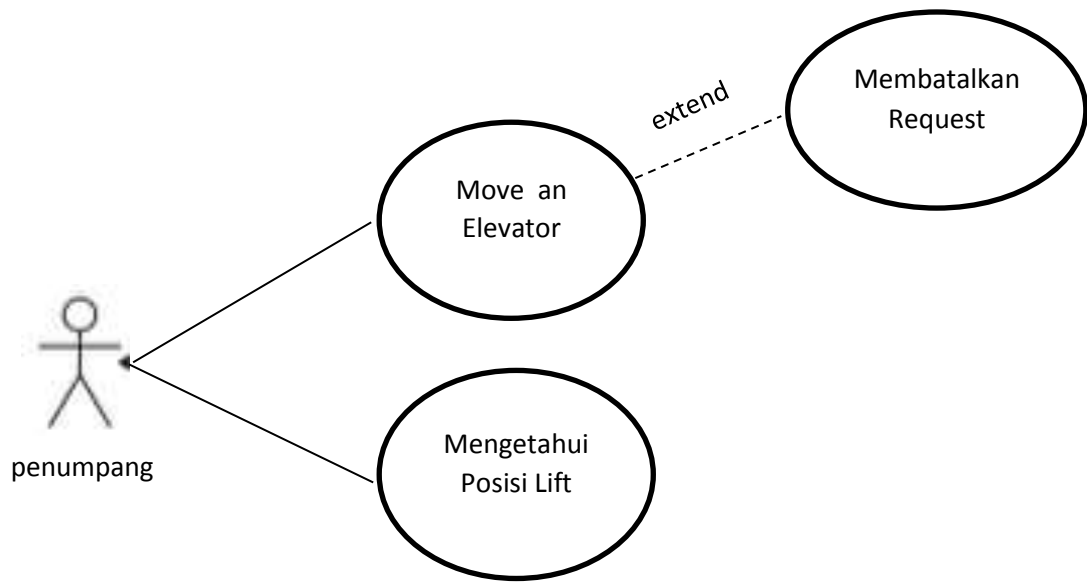
Activity diagram memodelkan alur kaja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*).

Seringkali bermanfaat bila kita membuat sebuah *activity diagram* terlebih dahulu dalam memodelkan sebuah proses untuk membantu kita memahami proses secara keseluruhan. *Activity diagram* juga sangat berguna ketika kita ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi.

BAB III

ANALISA DAN PERANCANGAN

3.1 Use Case Diagram Pemodelan Lift



Gambar 3.1 Use case diagram Pemodelan Lift

Gambar *use case diagram* di atas menunjukkan bahwa terdapat aktor yang terlibat secara langsung dengan *use case* tersebut. Untuk melakukan proses proses pada sistem yang akan dibuat, penumpang dapat memindahkan elevator (menekan tombol elevator) penumpang juga bisa membatalkan Requestnya . Selain itu Penumpang juga dapat melihat Posisi Lift.

3.5.5 Activity Diagram Pemodelan Lift

Untuk lebih memperjelas proses-proses yang terjadi pada *use case diagram* di atas, berikut ini adalah *activity diagram* yang menggambarkan secara lebih detail proses yang terjadi pada *use case*.

