BAB II

LANDASAN TEORI

Pada bab ini berisi tentang pembahasan teori yang mendasari penyelesaian tugas akhir penulis. Landasan teori yang digunakan antara lain, *Wayfinding*, *Microsoft Windows OS*, *National Instruments LabVIEW*, *Google Drive*, *Winguard*, *Symantec Ghost*, algoritma *Levensthein Distance*, metode *N-Gram*, metode *Bayesian Hybrid*, USB wifi adaptor, basis data, jaringan internet, mikrokontroller, dan *Real Time Clock* (RTC).

1.1 *Wayfinding*

Secara etimologi, istilah *wayfinding* terdiri dari dua kata, yaitu *way* dan *find*. Menurut www.*dictionary*.com, salah satu pengertian kata *way* adalah "*a direction or vicinity; a path or course leading from one place to another*" suatu arah, suatu jalan yang menuntun dari satu tempat ke tempat lain. Sedangkan kata *find* merupakan kata kerja dari kegiatan menemukan atau menempatkan, "*to locate, attain, or obtain by search or effort*". Secara istilah *wayfinding* dapat diartikan sebagai penemuan arah dari satu tempat ke tempat lain [2].

Dari sini dapat disimpulkan bahwa *wayfinding* merupakan suatu cara dimana manusia bernavigasi dan mengorientasikan diri dalam ruang fisik. Proses tersebut mencakup usaha manusia memperoleh, menetapkan, dan menemukan serangkaian jalur yang akan dilalui untuk berangkat dari satu tempat ke tempat lain.

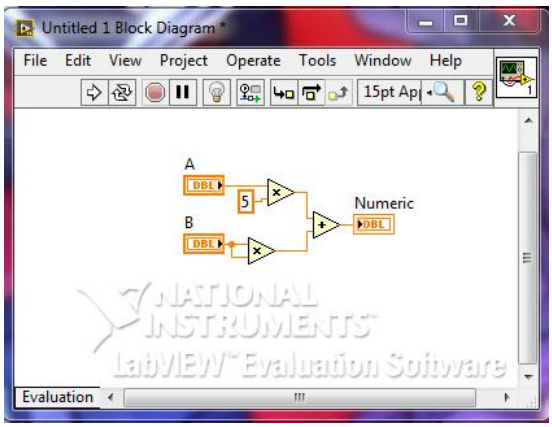
1.2 *Microsoft Windows OS*

Microsoft Windows adalah salah satu jenis *operating system* (OS). *Microsoft Windows* adalah OS milik Mirosoft Corporation. Versi terbaru dari *Microsoft Windows* saat ini adalah Windows 10 yang rilis pada 29 Juli 2015. Windows 10 merupakan langkah pertama dari generasi baru dari Windows. Windows 10 membuka pengalaman baru bagi pelanggan untuk bekerja, bermain dan terhubung. Windows 10 mewujudkan apa yang pengguna inginkan.

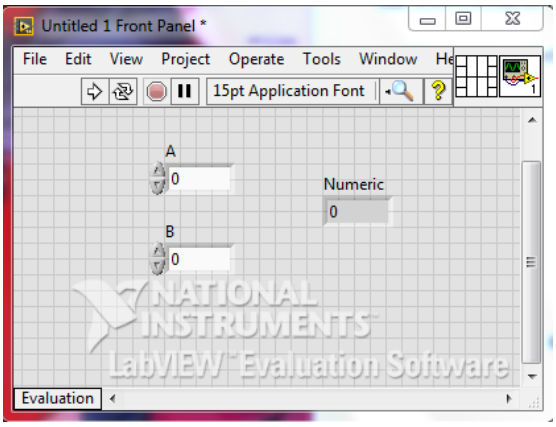
1.3 *National Instruments* LabView

LabVIEW adalah sebuah *software* pemrograman yang diproduksi oleh *National instruments* dengan konsep yang berbeda. *LabVIEW* adalah bahasa pemrograman grafis di mana program yang ditulis menggunakan gambar, bukan teks. Program LabView dikenal dengan sebutan VI (*Visual Instruments*). Sebuah VI terdiri dari tiga bagian utama yaitu *Front Panel*, *Block Diagram*, dan *Icon/Connector* [3].

Sebuah VI terdiri dari tiga bagian utama yaitu *Front Panel*, *Block Diagram*, dan *Icon/Connector*. Berikut di bawah ini contoh *block diagram* dan *front panel* untuk program sederhana.



