**BAB II**

**TINJAUAN PUSTAKA**

**2.1 Analisis terdahulu**

Sebagai pembanding penelitian yang hendak dilakukan maka penting untuk mencantumkan beberapa penelitian terdahulu sebagai berikut:

1. Penelitian yang dilakukan oleh Eko Puji Laksono (2014), “Analisis komparasi algoritma kriptografi antara metode DES dan AES”. Penelitian ini fokus menganalisis pada perbandingan antara program Des dan AES.

Hasil penelitian :

1. Analisis enkripsi dan deskripsi menggunakan program DES lebih sederhana dibandingkan AES.
2. Langkah yang panjang membuat enkripsi dan deskripsi AES lebih aman
3. Penggunaan kriptografi AES 192 bit menghasilkan literasi lebih panjang dan komplek sehingga sulit untuk diretas

Kesimpulan yang didapat :

1. Analisis enkripsi dan deskripsi menggunakan DES lebih sederhana dibandingkan AES. Data DES diolah hanya dengan 4 langkah tanpa menggunakan ronde, sehingga hasilnya lebih sederhana dalam bentuk huruf yang jumlahnya sama dengan data yang diolah
2. DES tidak memiliki ronde. Enkripsi AES memiliki 9 ronde dan deskripsi AES memiliki 9 ronde
3. Analisis enkripsi dan deskripsi AES menggunakan algoritma kriptografi AES dengan panjang kunci 128 bit memiliki literasi yang cukup panjang sehingga lebih aman
4. Karakteristik enkripsi DES menggunakan rumus C=P+K+1)mod26 dan deskripsi DES menggunakan rumus P=(C+26-K-1)mod26. Karakteristik enkripsi AES setiaprondenya memiliki SubBytes, ShiftRows, MixColomn, dan RoundKey. Karakteristik deskripsi AES setiap rondenya memiliki invShiftRows, invSubBytes, RoundKey dan addRoundKey.
5. Penelitian yang dilakukan oleh Yusuf Kurniawan (2007), “Perbandingan analisis sandi linier terhadap AES, DES dan AE1”. Penelitian ini mengkaji tentang perbandingan analisis sandi AES, DES dan AE1.

Hasil penelitian :

1. AES memiliki ketahanan lebih besar untuk menghadapi ASL.
2. Ketahanan cipher terhadap analisis sandi tidak otomatis sehingga tahan terhadap sandi lain.
3. AE1 memiliki ketahanan yang lebih besar.
4. Pembuatan keamanan algoritma kriptografi lebih sulit dari pada pembuktiaannya.

Kesimpulan yang didapat :

1. AES-128 memiliki ketahanan yang besar untuk menghadapi ASL, karena analisis sandi tersebut hanya mampu memecahkan AES hingga 6 ronde, sedangkan AES-128 memiliki 10 ronde. Bandingkan dengan DES lengkap yang dapat dipecahkan ASL dengan 243 plaintext, di mana DES memiliki masukan 64 bit.
2. Ketahanan cipher terhadap sebuah analisis sandi tidak otomatis menyebabkannya tahan terhadap analisis sandi lainnya. Ini terlihat dari ketahanan Rijndael terhadap ASL setelah 4 ronde, namun dengan Square attack, 4 Ronde Rijndael dapat dipecahkan dengan mudah. Sebaliknya, DES yang lemah menghadapi ASD dan ASL, ternyata memiliki ketahanan yang besar terhadap square attack.
3. Pembuktian keamanan algoritma kriptografi lebih sulit dari pada pembuatan algoritmanya.

**2.2 Kriptografi**

Kriptografi (cryptography) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu kripto dan graphia. Kripto artinya menyembunyikan, sedangkan graphia artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data (Menezes, Oorschot and Vanstone, 1996). Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi. Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan.

Menurut Munir (2006) ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

* [Kerahasiaan](http://id.wikipedia.org/wiki/Kerahasiaan), adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau [kunci rahasia](http://id.wikipedia.org/wiki/Kunci_%28kriptografi%29) untuk membuka/mengupas informasi yang telah [disandi](http://id.wikipedia.org/wiki/Enkripsi).
* [Integritas](http://id.wikipedia.org/w/index.php?title=Integritas&action=edit&redlink=1) data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
* [Autentikasi](http://id.wikipedia.org/w/index.php?title=Autentikasi&action=edit&redlink=1), adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
* [Non-repudiasi.](http://id.wikipedia.org/w/index.php?title=Non-repudiasi.&action=edit&redlink=1), atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

Berikut adalah istilah-istilah yang digunakan dalam bidang kriptografi (Munir, 2006) :

* **Plaintext** (M) adalah pesan yang hendak dikirimkan (berisi data asli).
* **Ciphertext** (C) adalah pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi.
* **Enkripsi** (fungsi E) adalah proses pengubahan *plaintext* menjadi *ciphertext*.
* **Dekripsi** (fungsi D) adalah kebalikan dari enkripsi yakni mengubah *ciphertext* menjadi *plaintext*, sehingga berupa data awal/asli.
* **Kunci** adalah suatu bilangan yang dirahasiakan yang digunakan dalam proses enkripsi dan dekripsi.

Kriptografi itu sendiri terdiri dari dua proses utama yakni proses enkripsi dan proses dekripsi. Seperti yang telah dijelaskan di atas, proses enkripsi mengubah *plaintext* menjadi *ciphertext* (dengan menggunakan kunci tertentu) sehingga isi informasi pada pesan tersebut sukar dimengerti (Munir, 2006).

plaintext ciphertext plaintext

**enkripsi**

**dekripsi**

kunci enkripsi kunci dekripsi

Gambar1 Diagramproses enkripsi dan dekripsi

Peranan kunci sangatlah penting dalam proses enkripsi dan dekripsi (disamping pula algoritma yang digunakan) sehingga kerahasiaannya sangatlah penting, apabila kerahasiaannya terbongkar, maka isi dari pesan dapat diketahui.