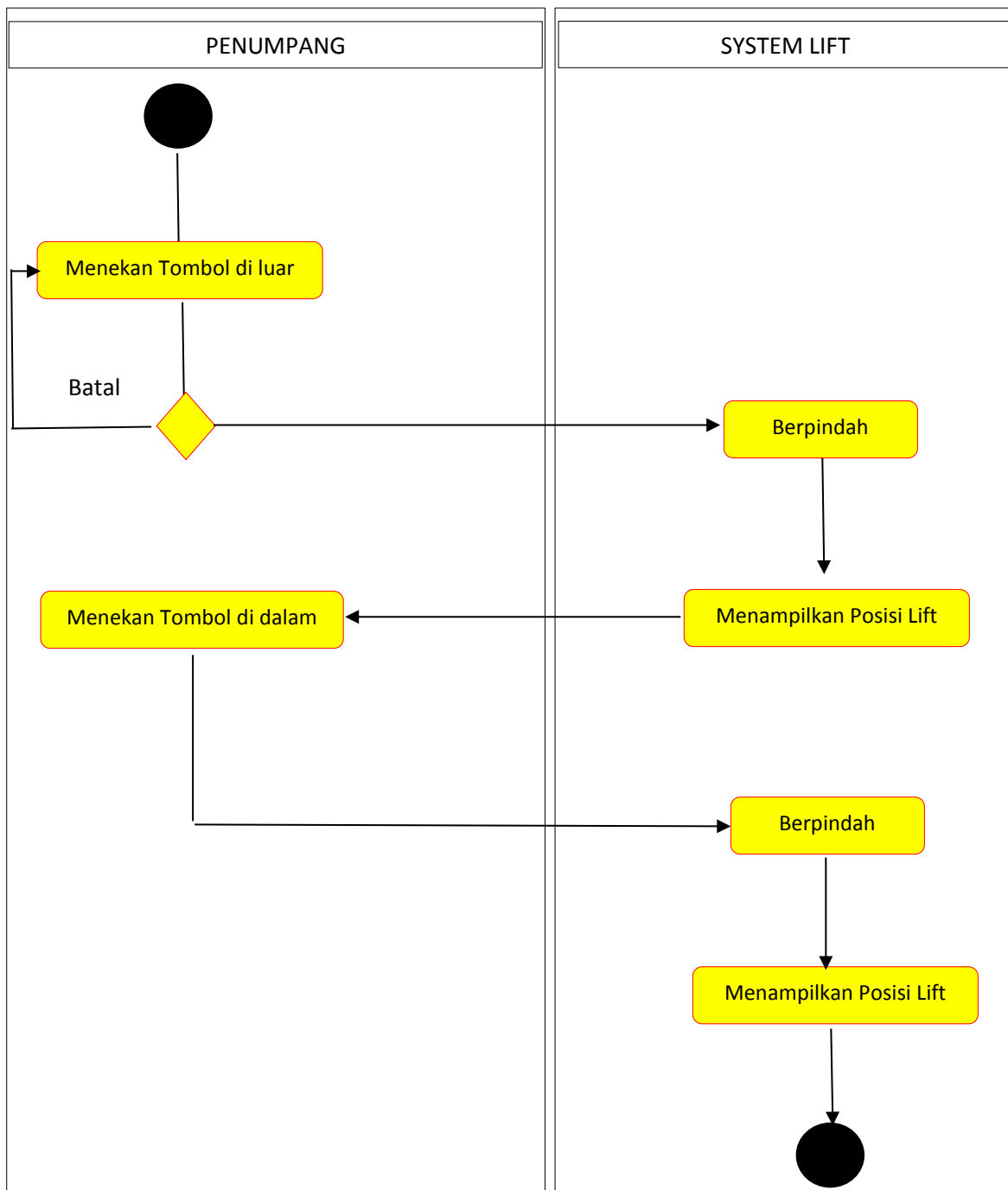
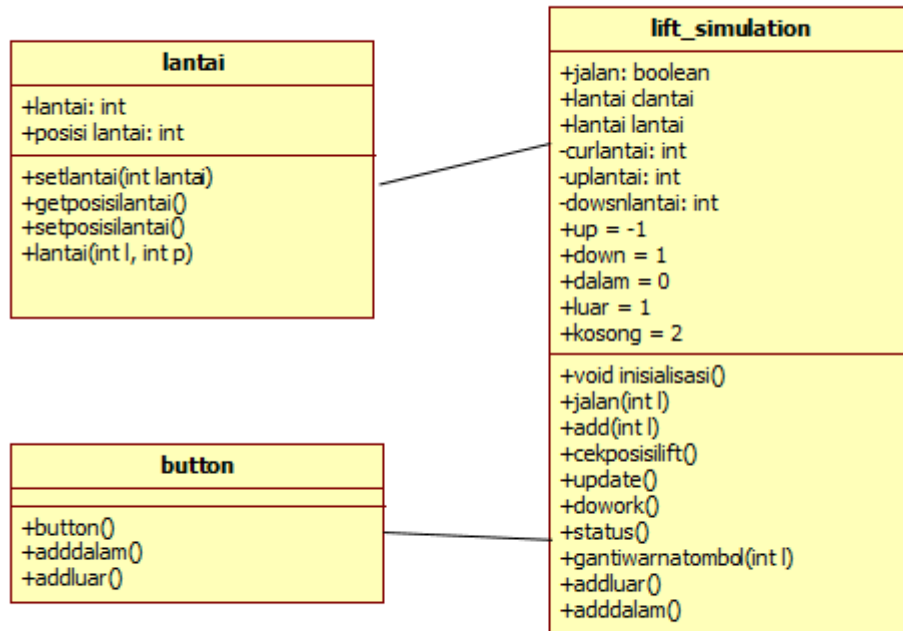


Diagram di atas menjelaskan proses yang terjadi di dalam sistem, yaitu antara pen.umpang dan sistem. Pada saat Penumpang menekan tombol dari luar sistem langsung dijalankan. Kemudian penumpang menekan tombol di dalam lift, perintah dari penumpang membuat lift bergerak sesuai perintah lantai yang diinputkan penumpang. Lift lalu menampilkan Posisi lantai

