



**PENERAPAN DIGITAL SIGNATURE DENGAN
ADLER32 PADA SURAT MASUK DAN SURAT
KELUAR**

Disusun dan Diajukan Untuk Memenuhi Persyaratan Ujian
Akhir Memperoleh Gelar Sarjana Komputer
Fakultas Sains Dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA : DANI SAFITRI
N.P.M : 1514370801
Program Studi : SISTEM KOMPUTER

**PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020**

ABSTRAK
DANI SAFITRI
PENERAPAN DIGITAL SIGNATURE DENGAN ALDER32 PADA SURAT
MASUK DAN SURAT KELUAR.
2020

Untuk mengetahui latar belakang keabsahan sebuah surat pada saat ini legalisasi menggunakan surat pengesahan dipilih menjadi salah satu alternatif, di dalam menghadapi kebutuhan validasi yang cepat untuk proses legalisasi surat. Namun di dalam legalisasi tersebut harus memiliki keamanan yang baik. Seperti integritas data, autentikasi entitas, autentikasi asal data, dan antipenyangkalan. Dimana hal tersebut bertujuan untuk mengatasi jenis ancaman modifikasi dan pemalsuan surat legalisasi digital. Untuk dapat membuat surat legalisasi yang baik, tentunya dapat menggunakan metode Digital Signature. Untuk memberikan keamanan data yang maksimum untuk keabsahan suatu digital signature, pada penelitian ini penulis akan meneliti mengenai metode sebuah konfersi bilangan dari *Adller32dec* ke *Adler32hex* untuk dapat mengamankan suatu data yang dibuat dengan digital signature. Adapun metode pembuatan digital signautre yang dirancang untuk diterapkan dalam keabsahann suatu surat masuk dan surat keluar dengan menerapkan algoritma adler32. Dimana sebuah data berbentuk surat dijadikan keabsahan.

Kata Kunci : Digital Signature, *ADLER32*, Surat masuk dan Surat Keluar.

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
DAFTAR LAMPIRAN	vi

BAB I PENDAHULUAN

1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3

BAB II LANDASAN TEORI

2.1 Digital Signature.....	5
2.2 Fungsi Hash	7
2.3 Adler32	8
2.4 <i>Processing Image</i>	14
2.5 <i>Proposed Work</i>	15
2.6 Surat.....	17
2.7 Bahasa Pemrograman	18

2.8	Aplikasi.....	20
2.9	<i>Microsoft Visual Studio</i>	21
2.10	<i>Visual Basic 2010</i>	21
2.11	<i>Pengertian UML (Unifed Modeling Language)</i>	23
2.12	Konversi Bilangan	29
2.13	<i>Flowchart</i>	32
2.14	Tabel Karakter ASCII.....	34

BAB III METODE PENELITIAN

3.1	Tahapan Penelitian.....	38
3.2	Metode Pengumpulan Data	38
3.3	Analisis Sistem Berjalan.....	39
3.4	Rancangan Penelitian.....	42
3.5	Struktur Program	44
3.6	Rancangan Tampilan	45

BAB IV HASIL DAN PEMBAHASAN

4.1	Implementasi Sistem	48
4.2	Spesifikasi Sistem	48
4.3	Pengujian Sistem.....	49
4.4	Hasil Rancangan Sistem.....	50
4.5	Kelebihan Dan Kekurangan Sistem	59

BAB V PENUTUP

5.1 Kesimpulan	60
5.2 Saran.....	61

DAFTAR PUSTAKA

BIOGRAFI PENULIS

LAMPIRAN-LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan data dan informasi merupakan bagian yang sangat penting dari sebuah sistem dalam jaringan komputer terutama yang terhubung ke internet. Saat pengiriman dokumen, seseorang bisa saja dengan ilegal mengubah isi dokumen itu tanpa diketahui pengirim atau penerima. Tanpa fasilitas keamanan yang baik, penerima dokumen tersebut dapat mencurigai adanya perubahan yang terjadi pada surat, sehingga legalitas surat dipertanyakan.

Pengelolaan dan pengarsipan surat masih banyak yang dilakukan secara manual karena data yang diproses masih berupa dokumen, maka akan beresiko dokumen hilang dan memakan waktu yang lama untuk mencari dokumen. Salah satu solusi adalah dengan membuat suatu tanda khusus yang memastikan bahwa data tersebut adalah data yang benar. Oleh karena kita bisa menggunakan salah satu teknologi keamanan jaringan yang disebut *Digital Signature*. *Digital Signature* dapat artikan sebagai bagian dari tanda tangan elektronik. Tanda tangan elektronik merupakan sebuah suara elektronik, simbol, atau proses, secara logika dihubungkan dengan kontrak atau lainnya.

Untuk dapat mengetahui surat legalisasi yang baik, tentunya dapat menggunakan metode *Digital Signature*, yang tentunya berhubungan erat dengan ilmu kriptografi. Untuk memberikan metode proteksi maksimum untuk keabsahan

suatu *digital signature*, pada penelitian ini penulis akan meneliti mengenai metode enkripsi untuk membuat suatu *digital signature*.

Adapun metode pembuatan *digital signautre* yang dirancang untuk diterapkan dalam keaslian data surat masuk dan surat keluar dengan menerapkan algoritma *adler32* dimana sebuah berkas berbentuk surat dijadikan pengesahan pada surat masuk dan surat keluar.

Berdasarkan uraian diatas, penulis tertarik untuk melakukan penelitian dan memilih judul **“Penerapan *Digital Signature* dengan Algoritma Adler32 pada Surat Masuk dan Surat Keluar”**

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan di atas, maka penulis merumuskan permasalahan yang akan dihadapi yaitu :

1. Bagaimana menerapkan *digital signature* pada surat masuk dan surat keluar.
2. Bagaimana menerapkan metode algoritma Adler32 untuk menentukan keamanan data surat masuk dan surat keluar.
3. Bagaimana merancang aplikasi konfersi bilangan dari desimal ke hexadesial menggunakan algoritma adler32 melalui pemograman Visual Studio.

1.3 Batasan Masalah

Berikut ini beberapa batasan masalah yang akan dibahas dalam penelitian ini yaitu sebagai berikut:

1. Algoritma yang digunakan adalah Adler32.
2. *Visual Basic (VB)* yang digunakan sebagai bahasa pemrograman pembuatan aplikasi.
3. Desimal dan hexadesimal sebagai konfersi bilangan dari surat yang akan di proses.

1.4 Tujuan Penelitian

Berikut ini beberapa tujuan penelitian yang akan dibahas dalam penelitian ini yaitu sebagai berikut:

1. Untuk menerapkan *digital signature* pada surat masuk dan surat keluar.
2. Untuk menerapkan metode algoritma Adler32 untuk menentukan keamanan data surat masuk dan surat keluar dengan menggunakan mickosoft visual studio.
3. Untuk mengetahui proses konfersi bilangan desimal ke hexadesimal pada sebuah surat dengan menggunakan algoritma adler32.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah

1. Mengembangkan sistem yang sedang berjalan, yang sebelumnya manual menjadi berbasis komputerisasi.

2. Pembuatan *digital signature* dapat digunakan pada surat surat masuk dan surat keluar.
3. Dengan algoritma Adler32 *digital signature* dapat dijadikan surat pengesahan institusi terhadap legalisasi surat masuk.

Adapun metode pembuatan *digital signautre* yang dirancang untuk diterapkan dalam keaslian data surat masuk dan surat keluar dengan menerapkan algoritma Adler32 dimana sebuah berkas berbentuk surat dijadikan pengesahan pada surat masuk dan surat keluar.

Berdasarkan uraian diatas, penulis tertarik untuk melakukan penelitian ini dan memilih judul **“Penerapan *Digital Signature* dengan Algoritma Adler32 pada Surat Masuk dan Surat Keluar”**

BAB II

LANDASAN TEORI

2.1 Digital Signature

Tanda tangan digital merupakan tanda tangan yang dilakukan dengan memakai alat elektronik yang berfungsi sama dengan tanda tangan manual, tanda tangan digital merupakan kumpulan bit yang bisa melakukan fungsi elektronik yang memakai fungsi *hash* satu arah (Ariyus.D 2006).

Pada dasarnya tanda tangan digital dari setiap dokumen berbeda dengan dokumen lainnya, karena diambil dari dokumen itu sendiri dan tentunya dengan adanya perubahan pada dokumen akan menciptakan tanda tangan digital yang berbeda. Tanda tangan digital juga mempunyai fungsi yang sama dengan tanda tangan manual yang mengabsahkan sesuatu dokumen yang bisa dijadikan suatu persetujuan, tanda terima atau lain sebagainya. Fungsi tanda tangan pada dokumen kertas juga diterapkan untuk otentikasi pada data digital seperti pesan yang dikirim melalui saluran komunikasi dan dokumen elektronik yang disimpan di dalam memori komputer. Tanda tangan pada data digital ini dinamakan tanda tangan digital (*digital signature*). Yang dimaksud dengan tanda tangan digital bukanlah tanda tangan yang di-digitalisasi dengan alat *scanner*, tetapi suatu nilai kriptografis yang bergantung pada pesan dan pengirim pesan (hal ini kontras dengan tanda tangan pada dokumen kertas yang bergantung hanya pada pengirim dan selalu sama untuk semua dokumen). Dengan tanda tangan digital, maka

integritas data dapat dijamin, disamping itu ia juga digunakan untuk membuktikan asal pesan (keabsahan pengirim), dan nirpenyangkalan. Sifat dari tanda tangan digital diantaranya:

a. *Authentication*

Authentication adalah jaminan dari suatu pesan yang belum dimodifikasi di dalam pengiriman (keaslian pesan atau integritas pesan). Juga kunci yang membuktikan keaslian untuk kunci publik, pemakai atau identifikasi sumber, yang boleh memverifikasi hak untuk mengirimkan pesan. Penerima pesan dapat memastikan keaslian pengirimannya. *Cryptanalyst* tidak bisa berpura-pura mengirim pesan tersebut. Selain itu juga merupakan suatu proses dalam rangka validasi *user* pada saat masuk ke dalam sistem.

- b. Hanya berlaku untuk sekali pengiriman Dokumen, yang mana tanda tangan digital tersebut tidak bisa dipindahkan ke dokumen lainnya, dan jika dokumen tersebut dirubah maka dokumen yang dikirim tidak berlaku lagi.
- c. Keabsahan tanda tangan digital itu dapat diperiksa oleh pihak yang menerima pesan, walaupun belum pernah saling bertemu sekalipun.

2.2 Fungsi Hash

Fungsi hash sering disebut dengan fungsi hash satu arah (one way function), message digest (pesan pendek), Fingerprint, fungsi matematika yang mengambil masukan panjang variable dan mengubahnya kedalam urutan biner dengan panjang yang tetap. Fungsi hash biasanya diperlukan bila ingin membuat sidik jari dari suatu pesan. Sidik jari pada pesan merupakan suatu tanda bahwa pesan tersebut benar-benar berasal dari orang yang diinginkan.

