BAB 2
LANDASAN TEORI

2.1 Profil Perusahaan

Profil perusahaan konsultan *Art and Design House The House* meliputi sejarah berdirinya, visi dan misi serta struktur organisasi.

2.1.1 Sejarah Perusahaan

House The House adalah sebuah konsultan seni dan desain terintegrasi yang menitikberatkan praktik seni dan desain untuk produk, brand, dan lingkungan terbangun dengan fokus pada kualitas pengalaman pengguna dan strategi yang berbasiskan pada kultur regenerasi ruang desa dan perkotaan.

Bertempat di Bandung, praktik desain yang perusahaan ini angkat sebagai keutamaan adalah Arsitektur, Lingkungan Terdesain, Desain Produk, Tampilan AV/Multimedia, Dokumenter Konseptual, Strategi Branding, dan Instalasi Intermedia.

House The House didirikan pada 24 Februari 2011 oleh 9 orang sahabat yaitu Meizan Diandra Nataadiningrat, Prananda Luffiansyah, Raditya Ardianto Taepur, Sakti Nuzan, Kukuh Rizal Arfianto, Yanuar Pratama Firdaus, Artiandi Akbar, dan Findhika Putra. Ke-9 orang ini memiliki basis keilmuan dan ketertarikan yang berbeda-beda di bidang arsitektur, desain interior, desain komunikasi visual, desain produk, kriya keramik, dan teknologi informasi.

Sebelumnya mereka tergabung dalam 3 media yang fokus pada keilmuan masing-masing, seperti Fold Magazine (http://foldmagazine.com) sebuah media online yang membahas desain produk dan kulturnya, Orders yang merupakan media cetak 3 bulanan berbicara tentang arsitektur dan perkotaan, dan Bandung Affairs sebuah platform digital dengan metode crowd-sourcing yang mengungkap seluk beluk Kota Bandung. Kemudian mereka memutuskan untuk menggabungkan ke tiga platform tersebut menjadi satu wadah yang memiliki keunggulan dalam

berkarya sekaligus memiliki keunikan dan pertanggungjawaban dalam interaksi sesama manusia dan lingkungan terbangun.

2.1.2 Visi dan Misi Perusahaan

Berikut merupakan visi dan misi dan perusahaan *House The House* :

a. Visi

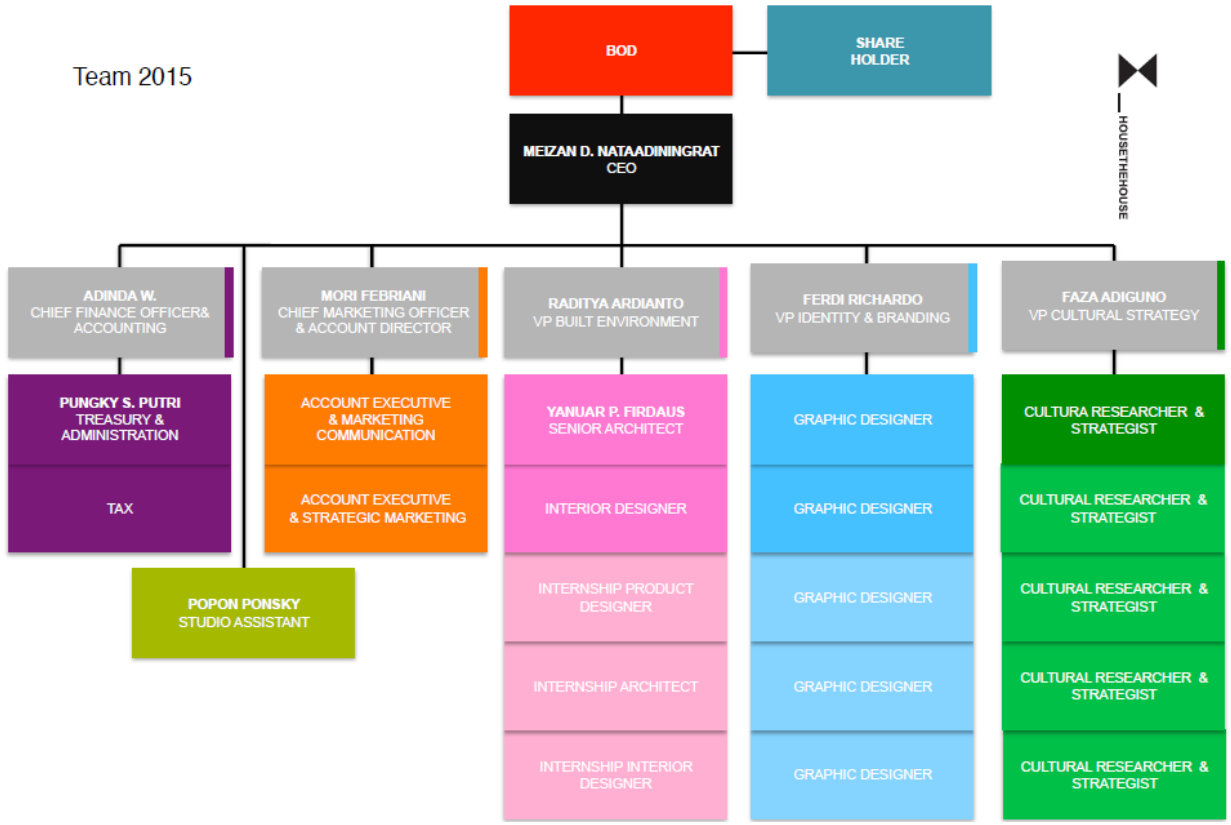
"House The House sebagai penyedia jasa creative collective desain yang bergerak dalam bidang regenerasi ruang kota dan daerah pedesaan terkemuka di Indonesia"