Secara matematis (Munir, 2006), proses enkripsi merupakan pengoperasian fungsi E (enkripsi) menggunakan e (kunci enkripsi) pada M *(plaintext)* sehingga dihasilkan C *(ciphertext),* notasinya

**Ee(M) – C**

Sedangkan untuk proses dekripsi (Munir, 2006), merupakan pengoperasian fungsi D (dekripsi) menggunakan d (kunci dekripsi) pada C *(ciphertext)* sehingga dihasilkan M *(plaintext),* notasinya :

**Dd(C) = M**

Sehingga dari dua hubungan diatas berlaku :

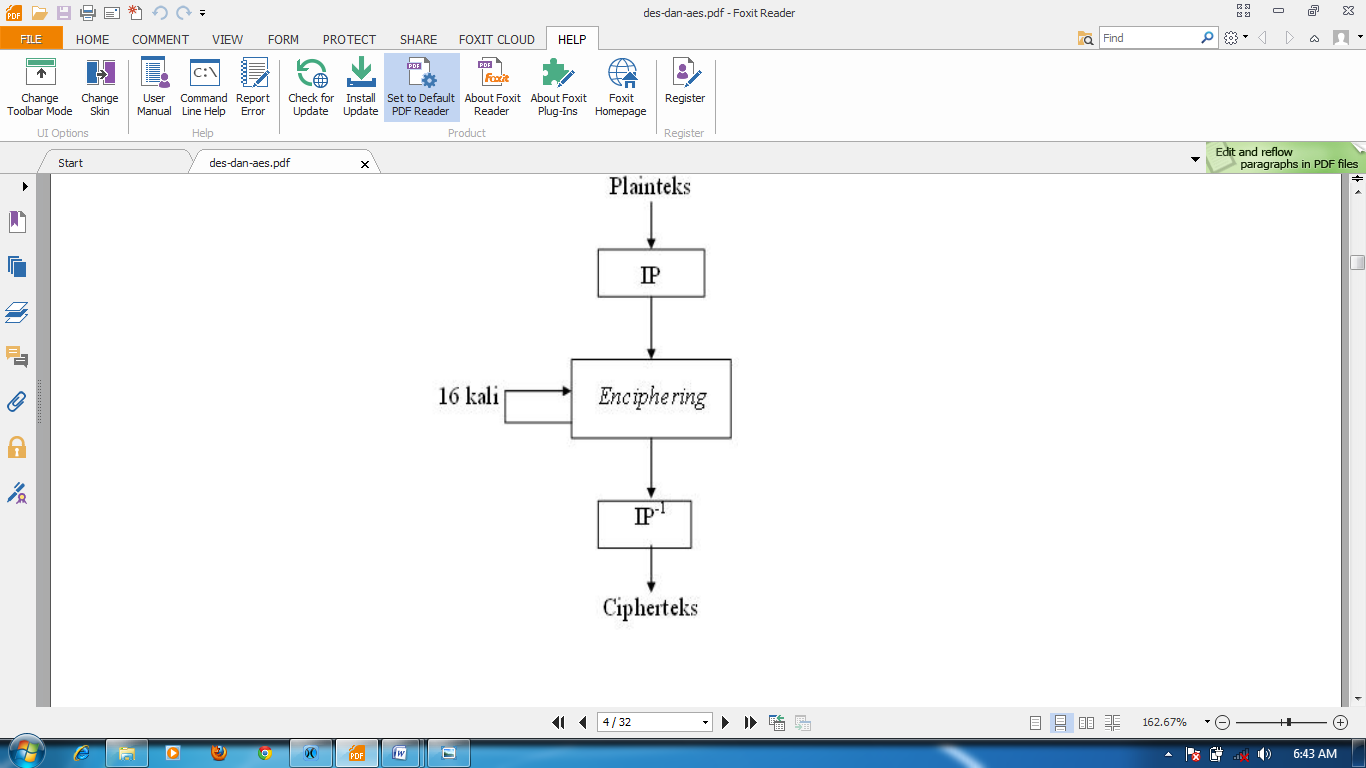
**Dd(Ee(M)) = M**

**2.3 *Data Encryption Standard* (DES)**

DES merupakan salah satu algoritma *Chiper Block* dengan ukuran 64 bit dan ukuran kunci 56 bit. DES dibuat dengan menggunakan algoritma terdahulu yang bernama Lucifer oleh IBM. Des pertama kali dipublikasikan di Federal Register pada 17 Maret 1975, setelah itu DES diadopsi sebagai algoritma standar yang digunakan oleh NBS (*National Bureau of Standards*) 15 januari 1977. Sejak saat itu DES digunakan untuk melindungi data agar tidak bisa dibaca oleh orang lain.

Namun demikian, DES juga mengundang banyak kontroversi dari para ahli di seluruh dunia. Salah satu kontroversi tersebut adalah *S-Box* yang digunakan pada DES. *S-Box* merupakan bagian vital dari DES karena merupakan bagian yang paling sulit dipecahkan. Hal ini disebabkan karena *S-Box* merupakan satu – satunya bagian dari DES yang komputasinya tidak linear. Sementara itu, rancangan dari *S-Box* sendiri tidak diberitahukan kepada publik. Karena itulah, banyak yang curiga bahwa *S-Box* dirancang sedemikian rupa sehingga memberikan trapdoor kepada NSA agar NSA bisa membongkar semua ciphertext yang dienkripsi dengan DES kapan saja. Kontroversi yang kedua adalah jumlah bit pada kunci DES yang dianggap terlalu kecil, hanya 56 bit. Akibatnya DES rawan terhadap serangan brute force.

**2.3.1 Panjang kunci dan ukuran Blok DES**

**** Algoritma DES dikembangkan di IBM dibawah kepemimpinan W.L. Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma Lucifer yang dibuat oleh Horst Feistel. Algoritma ini telah disetujui oleh National Bureau of Standard (NBS) setelah penilaian kekuatannya oleh National Security Agency (NSA) Amerika Serikat. DES termasuk ke dalam kriptografi kunci-simetri dan tergolong jenis cipher blok. DES beroperasi pada ukuran blok 64 bit. Panjang kunci ekternal = 64 bit (sesuai ukuran blok), tetapi hanya 56 bit yang dipakai (8 bit paritas tidak digunakan). Setiap blok (plainteks atau cipherteks) dienkripsi dalam 16 putaran. Setiap putaran menggunakan kunci internal berbeda. Kunci internal (56-bit) dibangkitkan dari kunci eksternal. Setiap blok mengalami permutasi awal (IP), 16 putaran enciphering, dan inversi permutasi awal (IP-1). (lihat Gambar 2)

Gambar 2 Skema Global Algoritma DES

* 1. Blok plainteks dipermutasi dengan matriks permutasi awal (initial permutation atau IP).
  2. Hasil permutasi awal kemudian di-enciphering - sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
  3. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (invers initial permutation atau IP-1 ) menjadi blok cipherteks.

