

RANCANG BANGUN SIMULATOR SISTEM KONTROL ELEVATOR 3 LANTAI

Zaiyan Ahyadi ¹⁾, Sarifudin ²⁾,
Ivan Maududy ³⁾

z.ahyadi@poliban.ac.id¹⁾, sarif.poliban@gmail.com²⁾, hidayat.ivan@gmail.com³⁾

1, 2,3) Teknik Elektronika, Politeknik Negeri Banjarmasin

Abstrak

Sistem elevator merupakan suatu sistem kontrol digital yang kompleks, sehingga menjadi obyek yang menarik baik dalam pembelajaran praktikum kontrol dan sistem digital. Namun suatu sistem elevator sangat mahal harganya meskipun dalam bentuk miniatur. Selain itu referensi tentang cara kerja elevator yang menjelaskan sampai detail pada tahap hardware masih sedikit. Penelitian ini bertujuan untuk merancang bangun simulator sistem kontroller elevator 3 lantai berbasis program visual. Plant system elevator dan kontrollernya berupa simulator yang dibuat menggunakan program Delphi. Sistem elevator yang dibuat mengikuti cara kerja peralatan modul praktikum elevator 3 lantai yang ada pada laboratorium Sistem Kendali. Indikator dari penelitian ini adalah bekerjanya sistem elevator hasil rancangan dengan baik. Kelebihan dari simulator yang dibuat pada penelitian ini antara lain kecepatan car elevator mempunyai dua kecepatan, visualisasi yang menarik dan status sensor dan sinyal kendali yang real time dan logika kontrol yang modular terbasusun atas beberapa blok. Hasil penelitian ini memperlihatkan bahwa simulator sistem kontrol elevator berjalan dengan baik sesuai yang diharapkan.

Kata Kunci : Elevator, Sistem Kontrol, Simulator

1. PENDAHULUAN

Sistem elevator adalah merupakan sistem yang kompleks karena melibatkan banyak sensor, indikator dan sinyal kontrol. Untuk pembelajaran biasanya digunakan Model Elevator yang harganya tidak murah. Penelitian ini bertujuan untuk membuat simulator sistem elevator dengan menggunakan program Delphi agar mudah digunakan dan murah.

Penulis sampai saat ini belum menemukan simulator elevator yang mempunyai semua komponen yang harusnya terdapat pada sistem elevator sesungguhnya. Pada umumnya model dan simulator elevator menyederhanakan sistem dan fokus pada bagian tertentu pada sistem tersebut. penulis dalam pembuatan Simulator Elevator dengan menggunakan program Delphi. Pada studi pustaka, sebagai bahan perbandingan awal penulis merangkum beberapa publikasi penelitian tentang model atau simulator elevator dengan beberapa kekurangannya.

Pada penelitian [1], model elevator yang dibuat sangat sederhana dengan menggunakan bandul yang bergerak naik turun digerakkan oleh motor stepper. Sensor posisi car menggunakan sensor infra merah dengan kontroller menggunakan AT89S52. Sinyal-sinyal yang terdapat pada model ini sangat sedikit dan fokus utama adalah mengontrol posisi car. Pada penelitian [2], fokus utama adalah membuat chip yang berfungsi sebagai pengontrol sistem elevator. Namun sistem elevator tidak mencakup kontrol kecepatan ketika car akan berhenti.

Pada penelitian [3], kontroller sistem elevator menggunakan PLC, simulasi menggunakan program visual, namun tidak ada kontrol kecepatan motor. Pada penelitian [4], model sistem elevator sangat sederhana dengan penggerak motor stepper, tanpa kontrol kecepatan dan pintu. Juga tidak terdapat indikator.

Pada penelitian [5], model elevator 8 lantai berupa simulator dengan menggunakan Delphi. Tampilan bagus dan interaktif, namun tidak ada kontrol kecepatan motor dan sinyal indikator dan

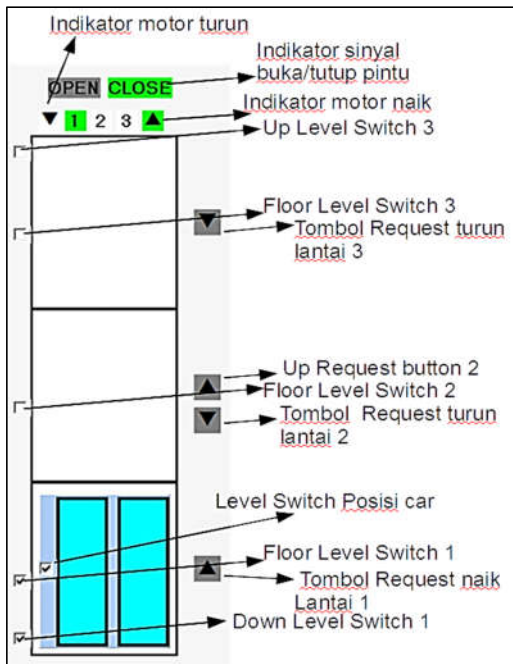
sensor posisi yang kurang lengkap. Fokus penelitian adalah teknik pemrograman controller dengan berbasiskan *statechart*.

