Basit: Algoritma Cipher Blok dengan Menggunakan Fungsi Hash Quark

Jauhar Arifin (13515049)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
jauhararifin10@gmail.com

Fadhil Imam Kurnia (13515146)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
fadhilimamk@gmail.com

Abstrak—Dalam makalah ini diajukan sebuah algoritma cipher blok (*block cipher*) baru menggunakan fungsi hash Quark yang ringan. Algoritma yang dinamakan Basit ini dapat mengenkripsi data dengan ukuran blok sebesar 64-bit. Keunikan algoritma Basit adalah masukan kunci yang ukurannya tidak ditentukan, pengguna dapat memasukan kunci yang mudah diingatnya. Algoritma Basit juga mudah diimplementasikan karena hanya menggunakan operasi-operasi sederhana, terdapat penggunaan fungsi hash Quark yang ringan dalam algoritma ini. Algoritma Basit juga dapat menghasilkan cipherteks yang sulit untuk dipecahkan dengan menggunakan analisis frekuensi, karena cipherteks yang dihasilkan memiliki kemunculan frekuensi yang relatif sama untuk setiap karakter. Oleh karena itu, Basit cocok digunakan untuk enkripsi data yang memerlukan keamanan tinggi.

Kata Kunci—Cipher Blok, Hash, Jaringan Fiestel, Quark

I. PENDAHULUAN

Salah satu aspek penting dalam komunikasi di era digital adalah masalah keamanan, terutama jika pesan yang dikomunikasikan merupakan pesan rahasia yang tidak boleh diketahui oleh pihak lain. Keamanan menjadi sangat penting karena pada umumnya komunikasi dilakukan melalui saluran publik yang digunakan oleh banyak orang. Bentuk serangan yang terjadi pada proses komunikasi dapat berupa serangan pasif atau serangan aktif. Pada serangan pasif, pihak ketiga berusaha mendapatkan sebanyak banyaknya informasi dengan melakukan penyadapan. Sedangkan pada serangan aktif, pihak penyerang berusaha mengintervensi komunikasi dan ikut mempengaruhi untuk kepentingan penyerang tersebut. Penyerang dapat mengintervensi dengan menghapus atau mengubah pesan, menyisipkan tambahan pesan, atau mengirim ulang pesan yang lama.

Teknik yang dapat digunakan untuk mengamankan pesan adalah dengan menggunakan kriptografi. Pesan yang dikirimkan dapat diubah terlebih dahulu menggunakan algoritma kriptografi tertentu sehingga hanya pengirim dan penerima saja yang dapat membaca pesan tersebut. Seiring dengan berkembangnya teknologi komputer digital, teknik kriptografi juga sudah dikembangkan untuk komunikasi dengan komputer. Kriptografi yang memanfaatkan komputer biasa disebut dengan kriptografi modern. Kriptografi modern memanfaatkan teori matematis dan aplikasi komputer, dengan pengoperasian dalam mode bit atau biner. Penggunaan kriptografi modern sudah sangat luas, teknik tersebut dapat ditemukan dalam sistem perbankan, koneksi internet aman (HTTPS), dan lain sebagianya.

Salah satu metode pada enkripsi digital adalah *block cipher*. *Block cipher* membagi bit-bit plainteks menjadi blok-blok bit dengan panjang yang sama, misalkan satu blok terdiri dari 64 bit. Proses enkripsi yang dilakukan menggunakan kombinasi operasi bit sederhana seperti permutasi dan substitusi, operasi tersebut juga dilakukan berulang-ulang dalam beberapa putaran. Beberapa contoh block cipher yang terkenal diantaranya AES, DES, Blowfish, dan lain sebagainya.

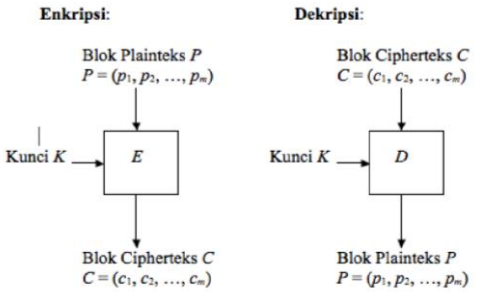
Pembuatan algoritma *block cipher* mengharuskan penentuan ukuran kunci yang akan digunakan untuk proses enkripsi dan dekripsi. Namun hal tersebut dapat membuat pengguna merasa kesulitan untuk menentukan kunci yang mudah diingat dan aman untuk digunakan. Oleh karena itu kami berinisiatif untuk menggunakan fungsi hash pada pemrosesan kunci yang diberikan oleh pengguna. Penggunaan fungsi hash tersebut memungkinkan pengguna untuk menggunakan kunci yang mudah diingat tanpa mengurangi aspek keamanan. Agar proses enkripsi yang dilakukan tetap ringan, kami menggunakan fungsi hash Quark yang dipublikasikan oleh Jean-Philippe Aumasson, Luca Henzen, Willi Meier, dan Mar´ıa Naya-Plasencia pada tahun 2012[1].

Pada makalah ini pembahasan dasar teori akan dijelaskan pada bagian II, kemudian akan dibahas mengenai rancangan Basit pada bagian III. Pada bagian IV akan dijelaskan hasil percobaan dan analisis dari algoritma *block cipher* Basit yang sudah diimplementasikan. Terakhir pada bagian V akan disimpulkan hasil penelitian yang kami lakukan.

II. DASAR TEORI

A. Cipher Blok

Cipher blok merupakan algoritma kriptografi yang beroperasi dengan memroses data dalam satuan blok yang sudah ditentukan. Setiap blok yang diproses dapat berisi sejumlah bit pesan atau byte pesan yang panjangnya sudah ditentukan oleh pembuat cipher blok. Hasil pemrosesan cipher blok adalah cipherteks dengan ukuran blok yang sama seperti panjang blok masukan yang diberikan, atau dapat dikatakan bahwa panjang blok cipherteks sama dengan panjang blok plainteks. Skema enkripsi dan dekripsi pada *blok cipher* dapat dilihat pada Gambar 1.



Gambar 1. Skema enkripsi dan dekripsi pada *block cipher*

Pada cipher blok dikenal lima mode operasi, yaitu Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), dan mode counter.

1. Electronic Code Book (ECB)
   Dengan mode ECB, setiap blok dienkripsi dan didekripsi secara independen satu persatu. Setiap blok menjadi input dari suatu fungsi yang akan menghasilkan cipher. Hasil enkripsi untuk suatu blok tidak akan mempengaruhi proses enkripsi blok yang lain. Karena sifat pemrosesannya yang independen, mode ini akan menghasilkan cipher yang sama untuk masukan blok yang sama. Hal tersebut dapat mempermudah pemecahan pesaan menggunakan metode kriptanalis, terutama dengen mendeteksi kemunculan cipher yang sama.