Cara untuk menguji integritas data adalah dengan menyediakan sebuah "checksum" atau tanda bahwa data tidak diubah. Yang termudah cara yang harus dilakukan adalah menghitung karakter yang ada sehingga jika ada perubahan, hasilnya akan berbeda. Fungsi hash adalah fungsi satu arah yang menghasilkan sebuah "checksum" atau "sidik jari" dari data. Pesan itu lolos ke fungsi hash akan menghasilkan output yang disebut Message Authenticated Code (MAC). Fungsi hash memetakan satu set data ke ukuran yang lebih kecil dan terbatas. Itu algoritma perhitungan menggunakan matriks untuk memetakan array byte. Mari kita ambil contoh sederhana, matematika fungsi modulus. Hasil dari ekspresi modular adalah sisa dari divisi integer. Misalnya, " $12 \bmod 5$ " menghasilkan nilai 2, karena 12 dibagi 5 untuk menghasilkan nilai 2 dan sisanya adalah 2. Setiap hari kita menggunakan operasi modulo untuk mengekspresikan jam di mana modulo adalah 12. Operator mod tidak dapat digunakan sebagai hash yang baik berfungsi tanpa terintegrasi ke formula lain. Sana adalah beberapa persyaratan untuk digunakan secara praktis. Untuk contoh, rentang hasil dari fungsi hash harus cukup sehingga kemungkinan dua berbeda pesan akan menghasilkan output fungsi

hash yang sama. Saya harus ditekankan kata "probabilitas", karena akan ada dua bagian data yang dapat menghasilkan output fungsi hash yang sama. Ini disebabkan oleh kisaran fungsi hash lebih kecil dari ruang input. Untuk buat dua pesan yang bisa dimengerti dan punya yang sama Output fungsi hash tidak mudah. Persyaratan lain fungsi hash yang baik adalah perubahan karakter atau bit tunggal dalam data harus menghasilkan output yang berbeda. Properti ini disebut efek longsor. (Andysah 2016).

2.3 Adler32

Mark Adler menemukan fungsi hash Adler-32. Dia dibuat pada tahun 1995 dan memodifikasi checksum Fletcher. Panjangnya sama dengan CRC. Ini menawarkan kecepatan proses validasi. Dia mengklaim bahwa Adler-32 lebih andal daripada Fletcher-16 dan sedikit kurang andal dibanding Fletcher-32. Itu diperoleh dengan menghitung dua 16-bit checksum A dan B dan menggabungkan bit-bit mereka menjadi sebuah Bilangan bulat 32-bit. Ini berjalan pada platform heksadesimal. A adalah jumlah semua byte dalam aliran plus satu, dan B adalah jumlah nilai individual A dari setiap langkah. Pada mulai menjalankan Adler-32, A diinisialisasi ke 1, B ke 0. Jumlahnya dilakukan modulo 65521. Bytes disimpan dalam urutan jaringan, B menempati dua yang paling signifikan byte.

Fungsi dapat dinyatakan sebagai : $A = 1 + D1 + D2 + \dots + Dn \pmod{65521}$
 $B = 1 + D1) + (1 + D1 + D2) + \dots 1 + D1 + D2 + \dots + Dn) \pmod{65521} = n \times D1$
 $+ (n - 1) \times D2 + (n - 2) \times D3 + \dots + Dn + n \pmod{65521}$

$$\text{Adler-32 (D)} = B \times 65536 + A$$

Tabel ini akan menjadi data lebih lanjut untuk temukan nilai Adler-32. Nilai tersebut diperoleh oleh menerapkan rumus Adler-32 sebelumnya, Tabel ini menggambarkan keseluruhan proses Adler-32.

Tabel 2.1 Nilai Adler32

No	A	B
1	1	0
2	201	201
3	439	640
4	638	127
5	799	2077
6	980	3057
7	1196	4253
8	1388	5641
9	1495	7136
10	1594	8730
11	1736	10466
12	1913	12379
13	2074	14453
14	2236	16689
15	2449	19138
16	2660	21798
17	2828	24626
18	3022	27648
19	3257	30905
20	3454	34359
21	3533	37892
22	3630	41522
23	3796	45318

24	3958	49276
25	4123	53399
26	4362	577661
27	4576	62337
28	4762	67099
29	4851	71950
30	4908	76858
31	5150	81963
32	5303	87266
33	5399	92665
34	5510	98175
35	5688	103863
36	5922	109785
37	6161	115946

NO	A	B
38	6415	122361
39	6614	128975
40	6654	135629
41	6682	142311
42	6674	1490
43	6834	155889
44	6960	162849
45	7104	169953
46	7248	177201
47	7471	184672
48	7726	192398
49	7980	200378
50	8235	208613
51	8475	217088
52	8630	2255718

53	8694	234412
54	8764	243176
55	8878	252054
56	9009	261063
57	9100	270163
58	9228	279391
59	9463	28885
60	9705	298559
61	9958	308517
62	10212	318729
63	104666	329195
64	10703	339898
65	10825	350723
66	10885	361608
67	10975	372583
68	11044	383627
69	11092	394719
70	11146	405865
71	11233	417098
72	11310	428408
73	11479	439887
74	11726	451613

NO	A	B
75	11980	463593
76	12194	475787
77	12297	488084
78	12350	500434
79	12378	512812
80	12414	525226
81	12457	537683
82	12492	550175
83	12548	562723
84	12585	575308
85	12661	57869
86	128896	600865
87	131444	614009
88	13387	627396
89	13502	640898
90	13537	654435
91	13596	668031
92	13622	681653
93	13650	695303
94	13681	708984
95	13710	722694
96	13814	736508
97	13928	7504366

Sumber : Andysah P.U Siahaan (2016)

Ada 96 (12 x 8 piksel) + 1 perhitungan. 1, A =1 dan B = 0 adalah kondisi inital. A terakhir menunjukkan 13928 dan B terakhir menunjukkan 750436. Perhitungannya tidak berakhir.

A	= A % MOD_ADLER = A % 65521
=	13928 % 65521
=	13928

B	= B % MOD_ADLER = B % 65521
---	-----------------------------------

$$= 750436 \% 65521$$

$$= 29705$$

$$AD = B \cdot 65536 + A$$

$$= 29705 \cdot 65536 + 13928$$

$$= 1946760808 \text{ (decimal)}$$

$$= 74093668 \text{ (hexadecimal)}$$

Nilai Adler-32 yang ditunjukkan di atas masih dalam desimal format. Adler-32 berjalan dalam heksadesimal. Nilai dalam heksadesimal adalah 74093668. Ini adalah kombinasi dari dua 16-bit nilai. Bagian pertama adalah 7409 dan yang terakhir adalah 3668. Saat mengirim gambar ini ke penerima, pengirim harus mengirim nilai Adler-32 ini (74093668) serentak. Setelah itu, penerima akan melakukan sinkronisasi hash mereka dengan pengirim. Setelah nilainya berbeda, harus ada kesalahan atau intersepsi saat transmisi di udara. Bagaimana jika kontennya sudah dimodifikasi atau ada objek kecil yang tidak terdeteksi telah dimasukkan ke dalam gambar. Sudah waktunya untuk membuktikan fungsi hash. Menganggap bahwa kita memodifikasi angka piksel 1. Nilai sebelumnya adalah R= 192, G = 203 dan B = 202. Nilai baru adalah R =

190, G = 203, dan B = 201. Ini adalah perubahan kecil. Saya t tidak dapat dideteksi dengan mata telanjang. Kami baru saja memodifikasi warna merah dan biru; hijau tetap sama. Itu orisinalitas hanya dapat dideteksi dengan menggunakan komputer program. Itu sebabnya pengirim selalu mengirim nilai integritas dengan gambar itu untuk melindungi informasi di dalam.

2.4 Preprocessing Image

Hal terpenting dalam fungsi hash visual adalah proses skala abu-abu itu sebagian besar dilakukan dalam pengolahan gambar adalah mengubah gambar warna menjadi gambar skala abu-abu. Digunakan untuk menyederhanakan gambar model. Gambar arna terdiri dari tiga lapisan, Merah, Hijau dan Biru. Proses skala abu-abu adalah untuk mencampur lapisan dan menghasilkan warna tunggal. Ketika ada perhitungan yang dilakukan menggunakan tiga lapisan itu akan diubah dengan mengelompokkan lapisan ketiga menjadi skala abu-abu. Tidak ada warna, hanya kelulusan hitam dan putih. Ada tiga cara untuk mendapatkan intensitas skala abu-abu.

$$\text{Skala abu-abu} = \frac{\max(R,G,B) + (R,G,B)}{2} \quad (1)$$

$$\text{Skala abu-abu} = \frac{R+G+B}{3} \quad (2)$$

$$\text{Skala abu-abu} = (0.21 * R) + (0.72 * G) + (0.07 * B) \quad (3)$$

Dari rumus diatas dapat dijelaskan bagaimana cara mnedapatkan intensitas skala abu-abu. Rumus 1 menyangkut penerangan rumus 2 menyangkut rata-rata

dan rumus 3 memyangkut lumitas. Rumus 1 adalah untuk menemukan nilai tertinggi dan terendah dari nilai Merah Hiau dan Biru kemudian nilai tertinggi dan terendah dijumlahkan dan kemudian dikalikan dengan 0.5. Formula 2 adalah menjumlahkan semua nilai merah hijau dan biru lalu di bagi 3 untuk mendapatkan nilai rata-rata merah hijau dan biru. Formula 3 adlah untuk mengalikan setiap nilai merah, hijau dan biru dengan nilai yang telah ditentukan sebelumnya, kemudian hasil dari pengkalian seluruh nilai tambah merah, hijau dan biru satu sama lain.

2.5 Proposed Work

Dalam penelitian ini, cara untuk menghitung intensitas warna setiap lapisan gambar memiliki tiga lapisan intensitas warna yaitu, Merah, Hijau dan Biru dan harus di kombinasikan dengan warna rata-rata. Wara rata-rata disimpan dalam bagian skala abu-abu. Dengan menggabungkan 3 lapisan menjadi 1 lapisan yang dikonversi dan di hitung pikselnya. Langkah pertama adalah memecahkan warna dan membangun intensitas baru.

