BAB III

ANALISIS DAN PERANCANGAN MODEL SEDERHANA SISTEM KONTROL ELEVATOR

Pada Bab III akan dijelaskan mengenai bagaimana sistem kontrol elevator bekerja. Mula-mula masalah elevator dianalisis terlebih dahulu. Setelah menganalisis masalah sistem kontrol elevator, kemudian dimodelkan dalam bentuk FSM.

3.1 ANALISIS MASALAH SISTEM KONTROL ELEVATOR

Mula-mula akan dijelaskan bagian-bagian dari elevator beserta fungsinya masing-masing.

3.1.1 Struktur elevator [3]

Struktur elevator terdiri dari beberapa bagian utama yaitu *traction machine*, *control panel*, *elevator car*, *landing door* dan *counterweight*, sebagaimana tampak pada Gambar 3.1.

Definisi 3.1

Traction machine berfungsi mengendalikan katrol dengan motor listrik untuk menurunkan atau menaikkan *elevator car* dengan tali.

Definisi 3.2

Control panel adalah tempat yang didalamnya terdapat sirkuit-sirkuit yang mengontrol kerja dari elevator dan terdapat sistem di dalamnya yang mengontrol elevator.

Definisi 3.3

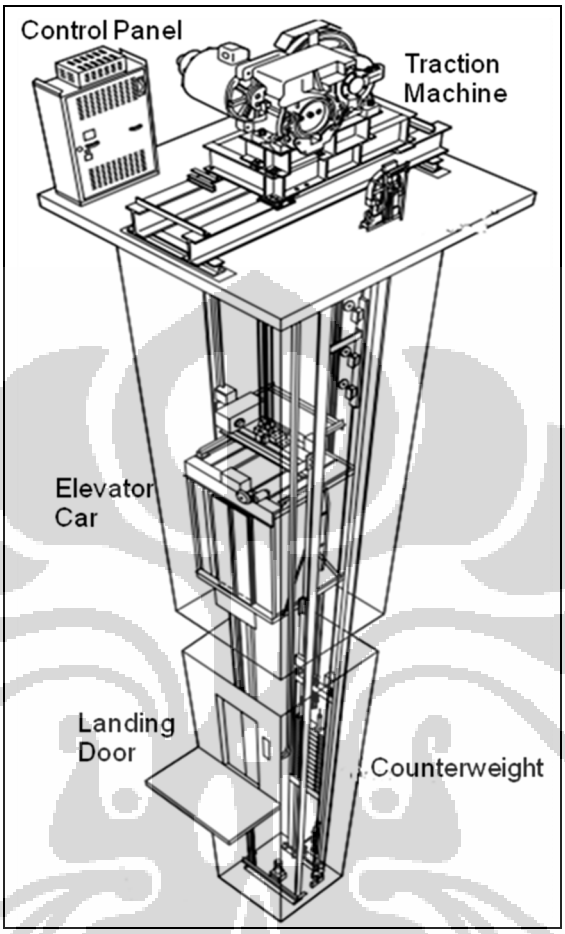
Elevator car adalah ruangan tempat pengguna elevator, di dalamnya terdapat tombol *car*. Tombol *car* berfungsi untuk menentukan lantai yang ingin dituju pengguna elevator.

Definisi 3.4

Landing door adalah tempat pengguna menunggu elevator yang terdapat pada setiap lantai. Terdapat juga tombol naik lantai dan tombol turun lantai. Kedua tombol tersebut berfungsi untuk menunjukkan pengguna meminta naik lantai atau turun lantai.

Definisi 3.5

Counterweight berfungsi untuk menyeimbangkan berat dari *elevator car*. Tujuan dari penyeimbangan adalah untuk menghemat energi. Dengan berat yang sama di antara kedua sisi katrol, hanya dibutuhkan sedikit gaya untuk menyeimbangkan sisi yang satunya[2].



Gambar 3.1 Struktur elevator modern
[Sumber: Markon, Sandor A. [3]: 9]

Pada penelitian ini akan dibuat model dari sistem kontrol elevator hanya pada bagian *landing door* dan *elevator car.*

3.1.2 Jenis-jenis permintaan dalam elevator

Saat orang menggunakan elevator terdapat dua permintaan yang mempengaruhi arah kerja elevator, yaitu permintaan lantai dan permintaan elevator.