2. Cipher Block Chaining (CBC)
   Menggunakan mode CBC, setiap blok akan dienkripsi dengan memanfaatkan hasil enkripsi (cipher) blok sebelumnya. Cipher yang dihasilkan pada proses enkripsi akan di-XORkan dengan blok selanjutnya, kemudian hasil operasi tersebut akan dienkripsi. Metode ini sering digunakan karena blok yang sama tidak akan dienkripsi menjadi cipher yang sama, sehingga proses kriptanalisis dapat menjadi lebih sulit. Namun jika terdapat kesalahan satu bit saja pada suatu proses enkripsi, maka kesalahan tersebut akan merambat ke pemrosesan blok-blok selanjutnya.

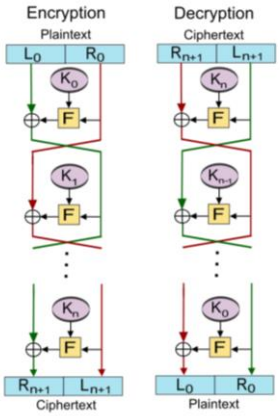
3. Cipher Feedback (CFB)
   Mode ECB dan CBC harus dilakukan jika jumlah bit dalam satu blok sudah lengkap, sehingga proses akan menunggu terlebih dahulu hingga jumlah bit lengkap. Mode CFB berusaha mengatasi kelemahan tersebut dengan melakukan proses dalam unit yang lebih kecil daripada ukuran blok. Ukuran data yang diproses dapat berupa bit per bit, 2 bit, 3 bit, dan sebagainya. Penggunaan mode CFB ini memerlukan struktur antrian (*queue*) yang berukuran sama dengan ukuran blok masukan.

4. Output Feedback (OFB)
   Pada mode OFB, operasi yang digunakan mirip dengan operasi pada mode CFB, namun n-bit dari hasil enkripsi terhadap antrian disalin menjadi elemen posisi paling kanan di antrian.

5. Mode *counter*
   Mode *counter* diusulkan oleh Diffie dan Hellman pada tahun 1979. Pada mode ini tidak dilakukan proses perantaian (*chaining*) seperti pada mode CBC. Untuk melakukan proses enkripsi, digunakan *counter* berupa blok bit yang ukurannya sama dengan ukuran blok plainteks. Nilai awal *counter* tersebut harus berbeda dari pada setiap blok yang dienkripsi, kemudian nilai *counter* tersebut dinaikan nilainya satu persatu pada tiap proses enkripsi.

B. Jaringan Fiestel (*Fiestel Network*)

Jaringan Fiestel merupakan struktur simetris yang digunakan dalam konstruksi Block Cipher. Jaringan ini ditemukan oleh kriptografer Horst Feistel selama pelaksanaan penelitiannya di IBM. Jaringan Fiestel bersifat *reversible*, karena operasi untuk melakukan proses enkripsi dan dekripsi sama, sehingga tidak perlu membuat algoritma baru untuk mendekripsi cipherteks menjadi plainteks. Ilustrasi operasi yang dilakukan pada diagram Fiestel dapat dilihat pada Gambar 2.



Gambar 2. Ilustrasi operasi pada jaringan Fiestel
(sumber: https://en.wikipedia.org/wiki/Feistel_cipher)

Jaringan Feistel terbentuk dari sejumlah putaran yang terdiri dari opweasi-operasi berulang, seperti permutasi, substitusi, dan operasi aljabar menggunakan XOR. Fungsi yang digunakan pada setiap putaran ini disebut *round function*.

C. Properti Konfusi dan Difusi

Konfusi (*confusion*) dan difusi (*diffusion*) merupakan dua properti dalam kriptografi yang menjamin keamanan cipherteks dengan menyulitkan proses analisis statistik. Kedua prinsip ini diperkenalkan oleh Claude Shannon pada tahun 1949 dalam makalah yang dipublikasikannya.

Prinsip konfusi bekerja dengan menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dan kunci. Hal tersebut dapat direalisasikan dengan menggunakan algoritma substitusi yang kompleks.

Sedangkan prinsip difusi menyebarkan pengaruh satu bit

plainteks atau kuncu ke sebanyak mungkin cipherteks. Misalnya jika perubahan dilakukan pada satu bit plainteks maka akan ada banyak bit pada cipherteks yang juga berubah, begitu pula sebaliknya. Kedua prinsip ini merupakan panduan dalam merancang berbagai algoritma kriptografi, dan juga menjadi konsep yang penting dalam merancang fungsi *hash* dan *pseudorandom generator*.

D. Fungsi Hash Quark

Fungsi hash adalah fungsi yang menghasilkan data keluaran dengan panjang yang selalu sama dari data yang panjangnya sembarang. Hasil keluaran fungsi hash disebut *message digest*. Fungsi hash bersifat satu arah karena kita tidak bisa mengembalikan *message digest* ke data atau string awal. Perubahan sedikit saja dalam data dapat mengakibatkan nilai hash yang berubah drastis. Dalam kriptografi, dikenal beberapa fungsi hash antara lain MD5, SHA-1, dan lain sebagainya.

Salah satu fungsi hash yang ringan adalah Quark. Fungsi hash tersebut membutuhkan kemampuan komputasi yang ringan[1]. Penggunaan fungsi hash Quark dapat diterapkan pada penggunaan teknologi RFID, NFC, dan *smartcard* yang membutuhkan algoritma hash yang cepat dan ringan.

III. RANCANGAN ALGORITMA

Algoritma *block cipher* Basit merupakan modifikasi algoritma *block cipher* yang memanfaatkan jaringan Fiestel. Besar blok pesan yang diterima adalah 64-bit atau 8 karakter, dan ukuran kuncinya dibebaskan. Panjang kunci dibebaskan karena ada tahap pemrosesan yang menggunakan fungsi hash, sehingga kunci yang digunakan akan selalui menghasilkan kunci baru yang panjangnya sama. Penggunaan fungsi hash juga membuat kunci mudah diingat tanpa mengurangi kekuatan algoritma *block cipher* Basit. Terdapat 14 (empat belas) putaran yang dilakukan pada *block cipher* ini. Secara garis besar, algoritma ini terdiri dari dua bagian utama yaitu tahap pembangkitan kunci internal dan tahap pemrosesan *round function* pada jaringan Fiestel. Proses dalam round function juga menggunakan S-Box serta P-Box yang sudah didefinisikan. Algoritma ini dapat berjalan relatif cepat karena menggunakan operasi bitwise sederhana seperti shift right, operasi XOR, substitusi, dan permutasi.