Tabel 2.2 Nilai Intensistas warna Merah

RED				
R^{11}	R^{12}	R^{13}	R^{14}	R^{15}
R^{21}	R^{22}	R^{23}	R^{24}	R^{25}
R^{31}	R^{32}	R^{33}	R^{34}	R^{35}
R^{41}	R^{42}	R^{43}	R^{44}	R^{45}
R^{51}	R^{52}	R^{53}	R^{54}	R^{55}

Tabel 2.3 Nilai Intensitas warna Hijau

GREEN				
G^{11}	G^{12}	G^{13}	G^{14}	G^{15}
G^{21}	G^{22}	G^{23}	G^{24}	G^{25}
G^{31}	G^{32}	G^{33}	G^{34}	G^{35}
G^{41}	G^{42}	G^{43}	G^{44}	G^{45}
G^{51}	G^{52}	G^{53}	G^{54}	G^{55}

Tabel 2.4 Nilai IntesitasWarna Biru

BLUE				
B^{11}	B^{12}	B^{13}	B^{14}	B^{15}
B^{21}	B^{22}	B^{23}	B^{24}	B^{25}
B^{31}	B^{32}	B^{33}	B^{34}	B^{35}
B^{41}	B^{42}	B^{43}	B^{44}	B^{45}
B^{51}	B^{52}	B^{53}	B^{54}	B^{55}

2.6 Surat

Menurut pengertian umum surat adalah untuk menyampaikan maksud secara tertulis. Dalam pengertian sehari-hari, surat umumnya hanya dikenal sebagai alat untuk menyampaikan berita secara tertulis. Pengertian tersebut merupakan pengertian dalam arti sempit, akibat dari anggapan bahwa surat hanya alat untuk menyampaikan berita, padahal surat mengandung aspek yang jauh lebih luas mencakup informasi tertulis. Adapun pengertian dengan informasi tertulis disini adalah informasi berupa kabar atau berita, misalnya penawaran, pesanan, panggilan dan permohonan. Surat juga bisa sebagai informasi rekaman berita secara tertulis, misalnya surat tanda bukti, kartu identitas, akta dan kontrak. Maka dari uraian di atas dapat disimpulkan bahwa pada dasarnya surat adalah informasi tertulis yang dapat dipergunakan sebagai alat komunikasi tulisan yang dibuat dengan persyaratan tertentu yang khusus yang berlaku untuk surat menyurat. Penyampaian maksud melalui surat dari suatu pihak ke pihak lain dapat di atas namakan perseorangan (pribadi) dan dapat juga di atasnamakan jabatan dan juga suatu organisasi. Kegiatan saling berkirim oleh perseorangan (pribadi) atau oleh organisasi disebut surat menyurat korespondensi dan para pelakunya disebut koresponden.

a. Pengertian Surat Masuk

Surat masuk yaitu semua surat yang dialamatkan, ditunjukkan dan diterima oleh organisasi, perusahaan atau instansi, baik yang berasal dari perorangan maupun dari suatu organisasi perusahaan instansi dan lainnya.

b. Pengertian Surat Keluar

Surat yang dibuat oleh instansi atau perusahaan yang ditujukan kepada instansi atau perusahaan lain atau untuk kepentingan internal. (Meliana,2008)

2.7 Bahasa Pemrograman

Bahasa pemrograman adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas tertentu”. Bahasa pemrograman merupakan sebuah instruksi untuk memerintah komputer agar bisa menjalankan fungsi tertentu, namun hanya instruksi standar saja. Bahasa pemrograman juga memiliki perhimpunan dari aturan sintaks dan semantik yang tugasnya untuk mendefinisikan program komputer. Bahasa pemrograman komputer yang kita kenal antara lain adalah *Java*, *Visual Basic*, *C++*, *C*, *PHP*, dan bahasa pemrograman lainnya. Namun kebutuhan bahasa pemrograman ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya. Menurut generasi bahasa pemrograman digolongkan menjadi 4 generasi, yaitu:

1. Generasi pertama : *machine language*
2. Generasi kedua : *assembly language: Assembler*
3. Generasi ketiga: *high level programming language*, contoh: *C* dan *Pascal*
4. Generasi keempat : *4 GL (fourth-generation language)*, contoh: *SQL*

5. Generasi kelima : *Programming Language Based Object Oriented & Web Development.*

Secara umum bahasa pemrograman dibagi menjadi 4 kelompok, yaitu :

1. *Object Oriented Language* : Seperti bahasa *Visual C, Delphi, Visual dBase, Visual FoxPro.*
2. *Low Level Language* : Bahasa *Assembly.*
3. *Middle Level Language* : Bahasa *C.*
4. *High Level Language* : Bahasa *Basic dan Pascal.*

Menurut dari tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari sebagai berikut:

- a. Bahasa Mesin, yaitu untuk memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya adalah 01100101100110.
- b. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah. Inggris *Assembly*), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode *mnemonic*), contohnya *MOV, SUB, CMP, JMP, JGE, JL, LOOP,* dsb.
- c. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dengan kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&.

- d. Bahasa Tingkat Tinggi, yaitu bahasa yang merupakan komputer yang memakai instruksi yang berasal dari unsur kata-kata bahasa manusia, contohnya adalah : *begin, end, if, for, while, and, or, dsb.* Fungsi dari bahasa pemrograman adalah untuk memerintahkan sebuah komputer agar dapat mengolah data yang sesuai dengan di inginkan. *Output* dari bahasa pemrograman ini dapat berupa aplikasi ataupun program khusus. (Jusuf Wahyudi 2013)

2.8 Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan pembuatan aplikasi tersebut”, Secara umum aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang dilakukan oleh manusia. Menurut kamus besar bahasa Indonesia “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”.

Aplikasi komputer juga disebut perangkat lunak yang mana program komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Pada umumnya aplikasi digunakan untuk mengontrol perangkat keras atau yang sering disebut sebagai *service device driver*. (Ninuk Wiliani dan Syadid Zambani : 2017)

2.9 *Microsoft Visual Studio*

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi”. Baik itu aplikasi bisnis, aplikasi personal, ataupun komponen komponen aplikasi dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. *Visual Studio* mencakup compiler dan kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain adalah *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic.NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe* *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* bahasa mesin yang berjalan di *Windows*.

Visual studio dapat digunakan untuk membuat sebuah aplikasi yang berbasis desktop yang merupakan *platform windows*, namun juga dapat dijalankan dalam bentuk *Microsoft Intermediate Language*, *Net Framework*. Selain itu *Visual Studio* juga dapat digunakan untuk membuat sebuah aplikasi yang dapat dijalankan di *windows mobile* yang juga berjalan diatas *Net Compact Framework*. (Ninuk Wiliani : 2017)

2.10 *Visual Basic 2010*

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer”. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual Studio 2010* yang sering disebut dengan *VB.Net 2010*, yaitu

sebagai bahasa pemrograman sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows. *Visual basic* merupakan sebuah bahasa pemrograman yang berpusat pada object (*Object Oriented Programming*) digunakan dalam pembuatan aplikasi *Windows* yang berbasis *Graphical User Interface*, hal ini menjadikan *Visual Basic* menjadi bahasa pemrograman yang wajib diketahui dan dikuasai oleh setiap programmer. Beberapa karakteristik obyek tidak dapat dilakukan oleh *Visual Basic* misalnya seperti *Inheritance* tidak bisa *module* dan *Polymorphism* secara terbatas bisa dilakukan dengan deklarasi class module yang mempunyai *Interface* tertentu.

Adapun beberapa kemampuan atau beberapa manfaat dari *Visual Studio* 2010 diantaranya :

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : aplikasi Internet .
3. Untuk menguji program (*debugging*) yang menghasilkan program berakhiran *EXE* yang bersifat executable atau juga dapat langsung dijalankan.

Visual Studio 2010 “merupakan bahasa yang dapat cukup mudah untuk dipelajari. Bagi programmer pemula yang baru ingin belajar program, lingkungan *Visual Studio* dapat membantu membuat program dalam sekejap mata. Sedang bagi programmer tingkat lanjut, kemampuan yang besar dapat digunakan untuk membuat program-program yang kompleks, misalnya lingkungan net-working atau client server”. *Bahasa Visual Studio* cukup sederhana dan menggunakan kata-kata

bahasa *Inggris* yang umum digunakan. dan di dalam *Visual Basic* semuanya sudah disediakan dengan pilihan-pilihan yang tinggal diambil sesuai dengan kebutuhan. Selain itu, sarana pengembangannya yang bersifat visual memudahkan kita untuk mengembangkan aplikasi berbasis Windows.

Adapun kelebihan dan kekurangan dari *Visual Basic* ini yaitu :

1. Kelebihan *Visual Basic*
 - a. *VB.Net* menyediakan fasilitas *Real Time Background Compiler* yaitu sebagai penanganan dalam error atau bug.
 - b. Lebih cepat dalam pembuatan aplikasi berbasis desktop
 - c. Menyediakan untuk *developer* pemrograman data akses *ActiveX Data Object (ADO)*.
2. Kelemahan *Visual Basic*
 - a. Untuk versi *VB.Net 2010* dengan seterusnya tidak mempunyai Komponen *Crystal Report* karena sudah terpisah.
 - b. Harus ada *Net framework* agar aplikasi bisa berjalan
 - c. Tidak mempunyai database sendiri.
 - d. Memerlukan kapasitas yang besar untuk instalasi *VB.Net*.

(Ninuk Wiliani : 2017)

2.11 Pengertian UML (*Unified Modeling Language*)

Unified Modeling Language (UML) “Merupakan standarisasi bahasa pemodelan untuk membangun perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. Diagram diagram yang

digunakan pada *UML* antara lain adalah *class diagram*, *object diagram*, *use case diagram*, *activity diagram*, dan *sequence diagram*”.

Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek yaitu *UML (Unified Modeling Language)*. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung. (Rosa A.S et al 2013).

Berikut beberapa tujuan atau fungsi dari penggunaan *UML*, yang diantaranya:

1. Dapat memberikan bahasa permodelan untuk *visual* kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan *visual* yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
4. Dapat berguna sebagai *blue print*, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.

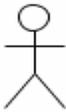
6. Dapat berguna untuk menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

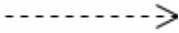
Sistem yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use Case Diagram*

Menurut Yunahar Heriyanto (2018) *Use Case Diagram* “adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan. *Diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* merupakan sesuatu yang mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem”. Berikut adalah simbol-simbol yang ada pada *diagram use case* :

Tabel 2.5 Simbol-Simbol *Use Case Diagram*

NO	Simbol	Deskripsi
1		Orang/Actor, proses himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

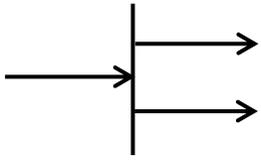
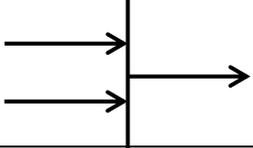
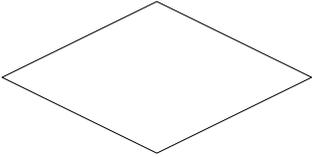
2		Dependency, hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		Generalzation, hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		Include, menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		Extend, menspesifikasikan bahwa <i>use case</i> target, memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		Association, merupakan Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		Collaboration, interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
8		<i>Use Case</i> , deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Note</i> , Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
10		System, Menspesifikasikan paket yang menampilkan sistem secara terbatas.

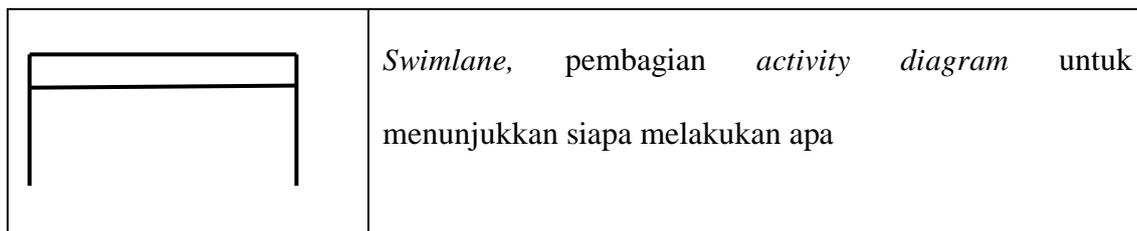
Sumber: Winda Aprianti (2016)

2. Diagram Aktifitas (*Activity Diagram*)

Activity diagram adalah “menggambarkan *workflow* (aliran kerja) atau dengan aktivitas dari sebuah sistem atau proses bisnis”. Berikut simbol-simbol yang digunakan dalam *activity diagram* yaitu :

Tabel 2.6 Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> /percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .



Sumber: Winda Aprianti (2016)

3. Diagram Kelas (*Class Diagram*)

“Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram*, juga menunjukkan sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*”.

2.12 Konfersi Bilangan

Konfersi bilangan adalah suatu proses dimana suatu sistem bilangan dengan basis tertentu akan di jadikan bilangan basis yang lain. Pada perancangan aplikasi konfersi bilangan, penyajian bilangan disusun berdasarkan pokok pembahasan sesuai kebutuhan.

Materi pembelajaran konversi bilangan termasuk dalam teori algoritma genetika, dalam bentuk biner yang banyak digunakan dalam bahasa pemrograman dan mutasi algoritma.

Pembelajaran yang dilakukan saat ini masih banyak yang menggunakan pembelajaran bersifat konvensional, dimana pendidik hanya mencatat materi ke papan tulis dan peserta didik menulis catatan tersebut ke buku catatan, dengan durasi pencatatan setiap peserta didik berbeda-beda sehingga apabila peserta didik yang memiliki durasi paling lama dalam mencatat tentunya akan mengurangi durasi proses belajar mengajar, sehingga proses penyampaian materi oleh pendidik ke peserta didik lebih sedikit, cara seperti ini tentunya tidak efisien. Konversi bilangan adalah suatu proses dimana satu system bilangan dengan basis tertentu akan dijadikan bilangan dengan basis yang lain. $010101011111_2 = 2537_8$
Konversi dari bilangan Desimal. Adapun materi yang di bahas ialah :

a. Bilangan desimal

Bilangan desimal adalah sistem bilangan yang paling umum digunakan dalam kehidupan sehari-hari. Sistem bilangan desimal menggunakan basis 10 dan menggunakan 10 macam simbol bilangan yaitu : 0, 1, 2, 3, 4,5,6,7,8 dan 9. Sistem bilangan desimal dapat berupa integer desimal (decimal integer) dan dapat juga berupa pecahan decimal (decimal fraction). Konversi dari bilangan Desimal ke biner Yaitu dengan cara membagi bilangan desimal dengan dua kemudian diambil sisa pembagiannya.

Contoh :

$$45 (10) = \dots(2)$$

$$45 : 2 = 22 + \text{sisa } 1$$

$$22 : 2 = 11 + \text{sisa } 0$$

$$11 : 2 = 5 + \text{sisa } 1$$

$$5 : 2 = 2 + \text{sisa } 1$$

$$2 : 2 = 1 + \text{sisa } 0 \quad 101101(2) \text{ ditulis dari bawah ke atas.}$$

b. Bilangan biner

Bilangan biner adalah bilangan *radix* (dasar) dua, angka terendahnya 0 dan angka tertingginya 1. Setiap kali perhitungan dengan menambahkan satu satuan, maka akan berturut-turut akan diperoleh: 0, 1, 10, 11, 100, 101, 110, 111, 1000 dan seterusnya atau dengan cara membagi bilangan desimal dengan 2 kemudian diambil sisa pembagiannya

Contoh :

$$385 (10) = \dots(2)$$

$$385 : 2 = 192 + \text{sisa } 1$$

$$192 : 2 = 96 + \text{sisa } 0$$

$$601 (8)$$

c. Bilangan octal

Sistem bilangan oktal adalah salah satu sistem bilangan yang dipakai pada beberapa mini komputer seperti PDP-8. Bilangan oktal

merupakan bilangan dengan dasar 8 dan memiliki 8 simbol bilangan yakni 0, 1, 2, 3, 4, 5, 6 dan 7. Jadi, dalam perhitungan berturut-turut setelah 7 adalah 10.

d. Bilangan heksadesimal

Bilangan heksadesimal adalah bilangan dasar 16, simbol bilangannya 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E dan F. Angka terendah adalah 0 dan angka tertinggi adalah F. Bilangan ini dipergunakan sebagai bahasa mesin baik pada mikroprosesor maupun dalam komputer.

Berikut beberapa tujuan atau fungsi dari penggunaan *UML*, yang diantaranya:

1. Dapat memberikan bahasa permodelan untuk *visual* kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan *visual* yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
4. Dapat berguna sebagai *blue print*, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.

5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
 6. Dapat berguna untuk menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.
- Famalua Gulo (2016:34)

2.13 Flowchart

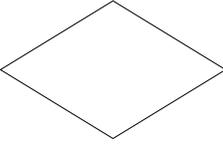
Flow chart atau *diagram* alir “adalah sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, dan beserta urutannya dengan menghubungkan masing-masing langkah tersebut dengan menggunakan tanda panah. Diagram ini juga bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut”.

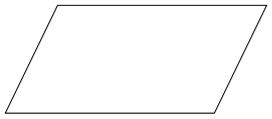
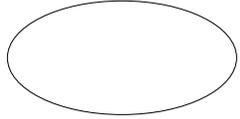
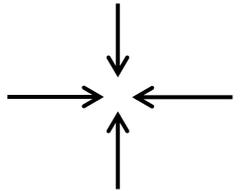
Flow chart digunakan dalam merancang dan mendokumentasikan proses yang kompleks atau program. Seperti jenis lain *diagram*, mereka membantu memvisualisasikan apa yang terjadi dan dengan demikian membantu pengunjung untuk memahami proses, dan mungkin juga menemukan kelemahan, kemacetan, dan ketidakjelasan lain di dalamnya. Ada berbagai jenis diagram alur yang masing-masing memiliki repertoire kotak sendiri dan ketentuan notasinya. Dua jenis yang paling umum dari kotak di *flow chart* adalah :

1. Langkah pengolahan, biasanya disebut aktivitas, dan dilambangkan sebagai persegi panjang.
2. Keputusan, biasanya dilambangkan sebagai belah ketupat.

Flowchart memberikan solusi langkah demi langkah dari sebuah masalah yang ingin dipecahkan. Kegunaan *flowchart* adalah untuk menganalisis, merancang, mendokumentasikan serta mengelola suatu proses atau program di berbagai bidang. Sebuah proses atau *action* direpresentasikan dalam sebuah kotak, dan tanda panah yang menghubungkan kotak-kotak ini mewakili aliran atau arah aliran data. Berikut ini adalah simbol-simbol *flowchart* yang sering digunakan beserta deskripsinya (Mohamad Natsir 2017):

Tabel 2.7 Simbol-Simbol *Flowchart*

No	Simbol <i>Flowchart</i>	Fungsinya
1		<i>Terminal</i> atau <i>Start</i> , berfungsi untuk memulai dan mengakhiri alur program.
2		<i>Process</i> , adalah untuk mengolah dan mengubah data yang ada didalam komputer.
3		<i>Decision</i> , digunakan untuk menentukan operasi perbandingan logika ketika masuk pada alur program.

4		<i>Input</i> dan <i>Output</i> , adalah simbol yang digunakan untuk memasukan data yang biasanya berupa <i>username</i> dan <i>password</i> , dimana hasil dari proses.
5		<i>Connector</i> , adalah menentukan hubungan arus proses program yang berjalan dalam halaman yang sama.
6		<i>Arrow Flow</i> , adalah untuk menunjukkan alur proses program yang terdiri dari, alur atas ke bawah, kanan ke kiri dan juga sebaliknya.
7		<i>Document</i> , adalah sebuah simbol untuk data atau informasi.

Sumber : Mohamad Natsir (2017)

2.14 Tabel Karakter ASCII

Kode standar Amerika untuk pertukaran informasi atau *American Standard Code Information Interchange* (ASCII) merupakan suatu standar internasional dalam kode huruf dan simbol seperti hexadesimal dan unicode tetapi ASCII lebih bersifat universal contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun, ASCII disimpan sebagai sandi 8 bit dengan menambahkan satu angka 0 sebagai bit significant paling tinggi. Bit tambahan ini sering digunakan untuk uji paritas. Karakter control pada ASCII dibedakan menjadi 5 kelompok sesuai dengan

penggunaan yaitu berturut-turut meliputi logical communication, Device control, Information separator, Code extention, dan physical communication. Code ASCII ini banyak dijumpai pada papan ketik (keyboard) computer atau instrument-instrument digital.

Jumlah kode ASCII adalah 255 kode. Kode ASCII 0..127 merupakan kode ASCII untuk manipulasi teks; sedangkan kode ASCII 128..255 merupakan kode ASCII untuk manipulasi grafik. Kode ASCII sendiri dapat dikelompokkan lagi kedalam beberapa bagian:

1. Kode yang tidak terlihat simbolnya seperti Kode 10(Line Feed), 13(Carriage Return), 8(Tab), 32(Space).
2. Kode yang terlihat simbolnya seperti abjad (A..Z), numerik (0..9), karakter khusus (~!@#\$\$%^&*()_+?:'"})
3. Kode yang tidak ada di keyboard namun dapat ditampilkan. Kode ini umumnya untuk kode-kode grafik.

Dalam pengkodean kode ASCII memanfaatkan 8 bit. Pada saat ini kode ASCII telah tergantikan oleh kode UNICODE (Universal Code). UNICODE dalam pengkodeannya memanfaatkan 16 bit sehingga memungkinkan untuk menyimpan kode-kode lainnya seperti kode bahasa Jepang, Cina, Thailand dan sebagainya.

Pada papan keyboard, aktifkan numlock, tekan tombol ALT secara bersamaan dengan kode karakter maka akan dihasilkan karakter tertentu. Misalnya: ALT + 44 maka akan muncul karakter koma (.). Mengetahui kode-kode

ASCII sangat bermanfaat misalnya untuk membuat karakter-karakter tertentu yang tidak ada di keyboard.

Tabel berikut berisi karakter-karakter ASCII . Dalam sistem operasi Windows dan MS-DOS, pengguna dapat menggunakan karakter ASCII dengan menekan tombol Alt+[nomor nilai ANSI (desimal)]. Sebagai contoh, tekan kombinasi tombol Alt+87 untuk karakter huruf latin "W" kapital.

