

## **BAB II**

### **LANDASAN TEORI**

#### **1.1 Simulasi**

Menurut The Oxford American Dictionary (1980) yang dikutip oleh Harrell, C., Ghosh, B. K., & Bowden, R (2004) di dalam buku *Simulation Using Promodel* simulasi didefinisikan sebagai cara untuk mereproduksi kondisi situasi dengan menggunakan model, untuk pembelajaran, pengujian atau pelatihan. Model yang digunakan adalah model komputer. Lebih lanjut Harrel mengatakan bahwa simulasi dapat didefinisikan sebagai imitasi dari sistem dinamis menggunakan model komputer untuk mengevaluasi dan meningkatkan kinerja sistem. Namun seiring perkembangannya simulasi memiliki definisi yang beragam. Berikut ini adalah beberapa definisi simulasi menurut beberapa ahli.

Menurut Emshoff dan Simun (1970) simulasi adalah sebagai suatu model sistem dimana komponennya di presentasikan oleh profesor-profesor aritmatika dan logika yang dijalankan komputer untuk memperkirakan sifatsifat dinamis sistem tersebut. Menurut Shannon (1975) simulasi merupakan proses perencanaan model dari sistem nyata yang dilanjutkan dengan pelaksanaan eksperimen terhadap model untuk mempelajari perilaku sistem sistem atau evaluasi strategi. Menurut Banks dan Carson (1984) simulasi adalah tiruan sistem nyata yang dikerjakan secara manual oleh komputer dan kemudian diobservasi dan disimpulkan untuk mempelajari krakterteristik sistem. Menurut Law dan Kelton (1991) simulasi adalah sekumpulan metode dan aplikasi untuk menirukan atau merepresentasikan

perilaku dari suatu sistem nyata, yang biasanya dilakukan pada komputer dengan menggunakan perangkat lunak tertentu. Menurut Khosnevis (1994) simulasi merupakan proses aplikasi membangun model dari sistem nyata atau usulan sistem, melakukan eksperimen dengan model dari sistem nyata atau usulan sistem, melakukan eksperimen dengan model tersebut untuk menjelaskan perilaku sistem, mempelajari kinerja sistem atau untuk membangun sistem baru sesuai dengan kinerja yang diinginkan[3].

## **2.2 Sistem *Monitoring***

*Monitoring* adalah proses pengumpulan dan analisis informasi berdasarkan indikator yang ditetapkan secara sistematis dan kontinu tentang kegiatan/ program sehingga dapat dilakukan tindakan koreksi untuk penyempurnaan program/ kegiatanitu selanjutnya. *Monitoring* adalah pemantauan yang dapat dijelaskan sebagai kesadaran (*awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan kearah tujuan atau menjauh dari itu[4]. *Monitoring* akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan.

Proses *monitoring* adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan yang focus pada proses dan keluaran [5]. *Monitoring* memiliki beberapa tujuan, yaitu:

1. Mengkaji apakah kegiatan- kegiatan yang dilaksanakan telah sesuai dengan rencana.
2. Mengidentifikasi masalah yang timbul agar langsung dapat diatasi
3. Melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan kegiatan.
4. Mengetahu ikaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan.
5. Menyesuaikan kegiatan dengan lingkungan yang berubah, tanpa menyimpang dari tujuan.

### **2.3 Perangkat lunak (*Software*) LabVIEW untuk simulasi**

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) adalah bahasa pemrograman grafis yang menggunakan ikon bukan baris teks untuk membuat aplikasi. Software ini digunakan khusus untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali dan instrumentasi, serta otomatisasi industry. Berbeda dengan bahasa pemrograman berbasis teks, di mana petunjuk menentukan eksekusi program. LabVIEW menggunakan dataflow pemrograman, di mana aliran data menentukan eksekusi[6]. LabVIEW dapat digunakan untuk mengontrol aliran data antara komputer dan instrumen, memiliki banyak fitur penting untuk pemrosesan sinyal dan analisis, dan membantu mahasiswa untuk

mudah dan cepat mengerti bagaimana instrumen virtual dapat digunakan di tempat instrumentasi fisik. Pengembangan komputer pribadi dan workstation yang kuat telah mengubah cara fisikawan, insinyur dan ilmuwan lainnya bekerja.



