BAB 2
TINJAUAN PUSTAKA

2.1 Struktur Analisis dan Desain

Struktur analisis dan desain adalah sebuah metodologi yang di gunakan pada rekayasa perangkat lunak untuk mendeskripsikan sistem kearah fungsional. Pendekatan ini memecahkan masalah-masalah dalam aktifitas bisnis menjadi bagian-bagian kecil yang dapat disatukan kembali menjadi satu kesatuan yang utuh untuk memecahkan masalah [3].

Tools yang dapat digunakan pada pendekatan analisis pengembangan sistem secara terstruktur adalah:

1. ERD (*Entity Relationship Diagram*) adalah satu model jaringan yang menggunakan susunan data yang disimpan dalam sistem. ERD merupakan model jaringan data yang menekankan pada struktur dan hubungan antar data [4].
2. Diagram Konteks adalah diagram yang menggambarkan sebuah sistem secara menyeluruh yang selanjutnya akan lebih didetailkan oleh DFD. Konteks diagram juga biasa disebut sebagai DFD level 0.
3. Data Flow Diagram atau biasa di singkat DFD merupakan serangkaian diagram yang menggambarkan kegiatan-kegiatan yang ada dalam satu sistem. Teknik pembangunan DFD dimulai dengan menggambarkan sistem secara global dan dilanjutkan dengan analisis masing-masing bagian. Pada awalnya digambarkan konteks diagram yang menggambarkan sebuah sistem secara menyeluruh yang akan diinvestigasi. Konteks diagram tersebut dapat dikatakan sebagai DFD level 0. Analisis sistem yang lebih detail selanjutnya dapat dilakukan dengan menggambarkan DFD level 1,2 dan seterusnya.
4. Spesifikasi Proses merupakan tabel yang berisi keterangan deskripsi dari semua proses yang terdapat di DFD. Logika proses harus dituliskan secara jelas baik menggunakan bahasa deskriptif atau *pseudo code* (tidak boleh campuran.

5. Kamus Data (*Data Dictionary*) merupakan fakta tentang data dan kebutuhan-kebutuhan informasi dari sistem informasi. Dengan menggunakan *data dictionary*, analis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap [5].

2.2 Web Service

Web Service adalah sekumpulan *application logic* beserta objek-objek dan metode-metode yang dimilikinya yang terletak di satu server yang terhubung ke internet [6]. Tujuan dari teknologi ini adalah untuk memudahkan beberapa aplikasi atau komponennya untuk saling berhubungan dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi menggunakan standar yang tidak terikat platform (*platform-neutral*) dan tidak terikat akan bahasa pemrograman yang digunakan (*language-neutral*). Hal tersebut dapat terjadi karena penggunaan XML standar yang didukung oleh banyak perusahaan besar di dunia, yang digunakan untuk bertukar data. Selain XML, terdapat pula format pertukaran data lain yaitu JSON.

2.3 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer [7]. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari dua struktur [8]:

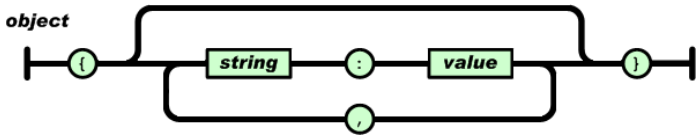
1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel *hash* (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.

2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-stuktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut:

a. Objek

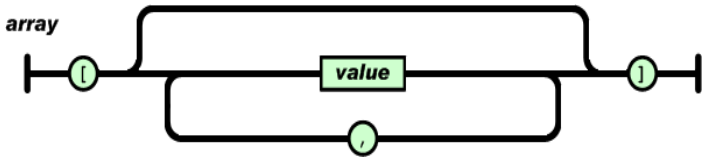
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