Tabel 2.8 Nilai ASCII

Karakter	Hex	Desimal	Karakter	Hex	Desimal	Karakter	Hex	Desimal	Karakter	Hex	Desimal
NUL (null)	0	0	Space	20	32	@	40	64	`	60	96
Start Heading	1	1	!	21	33	A	41	65	a	61	97
Start Text	2	2	"	22	34	B	42	66	b	62	98
End Text	3	3	#	23	35	C	43	67	c	63	99
End Transmit.	4	4	\$	24	36	D	44	68	d	64	100
Enquiry	5	5	%	25	37	E	45	69	e	65	101
Acknowledge	6	6	&	26	38	F	46	70	f	66	102
Bell	7	7	`	27	39	G	47	71	g	67	103
Backspace	8	8	(28	40	H	48	72	h	68	104
Horiz. Tab	9	9)	29	41	I	49	73	i	69	105
Line Feed	A	10	*	2A	42	J	4A	74	j	6A	106
Vert. Tab	B	11	+	2B	43	K	4B	75	k	6B	107
Form Feed	C	12	,	2C	44	L	4C	76	l	6C	108
Carriage Return	D	13	-	2D	45	M	4D	77	m	6D	109
Shift Out	E	14	.	2E	46	N	4E	78	n	6E	110
Shift In	F	15	/	2F	47	O	4F	79	o	6F	111
Data Link Esc	10	16	0	30	48	P	50	80	p	70	112
Direct Control 1	11	17	1	31	49	Q	51	81	q	71	113
Direct Control 2	12	18	2	32	50	R	52	82	r	72	114
Direct Control 3	13	19	3	33	51	S	53	83	s	73	115
Direct Control 4	14	20	4	34	52	T	54	84	t	74	116
Negative ACK	15	21	5	35	53	U	55	85	u	75	117
Synch Idle	16	22	6	36	54	V	56	86	v	76	118
End Trans Block	17	23	7	37	55	W	57	87	w	77	119
Cancel	18	24	8	38	56	X	58	88	x	78	120
End of Medium	19	25	9	39	57	Y	59	89	y	79	121
Substitutue	1A	26	:	3A	58	Z	5A	90	z	7A	122
Escape	1B	27	;	3B	59	[5B	91	{	7B	123
Form separator	1C	28	<	3C	60	\	5C	92		7C	124
Group separator	1D	29	=	3D	61]	5D	93	}	7D	125
Record Separator	1E	30	>	3E	62	^	5E	94	~	7E	126
Unit Separator	1F	31	?	3F	63	_	5F	95	Delete	7F	127

BAB III

TAHAPAN PENELITIAN

3.1 Tahapan Penelitian

Dalam penyelesaian Skripsi ini, teknik pengumpulan data merupakan faktor penting demi keberhasilan penelitian. Hal ini berkaitan dengan bagaimana cara mengumpulkan data, Siapa sumbernya, dan apa alat yang di gunakan.

3.2 Metode Pengumpulan Data

1. Analisa

Menganalisa sistem merupakan langkah dalam membuat sistem baru. Dalam analisa sistem digunakan metode-metode yang telah dijelaskan pada rancangan penelitian.

2. Perancangan

Perancangan sistem merupakan tahap pengembangan setelah analisis sistem dilakukan. Beberapa proses dalam perancangan aplikasi tersebut, seperti perancangan input dan output flowchart keamanan data menggunakan algoritma *Adler32*.

3. Studi perpustakaan

Pada tahap ini dilakukan studi literatur yang bertujuan mengumpulkan mempelajari serta menyeleksi bahan-bahan atau sumber dan buku-buku

baik text book, ebook atau jurnal yang di perlukan untuk penulisan skripsi ini.

4. Ujicoba program

Setelah program selesai, maka akan dilakukan uji coba program untuk mengetahui apakah program tersebut telah bekerja dengan benar dan sesuai dengan sistem yang dibuat.

5. Pembuatan kesimpulan

Pada tahap akhir ini adalah pembuatan kesimpulan atau ringkasan dari makalah skripsi ini dan kesimpulan tentang program yang telah dibuat.

3.3 Analisa Sistem Yang Berjalan

Dalam materi perkuliahan keamanan komputer terdapat pada bab mengenai konfersi bilangan. Salah satu bentuk konfersi bilangan adalah menggunakan metode desimal ke hexadesimal Untuk mendapatkan hasil dari *Adler32*, menggunakan angka dan tabel-tabel untuk konfers. Algoritma *Adler32* nantinya akan menganalisa langkah-langkah kerja algoritma *Adler32* tersebut, sehingga nantinya algoritma *Adler32* yang penulis bangun akan memiliki tingkat kesulitan yang lebih tinggi untuk di pecahkan.

Visual basic 2010 akan menjadi sarana untuk menciptakan perangkat lunak ini. Pada analisa proses ini penggunaan digunakan sebagai metode yang di dalam nya terdapat kombinasi dari algoritma *Adler32*. Algoritma *Adler32* digunakan

oleh Admin untuk menghitung pesan yang akan dikirimkan. Perhitungan secara matematis dilakukan sebagai penggambaran proses yang akan terjadi pada metode ini yang didalamnya terdapat algoritma *Adler32*.

Menjelaskan sistem yang sedang berjalan baik manual maupun komputerisasi pada objek penelitian meliputi yaitu:

1. Proses Perhitungan panjang surat

Tahap ini dilakukan dengan menggunakan Algoritma *Adler32* yang akan digunakan untuk meng-hitung pesan asli (*Plaintext*) pengirim.

Diketahui *Plaintext* "MEDAN" dengan memakai rumus ASCII. Maka untuk mendapatkan Bilangan Desimal dan hexadesimal harus menggunakan penghitungan seperti di bawah ini:

Langkah pertama membuat table konversi *ASCII*.

Penerima memilih kata MEDAN sebagai Plaintext yang akan di gunakan untuk melakukan proses hitung menggunakan *Algoritma Adler32*, sehingga pada proses hitung kata MEDAN akan mengikuti Plaintext yang didapat.

Plaintext = MEDAN

Selanjutnya kata MEDAN akan di konfersi kan dengan tabel ASCII

Parameter Adler32

A = 1

B = 0

A (A + Plaintext =)

B (B + A =)

MEDAN

M = 77

E = 69

D = 68

A = 65

N = 78

A (1 + Plaintext =)

A = 1 + 77 = 78

A = 78 + 69 = 147

A = 147 + 68 = 215

A = 215 + 65 = 280

A = 280 + 78 = 358

B (0 + A =)

B = 0 + 78 = 78

B = 78 + 147 = 225

B = 225 + 215 = 440

$$B = 440 + 280 = 720$$

$$B = 720 + 358 = 1078$$

$$\text{Adler32} = (B * 65536) + A$$

$$\text{Adler32} = (1078 * 65536) + 358$$

$$\text{Adler32} = 70648166$$

$$\text{Nilai Adler32Dec} = 70648166$$

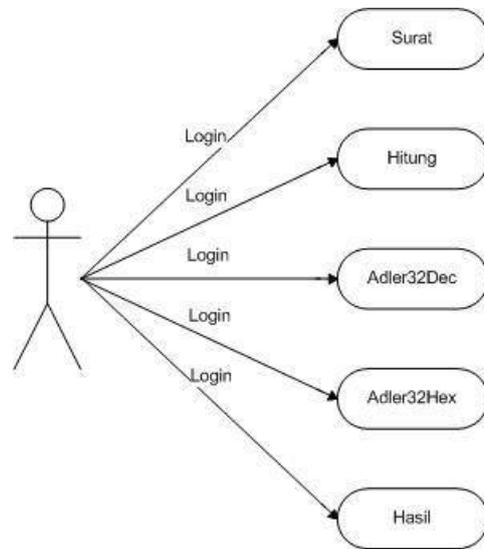
$$\text{Nilai Adler32Hex} = 4360166$$

Dengan panjang surat dari Plaintext MEDAN = 5

3.4 Rancangan Penelitian

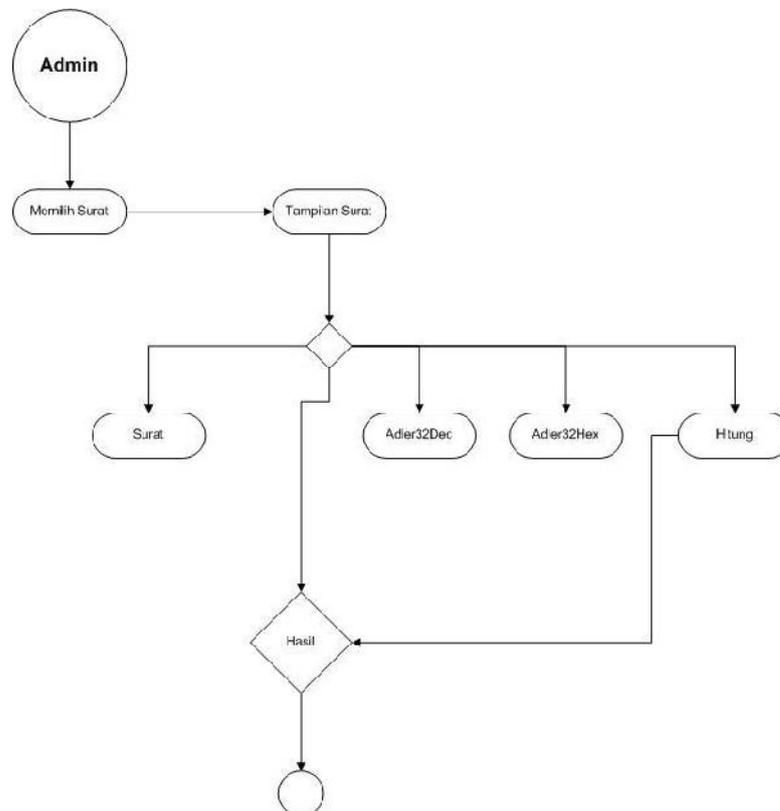
Perancangan merupakan proses mendapatkan informasi dari model dan menampilkannya secara grafik dengan menggunakan sebuah standar elemen grafik. Tujuan dari perancangan ini memungkinkan adanya komunikasi yang lebih berkualitas antara pengguna, pengembang, penganalisis, tetster, manajer dan siapapun yang terlibat dalam proyek pengembangan system informasi.

- a. Use Case Diagram : *use case* diagram terdiri atas, *user/* pengguna sebagai *actor* yang mempunyai *use case* dengan isi Surat, Hitung, *Adler32Dec* *Adler32Hex* dan Hasil.



Gambar 3.1 Use case Diagram

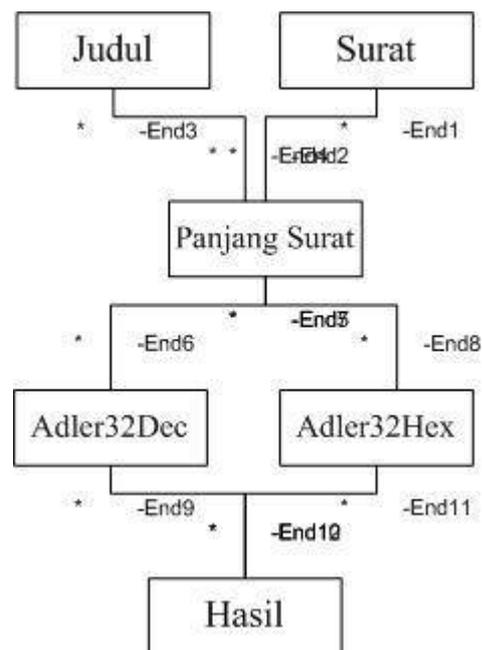
- b. Activity Diagram : menggambarkan aktifitas-aktifitas yang terjadi dalam aplikasi dari aktivitas dimulai sampai aktivitas berhenti.



Gambar3.2. Activity Diagram

3.5 Struktur Program

Struktur program mempresentasikan organisasi komponen program (modul) serta mengimplementasikan suatu hirarki kontrol. Hirarki kontrol tidak mengimplementasikan aspek prosedural dari perangkat lunak seperti urutan proses, kejadian atau urutan dari keputusan atau perulangan operasi.



