

2. TEORI PENUNJANG

Pada bab dua ini akan dijelaskan teori penunjang yang melandasi perancangan aplikasi SCADA sistem pada *plant* parkir mobil otomatis dengan menggunakan LabVIEW.

2.1. SCADA (*Supervisory Control and Data Acquisition*)

SCADA *system* adalah suatu metode dalam sistem kontrol, dimana operator dapat melakukan fungsi kontrol (*Controlling*), pengawasan (*Monitoring*) dan pengambilan serta perekaman data (*Data Acquisition*) dari sebuah sistem yang sedang berkerja. SCADA merupakan sebuah sistem yang mengumpulkan informasi atau data-data dari lapangan dan kemudian mengirimkannya ke sebuah komputer pusat yang akan mengatur dan mengontrol data-data tersebut. Sistem SCADA tidak hanya digunakan dalam proses-proses industri, misalnya, pabrik baja, pembangkit dan pendistribusian tenaga listrik (konvensional maupun nuklir), pabrik kimia, tetapi juga pada beberapa fasilitas eksperimen seperti fusi nuklir.

SCADA *system* mengacu pada kerja PLC, dimana pada PC akan ditunjukkan dan ditampilkan simulasi dan tombol kontrol pada *plant* secara *real-time* dari sistem dengan bantuan SCADA *software* (dalam hal ini menggunakan program LabView). Jadi PC akan memiliki fungsi untuk melakukan *controlling* dan *monitoring plant*. SCADA *software* didukung oleh fitur-fitur untuk menampilkan proses dari sistem dengan memanfaatkan *data acquisition*. Sedangkan untuk menghubungkan SCADA *software* dengan PC agar dapat dikontrol dan diamati oleh operator serta dengan PLC yang berkerja pada *plant*, maka dibutuhkan media komunikasi seperti jalur komunikasi serial pada PC (*Serial Port PC*).

Ada banyak bagian dalam sebuah sistem SCADA. Sebuah sistem SCADA biasanya memiliki perangkat keras sinyal untuk memperoleh dan mengirimkan I/O, kontroler, jaringan, antarmuka pengguna dalam bentuk HMI (*Human Machine Interface*), piranti komunikasi dan beberapa perangkat lunak pendukung. Semua itu menjadi satu sistem, istilah SCADA merujuk pada sistem

pusat keseluruhan. Sistem pusat ini biasanya melakukan pemantauan data-data dari berbagai macam sensor di lapangan atau bahkan dari tempat-tempat yang lebih jauh lagi (*remote locations*).

Sistem pemantauan dan kontrol industri biasanya terdiri dari sebuah *host* pusat atau *master* (biasa dinamakan sebagai *master station*, *master terminal unit* atau MTU), satu atau lebih unit-unit pengumpul dan kontrol data lapangan (biasa dinamakan *remote station*, *remote terminal unit* atau RTU) dan sekumpulan perangkat lunak standar maupun customized yang digunakan untuk memantau dan mengontrol elemen-elemen data di lapangan. Sebagian besar sistem SCADA banyak menggunakan komunikasi jarak jauh, walaupun demikian ada beberapa elemen menggunakan komunikasi jarak dekat. Ada dua elemen dalam aplikasi SCADA, yaitu:

- a. Proses, sistem, mesin yang akan dipantau dan dikontrol, bisa berupa *power plant*, sistem pengairan, jaringan komputer, sistem lampu trafik lalu-lintas atau *plant* apa saja;
- b. Sebuah jaringan peralatan ‘cerdas’ dengan *interface* ke sistem melalui *sensor* dan kontrol *output*. Jaringan yang merupakan sistem SCADA, akan mempermudah untuk melakukan pemantauan dan pengontrolan komponen-komponen sistem yang melalui *sensor* dan kontrol *output* tersebut.

SCADA dapat digunakan untuk mengatur berbagai macam peralatan. Biasanya sistem SCADA pada PLC digunakan untuk melakukan proses industri yang kompleks secara otomatis, dapat menggantikan tenaga manusia dan biasanya merupakan proses-proses yang melibatkan faktor-faktor kontrol yang lebih banyak, serta faktor-faktor kontrol gerakan-cepat yang lebih banyak, dan lain sebagainya. SCADA dapat digunakan dalam aplikasi-aplikasi yang membutuhkan kemudahan dalam pemantauan sekaligus juga pengontrolan, dengan berbagai macam media *interface* dan komunikasi yang tersedia saat ini. Berikut ini beberapa hal yang bisa dilakukan dengan Sistem SCADA:

- Mengakses pengukuran kuantitatif dari proses-proses yang penting, secara langsung saat itu maupun sepanjang waktu.
- Mendeteksi dan memperbaiki kesalahan secara cepat.

- Mengontrol proses-proses yang lebih besar dan kompleks dengan staf-staf terlatih yang lebih sedikit.

Sebuah sistem SCADA memberikan keleluasaan dalam mengatur maupun mengkonfigurasi sistem. Semakin banyak hal yang bisa dipantau, semakin detail operasi yang bisa dilihat, dan semuanya bekerja secara *real-time*. Sehingga sekompleks apapun proses yang ditangani oleh PLC, operator dari *plant* bisa melihat operasi proses dalam skala yang besar maupun kecil, dan operator bisa melakukan penelusuran jika terjadi kesalahan untuk meningkatkan efisiensi.

2.2. PLC (*Programmable Logic Controller*)

Dalam dunia industri automasi, proses kontrol pada mulanya dilakukan dengan menggunakan *relay conventional*. *Coil relay* masing-masing dihubungkan dengan sensor-sensor, sedangkan *output contact*-nya dihubungkan pada bagian mesin yang akan dikendalikan. Proses kontrol menggunakan *relay* seperti ini sangatlah rumit pada bagian *wiring* sehingga sangatlah sulit untuk memperbaiki sistem yang sudah ada. Selain itu, kontrol dengan menggunakan *relay* ini sangatlah terbatas. Oleh karena itu sekarang dibuat sebuah alat yang dapat menggantikan fungsi *relay* tersebut yaitu PLC (*Programmable Logic Controller*).