Gambar 2.1 Contoh *Block Diagram* LabVIEW



Gambar 2.2 Contoh *Front Panel* LabVIEW

1.4 *Google Drive*

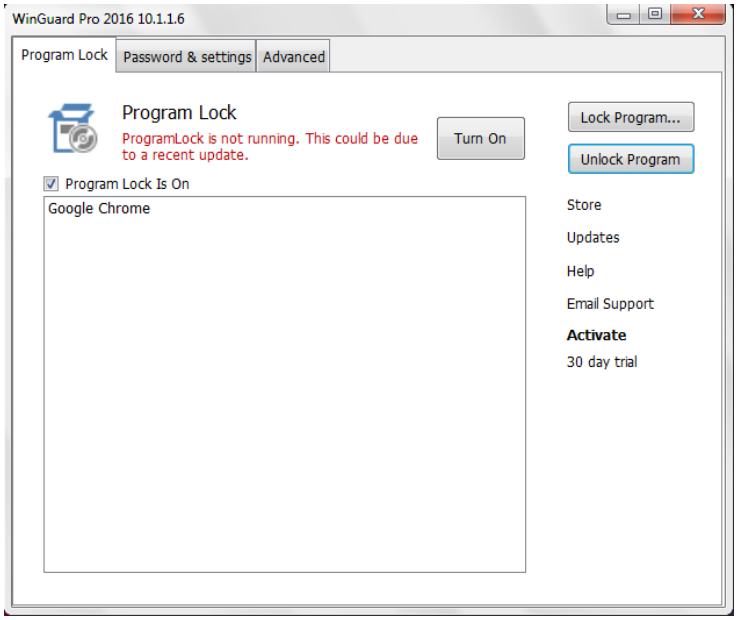
Google Drive merupakan sebuah layanan penyimpanan data *online* dari Google. *Google drive* sendiri menyediakan penyimpanan gratis sebesar 15GB dan mampu menyimpan file apa saja didalamnya. Selain berfungsi sebagai media penyimpanan *google drive* juga bisa langsung membuka file yang tersimpan tanpa harus mengunduhnya terlebih dahulu dan memiliki fitur edit file untuk jenis-jenis file tertentu. Google drive juga dapat terhubung langsung ke desktop pengguna sehingga pengguna mampu mengunduh ataupun mengunggah file tanpa melalui browser.



Gambar 2.3 Logo Google Drive

1.5 *Winguard*

Winguard adalah salah satu *security program* yang mampu mengamankan *software* pada komputer dengan *password*. *Winguard* berfungsi menghindarkan dari terjadinya pencurian data pribadi dan program-program tanpa harus menutup atau mengunci komputer, juga memungkinkan untuk menjalankan program dan file terbuka bahkan ketika data atau program sedang terkunci. Berikut di bawah ini contoh tampilan *Winguard Pro 2016*.



Gambar 2.4 Tampilan *Winguard Pro 2016*

1.6 *Symantec Ghost*

Symantec Ghost adalah *software cloning disk* dan juga *backup* data *hardisk* secara keseluruhan maupun hanya salah satu partisinya saja. Data yang di-*clone* mulai dari *registry*, *driver*, dan sebagainya. Jika *clone hardisk* tidak sesuai dengan komputer maka kemungkinan besar akan mengalami *bluescreen*. Sehingga 1 *hardisk* hasil *cloning* hanya cocok pada komputer yang memiliki spesifikasi yang sama.

1.7 Algoritma *Levenshtein Distance*

Levenshtein distance adalah suatu matriks untuk mengukur jumlah perbedaan antara dua string[4]. *Levenshtein distance* dua buah *string* adalah jumlah minimum operasi yang dibutuhkan untuk mengubah satu string (*source string*) menjadi string yang lain (*target string*), dimana suatu operasi melibatkan penyisipan (*insertion*), penghapusan (*deletion*), dan penggantian (*substitution*) dari suatu karakter tunggal. *Levenshtein distance* sering digunakan pada aplikasi untuk menentukan seberapa mirip atau berbedanya dua buah string, seperti aplikasi pengecekan suatu ejaan, atau yang biasa dikenal dengan *spell checkers*. *Levenshtein distance* melibatkan penggunaan matriks berukuran (n+1) x (m+1), dimana n dan m adalah panjang dari dua buah *string*. Algoritma *levenshtein distance* dapat diterapkan pada sebuah *search engine*. Dalam metode ini, dilakukan suatu penghitungan perbedaan antara dua *string*. Penghitungan tersebut meliputi tiga operasi *string* seperti di bawah ini. Untuk contoh yang akan digunakan,

diasumsikan *S* adalah *string* sumber (*source string*) pencarian dan *T* adalah *string* yang ingin dicari (*target string*).

a. Operasi Penyisipan (*Insertion*)

Misalnya *S* = *rima* dan *T* = *terima*. Operasi sisip dilakukan dengan menyisipkan *t* dan *e* pada posisi 1 dan 2 yang dapat ditunjukkan sebagai berikut.

