

## **BAB II**

### **LANDASAN TEORI**

Pada bab ini ini akan dijelaskan beberapa landasan-landasan teori sebagai hasil dari studi literatur yang berhubungan dengan perancangan dan pembuatan simulasi ini. Teori-teori itu ialah tentang lalu lintas, persimpangan, beberapa istilah lalu lintas, metode *webster*, *programmable logic controller (PLC)*, *Cx-Programmer*, *Cx-Designer*, dan *LabView*.

#### **2.1 Lampu Lalu Lintas**

Lampu lalu lintas merupakan suatu metode untuk mengatur ketertiban di persimpangan jalan. Pengaturan arus lalu lintas pada persimpangan pada dasarnya dimaksudkan untuk mengatur pergerakan kendaraan pada masing-masing sisi jalan agar dapat bergerak secara bergantian sehingga tidak saling mengganggu antar arus yang ada. [3]

#### **2.2 Persimpangan**

Persimpangan adalah bagian yang sulit dihindarkan dalam jaringan jalan karena persimpangan jalan merupakan tempat bertemu dan berganti arah arus lalu lintas dari dua jalan atau lebih. Persimpangan biasa ditemui di dalam jalan perkotaan dimana pengemudi dapat memutuskan untuk jalan terus, berbelok atau pindah jalan. [4] Persimpangan jalan merupakan suatu hal yang penting untuk di analisa karena sangat berpengaruh terhadap aliran dan keselamatan berlalu lintas. [5]

### 2.3 Beberapa Istilah Lalu Lintas

Pada perencanaan lalu lintas, dikenal beberapa istilah:

1. Waktu Siklus (*cycle time*): waktu satu periode lampu lalu lintas, dimana satu periode dihitung dari waktu yang dibutuhkan satu sisi jalan untuk mengulang menyalanya lampu hijau.

2. Fase: suatu tahapan sinyal dalam waktu satu atau lebih kesempatan bergerak melaju pada persimpangan lalu lintas untuk meminimalisir konflik yang terjadi [6]

Pemilihan dan penggunaan fase terlihat pada kejadian konflik, apabila pada suatu persimpangan ada dua konflik utama dapat diselesaikan dengan dua fase, jika ada tiga konflik utama akan diselesaikan dengan tiga fase, dan jika ada empat konflik maka diselesaikan dengan empat fase. [7]

### 2.4 *Websters*

F. V. *Webster* mengembangkan suatu persamaan untuk menghitung penundaan rata-rata per kendaraan ketika mendekati persimpangan. F. V. *Webster* juga menurunkan sebuah persamaan untuk memperoleh waktu siklus optimum yang menghasilkan penundaan kendaraan minimum. Penundaan kendaraan terjadi karena jumlah kendaraan yang masuk ke dalam sebuah persimpangan lebih besar dibandingkan dengan jumlah kendaraan yang keluar dari persimpangan tersebut [8].

Kelebihan dari metode *Webster* adalah sangat sederhana dalam pengaplikasiannya. Selain itu, pada metode ini mempertimbangkan penambahan durasi waktu lampu hijau jika ada kasus tingginya volume kendaraan yang terjadi.[4]

### 2.4.1 Volume Kendaraan (q)

Volume kendaraan adalah jumlah kendaraan yang melewati suatu titik dalam suatu ruang tertentu pada suatu interval waktu. [10] Semakin banyak jumlah kendaraan yang melalui suatu titik dalam suatu ruang tertentu pada suatu interval waktu berarti semakin besar volume lalu lintas pada titik tersebut. Peningkatan volume kendaraan tidak akan menimbulkan permasalahan apabila kapasitas ruas jalan tersebut tidak terlampaui, namun masalah akan timbul seandainya kapasitas ruas jalan tidak mampu menampung jumlah kendaraan yang melaluinya. Permasalahan yang dapat dilihat secara nyata adalah kemacetan yang dapat berujung munculnya permasalahan lain yaitu kecelakaan dan polusi [10].

### 2.4.2 Arus Jenuh (s)

Arus jenuh adalah banyaknya kendaraan yang melaju pada antrian saat kendaraan berada di kecepatan konstan, yakni setelah melakukan percepatan, sampai kendaraan melakukan perlambatan selama periode hijau dan kuning. Arus jenuh biasanya dinyatakan dalam kendaraan per jam waktu hijau [5].