Mengutip dari buku panduan pengoperasian PLC Omron CPM1, *CQM1/CPM1 Programmable Controller Operation Manual*. USA: OMRON, Revised Februari 1998. *Programmable Logic Controller* diperkenalkan pertama kali pada tahun 1969 oleh Richard E. Morley yang adalah pendiri Modicon (*Modular Digital Controller*) corporation. Alat ini berbasis komputer, standar industri, dan menggantikan alat elektronik serta rangkaian yang mengontrol proses dan peralatan mesin-mesin.

PLC (*Programmable Logic Controller*) merupakan perkembangan dari pengontrol berbasis-mikroprocessor yang dapat diprogram untuk menyimpan instruksi-instruksi dan untuk mengimplementasikan fungsi-fungsi. PLC adalah suatu perangkat elektronik digital dengan *memory* yang dapat diprogram untuk menyimpan instruksi-instruksi yang menjalankan fungsi-fungsi khusus, seperti: logika, *timing*, *counting*, dan aritmatika untuk mengontrol mesin dan proses. PLC merupakan “komputer khusus” untuk aplikasi di industri, digunakan untuk

memonitor dan mengontrol proses industri untuk menggantikan rangkaian relai atau kontaktor, dan memiliki bahasa pemrograman sendiri. Berbeda dengan PC (*Personal Computer*), PLC sudah dilengkapi oleh *unit input* dan *output digital* yang bisa langsung dihubungkan ke perangkat luar seperti: *switch*, *limit switch*, sensor, *relay*, sedangkan PC tidak memiliki *unit input* dan *output digital* yang bisa langsung dihubungkan ke perangkat luar.

Kelebihan dari PLC adalah memiliki bentuk bahasa pemrograman yang sederhana dan intuitif. Selain itu PLC (*Programmable Logic Controller*) merupakan suatu *controller* dalam bidang automasi yang biasa digunakan untuk mengontrol suatu mesin.

PLC itu sendiri mempunyai definisi sebagai berikut :

- *Programmable* :

Artinya dapat diprogram (diubah-ubah) sesuai dengan program yang diinginkan, kemudian menyimpan program tersebut pada memori.

- *Logic* :

Artinya dapat memproses *input* secara aritmatik/mampu melakukan operasi matematika.

- *Control* :

Artinya dapat mengontrol dan mengatur suatu proses sehingga dapat menghasilkan *output* yang diinginkan.

Perancangan program dari PLC ini dapat dirancang dengan menggunakan *statement list* ataupun *ladder diagram*. Untuk memprogram suatu PLC, hal pertama yang harus dilakukan adalah program ditulis di PC dengan menggunakan program khusus PLC, setelah itu program yang sudah selesai dibuat itu di *download* ke PLC dengan menggunakan kabel serial yang merupakan sarana komunikasi antara PC dengan PLC. Setelah itu PLC yang sudah diprogram dapat bekerja sebagai pengontrol yang *independent*.

Karena harga dari PLC ini tergolong mahal, maka untuk menentukan jenis/tipe dari PLC yang akan digunakan sebaiknya memperhatikan kriteria-kriteria sebagai berikut :

- *Hardware*, yang meliputi jumlah *input/output* maksimum, tegangan operasi dari *input* dan *output*, indikator status *I/O*, jumlah *timer* dan *counter*, jumlah

flag, ukuran memori, jenis memori, manipulasi *bit/word*, waktu siklus per 1x *statements*, *interface* ke PC, kondisi sekitar (suhu, kelembaban, pendinginan, getaran, *supply* arus), *housing*, ukuran.

- *Software*, yang meliputi bahasa pemrograman, kualitas dokumentasi, kualifikasi kemampuan operator yang dibutuhkan.

Penggunaan PLC memungkinkan untuk mengubah suatu sistem kontrol tanpa harus terlebih dahulu mengubah instalasi yang sudah ada sebelumnya. Jika ingin mengubah jalannya proses maka yang harus diubah hanyalah program yang ada dalam memori PLC saja (tanpa harus menggubah *hardware* yang telah jadi). Penggunaan PLC memudahkan pemakai dalam melakukan instalasi dan dapat mempersingkat waktu untuk mengubah jalannya proses kontrol. PLC dapat bekerja pada lingkungan industri dengan kondisi yang cukup berat, seperti temperatur yang tinggi, pengaruh dari peralatan-peralatan lain yang berada di sekitarnya.

PLC yang diproduksi oleh berbagai perusahaan sistem kontrol terkemuka saat ini biasanya mempunyai ciri-ciri sendiri yang menawarkan keunggulan sistemnya, baik dari segi aplikasi (perangkat tambahan) maupun modul utama sistemnya. Meskipun demikian pada umumnya setiap PLC (sebagaimana komputer pribadi yang cenderung mengalami standarisasi dan kompatibel satu sama lain) mengandung empat bagian (piranti) berikut ini:

- a. Modul catu daya
- b. Modul *CPU*
- c. Modul perangkat lunak
- d. Modul *I/O*

2.2.1. Modul Catu Daya (*Power Supply*)

Power Supply berguna sebagai penyedia daya bagi PLC, tegangan pada *power supply* bisa berupa tegangan AC (120/240 V) dan tegangan DC (24 V). Daya tersebut digunakan untuk berbagai modul PLC lainnya selain modul tambahan yang digunakan sebagai *memory backup* pada PLC. Jadi seandainya *power supply* mati, dengan adanya *memory backup* tersebut, maka isi memori

akan tetap terjaga. PLC juga memiliki *power supply* (24V DC) internal yang bisa digunakan untuk menyediakan daya bagi *input/output* PLC

2.2.2. Modul CPU

Modul *CPU* (*Central Processing Unit*) yang disebut juga modul *controller* atau *processor* terdiri dari dua bagian:

- a. *Processor* berfungsi untuk :
 - Mengoperasikan dan mengkomunikasikan modul-modul PLC melalui *bus-bus* serial atau paralel yang ada.
 - Mengeksekusi program kontrol.
- b. Memori, berfungsi untuk:
 - Menyimpan informasi digital yang bisa diubah dan berbentuk tabel data, *register* citra, atau RLL (*Relay Ladder Logic*), yang merupakan program pengendali proses.