NI LabVIEW 2016 (64-bit)

App

**Gambar 2.1** Software LabVIEW

Beberapa kelebihan LabVIEW dibandingkan dengan bahasa pemrograman lainnya adalah:

1. Bahasa pemrograman LabVIEW jelas dan mudah dipahami, karena berbentuk grafis, dengan instruksi berbentuk ikon-ikon, yang dihubungkan dengan garis/kawat untuk menunjukkan aliran data, mirip seperti *flowchart*.
2. Pembuatan program mudah, yaitu hanya dengan menarik keluar ikon instruksi yang sudah tersedia dipalet (kotak instruksi), dan menghubungkannya dengan kawat ke ikon lain. Kawat ini sama seperti *variable* pada bahasa pemrograman teks. Dengan cara ini, LabVIEW menyederhanakan pemrograman, karena kawat hanya akan terhubung apabila tipe datanya sesuai sehingga menghilangkan kebutuhan manajemen memori dan deklarasi tipe data setiap variabel seperti dalam bahasa pemrograman teks. Juga tidak perlu mengingat nama-nama instruksi, karena semua ditampilkan pada palet. Jadi, pembaca

hanya perlu mencarinya dari kategori yang disediakan, atau dengan menggunakan bantuan tombol *Search* untuk menemukannya.

3. Karena mudah dipahami dan mudah dibuat, maka mempersingkat waktu pembuatan program. Begitu pula untuk perbaikan program-programnya, karena dibuat dalam bentuk gambar yang bersifat interaktif, maka perbaikan programnya menjadi lebih cepat yang memungkinkan pengembangan program menjadi lebih baik.
4. Dari tahun 1986 hingga sekarang, LabVIEW telah memiliki integrasi dengan ribuan *hardware* dan ratusan library yang siap digunakan untuk aplikasi dibidang instrumentasi, pengolah sinyal, analisis dan visualisasi data serta koneksi ke internet.
5. Telah terbukti di industri sebagai *software* yang handal, *powerful*, dan fleksibel, yang dapat digabungkan dengan *library* eksternal dari *software* lain, seperti *MATLAB*, *Simulink*, *SPICE*, *ADAMS*, *SolidWorks*, *Lego Mindstorms*.
6. Menjembatani dunia pendidikan dengan industri, karena *software* yang digunakan sama, sehingga transisi dan transfer teknologi dari duni pendidikan ke industri menjadi lebih mudah.
7. LabVIEW didesain sebagai sebuah bahasa pemrograman parallel (*multicore*) yang mampu menangani beberapa instruksi sekaligus dalam waktu bersamaan. Hal ini sangat sulit dilakukan dalam bahasa pemrograman teks, karena biasanya bahasa pemrograman teks mengeksekusi instruksi secara berurutan perbaris, satu demi satu. Dengan LabVIEW, pengguna dapat membuat aplikasi

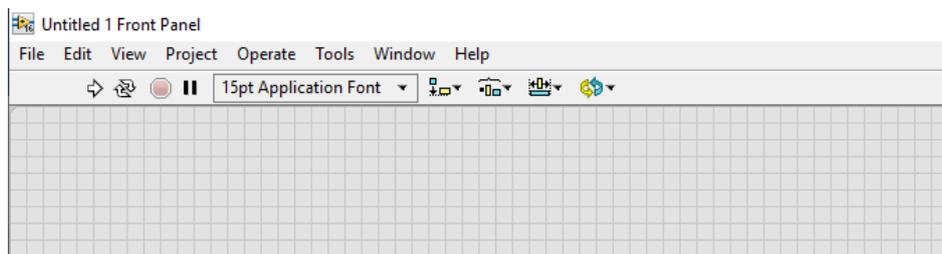
eksekusi paralel ini secara mudah dengan menempatkan beberapa struktur loop secara terpisah dalam block diagram.

8. Sifat modular LabVIEW memungkinkan pengguna untuk membuat program yang kompleks dan rumit menjadi sederhana, yaitu dengan cara membuat subprogram, atau di LabVIEW disebut SubVI. Ikon-ikon di dalam LabVIEW sebenarnya merupakan SubVI. Beberapa subVI dapat digabungkan menjadi subVI. *SubVI-subVI* gabungan tersebut dapat digabungkan lagi menjadi sebuah *subVI* lain. Demikian seterusnya dengan tingkat hierarki yang tidak terbatas.
9. Satu alat untuk berbagai bidang, meliputi otomotif, biomedis, komunikasi data, *energy*, *control*, *vision*, dengan penggunaan mulai dari perencanaan pengukuran, *prototype*, pengujian, hingga implementasi dan pengembangannya[7].