Dari beberapa publikasi penelitian di atas tidak ada yang memodelkan simulator secara lengkap seperti aslinya. Semuanya tidak menggunakan kontrol kecepatan motor penggerak, padahal dalam sistem elevator sebenarnya kecepatan motor akan berkurang ketika akan berhenti, sehingga tidak menyebabkan car berhenti mendadak. Selain itu juga pada umumnya fokus penelitian hanya pada kontrol posisi tanpa melakukan kontrol buka tutup pintu elevator.

Elevator trainer model ED-4000E [6] merupakan model terlengkap karena merupakan miniatur dari sistem elevator 3 lantai yang sesungguhnya. Penelitian ini berdasarkan pada *model trainer elevator* tersebut.

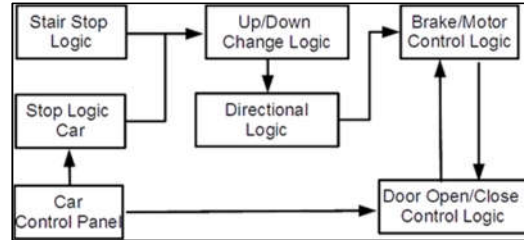
2. METODE PENELITIAN

Suatu sistem Elevator dapat dibedakan menjadi 3 bagian yaitu lantai bangunan (*stair*), kotak elevator (*car*) yang bergerak naik turun, dan sistem kontrol. Pada bagian *stairs* terdapat tombol permintaan (*request*) dan lampu indikator serta sensor-sensor. Pada bagian *car* terdapat panel kontrol dan lampu indikator. Sinyal sensor dan indikator pada bagian *stair* dapat dilihat pada Gambar 1.



Gambar 1. Sensor dan Indikator Sistem Elevator

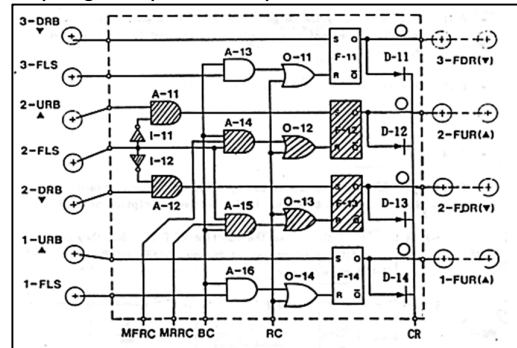
Bagian kontrol terdiri dari rangkaian logika, yang dalam penelitian ini diganti dengan program komputer. Rangkaian Controller elevator dapat dibagi menjadi 7 bagian seperti yang dilihat pada blok diagram Gambar 2.



Gambar 2. Blok Diagram Bubungan Seluruh Bagian Kontrol

2.1 Stair Stop Logic

Rangkaian *stair stop logic* merupakan bagian yang berfungsi untuk mengontrol car elevator harus berhenti karena adanya permintaan (*request*) pada lantai bangunan (kotak *elevator*). Permintaan berhenti terjadi karena penekanan tombol *request* (DRB3, URB2, DRB2 dan URB1) yang terdapat pada dinding kotak elevator. Rangkaian digital *Stair stop logic* dapat dilihat pada Gambar 3.