1. Permintaan lantai

   Ketika pengguna telah sampai pada *landing door* namun tidak menemukan *elevator car* di lantai tersebut, kemudian pengguna tersebut menekan tombol lantai (tombol naik atau tombol turun) sesuai keinginan. Sehingga e*levator car* akan menuju ke permintaan pengguna tersebut. Setelah sampai pada lantai yang dituju, *elevator car* akan membuka pintunya beberapa waktu, supaya pengguna dapat memasuki e*levator car*.

2. Permintaan *car*

   Ketika pengguna berada di dalam *elevator car*, dia akan meminta elevator untuk membawanya ke lantai yang ingin dituju, inilah yang disebut permintaan *car*. Orang tersebut akan menekan tombol *car* yang berupa angka lantai, sehingga elevator akan membawanya ke lantai tersebut. Ketika telah sampai pada lantai tujuan, *elevator car* kembali membuka pintunya untuk beberapa saat, sehingga orang yang menggunakan elevator dapat keluar dari *elevator car* ke lantai tujuannya.

Pernyataan permintaan lantai dan permintaan *car* ini hanya menggambarkan penggunaan elevator untuk satu orang saja. Untuk memenuhi permintaan yang berbeda lebih dari satu orang, elevator memiliki kondisi-kondisi dalam melayani banyak permintaan tersebut. Selanjutnya

akan dibahas kondisi-kondisi yang mempengaruhi pergerakan elevator dalam menangani banyaknya permintaan

3.1.3 *Selective collective control* [3]

Kondisi-kondisi yang dialami elevator dalam menangani permintaan adalah sebagai berikut:

1. *Elevator car* yang sedang diam, melayani permintaan pertama yang masuk, kemudian *elevator car* digerakkan menuju lantai tersebut dengan arah naik atau turun sesuai permintaan pertama.
2. Ketika ada permintaan yang searah dengan elevator, elevator melayani permintaan tersebut secara berurutan.
3. Jika elevator sudah melayani semua permintaan yang searah maka permintaan yang tidak searah akan dilayani oleh elevator.
4. Ketika tidak ada lagi permintaan maka *elevator car* akan diam pada lantai terakhir yang dikunjungi sampai ada permintaan yang masuk.

Kondisi-kondisi ini dinamakan *selective collective control* (S/C). Permintaan yang searah adalah permintaan yang sesuai dengan arah bekerja elevator dan lantai permintaan tersebut belum dilewati oleh elevator.

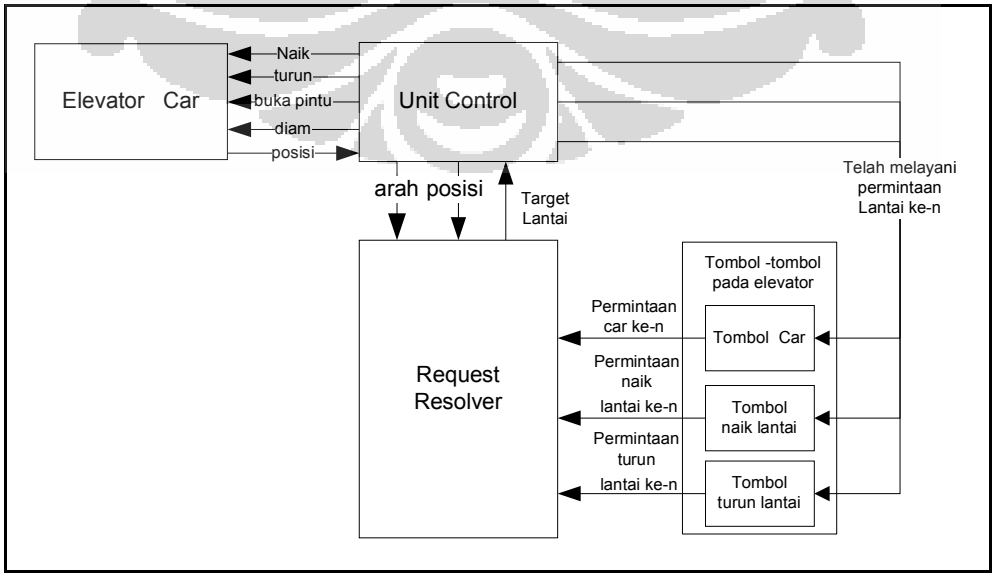
Masalah dalam sistem kontrol elevator adalah bagaimana menggerakan *elevator car* pada n lantai dengan kendala:

1. Dalam *elevator car* terdapat n buah tombol *car*, satu tombol untuk setiap lantai. Saat tombol ditekan, lampu dari tombol ini menyala kemudian elevator menuju lantai yang ingin dikunjungi, setelah elevator sampai pada lantai yang diminta, lampu dari tombol tersebut mati ketika elevator membukakan pintunya untuk melayani permintaan di lantai tersebut.
2. Setiap lantai pada *landing door*, kecuali lantai ke-1 dan lantai ke-n, memiliki dua tombol lantai, yaitu tombol permintaan untuk naik lantai dan tombol permintaan untuk turun lantai. Sedangkan untuk lantai ke-1 hanya terdapat tombol permintaan naik dan lantai ke-n hanya terdapat tombol permintaan turun. Saat ditekan tombol tersebut menyala dan akan mati apabila elevator telah sampai pada lantai yang diminta ketika elevator membuka pintunya untuk melayani permintaan di lantai tersebut.
3. Ketika tidak ada permintaan lagi untuk elevator, elevator diam pada lantai terakhir yang dikunjungi dengan pintu tertutup.
4. *Elevator car* digerakkan sesuai dengan kondisi S/C.

Sistem kontrol elevator mengatur pergerakan *elevator car* terhadap permintaan pengguna baik permintaan lantai atau permintaan *car.* Ketika

tombol-tombol lantai atau *car* ditekan maka tombol tersebut memberikan sinyal kepada sistem kontrol bahwa lantai tersebut ingin dilayani.

Model sederhana sistem kontrol elevator dibagi menjadi dua bagian utama yaitu *unit control* dan *request resolver*. *Unit control* memberikan perintah kepada *elevator car* untuk naik, turun, diam atau buka pintu, sementara *elevator car* menginformasikan keberadaannya di suatu lantai tertentu kepada *unit control*. *Request resolver* menerima input-input dari tombol permintaan *car* dan tombol permintaan lantai, kemudian dengan memperhatikan arah dan posisi *elevator car* yang diterima dari *unit control, request resolver* menentukan lantai berikutnya yang akan dituju oleh *elevator car* . Setelah *elevator car* melayani lantai, *unit control* memberikan sinyal ke tombol-tombol permintaan bahwa permintaan telah dilayani, kemudian lampu-lampu tombol lantai yang dilayani dimatikan. Hal tersebut tampak pada Gambar 3.2.



Gambar 3.2 Diagram kerja sistem kontrol elevator sederhana

3.2 PERANCANGAN MODEL SISTEM KONTROL ELEVATOR

Model sistem kontrol elevator dibuat dengan menggunakan FSM. FSM umumnya digunakan untuk memodelkan sistem yang didominasi kontrol, yaitu sistem yang menerima input dan mengeluarkan output. FSM menggambarkan perilaku sebuah sistem dengan menggunakan keadaan-keadaan yang dialami oleh sistem dan bagaimana sistem tersebut berpindah keadaan atau bertransisi. Keuntungan pembuatan sebuah model dengan menggunakan FSM sebagai model komputasi adalah mudah untuk dimengerti dan dijadikan ke dalam bahasa pemrograman.

3.2.1 *Finite State Machine with Datapath*

Dalam pembuatan model ini digunakan *extended* FSM, yaitu *Finite State Machine with Datapath* (FSMD). Model ini ditambahkan kata *extended* karena FSM ini ditambahkan komponen baru yaitu himpunan variabel. Kelebihan dari FSMD adalah kemampuan untuk menyimpan variabel, variabel ini mempengaruhi fungsi transisi kemudian fungsi output dapat merubah nilai dari variabel.

Berikut adalah definisi dari FSMD:

Definisi 3.6

FSMD M = (S, I, O, V, f, g, s_0) terdiri dari:

1. Himpunan berhingga keadaan S.
2. Himpunan berhingga input I .
3. Himpunan berhingga output O.
4. Himpunan berhingga variabel V.
5. Fungsi transisi f yang memasangkan pasangan keadaan, input dan variabel pada keadaan berikutnya. Dinotasikan $f : S \times I \times V \rightarrow S$.
6. Fungsi output g yang memasangkan pasangan keadaan pada output atau variabel. Dinotasikan $g : S \rightarrow O \mid V$.
7. s_0 merupakan keadaan awal, $s_0 \quad S$.

Adapun langkah-langkah dalam pemodelan sistem kontrol elevator dengan menggunakan FSMD, yaitu:

1. Daftarkan semua keadaan yang mungkin, berikan setiap keadaan nama yang deskriptif.
2. Nyatakan semua input, output dan variabel.
3. Daftarkan transisi kepada keadaan lain yang mungkin dari setiap keadaan, dengan kondisi seperti apa transisi itu terjadi.
4. Daftarkan aksi-aksi yang berhubungan untuk setiap keadaan.
5. Untuk setiap keadaan, yakinkan bahwa kondisi transisi ekslusif (tidak ada dua kondisi yang dapat terjadi secara bersamaan).