3.5.6 Class Diagram Sistem Lift

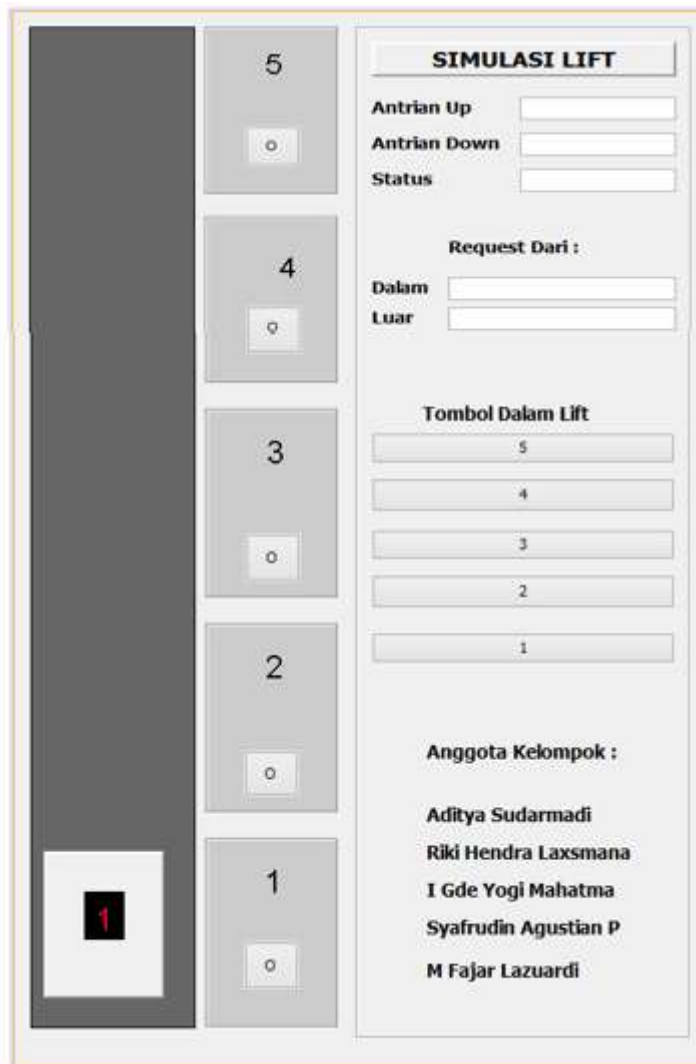


BAB IV

IMPLEMENTASI DAN PEMBAHASAN

Implementasi Antarmuka Program

Implementasi antarmuka program pada sistem pemodelan Lift dilakukan dengan menampilkan interface program beserta pembahasannya dan kode programnya di dokumentasikan dalam bentuk file program serta dituliskan pada halaman lampiran. Berikut ini beberapa tampilan antarmuka program beserta pembahasannya.



Gambar 4.1 Tampilan Interface Pemodelan system lift yang kami buat



Gambar 4.2 Objek Lift berisi tentang Status Lantai (Current Lantai)



Gambar 4.3 Keterangan Lantai Beserta Tombol di Luar Lift

SIMULASI LIFT

Antrian Up [3, 4, 5]

Antrian Down []

Status UP

Gambar 4.4 Menjelaskan tentang Antrian dari Sebuah Lantai pada Lift, UP jika menuju Ke atas, Down Jika menuju Ke Bawah

Tombol Dalam Lift

5

4

3

2

1

Gambar 4.5 Menjelaskan Tombol di dalam Lift, Berwarna merah jika terdapat Request dari User, akan berwarna abu kembali jika request tersebut telah selesai dijalankan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pembuatan sistem pemodelan lift ini dan sebagai akhir dari penulisan laporan, kami dapat menarik beberapa kesimpulan sebagai berikut :

1. Dengan menggunakan sistem ini, pengguna dapat mengetahui cara / mekanisme dari sistem lift
2. Pengguna Lift dapat dengan mudah menuju lantai yang akan ia tuju
3. Penumpang/ pengguna Lift dapat membatalkan request yang dijalankan sehingga lift tidak dieksekusi

5.2 Saran

Dari berbagai fakta yang ada, setiap sistem pasti memiliki pengembangan dalam bentuk versi terbaru dari sistem yang pertama kali dibuat. Sebaik apapun sistem yang dibuat pasti memiliki kekurangan yang perlu untuk dikembangkan, yaitu dengan membuat versi terbaru dari sistem tersebut. Untuk itu dibuatlah penyampaian dalam bentuk saran untuk kesempurnaan sistem ini di masa mendatang. Adapun saran tersebut adalah sebagai berikut :

1. Sistem Pemodelan Lift ini perlu adanya status pintu terbuka dan tertutup yang belum ada pada sistem kami
2. Perlunya ditambahkan fasilitas alarm yang berguna ketika terjadi gangguan teknis daripada lift tersebut
3. Kecepatan Lift perlu ditambah

DAFTAR PUSTAKA

Raharjo B. 2012. Mudah Belajar Java Revisi Kedua. Jakarta: Informatika Jakarta.

Zainul R. 2010. Perancangan Sistem Informasi berbasis Java
<http://repository.amikom.ac.id> , diakses pada tanggal 10 Oktober 2013.