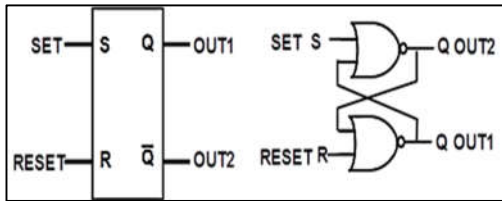


Gambar 3. Rangkaian Stairs Stop Logic

Dari rangkaian tersebut dibuat program Delphi seperti berikut ini

```

DRB3 := DRB3_CP.checked;
FLS3 := FLS3_CP.checked;
URB2 := URB2_CP.checked;
DRB2 := DRB2_CP.checked;
FLS2 := FLS2_CP.checked;
DRB2 := DRB2_CP.Checked;
URB1 := URB1_CP.checked;
FLS1 := FLS1_CP.checked;
setF11:= DRB3 ;
resF11:= (FLS3_ and BC_) or RC_ ;
FDR3 := rsff(setF11,resF11,FDR3_);
setF12:= (URB2_ and not FLS2_);
resF12:= (BC_ and MFR_ and FLS2_) or RC_ ;
FUR2 := rsff(setF12,resF12,FUR2_);
setF13:= (DRB2_ and not FLS2_);
resF13:= (FLS2_ and MRRC_ and BC_) or RC_ ;
FDR2 := rsff(setF13,resF13,FDR2_);
setF14:= URB1 ;
resF14:= (BC_ and FLS1_) or RC_ ;
FUR1 := rsff(setF14,resF14,FUR1_);
CR := FDR3_ or FUR2_ or FDR2_ or FUR1 ;
    
```



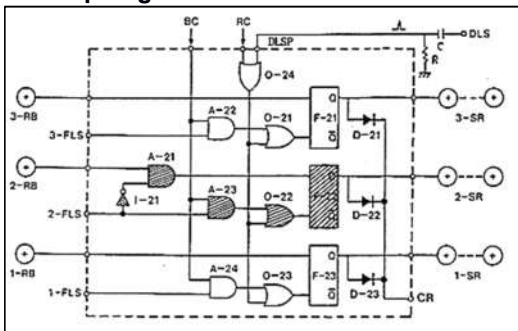
Gambar 4. RS Flip flop dan rangkaian internal dari gerbang NOR

Implementasi RS Flip flop pada Gambar 4 pada program Delphi adalah dengan membuat dua buah fungsi untuk dua output (Q dan not Q) yaitu sebagai berikut :

```
function
rsff (inS:boolean;inR:boolean;outQ:boolean)
:boolean;
var
outQ_:boolean;
begin
outQ_:= not(inS or outQ);
outQ:= not(inR or outQ_);
result := outQ;
end;

function
_rsff (inS:boolean;inR:boolean;outQ_:boolean)
:boolean;
var
outQ_:boolean;
begin
outQ_:= not(inR or outQ_);
outQ := not(inS or outQ);
result := outQ_;
end;
```

2.2 Stop Logic Car



Gambar 5. Rangkaian Stop logic of Car

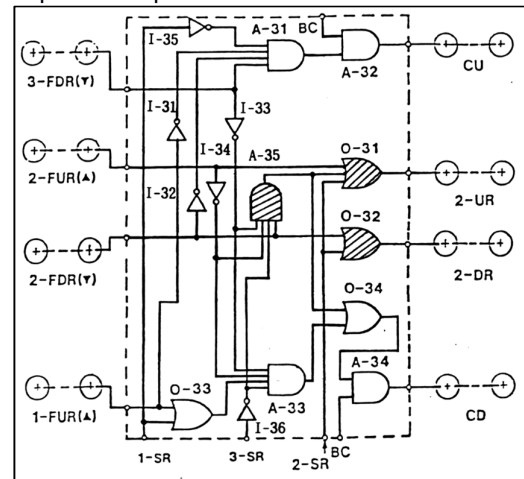
Stop logic Car adalah rangkaian yang berfungsi untuk membuat car elevator stop berdasarkan penekanan tombol permintaan yang terdapat pada car elevator. Tombol ini terletak di dalam car yang ditekan setelah pengguna elevator masuk ke dalam car dan menekan nomor lantai tujuan. Implementasi Rangkaian Stop Logic car dapat dilihat pada Gambar 5 pada program Delphi dituliskan sebagai berikut :

```
setF21 := RB3 ;
resF21 := (FLS3_ and BC_) or RC ;
SR3_ := rsff (setF21, resF21, SR3_);
setF22 := RB2_ and not FLS2_ ;
resF22 := (BC_ and FLS2_) or RC ;
```

```
SR2_ := rsff (setF22, resF22, SR2_);
setF23 := RB1 ;
resF23 := (FLS1_ and BC_) or RC_ ;
SR1_ := rsff (setF23, resF23, SR1_);
CR_ := SR1 or SR2 or SR3 or CR ;
```

2.3 Up / Down Change Logic

Up/ Down Change Logic merupakan rangkaian yang berfungsi untuk mengatur perubahan arah motor penggerak car elevator. Sinyal out dari rangkaian ini belum berupa sinyal Change Up (CU) yang mengubah arah motor menjadi arah naik dan mengubah arah menjadi arah turun Change Down (CD). Rangkaian logika untuk blok ini dapat dilihat pada Gambar 6.