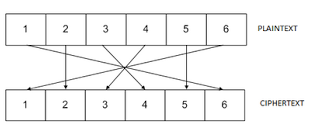
**2.3.2 Teknik Dasar Kriptografi**

1. Substitusi

Dalam kriptografi, sandi substitusi adalah jenis metode enkripsi dimana setiap satuan pada teks terang digantikan oleh teks tersandi dengan siste yang teratur. Metode penyandian substitusi telah dipakai dari zaman dulu (kriptografi klasik) hingga kini (kriptografi modern), langkah pertama adalah membuat suatu tabel substitusi. Tabel substitusi dapat dibuat sesuka hati, dengan catatan bahwa penerima pesan memiliki tabel yang sama untuk keperluan decrypt. Bila tabel substitusi dibuat secara acak, akan semakin suli pemecahan *chipertext* oleh orang yang tidak berhak. Metode ini dilakukan dengan mengganti setiap huruf dari teks asli dengan huruf lain sebagai huruf sandi yang telah didefinisikan sebelumnya olehalgoritma kunci. (www.mercubuana.ac.id)

1. Permutasi

Salah satu teknik yang penting pada enkripsi adalah permutasi atau sering juga disebut transposisi. Teknik ini memindahkan atau menotasikan karakter dengan aturan tertentu dengan prinsip yang berlawanan dengan teknik substitusi. Dalam teknik substitusi, karakter berada pada posisi yang tetap tapi identitasnya yang diacak. Pada teknik permutasi, identitas karakternya tetap, namun posisinya yang diacak. Sebelum dilakukan permutasi, umumnya plaintext terlebih dahulu dibagi menjadi blok – blok dengan panjang yang sama. *Plaintext* akan dibagi menjadi blok – blok yang terdiri dari 6 karakter, dengan aturan permutasi (www.mercubuana.ac.id), sebagai berikut :



CHIPERTEXT

PLAINTEXT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |

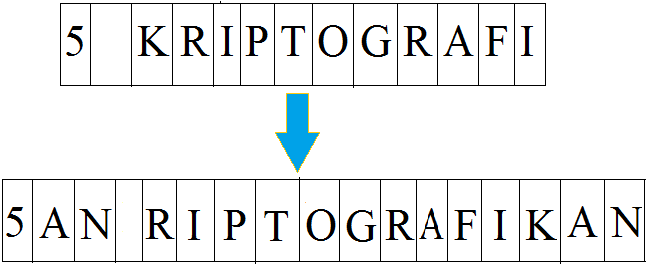
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |

Gambar 4 Enkripsi dengan Permutasi

Dari gambar diatas dapat kita lihat proses permutasi yang menjadikan plainteks : 1 2 3 4 5 6 menjadi chiperteks : 6 2 4 3 5 1

1. Ekspansi

Suatu metode sederhana untuk mengacak pesan adalah dengan memperlebar pesan itu dengan aturan tertentu. Salah satu contoh penggunaan teknik ini adalah dengan meletakkan huruf konsonan atau bilangan ganjil yang menjadi awal dari suatu kata di akhir kata itu dan menambahkan akhiran “an”. Bila suatu kata dimulai dengan huruf vokal atau bilangan genap, ditambahkan akhiran “i”. proses enkripsi dengan cara ekspansi terhadap plaintext. (www.mercubuana.ac.id)



Gambar 5 Enkripsi dengan Ekspansi

*Chipertext* adalah “5AN RIPTOGRAFIKAN”. Aturan ekspansi dapat dibuat lebih kompleks dengan menggabungkan dengan teknik lainnya.

***2.4 Triple Data Encryption* (3DES)**

Menurut Hidayat, 2010 *Triple* DES adalah sebuah *chiper* blok yang dibentuk oleh DES dengan menggunakan tiga kali. *Triple* DES atau 3DES menggunakan DES 3 kali. Penggunaan tiga langkah ini penting untuk mencegah *meet* – *in* – *middle* *attack* sebagai mana pada *Double* DES.

*Triple* DES memiliki 2 varian yakni EEE dan EDE.

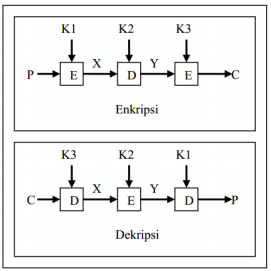
1. EEE adalah enkripsi menggunakan 3 kunci yaitu K1, K2, dan K3 dimana semua kunci berbeda sehingga panjang kunci yang didapat 128 bit, 3x dari panjang kunci DES.

Bentuk sederhana dari 3DES EEE adalah :

Enkripsi : C = Ek3 (Ek2(Ek1(P)))

Dekripsi : P = Dk1 (Dk2(Dk3(C)))

1. EDE adalah enkripsi DES tunggal dengan kunci K1 = K2 = K3. Gambar dibawah ini memperlihatkan versi 3DES dengan 2 buah kunci. Penggunaan enkripsi tidak mempengaruhi keamanan algoritma.



Gambar 6 Diagram enkripsi dan dekripsi 3DES (Hidayat,2010)

Perlu diingat bahwa DES bukanlah sebuah grup (dalam matematika), karena jika merupakan grup, pembangunan algoritma 3DES akan ekuivalen dengan operasi algoritma DES yang berarti tidak aman. Varian ini umum dikenal dengan metode penyandian EEE. Untuk menyederhanakan *interoperability* antara DES dan 3DES, maka pada langkah di tengah (pada proses enkripsi 3DES) diganti dengan dekripsi (mode EDE).

* + 1. **Algoritma *Triple* DES**

Algoritma enkripsi atau dekripsi *Triple* DES seperti algoritma kriptografi lainnya yaitu memiliki algoritma umum. 3DES merupakan suatu algorirma pengembangan dari algoritma DES.