Tabel 2.1 Operasi Penyisipan

	1	2	3	4	5	6
T	t	e	r	i	m	a
S	-	-	r	i	m	a
Sisipan	t	e				

b. Operasi Penghapusan (*Deletion*)

Misalnya *S* = *jaringan* dan *T* = *jangan*. Penghapusan dilakukan untuk karakter *r* dan *i* pada posisi 3 dan 4. Dua operasi penghapusan tersebut menunjukkan transformasi *S* ke *T* yang diilustrasikan sebagai berikut.

Tabel 2.2 Operasi Penghapusan

	1	2	3	4	5	6	7	8
T	j	a	-	-	n	g	a	n
S	j	a	r	i	n	g	a	n
Penghapusan			r	i				

c. Operasi Penggantian (*Substitution*)

Misalnya *S* = *peramal* dan *T* = *pejabat*. String *T* ditransformasikan menjadi *S* dengan melakukan penggantian (subtitusi) pada posisi 3, 5, dan 7. Huruf *j, b*

dan *t* pada *S* digantikan dengan *r, m* dan *l* pada *T*. Prosesnya dapat ditunjukkan sebagai berikut.

Tabel 2.3 Operasi Penggantian

	1	2	3	4	5	6	7
T	p	e	r	a	m	a	l
S	p	e	j	a	b	a	t
Penggantian			r		m		l

Ketiga operasi di atas dapat digunakan dalam contoh berikut ini. Misalnya *T* = *abrakadabra* dan *S* = *avrakhadabah*. *S* dapat ditransformasikan menjadi *T* sebagai berikut.

Tabel 2.4 Operasi Campuran

	1	2	3	4	5	6	7	8	9	10	11	12	13
T	a	b	R	a	k	-	a	d	a	b	r	a	-
S	a	v	R	a	k	h	a	d	a	b	-	a	h
		b				h					r		h

Pada dasarnya, algoritma ini menghitung jumlah minimum pentransformasian suatu *string* menjadi *string* lain yang meliputi penggantian, penghapusan, dan penyisipan. Algoritma ini digunakan untuk mengoptimalkan pencarian tersebut karena sangat tidak efisien jika dilakukan pencarian setiap kombinasi operasi - operasi *string* tersebut. Oleh karena itu, algoritma ini tergolong program dinamis dalam pencarian nilai minimal tersebut. Dalam algoritma ini, dilakukan penyeleksian panjang kedua *string* terlebih dahulu. Jika salah satu atau kedua *string* merupakan *string* kosong, jalannya algoritma ini berhenti dan memberikan hasil *edit distance* yang bernilai nol atau panjang *string* yang tidak

kosong. Jika panjang *string* keduanya tidak nol, setiap *string* memiliki sebuah karakter terakhir, misalnya c1 dan c2. Misalnya bagian *string* pertama tanpa c1 adalah s1 dan bagian *string* kedua tanpa c2 adalah s2, dapat dikatakan penghitungan yang dilakukan adalah cara mentransformasikan s1+c1 menjadi s2+c2. Jika c1 sama dengan c2, dapat diberikan nilai *cost* 0 dan nilai *edit distance*-nya adalah nilai *edit distance* dari pentransformasian s1 menjadi s2. Jika c1 berbeda dengan c2, dibutuhkan pengubahan c1 menjadi c2 sehingga nilai *cost*-nya 1. Akibanya, nilai *edit distance*-nya adalah nilai *edit distance* dari pentransformasian s1 menjadi s2 ditambah 1. Kemungkinan lain adalah dengan menghapus c1 dan mengedit s1 menjadi s2+c2 sehingga nilai *edit distance*-nya dari pentransformasian s1 menjadi s2+c2 ditambah 1. Begitu pula dengan penghapusan c2 dan mengedit s1+c1 menjadi s2. Dari kemungkinan-kemungkinan tersebut, dicarilah nilai minimal sebagai nilai *edit distance*.

Misalkan *S* = erlena dan *T* = arlene. Maka dengan berpatokan pada kesamaan *string* berurutan "rlen" pada keduanya dihasilkan transformasi dari *S* ke *T*.

Tabel 2.5 Contoh Kasus

	1	2	3	4	5	6
T	a	r	l	e	n	e
S	e	r	l	e	n	a
	a					e

String s ditransformasikan menjadi *t* dengan melakukan 2 penggantian. Sehingga jumlah perubahannya adalah 2. Algoritma di atas dapat menentukan

jumlah perubahan yang diperlukan yang minimal, yaitu 2. Pemerolehan nilai *edit distance* ini dapat ditunjukkan sebagai berikut.

Tabel 2.6 Pencarian Nilai *Edit Distance*

		e	r	l	e	n	a
	0	1	2	3	4	5	6
a	1	1	2	3	4	5	5
r	2	2	1	2	3	4	4
l	3	3	2	1	2	3	3
e	4	3	3	2	1	2	3
n	5	4	4	3	2	1	2
e	6	5	5	4	3	2	**2**

Jaraknya adalah nilai yang berada pada pojok kanan-bawah matriks yaitu 2. Artinya diperlukan 2 proses substitusi untuk mengubah string "erlena" menjadi string "arlene". Dengan demikian dapat diketahui nilai *Levenshtein Distance*-nya adalah 2.