Gambar 6. Rangkaian Up/Down Change Logic

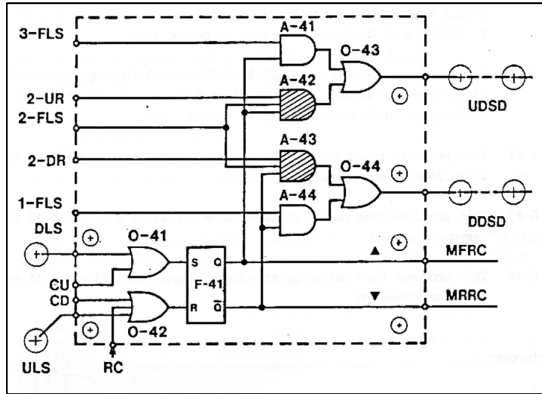
Implementasi rangkaian Up/Down Change Logic pada Delphi adalah sebagai berikut :

```
CU_ := (not SR1_ and not FUR1_ and not
FDR2_ and FDR3_) and BC ;
A35 := (not FDR3_ and not FUR2_ and not
SR3_ and FDR2_);
UR2 := FUR2_ or A35 or SR2_ ;
DR2_ := FDR2_ or SR2_ ;
CD_ := (A35 or (not FDR3_ and not FUR2_
and (FUR1_ or SR1_) and
not SR3_)) and BC ;
```

2.4 Rangkaian Down Speed and Directional

Rangkaian ini berfungsi untuk mengatur arah motor dan sinyal untuk memperlambat atau mempercepat putaran motor. Rangkaian ini dapat dilihat pada Gambar 7.

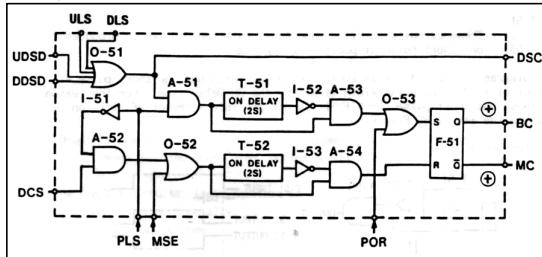
```
FLS1_:=FLS1_CP.Checked;
FLS2_:=FLS2_CP.Checked;
FLS3_:=FLS3_CP.Checked;
setF41 := DLS_ or CU_ ;
resF41 := CD_ or ULS_ or RC_ ;
MFRC_ := rsff (setF41, resF41, MFRC_);
MRRC_ := _rsff (setF41, resF41, MRRC_);
UDSD_ := (FLS3_ and MFRC_) or (UR2_ and
FLS2_ and MFRC_);
DDSD_ := (DR2_ and FLS2_ and MRRC_) or
S (FLS1_ and MRRC_);
```



Gambar 7. Down Speed and Directional Logic Circuit

2.5 Brake/Motor Control Logic

Blok ini berfungsi untuk mengatur motor untuk berhenti atau bekerja dan juga untuk mengatur kecepatan. Rangkaian dapat dilihat pada Gambar 8.



Gambar 8. Brake / Motor Kontrol Logic Circuit

Implementasi program pada Delphi adalah sebagai berikut

```

PLS_:=PLS_CP.checked;
QA51:= PLS_ and (DLS_ or ULS_ or UDSD_ or
DDSD_);
QO52:= (not PLS_ and DCS_) or MSE_;
T51.enabled:=QA51;
QT51:=QT51 and T51.Enabled;
T52.enabled:=QO52;
QT52:=QT52 and T52.Enabled;
QA53:= (not QT51 and QA51);
QA54:= (not QT52 and QO52);
setF51 := QA53 or POR_;
resF51 := QA54;
DSC_:=ULS_ or DLS_ or UDSD_ or DDSD_;
BC_ := rsff(setF51,resF51,BC_);
MC_ := rsff(setF51,resF51,MC_);
    
```

Pada rangkaian dapat dilihat terdapat komponen *Timer On Delay*. Komponen ini berfungsi untuk menghasilkan sinyal ON yang tertunda ketika timer mulai diaktifkan. . Timer akan aktif jika sinyal input berlogika 1 dan output akan ikut bernilai 1 setelah 2 detik. Dan ketika timer mati maka output akan bernilai logika 0 bersamaan dengan timer tidak aktif lagi. Pada gambar terlihat tundaan selama 2 detik.