3.2.2 Perancangan Model *Unit control* dari Elevator [8]

Unit control bekerja sebagai berikut, mula-mula *elevator car* dalam keadaan diam dengan menunggu input target lantai yang ingin dituju. Ketika target masuk dilihat apakah target lantai lebih besar atau lebih kecil dari posisi *elevator car*. Apabila target lebih besar dari posisi *elevator car* maka *unit control* memberikan perintah kepada elevator untuk naik satu lantai, sedangkan jika target lebih kecil dari posisi elevator maka *unit control* memberikan perintah kepada elevator untuk turun satu lantai. Hal tersebut terus berlangsung dengan target selalu diperbaharui setiap terjadi transisi. Ketika target telah sama dengan posisi elevator maka elevator akan melayani lantai tersebut dengan membuka pintunya untuk beberapa saat. Setelah itu elevator dalam keadaan diam lagi dengan menutup pintunya. Lalu memeriksa target lantai berikutnya.

Berdasarkan pernyataan di atas maka keadaan dalam *unit control* dibagi menjadi 4 yaitu diam, naik, turun dan bukapintu dengan keadaan awal diam. Himpunan keadaan dinotasikan dengan

S = {diam, naik, turun, bukapintu}

Penjelasan masing-masing keadaan adalah sebagai berikut:

1. Keadaan diam adalah keadaan ketika *elevator car* diam dengan menutup pintunya dan melihat target permintaan yang diterima dari *request resolver.*

2. Keadaan naik adalah keadaan ketika *elevator car* menaiki satu lantai kemudian mengubah arah operasinya menjadi naik dan memeriksa target lantai berikutnya.
3. Keadaan turun adalah keadaan ketika *elevator car* menuruni satu lantai kemudian mengubah arah operasinya menjadi turun dan memeriksa target lantai berikutnya.
4. Keadaan bukapintu adalah keadaan ketika *elevator car* melayani permintaan di lantai tersebut dengan membuka pintunya selama beberapa saat.

Inputnya berupa target lantai (trgt) dan dapat bernilai antara 1 sampai dengan n (berdasarkan jumlah lantai) atau bernilai 0 jika tidak terdapat target lantai yang ingin dituju. Himpunan input dinotasikan dengan

$$I = \{trgt\}$$

Outputnya adalah berupa perintah kepada *elevator car* berupa naik satu lantai (k), turun satu lantai (t), buka pintu (bp) dan tutup pintu (tp). Himpunan output dinotasikan dengan

$$O = \{k, t, bp, tp\}$$

Variabelnya adalah posisi lantai *elevator car* (pos), arah elevator bekerja (arah) dan timer (tmr) untuk menentukan berapa lama elevator membuka pintunya. Variabel pos dapat bernilai 1 sampai n (berdasarkan jumlah lantai). Untuk variabel arah bernilai 1 untuk naik, 0 untuk turun.

Himpunan variabel dinotasikan dengan

$$V = \{pos, tmr, arah\}$$

Berikut ini fungsi transisi keadaan pada *unit control* yang terjadi saat:

- Keadaan diam yaitu:
  1. Jika *elevator car* tidak memiliki target lantai maka *elevator car* akan tetap diam. Dinotasikan dengan

     $$diam \text{ dan } trgt = 0 \rightarrow diam$$

  2. Jika target lantai sama dengan posisi lantai *elevator car* maka *elevator car* akan melayani lantai tersebut dengan membuka pintunya. Dinotasikan dengan

     $$diam \text{ dan } trgt = pos \rightarrow bukapintu$$

  3. Jika target lantai lebih besar dari posisi lantai *elevator car* maka keadaannya akan menjadi naik. Dinotasikan dengan

     $$diam \text{ dan } trgt > pos \rightarrow naik$$

  4. Jika target lantai lebih kecil dengan posisi lantai *elevator car* maka keadaannya akan menjadi turun. Dinotasikan dengan

     $$diam \text{ dan } trgt < pos \rightarrow turun$$

- Keadaan naik yaitu:
  1. Jika target lantai lebih besar dari posisi lantai *elevator car* maka keadaannya akan menjadi naik. Dinotasikan dengan

     $$naik \text{ dan } trgt > pos \rightarrow naik$$

2. Jika target lantai sama dengan posisi lantai *elevator car* maka keadaannya akan melayani lantai tersebut dengan membuka pintunya. Dinotasikan dengan

naik dan trgt = pos → bukapintu

- Keadaan turun yaitu:

1. Jika target lantai lebih kecil dari posisi lantai *elevator car* maka keadaannya akan menjadi turun kembali. Dinotasikan dengan

turun dan trgt < pos → turun

2. Jika target lantai sama dengan posisi lantai *elevator car* maka keadaannya akan melayani lantai tersebut dengan membuka pintunya. Dinotasikan dengan

turun dan trgt = pos → bukapintu

- Keadaan bukapintu yaitu:

1. Jika masih belum 10 detik maka keadaannya akan tetap membuka pintunya. Dinotasikan dengan

bukapintu dan tmr < 10 → bukapintu

2. Jika sudah melewati 10 detik maka keadaanya akan kembali menjadi diam. Dinotasikan dengan

bukapintu dan tmr =10 → diam

Sedangkan fungsi output pada semua keadaan di *S* yaitu:

1. Saat keadaan diam maka *elevator car* akan menutup pintunya. Dinotasikan dengan

   $$\text{diam} \rightarrow \text{tp}$$

2. Saat keadaan naik maka *unit control* memerintahkan *elevator car* untuk naik satu lantai dan arah elevator menjadi naik. Dinotasikan dengan

   $$\text{naik} \rightarrow \text{k dan arah} = 1$$

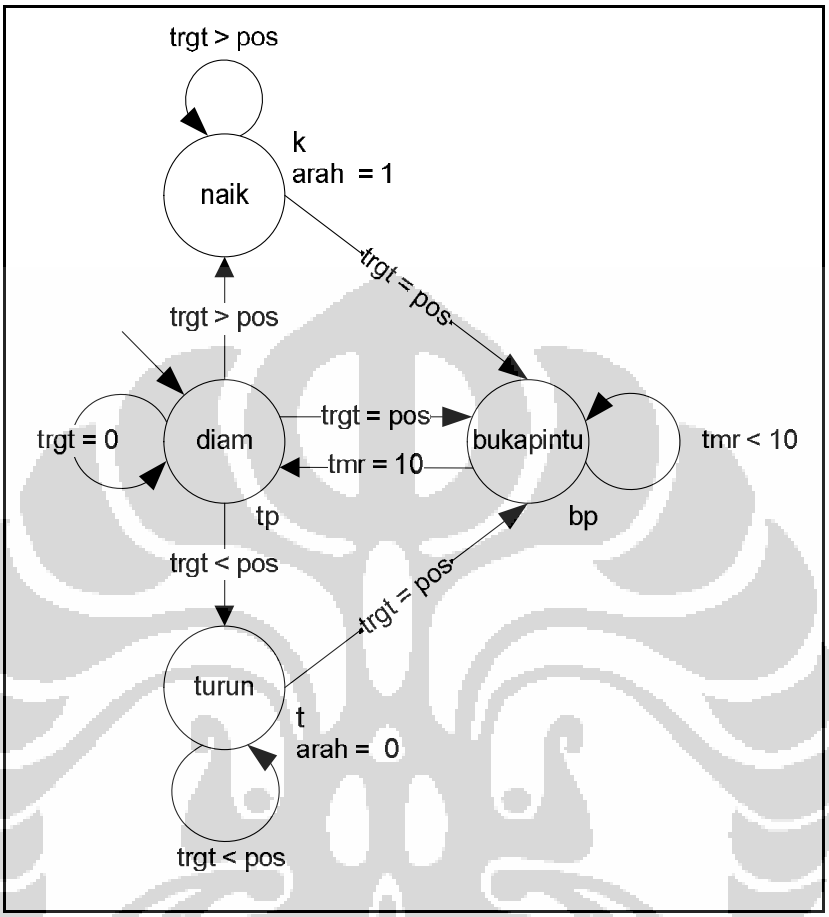
3. Saat keadaan turun maka *unit control* memerintahkan *elevator car* untuk turun satu lantai dan arah elevator menjadi turun. Dinotasikan dengan

   $$\text{turun} \rightarrow \text{t dan arah} = 0$$

4. Saat keadaan bukapintu maka *elevator car* akan membuka pintunya. Dinotasikan dengan

   $$\text{bukapintu} \rightarrow \text{bp}$$

Gambar 3.3 menunjukkan diagram transisi untuk *unit control*



Gambar 3.3 Diagram transisi untuk *unit control*

3.2.3 Perancangan *request resolver* [7]

Elevator car akan digerakkan sesuai dengan S/C maka elevator akan bekerja menghabiskan permintaan yang searah terlebih dahulu, setelah semua permintaan yang searah dilayani lalu permintaan yang tidak searah dilayani.