A. Pembangkitan Kunci Internal (*subkey*)

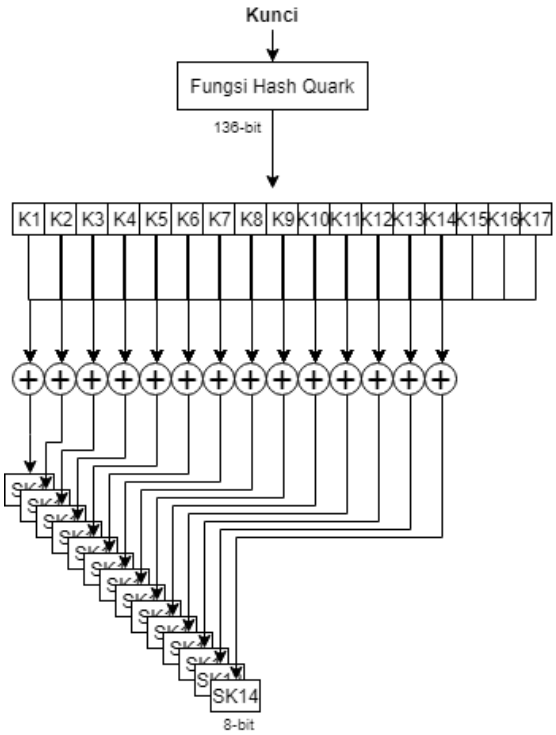
Setiap iterasi pada Basit memerlukan kunci yang berbeda-beda, maka dari kunci yang diberikan oleh pengguna perlu dihasilkan 14 kunci internal (*subkey*) yang akan digunakan pada masing-masing putaran. Langkah pembangkitan *subkey* adalah sebagai berikut:

1. Kunci yang diberikan oleh pengguna dimasukan dalam fungsi hash quark yang relatif ringan. Fungsi hash tersebut akan menghasilkan 136-bit kunci baru.
2. Kemudian dari 136-bit kunci baru tersebut akan dibagi-bagi menjadi 17 bagian yang masing-masing sebesar 8-bit.

   Misalkan bagian kunci baru ke-n kita sebut dengan notasi Kn, dan subkey ke-n yang dihasilkan pada pembangkitan kunci kita sebut dengan notasi SKn.

3. Dari 17 bagian yang ada kemudian akan dibentuk 14 subkey dengan menggunakan operasi XOR pada 3 bagian yang bersebelahan. Hasil XOR K1, K2, dan K3 akan menjadi SK1, lalu hasil XOR K2, K3, dan K4, akan menjadi SK2, begitu seterusnya hingga dihasilkan SK1 hingga SK14 yang akan digunakan pada fungsi putaran.

Proses pembangkitan kunci internal ini dapat dilihat pada Gambar 3.

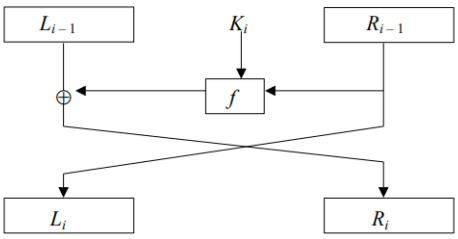


Gambar 3. Proses pembangkitan kunci pada Basit

B. Fungsi Putaran (*round function*)

Fungsi putaran *f* merupakan bagian dalam jaringan Fiestel yang digunakan pada Basit. Jaringan Fiestel yang digunakan pada algoritma ini membagi blok plainteks menjadi dua bagian sama besar yaitu 64-bit pada bagian kanan, dan 64-bit pada bagian kiri. Lalu 64-bit plainteks bagian kanan akan digunakan sebagai input dari fungsi putaran beserta *subkey* yang dihasilkan pada tahap pembangkitan kunci internal. Kemudian hasil fungsi putaran tersebut di-XOR-kan dengan 64-bit plainteks bagian kiri untuk kemudian menjadi plainteks bagian kanan pada putaran selanjutnya. Plainteks bagian kiri pada

putaran selanjutnya diambil dari plainteks bagian kanan putaran sebelumnya. Proses tersebut dapat dilihat pada Gambar 4. Terdapat 14 putaran yang akan dilakukan hingga proses enkripsi sebuah blok selesai dilakukan.



Gambar 4. Penggunaan jaringan Fiestel pada Basit

Fungsi putaran (*round function*) *f* yang digunakan pada Basit memanfaatkan beberapa operasi bitwise ringan seperti *shift right*, operasi XOR, substitusi, dan permutasi. Operasi substitusi dan permutasi yang dilakukan memanfaatkan S-Box dan P-Box yang sudah didefinisikan sedemikian rupa sehingga properti difusi dapat diperoleh. Urutan proses pada fungsi putaran dalam Basit adalah sebagai berikut:

1. Lakukan operasi XOR terhadap *subkey* dengan 32-bit plainteks bagian kanan yang dimasukan ke dalam fungsi putaran.
2. Lakukan proses substitusi pada hasil yang didapatkan dari operasi XOR sebelumnya. S-Box yang digunakan adalah sebagai berikut:

d9 9f c4 6d a8 d2 67 66 51 1e ba d5 b2 a1 95 10
f4 b0 4a 25 a9 ca 4 15 c0 9a 70 2a 49 1d 92 b6
21 87 f6 c6 b9 ae 83 5f 1a 1f 41 a6 27 5d 6a 3b
1b 5c b8 eb b1 b4 4e 60 2 55 a0 62 76 d8 d1 94
b3 ad f5 77 79 ee 9e 4d 7c f9 90 34 8a 7a 8e 63
c9 89 99 b 1c 73 8d f1 16 6e d7 43 72 17 bb 9b
4c f8 39 2f 20 22 80 ce 18 45 3f 3e 81 e6 e5 56
82 dc bc d4 7e cd 65 38 5 ab da 6 f2 30 1 36 df
54 bf 78 ef b5 e2 c1 0 86 98 24 96 be fb e0 ff 7
12 3c 50 61 85 6f 3d a3 cb a cf ea db 5b e9 e7
f7 fe c8 c7 40 84 6b d0 97 3 42 2e 31 2b e8 29
2d ac ec 13 74 19 8f 8b a2 37 7f f 58 9c a5 cc
75 88 4b e b7 46 f0 bd c 7d 93 8c e1 23 5e 44
d3 4f a4 47 a7 11 e3 6c 26 aa af 9 52 c3 68 ed fc
c2 c5 14 de 59 fa d 71 35 9d 2c 57 8 dd 69 5a f3
e4 fd d6 48 64 53 7b 33 28 3a

Tabel 1. S-Box pada Basit

3. Lakukan proses cyclic shift right pada 32-bit yang diperoleh dari hasil substitusi dengan aturan sebagi berikut.
   a. Bagi 32-bit hasil proses sebelumnya menjadi 4 bagian, masing-masing bagian memiliki panjang 8-bit.