Adapun *Software* LabVIEW terdiri dari beberapa komponen utama yaitu:

### 2.3.1 Front Panel

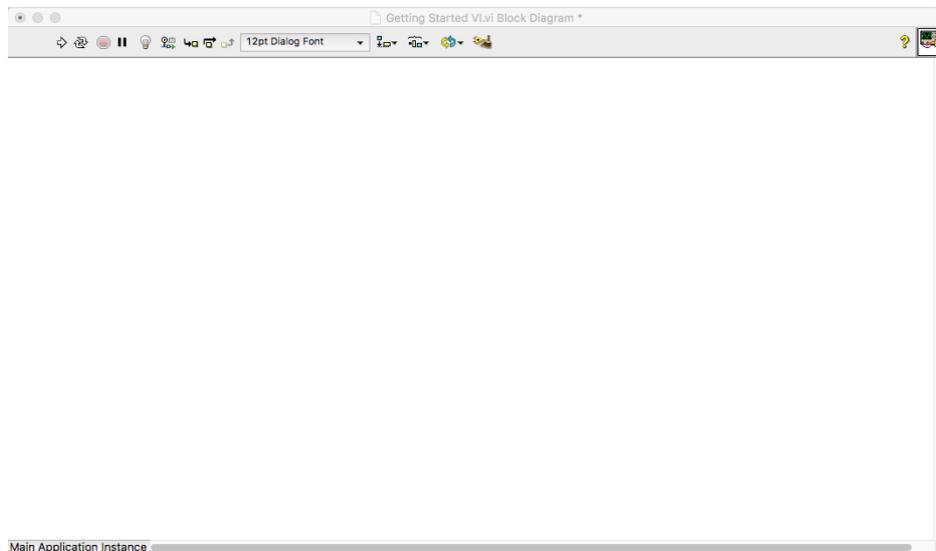
*Front panel* adalah bagian *window* yang berlatar belakang abu-abu serta mengandung kontrol dan indikator seperti yang ada di **Gambar 2.2**. *Front panel* digunakan untuk membangun sebuah VI, menjalankan program dan *mendebug* program.



**Gambar 2.2** *Front Panel*

### 2.3.2 Block Diagram dari VI

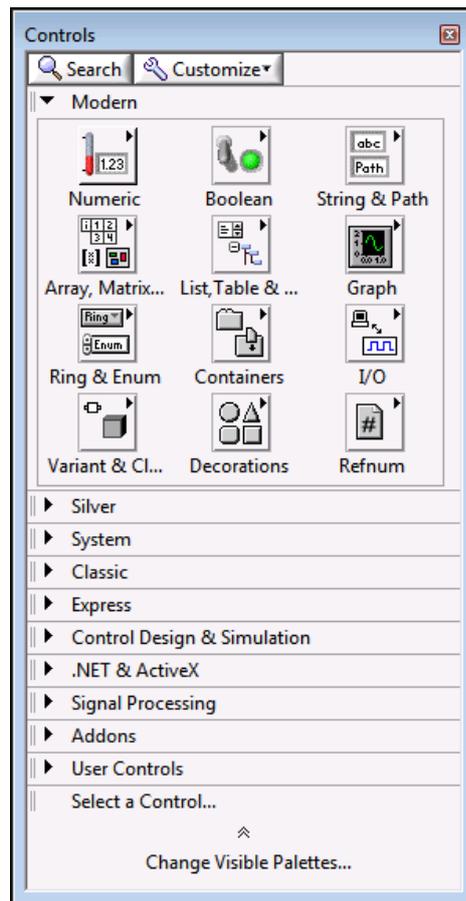
Blok diagram adalah bagian window yang berlatar belakang putih berisi *source code* yang dibuat dan berfungsi sebagai instruksi untuk *front panel* seperti yang ada di **Gambar 2.5**.



**Gambar 2.3** Block Diagram dari VI

### 2.3.3 Control Pallate

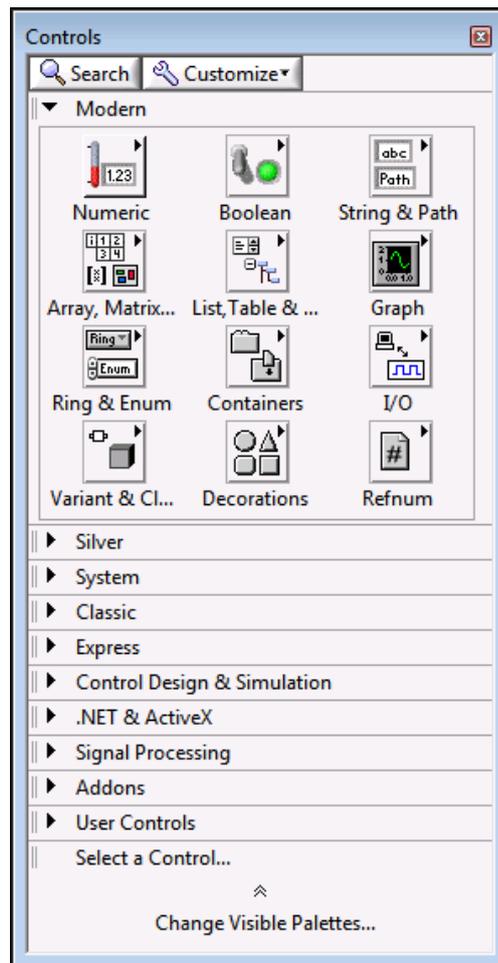
*Control pallette* merupakan tempat beberapa kontrol dan indikator pada *front panel*, *control pallette* hanya tersedia di *front panel* seperti yang ada pada **Gambar 2.6**. Untuk menampilkan *control pallette* dapat dilakukan dengan mengklik *windows >> show control pallette* atau klik kanan *front panel*.



