

PERANCANGAN PROGRAM SIMULASI LIFT SEBAGAI ALAT BANTU PEMBELAJARAN ALGORITMA LOOK

Youllia Indrawaty^[1], M.Ichwan^[2], Galih Pratiwi^[3]

Jurusan Teknik Informatika
Institut Teknologi Nasional Bandung

ABSTRAK

Algoritma LOOK adalah algoritma penjadwalan disk pada sistem operasi yang memiliki prinsip kerja sesuai dengan namanya, yaitu seolah-olah seperti dapat "melihat" apakah di depan arah pergerakannya masih ada permintaan atau tidak. Disk arm pada algoritma LOOK tidak berjalan sampai ujung disk, tetapi hanya berjalan sampai pada permintaan yang paling dekat dengan ujung disk. Setelah melayani permintaan tersebut, disk arm akan berbalik arah dan berjalan serta melayani permintaan-permintaan yang ada di depan yang sesuai dengan arah pergerakannya. Prinsip kerja algoritma LOOK ini hampir sama dengan prinsip kerja algoritma penjadwalan SCAN atau biasa juga disebut dengan algoritma elevator (lift), karena prinsip kerjanya sama seperti yang digunakan lift di sebuah gedung tinggi dalam melayani permintaan penggunaannya. Bila pada umumnya prinsip kerja lift diidentikan dengan algoritma SCAN, maka pada penelitian ini akan dibangun program simulasi lift dengan menerapkan algoritma LOOK. Pada lift terdapat 2 aspek utama yang dapat ditinjau, yaitu komponen pembangun serta prinsip kerja dalam mengeksekusi antrian permintaan. Karena beragamnya komponen pada lift dan terdapat beberapa objek yang memiliki perilaku sama maka diperlukan metode analisis berorientasi objek (OOP) untuk mendapatkan komponen-komponen yang dibutuhkan dalam program simulasi lift yang di bangun. Program simulasi lift dengan algoritma LOOK ini diharapkan dapat membantu proses pemahaman pada pembelajaran algoritma penjadwalan disk khususnya algoritma penjadwalan LOOK pada sitem operasi komputer.

Kata Kunci : Algoritma LOOK, Program Simulasi, Lift, OOP.

ABSTRACT

LOOK algorithm is a disk scheduling algorithm on the operating system working principles as the name suggests, that seems like a can "see" whether in front of the direction of movement is still there is demand or not. Disk arm LOOK algorithm does not run until the end of the disk, but only runs until the request closest to the end of the disk. After serving the request, the disk arm will reverse direction and run while serving the demands that exist in front of him in accordance with the direction of movement. LOOK algorithm working principle is similar to the working principle Scan scheduling algorithm or algorithms also called elevator (lift), because the principle works the same as that used an elevator in a tall building in serving the demand of users. If the general principle is identified with the working

elevator algorithm SCAN, then the research is to construct an elevator simulation program by applying the algorithm LOOK. In the elevator there are 2 main aspects to be reviewed, namely the component builder and working principles in executing the request queue. Due to different components on the elevator it needs an object-oriented analysis methods (OOP) to obtain the necessary components in the elevator simulation program in the wake. Elevator Simulation Program with LOOK algorithm is expected to help the process of understanding the learning algorithm, disk scheduling algorithm LOOK particularly on the operating system

Key Word : LOOK algorithm, Simulation Program, elevator, OOP.

I. Pendahuluan

1.1 Latar Belakang

Dalam mempelajari sistem operasi komputer banyak materi yang harus dibahas, salah satunya adalah mengenai sistem penjadwalan proses atau biasa disebut dengan penjadwalan *disk*. Penjadwalan proses merupakan basis sistem operasi *multiprogramming*. Penjadwalan merupakan fungsi dasar dari sistem operasi, dimana di dalamnya terdapat kumpulan kebijakan dan mekanisme yang berkaitan dengan urutan kerja dalam sistem komputer. Penjadwalan bertugas untuk memutuskan proses yang harus berjalan, kapan dan berapa lama proses itu harus berjalan. Dalam penjadwalan, proses yang belum mendapat jatah alokasi dari CPU akan mengantri di *ready queue*. Pada tahap ini algoritma dibutuhkan untuk mengatur giliran proses-proses yang akan dijalankan.

Algoritma penjadwalan dalam sistem operasi terbagi menjadi beberapa jenis, seperti algoritma penjadwalan FCFC, SSTF, SCAN, algoritma penjadwalan LOOK dan lain sebagainya. Setiap algoritma penjadwalan memiliki sistem kerja yang berbeda dalam melayani antrian proses, karena setiap algoritma tersebut memiliki skala prioritas yang berbeda. Prinsip kerja algoritma LOOK sesuai dengan namanya, seolah-olah seperti dapat "melihat" apakah di depan arah pergerakannya masih terdapat permintaan atau tidak. Algoritma LOOK ini memprioritaskan antrian berdasarkan

posisi permintaan (permintaan yang ada di depan), melayani permintaan yang dilalui selama pergerakan sampai pada ujung permintaan kemudian berbalik arah dan bergerak kembali untuk melayani permintaan yang berada didepannya dan searah dengan arah geraknya.

Prinsip kerja algoritma penjadwalan dapat diterapkan pada alat, sarana atau kegiatan yang memiliki konsep kerja berupa antrian. Seperti pada sistem kerja lift, yang alur kerjanya dipengaruhi oleh antrian permintaan dari pengguna. Prinsip kerja lift dipengaruhi oleh kondisi antrian permintaan yang beragam, yaitu adanya permintaan dari dalam maupun dari luar kabin, serta waktu kedatangan permintaan yang tidak bersamaan. Pada umumnya prinsip kerja lift berdasarkan algoritma penjadwalan SCAN, namun ada pula lift yang bekerja berdasarkan algoritma LOOK. Algoritma LOOK secara konsep prinsip kerjanya hampir sama dengan algoritma SCAN karena algoritma LOOK ini merupakan algoritma penyempurna dari algoritma SCAN. Perbedaannya terdapat pada pegerakannya, jika algoritma LOOK *disk arm* tidak harus bergerak sampai ujung disk, tetapi hanya bergerak sampai pada permintaan terakhir.