      Misalkan bagian ke-n disebut dengan menggunakan notasi Pn.

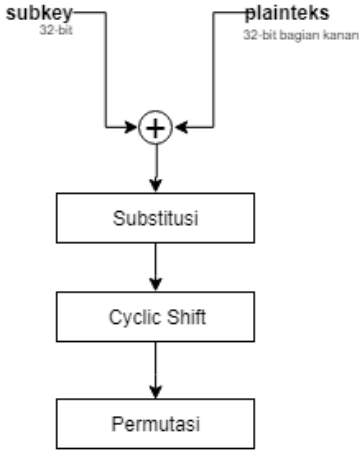
   b. Untuk setiap Pn, lakukan *shift right* n (>>n), sehingga

      P1 ← P1 >> 1
      P2 ← P2 >> 2
      P3 ← P3 >> 3
      P4 ← P4 >> 4

   c. Gabungkan 4 bagian 8-bit tersebut menjadi 32-bit kembali (P1||P2||P3||P4).

4. Lakukan proses permutasi menggunakan P-Box yang terdapat pada lampiran.

Proses fungsi putaran tersebut dapat dilihat pada Gambar 5.



Gambar 5. Proses fungsi putaran pada Basit

IV. Percobaan Eksekusi AlgoRitma

Untuk melakukan percobaan terhadap algoritma Basit yang sudah dirancang, kami mengimplemantasikannya menggunakan bahasa C[5] dengan tambahan program hash Quark yang telah dirancang sebelumnya[1]. Proses pengujian dilakukan dengan mode ECB, CBC, CFB, OFB, dan counter. Dalam percobaan ini digunakan kunci sebesar 8 bytes dan plainteks sebesar 44 bytes. Kunci dan plainteks yang digunakan dapat dilihat pada Tabel 2.

Kunci	67 61 6e 65 73 68 61 0a	ganesha
Plainteks	5468 6520 7175 6963 6b20 6272 6f77 6e20 666f 7820 6a75 6d70 7320 6f76 6572 2074 6865 206c 617a 7920 646f 670a	The quick brown fox jumps over the lazy dog

Tabel 2. Kunci dan plainteks yang digunakan untuk percobaan

A. Hasil dengan mode ECB

Cipherteks	0277 c2e5 ec5a 5bcb e284 7ff7 8b75 9469 b4b0 2d4b 8539 2046 144b c8be 8982 5448 b914 c4a7 358f 49bd 4a76 2818 8ce8 86bd
Hasil dekripsi	5468 6520 7175 6963 6b20 6272 6f77 6e20 666f 7820 6a75 6d70 7320 6f76 6572 2074 6865 206c 617a 7920 646f 670a

Tabel 3. Hasil percobaan menggunakan mode ECB

B. Hasil dengan mode CBC

Cipherteks	6188 9952 7350 1f43 92c4 8d53 e23e cc76 5ac3 fcc3 87b7 6e20 d047 4e7b 9cb7 5d6a 9366 c87c 9047 097b 2660 0ef3 3aee 1068
Hasil dekripsi	5468 6520 7175 6963 6b20 6272 6f77 6e20 666f 7820 6a75 6d70 7320 6f76 6572 2074 6865 206c 617a 7920 646f 670a

Tabel 4. Hasil percobaan menggunakan mode CCB

C. Hasil dengan mode CFB

Cipherteks	83d0 4822 8888 b02a 3e20 a863 e201 76fa 5516 80fb 8b01 29b8 9d86 8edb 2d9e 7951 dfa3 f41f 1900 96fc 44c2 906c 93eb 0880
Hasil dekripsi	5468 6520 7175 6963 6b20 6272 6f77 6e20 666f 7820 6a75 6d70 7320 6f76 6572 2074 6865 206c 617a 7920 646f 670a

Tabel 5. Hasil percobaan menggunakan mode CFB

D. Hasil dengan mode OFB

Cipherteks	83d0 4822 8888 b02a ce4b ae64 4190 78f6 6ff2 2510 512b 6afa fc98 9998 eb2a aa61 1cdb c647 c948 ff91 9ac9 10b5 8b5b c2dd
Hasil dekripsi	5468 6520 7175 6963 6b20 6272 6f77 6e20 666f 7820 6a75 6d70 7320 6f76 6572 2074 6865 206c 617a 7920 646f 670a

Tabel 6. Hasil percobaan menggunakan mode OFB

E. Hasil dengan mode *counter*

Cipherteks	83d0 4822 8888 b02a b10a fdf7 6f25 0faa eb6a 443d 9068 7a2f fbb9 db43 aa14 ce1b b57d c701 99b8 9c5e 955d f8ce a3a4 0f2e
Hasil dekripsi	5468 6520 7175 6963 6b20 6272 6f77 6e20 666f 7820 6a75 6d70 7320 6f76 6572 2074 6865 206c 617a 7920 646f 670a

Tabel 7. Hasil percobaan menggunakan mode counter

Selain mencoba eksekusi program dengan berbagai mode, kami juga mencoba menggunakan plainteks kecil dan plainteks besar untuk diproses menggunakan Basit. Plainteks kecil yang digunakan berukuran 13 bytes, sedangkan plainteks besar yang digunakan adalah gambar berukuran 9990 bytes. Plainteks gambar yang digunakan dapat dilihat pada Lampiran 2. Percobaan menggunakan dua plainteks tersebut berhasil di lakukan dengan baik.

V. Analisis Keamanan dan Waktu Eksekusi

Setelah memastikan program yang diimplementasikan dapat digunakan dengan benar, kemudian dilakukan berbagai analisis yang terkait dengan keamanan dan waktu eksekusi dari algoritma blok cipher Basit.

A. Analisis *Brute Force Attack*

Karena pemrosesan kunci menggunakan fungsi hash Quark yang relatif baru, oleh karena itu belum ada kolisi yang ditemukan. Fungsi hash tersebut membuat blok cipher Basit dapat menerima kunci dengan panjang berapa saja, kemudian akan dihasilkan 136 bit *message digest* untuk kemudian diproses dalam pembangkitan sub-kunci. Untuk memecahkan kunci yang digunakan, diperlukan *brute force* untuk kunci dengan panjang beragam. Oleh karena itu proses brute force dapat memakan waktu yang sangat lama.