b. Misi

House The House menjadi pilihan utama sebagai penyedia jasa desain terintegrasi di Indonesia yang bekerjasama dengan *corporate*, memegang saham, dan mitra usaha lainnya untuk dapat menghasilkan nilai tambah bagi *stakeholder*, karyawan, dan negara.

2.1.3 Struktur Organisasi

Struktur Organisasi di *House The House* sebagai berikut :



Bagan 2.1 Struktur Organisasi House The House

2.2 Landasan Teori

Landasan Teori bertujuan memberikan gambaran dari teori yang terkait dalam pembangunan aplikasi. Landasan Teori yang dibahas yaitu Pengertian Citra Digital, Kriptografi, Metode yang digunakan dan Bahasa Pemrograman yang digunakan.

2.2.1 Citra Digital

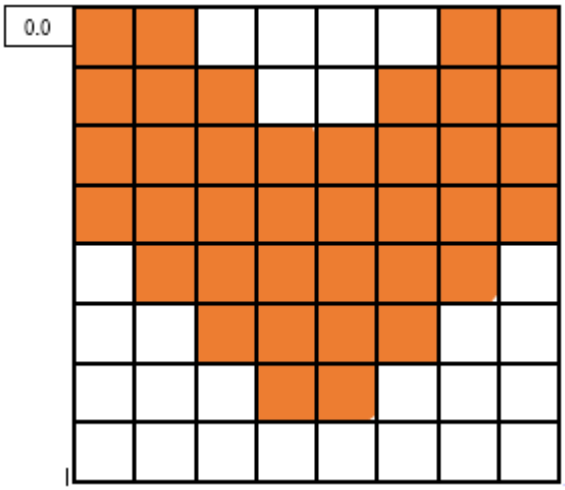
Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Sedangkan jika ditinjau dari sudut pandang matematis, citra merupakan fungsi terus menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata

pada manusia, kamera, pemindai (*scanner),* dan sebagainya, sehingga banyak objek yang disebut citra tersebut terekam. [7]

Sedangkan citra digital sendiri merupakan sebuah gambar yang dibentuk kumpulan titik-titik piksel [8]. Dalam piksel terdapat kumpulan angka biner yang merepresentasikan nilai dari setiap warna. Dalam komputer, setiap piksel diwakili oleh dua buah bilangan bulat (integer) untuk menunjukkan lokasinya dalam bidang citra (koordinat (0,0)). Piksel disusun dalam sebuah larik dua dimensi (*array* dua dimensi) [9].