Jika prinsip kerja setiap jenis algoritma penjadwalan hanya dikemukakan secara teori, memungkinkan proses pembelajaran dan pemahaman mengenai algoritma penjadwalan dalam sistem operasi ini

menjadi sulit dipahami, untuk itu penulis merancang sebuah program simulasi dengan menerapkan algoritma LOOK pada pergerakan kabin lift.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka dapat dirumuskan beberapa permasalahan diantaranya:

1. Prinsip kerja penjadwalan algoritma LOOK yang di kemukakan secara teori, memungkinkan proses pembelajaran tentang prinsip kerjanya menjadi sulit dipahami.
2. Algoritma LOOK sebagai penyempurnaan dari sistem kerja algoritma SCAN, sehingga terdapat persamaan dan perbedaan pada prinsip kerjanya.

1.3 Tujuan Penelitian

Tujuan dalam melakukan penelitian ini adalah merancang dan membuat sebuah program simulasi sebagai alat bantu untuk pembelajaran mengenai prinsip kerja algoritma penjadwalan LOOK.

1.4 Batasan Masalah

Penulis membatasi permasalahan yang ada dalam penelitian sebagai berikut :

1. Menerapkan algoritma penjadwalan LOOK pada sebuah program simulasi berupa pergerakan kabin lift.
2. Evaluasi program simulasi dilakukan dengan asumsi gedung enam lantai.
3. Evaluasi dilakukan dengan asumsi kondisi untuk satu kabin lift.
4. Simulasi lift ini dilengkapi dengan delapan tombol, diantaranya enam tombol sebagai penunjuk nomor lantai, tombol buka dan tutup pintu kabin lift di dalam kabin lift masing-masing, sedangkan pada dinding gedung (di luar lift) terdapat dua

tombol penunjuk arah permintaan bagi *user*.

5. Dalam simulasi ini belum mengevaluasi masalah perubahan kecepatan yang terjadi pada setiap kondisi gerak kabin lift (kondisi kecepatan pada saat kabin mulai bergerak dan pada saat kabin telah dekat pada lantai tujuan maka kecepatan gerak akan menurun, dan kabin akan bergerak cepat apabila posisi kabin masih jauh dari lantai tujuan).
6. Penelitian ini terfokus pada algoritma penjadwalan pengeksekusian permintaan, dan belum mengevaluasi tentang baban angkut, sensor dan lain sebagainya.
7. Aplikasi ini dirancang dan dikembangkan dengan aplikasi pemrograman yang berbasis obyek Pascal yaitu Borland Delphi 7 Lite Edition (*Personal Version Free*).

1.5 Tinjauan Pustaka

Penulis meninjau penelitian ini berdasarkan dari hasil-hasil penelitian yang telah ada sebelumnya. Latar belakang dari penelitian tugas akhir ini adalah berdasarkan hasil penelitian yang berkaitan dengan sistem kerja lift. Adapun penelitian-penelitian yang pernah dilakukan sebelumnya adalah sebagai berikut :

(**Adhia UI Wahyudi, 2006**) dalam tugas akhir dengan judul “Simulasi Lift Berbasis Mikrokontroler AT89S52 dan dapat Dikontrol Melalui PC”. (Tugas Akhir Jurusan Teknik Elektro Institut Teknologi Nasional tahun 2006).

Sedangkan (**R. Agung Linggaponti, 2007**) telah melakukan penelitian dalam tugas akhirnya yang berjudul “Perancangan Ladder Diagram Sistem Prototype Lift 3 Lantai Berbasis PLC OMRON CPM1A-3OCDR”. Dalam penelitian ini ditetapkan pada saat terjadi

emergency maka perintah reset akan mengembalikan posisi lift pada keadaan *default* lantai 1.

Lalu (Muhtar Ariadi, 2006) telah melakukan penelitian dalam Skripsi yang berjudul “Simulasi Sistem Kendali Pada Lift Berbasis PLC Siemens Logo Soft 230 RCL”. Dalam penelitian tersebut beliau merancang program untuk sistem kendali pada lift serta dilanjutkan dengan mengaplikasikannya ke dalam simulator. Perancangan simulasi ini dilakukan dalam tiga tahap. Pertama, perancangan *software* mulai dari pendeskripsian kerja dari sistem pada lift kemudian pengalamanan komponen-komponen yang digunakan ke dalam bahasa yang dimengerti oleh PLC, pembuatan *flowchart* dan *timing diagram*, selanjutnya masuk ke tahap pembuatan program pada komputer. Kedua, perancangan *hardware* yang dilakukan dengan melakukan pembuatan simulasi dalam bentuk miniatur. Kemudian tahap yang terakhir adalah penggabungan *software* dan *hardware* yakni dengan cara memasukkan program yang telah dibuat atau dirancang dalam komputer ke dalam simulator.

Kemudian (Albertus David Kho, 2003)^[1] dalam Tugas Akhir dengan judul “Simulasi Kontrol Lift Menggunakan PLC Berbasis *Fuzzy Logic*”. Dalam penelitian

tersebut digunakan PLC Omron C200HG sebagai perancangan sistem kendali lift. Lift yang digunakan adalah simulasi lift satu *shaft* yang terdiri dari 5 lantai. Sedangkan untuk mengontrol posisi dari lift digunakan *fuzzy logic* sebagai metode kontrol, sehingga sistem yang dibuat memiliki *settling time* yang tepat, *error* yang kecil, dan stabilitas yang baik.

Dari beberapa riset dalam jurnal di atas dapat ditarik kesimpulan bahwa dalam pembuatan simulasi lift diperlukan perhatian terhadap sistem pemilihan algoritma gerakan lift dan masalah *traffic* dari lift tersebut. Simulasi yang dibuat harus dapat mengkonfigurasi dan menentukan algoritma gerak lift yang sesuai kondisi permintaan.

Berdasarkan beberapa penelitian yang telah dipaparkan tersebut, perbandingan sistem yang dibangun oleh penulis dengan penelitian diatas memiliki konsep yang sama dengan Adhia UI Wahyudi, Muhtar Ariadi, dan Albertus David Kho, yaitu persamaan dalam pembuatan program simulasi, namun penulis lebih menekankan pada program simulasi yang menerapkan algoritma penjadwalan LOOK. Adapun hasil tinjauan pustaka yang telah dilakukan dapat dilihat pada Tabel 1.1.