Implementasi *timer On Delay* pada Delphi dapat menggunakan komponen *Timer* dimana pengaktifan dihubungkan dengan sinyal input. Kemudian pada event onTimer diisi baris perintah yang menyebabkan output timer menjadi logika 1. *Interval timer* harus diisi dengan nilai 2000 (untuk tundaan 2 detik). Output *timer* kemudian di-and-kan dengan input timer. Berikut adalah cuplikan program untuk T-51:

```

T51.enabled :=QA51;
QT51 := QT51 and T51.Enabled;
    
```

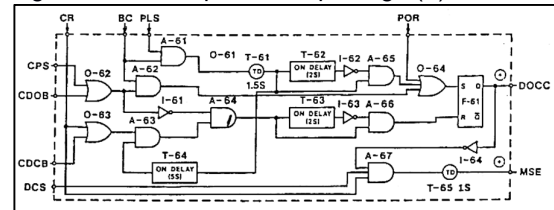
Dua baris program di atas diletakkan pada bagian timer yang memanggil aksi kontrol. Untuk prosedur event timer dituliskan sebagai berikut :

```

Procedure TForm1.T51Timer(Sender: TObject);
begin
    QT51:=true;
end;
    
```

2.6 Door Open Close Control logic Circuit Rangkaian Door Open/Close

Control Logic berfungsi untuk mengatur buka tutup pintu car elevator. Rangkaian logika blok ini dapat dilihat pada gb (9)



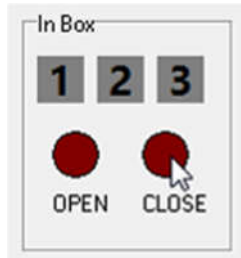
Gambar 9 Rangkaian Door Open/Close Control Logic

```

QA61 := PLS_ and BC_ ;
QA62 := BC_ and (CPS_ or CDOB_);
QA63 := (CR_ or CDCB_) and QT64;
T64.enabled:= QT61;
QT64 := QT64 and T64.enabled;
QA64 := not (CPS_ or CDOB_) and QA63;
T62.enabled:=QT61;
QT62 := QT62 and T62.enabled;
T63.enabled:=QA64;
QT63 := QT63 and T63.enabled;
QA65 := not QT62 and QA61;
QA66 := not QT63 and QA64;
setF61:=POR_ or QA65 or QA62;
resF61:= QA66;
DOCC_ := rsff(setF61,resF61,DOCC_);
QA67 := not DOCC_ and DCS_ and CR_;
MSE_ := QT65;
    
```

2.7 Car Control panel

Panel tombol yang terdapat pada car elevator sebenarnya merupakan bagian dari panel kontrol, dan merupakan blok kontrol Stop Logic Car. Namun pada simulator ini memisahkan tampilan panel tombol ini terpisah dari yang lain, seperti yang ditunjukkan pada Gambar 10.



Gambar 10 Car control panel

Terdapat 3 buah tombol *Floor Request* sesuai dengan jumlah lantai dan 2 buah tombol untuk *Open* dan *Close door*. Implementasi tombol *Floor Request* pada Delphi adalah dengan komponen Label seperti tombol pada kotak elevator. Sedangkan tombol *Open / Close Door* diimplementasikan dengan komponen shape. Program pembacaan *tombol Open / Close Door* menggunakan prosedur event *mouseDown* dan *mouseUp*.

```

procedure
TForm1.closeBshapeMouseDown(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
closeBshape.Brush.Color:=clred;
CDCB_:=true;
end;

procedure
TForm1.closeBshapeMouseUp(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
closeBshape.Brush.Color:=clmaroon;
CDCB_:=false;
end;
    
```

Untuk tombol *Floor request* juga menggunakan prosedur *event mouseDown* dan *mouseUp* sebagai berikut :

```

procedure TForm1.RB1LabelMouseDown(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
RB1 := true;
end;

procedure TForm1.RB1LabelMouseUp(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
RB1:= False;
end;
    
```

Prosedur event untuk tombol *Floor Request* tidak ada perintah untuk mengubah warna komponen label. Perubahan warna dilakukan pada blok *loop (Timer)* kontrol. Karena perubahan warna harus dipastikan bahwa request telah terbaca oleh panel kontrol.