Hubungan antara koordinat citra dan indeks larik untuk citra yang bersangkutan dapat ditunjukkan dengan dalam bentuk diagram berupa kumpulan segi-empat dengan sisi-sisi yang sama (*square tesselation*). *Square Tessalation* hanya sebuah teknik visualisasi dari citra yang tersimpan dalam memori komputer sebab kenyataannya di dalam komputer memori yang berisi data intensitas citra tidak terletak berdampingan seperti pada teknik *tesselation.*

Berikut merupakan gambaran teknik *tessalation* pada suatu objek citra digital pada larik dua dimensi dengan ukuran array 8x8 :



Bagan 2.2 Penggambaran Citra pada Square Tesselation

Untuk mengambil nilai dari suatu citra digital sangat beragam caranya. Hal ini sangat bergantung pada bahasa pemrograman dan *tools* tambahan dari setiap programnya. Contoh aplikasi dan *tools* untuk mengambil nilai dari suatu citra adalah octave pada matlab, fungsi bawaan dari C++ dan fungsi bawaan dari C# menggunakan visual studio.

File citra beragam, ada yang memiliki header dan ada yang tidak memiliki header namun secara umum suatu file citra terdiri atas header dan palet warna. Sebagai contoh kita ambil struktur penyusun file .bmp yaitu sebagai berikut :

Tabel 2.1 Struktur penyusun bitmap

Header berkas	Header bitmap	Informasi palet	Data bitmap

Secara umum header berisi tentang cara membaca suatu file citra. Sedangkan untuk informasi palet akan menyimpan nilai warna dalam bentuk RGB. Dan untuk data bitmap menyimpan data bitmap yang digambarkan secara susunan terbalik yaitu baris ke-0 pada matriks menyatakan data pixel citra baris terbawah dan sebaliknya untuk baris k-n matriks menyatakan data pixel teratas.

Citra digital sendiri memiliki tiga jenis penggambaran warna. Ketiga jenis tersebut adalah citra berwarna, citra berskala keabuan dan citra biner. Berikut merupakan penjelasan dari setiap jenis tersebut :

a. Citra berwarna

Citra berwarna atau biasa dinamakan citra RGB merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau) dan B (biru). Setiap komponen menggunakan delapan bit (nilainya berkisar 0 sampai dengan 255). Dengan demikian, kemungkinan warna yang dapat disajikan mencapai 255 x 255 x 255 atau 16.581.375 warna. Sehingga warna dapat direpresentasikan sebagai berikut :

Tabel 2.2 Warna dan nilai penyusunnya

Warna	**R**	**G**	**B**
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255

Berikut merupakan contoh penggambaran citra RGB



Bagan 2.3 Contoh citra dengan warna RGB

b. Citra Berskala Keabuan

Citra jenis ini hanya menggambarkan suatu citra dengan dua warna yaitu hitam dan putih serta gradasinya. Pada jenis ini warna dinyatakan dengan intesitas yang berkisar antar 0 – 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih. Sehigga efeknya pada citra ini terdapat warna abu yang merepresentasikan warna antara 0 dan 255.

Berikut merupakan contoh citra berskala keabuan :



Bagan 2.4 Contoh citra berskala keabuan

c. Citra Biner

Citra biner adalah citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dia kemungkinan (yaitu 0 dan 1). Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Warna jenis ini banyak digunakan dalam pemrosesan citra karena sangat memungkinkan untuk dimanipulasi.

Berikut merupakan contoh citra berskala biner :



Bagan 2.5 Contoh citra biner

Dalam penelitian kali ini akan digunakan jenis citra RGB yang merupakan citra yang sering digunakan dalam kehidupan sehari-hari. Selain itu citra ini juga merupakan citra yang digunakan di tempat studi kasus *House The House.*

2.2.2 Kriptografi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan infromasi seperti kerahasiaan, keutuhan data dan otentifikasi. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan

penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi [2].

Secara umum kriptografi dapat dibedakan menjadi dua berdasarkan era nya, yaitu :

a. Kriptorafi klasik

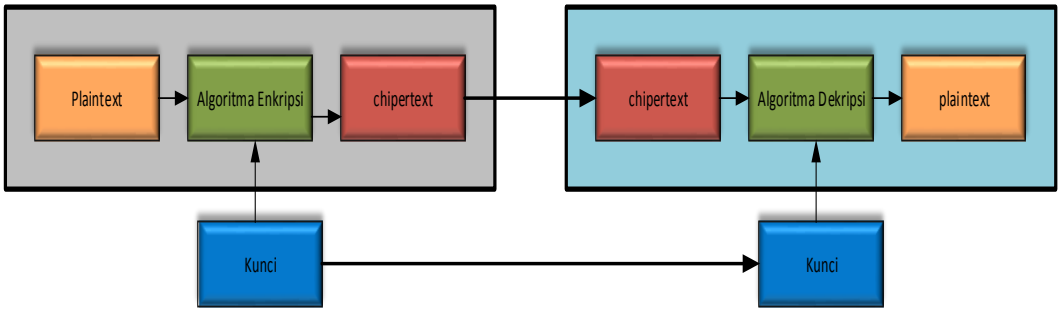
Kriptografi klasik merupakan kriptografi yang ada sebelum era komputer. Kriptografi pada era ini menggunakan kunci simetrik dengan menggunakan metode substitusi (pergantian huruf) dan/atau transposisi.