Tabel 2. 1 Arus Jenuh pada Persimpangan untuk lebar pendekat

Lebar jalan (m)	3,05	3,35	3,65	3,95	4,25	4,6	4,9	5,2
Arus Jenuh (smp/jam)	1850	1875	1900	1950	2075	2250	2475	2700

Jika lebar jalan melebihi nilai yang ditetapkan, maka arus jenuh (s) = Lebar Jalan  $\times$  525 (smp/jam). Satuan mobil penumpang (smp) adalah satuan kendaraan di dalam arus lalu lintas yang disetarakan dengan

kendaraan ringan atau mobil penumpang. Besaran smp dipengaruhi oleh tipe atau jenis kendaraan, dimensi kendaraan, dan kemampuan olah gerak.

[4] Dan untuk percobaan simulasi ini menggunakan lebar jalan 3,75 maka nilai  $S=1950$

### 2.4.3 Rasio Arus ( $y$ )

Untuk mengetahui tingkat arus lalu lintas dengan cara membagi volume lalu lintas dengan arus jenuh pada masing masing fase.

$$y_i = \frac{q_i}{S_i} \dots\dots\dots(\text{Persamaan 1})$$

Ket:

$y_i$  = Rasio Arus

$q_i$  = Volume Kendaraan

$s$  = Arus Jenuh

### 2.4.4 Ratio Fase (Y)

rasio fase yaitu hasil rasio arus pada masing-masing fase.

$$FR = \sum y \dots\dots\dots(\text{persamaan 2})$$

Ket:

$y$  = Rasio

FR = Rasio Fase

### 2.4.5 Waktu Hilang (L)

Faktor yang diperlukan untuk menghitung siklus waktu maksimum adalah waktu hilang ( $L$ ), yaitu lama waktu satu siklus penuh pada saat tidak ada kendaraan. Hal ini dilakukan tidak hanya waktu merah dan waktu merah/merah/kuning tetapi juga sebagai waktu persiapan jalan (*Starting-Up*) dan persiapan berhenti (*Tailing-Off*) yang terjadi pada saat perubahan warna lampu. [4]

Waktu yang terbuang dihitung dengan rumus:  $L=2n+R$ .

$$\begin{aligned} L &= 2n + R \\ L &= 2(2) + 2 \\ L &= 6 \dots\dots\dots \text{(persamaan 3)} \end{aligned}$$

Ket:

$n$  = Banyaknya fase

$R$  = Waktu hilang yang diakibatkan antrian (waktu kuning)

### 2.4.6 Waktu Siklus Optimum ( $C_0$ )

Waktu siklus optimum adalah waktu satu siklus yang memberikan paling sedikit rata-rata waktu penundaan bagi semua kendaraan yang melewati persimpangan.

*Webster* memberikan rumus untuk menghitung waktu siklus optimum ( $C_0$ ) adalah [9].

$$C_0 = \frac{1,5 \times L + 5}{1 - FR} \dots\dots\dots \text{(persamaan 4)}$$

Keterangan:

$C_0$  = Waktu siklus optimum (detik)

$L$  = Waktu hilang total per siklus

$Y$  = Jumlah  $y$  maksimum untuk semua fase

#### 2.4.7 Waktu Hijau Efektif

Waktu hijau efektif yakni ketika kendaraan bergerak melintas selama periode nyala lampu hijau.

$$g = \frac{y(C_0 - L)}{FR} \dots\dots\dots \text{(persamaan 5)}$$

Keterangan:

$g$  = waktu hijau efektif

$y$  = tingkat arus lalu lintas tiap persimpangan

$C_0$  = waktu siklus Optimum

$L$  = waktu hilang total per siklus

$Y$  = jumlah  $y$  maksimum untuk seluruh fase [3]

#### 2.5 Metode Adaptive

Metode adaptive yaitu untuk meentukan durasi lampu hijau akan bertambah tergantung pada kondisi kepadatan berdasarkan jumlah kendaraan. Pada saat keadaan normal maka durasi lampu hijau akan normal, jika kepadatan di jalur 1

maka durasi lampu hijau akan bertambah. untuk durasi lampu hijaunya sesuai ketentuan keadaan kondisi 1 dan seterusnya.