Gambar3.3 Struktur Proses Konfersi Bilangan

3.6 Rancangan Tampilan

1. Rancangan Halaman Judul

Halaman judul merupakan halaman yang pertama muncul pada saat program dijalankan. Pada rancangan di atas akan menampilkan judul yang kemudian akan pindah ke *form* menu utama dengan menggunakan timer.



Gambar 3.4 Rancangan Halaman Judul

2. Rancangan Halaman Menu Utama

Form ini berisi tombol-tombol seperti menu Materi, *Enkripsi*, *Deskripsi*, tentang, dan Keluar.

Surat	<input type="text"/>
Adler32Dec	<input type="text"/>
Adler32Hex	<input type="text"/>
	<input type="button" value="Proses"/>
Hasil	<input type="text"/>

Gambar 3.5 Rancangan Halaman Menu Utama

Pada tampilan di atas terdapat 5 tombol yaitu Surat, *Adler32Dec*, *Adler32Hex*, Proses dan Hasil.

- a. Tombol Surat berfungsi untuk menampilkan Isi Surat pengguna ke menu.
- b. Tombol *Adler32 Dec* berfungsi untuk menampilkan hasil dari isi surat.
- c. Tombol *Adler32Hex* berfungsi untuk menampilkan hasil dari isi surat.
- d. Tombol proses berfungsi untuk menampilkan perhitungan konfersi bilangan dari isi surat.
- e. Tombol hasil berfungsi untuk menampilkan hasil perhitungan dari program.

3. Rancangan Halaman *Adler32De* dan *Adler32Hex*

Berisi penjelasan mengenai konfersi Bilangan. Pememasukkan Surat asli atau *plaintext* kedalam tombol masukan *plaintext* kemudian dihitung jumlah panjang Surat. Setelah itu, ditekan tombol Hitung yang kemudian akan menampilkan hasil bilangan *Adler32Dec* dan hasil bilangan *Adler32Hex*.

Surat	<input type="text"/>
Adler32Dec	<input type="text"/>
Adler32Hex	<input type="text"/>
	<input type="button" value="Proses"/>
Hasil	<input type="text"/>

3.6 Gambar Rancangan Halaman *Adler32Dec* dan *Adler32Hex*

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap dimana aplikasi yang telah dirancang dijalankan. Tahap ini menunjukkan apakah setiap proses dapat berjalan dengan baik dan mampu memberikan hasil yang di harapkan. Proses perancangan aplikasi menggunakan *Microsoft Visual Studio* ditampilkan dalam bentuk form-from yang menjadi sarana bagi pengguna untuk melakukan proses implementasi.

4.2 Spesifikasi Sistem

Analisis kebutuhan sistem merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi merupakan elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

a. Analisis Perangkat Keras (Hardware)

Perangkat keras minimum yang digunakan untuk membangun Sistem

Informasi Penjualan ini adalah:

1. Processor Berkecepatan 3.0 Ghz
2. RAM 4 Gb
3. Hardisk minimal 10 Gb untuk menyimpan data
4. LAN Card
5. Keyboard dan Mouse
6. Monitor 20 inch.

b. Analisis Perangkat Lunak (Software)

Untuk mendukung dalam penyimpanan informasi, dibutuhkan suatu fasilitas yang memadai. Yaitu berupa perangkat lunak (software) yang dirancang untuk memudahkan dalam pembangunan dan menjalankan sistem nantinya.

Adapun perangkat lunak yang digunakan adalah sebagai berikut :

1. Microsoft Windows 10 , Windows 10 sebagai sistem operasi
2. Visual Studio 2010, Sebagai Perancangan Program Aplikasi.

4.3 Pengujian Sistem

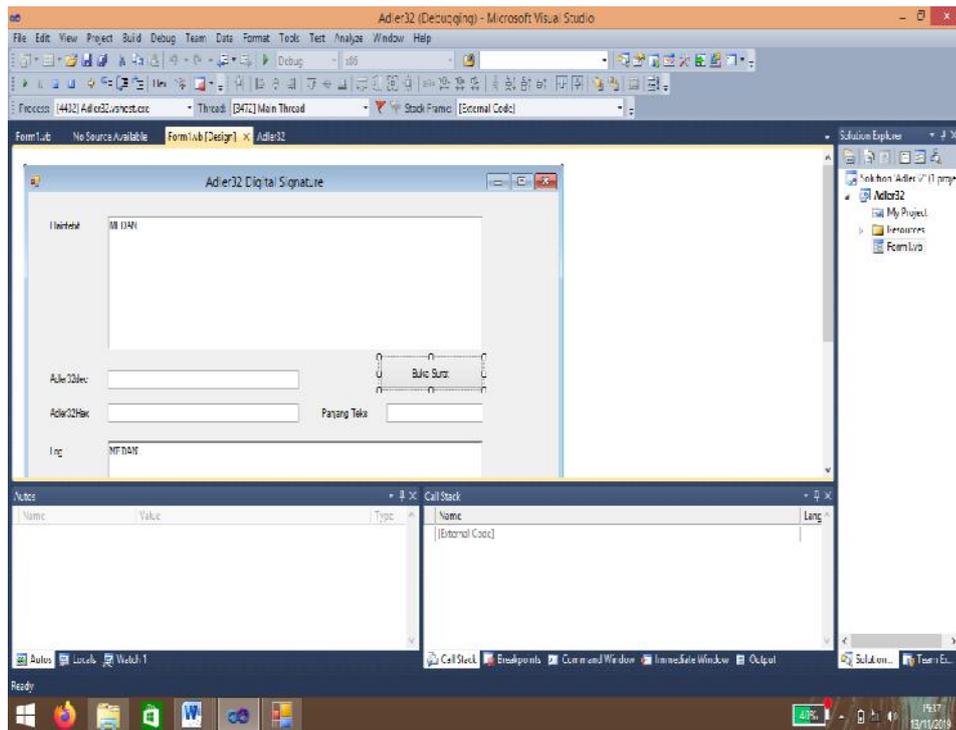
Pengujian sistem dilakukan untuk menunjukkan apakah sistem yang telah di rancang dapat berjalan sesuai harapan. Selain itu tujuan pengujian adalah untuk dapat menemukan kesalahan fungsi pada aplikasi yang di bangun dan memperbaikinya.

Pengujian dapat dilakukan dengan memasukan karakter atau huruf dari file berformat txt selanjutnya diproses oleh aplikasi, apakah aplikasi tersebut dapat memberikan hasil yang sesuai. Proses yang akan dilakukan pengujian dalam aplikasi ini adalah simulasi pengiriman pesan dapat menggunakan metode algoritma *Adler32* antara pengirim kepada si penerima dengan konversi bilangan.

4.4 Hasil Rancangan Sistem

a. Tampilan Menu Awal

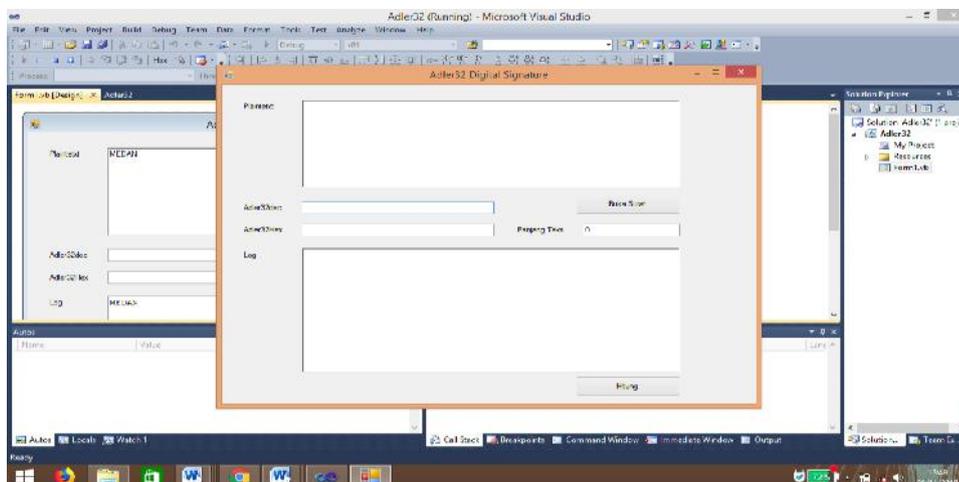
Rancangan tampilan adalah tampilan awal sebelum masuk ke aplikasi. Halaman ini berfungsi untuk memberikan hak akses bagi seorang user sebelum menggunakan aplikasi. Berikut tampilan Menu awal dapat dilihat pada gambar dibawah ini :



Gambar 4.7 Tampilan Menu Awal

b. Tampilan Menu Utama

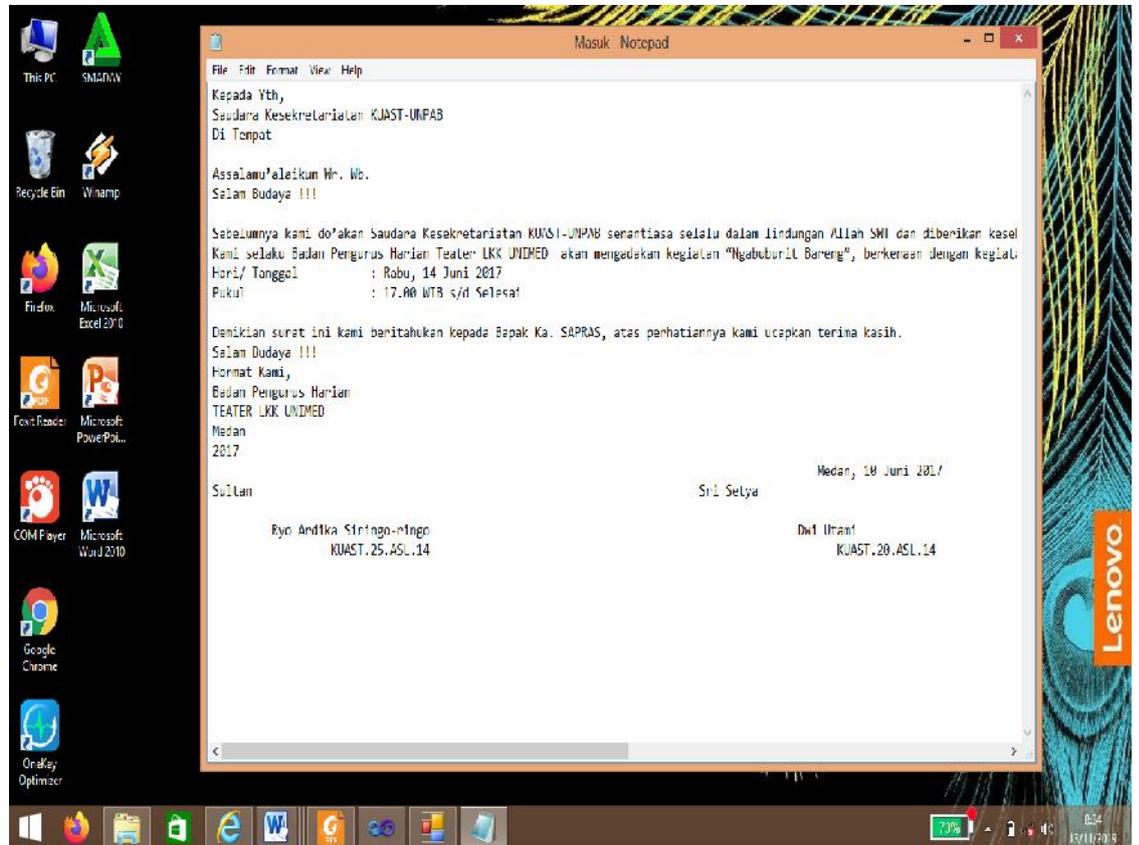
Tampilan Menu Utama merupakan tampilan yang pertama muncul saat program dijalankan. Di dalam menu utama terdapat menu seperti *plaintext*, *adler32dec*, *adler32hex*, *panjang teks*, dan *log*:



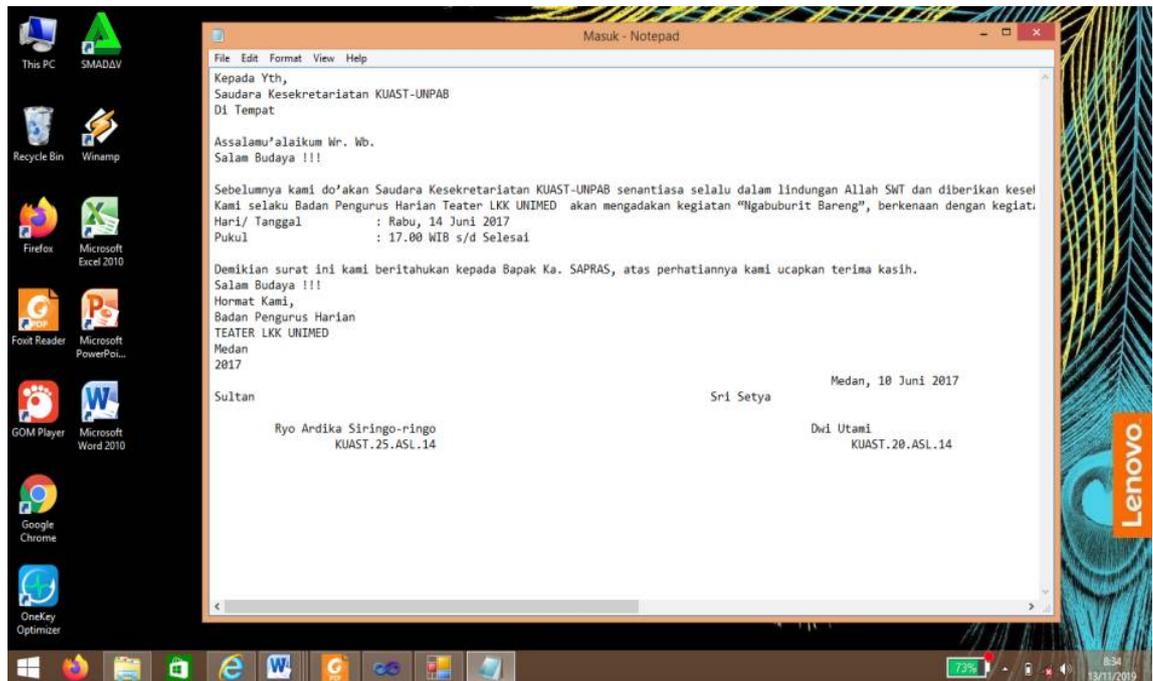
Gambar 4.8 Tampilan Menu Utama

c. Tampilan Surat

Tampilan Surat yang akan dihitung

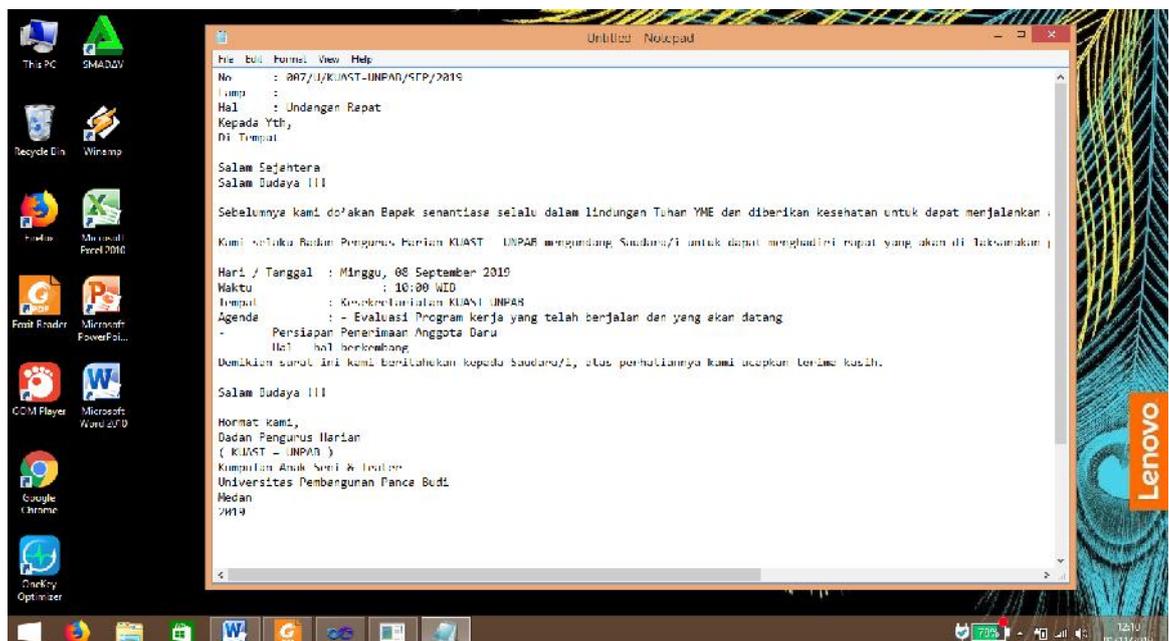


1. Surat Masuk



Gambar 4.9 Surat Masuk

2. Surat Keluar

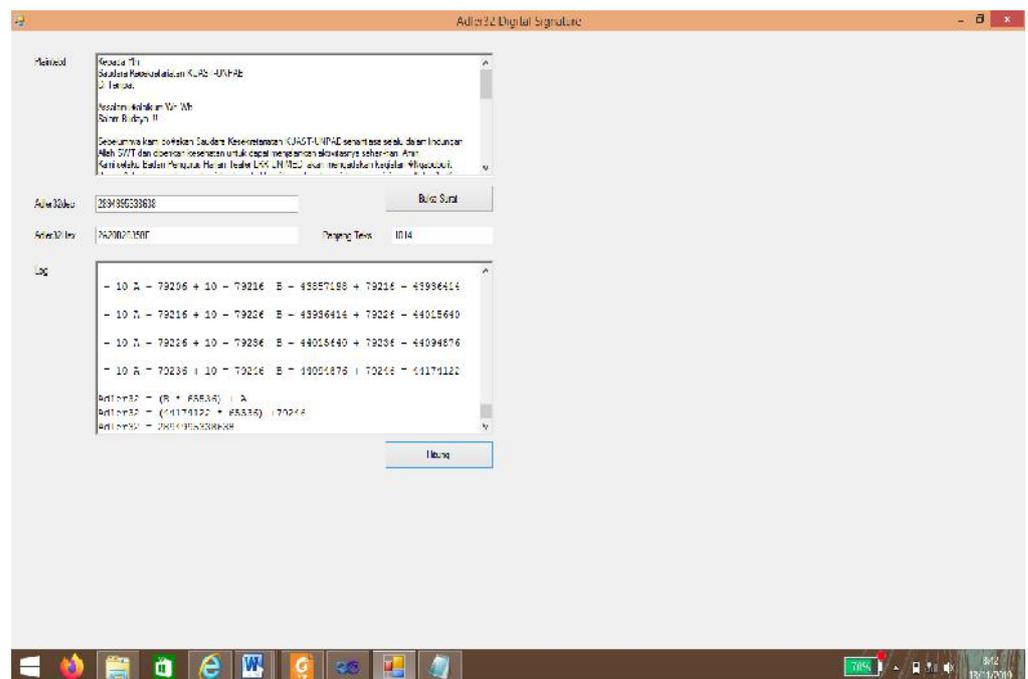


Gambar 4.10 Tampilan Surat Keluar

d. Tampilan Menu Konfersi Bilangan

Tampilan konfersi bilangan berfungsi untuk menggantikan tulisan asli menjadi tulisan yang disandikan dengan menggunakan algoritma *Adler32*. Untuk mengkonversikan tulisan tersebut, digunakan dengan cara menghitung dari desimal ke hexadesimal agar tidak mudah untuk dibuka:

1. Perhitungan Surat Masuk



Gambar 4.11 Tampilan hasil perhitungan Surat Masuk *adler32*

$$A = 1$$

$$B = 0$$

$$A (A + Plaintext =)$$

$$B (B + A =)$$

$$\text{Panjang Surat masuk} = 1014$$

Dengan hasil Log

$$\text{Adler32} = (B * 65536) + A$$

$$\text{Adler32} = (44174122 * 65536) + 79246$$

$$\text{Adler32} = 2894995338638$$

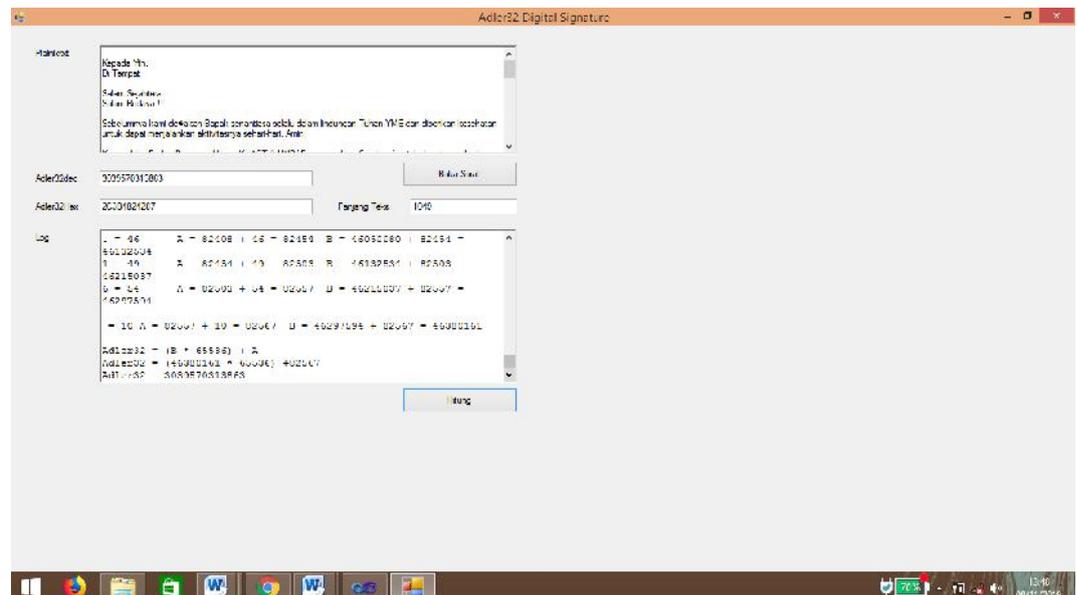
Hasil hitung konversi Bilangan Dari Surat Masuk Adler32Dec dan Adler32Hex

ialah:

$$\text{Adler32Dec} = 2894995338638$$

$$\text{Adler32Hex} = 2A20B2B358E$$

2. Perhitungan Surat Keluar



Gambar 4.12 Tampilan hasil perhitungan Surat Keluar *adler32*

Parameter *Adler32* :