Permintaan pada elevator dibagi 3 yaitu:

1. Permintaan *car*

   Jika tombol *car* ditekan pada lantai ke-i, i = 1, ..., n maka car(i) akan bernilai 1. Hal ini menyatakan terdapat permintaan *car* pada lantai ke-i.

2. Permintaan naik

   Jika tombol naik ditekan pada lantai ke-i, i = 1, ..., n – 1 maka naik(i) akan bernilai 1. Hal ini menyatakan terdapat permintaan naik pada lantai ke-i.

3. Permintaan turun

   Jika tombol turun ditekan pada lantai ke-i, I = 2, ..., n maka turun(i) akan bernilai 1. Hal ini menyatakan terdapat permintaan turun pada lantai ke-i.

Pada awalnya semua permintaan bernilai 0. Setelah permintaan dilayani oleh elevator maka permintaan akan bernilai 0. Jika tidak terdapat permintaan maka *request resolver* mengembalikan nilai 0 yang berarti tidak terdapat target lantai yang dituju oleh *elevator car*.

Pada saat *elevator car* bergerak naik maka *request resolver* akan memeriksa permintaan *car* atau permintaan naik dari posisi lantai *elevator car* berada sampai lantai ke-n. Ketika tidak ditemukan permintaan *car* atau naik maka *request resolver* akan memeriksa permintaan yang pertama kali masuk dari lantai ke-n sampai lantai ke-1. Ketika mendapatkan permintaan yang

masuk, *request resolver* langsung menjadikan lantai tersebut target lantai selanjutnya yang akan dituju.

Pada saat *elevator car* nergerak turun maka *request resolver* akan memeriksa permintaan *car* atau permintaan turun dari posisi lantai *elevator car* berada sampai lantai ke-1. Ketika tidak ditemukan permintaan *car* atau naik maka *request resolver* akan memeriksa permintaan yang pertama kali masuk dari lantai ke-1 sampai lantai ke-n. Ketika mendapatkan permintaan yang masuk, *request resolver* langsung menjadikan lantai tersebut menjadi target lantai selanjutnya yang akan dituju.

Pemeriksaan dilakukan supaya elevator melayani lantai dengan permintaan yang searah secara berurutan. Jika tidak ditemukan permintaan searah maka *request resolver* mencari permintaan yang masuk pertama kali. Pada saat arah naik elevator mengutamakan permintaan dari atas terlebih dahulu dan pada saat arah turun elevator mengutamakan permintaan dari lantai bawah terlebih dahulu. Hal ini supaya elevator bekerja sesuai dengan S/C, yaitu menghabiskan permintaan yang searah terlebih dahulu.

Jika dituliskan dalam bentuk pseudocode maka algoritma dari *request resolver* adalah sebagai berikut:

```
procedure target( car(i), naik(i), turun(i): {0, 1},
    posisi: positive integer, arah: {0, 1})
target := 0
if arah := 1 then
     for  i:= posisi to n
     begin
          if car(i) or naik(i) then
          begin
               target := i
               break
          end
          if i = n then
               for j:= n to 1
                    if car(j) or turun(j) or naik(j) then
                    begin
                         target := j
                         break
                    end
     end
else for  i:= posisi to 1
     begin
          if car(i) or turun(i) then
          begin
               target := i
               break
          end
          if i = 1 then
               for j:= 1 to n
                    if car(j) or turun(j) or naik(j) then
                    begin
                         target := j
                         break
                    end
     end
```

3.2.4 Perancangan model tombol-tombol pada elevator

Tombol-tombol digunakan oleh sistem kontrol elevator untuk mengetahui adanya permintaan pada suatu lantai. Jika tombol ini ditekan maka lampu tombol ini menyala dan tombol ini akan memberikan pesan kepada sistem bahwa lantai ingin dilayani. Setelah itu elevator akan berjalan ke lantai tersebut untuk melayani permintaan.

Dalam elevator terdapat dua tombol yaitu tombol *car* dan tombol lantai. Mula-mula akan dibahas untuk tombol *car*, misalkan elevator melayani n buah lantai maka terdapat n buah tombol *car* yang berisi masing-masing permintaan tiap lantai di dalam *elevator car*. Ketika tombol *car* lantai ke-i ditekan dan *elevator car* tidak berada pada lantai ke-i maka lampu tombol *car* lantai ke-i akan menyala kemudian memberikan sinyal kepada elevator bahwa lantai ke-i ingin dilayani. Ketika lampu tombol *car* hidup kemudian *elevator car* sampai pada lantai ke-i dan melayani lantai tersebut dengan membuka pintunya maka lampu tombol *car* lantai ke-i tersebut akan mati.