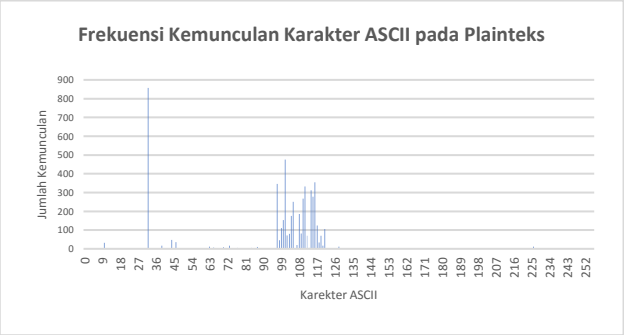
Brute force attack juga dapat dilakukan dengan memanfaatkan kolisi yang mungkin terjadi karena adanya penggunaan fungsi hash Quark. Berdasarkan *Birthday Problem*[6], peluang kolisi ditemukan ketika dilakukan *brute force* sekitar

$$1{,}25\sqrt{2^{136}} = 3{,}25 \times 10^{20}$$

kali. Misalkan sebuah komputer mampu menghasilkan 10 juta kunci / detik, maka untuk memecahkan kolisi tersebut dibutuhkan waktu hingga sekitar $1{,}03 \times 10^{6}$ tahun.

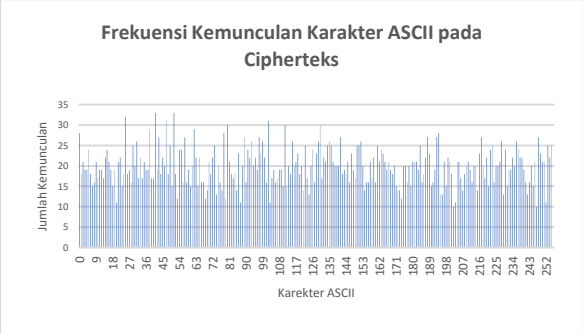
B. Analisis Frekuensi

Kami melakukan analisis terhadap frekuensi kemunculan pada plainteks dan cipherteks. Hasil analisis yang kami lakukan sangat memuaskan. Dari teks yang berisi cerita berbahasa inggris, terdapat beberapa karakter yang cukup sering muncul seperti huruf A, E, dan lain sebagainya. Plainteks yang digunakan untuk pengujian ini dapat dilihat pada Lampiran 3. Grafik kemunculan setiap karakter ASCII, dari 0 hingga 225, pada plainteks dapat dilihat pada Gambar 6.

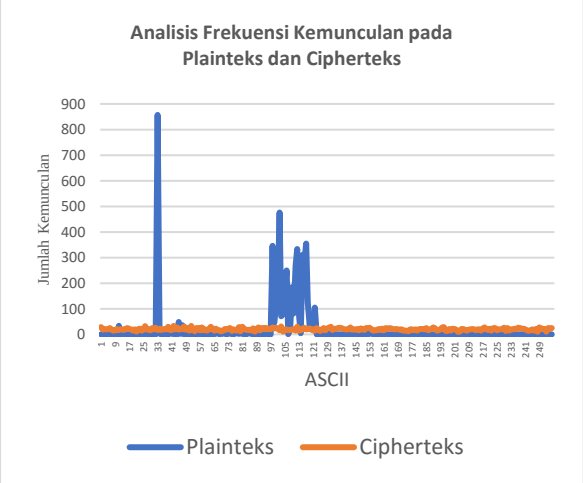


Gambar 6. Frekuensi kemunculan setiap karakter ASCII pada plainteks

Pada plainteks, karakter ASCII yang sering muncul diantaranya adalah spasi dengan nilai ASCII 32, dan huruf e dengan nilai ASCII 101. Setelah dienkripsi menggunakan blok cipher Basit, kami berhasil mendapatkan cipherteks dengan frekuensi yang relatif sama untuk setiap karakter ASCII. Frekuensi kemunculan tersebut sangat berbeda jika dibandingkan dengan plainteks awal. Frekuensi kemunculan karakter ASCII dalam cipherteks dapat dilihat pada Gambar 7, sedangkan perbandingan keduanya dapat dilihat pada Gambar 8. Dari hasil analisis ini dapat disimpulkan bahwa algoritma blok cipher Basit cukup aman, karena analisis frekuensi akan sulit dilakukan untuk memecahkan pesan asli.



Gambar 7. Frekuensi kemunculan setiap karakter ASCII pada cipherteks



Gambar 8. Hasil analisis frekuensi menggunakan algoritma *cipher block* Basit

C. Analisis Sedikit Perubahan Plainteks

Untuk memastikan sifat difusi dari Basit, kami melakukan percobaan dengan mengubah satu karakter pada plainteks, dan mengamati dampaknya. Plainteks dan cipherteks yang digunakan sebagai pembanding dapat dilihat pada Tabel 8. Cipherteks tersebut dihasilkan dengan menjalankan Basit menggunakan mode EBC.

Kemudian kami melakukan perubahan pada plainteks dengan mengubah 1 huruf. Plainteks yang sudah berubah tersebut kemudian dienkripsi menggunakan mode EBC dan hasilnya dapat dilihat pada Tabel 9. Menggunakan mode EBC tersebut hanya ada sedikit cipherteks yang berubah. Hal tersebut berbeda jika kita menggunakan mode CBC.

Kunci	ganesha
Plainteks	The quick brown fox jumps over the lazy dog
Cipherteks	0277 c2e5 ec5a 5bcb e284 7ff7 8b75 9469 b4b0 2d4b 8539 2046 144b c8be 8982 5448 b914 c4a7 358f 49bd 4a76 2818 8ce8 86bd

Tabel 8. Kunci dan plainteks yang digunakan untuk pembanding menggunakan mode EBC

Kunci	ganesha
Plainteks	The quick brown fo**k** jumps over the lazy dog
Cipherteks	0277 c2e5 ec5a 5bcb e284 7ff7 8b75 9469 **6718 c55e 3693 6476** 144b c8be 8982 5448 b914 c4a7 358f 49bd 4a76 2818 8ce8 86bd

Tabel 9. Cipherteks yang berubah setelah plainteks dirubah meggunakan mode EBC

Menggunakan mode CBC, perubahan 1 huruf pada plainteks dapat menghasilkan perubahan yang lebih banyak di cipherteksnya. Hasil percobaan menggunakan mode CBC ini dapat dilihat pada Tabel 10 dan Tabel 11.