$A = 1$

$B = 0$

$$A (A + Plaintext =)$$

$$B (B + A =)$$

$$\text{Panjang Surat Keluar} = 1014$$

Dengan hasil Log

$$\text{Adler32} = (B * 65536) + A$$

$$\text{Adler32} = (44174122 * 65536) + 79246$$

$$\text{Adler32} = 2894995338638$$

Hasil hitung konversi Bilangan Dari Surat Keluar *Adler32Dec* dan *Adler32Hex*

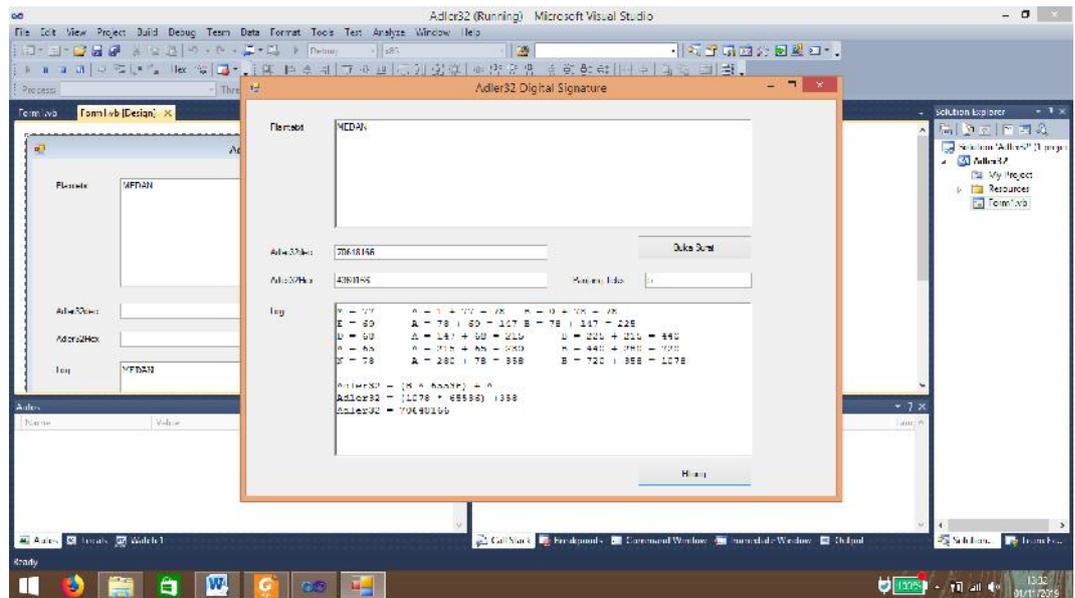
ialah:

$$\text{Adler32Dec} = 3039570313863$$

$$\text{Adler32Hex} = 2C3B4824287$$

e. Tampilan Hasil Hitung Adler32dec Ke Adler23Hex

Tampilan konversi Bilangan dari desimal ke hexadesial menampilkan tulisan dikembalikan ke dalam bentuk semula menggunakan teks MEDAN:



Gambar 4.13 Tampilan Hasil Hitung desimal ke hexadesimal

Parameter Adler32

$$A = 1$$

$$B = 0$$

$$A (A + Plaintext =)$$

$$B (B + A =)$$

Panjang Surat masuk = 5

Dengan hasil Log

$$\text{Adler32} = (B * 65536) + A$$

$$\text{Adler32} = (1078 * 65536) + 358$$

Adler32 = 70648166

Hasil hitung konversi Bilangan Dari plaintext MEDAN Adler32Dec dan

Adler32Hex ialah:

Adler32Dec = 70648166

Adler32Hex = 4360166

Tabel 4.9 konfersi Pesan kedalam kode ASCII

NO	KARAKTER	ASCII
1	M	77
2	E	69
3	D	68
4	A	65
5	N	78
HASIL		
$A = 1 + 77 = 78$ $B = 0 + 78 = 78$ $A = 78 + 69 =$ $B = 78 + 147 = 225$ $A = 147 + 68 = 215$ $B = 225 + 215 = 440$		

$$\begin{aligned}
 A &= 215 + 65 = 280 \\
 B &= 440 + 280 = 720 \\
 \\
 A &= 280 + 78 = 358 \\
 B &= 720 + 358 = \\
 &1078 \\
 \\
 \text{Adler32} &= (B * \\
 &65536) + A \\
 \text{Adler32} &= (1078 * \\
 &65536) + 358 \\
 \text{Adler32} &= 70648166
 \end{aligned}$$

4.3 Kelebihan dan Kekurangan Sistem

Adapun kelebihan dan kekurangan dari aplikasi Alder32 ini adalah sebagai berikut:

a. Kelebihan Sistem

1. Dapat menghitung nilai hash tulisan dengan algoritma adler32 dengan cepat.
2. Menjaga keaslian isi surat.

b. Kekurangan Sistem

1. Tidak dapat dijalankan dalam jaringan pada Komputer.
2. Hanya menggunakan format txt.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan dalam Penerapan digital signature pada surat masuk dan surat keluar dengan algoritma *adler32*, Nilai *adler-32* diperoleh oleh menggabungkan dua 16-bit A dan B. Ini akan menjadi 32-bit bilangan bulat. Karena tidak menggunakan aritmatika kompleks Ekspresi, itu dapat diterapkan pada gambar yang lebih besar. *Alder-32* dapat menghitung keaslian pengirim apa kirim ke penerima. Penelitian ini tidak menyediakan menyembunyikan informasi, itu hanya untuk memastikan apa yang dikirim kirim adalah apa yang penerima dapatkan. Gambar bisa membuktikan apapun dalam kehidupan nyata. Jadi harus asli jika digunakan sebagai bukti. Umpan baliknya pada penelitian ini membuat algoritma dapat bekerja bersama dengan pemrosesan gambar. maka dapat diambil kesimpulan sebagai berikut :

1. Penerapan *digital signature* pada surat masuk dan surat keluar dilakukan dengan cara menghitung pesan teks dari desimal ke hexadesimal yaitu setiap huruf yang ada.
2. Penggunaan algoritma *adler32* memiliki manfaat bagi pengguna aplikasi.
3. Pengamanan aplikasi menggunakan kriptografi dengan algoritma *adler32* ini sangat berguna dikarenakan proses dari angka desimal ke hexadesimal sulit untuk ditebak dan di bobol.

5.2 Saran

Adapun saran-saran yang dapat dilakukan penelitian ataupun pengembangan selanjutnya adalah sebagai berikut:

1. Perangkat lunak ini hanya dapat dikembangkan dengan menggunakan metode *adler32*.
2. Perangkat lunak ini dapat dikembangkan dan terhubung ke jaringan sehingga dapat dijalankan di lebih dari satu komputer.
3. Perangkat lunak ini dapat dikembangkan menggunakan algoritma-algoritma lain yang lebih kompleks.

Daftar pustaka

- Andysah putera utama siahaan(2016) a three-layer visual hash function using adler-32. International journal of computer science and software engineering (ijcsse), volume 5, issue 7, july 2016. Issn (online): 2409-4285 www.ijcsse.org page: 142-147
- Fachri, barany, agus perdana windarto, and ikhsan parinduri. "penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik." jepin (jurnal edukasi dan penelitian informatika) 5.2 (2019): 202-208.
- Fachri, b., windarto, a. P., & parinduri, i. (2019). Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. Jepin (jurnal edukasi dan penelitian informatika), 5(2), 202-208.
- Fachri, barany; windarto, agus perdana; parinduri, ikhsan. Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. Jepin (jurnal edukasi dan penelitian informatika), 2019, 5.2: 202-208
- Famalua gulo (2016) aplikasi pembelajaran konversi bilangan menggunakan metode computer assisted instruction (cai) jurnal riset komputer (jurikom), vol. 3 no. 6, desember 2016 issn 2407-389x (media cetak) hal : 34-37
- Hamdi, nurul. "model penyiraman otomatis pada tanaman cabe rawit berbasis programmable logic control." jurnal ilmiah core it: community research information technology 7.2 (2019).
- Mohamad natsir.(2017). Pengembangan prototype sistem kriptografi untuk enkripsi dan Dekripsi data office menggunakan metode blowfish dengan bahasa pemrograman Java. Jurnal format volume 6 nomor 1.95. <http://publikasi.mercubuana.ac.id/index.php/format/article/download/1532/1209>
- Ninuk wiliani & syadid zambi (2017). Rancangan bangun aplikasi kasir tiket nonton bola bareng pada x dengan visual basic 2010 dan mysql. Jurnal rekayasa informasi, vol. 6. No.2,278
- Permana, aminuddin indra. "kombinasi algoritma kriptografi one time pad dengan generate random keys dan vigenere cipher dengan kunci em2b." (2019).

- Putra, randi rian. "sistem informasi web pariwisata hutan mangrove di kelurahan belawan sicanang kecamatan medan belawan sebagai media promosi." jurnal ilmiah core it: community research information technology 7.2 (2019).
- Putra, randi rian, et al. "decision support system in selecting additional employees using multi-factor evaluation process method." (2019).
- Putra, randi rian. "implementasi metode backpropagation jaringan saraf tiruan dalam memprediksi pola pengunjung terhadap transaksi." jurti (jurnal teknologi informasi) 3.1 (2019): 16-20.
- Saputra, muhammad juanda, and nurul hamdi. "rancang bangun aplikasi sejarah kebudayaan aceh berbasis android studi kasus dinas kebudayaan dan pariwisata aceh." journal of informatics and computer science 5.2 (2019): 147-157
- Sidik, a. P., efendi, s., & suherman, s. (2019, june). Improving one-time pad algorithm on shamir's three-pass protocol scheme by using rsa and elgamal algorithms. In journal of physics: conference series (vol. 1235, no. 1, p. 012007). Iop publishing.
- Sitepu, n. B., zarlis, m., efendi, s., & dhany, h. W. (2019, august). Analysis of decision tree and smooth support vector machine methods on data mining. In journal of physics: conference series (vol. 1255, no. 1, p. 012067). Iop publishing.
- Sri wahyuni (2013) penerapan digital signature dengan algoritma sha-1 pada surat legalisasi ijazah dan transkrip nilai mahasiswa. Pelita informatika budi darma, volume : vii, nomor: 2, agustus 2014
- Tasril, v., wijaya, r. F., & widya, r. (2019). Aplikasi pintar belajar bimbingan dan konseling untuk siswa sma berbasis macromedia flash. Jurnal informasi komputer logika, 1(3).
- Yunahar heriyanto.(2018).perancangan sistem informasi rental mobil berbasis web pad apt.apm rent car. Jurnal intra-tech. Volume 2, no.2 oktober 2018.67-68.