Tabel 1.1 Matriks perbandingan Tinjauan Pustaka

Peneliti	Judul	Perbedaan				
		Program Simulasi	Prototype Hardware	PLC	Fuzzy Logic	Algoritma LOOK
Adhia UI Wahyudi	Simulasi Lift Berbasis Mikrokontroler AT89S52 dan dapat Dikontrol Melalui PC	✓	✓	-	-	-
R. Agung Linggaponti	Perancangan Ladder Diagram Sistem Prototype Lift 3 Lantai Berbasis PLC OMRON CPM1A-3OCDR	-	✓	✓	-	-
Muhtar Ariadi	Simulasi Sistem Kendali Pada Lift Berbasis PLC Siemens Logo Soft 230 RCL	✓	✓	✓	-	-
Albertus David Kho	Simulasi Kontrol Lift Menggunakan PLC Berbasis <i>Fuzzy Logic</i>	✓	-	✓	✓	-
Galih Pratiwi, Youllia Indrawaty, M. Ichwan	Pembangunan Program Simulasi Pergerakan Lift 6 Lantai dengan menerapkan Algoritma LOOK.	✓	-	-	-	✓

1.6 Cara Penelitian

Pada penelitian ini, penulis menggunakan beberapa metode penelitian, yaitu metode berorientasi objek (OOP) dan model *prototype*.

1.6.1 Metode Berorientasi Objek

Dalam metode berorientasi objek proses pengembangan perangkat lunak dapat dilihat sebagai urutan transformasi dimana *output* dari suatu transformasi menjadi masukan pada transformasi selanjutnya. Berikut tahapan transformasi :

Transformasi 1 (Analisis)

Menerjemahkan kebutuhan kedalam spesifikasi kebutuhan sistem (SRS-Syatem/ Software Requirement Specification). Spesifikasi kebutuhan sistem ini bersifat menangkap semua yang dibutuhkan sistem dan dapat terus

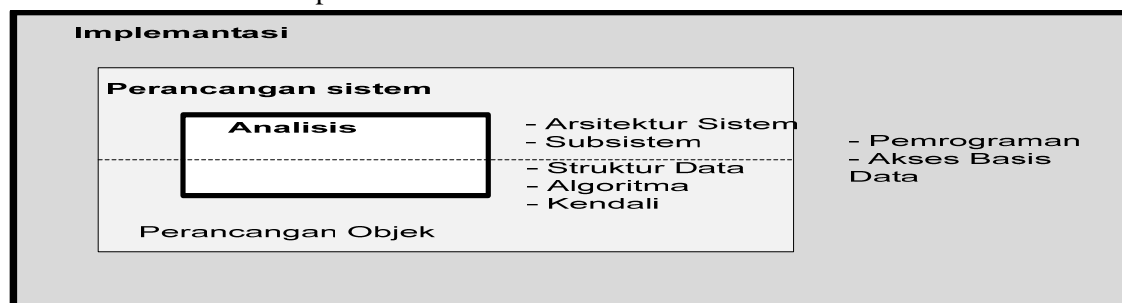
diperbaharui selama berjalannya pengembangan sistem.

Transformasi 2 (Perancangan)

Dimulai dengan pernyataan masalah dan diakhiri dengan rincian perancangan yang dapat ditransformasikan ke sistem operasional. Transformasi ini mencakup seluruh aktivitas pengembangan perangkat lunak, mencakup definisi-bagaimana mengembangkan perangkat lunak serta pengujiannya.

Transformasi 3 (Implementasi)

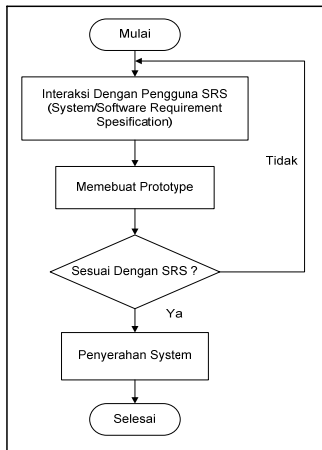
Melakukan penghalusan rincian perancangan ke penyebaran sistem sesuai dengan kebutuhan. Transformasi ini juga mencakup perancangan peralatan yang digunakan, prosedur-prosedur pengoperasian, deskripsi orang yang menggunakan sistem, dan lain sebagainya.



Gambar 1.1 Siklus Pengembangan Sistem Berorientasi Objek

1.6.2 Model Prototype

Dalam model pengembangan *prototype* ini program simulasi lift dapat dianalisis dan dikembangkan secara bertahap sehingga didapat program simulasi yang sesuai dengan yang dikehendaki. Dibawah ini merupakan gambar tahapan model pengembangan sistem *prototype*.



Gambar 1.2 Model *Prototype*

II. Teori Penunjang

2.1 Lift^[1]

Lift adalah suatu pesawat pengangkat yang mempunyai gerakan periodik untuk menaikkan atau menurunkan beban melalui *guide rail* yang vertikal. Ada beberapa jenis lift dan salah satunya adalah lift, lift digunakan untuk perpindahan orang atau meterial secara vertikal.

2.1.1 Komponen Lift

➤ **Kabin**

Tempat penyimpanan beban, berikut adalah syarat utama yang harus dimiliki kabin :

- Desain sederhana dan ringan
- Kabin tertutup dan aman
- Ventilasi yang cukup
- Kabin kuat dan kokoh

- Luas kabin standar dan sesuai dengan beban
- memiliki *safety door*.

➤ **Guide railsha**

Guide railsha merupakan jalan pergerakan kabin yang vertikal agar kabin tahan dari goncangan samping.

➤ **Shaftway (Hoistway)**

Shaftway merupakan lorong vertikal dari kabin atau *counterweight* yang bergerak vertikal pada saat *loading* ataupun *unloading*.

➤ **Counterweight**

Counterweight dalam lift memegang peranan penting karena berfungsi sebagai penyeimbang terhadap berat kabin, maka secara keseluruhan daya yang diperlukan untuk menggerakkan bisa ditekan, secara teoritis dapat dituliskan persamaan :

$$W = G_{cage} + Q$$

Dimana : W = Berat *Counterweight*

G_{cage} = Berat kabin kosong

Q = Berat penumpang maksimal yang diijinkan

➤ **Geared machine**

Digunakan sebagai reduksi putaran dari putaran motor hingga putaran *guide shave* yang mengangkat beban dengan kecepatan yang standar.

➤ **Hoisting machine**

Hoisting machine adalah penggerak utama pada lift dan merupakan kombinasi dari motor listrik, *guide shave* sebagai yang berfungsi sebagai pengangkat beban dengan putaran kabel.

➤ **Panel kontrol**

Panel kontrol ini merupakan *switch* yang bekerja secara otomatis serta sebagai pengaman elektrik sehingga lift dapat

bergerak sesuai ketentuan algoritma untuk setiap kondisi eksternal yang terjadi.

➤ **Peralatan pengaman**

Peralatan pengaman ini akan bekerja pada saat:

- Beban *overload*
- Kabel-kabel kendur
- Kecepatan gerak melebihi kecepatan gerak standar.

Semua alat pengaman lift ini dilengkapi dengan *switch* sehingga dapat bekerja dan memutuskan hubungan listrik secara langsung.