Kunci	ganesha
Plainteks	The quick brown fox jumps over the lazy dog
Cipherteks	6188 9952 7350 1f43 92c4 8d53 e23e cc76 5ac3 fcc3 87b7 6e20 d047 4e7b 9cb7 5d6a 9366 c87c 9047 097b 2660 0ef3 3aee 1068

Tabel 10. Kunci dan plainteks yang digunakan untuk pembanding menggunakan mode CBC

Kunci	ganesha
Plainteks	The quick brown fo**k** jumps over the lazy dog
Cipherteks	6188 9952 7350 1f43 92c4 8d53 e23e **cc76** **d75b b823 c9ec 1dda 37de 8812 3432 ce21** **094d aa4b fd30 d686 a7da efd1 3a82 e358**

Tabel 11. Cipherteks yang berubah setelah plainteks dirubah meggunakan mode CBC

D. Analisis Sedikit Perubahan Kunci

Setelah mengamati dampak perubahan cipherteks karena perubahan plainteks, kemudian kami melakukan percobaan untuk mengamati dampak perubahan kunci terhadap cipherteks yang dihasilkan blok cipher Basit. Percobaan ini dilakukan menggunakan mode EBC. Hasil percobaan dapat dilihat pada Tabel 12 dan Tabel 13.

Dari percobaan ini didapatkan hasil yang cukup baik. Dapat dilihat bahwa sedikit perubahan pada kunci mengakibatkan perubahan pada seluruh cipherteks.

Kunci	ganesha
Plainteks	The quick brown fox jumps over the lazy dog
Cipherteks	0277 c2e5 ec5a 5bcb e284 7ff7 8b75 9469 b4b0 2d4b 8539 2046 144b c8be 8982 5448 b914 c4a7 358f 49bd 4a76 2818 8ce8 86bd

Tabel 12. Kunci dan plainteks yang digunakan untuk pembanding menggunakan mode CBC

Kunci	ganecha
Plainteks	The quick brown fok jumps over the lazy dog
Cipherteks	**4c2e adfe 5707 3db5 cfcf 0fae f2d8 4b8f 7b2a 4110 da4b 7eaa ba89 38de c046 0152 e933 17e0 5537 a281 796b 7019 baee 36f4**

Tabel 13. Cipherteks yang berubah setelah plainteks dirubah meggunakan mode CBC

E. Analisis Blok Berulang (*Repeated Block*)

Serangan terhadap kriptografi dapat dilakukan jika ada blok yang berulang pada cipherteks. Keberadaan blok berulang itu dapat dianalisis untuk memecahkan pesan yang dienkripsi. Kami melakukan percobaan dengan algoritma blok cipher Basit menggunakan berbagai mode. Kunci dan plainteks yang digunakan untuk percobaan dapat dilihat pada Tabel 13, sedangkan hasil dari percobaan tersebut dapat dilihat pada Tabel 14.

Kunci	ganesha
Plainteks	abcdefghabcdefghabcdefghabcdefgh

Tabel 13. Kunci dan plainteks yang digunakan untuk menguji blok berulang

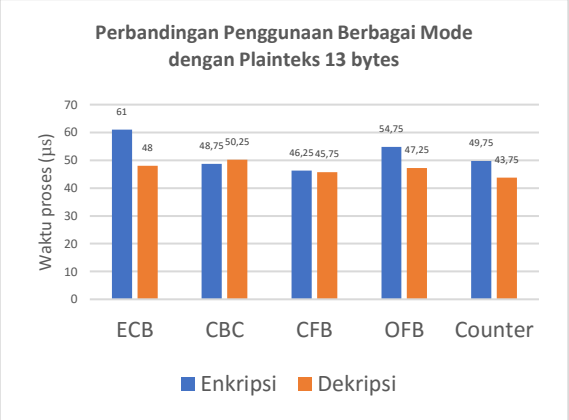
Mode EBC	**6c69 b1c3 9d67 5699 6c69 b1c3 9d67 5699 6c69 b1c3 9d67 5699 6c69 b1c3 9d67 5699** 94bc 848a 8aa2 8091
Mode CBC	4962 c907 f6b5 b450 82a9 f7bf fa88 e5ef c1c5 8d04 12a3 9b0b e7d3 68b4 887f 7316 281c f0f7 e513 bf47
Mode CFB	b6da 4e66 9c9b be21 525e 2f4e a75d f4ec 0f17 2ddf ae7e 1a81 9efc 77ce 363a 713a 7275 f0dd bcf6 1da8
Mode OFB	b6da 4e66 9c9b be21 c409 af72 4b81 71be 68ff 3e54 5e38 60e2 eeda 958a eb3e ed7d 7eb9 e12c af35 81b6
Mode counter	b6da 4e66 9c9b be21 bb48 fce1 6534 06e2 ec67 5f79 9f7b 7037 e9fb d751 aa00 8907 d71f e06a ffc5 e279

Tabel 14. Cipherteks yang berubah setelah plainteks dirubah meggunakan mode CBC

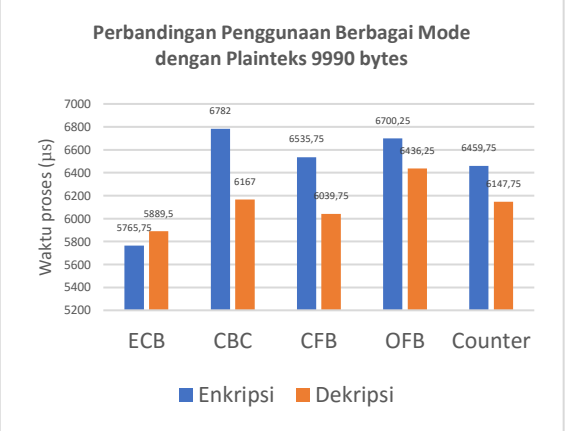
Dari percobaan yang dilakukan, menggunakan mode EBC terdapat blok berulang dalam cipherteks yang dihasilkan. Namun jika proses enkripsi dilakukan dengan mode lainnya, kami tidak menemukan blok yang berulang. Oleh karena itu untuk meningkatkan keamanan, maka disarankan untuk menggunakan mode selain EBC untuk mengenkripsi pesan.

F. Analisis Waktu Eksekusi dan Pemilihan Mode

Kami melakukan 2 kali percobaan, yaitu dengan data kecil (13 bytes) dan dengan data yang lebih besar (9990 bytes). Proses enkripsi pada data kecil membutuhkan waktu rata-rata selama 52,1 µs, sedangkan untuk dekripsi membutuhkan waktu selama 47 µs. Percobaan pada proses enkripsi data besar membutuhkan waktu 6448,7 µs, sedangkan untuk dekripsi membutuhkan waktu 6136,05 µs.