Jenis PLC yang akan digunakan adalah PLC OMRON CPM1 yang memiliki struktur memori sebagai berikut ini.

Tabel 2.1. Struktur Memori PLC OMRON CPM1

Data Area		Words	Bits	Fungsi
IR Area	<i>Input Area</i>	IR 000 - IR 009 (10 Words)	IR 00000 - IR 00915 (160 bits)	Bit – bit ini dapat dialokasikan ke terminal I/O diluar (<i>external</i>).
	<i>Output Area</i>	IR 010 - IR 019 (10 Words)	IR 01000 - IR 01915 (160 bits)	
	<i>Work Area</i>	IR 200 - IR 231 (32 Words)	IR 20000 - IR 23115 (512 bits)	Bit ini dapat digunakan dengan bebas.
SR Area		SR 232 – SR 255 (24 Words)	SR 23200 - SR 25507 (384 bits)	Bit ini disediakan untuk fungsi spesifik seperti <i>flag</i> dan <i>control bits</i> .
TR Area		----	TR 0 - TR 7 (8 bits)	Untuk menyimpan status On/Off saat <i>power</i> dimatikan.
HR Area		HR 00 - HR 19 (20 Words)	HR 0000 - HR 1915 (320 bits)	Untuk menyimpan data dan status On/Off saat <i>power</i> dimatikan.
AR Area		AR 00 to AR 15 (16 Words)	AR 0000 to AR 1515 (256 bits)	Menyediakan fungsi spesifik seperti <i>flags</i> dan <i>control bits</i> .
LR Area		LR 00 to LR 15 (16 Words)	LR 0000 to AR 1515 (256 bits)	Untuk data link 1:1 dengan PC yang lain.

Tabel 2.1. Struktur Memori PLC OMRON CPM1 (lanjutan)

<i>Timer / Counter Area</i>		TC 000 - TC 127 (<i>Timer / Counter numbers</i>)		<i>Timer dan Counter menggunakan penomoran yang sama.</i>
DM Area	<i>Read / Write</i>	DM0000 - DM0999 DM1022 - DM1023 (1002 Words)	----	DM Area data dapat diakses pada <i>word unit</i> saja. Digunakan untuk menyimpan waktu, kejadian, dan <i>error code</i> yang ditemukan. <i>Word-word</i> ini dapat digunakan sebagai DM <i>read/write</i> saat <i>error log</i> tidak digunakan. Nilai <i>Word</i> didapatkan ketika dalam keadaan mati.
	<i>Error Log</i>	DM 1000 - DM 1021 (22 Words)	----	Digunakan untuk menyimpan waktu dan <i>error code</i> yang terjadi. <i>Word-word</i> ini dapat digunakan sebagai DM <i>read/write</i> biasa ketika fungsi <i>error log</i> tidak dapat digunakan.
	<i>Read Only</i>	DM 6144 - DM 6599 (456 Words)	----	Tidak dapat ditulis dari program.
	<i>PC Setup</i>	DM 6600 - DM 6655 (56 Words)	----	Digunakan untuk menyimpan kondisi dari berbagai parameter yang berlainan yang dapat mengontrol kerja PC.

Sumber: OMRON (1996, p.111)

Pada PLC tertentu terkadang dijumpai pula adanya beberapa *processor* dalam satu modul sekaligus yang ditujukan untuk mendukung kehandalan sistem. Beberapa *processor* tersebut bekerja sama dengan suatu prosedur tertentu untuk meningkatkan kinerja pengendalian.

2.2.3. Modul Perangkat lunak / *Software*

Software merupakan modul yang pasti dimiliki oleh PLC. PLC mampu mengenali berbagai macam bahasa pemrograman / perangkat lunak seperti *State Language*, bahasa pemrograman C, dan yang paling populer digunakan adalah

RLL (*Relay Ladder Logic*). Sistem yang akan digunakan untuk mengontrol plant, dibuat dalam bahasa pemrograman yang *compatible* dengan PLC yang digunakan. Semua instruksi yang telah ditulis disimpan dalam memori PLC, dan akan dieksekusi oleh modul *CPU*. Penulisan program pada PLC dapat dilakukan pada keadaan *on line* maupun *off line*.

2.2.4. Modul I/O

Modul *I/O* merupakan modul masukan (*input*) dan modul keluaran (*output*) yang bertugas mengatur hubungan PLC dengan piranti *eksternal* atau *peripheral* yang bisa berupa suatu komputer *host*, saklar-saklar, unit penggerak motor, dan berbagai macam sumber sinyal yang terdapat dalam *plant*.

a. Modul Input

Modul *input* berfungsi untuk menerima sinyal dari unit pengindera *peripheral*, dan memberikan pengaturan sinyal, terminasi, isolasi, maupun indikator keadaan sinyal *input*. Sinyal-sinyal dari piranti *peripheral* akan di-*scan* dan keadaannya akan dikomunikasikan melalui modul antarmuka dalam PLC.

Beberapa jenis tegangan masukan pada PLC OMRON CPM1:

- Tegangan masukan *DC* (24V).
- Tegangan *AC* (120/240V)
- Masukan *TTL* (5V)

b. Modul Output

Modul *output* mengaktifasi berbagai macam piranti seperti *actuator hydrolic*, *pneumatic*, *solenoid*, *starter motor*, dan tampilan status titik-titik *peripheral* yang terhubung dalam sistem. Fungsi modul *output* lainnya mencakup *conditioning*, terminasi dan juga pengisolasian sinyal-sinyal yang ada. Proses aktivasi itu tentu saja dilakukan dengan pengiriman sinyal-sinyal diskret dan analog yang *relevan*, berdasarkan watak PLC sendiri yang merupakan piranti digital. Tegangan keluaran pada PLC OMRON CPM1 adalah tegangan *DC* (24V).