P

K2

K3

K1

Plaintext

f

K1

⊕

R2=L1⊕f(R1,K2)

L2=R1

L0

R0

f

K2

⊕

R2=L1⊕f(R1,K2)

L2=R1

R15=L14⊕f(R14,K15)

L15=R14

f

K16

⊕

R16=L15⊕f(R15,K16)

L16=R15

Chipertext

P

P1

Plaintext

f

K1

⊕

R2=L1⊕f(R1,K2)

L2=R1

L0

R0

f

K2

⊕

R2=L1⊕f(R1,K2)

L2=R1

R15=L14⊕f(R14,K15)

L15=R14

f

K16

⊕

R16=L15⊕f(R15,K16)

L16=R15

Chipertext

P

P1

Plaintext

f

K1

⊕

R2=L1⊕f(R1,K2)

L2=R1

L0

R0

f

K2

⊕

R2=L1⊕f(R1,K2)

L2=R1

R15=L14⊕f(R14,K15)

L15=R14

f

K16

⊕

R16=L15⊕f(R15,K16)

L16=R15

Chipertext

P

P1

C

Gambar 7 Tahapan algoritma 3 DES (Hidayat, 2010)

Gambar diatas menjelaskan tentang tahapan algoritma 3DES. Pada dasarnya algoritma yang digunakan sama. Hanya pada 3DES dikembangkan dengan melakukan enkripsi dengan implementasi algoritma DES sebanyak tiga kali.

3DES memiliki 3 buah kunci yang berukuran 168-bit (tiga kali 56 bit dari DES). Pada algoritma 3DES dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma DES.

Tahap pertama, plainteks yang diinputkan dioperasikan dengan kunci eksternal pertama (K1) dan melakukan proses enkripsi dengan menggunakan algoritma DES, sehingga menghasilkan pra-chiperteks pertama. Tahap kedua, pra-chiperteks pertama yang dihasilkan pada tahap pertama, kemudian dioperasikan dengan kunci eksternal (K2) dan melakukan proses enkripsi atau proses dekripsi (tergantung cara pengenkripsian yang digunakan) dengan menggunakan algoritma DES. Sehingga menghasilkan pra-chiperteks kedua. Tahap terakhir, pra-chiperteks kedua yang dihasilkan pada tahap kedua, dioperasikan dengan kunci eksternal (K3) dan melakukan proses enkripsi dengan menggunakan algoritma DES, sehingga menghasilkan chiperteks (C).

* + - 1. **Proses Enkripsi *Triple* DES**

Proses enkripsi algoritma 3DES dapat dicapai dengan menggunakan tiga kunci :

Enkripsi : C = EK3(EK2(EK1(P)))

Penjelasan :

Enkripsi pesan P mula – mula dengan kunci1 lalu hasilnya di enkripsi lagi dengan kunci K2, kemudian dienkripsi lagi dengan kunci K3 dan hasil enkripsi terakhir adalah chiperteks.

* + - 1. **Proses Dekripsi *Triple* DES**

Proses dekripsi algoritma 3DES dapat dicapai dengan menggunakan tiga kunci :

Dekripsi : P = DK1(DK2(DK3(C)))

Penjelasan :

Mula – mula kunci K3 digunakan untuk mendekripsi C, lalu hasilnya didekripsi lagi dengan kunci K2, kemudian didekripsi lagi dengan kunci K1 dan hasil dekripsi terakhir adalah pesan semula (P).

**2.4.2 Keamanan *Triple* DES**

Secara umum 3DES dengan tiga kunci berbeda memiliki kunci berukuran 168-bit (3 kali kunci 56-bit dari DES), namun dengan metode *meet-in-the-middle* keamanan yang diberikan hanyalah 112-bit. Sebuah varian, Double 3DES, menggunakan kunci K1=K3, yang berarti mengecilkan ukuran ke 112-bit dan ukuran storage menjadi 128-bit.

*Meet-in-middle attack* adalah salah satu serangan terhadap kriptografi yang mencari nilai dalam setiap jarak dan domain dari komposisi dua fungsi sehingga *forward mapping* dari fungsi pertama sama dengan *image inverse* yang lainnya ke fungsi kedua. (Dian Intania, 2006)

**2.4.3 Implementasi Kriptografi**

Menurut Munir (2004), kriptografi banyak diguakan dalam kehidupan sehari – hari antara lain :

* Pengiriman data melalui saluran komunikasi
* Penyimpanan data di dalam disk penyimpanan
* *Automatic Teller Machine* (ATM)
* Telepon genggam (*Handphone)*
* Keamanan Jaringan Internet

Contoh aplikasi kriptografi pada file dokumen :

Plainteks (plain.doc) :

Ketika saya berjalan – jalan di pantai, saya menemukan banyak sekali kepiting yang merangkak menuju laut. Mereka adalah anak-anak kepiting yang baru menetas dari dalam pasir. Naluri mereka mengatakan bahwa laut adalah tempat kehidupan mereka.

Chiperteks (chiper.doc) :