Berdasarkan analisis masalah tombol *car*, FSMD dari tombol car adalah sebagai berikut:

- Keadaannya dinotasikan dengan:
    1. TCH(i) : lampu tombol *car* lantai ke-i hidup.
    2. TCM(i): lampu tombol *car* lantai ke-i mati.

    Keadaan awalnya adalah TCM(i).

- Inputnya dinotasikan dengan:
    1. TCD(i): tombol *car* lantai ke-i ditekan.
    2. ETL(i): Elevator tiba pada lantai ke-i.
- Outputnya dinotasikan dengan:
    1. CH(i) : menyalakan lampu tombol *car* lantai ke-i.
    2. CM(i) : mematikan lampu tombol *car* lantai ke-i.
- Variabelnya dinotasikan dengan:
    1. EML(i) : Elevator melayani lantai ke-i.
    2. car(i): Terdapat permintaan *car* Lantai ke-i.

  Untuk EML(i) dan car(i) bernilai 1 jika keadaannya berlaku dan bernilai 0 jika keadaannya tidak berlaku.
- Fungsi transisinya adalah sebagai berikut:

1. Jika lampu tombol *car* lantai ke-i mati dan tombol *car* lantai ke-i ditekan sedangkan elevator tidak melayani lantai ke-i maka lampu tombol *car* lantai ke-i hidup. Dinotasikan dengan

$$TCM(i) \text{ dan } TCD(i) \text{ dan } EML(i) = 0 \rightarrow TCH(i)$$

2. Jika lampu tombol *car* lantai ke-i hidup kemudian elevator tiba pada lantai ke-i dan melayani lantai ke-i maka lampu tombol elevator lantai ke-i mati. Dinotasikan dengan

$$TCH(i) \text{ dan } ETL(i) \text{ dan } EML(i) = 1 \rightarrow TCM(i)$$

- Fungsi outputnya untuk setiap keadaan tombol *car* adalah sebagai berikut:

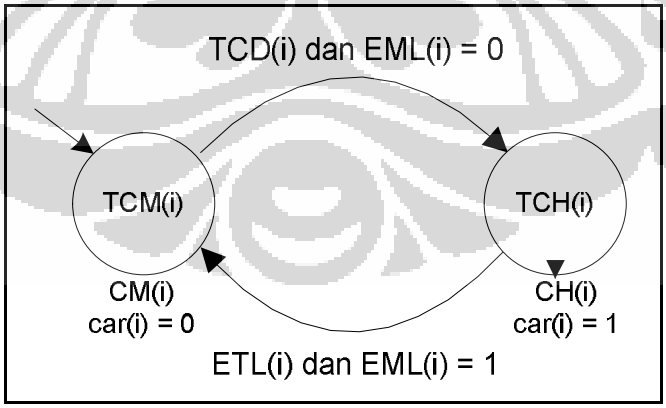
  1. Pada saat lampu tombol *car* lantai ke-i hidup maka lampu tombol *car* lantai ke-i menyala dan terdapat permintaan *car* lantai ke-i. Dinotasikan dengan

     $$TCH(i) \rightarrow CH(i) \text{ dan } car(i) = 1$$

  2. Pada saat lampu tombol *car* lantai ke-i mati maka lampu tombol *car* lantai ke-i tidak menyala dan tidak terdapat permintaan *car* lantai ke-i. Dinotasikan dengan

     $$TCM(i) \rightarrow CM(i) \text{ dan } car(i) = 0$$

Berdasarkan pemodelan FSMD tombol *car* tersebut dapat digambarkan model dengan diagram transisi sebagaimana Gambar 3.2.



Gambar 3.2 Diagram transisi untuk tombol *car*

Analisis masalah untuk tombol lantai adalah sebagai berikut. Ketika tombol naik lantai ke-i ditekan dan *elevator car* tidak sedang berada pada lantai ke-i maka lampu tombol naik lantai ke-i akan menyala dan akan

mengirimkan sinyal kepada sistem bahwa lantai ke-i ingin dilayani oleh elevator yang sedang naik. Ketika tombol naik lantai ke-i menyala kemudian *elevator car* tiba pada lantai ke-i dan melayani lantai tersebut maka lampu lantai naik ke-i tersebut akan mati dan elevator memberikan sinyal kepada sistem bahwa elevator telah melayani permintaan naik lantai ke-i.