b. Kriptografi modern

Kriptografi modern merupakan kriptografi yang ada setelah era komputer. Pada kriptografi modern sistem penyandian lebih bervariasi. Hal ini karena era komputer memungkinkan manipulasi dilakukan dalam satuan byte dan bit. Pada kriptografi modern ini pula terdapat metode baru penyandian yaitu sistem blok dan stream.

Penyandian dengan sistem blok mengakumulasikan sekumpulan teks atau byte menjadi sebuah blok kemudiaan baru dilakukan penyandian. Sedangkan metode stream penyandian dilakukan secara realtime. Hal ini berarti bahwa setiap bit atau byte akan dilakukan penyandian tersendiri.

Secara umum sistem penyandian dapat digambarkan dengan sederhana yaitu seperti berikut :



Bagan 2.6 Sistem Kriptografi

Berdasarkan gambar tersebut terlihat beberapa istilah sebagai berikut :

a. Plaintext

Plaintext meruapakan teks asli yang dapat terbaca. *Plaintext* merupakan masukan bagi algoritma enkripsi.

b. Kunci

Kunci merupakan masukan bagi algoritma enkripsi juga selain *plaintext.* Kunci dapat dibedakan menjadi tiga yaitu :

1) Kunci simetrik

Kunci simetrik merupakan kunci yang proses enkripsi dan dekripsi menggunakan kunci bernilai sama. Penyandian dengan kunci simetrik harus benar-benar menjaga kunci tersebut karena jika kunci tersebut diketahui orang lain maka proses dekripsi dapat dilakukan dengan mudah.

2) Kunci asimetrik

Kunci asimetrik merupakan kunci yang proses enkripsi dan dekripsinya menggunakan kunci bernilai beda. Penyandian ini akan membangkitkan dua buah kunci dimana masing-masing kunci digunakan untuk proses enkripsi dan proses dekripsi. Pada kunci asimetrik ini biasa menggunakan persamaan matematika untuk membuat sebuah kunci sehingga pada akhirnya meskipun suatu *plaintext* di enkripsikan dengan nilai kunci yang berbeda namun bisa didekripsikan dengan kunci yang beda.

3) Fungsi Hash

Fungsi *hash* merupakan fungsi yang melakukan pemetaan pesan dengan panjang sembarang ke sebuah teks khusus yang disebut *message digest* dengan panjang tetap. Fungsi *hash* umumnya dipakai sebagai nilai uji (*check value)* pada mekanisme keutuhan data.

c. Algoritma enkripsi

Algoritma enkripsi memiliki dua masukan yaitu teks asli (*plaintext)* dan kunci rahasia. Algoritma ini melakukan transformasi terhadap *plaintext* dengan menggunakan kunci yang ada, algoritma ini akan menghasilkan teks tersandi (*chipertext).*

d. Algoritma dekripsi

Algoritma ini memiliki dua masukan yaitu *chipertext* dan kunci rahasia. Algoritma ini memulihkan kembali *chipretext* menjadi teks asli jika kunci yang digunakan untuk melakukan dekripsi bernilai benar.

e. *Chipertext*

Chipertext adalah keluaran dari algoritma enkripsi. *Chipertext* dapat dikatakan sebagai pesan dalam bentuk tersembunyi. Algoritma enkripsi yang baik akan menghasilkan *chipertext* yang terlihat acak.

2.2.3 Metode RSA – CRT

Pada tahun 1977, Rhivest, Shamir dan Adleman merumuskan algoritma praktis yang mengimplementasikan sistem kriptografi kunci publik disebut dengan sistem kriptografi RSA [10]. Meskipun pada tahun 197 badan sandi Inggris mempublikasikan bahwa Clifford Cock telah merumuskan sistem kriptografi RSA 3 tahun lebih dahulu daripada Rivest, Shamir dan Adleman.