ꋥ䥂蕫候垭ꦆ㭕⏂䙢慒컒띈ᵬ㕞仗陹儕椞꼼뼌쟞薹䱂鱦뜵督⹬Ꙁ걧䡺΅⏆릗㔝୏㶕鑣ඉ먕㵁㡸㇄坅沬綄΀癶텃첁掻㇡ࢀ骋Ψ듦醼鏨黄鵮ᬝ⡰敓๥銂ׁත縆珇膃站魅菲庉잚ꑋ橽ꙶ⸅蓜℆鱌詜佄쀜膚㝧塔頺桳督㷓掅쇁∘⫭瘏褢惣幻揷ዠ렛ඦ틵擒醩⓱ਯࣞ叽쬛쮪蛺ꨩ뀹钆브褲捲惫윳⸭䷱緊얞꣘첱䃏券涏꿼霸입ᦺ⧄耍컝뭍쌧뒼ൊ桁롍㡲䯒笯呝按吁⪃햗ڟ丨긐蚀抧唌褥辄䶅婝臇问닁픲褍₍꽰궝ꔀ봈魟躗ꇘ픅핻燿ꩩ⬃시덠ቲ퀆巠摧ꎦ絣퀇㋚힯⥀ᑫ࡬쵮䤀ᩬ碴Ῡ奌哐휵ᔠ䎂㼕ᶞ䎬ヒﾟⴏ햯澩㢵麦攮넍嚹⍯暷슦᫛䫚㨐潋䡧ż⦥ᰒ閙诺魥ʢ㜥창敆짉꼶㾺橭앤톒馻Ძ༣䲽Ì嵣蓪წ銲偒ꎅ簑Ꮒ袌ꦙ훩〃ᦩ䀴㸜䪍鈆䑸ꤾ㝀ᓅᇎӰ禋팿驣㶖↲엳쳞䈩긝眉牑ꆹ笵勊ậ콨뙋ଢ଼셳屧䝽㼛拟跣靜᷃᩵帠ഐ鲷伛ᛱ晸䘚ꬉ啞㥙쩎厶檷橌닻螷⫨鰺ᄝ୕ꖿ뱫軙걌⡘鑈㘴芇ඉ娵⽌灟툓赲ﮭ탊섳摭ʍゅ斟ㄪ䈜녎晊⚓钅菉壦섉炆糄嵻覕廒⧚忈祳墵䲾ၧ場p争豜鶄칤㲻誜덆萳銅坄䩴鱇똲ᢓ㹊偺飑罈䕣䁥渫㰏䤯沜뷚䆾ƺ뭼폮鞌渡┓䛈硗⸛儤왂ၢ릥ڳ醢䝈Ɇಈ麦쓀㙧⭮뀸⻤᪌㶜ㄫ㗺ᓩ絽餆ꗖ翡癭솭ࡗ픔죚䂤鍵栺᥌麎릫翋㋇ᘋ떗⒯䀰ᚎ緩㑆鰒媣꯿㜙馯깿뎆㘆톼㸲⡀⡠凌㯑㼠抆㔸餯ᩍ摓ꏚ嚖읋의䚼给㽵դ辝쯧놅￝„醆ꪫ䜲ɵջᆁ芁鑛ⵄ免㟄兂뱪ꟓ诊ᤰᶿ膐䛁䮐쭡㫆墚䜹淐懵敛᫾긳鼄퇩哪ď⼜磴묶ﯕ댡鎞钴ꅚ沈즺뮆흁첝淍雛ꢬꆫ壳릱ᬬ蚩킼锟܃奼낍㿗槙媉인஗ࣽ雨殀혎崽䴏崹蔙⨨潏把핮㴜ན夽ￂへ뺋夰〛蒩㇚雘놾ﾈ阵␽č鎯⻬控嘹⚍윮ㆶஹ荇潍㝧眉ᮔ㜁᮪䄰菕杌ᢅ཰␐飿몸ൄ豈윁ᤚ뻰⬲緘뺮⇬⩣ⴼ칚ᮑ㣇駲撣쏑諹א㍱龉㏰𧻓饣䞺淘ၗ꒞퐎ᴄ廏ꢙ䬦鎛㽤쌢崝Ტ焦䈝꫕ꔺ蟟崓ᑒ捘딉넕㌉ꑭ뒬ඝ걫閪榰检耇ꨝ䗒⧫荸慾䡃猛菔┖ɳ袘㟛㋾冴㮗褫Ȝ餗鎦첱땤涹郼ﱺð輣䵭馧齿땋ꇁ蠚臧삇ﶽﭺἈᠼퟻ⢳둯䳗㋭뒤沌哹⮧፿賫⎔烾옧ኟ◦ᶉ쳪ꋮ润뛽❻㳞晲須䑀늴沋⾆䎮鉻챓ٱ㷭ꨢ遂襘ߐ쒑巎媩칈뼽ڂႦꔷ퀮芔塠쾗腋䄄᫰뷵ꡋ뚍῁䗻㌊喏訦欛㍷ኮ螼䲂絟馑ꯘ쿳꫃썲撲쁋ࡠ譎ꦟႦ鮆䧺୫礰㔴껎泋鋐詰㈵䊧㑁ԝ칻䋰튯䐫됑՝ສ멄༔᪖ꚰ鹐␊䦲嵉蓿稆晏廥漀ᕮ씃ᛅﻵ淸춛懝⾇鈅㔊뚊﬿욈亯ሬ惄䄮緆㍢礈幣戝ᢂ¾⋬ᓼ턑䇿≻ᐌᄦ뿝覂瘱ೈ笝吙풭蕈猫㉘‭聐澋ᭃ༪淥㘫萈펣ꃂ怌כ䫯ᒤ霉ﱪ끲쿣滀㗺ᬕᾜ퟿ꭱ瘨ꓫ춞蝧慼ㅪ۲ꏞ᯻幝傫賾Θ쾤냺堑됮ᩁ킑ⷽ᫤椺吥䦌ࣲ幌ᐓ絕ꈚ옎ꟓ笢䕝ભ⧜ጤ쿒⡣宂꫿U텻㰵쩰쬻쁽ᗓ仠㛍꫔銬ꊱ䎇ꕅ鹩暹㤂쵳㺢֘騃㩉ᩆⶰ쮜䠖ﵘ鹖迒︜ᛃ玿奧ﺁƆ䰖芕䰣⩊䑲튢轁긲뽝跋鯩는槒Ⓚ䕽祜䑷ٺ盹ꩪȞ≨⢖镼儇ᆍ뿪ᾬ䶲ᠧ꾹㾏偉籭ꚵ솰⤑遗钝ｳ婝箏ｋ塽即氇瞔䯷킨앸ꡈ퍦愫忼⨖↊횼䙧渲䬗ᯐ剩컀ↀ漡旖䑷隉ᦧﴭ뚵׃饐쾁ꈒᯎ픞↿ⰷ蜅᎛滉솨슪읪䉽䫧襂ࠇᜄ퀝넧᫊揜鞼⌓㏶绯梢䯘␾梄홬䉣㠮ꖎ꨻띣鋝奍ᇊ侁欳饯☦ଫᴢ蝐㟥ꂲ墢靓曾鱬斷髱❰⻂⸭皜ᧀ੹ꆘ嬎䪦ᮓᴗ䘠㚊藷⃳箣삇೙慱㲴噆䲇ꋏ믎㰸轀澉暝韤撶㺇뮯㰩可⩠侅骤᰻㐲气ﳓ㷦蚣纣췼㙟ꄧ冮湵Ҙ醦씕얠몯ﻋ來⨧礼끗ᙊ亻届䚱ષ廥빏ཪꮹ諼䁻큖퐋䑗㇇ळዟ斸糌䒄잞䋋લ黅䷢䪧廡序惞䜕愮ﰉ쑙챷佧꟨놡旓癃鹶レ雴꿃총꣜쯭㱷ꂠÒ훤༇✬ώＪ핦陨毐⮉弭䅷¼䍍由ⓡ銣क魪䧠阔켫㇟﹚叏