2.3. Protokol Komunikasi

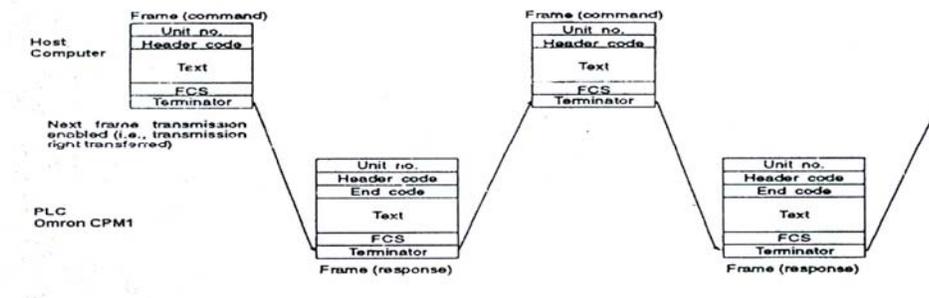
Host link Unit berfungsi untuk menjembatani PC dalam memonitor status pengoperasian dan lokasi data dari PLC. Dalam hal ini kita menggunakan

OMRON CPM1. Parameter komunikasinya sudah ter-*setting* standar yaitu :

- *Baud rate* = 9.600 bps (*bit per second*)
- Nomor *start bits* = 1
- Panjang data = 7 *bits*
- *Event (vertical) parity* = 1 *bits*
- Nomor *stop bits* = 2

PC di-*set* melalui pemrograman atau penulisan kode, yaitu dengan men-*set property* kontrol komunikasi. *Setting* parameter kecepatan transmisi dan format data pada *properti* harus sama dengan pengesetan pada *Host link Unit*.

Dengan adanya *host link communication* pada PLC Omron CPM1 satu *host computer* dapat mengontrol sekitar 32 PLC. Prosedur pada saat komunikasi data antara *host computer* dengan PLC Omron CPM1 adalah sebagai berikut:



Gambar 2.1. Blok diagram prosedur komunikasi antara *host* komputer dengan PLC Omron CPM1

Sumber : OMRON (1996, p.237)

Penjelasan dari Paket/frame dari data *Command* adalah sebagai berikut:

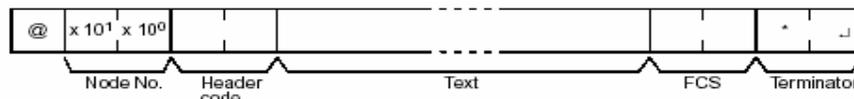
- a. *Unit no.* : nomor pengenalan identitas dari PLC.
- b. *Header Code* : kode khusus yang akan menentukan jenis operasi dan alamat tujuan, dan penunjuk operasi yang akan dilakukan oleh PLC.
- c. *Text* : berisi data *word* yang akan dikirimkan ke PLC.
- d. FCS (*Frame Check Sequence*) : prosedur pengecekan kesalahan pada *frame* data.
- e. *Terminator* : penanda dari akhir *frame* data, dan harus diberikan * dan ←

Penjelasan dari Paket/frame data *Response* adalah sebagai berikut:

- a. *Unit no.* : nomor pengenalan identitas dari PLC.
- b. *Header Code* : kode khusus yang akan menentukan jenis operasi dan alamat tujuan, dan penunjuk operasi yang akan dilakukan oleh PLC.
- c. *End Code* : kode yang menunjukkan pengiriman berjalan baik atau tidak.
- d. *Text* : berisi data *word* yang akan dikirimkan ke PLC.
- e. FCS (*Frame Check Sequence*) : prosedur pengecekan kesalahan pada *frame* data.
- f. *Terminator* : penanda dari akhir *frame* data, dan harus diberikan * dan ↵

Agar dapat membaca ataupun menulis, satu rangkaian data harus dikirim dengan bentuk paket terstruktur yang disebut *frame*. Masing-masing lokasi data atau memori data mempunyai bentuk *frame* yang berbeda.

Berikut contoh penulisan *Command Frame*:



Gambar 2.2. Satu *command frame* data

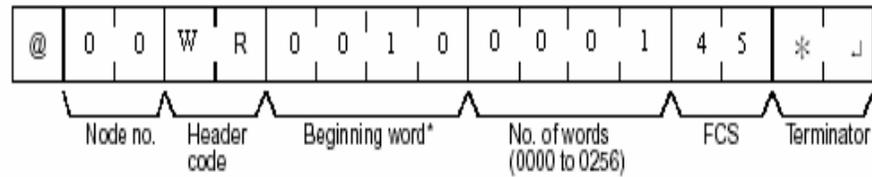
Sumber: OMRON (1996, p.237)

Penjelasan dari *Frame* di atas adalah sebagai berikut:

- a. @ : kode awal yang harus diberikan
- b. *Node No.* : nomor sebagai identitas PLC
- c. *Header Code* :
 - penunjuk operasi yang dilakukan (*READ/WRITE*)
 - penanda *area memori* PLC
- d. *Text* : *word* yang dituju dan jumlahnya
 - Read : *word* awal yang dituju dan jumlah *word*
 - Write : *word* awal yang dituju dan data yang akan ditulis
- e. FCS (*Frame Check Sequence*) : untuk mengecek ada tidaknya kesalahan pada *frame* data
- f. *Terminator* : harus diberikan * dan ↵

Berikut ini adalah contoh perintah menulis pada alamat IR 010 sebanyak 1 word:

Command Format



Gambar 2.3. Contoh *command frame* untuk menulis IR

Sumber: OMRON (1996, p.237)

Tabel 2.2. Contoh *Header Code*

Header Code	PC Mode			Name
	RUN	MON	PRG	
RR	Valid	Valid	Valid	IR/SR Area Read
RC	Valid	Valid	Valid	PV Read
WR	Not Valid	Valid	Valid	IR/SR Area Write
WD	Not Valid	Valid	Valid	DM Area Write

Sumber: OMRON (1996, p.314)