➤ **Governor**

Governor berfungsi untuk membatasi kecepatan kabin lift pada batas kecepatan standar.

➤ **Buffer**

Buffer berfungsi untuk mengurangi hentakan dan memperkecil kerusakan pada saat kabin *landing* pada lantai paling atas atau paling bawah.

➤ **Rem Listrik Magnetik**

Tujuan pemasangan rem adalah agar lift dapat berhenti sesuai poros motor penggerak *drive shave* untuk kabin yang akan dan selama berhenti.

2.1.2 Spesifikasi Teknis Pemakaian Lift

Lift memiliki beberapa karakteristik yang ditunjukkan oleh variabel-variabelnya, yang meliputi:

- **Kapasitas**
Menurut “Material Handling Equipment”, Rudenko, kapasitas lift penumpang berkisar antara 0,25 ton – 1,5 ton.
- **Kecepatan angkat**
Dalam standar kecepatan angkat dari lift adalah diambil dari harga kecepatan konstan yang dicapai dan dapat dilihat pada tabel 2.1.
Dari tabel di bawah ini dapat dilihat bahwa untuk suatu bangunan

bertingkat dengan jumlah lantai yang cukup banyak, maka dianjurkan memilih kecepatan yang lebih tinggi standar.

Tabel 2.1 Standar Kecepatan Lift Penumpang untuk Gedung Perkantoran

Floors	Passenger (fpm)	Service (fpm)
2 - 5	250 - 400	200
5 - 10	350 - 500	300
10 - 15	500 - 700	350 - 500
15 - 25	700 - 800	500
25 - 35	800 - 1000	500
35 - 45	1000 - 1200	700 - 800
45 - 60	1200 - 1600	800 - 1000
60 or more	1800	1000

1 fpm (*feet per minute*) = 0,505 cm/dt

Sumber : Adler, Rodney R, Vertikal Transportation for Buildings, New York

- **Stabilitas**

Instalasi lift melekat pada induk bangunan gedung yang ditempati sehingga stabilitas *frame* penyangga pada saat lift bekerja dapat stabil bergerak.

2.1.3 Cara Kerja Lift

Konstruksinya berupa kabin atau kereta yang dinaikturunkan oleh mesin traksi, dengan menggunakan tali baja tarik, melalui ruang luncur (*hoistway*) di dalam bangunan yang dibuat khusus untuk lift. Agar kereta lift tidak bergoyang digunakan rel pemandu setinggi ruang luncur (*hoistway*) yang diikat dengan tembok ruang luncur lift. Untuk mengimbangi berat kereta dan bebannya digunakan bandul pengimbang (*counterweight*), beratnya sama dengan berat kereta ditambah dengan setengah berat beban maksimum yang diizinkan. Hal ini untuk memperingan kerja mesin traksi, karena pada saat kereta dipenuhi dengan beban

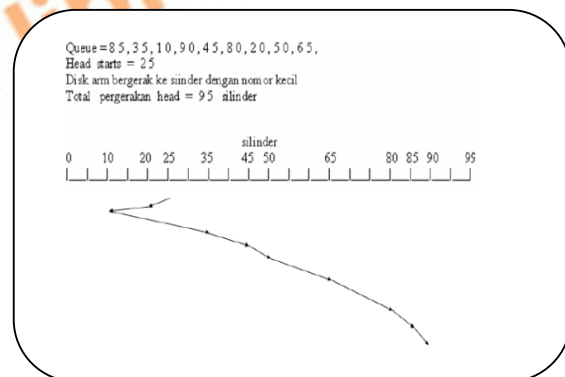
maksimum, mesin traksi hanya berupaya mengangkat atau menaikkan setengah dari beban maksimumnya. Sebaliknya pada saat kereta kosong, mesin traksi hanya perlu mengangkat atau menaikkan setengah dari beban maksimum yang berlebih pada *counterweight*.

Pada sistem geared atau gearless (yang masing-masing digunakan pada instalasi gedung dengan ketinggian menengah dan tinggi), kereta lift tergantung di ruang luncur oleh beberapa steel hoist ropes, biasanya dua puli katrol, dan sebuah bobot pengimbang (counterweight). Bobot kereta dan counterweight menghasilkan traksi yang memadai antara puli katrol dan hoist ropes sehingga puli katrol dapat menggegam hoist ropes dan bergerak serta menahan kereta tanpa selip berlebihan.

Mesin Lift Gearless adalah mesin untuk menggerakkan elevator terletak di ruang mesin yang biasanya tepat di atas ruang luncur kereta. Untuk memasok listrik ke kereta dan menerima sinyal listrik dari kereta ini, dipergunakan sebuah kabel listrik multiwire untuk menghubungkan ruang mesin dengan kereta. Ujung kabel yang terikat pada kereta turut bergerak dengan kereta sehingga disebut sebagai kabel bergerak (traveling cable) Jalur Lift (Hoistway) dan ruang mesin di atasnya. Mesin geared memiliki motor dengan kecepatan lebih tinggi dan drive sheave dihubungkan dengan poros motor melalui gigi-gigi di kotak gigi, yang dapat mengurangi kecepatan rotasi poros motor menjadi kecepatan drive-sheave rendah. Mesin gearless memiliki motor kecepatan rendah dan puli katrol penggerak dihubungkan langsung ke poros motor.

2.2 Algoritma Penjadwalan LOOK^[2]

Algoritma LOOK adalah algoritma penjadwalan disk yang secara konsep hampir sama dengan algoritma SCAN. Sesuai dengan namanya, algoritma ini seolah-olah seperti dapat "melihat". Algoritma ini memperbaiki kelemahan SCAN dan C-SCAN dengan cara melihat apakah di depan arah pergerakannya masih ada permintaan lagi atau tidak. Bedanya pada algoritma LOOK, *disk arm* tidak berjalan sampai ujung disk, tetapi hanya berjalan sampai pada permintaan yang paling dekat dengan ujung disk. Setelah melayani permintaan tersebut, *disk arm* akan berbalik arah dari arah pergerakannya yang pertama dan berjalan sambil melayani permintaan-permintaan yang ada di depannya sesuai dengan arah pergerakannya.



Gambar 2.1 Algoritma Penjadwalan LOOK

2.3. Thread^[2]

Thread merupakan cara dari komputer untuk menjalankan dua atau lebih *task* dalam waktu bersamaan, sedangkan *multithreading* adalah cara pengeksekusian yang mengizinkan beberapa *thread* terjadi dalam sebuah proses, saling berbagi sumber daya tetapi dapat dijalankan secara independen.