## **2.6 Metode Fuzzy Logic**

Metode logika fuzzy menentukan durasi lampu hijau berdasarkan variabel input kepadatan kendaraan dan waktu kemacetan terjadi. Serta satu variabel output, yaitu variabel delay penyalan lampu hijau. Setelah itu dilakukan tahapan fuzzifikasi, Penalaran (Inference Machine), Aturan Dasar (Rule Based), dan Defuzzifikasi. [14]

## **2.7 Programmable Logic Controller (PLC)**

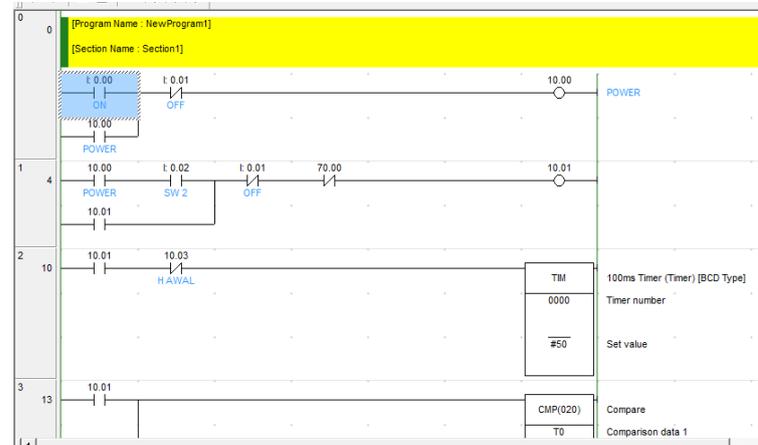
Menurut *National Electrical Manufacturers Association* (NEMA), PLC merupakan suatu peralatan elektronik digital yang menggunakan memori dan dapat diprogram untuk menyimpan arahan suatu fungsi tertentu, seperti logika, sekuensial, waktu, penghitungan, dan aritmatik untuk mengendalikan suatu. [13]

### **2.7.1 Ladder Diagram Pada Program PLC**

*Ladder diagram* adalah bahasa pemrograman yang yang dibuat dari persamaan fungsi logika dan fungsi-fungsi lain berupa pemrosesan data atau fungsi waktu dan pencacahan. *Ladder diagram* terdiri dari susunan kontak-kontak dalam satu grup perintah secara horizontal dari kiri ke kanan, dan terdiri dari banyak grup perintah

secara vertikal. Contoh dari *ladder diagram* ini adalah kontak *normaly open*, kontak *normaly close*, *output coil*, pemindahan data, dll. Garis vertikal paling kiri dan paling kanan diasumsikan sebagai fungsi

tegangan, bila fungsi dari grup perintah menghubungkan dua garis vertikal tersebut maka rangkaian perintah akan bekerja [13].

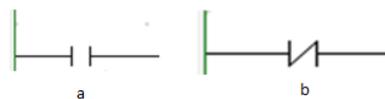


**Gambar 2. 1** Contoh program ladder

Berikut merupakan simbol yang digunakan sebagai instruksi pada sebuah *ladder diagram*:

#### 1. LOAD (LD) dan LOAD NOT

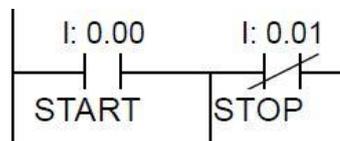
LOAD (LD) merupakan perintah yang digunakan jika urutan kerja pada sistem kontrol hanya membutuhkan satu kondisi logika. Logika ini hampir menyerupai dengan kontak relay NO. Sedangkan LOAD NOT merupakan perintah yang digunakan jika urutan kerja pada sistem kontrol hanya membutuhkan satu kondisi logika. Logika ini hampir menyerupai kontrak relay NC.



**Gambar 2. 2** LOAD (a) LOAD NOT (b).

#### 2. AND dan AND NOT

AND merupakan perintah yang digunakan jika urutan kerja pada sistem kontrol lebih dari satu kondisi logika yang harus dipenuhi semuanya untuk memberikan satu output. Logika ini hampir menyerupai relay NO. Sedangkan AND NOT merupakan perintah yang digunakan jika urutan kerja sistem kontrol lebih dari satu kondisi logika harus terpenuhi semuanya untuk memberikan satu output. Logika ini hampir menyerupai kontak relay NC.