䵷먢뚉ﶮ䝐㔜菉텕鼄礐⢮൅ꅲꦒ琒餁긂堈욐㈐䤭댞糀㴲䙩啀ﱞ꙯ꎼ頸⇚鱡탎蛇⌙㡜㙇䙃瘟䧃㪁㋈끝뵝䝗ﻁ㓸솅톐瞝莀띤⾬ቨￎ켓ﬗ㨰쐊崵溠䙁裭ㆃ蒤൵䃜ꕏ憐禾㑀鬸糺﻽㴬脓墼앸惛웳蠻쿨嚇秩Ꭾ刷忓㠛붤矋Ჽ䣆궑൐ㄆ胴뵾孋헊礵П◞뒻곉쌠✨粊礴հ쵫粦虴뭘ࡇ돨훇睊喝へ쇭션㌙欓쉯窗溣醛Ｖ㱫㇅ࡗᵅ￙ᑮિᕡ쀉켩ᩙ씋ꋄԢ뙵ᷩ껯㜳ꇉᢱሒ龑ᦨᒓ⸱紩⚎爏䶹飖⿃穈㼸☨⫓ꓶ嬇냷룵犀翖㈙쵌洌㔝譝誵为銷苣ꍺ儤獢﷤퉒ꟛₗ揾擒엾ﾠ䀖ផ痀〈஁䣞ఢ᧢ҕꄝ歹⺼埞㯎㼚陧Ჳ꧎䷥攩㘶姳恷꟡羡ꭉ㹋峷鿈額ᨎ㻰镱䚑쩗겙Ḩ뇦째ᒝ쾴ṕ㟢๖拮碓⑴쇬Ǯ᯻툶ᒻ熰鰂⫸Ҩ䣺鈽䧝鎍⩾믈õ㴯심受㷙莏⟋䕶况⨝ꥦ樂鹔▔㯗囘釟ήꭣ죫俑兂矛괨櫾䜠颐옉젭墇聕뾍孤쐀귚敛䰨塎ԫ⣥飁ᰦ뙙胲盖ݰ㯋퇬굽꿣ܱ䫫Ǚᵵ꿊ᶡၧ쯳딾盐ꪕ쫾숻벿䮲眕샾⏽늅蒹Ᵹ庠ꚛᘉ᳓痔甽쁗䅡࣎꟪୮纱⧀질ʬ߫㈐䘰⯿灶↷㙢産죈쓷ﰤຜ冯☁泵ˈᕸ옻ⶆ꓌㲢㫔趟Іྃඬᔒ䊲謃깬翜㔀ࢣ䀴ꢆ⟨ุ둠쐾로俀砲됓懞䚵릅獴훤呠讷뜲ㅎ⃛慩퐽觯䏲᳊뎿鬿와欠ቼ殶䛶홲⋋캇ᔙî斗္丄뺐琢尾委춲⑬ۚ詨✅꿭둺챼ꛞ㣍⁜㝙袎鸊羷ɥ꾞妘䙨쫌귢焾纭ޚٔᮿ㩩ᶮ៼ቅ걦머퓴똅컳쁮䢍쥇ࢃ价촗ۊ慒⋣䤷꜈荈붙ݻ偊쀿뒥镵絉뼞ꮍ钾䶸됍ᦼ뷙凞외ৗ虵▦蜠誘〆蔟ꡞ䡩ࡄ㝬ꭝ቙ꃶӈ硒悛곉⛃䏘薯허Ꭱ퇦ꎎۭꜟ击飛㨖⍍뷦盚쇟濴፣ꗌ㟭ⵑꎈ콣晤ﴟ䳸챬跸兼ﰉ燰ꜥ䟑酈퟽䂖ꪇ驨藡㔎戙ꨎ鿑禗垻洴퓉두蠧騰玸䕳쁆¤Ố獗⯦̆秐澷弭濦鏈釾뒟藱ҝ訷河ꒅᙞ䖃ﰨ헔驒ೆ忭䠼ๆᔙ儑ํ汋페﷛㹟⡔ⱶ媳四먁餆硋啲䏺ὅ䇬檛폃篮툩猎ᆑ웒頱픩⣔糧楢᧚䶉ﳢ︇㐌뚒㢚㝣卸劢⢈ꦶ⃴鐭辄瞦뼅쟔坖砏김鞵㵔懧䪙ꃡ爇য㬵퉍甛츺礐疹Ⅲ쨿܊諒쾿頉ꔇ㾖ﱧ이䦈᪂껕ꎕ읹໲岙⑁냵畟녇豪랛뇫╹Ǹꇾ䔰鍸櫋⌸튦彇饎鬬엧ᇮꞌ䟡ꋜ啔ⱀ嗻璻籣끢즓㐊菘Ὀ慓黬㮊翕ȃ㓍鵐㶪㳟ჿ澼㼗僗銽崂ⴈ䬗恏竏뭙﫢寘防巏䍺䳺긗乓咑᫝礱彼봉ዖĮ㮨땟禵荜擄⁏⎩Ḁ魜嗖紐莣켑楞觕睠ᶷ袦咭䰫ᥰ⭁匇૶䧡瞐ヌ₀눒ෛ缇⫚チ푋炰둎鹛⬍叵䋔㲴㜭鬲䴘⾛鳆疦༢﹫窶㘹牔寠唪ር筑㴠ళ㛐䇟女泿攷䟯᫜Ř썄肴싚㗧뜙粥㽳믄厡풕᫓ᄑ審⍚偵䒖⤩䋻锵怒蓇装䞦ᢡ뿾﷿棾륀芚腤䫩凂삘蒼ࣸ磴᳂덀幻찜﮺姌崶땒ﻌ읐鷘ᩮ騜蛄뫉臖䷰腆컎긢罃緎㹊਎▦댻౑㛿缈隲ᤄ륷懄ㄳ骟⤜帞撯☖﫽妊櫸ꇠ혨姡旬哶ꕴ큹쨻쭧ꣲ硆꟯꒴ඵ赃͙랜폎ၲ츭䣿பẳ뜞ๅ뵒篔酗๢抌房돋ꯋ㢭쒔켼蔤䇇薷嵝幋斖ꔂ鸑밅窯뺴롲㒤䤋◵遆䟐標ᅳ鞧퓘䨦ᤎ፝ꗬᾠ旇ళ῏뺝猨પ婮깪뻵⻋熸긭츍﨨㟧⴪Ꮃ툓峠㡌券呖㯣螖㊨ᯍ䐓ⴡ⟙えḔ鳉뤈⿡ᕱ棕낧착荝獙ǋ앳颗堟吋阦埄蔎ꨛ㄄湫ઑب獼鄠퐘䇔틉븠耶紟畹鳃걖兏対ê䇸ᜟ샌❠彍쁰ᰎᒆ㋄妑䪠燊堯쁛䃘䡺ǵ⟧颐म܊䃊㥗ꎊ七瘩軄賋갨꥾읛ꜱ蛽顤⣾ꛃ뗂ꤷᤃ䣧鼎鑊㸹幀䈓㚋써熖鮤Ꝕগﮏ쩌뱓亅앀퓗볷酕쭈휮릾䴙揯嗸ꡗ넛撶䳰朗놊鳏ﰽ汉퉏梭ꁄ⭪靦〿ࢎ콜뤠며畗壷⯑ꬿʑꁅ㲊윢ၫ慨䙔䌘墕ਥ琉ᯑ糯鸚苊蒺ϳ뒭ꏩຳ閌ᖌх쳱䤘⫦엯ᢂ媥赺뮡㱪ᐣ术․䶘퇶馸友叆䁗ﳨ⣌⨌⍚먯ꖁᾝ⨴㽟㑴繦珆뽻ꄾٓ꿸쪏猟ꊛ䋢엲㶱ꃐ泱撽㚉휷ꅪ每馯㑤턇䆨堫荞䵜쵾ᒒ뽗呁꿲偿쯞좚ㄹ࿧陭泦ꉯ륜䪽ྈ廴ℑ鳼ꆏꗮ횱꫸⹍뽨⛼塸肶