Gambar 2.1 Objek JSON

b. Larik

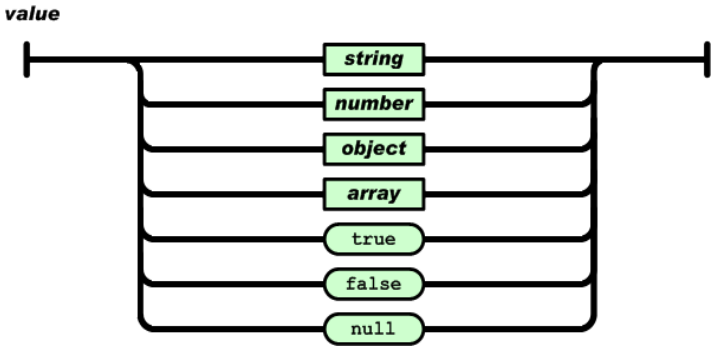
Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung siku buka) dan diakhiri dengan] (kurung siku tutup). Setiap nilai dipisahkan oleh , (koma).



Gambar 2.2 Larik JSON

c. Nilai

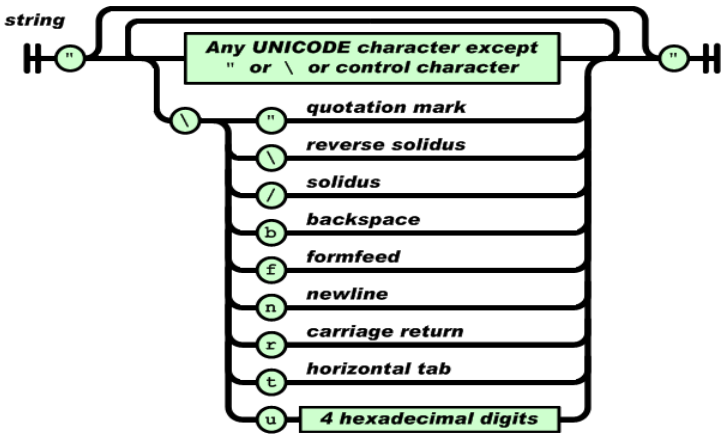
Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Gambar 2.3 Nilai JSON

d. String

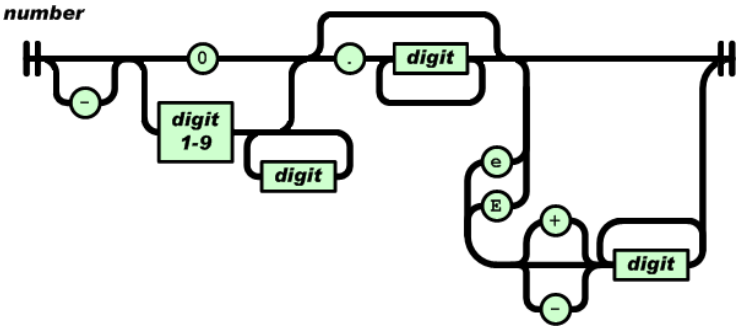
String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam *string* dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java.



Gambar 2.4 String JSON

e. Angka

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.