Setelah PC mengirimkan *Command Frame*, maka PLC akan memberi jawaban atau respon *frame* yang mirip dengan *Command Frame*. Namun, pada *respon frame* terdapat *end code* yang berfungsi sebagai penanda komunikasi berjalan normal atau tidak, dan jika *End Code* bernilai 00 berarti transmisi berjalan normal, berikut adalah gambar respon *frame* seperti berikut:



Gambar 2.4. Contoh *respon frame*

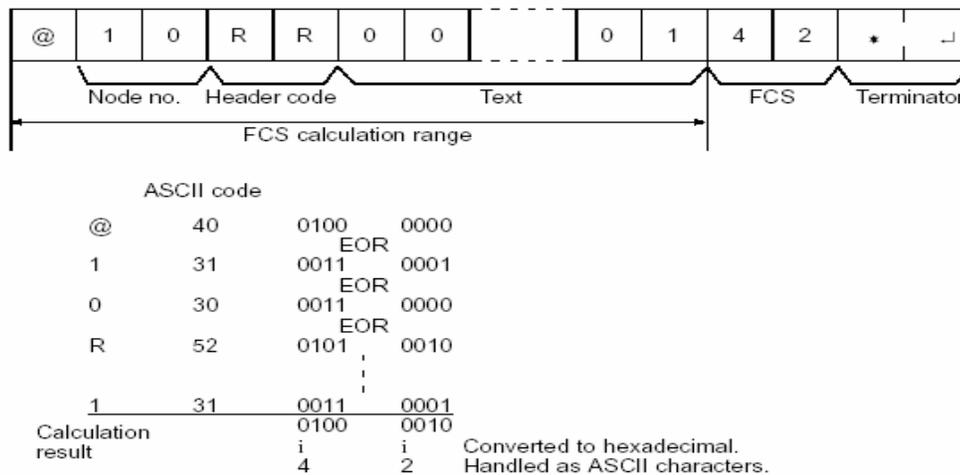
Sumber: OMRON (1996, p.237)

Tabel 2.3. Contoh *End Code*

<i>End Code</i>	Contents	Probable Cause	Corrective Measures
00	Normal Completion	---	---
13	FCS Error	The FCS is wrong. Either the FCS calculation is mistaken or there is adverse influence from noise.	Check the FCS calculation method. IF there was influence from noise, transfer the command again.

Sumber: OMRON (1996, p.382)

Ketika sebuah *frame* dikirim, sebuah kode FCS (*Frame Check Sequence*) ditempatkan sebelum *terminator* untuk mengecek apakah terjadi *error* pada saat transmisi dan untuk mengetahui adanya kesalahan *frame* data yang diterima PLC. FCS merupakan data 8 bit yang dikonversikan ke dalam 2 karakter ASCII. Data 8 bit tersebut merupakan hasil dari penjumlahan *Exclusive Or* yang dilakukan data dari awal hingga akhir *frame* sebelum FCS. Jadi mula-mula mengubah masing-masing *character* dari *character @* sampai data *text* dalam bentuk ASCII, kemudian melakukan operasi XOR satu sama lain mulai dari *character @*, sampai batas FCS, kemudian hasil akhirnya diubah ke dalam bentuk Heksa Desimal.



Gambar 2.5. Contoh perhitungan kode FCS

Sumber: OMRON (1996, p.318)

2.4. Komunikasi Serial RS 232

RS 232 merupakan *interface* paling umum yang digunakan pada komunikasi serial. RS 232 yang dikenalkan pada tahun 1962 ini sering digunakan diberbagai industri dan otomasi. Spesifikasi transmisi data dari *transmitter* ke *receiver* rata-rata lamban dan jarak transmisinya pendek. RS 232 populer karena harganya yang murah, dan dapat menggunakan kabel yang lebih panjang dibandingkan komunikasi menggunakan paralel. *Channel-channel independent* dibuat untuk komunikasi dua arah (*full-duplex*). Sinyal RS 232 diwakili oleh tegangan sistem umum, dan menspesifikasikan protokol komunikasi, dimana dapat bekerja baik dalam komunikasi *point to point* pada rata-rata transmisi data rendah. Suatu perangkat dapat menggunakan *port* yang ada pada komputer, atau perlu tambahan *port*, namun kebanyakan PC mempunyai *interface* RS 232, dan Port RS 232 pada PC merupakan *single device*. Sinyal RS 232 membutuhkan *ground* antara PC dan peralatan yang terhubung. Sedangkan jarak kabel harus dibatasi 1 sampai 200 kaki pada data *asynchronous* dan sekitar 50 kaki dengan data *synchronous*. Data *synchronous* mempunyai jam *transmite* dan *receive* yang membatasi jarak maksimum pada saat menggunakan suatu jalur data *synchronous*. *Port* RS 232 didesain untuk berkomunikasi dengan peralatan yang mendukung 1 *transmitter* dan 1 *receiver*. Pada umumnya RS 232 digunakan untuk komunikasi dua arah.

Duplex adalah suatu metode pengoperasian rangkaian komunikasi antara dua peralatan. *Full-Duplex*, memungkinkan kedua unit untuk mengirim dan menerima secara bersamaan atau serentak. *Half-Duplex*, memungkinkan suatu unit untuk mengirim informasi pada suatu saat meskipun sambungan mungkin mampu untuk melakukan transmisi dua arah, namun komunikasi terjadi secara bergantian.