Algoritma enkripsi dan dekripsi sistem kriptografi RSA bersandar pada asumsi fungsi satu arah (*one – way function*) yang dibangun oleh fungsi eksponensial modular pada grup perkalian (Z^*_n, x) dan grup perkalian $(Z^*_{\phi(n)}, x)$ dengan $n = p \times q$, p, q adalah bilangan prima dan $\phi(n) = (p - 1) \times (q - 1)$.

Sedangkan metode RSA – CRT (Rhivest – Shamir – Adleman dengan Chinese Remainder Theorem) merupakan suatu metode kriptografi yang sama dengan RSA biasa namun memanfaatkan teorem CRT untuk memperpendek ukuran bit eksponen dekripsi *d* dengan cara menyembunyikan *d* pada sistem kongruen sehingga mempercepat waktu dekripsi. Sistem kongruen pada CRT adalah dengan pemanfaatan penyelesaian masalah kongruen dengan modulus berbeda. Berikut penggambaran sistem kongruen pada CRT :

$x \equiv a_1 \mod m_1$

$x \equiv a_2 \mod m_2$

...

$x \equiv a_k \mod m_k$

Dengan nilai $a_1, \ldots, a_k$ dan $m_1, \ldots, m_k$ diketahui dan x tidak diketahui dan nilai x diketahui. Untuk mencari nilai x dapat menggunakan prosedur berikut :

$M = m_1 \times m_2 \times \ldots \times m_k$

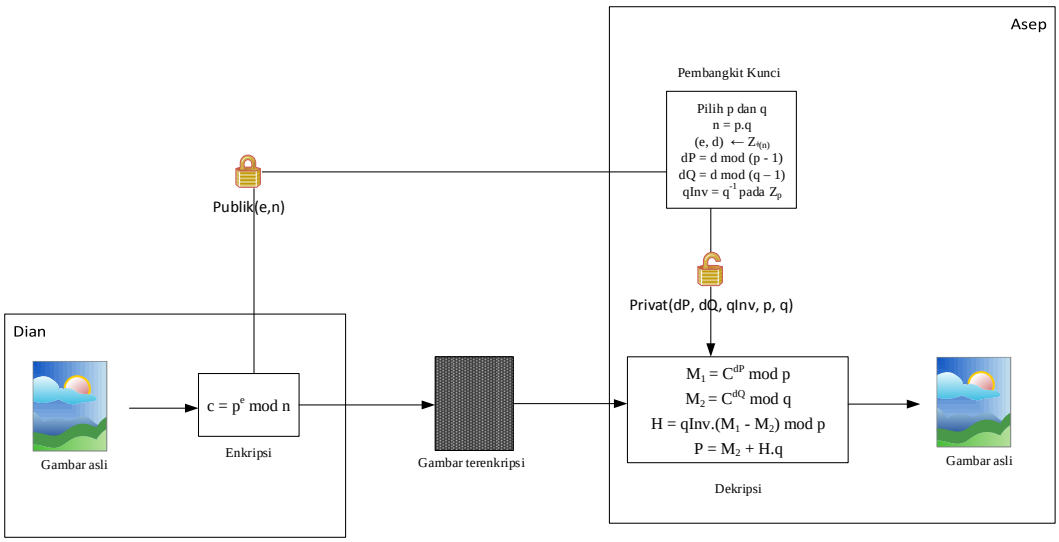
$M_1 = M/ m_1$, $M_2 = M/ m_2$, $M_3 = M/ m_3$

M_1^{-1} , M_2^{-1}, . . ., M_k^{-1}

$x = (a_1 \times M_1 \times M_1^{-1} + \ldots + a_k \times M_k \times M_k^{-1}) \mod M$

Secara sederhana penggunaan CRT pada nilai *d* mengakibatkan pemecahan kunci sehingga menghasilkan kunci baru yaitu parameter dP, dQ dan qInv yang memiliki ukuran setengah panjang bit *d*. Kunci privat RSA – CRT ditetapkan sebagai K_{privat} = (dP, dQ, qInv, p, q) [11].