2.8 Sinyal Transfer

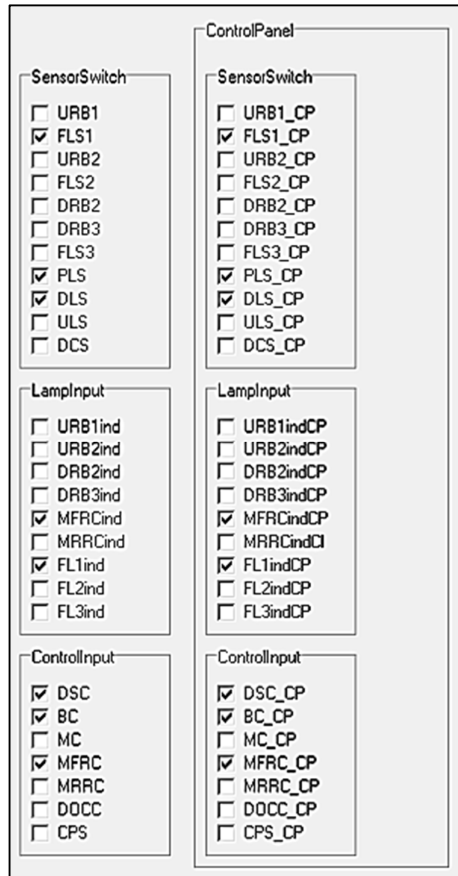
Simulator yang dirancang juga memperlihatkan kondisi sinyal-sinyal yang ditransfer antara kotak elevator dengan panel kontrol, seperti yang terlihat pada Gambar 11 berikut. Pada gambar dapat dilihat implementasi sinyal adalah berupa komponen checkbox dengan memanfaatkan property *checked* yang bertipe Boolean. Ini sesuai dengan kondisi sinyal yang mempunyai nilai logika 0 atau 1.

Sinyal terbagi menjadi 2 kolom. Kolom sebelah kiri memperlihatkan sinyal yang terdapat pada Port di kotak elevator, sedangkan kolom kiri mewakili sinyal yang terdapat pada Port kontrol Panel. Pemisahan ini digunakan untuk mempermudah pengelolaan sinyal untuk tahap selanjutnya, dimana panel kontrol bukan terdapat pada program Delphi, melainkan peripheral eksternal yang terhubung dengan computer seperti mikrokontroller.

Setiap kolom juga terlihat kumpulan sinyal terbagi menjadi tiga kelompok, hal ini didasarkan pada buku manual teknis Elevator Trainer yang menjadi panduan utama penelitian ini, yaitu kelompok sinyal sensor dari Kotak Elevator ke Panel Kontrol, kelompok sinyal indikator yang dikirim dari Panel Kontrol ke kotak Elevator dan kelompok sinyal kontrol dari Pnel Kontrol ke Kotak Elevator.

Pada perancangan simulator yang masih menjadi satu antara kotak Elevator dengan Panel kontrol, sinyal dari kotak Elevator ke Panel Kontrol di implementasikan dengan menuliskan variable kepunyaan Panel Kontrol di sebelah kanan tanda ':=', sedangkan variabel kepunyaan Kotak Elevator.

Adapun sinyal dituliskan dengan singkatan yang melambagnkan nama sinyal tersebut. Berikut adalah keterangan sinyal pada Gambar 11.



Gambar 11. Sinyal transfer antara kotak elevator dengan panel kontrol

Tabel 1. Kelompok sensor switch

ID	Deskripsi
URB1	Up Request Button1= tombol request untuk naik yang terletak di lantai 1
FLS1	Floor Level Switch 1 = sensor level switch yang terletak pada lantai 1
URB2	Up Request Button 2= tombol request untuk naik yang terletak di lantai 2
FLS2	Floor Level Switch 1 = sensor level switch yang terletak pada lantai 1
DRB2	Down Request Button 2 = tombol request untuk turun yang terletak di lantai 2
DRB3	Down Request Button 3 = tombol request untuk turun yang terletak di lantai 3
PLS	Position Limit Switch = sensor level switch yang menyatakan posisi car telah tepat
DLS	Down Limit Switch = sensor limit switch yang terletak paling bawah
ULS	Up Limit switch = sensor limit switch yang terletak paling atas
DCS	Door close switch = sensor limit switch menyatakan pintu tertutup

Tabel 2. Kelompok Lamp input

ID	Deskripsi
URB1ind	Up Request Button indikator 1 = indikator terdapat permintaan naik pada lantai 1
URB2ind	Up Request Button indikator 2 = indikator terdapat permintaan naik pada lantai 2
URB3ind	Up Request Button indikator 3 = indikator terdapat permintaan naik pada lantai 3
MFRCind	Motor Forward Rotation Kontrol indikator = indikator Sinyal motor ke atas
MRRCind	Motor Reverse Rotation Kontrol indikator = indikator Sinyal motor ke bawah
FL1ind	Floor Level1 Indikator = indikator level switch pada lantai 1
FL2ind	Floor Level2 Indikator = indikator level switch pada lantai 2
FL3ind	Floor Level2 Indikator = indikator level switch pada lantai 3