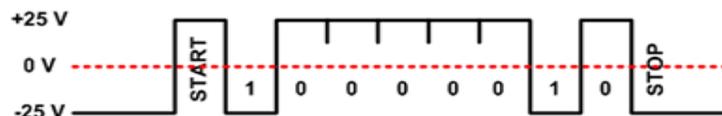
2.4.1. Karakteristik Sinyal Port Serial

Standar sinyal komunikasi serial yang banyak digunakan adalah RS 232 yang dikembangkan oleh *Electronic Industry Association and the Telecommunication Industry Association* (EIA/TIA) yang pertama kali dipublikasikan pada tahun 1962. Hal ini terjadi jauh sebelum IC TTL populer,

sehingga sinyal ini tidak ada hubungannya sama sekali dengan level tegangan IC TTL. Standar ini hanya menyangkut komunikasi data antara komputer (*Data Terminal Equipment – DTE*) dengan alat-alat perlengkapan komputer (*Data Circuit-Terminating Equipment – DCE*) saja. Standar sinyal serial RS 232 memiliki ketentuan level tegangan sebagai berikut:

- Logic '1' disebut 'mark' terletak antara -3 Volt hingga -25 Volt.
- Logic '0' disebut 'space' terletak antara +3 Volt hingga +25 Volt.
- Daerah tegangan antara -3 Volt hingga +3 Volt adalah *invalid level*, yaitu daerah tegangan yang tidak memiliki *level logic* pasti, sehingga harus dihindari. Demikian juga, *level* tegangan lebih negatif dari -25 Volt atau lebih positif dari +25 Volt juga harus dihindari karena *level* tegangan tersebut dapat merusak *line driver* pada saluran RS 232.

Gambar 2.6 adalah contoh level tegangan RS232 pada pengiriman huruf "A" dalam format ASCII tanpa bit paritas.

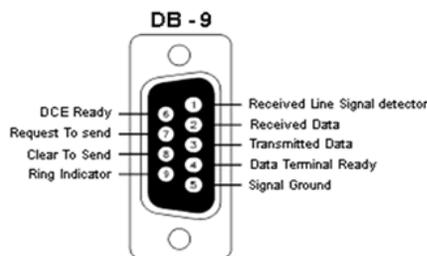


Gambar 2.6. Level tegangan RS232 pada pengiriman huruf "A"

Sumber : Prasetia, Widodo (2004, p. 131)

2.4.2. Konfigurasi Port Serial

Konektor DB-9 pada bagian belakang komputer adalah port serial RS232 yang biasa dinamai dengan COM1 dan COM2. Gambar 2.7 menunjukkan konektor *female* port serial DB 9.



Gambar 2.7. Konektor *female* port serial DB 9

Sumber : Prasetia, Widodo (2004, p. 131)

Keterangan mengenai fungsi saluran RS232 pada konektor *female* DB-9 :

- a. *Received Line Signal Detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa pada terminal masukan ada data masuk.
- b. *Receive Data*, digunakan DTE menerima data dari DCE.
- c. *Transmit Data*, digunakan DTE mengirimkan data ke DCE.
- d. *Data Terminal Ready*, DTE memberitahukan kesiapan terminalnya.
- e. *Signal Ground*, saluran *ground*
- f. *DCE ready*, sinyal aktif pada saluran ini menunjukkan bahwa DCE sudah siap.
- g. *Request to Send*, dengan saluran ini DCE diminta mengirim data oleh DTE.
- h. *Clear to Send*, dengan saluran ini DCE memberitahukan bahwa DTE boleh mulai mengirim data.
- i. *Ring Indicator*, pada saluran ini DCE memberitahukan ke DTE bahwa sebuah stasiun menghendaki hubungan dengannya.

Tabel 2.4. Konfigurasi Pin dan Nama Sinyal Konektor Serial DB 9

Pin	Nama	Keterangan	Arah	Fungsi
1	DCD	<i>Data Carrier Detect</i>	Masuk	Saat mendeteksi suatu ' <i>carrier</i> ' maka sinyal ini akan aktif.
2	RxD	<i>Receive Data</i>	Masuk	Untuk penerimaan data serial.
3	TxD	<i>Transmit Data</i>	Keluar	Untuk pengiriman data serial.
4	DTR	<i>Data Terminal Ready</i>	Keluar	Siap/tidak terjadi hubungan komunikasi.
6	DSR	<i>Data Set Ready</i>	Masuk	Siap/tidak melakukan pertukaran data.
7	RST	<i>Request to Send</i>	Keluar	Informasi siap/tidak pertukaran data.
8	CTS	<i>Clear to Send</i>	Masuk	Siap/tidak melakukan pertukaran data.
9	RI	<i>Ring Indicator</i>	Masuk	Aktif jika mendeteksi sinyal dering.

Sumber : Prasetya, Widodo (2004, p. 133)

2.4.3. Transmisi Data Pada RS232

Komunikasi pada RS-232 dengan PC adalah komunikasi asinkron. Pada komunikasi serial asinkron tidak diperlukan clock karena data dikirimkan dengan kecepatan tertentu yang sama baik pada pengirim / penerima. Masing-masing data disinkronkan menggunakan *clock* internal pada tiap-tiap sisinya. Kecepatan transmisi (baud rate) dapat dipilih bebas dalam rentang tertentu. Baud rate yang

umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/perderti). Dalam komunikasi data serial, baud rate dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya harus ditentukan panjang data (6,7 atau 8 bit), paritas (genap, ganjil, atau tanpa paritas), dan jumlah bit “Stop” (1, 1 ½, atau 2 bit). Bentuk data *Asynchronous*:

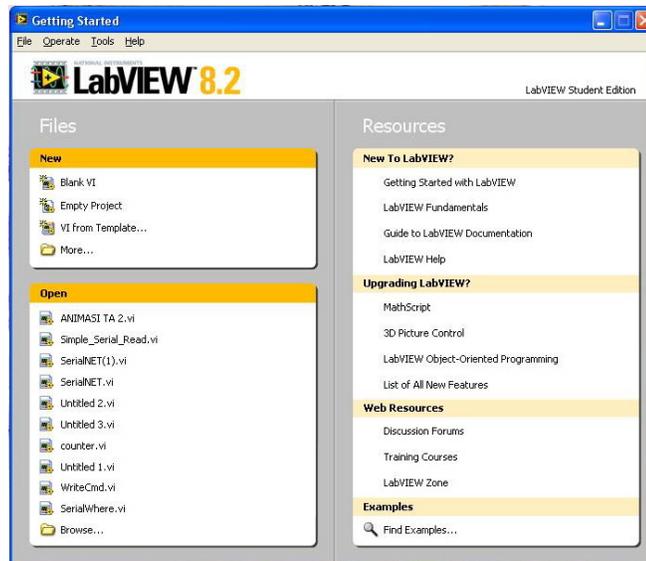
- a. Logic ‘1’ disebut ‘mark’ terletak antara -3 Volt hingga -25 Volt.
- b. Logic ‘0’ disebut ‘space’ terletak antara +3 Volt hingga +25 Volt.
- c. T (bit time) : Lama pengiriman 1 bit data.
- d. *Start bit* : bit sebagai penanda mulainya pengiriman data (selalu *low*).
- e. *Data bit* : bit – bit data.
- f. *Parity bit* : bit untuk mengecek kesalahan pengiriman data.
 - *Odd parity* (bernilai 1) = Jumlah data bit + parity bit = ganjil.
 - *Even parity* (bernilai 0) = Jumlah data bit + parity bit = genap.
- g. *Stop bit/stop interval* : panjangnya waktu sebelum interval berikutnya, untuk menjamin kondisi kembali ke MARK (sebelum start bit karakter berikutnya memberi kondisi SPACE), dan Besarnya adalah : 1/1.5/2 kali dari T (bit time).

2.5. National Instruments LabVIEW

National Instruments LabVIEW adalah sebuah *graphical programming environment* terbuka yang ditetapkan oleh standar industri untuk aplikasi-aplikasi pengujian, pengukuran, dan otomasi. Sebuah perkembangan pada pemrograman grafik, dimana teknisi dapat menggunakannya untuk mendisain suatu sistem menyerupai bentuk sistem aslinya, dapat melakukan pengamatan dengan hasil yang maksimal, dan dapat mengontrol suatu aplikasi dengan *programmable automation controllers* (PACs). *LabVIEW* adalah sebuah sistem pemrograman yang terbuka dan fleksibel, sehingga teknisi dapat menghubungkannya dengan alat lainnya, seperti *programmable logic controllers* (PLCs), dan *programmable automation controllers* (PACs) menggunakan satu paket *software*.

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) adalah suatu bahasa pemrograman yang menggunakan berbagai macam ikon yang merepresentasikan suatu instruksi. Jika bahasa pemrograman *text-based* mengeksekusi instruksi sesuai dengan urutan yang ditulis, *LabVIEW*

menggunakan metode *dataflow programming*, dimana alur data melalui berbagai ikon akan menentukan urutan eksekusi dari setiap instruksi. Dalam LabVIEW, VI atau *Virtual Instruments*, adalah program yang menyerupai instrumen yang sesungguhnya. Gambar 2.8 adalah tampilan awal dari program LabVIEW saat dibuka.



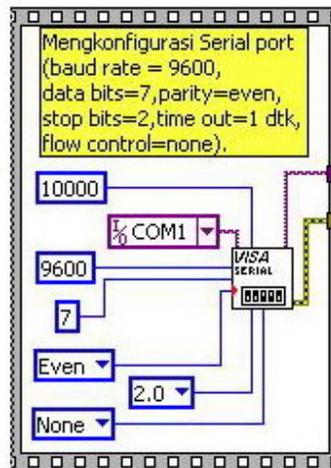
Gambar 2.8. Tampilan awal program LabVIEW

Karena fleksibilitasnya, sifatnya yang modular, dan kemudahan dalam pemrogramannya, LabVIEW telah menjadi sebuah bahasa pilihan untuk penggunaan laboratorium dalam universitas-universitas teratas di seluruh dunia. Ilmuwan-ilmuwan dan insinyur yang menjalankan penelitian dan pengajaran dapat memanfaatkan fungsionalitas I/O yang mudah bersama dengan dasar-dasar kemampuan-kemampuan analisa yang kuat. Instruktur-instruktur juga dapat menggunakan LabVIEW di ruang-ruang kelas untuk memecahkan masalah.

LabVIEW juga membantu teknisi untuk semakin mudah dalam mengaplikasikan sistem *programmable logic controller* (PLC) dengan cara penggabungan *personal computers* (PC) pada aplikasi mereka dengan bantuan perlengkapan *human machine interface* (HMI), atau *supervisory control and data acquisition* (SCADA). Dengan LabVIEW teknisi dapat memrogram *human machine interface* (HMI) dan *logic* pada daerah program yang sama, sehingga

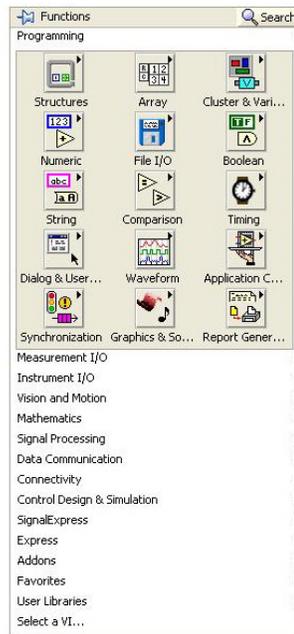
dapat meminimalkan biaya pembuatan dan waktu pembelajaran, dan dapat memaksimalkan ketrampilan pemrograman.

LabVIEW menawarkan sebuah variasi baru dalam dunia pemrograman pada mesin, pada kejadian *discrete*, pada aliran data yang dinamis, pada waktu yang kontinyu dan pada kontrol dengan rangkaian *logic*. LabVIEW mulai beranjak meninggalkan sifat lama dari bahasa pemrograman tradisional dan mulai menggunakan fitur-fitur dari sebuah daerah pemrograman grafik yang lebih mudah digunakan, termasuk di dalamnya terdapat semua peralatan yang dibutuhkan untuk *data acquisition* (DAQ), analisis data, dan hasil dari presentasi. Dengan bahasa pemrograman grafik seperti di atas, yang disebut bahasa pemrograman “G”, teknisi dapat memrogram dengan menggunakan sebuah blok diagram grafik, dimana blok diagram tersebut dapat dijalankan dan diubah kedalam kode mesin. Gambar 2.9 adalah contoh hasil pemrograman dengan menggunakan LabVIEW.