Untuk gambaran sistem terlihat seperti berikut :



Bagan 2.7 Gambaran Umum Sistem Kriptografi RSA – CRT

Berikut adalah algoritma dari proses pembangkit kunci :

Algoritma Pembangkit Kunci RSA – CRT

Bangkitkan bilangan prima besar *p* dan *q*.

$n = p \times q$

$\phi(n) = (p - 1) \times (q - 1)$

$e \leftarrow Z_{\phi(n)}$ dengan $gcd(e, \phi(n)) = 1$

$d = e^{-1}$ pada $Z_{\phi(n)}$

$dP = d \bmod (p - 1)$

$dQ = d \bmod (q - 1)$

$qInv = q^{-1}$ pada Z_p

$K_{publik} = (e, n)$, $K_{privat} = (dP, dQ, qInv, p, q)$

Bagan 2.8 Algoritma Pembangkit Kunci RSA - CRT [8]

Sedangkan untuk melakukan dekripsi diperlukan algortima seperti yang tertera di bawah ini :

Algoritma Dekripsi

Input : $C = P^e \bmod n$, $K_{privat} = (dP, dQ$

Output : P

$m_1 = C^{dP} \bmod p$

$m_2 = C^{dQ} \bmod q$

$h = qInv\ (m_1 - m_2) \bmod p$

$P = m_2 + h.q$

Bagan 2.9 Algoritma Dekripsi RSA - CRT

2.2.4 Model Sistem

Model sistem merupakan representasi grafis yang mendeskripsikan masalah yang akan ditangani dan sistem yang akan dikembangkan [6]. Karena digunakan representasi grafis, model seringkali dipahami jika dibandingkan dengan deskripsi bahasa naturalnya yang rinci tentang persyaratan sistem. Representasi ini juga merupakan jembatan antara proses analisis dan perancangan.

Metode pemodelan dapat dibedakan menjadi dua yaitu pemodelan terstruktur dan pemodelan objek. Dalam pemodelan terstruktur akan memandang pemecahan masalah berdasarkan setiap fungsional pemecahan. Sedangkan untuk pemodelan objek akan melihat pemecahan masalah berdasar objek-objek yang ada berdasarkan pemahaman *end-user*. Dalam hal ini dikatakan bahwa orientasi objek ini adalah cara pandang bukan sekedar algoritma yang ditetapkan pada bahasa berorientasi objek.

Menurut Coad – Yourdon, hanya untuk sistem-sistem yang sangat sepele saja maka pendekatan berorientasi objek akan tidak berguna. Pendekatan berorientasi objek memang ditujukan untuk menangani kompleksitas besar dan berpotensi terus untuk beradaptasi menjadi sistem yang lebih kompleks [12]

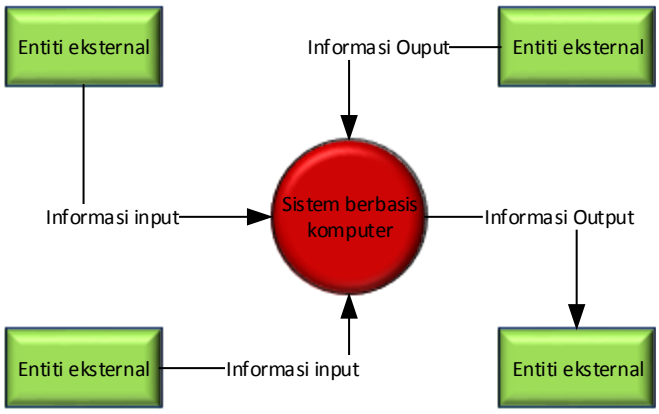
Kompleksitas dalam hal ini dipandang dari segi kompleksitas persoalan dan kompleksitas solusi. Dalam ini akan menimbulkan banyak persoalan seperti cakupan proyek yang besar, komunikasi antar tim, pengembangan yang memerlukan aplikasi yang fleksibel. Aplikasi yang fleksibel artinya aplikasi dapat dikembangkan dengan menggunakan bagian aplikasi yang sebelumnya (*reusability*) .

Berdasarkan pengertian diatas maka penelitian ini akan menggunakan pemodelan sistem terstruktur. Hal ini berdasarkan pertimbangan sistem yang akan dibangun cukup sederhana karena hanya melibatkan tiga buah fungsional dasar saja yaitu pembangkit kunci kriptografi, proses enkripsi dan proses dekripsi. Selain itu aplikasi yang akan dibuat tidak memiliki tingkat *reusability* yang tinggi, baik dari segi solusi maupun pada tingkat pengembangannya.