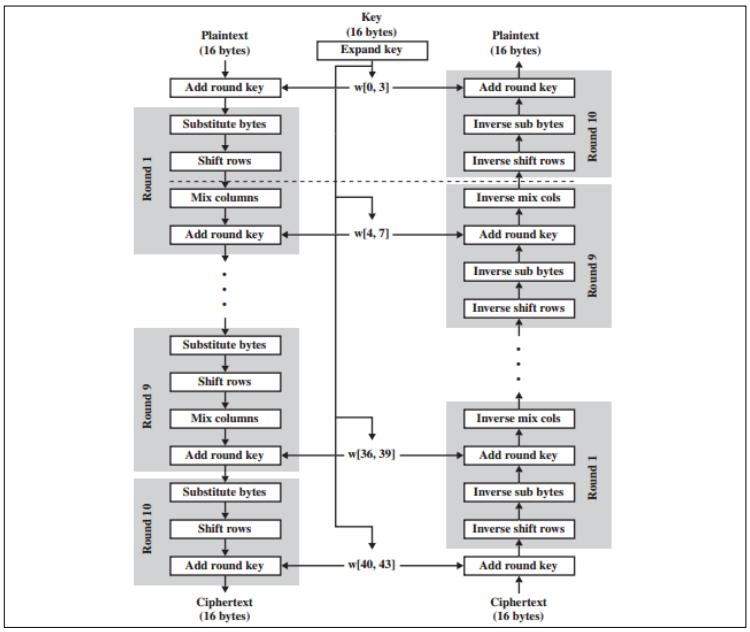


Gambar 2.5 String JSON

2.4 Kriptografi

Kriptografi didefinisikan sebagai ilmu dan pelajaran untuk tulisan rahasia dengan pertimbangan bahwa komunikasi dan data dapat dikodekan untuk mencegah dari mata-mata atau orang lain yang ingin mengetahui isinya, dengan menggunakan kode-kode dan aturan-aturan tertentu dan metode lainnya sehingga hanya orang yang berhak yang dapat mengetahui isi pesan sebenarnya. Kriptografi memiliki dua jenis cara dalam melakukan enkripsi dan dekripsi yaitu *symmetric* dan *asymmetric*. Algoritma *symmetric* disebut dengan algoritma klasik, karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsinya. Sedangkan algoritma *asymmetric* sering juga disebut dengan algoritma kunci publik, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsinya berbeda [9].

Pada penelitian ini jenis algoritma kriptografi yang digunakan adalah algoritma kriptografi simetrik dengan algoritma yang digunakan adalah AES (*Advance Encryption Standard*). Algoritma AES sendiri memiliki tiga jenis dengan panjang kunci yang berbeda, diantaranya AES-128, AES-192, dan AES-256 [10]. Proses enkripsi dan dekripsi pada algoritma AES dapat dilihat pada Gambar 2.6.



Gambar 2.6 Kriptografi pada Algoritma AES

2.5 Object Relational Mapping

Object Relational Mapping (ORM) adalah suatu teknik pemrograman yang digunakan untuk menghubungkan kode objek dengan database relasional. Kode objek ditulis dalam bahasa berorientasi objek pemrograman (OOP) seperti Java atau C#. Konsep ORM adalah melakukan pemetaan dari tabel menjadi objek. Kolom kolom yang ada pada tabel nantinya akan menjadi variabel-variabel dalam objek tersebut. Satu objek mewakili satu baris. Karena merupakan objek, untuk mengakses beberapa baris sekaligus dapat disamakan dengan mengakses array dari objek tersebut. ORM memiliki kemampuan untuk menciptakan objek database virtual, yaitu suatu model database yang direpresentasikan ke dalam sebuah objek pada bahasa pemrograman OOP. Adapun kelebihan yang dimiliki oleh ORM adalah mempercepat pengembangan program. Contohnya yaitu mengurangi perulangan kode query, dan memudahkan pemakaian karena tabel yang ada sudah tersusun dalam bentuk objek. Singkatnya ORM ini adalah sebuah konsep untuk menggabungkan metode pemrograman database dengan konsep Object Oriented Programming (OOP) [8].

2.6 Pengujian *Black Box*

Metode pengujian *blackbox* berfokus pada kebutuhan fungsional pada *software,* berdasarkan pada spesifikasi kebutuhan *software*. Pengujian dengan menggunakan metode *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

Blackbox memiliki banyak teknik pengujian diantaranya *Equivalence Partitioning, Boundary Value Analysis/Limit Testing, Comparison Testing, Sample Testing, Robustness Testing, Behavior Testing, Requirement Testing, Performance Testing, Endurance Testing, Cause-Effect Relationship Testing*. Salah satu teknik yang akan di bahas adalah *Equivalence Partitioning*.

Equivalence Partitioning merupakan metode black box testing yang membagi domain masukan dari suatu program menjadi beberapa kelas data dari kasus uji coba yang dihasilkan [11].