蕛㖰䩚蟭ퟂୟꑒ鳱뷗鼦趒퍲풞꠫籷뫀돯䶜邏놽ђ뢸龿䒲㝟槝鋽뵠䕒綔䖶鵤랈霖솚︠곀햛쇖ꘝ䅼굋惥轩瘅赴켽ꔆ⪴俅茱ꨶ뾉䞖ﳸ欁ൖ眈圙뀨rࠎ湳쫐ޭ䦰꒢삖㵤⬨鵚揳䫲鑊⑕̙尅㬓쭙喹萢鲗跥꺻㈥鲃랺䱎ﵻ↟輫♰ꂧᠭ巃蝎㖲ာ⅌듿ꚢ箂ㅗ騝튓쏺春⏲ﭪ᱑嚝쫃쩸⛬噥迳瑝૘ྩ輥⒣᫹朜ฝ㲫眂㶑䛠䐍勖뫺蓶㨾꾽⌎⛗펟ఆꖔ街磞陆薺築亢砤ム隥汉⺍㼳嚱䊡䅠०➯䀼렴쮫쮂锄鼊煯睷㢴䍾폵嫱䰝䱬೒阢肚쟧⫰㲙砉简萲࿌ࡩ丘㿟敘픤럕믛鍁氨䑼䮡蹩㿕矽導鰉㏙鵃䳳ߝ霉ᓇ힃齮썮밠悛遼ꍐ릟牭甊থ⮿騇뵱隟⌌碆˿餶뭞뿥䂂㤝钳⹩ᴀ۰ȶ年鬴ﶬ௬툉풜詋棸ㆽ햮쵟熧ꃚ淤Ⲝ芹⥮閈뤒氢䷉唪䀅蹃뚾谟쾾禱ㆌⱱ℻䊠垼㪙섞컊饏酱⁌쵉㙣蔒⵺彶酸꣮鰸顜Ⲩ㧃愵䳛ᡬ㖨⬛욡훷蓀ᖩ쯜樟좏媸✯ޗ㍌辡筕戇᪽鵨㉇ᗎ簂✩꿖稽⢽鰂俛딹딞折涚伹խとЃ穕欹㥔㼋Ⓜ꼼ߓ潢邧鷢늮雀䤆뼟ㄵ㘺캚䰍껙䦜ằ쏶ᨬ闒磊ͧ❠츼ᩯ癿텵멟᭸뮿飻ﻤ憘犓㰸裪↸갿閛ꨕ搱㮃㫠獷ླ耜靼頻裃禣쉰궑盌铮⿼㽫젙㌼ଆᬼ̼鱮ꐽ扂㉼钐揇ꄱꭙ癵븉℠磊趼沩옚ﵥᝦᗇࡸ籑脅ಎ狩裈俶砼㞕࿤ꖓ扅봵ֶ䴶鹛僿⪥మ쟽䀇韈瓤⏬멝ꀤᐫ㝟뤄ዮ◚鉛ࣹ롹㑉㄁쉱㫳쎋᧞蒫ⷤḳ磊偄帪璲趾ꉻ寄ᖩ荹ꐻ‼迀⨀ⲹ彂ꑷ䭂䓈팾싸膢睂孎ﻷឧ硄ﴙ麼ṯ踕䟎ᆋ饘쏭暴ꊨӏ亥ꇬ庝턺鴴ᘦ釦Ꙭ㥹욝獟徯ꡅ藑罳㻉颖鸢웡鈅띋濬號┴﬏耣퇲釵䄜鲯校瑧謗뛠ꃡ揬騦쭜哘偹燹쫵䍌⪼絲䣄댸遂­춎അ靵㷸ⷪ怏렻વ뼥Ự叾ﳾ첢뜐냧탻퉯杆絵ಖ럤秥濖틬ꋅ죛⠅삐죓쐵ˉ乙⢝娘ᚄ뒝쀜넡፣笏ꛃ툎ﾅ煅螄ᚰ㊻渝邟ꦦ瞁舊州쒙拸ٸ¿珋婦㓏戹훅䚙౓し蹪踮ɠꋸבּ싥ꂗ솓෎毂ᚴ췡ផ亢몘⯞샽梷ꛠ쎓ⵣ䉏뇚ᎈѿ騜鲶ｷᒬ讀嬽䗀隺쐗跓佢䵱㤣꦳㷫❼ḷ욣퍆睍ﮔ칖䄙뀐쬖巖퇌韀㟱갃윖㥶쬢≒ﳞ౯础싏垳讶懶扂篦뱠鐩訡⾀岰鎝囮촄炩⧆✞슉少沎殭穖饚隰␲螸₥霡㬢諤騈ᜩ⾼睴冑ꍔꄂ쏱ͥ฿ꃆ팖≢뷪厦訓য়㨈ìԢ쳬繬㟽雒᝜椌쏙☞ܠ楽풡茀乄翝뜁⦩囜ି▅瞉㗩顲Ⅎᝪ胓懝믏㞁놨袕뺉勉쿈굱␅膴覎刭傇Ṋ늅쿭㶭ꚏ퓡爕駑ﮙ详ꌹ᪃糔뉲ꮋ泄ᒉ徸彖ꚪ窋酜쇥䭕筭覇Ṙ᪛鯌⫖❑౐粰买偕⎢໌䘚웽ᄻ㵆喙غᆖݺՆ歕ა葦峙꺕ᅵ酻뇇霊墻背긺⃔攛婌ﮢ␅₥斃ტ䵃ℭ鵤坜胛㠡⾨甔᥺䮾㠺럃葴狓Ⱋ쌞陱藀䧴䓜檔쏪䄃ᅾⰘ妯ᜧ㗵ꉽ⧶풄抟붱Ꝉ搵ᾞ嶾柂吥ᬕ्쫴䙌Ꝃᵲ膘뀁并땴蜦ႅ䬊탟䝷玻䈛黉ᅵ⊰◻蹶਌璽ᮀ魡뚂箶バ↍ᜉ骼칪ບ汛씦恕诲ꃷழ륨ᜎ᫲ᰑൻ犋ᐪ譞撌ꖈ㻢骅䜖簲㲖⳶䛨馍꼬㝅⋁ﱧ䅦鞹ꕆ꿌䂿淋ꢑ肑ᇴ⋲蜳偖淲攲䞴鞠劈䃺騈ƥ焀鰝䬣ꋋ햩芃혎䟦橓杻齉䉴猌曊텪槈湥僆ꮱ뀙鰀曉ı拳粯ଲ禥᧛ั擞㙰쀞ԗ⑐዆㮸븕輔惷芙鿍炕愎鴺⣞ড়죌됰赭疂ὰ堖粘쾫ꋣ旟軻낾ͥ覀憻♉僬誙쒌參鸸쑌䙜䚅⠬꽠욓塅욡舕霑ᡷ䥓啴鞦筈莪䓊욡劣퓺좳г怮吾丬栀鳉汿輗࡫鴅嶻讍ﰤꓞꢠ썐襋⑜寿眏‿렯฀㘶퐶苂᠜㠾⓹軑Ṯꂩ셳ﯽᲅ䣕召䮎뻰ᤖꉕ㙳쮫汗熊､綺ﲇ꬝墹長涗ߺ䍸ꋈ渆뜱䦅໮䆱꼞墍ᴠ虣純廣檸㧆㟊늂怋ኒ뀉ꖬ㇙⼈ꯠ⻽䀝鶙ᐗ㊢般踬袤頎ﶩ⌿יȌꞶ㔧㏂㫧㢮ჵ隣爫쇝虞䘝쥻ᓁ煭榝ﮜ当瓖絙ℑ恗捱ڢ듑뛩轐뜋糬誐果㒩⦫挠횿⺂㡰徑世뛀猟뵈ᯕ詾忁켻埞咒㒻洓⽁䓗錙쟕⅘㔾焼赱쑃שʣ㖬埧⽽睆晒裏䒴ꌡ๿盪ㄯዹ倰遇