Keuntungan dari sistem yang menerapkan *multithreading* dapat kita kategorikan menjadi 4 bagian:

- a. Responsif. Aplikasi interaktif menjadi tetap responsif.
- b. Berbagi sumber daya. Beberapa *thread* yang melakukan proses yang sama akan berbagi sumber daya.
- c. Ekonomis. *thread* membagi memori dan sumber daya yang dimilikinya sehingga lebih ekonomis.

III. Analisis dan Perancangan Program Simulasi Lift

3.1 Analisis

Dalam pembangunan program simulasi lift penulis melakukan analisis terhadap beberapa kebutuhan meliputi:

Tabel 3.1 Analisis kebutuhan

No.	Nama Kebutuhan
1.	Pemilihan Metodologi
2.	Objek-objek pada Lift
3.	<i>Traffic</i> Lift
4.	Algoritma penjadwalan

3.1.1 Analisis Objek pada Lift

penulis memilih metodologi pengembangan sistem berorientasi objek. Karena jika menggunakan metodologi yang dikembangkan dengan Diagram Alir Data (DFD) kurang representatif untuk mempresentasikan masalah simulasi yang memiliki kondisi melibatkan objek-objek dan operasi-operasi ada dalam satu program aplikasi, dimana setiap objek dan operasi yang ada akan saling bertukar pesan satu sama lain dalam menjalankan program tersebut. Selain itu program simulasi ini berupa sistem interaktif, pada aksi-aksi tertentu program harus menjamin bahwa ia dapat melakukan tanggapan dalam selang waktu yang absolut.

3.1.2 Analisis Objek pada Lift

Dalam dunia nyata lift terdiri dari banyak komponen, namun penulis tidak akan menerapkan seluruh komponen lift pada program simulasi. Penulis melakukan analisis pada komponen lift berdasarkan fungsinya, kemudian mengambil beberapa objek yang dibutuhkan dalam pembangunan program simulasi ini.

Tabel 3.2 Objek (komponen lift)

No.	Nama Objek/ Komponen Lift	Fungsi
1.	Kabin	Tempat penyimpan beban angkut
2.	<i>Guide railsha</i>	jalannya pergerakan kabin
3.	<i>Shaftway</i> (<i>Hoistway</i>)	lorong vertikal dari kabin
4.	<i>Counterweight</i> (beban pengimbang)	penyeimbang terhadap berat kabin
5.	<i>Geared machine</i>	putaran motor
6.	<i>Hoisting machine</i>	pengangkat beban dengan putaran kabel.
7.	Kontrol Panel	<i>switch</i>
8.	Peralatan pengaman	pendukung pengamanan kabin
9.	<i>Governor</i>	Pembatas kecepatan
10.	<i>Buffer</i>	pengurang hentakan saat <i>landing</i>
11.	Rem ListrikMagnetik	Pemberhenti kabin sesuai poros

Berdasarkan fungsi masing komponen pada lift sebenarnya, maka penulis hanya mengambil beberapa komponen yang dibutuhkan dan menambah beberapa komponen lain, berikut adalah komponen objek yang digunakan dalam program simulasi lift ini, yaitu :

1. **Lorong (*Shaftway Hoistway*)** sebagai tempat kabin berada.
2. **Kabin** sebagai tempat penyimpanan beban angkut. Dalam kabin terdapat properti, yaitu :
 - Tombol dalam kabin (8tombol yaitu, 6tombol nomor lantai, tombol tutup pintu, tombol buka pintu)
 - Display arah gerak kabin
 - Display nomor lantai
 - Display status pintu
 - Pintu : terdiri dari pintu kanan dan pintu kiri (lebar pintu yang sama)
 - metode gerak pintu bergerak *center opening*
 - status pintu : open, close, on open, on close
3. **Counterweight (beban pengimbang)**
Beban pengimbang dilengkapi dengan motor penggerak, yaitu :
 - Tali pengangkut sebagai penghubung kabin dengan beban pengimbang
 - 2 buah roda katrol
4. **Tombol** pada setiap lantai, berfungsi sebagai penunjuk arah permintaan tujuan dari *user*, berfungsi juga sebagai pemberhenti kabin. Terdiri dari :
 - Tombol naik (panah atas)
 - Tombol turun (panah bawah)
5. **Konfigurasi**
Sebagai pengatur beberapa gerak lift, yaitu
 - Pengaturan kecepatan gerak pintu (*door speed*).
 - Pengaturan kecepatan gerak perpindahan kabin (*move speed/* kecepatan gerak naik turun kabin).

- Pengaturan lamanya waktu pintu terbuka (*open time*).

6. Tabel request

Tempat megumpulkan dan memanej setiap permintaan yang masuk

Metode : mengurutkan permintaan sesuai dengan urutan permintaan user, namun pengekseskuan permintaan tidak sesuai urutan permintaan, melainkan berdasarkan algoritma yang telah ditentukan.

3.1.3 Analisis *Traffic Lift*

Dalam perencanaan *traffic lift* digunakan arah gerak kabin lift sebagai prioritas utama dalam menentukan lantai tujuan kabin lift. Sehingga kabin lift akan bergerak ke lantai permintaan dengan syarat posisi lantai permintaan lebih besar (untuk arah naik), lebih kecil (untuk arah turun), dan arah permintaan sama dengan arah gerak kabin lift. Apabila ada permintaan yang arahnya sesuai namun posisi lantai lebih kecil (arah naik)/ lebih besar (arah turun), maka permintaan tersebut akan diabaikan terlebih dahulu sampai kabin lift berbalik arah dan mencapai kondisi syarat seperti yang telah dijelaskan. Apabila dalam semua kondisi syarat sudah terpenuhi maka kabin lift akan memilih bergerak ke lantai terdekat sebagai tujuannya.

Apabila semua kondisi diatas tidak terpenuhi kabin lift akan diam pada lantai dimana permintaan yang terakhir dieksekusi. Apabila ada permintaan yang searah dengan arah gerak kabin lift terakhir dan telah terlewati, maka lift akan mengecek lantai mana yang melakukan permintaan, kemudian lantai tersebut di setting sebagai *setting-point* (SP) dan arah kabin lift disesuaikan dengan *setting-point* dan *present-vallue*(VP). Apabila $SP > VP$, maka arah lift menjadi naik, dan apabila

SP < VP maka arah gerak kabin lift menjadi turun.

Apabila dalam perjalanan kabin lift menuju lantai permintaan yang diinginkan sejak awal, ada penekanan tombol, maka kabin lift akan langsung mengecek posisi permintaan yang diinginkan. Seandainya posisi yang baru diinginkan belum terlewati, maka kabin akan berhenti dahulu di lantai yang diminta tersebut.