**Gambar 2. 3** AND (a) AND NOT (b).

### 3. OR dan OR NOT

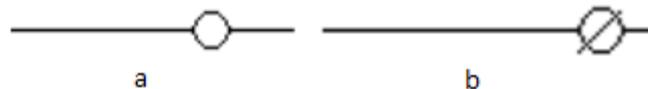
OR merupakan perintah yang digunakan apabila urutan kerja pada sistem kontrol lebih dari salah satu kondisi logikanya terpenuhi semuanya untuk memberikan satu output. Logika ini menyerupai kontak relay NO. Sedangkan OR NOT merupakan perintah yang digunakan apabila urutan kerja pada sistem kontrol lebih dari salah satu kondisi logikanya terpenuhi semuanya untuk memberikan satu output. Logika ini menyerupai kontak relay NC.



**Gambar 2. 4** OR (a) OR NOT (b).

#### 4. *OUT*

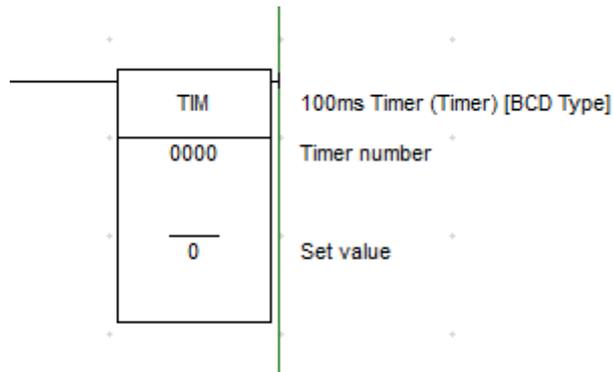
*OUT* merupakan simbol perintah, Jika kondisi logikanya terpenuhi maka perintah ini akan mengeluarkan satu output. Logika ini menyerupai kontak relay NO.



**Gambar 2. 5** *OUT*

#### 5. *Timer*

*Timer* merupakan perintah *countdown* (menghitung mundur) yang nilai awalnya ditentukan oleh program, *timer* mempunyai batas angka 0000 sampai 9999 dalam bentuk BCD dan dalam order 100ms.



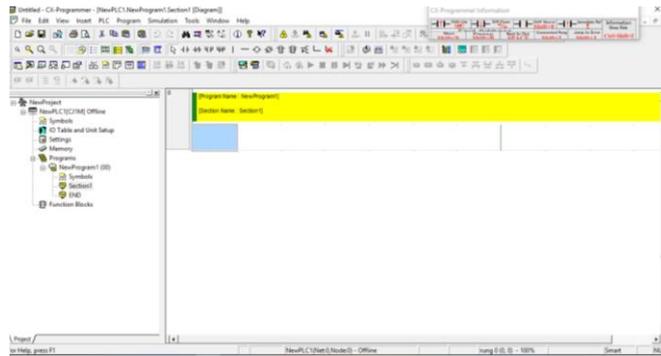
**Gambar 2. 6** Timer (a)

## 6. Binary Coded Decimal (BCD)

BCD adalah teknik pengkodean bilangan desimal, dimana masing-masing digit bilangan desimal tersebut direpresentasikan dengan bilangan biner. Dalam dunia elektronik dan komputer, pengkodean ini bertujuan untuk memudahkan pengkonversian data bilangan desimal ketika akan ditampilkan pada suatu *display* (seven-segment misalnya). Selain itu, BCD juga digunakan untuk mempercepat suatu proses kalkulasi. Dengan teknik pengkodean BCD ini, maka akan dapat dihindari kerumitan baik dari sisi hardware maupun software.