Tabel 3. Kelompok Kontrol Input

ID	Deskripsi
DSC	Down Speed Kontrol = Sinyal kontrol kecepatan rendah
BC	Break Kontrol = sinyal kontrol motor berhenti
MC	Motor Kontrol = sinyal kontrol motor bekerja
MFRC	Motor Forward Rotation Kontrol = Sinyal kontrol motor berputar ke atas
MRRC	Motor Reverse Rotation Kontrol = Sinyal kontrol motor berputar ke bawah
DOCC	Door Open / Close Kontrol = Sinyal kontrol pintu car tertutup
CPS	Car Parking Switch = sinyal car pada posisi parkir

3. HASIL DAN PEMBAHASAN

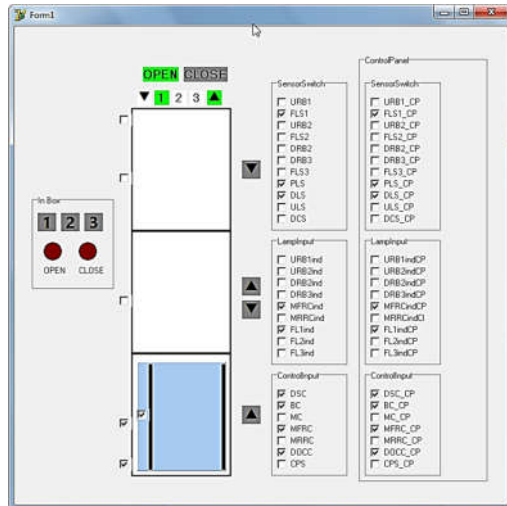
Uji coba simulator dilakukan dengan memberikan tombol request dan melihat apakah simulator berkerja seperti dengan model simulator ED 4000.

3.1 Posisi awal car

Ketika program simulator pertama kali dijalankan maka posisi car berada pada lantai 1 dan pintu terlihat membuka seperti yang ditunjukkan pada Gambar 12.

3.2 Request Tunggal

Pada tahap ini dicoba request dilakukan bergantian satu-satu tanpa ada request terjadi sebelum pengguna selesai menggunakan elevator.



Gambar 12. Tampilan awal program

3.2.1 Request di lantai 1 ke lantai 2

Pada saat awal car di lantai 1 dan pintu terbuka. Ditekan tombol 2 dalam car. Pintu elevator tertutup dan car bergerak ke lantai 2 dengan kecepatan lambat. Sinyal DSC (Down Speed Kontrol) terlihat aktif. Ketika posisi car telah tepat di lantai 2 (ditandai dengan aktifnya sinyal PLS) pintu car akan terbuka.

3.2.2 Request di lantai 3 menuju lantai 1

Posisi car berada di lantai 2 dan pintu terbuka. Tombol request di lantai 3 (URB3) ditekan. Indikator URB3 menyala, pintu car menutup car bergerak ke atas menuju lantai 3. Sampai di lantai tiga car berhenti, indikator request (URB3ind) mati dan pintu terbuka. Pengguna masuk dan menekan tombol 1 yang terletak dalam car. Pintu menutup dan car bergerak ke bawah menuju lantai 1. Sampai di lantai 1 pintu terbuka.

3.2.3 Request di lantai 2 menuju lantai 1

Posisi car ada di lantai 1, ada request di lantai 2 mau menuju lantai 1 (DRB2). Car bergerak menuju lantai 2. Sampai di lantai 2 indikator DRB2 mati dan pintu terbuka. Pengguna menekan tombol 1 dalam car. Pintu menutup dan bergerak ke bawah ke lantai 1.

3.3 Request jamak

Pada tahap ini dilakukan percobaan untuk melihat yang terjadi ketika ada le request dimana request sebelumnya belum diselesaikan.

3.3.1 Request di lantai 2 mau ke lantai 1, request di lantai 3 mau ke lantai 1

Posisi car di lantai 1. Request DRB2 (mau ke lantai 1). Beberapa saat kemudian terjadi request DRB3 (mau ke lantai 1). Car bergerak ke atas ke lantai 3 pintu membuka, tombol1 ditekan. Car bergerak ke lantai 2, pintu terbuka, pengguna masuk car. Pintu tertutup, car bergerak ke bawah ke lantai 1.