Untuk tombol turun, ketika tombol lantai turun ke-i ditekan dan *elevator car* tidak sedang berada pada lantai ke-i maka lampu tombol lantai turun ke-i akan menyala dan akan mengirimkan sinyal kepada sistem bahwa lantai ke-i ingin dilayani oleh elevator yang sedang turun. Ketika tombol turun lantai ke-i menyala kemudian *elevator car* tiba pada lantai ke-i dan melayani lantai tersebut maka lampu lantai turun ke-i tersebut akan mati dan elevator memberikan sinyal kepada sistem bahwa elevator telah melayani permintaan lantai turun ke-i.

Berdasarkan analisis masalah tombol lantai, FSMD dari tombol lantai adalah sebagai berikut:

- Keadaannya dinotasikan dengan:
    1. TLH(d,i) : lampu tombol d lantai ke-i hidup, d = naik, turun.
    2. TLM(d,i): lampu tombol d lantai ke-i mati, d = naik, turun.

    Dengan keadaan awal TLM(d,i).
- Inputnya dinotasikan dengan:
    1. TLD(d,i): tombol d lantai ke-i ditekan d = naik, turun.
    2. ETL(i): Elevator tiba pada lantai ke-i.

- Outputnya dinotasikan dengan:
  1. LH(d,i) : menyalakan lampu tombol d lantai ke-i, d = naik, turun.
  2. LM(d,i) : mematikan lampu tombol d lantai ke-i, d = naik, turun.
- Variabelnya dinotasikan dengan:
  1. EML(i) : Elevator melayani lantai ke-i.
  2. L(d,i): terdapat permintaan d lantai ke-i, d = naik, turun.

  Untuk EML(i) dan L(d,i) bernilai 1 jika keadaannya berlaku dan bernilai 0 jika keadaannya tidak berlaku.
- Untuk fungsi transisinya adalah sebagai berikut:

1. Jika lampu tombol d lantai ke-i mati dan tombol d lantai ke-i ditekan sedangkan elevator tidak melayani pada lantai ke-i maka lampu tombol d lantai ke-i hidup. Dinotasikan dengan

   $$TLM(d,i) \text{ dan } TLD(d,i) \text{ dan } EML(i) = 0 \rightarrow TLH(d,i)$$

2. Jika lampu tombol d lantai ke-i hidup kemudian elevator tiba pada lantai ke-i dan melayani lantai ke-i maka lampu tombol d lantai ke-i mati. Dinotasikan dengan

   $$TLH(d,i) \text{ dan } ETL(i) \text{ dan } EML(i) = 1 \rightarrow TLM(d,i)$$

- Untuk fungsi outputnya untuk setiap keadaan tombol lantai adalah sebagai berikut:

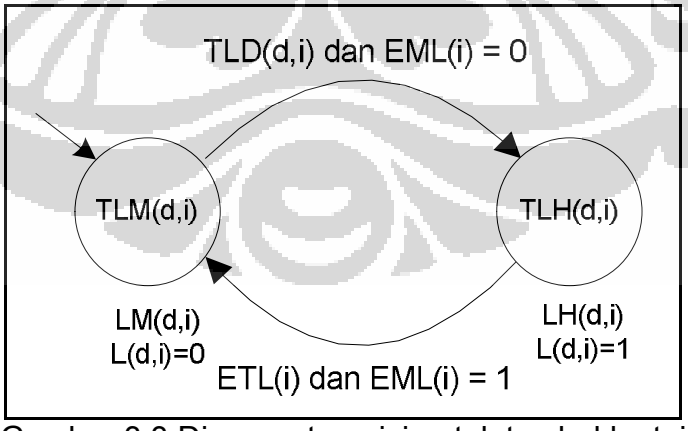
  1. Pada saat lampu tombol d lantai ke-i hidup maka menyalakan lampu tombol d lantai ke-i dan terdapat permintaan d lantai ke-i. Dinotasikan dengan

     $$TLH(d,i) \rightarrow LH(d,i) \text{ dan } L(d,i) = 1$$

  2. Pada saat lampu tombol d lantai ke-i mati maka mematikan lampu tombol d lantai ke-i dan permintaan d lantai ke-n tidak minta dilayani. Dinotasikan dengan

     $$TLM(d,i) \rightarrow LM(d,i) \text{ dan } L(d,i) = 0$$

     Diagram transisi dari tombol lantai tampak pada Gambar 3.3.



Gambar 3.3 Diagram transisi untuk tombol lantai