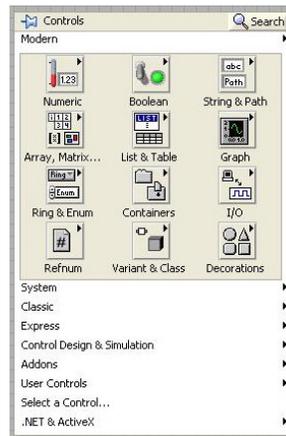
Gambar 2.9. Contoh bahasa pemrograman LabVIEW

Gambar 2.10 adalah *Functions* pada bahasa pemrograman LabVIEW untuk merancang program pada LabVIEW.



Gambar 2.10. *Functions* pada bahasa pemrograman LabVIEW

Gambar 2.11 adalah *controls* pada bahasa pemrograman LabVIEW untuk merancang tampilan program pada LabVIEW.



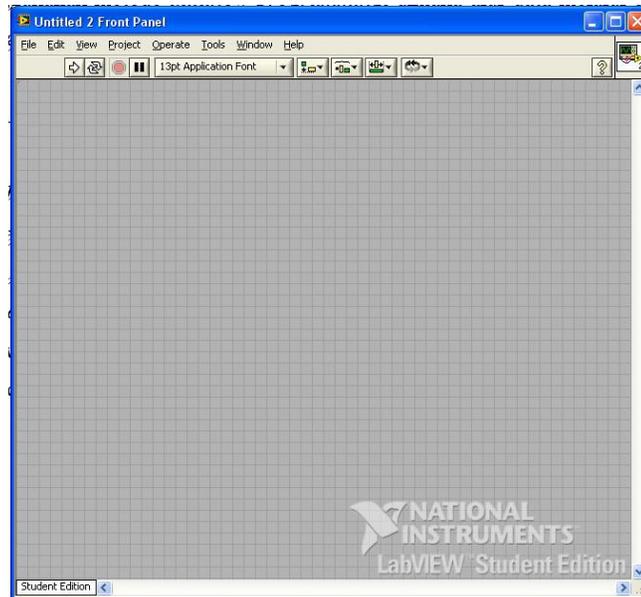
Gambar 2.11. *Controls* pada bahasa pemrograman LabVIEW

LabVIEW mulai digunakan pertama kali di dalam laboratorium dan sampai sekarang masih tetap dipakai di beberapa laboratorium. Mulai dari penelitian yang besar sampai perkembangan laboratorium di seluruh dunia (seperti di Lawrence Livermore, Argonne, Batelle, Sandia, Laboratorium Jet Propulsion,

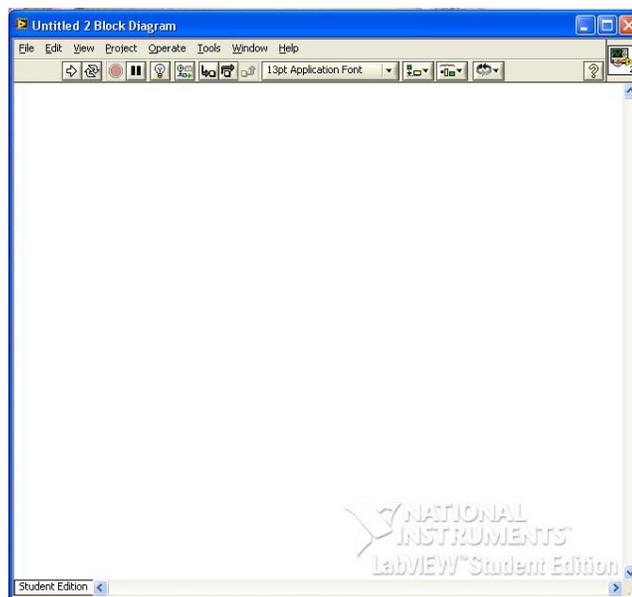
White Sands, Oak Ridge di Amerika, dan CERN di Eropa). LabVIEW dipakai juga pada laboratorium R&D di beberapa industri, dan dipakai juga untuk kepentingan laboratorium belajar-mengajar di beberapa universitas seluruh dunia, terutama pada disiplin ilmu elektronika, teknik mesin dan fisika.

Perangkat pemrograman untuk *supervisory control and data acquisition* (SCADA) pada LabVIEW dapat digunakan untuk memperoleh data dari sebuah proses, untuk menampilkan secara grafik kondisi sekarang dan yang lalu dari sebuah variabel proses pada beberapa layar komputer dan dapat mencetak hasilnya, serta dapat mengontrol jalannya proses dari sistem. Perangkat pemrograman ini telah menjadikannya sebuah program yang handal, karena telah menyediakan fungsi pemrograman SCADA juga pada program LabVIEW. Selain itu LabVIEW juga memiliki *real time data base* (RTDB) agar program dari sistem lain juga dapat mengakses data yang dihasilkan dari proses eksternal sensor dan aktuator yang berupa arus. LabVIEW juga dapat diakses oleh beberapa jenis alat eksternal, seperti PLC, *data acquisition card*, GPIB atau *serial port* pada *personal computer* (PC).

Dalam program LabVIEW, seorang *programmer* yang akan membuat program, harus membuat dua bagian utama pada program LabVIEW, yaitu bagian *front panel* program dan bagian *block diagram* program. Bagian *Front panel* program adalah tampilan yang merupakan *interface* dengan pengguna pada PC, dimana pengguna dapat mengontrol program yang sedang dijalankan, dapat mengganti *input*, dan dapat melihat data maupun indikator yang terus menerus di-*update*. Bagian *Block diagram* program adalah bagian dimana urutan instruksi dari program yang berupa ikon-ikon program dibuat untuk menjalankan program tersebut. Gambar 2.12 dan 2.13 adalah tampilan dari bagian *front panel* program dan bagian *block diagram* program pada bahasa pemrograman LabVIEW.



Gambar 2.12. *Front panel* pada LabVIEW



Gambar 2.13. *Block diagram* pada LabVIEW