Pemodelan sistem terstruktur akan menggunakan analisis terstruktur dan perancangan terstruktur. Tentu saja untuk mendukung pemodelan akan digunakan beberapa alat seperti diagram konteks dan *Data-Flow-giagram*.

2.2.5 Diagram Aliran Data (DFD – *Data Flow Diagram*)

Data Flow Diagram adalah sbuah teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output [5]. Bentuk dasar dari suatu DFD dapat digambarkan sebagai berikut :



Bagan 2.10 Model Aliran Informasi [5]

Model aliran data menunjukkan sudut pandang fungsional dimana setiap transformasi merepresentasikan satu fungsi. Selain mendeskripsikan pemrosesan di dalam sistem, kadangkala akan berguna bagi deskripsi aliran data untuk mengilustrasikan konteks sistem. Model aliran data dapat menunjukkan bagaimana sistem dan subsistem yang berbeda bertukar informasi. Subsistem-subsistem ini tidak harus berupa satu fungsi [6].

DFD tingkat 0 disebut juga dengan *model sistem fundamental* atau model konteks, merepresentasikan seluruh elemen sistem sebagai sebuah *bubble* tunggal dengan data input dan output yang ditunjukkan oleh anak panah yang masuk dan keluar secara berurutan [5].

2.2.6 Bahasa Pemrograman C#

Pertama kali sistem operasi windows diperkenalkan oleh *Microsoft Corp* untuk menggantikan sistem operasi DOS (*Disk Operating System*), para pemrogram umumnya menggunakan bahasa pemrograman C untuk membuat program aplikasi diatas sistem operasi tersebut. Hal ini karena API (*Application Programming Interface*) Windows yang banyak dimanfaatkan oleh program aplikasi juga ditulis menggunakan bahasa C [13]. Bagi beberapa kalangan pemrograman C dianggap rumit karena penulis program harus mengalokasikan secara manual memori yang akan digunakan dan lebih kompleks karena adanya pointer-pointer aritmetika serta struktur program yang sulit dibaca.

Bahasa C merupakan bahasa pemrograman terstruktur dimana kita tidak bisa memanfaatkan berbagai kelebihan bahasa pemrograman yang memiliki paradigma pemrograman berorientasi objek (OOP – *Object Oriented Programming*) yang berkaitan dengan penggunaan ulang komponen (*component reusable*) seperti penyembunyian/pembungkusan (*encapsulation*), pewarisan (*inheritance*), polimorfisme dan sebagainya.

Pada masa-masa berikutnya para pemrogram bahasa C beralih ke bahasa C++ untuk melakukan pemrograman di atas sistem operasi Windows [13]. Dalam hal ini bahasa C++ dapat dianggap sebagai perluasan dari C yang diadaptasi dengan konsep pemrograman berorientasi objek. Bahasa pemrograman C++ ini sama rumitnya (bahkan beberapa orang menyebutnya lebih rumit) dibanding dengan bahasa C, meski memiliki keunggulan karena menggunakan paradigma pemrograman berorientasi objek. Meski sangat rumit, beberapa framework C++ sudah hadir saat ini, misalnya MFC (*Microsoft Foundation Classes*) yang menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi

Win32 API (kelas-kelas yang digunakan untuk mengembangkan aplikasi-aplikasi diatas sistem operasi windows yang memiliki modus 32bit)

Dalam hal tersebut, *Microsoft Corp* cukup tanggap dalam menghadapi keluhan para pemrogram akan kesulitan bahasa pemrograman C/C++ dengan mengembangkan bahasa pemrograman Visual Basic yang sangat populer dalam hal kemampuan dan kemudahannya dalam mengembangkan antarmuka pengguna (*user interface*) yang kompleks [14]. Selain itu juga *Microsoft Corp* menyediakan pustaka kelas yang lengkap dalam bentuk COM (*Component Object Module*), serta logika akses basis data yang sanga mudah bagi para pemrogram Visual Basic. Di atas MFC (*microsoft foundation classes*), Visual Basic menyembunyikan kompleksitas Win32 API dari pandangan pemrogram [14]. Meski demikian, versi terakhir visual Basic (VB versi 6) tidak dapat secara sempurna beradaptasi dengan konsep pemrograman berorientasi objek sehingga mengurangi fleksibilitas dan penggunaan ulang komponennya.