Gambar 9. Hasil percobaan penggunaan Basit pada data yang ukurannya kecil

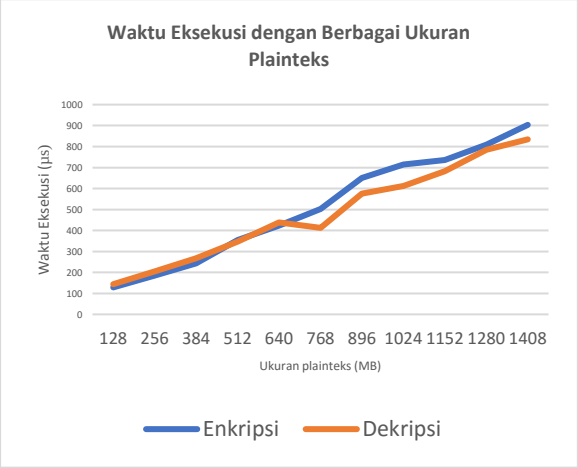


Gambar 10. Hasil percobaan penggunaan Basit pada data yang ukurannya besar

Dari percobaan tersebut, kenaikan jumlah data lebih dari 700 kali lipat tidak menaikan waktu eksekusi terlalu signifikan. Pemrosesan data kecil hanya memerlukan waktu 0,05 ms, sedangkan pada data besar memerlukan waktu 6.3 ms.

G. Analisis Waktu Eksekusi dan Ukuran Plainteks

Salah satu mode yang paling sering digunakan adalah mode CBC. Oleh karena itu percobaan ini dilakukan dengan menggunakan mode CBC. Percobaan ini dilakukan 4 kali untuk setiap ukuran plainteks yang sama. Kenaikan ukuran plainteks dilakukan secara linear. Hasil percobaan ini dapat dilihat pada Gambar 11.



Gambar 11. Hasil percobaan penggunaan Basit dengan ukuran plainteks yang berbeda-beda

Dari hasil percobaan ini diketahui bahwa semakin besar ukuran plainteks, maka waktu yang dibutuhkan untuk proses enkripsi dan dekripsi juga akan semakin besar. Untuk plainteks sebesar 1 MB hanya dibutuhkan waktu eksekusi dibawah 1 ms.

VI. Studi Lebih Lanjut

Penggunaan operasi-operasi dasar dalam blok cipher Basit membuat proses enkripsi dan dekripsi berjalan dengan ringan. Penggunaan fungsi hash Quark juga dimaksudkan agar blok cipher ini tetap ringan untuk diimplementasikan. Penggunaan operasi yang sederhana dan fungsi hash Quark membuat blok cipher ini ringan jika langsung diimplementasikan pada level perangkat keras. Kedepannya dapat dilakukan implementasi blok cipher ini langsung ke perangkat keras, sehingga dapat dilakukan penelitian lebih lanjut mengenai performa dari blok cipher ini jika digunakan pada perangkat dengan *resource* yang terbatas.

V. Kesimpulan

Algoritma *block cipher* Basit merupakan alternatif algoritma yang mudah untuk diimplementasi karena menggunakan operasi-operasi yang sederhana. Walaupun demikian, algoritma Basit dapat menghasilkan cipherteks yang sulit untuk dipecahkan menggunakan analisis frekuensi. Waktu yang digunakan untuk pemrosesan plainteks juga relatif cepat. Penggunaan fungsi hash Quark membuat algoritma ini dapat menerima berbagai kunci yang mudah diingat, tanpa ada ketentuan ukuran bit kunci.

Daftar Pustaka

[1] Jean-Philippe, Aumasson, Luca Henzen, Willi Meier, dan Mar´ıa Naya-Plasencia. 2012. *Quark: a lightweight hash*.

[2] Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi.* Yogyakarta: Penerbit Andi

[3] Munir, Rinaldi. 2015. Slide Kuliah IF4020 Kriptografi: Algoritma Kriptografi Modern

[4] Munir, Rinaldi. 2015. Slide Kuliah IF4020 Kriptografi: Serangan terhadap Kriptografi.

[5] Implementasi Basit: https://github.com/jauhararifin/basit

[6] Wegner, David. *A Generalized Birthday Problem.* University of California at Barkely.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 2 April 2018

Jauhar Arifin (13515049)

Fadhil Imam Kurnia (13515146)