**Gambar 2.4** Control Pallate

### 2.3.4 Function Pallate

*Function pallette* digunakan untuk membangun sebuah block diagram, *function pallette* hanya tersedia pada blok diagram seperti pada **Gambar 2.6**, untuk menampilkannya dapat dilakukan dengan mengklik *windows >> show control pallette* atau klik kanan pada lembar kerja block diagram[7].



**Gambar 2.5** Function Pallate

## 2.4 Mean

Rata-rata atau mean adalah nilai khas yang mewakili sifat tengah atau posisi pusat dari kumpulan nilai data. Terdapat beberapa ukuran yang termasuk mean, diantaranya[8]:

### 2.4.1 Mean aritmatik

Mean aritmatik atau sering disebut dengan mean dinotasikan dengan  $\bar{x}$ .

Mean aritmatik untuk data tidak berkelompok dirumuskan sebagai berikut:

$$\bar{x} = (x_1 + x_2 + \dots + x_n) \quad (2.1)$$

Jika dinotasikan dengan notasi sigma, maka rumus diatas menjadi

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.2)$$

Keterangan :

$\bar{x}$  : rata-rata hitung

$x_i$  : nilai sampel i

$n$  : jumlah sampel

#### 2.4.2 Mean geometric

Selain mean aritmatik, suatu penelitian terkadang menggunakan ukuran mean geometrik atau rata-rata ukur yang dinotasikan dengan MG. Mean geometrik cocok digunakan untuk menghitung perubahan return pada periode serial dan kumulatif (misalnya 5 atau 10 tahun berturut-turut). Mean geometrik untuk data return dapat dirumuskan sebagai berikut[8]:

$$M_G = \prod_{i=1}^n (1 + R_i)^{1/n} - 1 \quad (2.3)$$

dengan

$R_i$  = return saham ke-i

$n$  = banyaknya data pengamatan

MG = mean geometrik

Penambahan nilai 1 pada rumus tersebut berguna untuk menghilangkan nilai negatif dalam perhitungan rata-rata geometrik.

## 2.5 Sensor Infra Merah

*Infra red* (IR) detektor atau sensor infra merah adalah komponen elektronika yang dapat mengidentifikasi cahaya infra merah (infra red, IR). Sensor infra merah atau detektor infra merah saat ini ada yang dibuat khusus dalam satu module dan dinamakan sebagai IR Detector Photomodules. IR Detector Photomodules merupakan sebuah chip detektor inframerah digital yang di dalamnya terdapat fotodiode dan penguat (amplifier)[9].

Konsep dasar dari sensor inframerah yang digunakan sebagai detektor adalah memancarkan sinyal infra merah, sinyal infra merah ini memantulkan dari permukaan suatu benda dan sinyal tersebut diterima di penerima infra merah. Prinsip kerja sensor *infrared* adalah ketika sensor memancarkan radiasi, ia mencapai objek dan sebagian radiasi dipantulkan kembali ke penerima IR. Berdasarkan intensitas penerimaan oleh penerima IR, keluaran dari sensor ditentukan[10].



**Gambar 2.6** *infra red* (IR)

## 2.6 Arduino UNO

Arduino Uno adalah board yang memiliki 14 pin digital input/output, 6 analog input, sebuah resonator keramik 16MHz, koneksi USB, colokan power input, ICSP header, dan sebuah tombol reset. Arduino uno merupakan sebuah mikrokontroler yang dapat membuat program untuk mengendalikan berbagai komponen elektronika dan fungsi Arduino Uno ini dibuat untuk memudahkan peneliti dalam melakukan prototyping, memprogram mikrokontroler, membuat alat-alat canggih berbasis mikrokontroler[11].



**Gambar 2.7** Arduino UNO [11]