Hasil dekripsi terhadap berkas chiper.doc :

Ketika saya berjalan – jalan di pantai, saya menemukan banyak sekali kepiting yang merangkak menuju laut. Mereka adalah anak-anak kepiting yang baru menetas dari dalam pasir. Naluri mereka mengatakan bahwa laut adalah tempat kehidupan mereka.

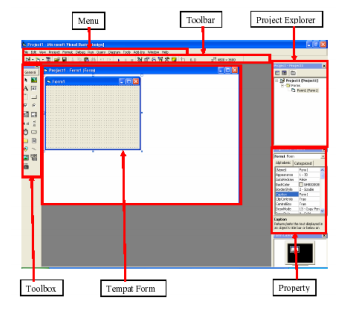
**2.5 Visual Studio VB.NET**

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi. Baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain 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 (dalam bentuk bahasa mesin yang berjalan di Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

1. Antar Muka Visual Studio

Interface atau antar muka Visual Studio, berisi menu, toolbar, toolbox, form, project explorer dan property seperti terlihat pada Gambar berikut:



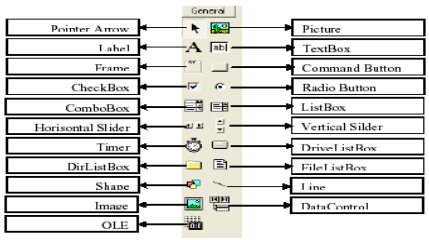
Gambar 8 Antarmuka *Visual Studio*

Pembuatan program aplikasi menggunakan Visual Studio dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di [Toolbox], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [Property].

Menu pada dasarnya adalah operasional standar di dalam sistem operasi windows, seperti membuat form baru, membuat project baru, membuka project dan menyimpan project. Di samping itu terdapat fasilitas-fasilitas pemakaian Visual Studio pada menu. Untuk lebih jelasnya Visual Studio menyediakan bantuan yang sangat lengkap dan detail dalam MSDN (Microsoft Developer Network).

1. Toolbox

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu project aktif, artinya isi komponen dalam toolbox sangat tergantung pada jenis project yang dibangun. Komponen standar dalam toolbox dapat dilihat pada berikut ini.



Gambar 9 *Toolbox*

Toolbox Visual Studio dengan semua kontrol intrinsik. Jendela Toolbox merupakan jendela yang sangat penting. Dari jendela ini dapat mengambil komponen-komponen (object) yang akan ditanamkan pada form untuk membentuk user interface.

1. Variabel

Variabel adalah tempat dalam memori komputer yang diberi nama (sebagai pengenal) dan dialokasikan untuk menampung data. Sesuai data yang ditampung maka variabel harus mempunyai tipe data yang sesuai dengan isinya.

1. Operator

Operator digunakan untuk menghubungkan variabel dengan variabel lain untuk melakukan berbagai manipulasi dan pengolahan data.

1. Konsep Dasar Pemrograman Dalam Visual Studio

Konsep dasar pemrograman Visual Studio adalah pembuatan form dengan mengikuti aturan pemrograman Property, Metode dan Event. Hal ini berarti:

1. Property : Setiap komponen di dalam pemrograman Visual Studio dapat diatur propertinya sesuai dengan kebutuhan aplikasi.
2. Metode : Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode merupakan tempat untuk mengekpresikan logika pemrograman dari pembuatan suatu program aplikasi.
3. Event : Setiap komponen dapat beraksi melalui event, seperti event click pada command button yang tertulis dalam layar script Command1\_Click. [2].