3.1.4. Input dan Output Program

Berikut adalah *input* yang dimasukkan ke dalam program simulasi dan *output* yang dihasilkan oleh program simulasi :

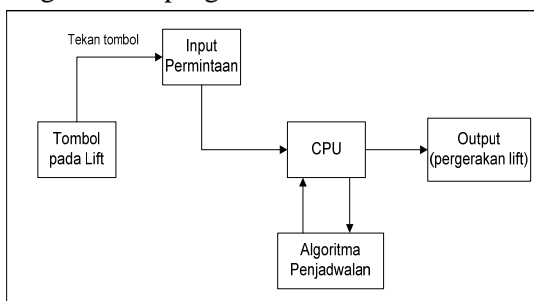
Tabel 3.3 Input dan Output Program Simulasi Lift

No.	Input	Output
1.	Data permintaan lantai dari tombol di dalam kabin lift.	<ul style="list-style-type: none"> Kabin lift akan berhenti dan membuka pintu untuk beberapa saat pada lantai yang diminta tersebut.
2.	Data permintaan naik/turun dari tombol di luar kabin.	<ul style="list-style-type: none"> Kabin lift akan berhenti dan membuka pintu untuk beberapa saat pada lantai dimana terdapat permintaan.
3.	Data permintaan tombol buka pintu dari dalam kabin.	<ul style="list-style-type: none"> Pintu kabin akan terbuka.
4.	Data permintaan tombol tutup pintu dari dalam kabin.	<ul style="list-style-type: none"> Pintu kabin akan tertutup.

3.2 Perancangan Program Simulasi Lift

3.2.1 Diagram Blok

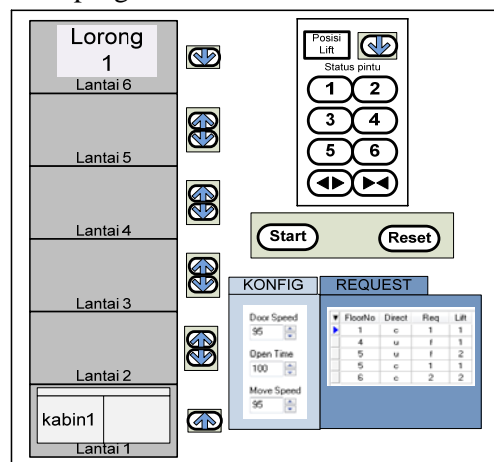
Berikut ini merupakan blok diagram dari program simulasi lift.



Gambar 3.1 Diagram Blok

3.2.2 Perancangan Program Simulasi Desain Form

Berikut ini merupakan gambaran form program simulasi lift.



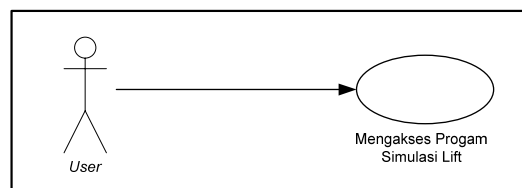
Gambar 3.2 Desain Form

3.2.3 UML (Unified Modelling Language)

Tahapan analisis sistem dengan menggunakan pemodelan UML (*Unified Modelling language*). UML merupakan pemodelan yang sangat tepat untuk menggambarkan perancangan sistem ini karena aplikasi ini dibangun dengan berorientasikan objek.

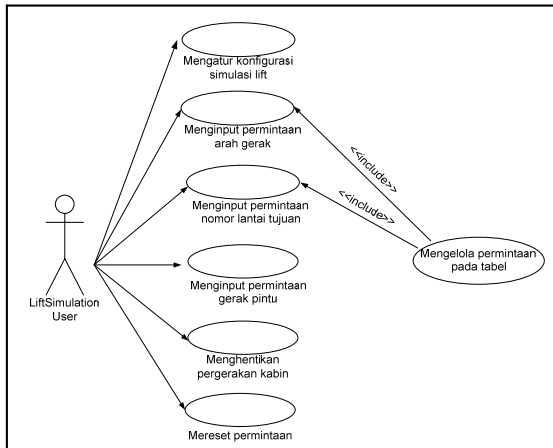
3.2.3.1. Use Case

Berikut ini merupakan gambaran proses model pengembangan program simulasi yang dimodelkan dalam bentuk *use case*.



Gambar 3.3 Diagram Use Case Program simulasi Lift

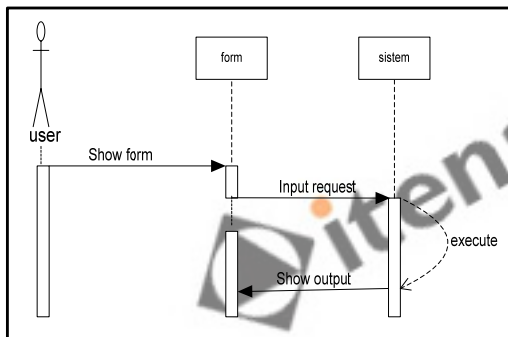
Adapun proses yang terjadi didalam proses pada program simulasi lift ini akan dijelaskan melalui diagram *use case* yang merupakan *break down* dari proses Diagram Use Case Program simulasi Lift seperti pada Gambar 3.4.



Gambar 3.4 breakdown Diagram Use Case Program simulasi Lift

3.2.3.2. Sequence Diagram

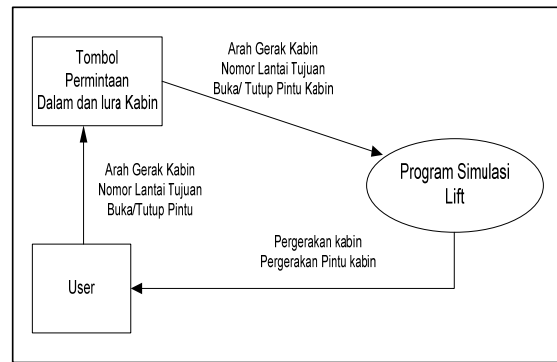
Sequence Diagram yang terbentuk pada program simulasi memperlihatkan event-event yang berurutan sepanjang berjalannya waktu.



Gambar 3.5 Sequence Diagram Program simulasi Lift

3.2.3.3. DAD

Diagram Aliran Data dirancang untuk memperlihatkan ketergantungan fungsional. Aliran-aliran pada model fungsional berhubungan langsung dengan nilai objek atau atribut pada diagram objek. Berikut ini adalah DAD untuk program simulasi lift yang dibangun :



Gambar 3.6 Diagram Konteks Program Simulasi Lift

3.2.4. Flowchart Pergerakan

Kabin pada Program Simulasi Rancangan flowchart program berfungsi untuk mengetahui alur kerja dari program. Berikut ini adalah Flowchart pergerakan kabin lift pada dari program simulasi.