## 2.8 Cx-Programmer

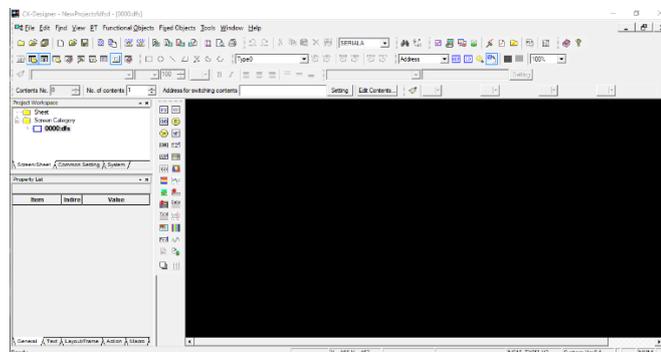
*CX Programmer Omron* merupakan sebuah software komputer yang digunakan dalam bidang pemrograman PLC yang berfungsi untuk membuat, memonitor dan merubah dari berbagai program PLC Omron. CX-Programmer dapat dijalankan dengan standar minimal komputer processor 486 MHz dengan sebuah sistem operasi Windows XP. [11]



**Gambar 2. 7** Gambar Awal Cx-Programmer

## 2.9 Cx-Designer

CX- Designer adalah sebuah *software* penampil data atau biasa disebut *human machine interface* yang dibuat oleh Omron yang berfungsi untuk menampilkan dan memvisualisasikan suatu kejadian, peristiwa, atau pun proses yang sedang terjadi di *plant* secara *real* (nyata) sehingga dengan memudahkan operator dalam melaksanakan pekerjaannya. [8]



**Gambar 2. 8** Gambar Awal Program Cx-Designer

## 2.10 LabView

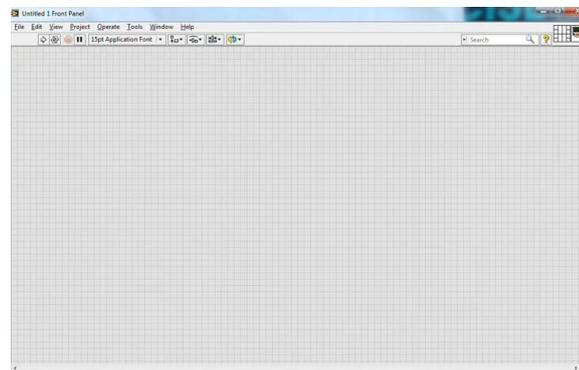
*Laboratory Virtual Instrument Engineering Workbench* (LabVIEW) merupakan produksi dari *National Instrument Company*. LabVIEW menggunakan bahasa program grafik atau G (*Grahical Programming Language*) dan tidak menggunakan

teks untuk membuat program seperti C++, Java, Basic, dan lain-lain. LabVIEW dapat digunakan untuk mengukur, menganalisa data, memberikan informasi nilai aktual, dan lain-lain. [9]

LabVIEW merupakan instrumentasi maya atau sistem instrumentasi berbasis Komputer. Untuk membangun sistem instrumentasi berbasis komputer (instrumentasi maya) di perlukan *hardware*: (Data akusisi (Daq-card), General Purpose Interface Bus (GPIB), komunikasi serial) dan *software*. Program *LabVIEW* dikenal sebagai *Virtual Instrument (VI)*. Untuk membuat VI terdiri dari:

### 1. *Front Panel*

Front Panel sebagai tampilan yang digunakan untuk berinteraksi dengan pengguna pada saat program berjalan. Disini menggunakan dapat melakukan kontrol, mengubah input dan indikator analisa data. [9]

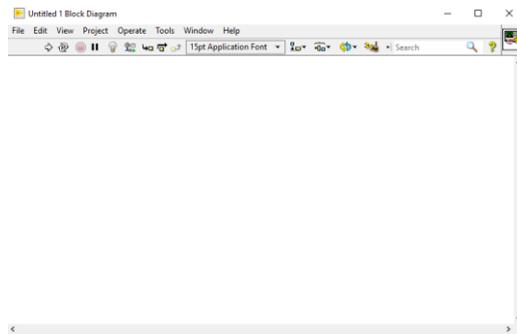


**Gambar 2. 9** Gambar front panel program LabView

Kontrol digunakan untuk mensimulasi instrumen input dan mengirimkan data ke blok diagram. Kontrol berupa pushbutton, knobs, dan lain-lain. Indikator digunakan sebagai output. Dapat berupa grafik, LED, dan *display* lainnya. Setiap kontrol dan indikator di *front panel* memiliki terminal yang saling berhubungan. [9]

## 2. Blok Diagram

Blok diagram berikan kode sumber bersifat grafis. Didalam blok terdapat blok diagram panel berbentuk terminal. [9]



**Gambar 2. 10** Gambar blok diagram program LabView