1.8 *USB Wifi* Adaptor

USB wifi adaptor adalah sebuah alat yang digunakan untuk dapat menangkap sinyal *wifi* di daerah yang memiliki wifi *hotspot* atau *hotspot zone*. *Usb wifi* ini dapat dipergunakan untuk *Personal Computer* (PC) atau laptop. Jika PC atau laptop belum dilengkapi dengan penangkap sinyal *wifi* di dalamnya, maka *usb wifi* adaptor bisa menjadi solusinya. *USB wireless* ini memerlukan power 5V dari *USB Port*. Cara menggunakan *USB wifi* adapter cukup mudah, namun juga memiliki kelebihan dan kekurangan. Kelebihannya adalah bentuknya praktis dan mudah dilepas sehingga pengoperasiannya mudah. Kekurangannya jangkauan alat

ini tidak terlalu luas dan kemampuan kinerja serta daya pancarnya sangat tergantung pada bentuk antena.

Wireless adapter adalah secara umum dibedakan menjadi wireless adapter untuk PC dan *wireless adapter* untuk gadget selain PC seperti *notebook*, PDA dan perangkat lainnya. *Wireless adapter* untuk PC umumnya menggunakan slot PCI. Namun untuk komputer *desktop* bisa dibantu menggunakan *Personal Computer Memory Card International Association* (PCMCIA) *card*. Alat ini bisa digunakan dengan cara menambah satu *holder* sehingga memerlukan biaya tambahan.

Wireless adapter adalah untuk gadget non PC bisa menggunakan PCMCIA *card* tanpa bantuan *holder* tambahan.

1.9 Basis data (*Database*)

Basis data atau *database* adalah kumpulan data yang saling berelasi yang disimpan dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasikan data, menghindari duplikasi data.

1.10 Jaringan Internet

Internet adalah suatu jaringan komputer global yang terbentuk dari jaringan-jaringan komputer lokal dan regional yang memungkinkan komunikasi data antar

komputer yang terhubung ke jaringan tersebut. Internet awalnya merupakan suatu rencana dari Departemen Pertahanan Amerika Serikat *(US Departement of Defense)* pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan *Advance Research Project Agency Network* atau ARPANET yang melakukan riset tentang cara menghubungkan komputer satu dengan komputer lainnya agar bisa saling berkomunikasi.

Pada tugas akhir ini jangkauan geografis jaringan internet yang dipakai adalah *local area network* (LAN) yang hanya mencakup wilayah kecil dengan radius 1 km. Jaringan LAN menggunakan teknologi IEEE 802.3 Ethernet yang mempunyai kecepatan transfer data hingga 1GB/s. Berdasarkan distribusi sumber data tugas akhir ini menggunakan jeringan terpusat yang terdiri dari komputer *client* dan komputer *server,* di mana komputer *client* bertugas sebagai perantara dalam mengakses sumber data yang berasal dari komputer *server*. Dan berdasarkan media transmisi data yang digunakan adalah jaringan nirkabel, yang menggunakan gelombang elektromagnetik sebagai media transmisi datanya.

1.11 Mikrokontroller

Mikrokontroller dapat di artikan sebagai salah satu komponen terpenting dalam suatu sistem yang membutuhkan pengontrolan, dengan ukuran mikro yang tak ubahnya seperti sebuah komputer dalam ukuran mini, yang terdiri dari sebuah *Central Processing Unit* (CPU) sebagai pengolah data, memori

untuk penyimpan data dan sarana input/output (I/O) yang terangkum dalam sebuah chip.

Jenis-jenis mikrokontroller terbagi menjadi 3 keluarga besar yaitu keluarga MCS51, AVR dan PIC, pada umumnya jenis AVR yang sering digunakan, karena melimpahnya *tutorial* penggunaannya dan mudah untuk memprogramnya, AVR di kelompokkan menjadi 4 kelas, yaitu ATTiny, AT90Sxx, ATMega dan AT86RFxx. Banyak sistem minimum yang chip pemogramannya menggunakan AVR khususnya jenis ATMega, contohnya Arduino yang secara universal sudah banyak yang memakainya untuk *controller* sebuah sistem.

1.12 Real Time Clock

Komponen Real Time Clock (RTC) adalah komponen penghitung yang dapat difungsikan sebagai penyimpan data waktu sebagai sumber data waktu baik berupa data jam, hari, bulan, maupun tahun. RTC memerlukan baterai sebagai suplai daya pada chip, sehingga jam akan tetap *up-to-date* walaupun komputer dimatikan.