Gambar 3.7 flowchart global pergerakan Kabin

3.2.5. Proses Threading

Pada program simulasi lift, dibutuhkan *threding* untuk menangani proses-proses yang harus berjalan pada waktu yang sama. Misalnya pada saat kabin sedang bergerak naik atau turun untuk mengeksekusi permintaan, di waktu yang program harus merekam semua permintaan dari tombol-tombol yang

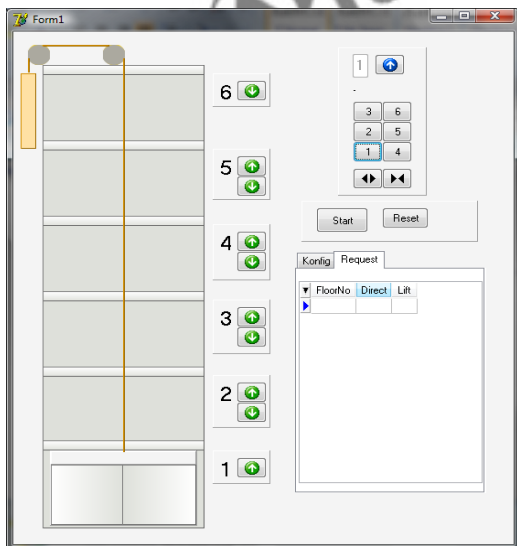
ditekan, kemudian mengurutkan data tersebut dan menampilkannya pada tabel *request* dan selanjutnya setiap data permintaan tersebut akan disortir untuk menentukan permintaan mana yang akan dieksekusi oleh program simulasi lift ini.

IV. Pengujian Program Simulasi Lift

Pengujian program simulasi lift ini menggunakan *black-box*. Pengujian *black-box* ini berfokus pada persyaratan fungsional program simulasi. Pengujian ini memungkinkan analisis program memperoleh kumpulan kondisi input yang akan mengerjakan seluruh kumpulan fungsional program.

4.1 Hasil Pengujian

Pada simulasi lift yang telah dirancang ini, dilakukan pengujian pada setiap modulnya. Pengujian dilakukan untuk mengetahui apakah program simulasi lift ini telah sesuai dengan keinginan. Sebelum melakukan pengujian program simulasi harus ditampilkan, berikut tampilan awal dari program simulasi lift.



Gambar 4.1 Tampilan Awal program simulasi

Berikut ini akan ditampilkan beberapa tabel pengujian yang dilakukan dengan metode pengujian *black-box* serta hasil dari pengujian tersebut.

4.1.1 Pengujian Inisialisasi Awal

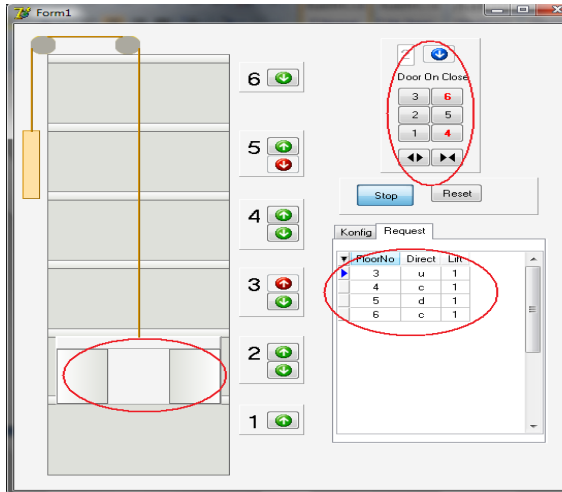
Tabel 4.1 pengujian inisialisasi awal

Kasus dan Hasil Uji			
Hal yang diuji	Harapan	pengamatan	kesimpulan
Semua objek dalam form	Semua objek tampil di form	Semua objek muncul di form	sukses

4.1.2 Pengujian Verifikasi Tombol Dalam Kabin

Tabel 4.2 pengujian properti dalam kabin

Kasus dan Hasil Uji			
yang diuji	Harapan	pengamatan	kesimpulan
Penambahan permintaan pada tabel	Saat tombol permintaan dalam kabin ditekan, maka permintaan langsung masuk tabel antrian permintaan	Penambahan permintaan pada tabel	sukses
Tombol buka/tutup pintu kabin	Pintu membuka dan menutup sesuai permintaan selama memenuhi syarat	Buka dan tutup pintu kabin sesuai permintaan dan memenuhi syarat	sukses
Display dalam kabin	Arah, status pintu, no lantai	Ketiganya tampil	sukses



Gambar 4.2 Hasil uji properti dalam kabin

4.1.3 Pengujian Verifikasi Tombol Luar Kabin

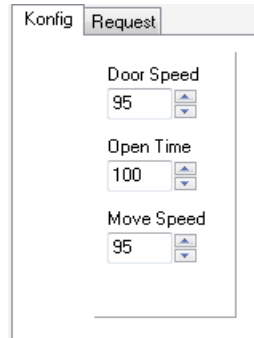
Tabel 4.3 pengujian properti dalam kabin

Kasus dan Hasil Uji			
Hal yang diuji	Harapan	pengamatan	kesimpulan
Tombol di luar kabin	Tombol luar kabin ditekan masuk ke tabel permintaan	Permintaan dari luar kabin masuk dalam antrian tabel permintaan	sukses

4.1.4 Pengujian Verifikasi Konfig

Tabel 4.4 pengujian properti dalam kabin

Kasus dan Hasil Uji			
Hal yang diuji	Harapan	pengamatan	kesimpulan
Pengaturan konfigurasi	Kecepatan gerakan kabin bisa berubah	Kecepatan gerakan berubah	sukses



Gambar 4.4 konfigurasi pengaturan kecepatan gerak

4.1.5 Pengujian Verifikasi Request

Tabel 4.5 pengujian tabel permintaan

Kasus dan Hasil Uji			
Hal yang diuji	Harapan	pengamatan	kesimpulan
Tabel Request	Antarian dalam tabel bertambah jika ditekan tombol, dan berkurang bila permintaan telah dieksekusi	Antrian permintaan dalam tabel bertambah dan berkurang jika telah dieksekusi	sukses