Di sudut dunia pemrograman yang lain, *Sun Mycrosystem Inc* mengembangkan bahasa permrograman Java yang memiliki sintak cukup mirip dengan bahasa C/C++ [11]. Dalam hal ini, meskipun bahasa pemrograman Java merupakan bahasa pemrograman yang elegan dalam hal adaptasinya dengan pemrograman berorientasi objek , integrasi antarprogram yang dikembangkan menggunakan bahasa pemrograman Java cukup sukar dilakukan karena sifatnya yang open source setiap vendor di seluruh dunia memiliki aneka produk yang kadang tidak kompetibel satu sama lain. Hal ini berbeda dengan produk *Microsoft Corp* yang terintegrasi sangat kuat.

Beberapa waktu yang lalu, *Microsoft Corp* mengembangkan objek-objek COM (*Component Object Module*) sedemikian rupa sehingga bisa diakses oleh pemrogram C/C++, delphi dan sebagainya sehingga integrasi antar aplikasi dapat berlangsung dengan mulus pada aplikasi-aplikasi internet yang didalamnya terkandung berbagai teknologi yang beragam. Hal ini memunculkan framework .Net yang berusaha sedemikian rupa sehingga integrasi antar aplikasi ini dapat berjalan dengan lebih mulus.

Dari sudut pandang pemrogram, .Net dapat dipandang sebagai lingkungan saat aplikasi dijalankan (*runtime enviroment*) dan sejumlah pustaka kelas (*class library*) yang memungkinkan aplikasi-aplikasi diatas sistem operasi Windows dapat berjalan dengan baik [14]. Lingkungan aplikasi yang dijalankan disebut dengan CLR (*Common Language Runtime*) yang bertugas mengelola aplikasi-aplikasi .Net (memuat ke memori, menjalankan/mengeksekusi, dan sebagainya). Selanjutnya , komponen lain yang bermanfaat untuk menspesifikasi berbagai tipe data yang digunakan dalam konturksi pemrograman disebut sebagai CTS (*Common Type System*). Dalam hal ini komponen yang bermanfaat sedemikian rupa sehingga bisa terjadi integrasi yang ketat antar aplikasi yang dibuat menggunakan berbagai bahasa keluarga .net (visual basic.net, visual c.net, visual c++.net dan sebagainya), dinamakan sebagai CLS (*Common Language Spesification*).

Bahasa pemrograman C# dirancang oleh *Microsoft Corp* sebagai bahasa pemrograman yang berdaya-guna, aman serta mudah digunakan. Sebagai bagian dari platform .Net, bahasa pemrograman C# juga dapat digunakan untuk mengembangkan aplikasi-aplikasi sarana bergerak (*mobile application*), aplikasi berbasis web (*web-based aplication*) setta aplikasi berskala besar (*Enterprise*) [15].

Dari sejarah diatas dapat disimpulkan bahwa bahasa pemrograman C# merupakan gabungan dari kecanggihan bahasa keluarga C (C, C++, Objective-C, Java dan Sebagainya) dengan kemudahan bahasa pemrograman Visual Basic [16].

Secara umum bahasa pemrograman C# memiliki karakteristik umum seperti tertulis dibawah ini [14] :

a. Tidak ada alokasi memori secara manual menggunakan pointer (hal ini mirip dengan bahasa pemrograman java).
b. Manajemen memori otomatis menggunakan salah satu fiturnya yang dinamakan *garbage collection* (hal ini juga mirip dengan bahasa java).
c. Mendukung konstruksi kelas, antarmuka, struktur dan enumerasi seperti bahasa pemrograman berorientasi objek lainnya (misalnya C++ atau Java).
d. Mendukung pemrograman berbasis atribut (*attribute-based programming*).
e. Mendukung LinQ (*Language Integrated Query*) yang memungkinkan aplikasi yang ditulis menggunakan bahasa pemrograman C# mampu

berinteraksi dan bekerjasama dengan berbagai jenis format data dimana hal ini sangat penting saat kita membuat aplikasi bahasa C# yang mengakses sistem basis data relasional (RDBMS – *Relational Database Management System*).

f. Mendukung tipe data dan kelas generik (mirip dengan C++ dan Java).
g. Mendukung operator delegasi (=>).