Lampiran 1 : P-Box yang digunakan dalam Basit Block Cipher

39	2	3f	20	12	0	33	15	f	b	32	3b	3c	27	1e	2a	2e	38	7	2c	1b	36	1a	1d	e	10	3d	1	1c	25	14	26	13	21	3e	2d	2f	30	d	1f	17	35	23	4	19	34	3a	a	9	16	6	29	31	8	24	3	c	2b	22	11	28	37	5	18
13	2b	24	37	29	10	28	33	d	3c	17	1a	14	15	1f	6	b	30	20	1b	36	12	9	1d	31	21	3e	1c	3f	2a	7	1e	3b	32	c	0	38	4	16	26	8	25	3	e	f	35	2c	27	18	23	11	2e	2d	1	22	39	a	5	3d	19	3a	2	2f	34
31	29	23	1	18	c	2	3e	2a	3c	36	33	0	22	30	21	3d	2d	2b	3b	32	1b	1c	24	39	2e	10	17	1d	38	2f	11	1f	19	13	8	37	3a	14	25	3f	16	34	d	27	12	b	9	f	2c	7	6	a	3	1e	5	15	28	35	26	20	1a	4	e
31	9	14	1e	34	39	2	2e	2c	e	10	1f	18	11	3	1a	37	21	20	1d	3d	c	4	3e	30	d	3a	22	6	2d	5	1b	2f	27	24	12	f	a	2b	3c	17	3b	3f	2a	23	19	33	35	38	32	b	16	29	13	1c	15	7	8	28	26	1	36	25	0
36	d	10	1e	16	1d	35	14	29	1f	e	9	7	2d	2c	5	38	6	27	15	c	1c	17	21	32	2a	3a	3	26	3b	1b	8	39	30	25	28	20	2e	3c	24	31	13	23	12	b	34	2f	37	2b	4	11	22	1	3e	18	f	1a	0	3f	19	3d	a	33	2
15	29	1a	22	37	11	26	d	17	27	7	21	16	30	33	12	20	4	3a	2	f	39	31	e	2c	b	14	9	0	6	28	3b	2f	2a	24	a	3e	1	1f	3f	25	1e	1b	5	2e	3	10	3d	38	1d	36	8	2b	23	32	13	2d	35	34	18	19	3c	1c	c
33	17	2d	13	14	11	28	3f	8	24	31	34	2b	1	1d	1e	2	30	1f	2c	22	7	c	e	b	1a	26	3c	29	2e	23	5	36	3e	10	2a	3b	f	27	12	20	38	37	3	6	35	3d	0	15	3a	39	2f	4	19	16	9	1c	d	32	21	18	25	a	1b
2e	2f	33	20	1a	10	3b	35	2b	39	1f	30	28	1b	29	3d	e	1	13	34	18	3c	38	37	1d	23	d	32	9	31	12	b	2c	2	2d	14	6	4	8	3f	21	1c	1e	24	19	26	5	0	3e	27	15	16	7	36	11	f	3a	a	25	22	17	3	c	2a
36	13	34	32	3e	27	20	3b	19	28	c	2	33	22	2c	2a	3f	1	9	d	2f	7	8	1a	38	18	12	31	2b	14	37	1f	3	1e	3c	16	23	3a	b	5	11	0	26	a	39	2d	2e	25	1c	30	21	6	35	1d	f	4	3d	e	29	1b	10	17	24	15
1f	3f	35	1d	34	37	20	3	2d	24	28	30	3d	2b	39	c	13	32	23	1c	3b	e	17	36	4	5	18	19	3e	2e	16	21	29	2	0	3a	15	1b	1e	22	d	3c	11	38	6	25	2f	7	10	a	27	9	1	2c	1a	14	31	33	8	12	2a	b	f	26
14	16	15	2a	4	b	36	2e	13	2c	17	37	27	19	2f	22	c	1f	24	23	7	2d	31	3c	25	38	20	29	1e	2	35	3e	32	3f	2b	18	5	3	3d	1a	f	9	21	1c	3a	1b	10	6	34	e	26	8	1	28	d	39	3b	12	11	33	30	a	1d	0
d	20	39	2d	a	30	3b	2b	3f	6	22	2c	8	2e	37	4	36	18	13	7	21	1a	3	3d	2a	28	3a	c	14	10	12	19	1e	32	11	2f	24	35	1	2	1c	e	1b	27	26	38	0	f	29	31	16	17	9	1f	3c	3e	b	5	25	23	33	34	1d	15
e	22	4	13	1	2a	f	2f	3a	31	7	39	36	2	23	1c	1f	15	19	26	1e	18	21	34	b	a	37	17	12	3b	8	5	14	6	0	2c	10	16	25	3d	1a	33	2e	2b	24	28	20	3	38	30	11	32	1d	29	1b	3e	35	3c	c	27	2d	3f	9	d
0	e	25	38	3f	33	3d	1f	13	28	2e	31	22	21	36	17	1e	18	c	1	26	2f	1b	a	34	f	3e	1c	12	19	3b	23	37	39	1d	2	2a	35	29	3	2c	10	b	9	27	d	32	7	30	3a	15	8	3c	2d	16	1a	20	2b	11	14	6	4	5	24

Lampiran 2 : Cipherteks besar berukuran 9990 bytes untuk percobaan

Lampiran 3 : Plainteks Bahasa Inggris untuk analisis frekuensi

As a child, I loved sitting on my grandfather's lap while he read me stories. I remember most of them even though I am now a grandparent, too! As a child, I was blissfully unaware that, as I listened to the stories, I was also learning new words and ways in which those new words combined to communicate ideas and life lessons.

A good story encourages us to turn the next page and read more. We want to find out what happens next and what the main characters do and what they say to each other. We may feel excited, sad, afraid, angry or really happy. This is because the experience of reading or listening to a story is much more likely to make us 'feel' that we are part of the story, too. Just like in our 'real' lives, we might love or hate different characters in the story. Perhaps we recognise ourselves or others in some of them. Perhaps we have similar problems.

Because of this natural empathy with the characters, our brains process the reading of stories differently from the way we read factual information. Our brains don't always recognise the difference between an imagined situation and a real one so the characters become 'alive' to us. What they say and do is therefore more meaningful. This is why the words and structures that relate a story's events, descriptions and conversations are processed in this deeper way.

In fact, cultures all around the world have always used storytelling to pass knowledge from one generation to another. Our ancestors understood very well that this was the best way to make sure our histories and information about how to relate to others and to our world was not only understood, but remembered too. (Notice that the word ‘history’ contains the word ‘story’ – More accurately, the word ‘story’ derives from ‘history’.)

Encouraging your child to read or listen to stories should therefore help them to learn a second language in a way that is not only fun, but memorable.

Let's take a quick look at learning vocabulary within a factual text or within a story. Imagine the readers are eight-year-olds interested in animals. In your opinion, are they more likely to remember AND want to continue reading the first or second text?

Many birds and animals live in the world, for example, parrots, pandas, lions, leopards and rabbits. In the sea we can find whales, dolphins, sharks and octopuses.

My younger brother is called Fred. Fred's very interested in animals. He talks and asks questions about animals ALL the time! Fred's really interested in parrots and pandas and lions and leopards and rabbits. But Fred's favourite animals live in the sea. He has pictures of whales, dolphins, sharks and octopuses on all the walls of his bedroom.

From: Do whales have stomach aches? (Storyfun for Movers, Cambridge University Press, 2011).

When choosing second language story books, you might consider questions like:

Will your child easily identify with the central characters? Are they of similar ages for example?
Will the events interest and excite, scare or amuse your child enough to motivate them to continue reading?
Is the story an appropriate length – not too short, not too long?
Will the layout – the font, the titles, the amount of text on each page – appeal to your child?
Is it supported by illustrations that your young reader will enjoy looking at?
For your child to gain the maximum benefit and language learning from reading stories, consider the story's language level carefully, too. Is the grammar and vocabulary not too easy but still accessible to the reader? Would the language be similar to that which your child might use in their first language? Would it support school work and help prepare for tests? Useful EFL publications such as Storyfun for Starters, Movers and Flyers and other graded readers are carefully written with these important considerations in mind.

But, of course, stories don't only offer the young reader a chance to read. The experience also creates an opportunity to talk about the story. As a parent, you can encourage your child to describe their favourite person, part of the story or picture. Their creativity might be developed by drawing new story pictures or even by writing their own short stories as a result.

If your child is reluctant to read or has little confidence in their ability to read in another language, you might help them by reading the story to them, stopping where necessary to interact and ask questions like 'What do you think will happen next?' If you read to your children in a relaxed and fun way, they will subconsciously relate to the reading and language learning process more confidently and positively. Of course, being read to by a parent, for whatever reason, is also simply a lovely way to share quiet and close time.

The experience of reading or listening to a story allows us to escape our own lives for a moment and live in another one in a fun and safe way. In the same magical experience, a goldmine of language may be learned, so do encourage your child to read stories in their second language as well as their first!