4.1.6 Pengujian Pergerakan Kabin

• **Asumsi Kasus**

Sebelum menguji keberhasilan penerapan algoritma look pada program simulasi, ada beberapa istilah yang pada penjadwalan sistem operasi dengan algoritma pada simulasi lift yang akan diasumsikan sama, yaitu :

- *Head start* diasumsikan sebagai posisi awal kabin.
- *Queue* sebagai antrian permintaan *user*.
- Pada sistem operasi pergerakan awal ditentukan oleh posisi silinder lebih kecil sedangkan pada simulasi lift arah gerakan ditentukan *Set Move Direction*. Pergerakan kabin pada simulasi lift ini dapat dilihat dengan

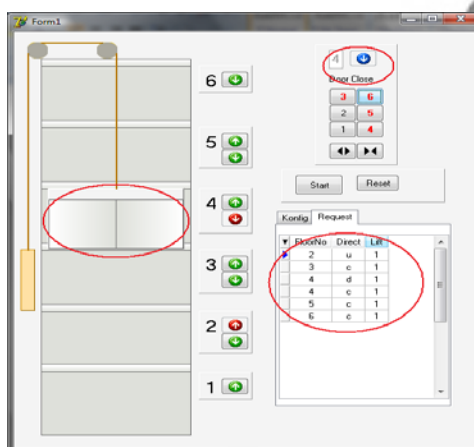
menyertakan contoh kasus berikut ini :

A. Percobaan Kasus 1

- **Asumsi awal**
 - Posisi kabin berada di lantai 4.
 - Arah gerak terakhir (Set MoveDirection) ke bawah.
 - Status pintu "Door Close".
 - Permintaan : 3, 2 atas, 4, 5, 4 bawah, 6

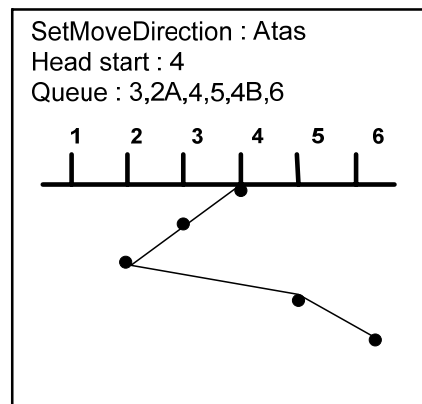
FloorNo	Direct	Req	Lift
2	u	f	1
3	c	1	1
4	d	f	1
4	c	1	1
5	c	1	1
6	c	1	1

Gambar 4.5 Asumsi permintaan program simulasi



Gambar 4.6 Asumsi Awal kasus pada program simulasi

- **Pengeksekusian permintaan oleh program simulasi**



Gambar 4.7 Pergerakan Kabin kasus 1 berdasarkan program dengan Algoritma LOOK

Gambar 4.7 merupakan kurva yang mewakili pergerakan kabin lift pada kasus 1 program simulasi lift dengan menggunakan algoritma penjadwalan LOOK. Terlihat pada gambar bahwa kabin lift akan mengeksekusi permintaan di lantai 4, karena *setmovedirection* ke arah bawah maka kabin akan bergerak turun ke lantai 3, kemudian ke lantai 2, setelah permintaan ke arah bawah habis, maka kabin lift berbalik arah gerak untuk mengeksekusi permintaan dari lantai 5 dan lantai 6. Berikut ini merupakan tabel pengeksekusian permintaan.

Tabel 4.6 Pengeksekusian permintaan pada kasus 1 dengan algoritma LOOK

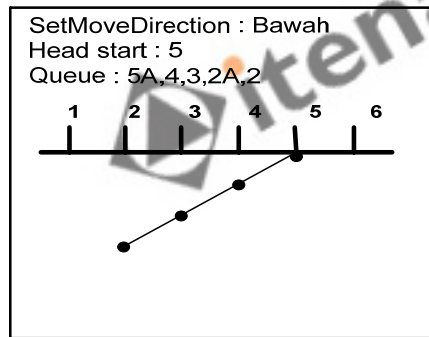
Next Floor Accesed	Number of Floor Traversed
4	0
3	1
2	1
5	3
6	1
Total	6

Pada tabel 4.6 dapat dilihat nomor lantai yang dieksekusi program jumlah lantai yang dilewati kabin yaitu 6 lantai.

B. Percobaan Kasus 2

- **Asumsi awal**
 - Posisi kabin berada di lantai 2.
 - Arah gerak terakhir (Set MoveDirection) ke atas.
 - Status pintu “Door Close”.
 - Permintaan : 5 bawah, 4, 3, 2 atas,2
- **Pengeksekusian permintaan oleh program simulasi**

Gambar 4.8 merupakan kurva yang mewakili pergerakan kabin lift pada kasus 2 program simulasi lift dengan menggunakan algoritma penjadwalan LOOK. Terlihat pada gambar bahwa kabin lift akan mengeksekusi permintaan di lantai 5, karena *setmovedirection* ke arah bawah maka kabin akan bergerak turun ke lantai 4, kemudian ke lantai 3, dan ke lantai 2.



Gambar 4.8 Pergerakan Kabin kasus 2 berdasarkan program dengan Algoritma LOOK

tabel 4.7 dapat dilihat nomor lantai yang dieksekusi program jumlah lantai yang dilewati kabin yaitu 4 lantai.

V. Kesimpulan

Berdasarkan hasil pengujian pada penelitian yang telah dilakukan, dapat dihasilkan kesimpulan, yaitu :

1. Dengan adanya perancangan dan pembangunan program simulasi lift ini dapat membantu proses pembelajaran mengenai algoritma penjadwalan LOOK yang ada pada sistem operasi komputer.
2. Algoritma LOOK telah berhasil diterapkan pada program simulasi lift ini. Dapat dilihat pada pengujian dengan beberapa kasus, dimana kabin lift akan bergerak sampai ke lantai terujung jika disana terdapat permintaan (kasus 1). Apabila tidak ada permintaan di lantai terujung tersebut, maka kabin tidak akan bergerak ke sana(kasus 2).

VI. Daftar Pustaka

[1] David Kho, Albertus, 2003, “Simulasi Kontrol Lift Menggunakan PLC Berbasis Fuzzy Logic”, Universitas Kristen Petra.

[2] Avi Silberschatz, Peter Galvin, dan Rag Gagne. 2005. [Silberschatz2005] *Operating Systems Concepts*. Seventh Edition. John Wiley & Sons.

[3] Nugroho, Adi, Agustus 2004, “Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorintasi Objek”, Informatika, Bandung.

[4] Pressman, Roger S., 1997, *Software Engineering: A Practitioner's Approach*, McGraw-Hill.