Pada percobaan ini car tidak berhenti di lantai 2 terlebih dahulu karena mengetahui bahwa pengguna di lantai dua mau ke bawah. Sehingga sistem membuat car terus ke atas karena ada request di lantai 3. Ini berbeda untuk request tunggal. jika tidak ada request di lantai 3 maka car akan berhenti di lantai 2 dan setelah pengguna menekan tombol 1 maka car akan menuju lantai 1.

Jika request di lantai 3 belum terjadi maka ketika FLS2 aktif maka kecepatan car akan melambat karena menganggap request tunggal yang terjadi. Tapi apabila car belum bergerak turun dan terjadi request DRB3, maka car bergerak naik terlebih dahulu. Demikian juga jika tombol DRB3 ditekan sebelum penekanan tombol 2 pada car, maka car akan bergerak ke lantai 3 untuk menyelesaikan putran motor naik. Namun apabila yang ditekan duluan adalah tombol 2 pada car daripada tombol DRB3 maka car akan bergerak ke bawah setelah penekanan tombol 2.

3.3.2 Request di lantai 3 mau ke lantai 2, kemudian diikuti request dilantai 2 mau ke lantai 3

Posisi car pada lantai 1. Terjadi request DRB3 mau ke lantai 2 dan sebelum car sampai lantai 2 (sebelum FLS2 aktif) terjadi penekanan tombol DRB2, maka car akan berhenti dulu di lantai 2. Apabila penekanan tombol DRB2 ketika FLS sudah aktif, maka car tidak akan berhenti di lantai 2. Ini untuk menghindari car berhenti mendadak dari motor bergerak cepat.

3.3.2 Request di lantai 3, kemudian diikuti request dilantai 2 mau ke lantai 1

Posisi car di lantai 1. Terjadi request DRB3 mau ke lantai 2. Kemudian terjadi request DRB2 mau ke lantai 1. Maka car akan terus naik sampai lantai 3. Setelah itu baru turun untuk berhenti di lantai 2.

4. KESIMPULAN

Dari kegiatan penelitian yang telah dilakukan dapat disimpulkan bahwa simulator sistem elevator berjalan dengan baik. Simulator berupa program Visual yang dirancang berdasarkan pedoman elevator trainer (berupa miniature elevator) ED-4000E. Sistem kendali elevator juga berjalan dengan baik. Sistem pengendali (kontrol Panel) yang dirancang menjadi satu program dengan Plant pada program Delphi berjalan dengan baik seperti cara kerja Elevator traine ED-4000E .

Penelitian ini masih perlu diperbaiki dan dikembangkan, diantaranya adalah perlu penambahan indikator performance terhadap sistem simulator elevator seperti waktu tunggu dan energy yang digunakan (lama waktu motor bekerja). Simulator ini dapat juga dikembangkan menjadi lebih dari tinggal lantai, seta perlu dicari metode baru pada program sistem kontrol. Pada penelitian ini program sistem kontrol tidak lain adalah konversi dari rangkaian logika digital hardware yang dijadikan software sehingga belum dapat memberikan keleluasaan untuk dikembangkan sistem kontrol yang lebih canggih.

5. DAFTAR PUSTAKA

- [1] Zhang Yajun, Chen Long, Fan Lingyan, A Design of Elevator Positioning Control System Model, IEEE Int. Conference Neural Networks & Signal Processing Zhenjiang, China, June 8~10, 2008
- [2] Robert Gulde and Michael Weeks, A POSITION CONTROL SYSTEM DESIGN, Proceedings of The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications ISBN 0-7695-1929-6/03 \$17.00 © 2003 IEEE
- [3] Sandar Htay, Su Su Yi Mon, Implementation of PLC Based Elevator Control System, International Journal of Electronics and Computer Science Engineering
- [4] Saurabh Sharma, T.Y.Ladakhi, A.P.Tiwary, Dr. B.B.Pradhan, R.Phipon, Application of PLC for Elevator Control System, International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC) 2011 Proceedings published by International Journal of Computer Applications® (IJCA)
- [5] Yi-Sheng Huang, Sheng-Luen Chung, and Mu-Der Jeng, MODELING AND CONTROL OF ELEVATORS BY STATECHARTS, Asian Journal of Control, Vol. 6, No. 2, pp. 242-252, June 2004
- [6] ED Laboratory, Operating Manual Elevator Trainer Experimental System model ED-4000E
- [7] Digital Systems, Principles and Applications, 8th ed, Ronald J. Tocci and Neas S. Widmer, Prentice Hall, ISBN 0130856347
- [8] Wahana Komputer , Panduan Praktis Delphi 2010 Programming : Konsep Dan Implementasi , Andi Publisher, ISBN 9789792918638