# BAB II
# LANDASAN TEORI

## II.1. Dodo Kids Browser

Dodo Kids Browser merupakan aplikasi parental *controlling* terhadap kegiatan internet anak. Aplikasi ini terdiri dari dua jenis platform yaitu berupa *extension* pada *web browser desktop* dan *mobile* Windows Phone. Dodo Kids Browser Extension pada *web browser desktop* digunakan sebagai fungsi tambahan pada aplikasi web browser untuk melakukan *filtering* terhadap aktifitas pencarian anak di mana jika anak melakukan pencarian dengan menggunakan *keyword* yang mengandung kata negative maka aplikasi ini akan mengirimkan notifikasi kepada orang tua yang menggunakan aplikasi Dodo Kids Browser versi *mobile* Windows Phone yang berguna untuk memberikan aksi kepada anak yang sedang melakukan *browsing* tersebut seperti memberikan suatu pesan atau nasihat. Selain untuk memberikan aksi aplikasi Dodo Kids Browser Windows Phone yang digunakan oleh orang tua dapat melakukan monitoring terhadap aktifitas browsing anak. Pada aplikasi *mobile* version-nya tersebut pun dapat memiliki *kids* mode yang dapat digunakan oleh anak untuk melakukan *browsing* aman dengan menggunakan perintah suara (*speech recognition*) atau teks seperti biasa.

## II.2. *Text Mining*

*Text Mining* merupakan suatu langkah dari analisis teks yang dilakukan secara otomatis oleh sistem komputer untuk menghasilkan informasi baru yang belum diketahui sebelumnya yang diambil dari suatu rangkaian teks yang terangkum dalam sebuah dokumen [5]. Text Mining adalah bidang multi disiplin yang melibatkan information retrieval, text analysis, information extraction, clustering, categorization, visualization, machine learning dan teknik lainnya [6]. Text mining menggunakan penerapan data mining untuk mengubah data tidak terstruktur menjadi data terstruktur melalui tahap-tahap yaitu [6]:

1. *Text Preprocess* yaitu pemecahan sekumpulan karakter ke dalam kata-kata.

2. *Feature Generation* / *Text Transformation* yaitu mengubah kata-kata ke dalam bentuk dasar sekaligus mengurangi jumlah kata-kata tersebut.
3. *Feature Selection* yaitu seleksi feature untuk mengurangi dimensi dari suatu kumpulan teks.
4. *Text Mining* / *Pattern Discovery* yaitu dapat berupa *unsupervised learning* (*clustering*) atau *supervised learning* (*classification*).
5. *Interpretation / Evaluation* yaitu pengukuran efektifitas untuk mengevaluasi metode yang diterapkan menggunakan parameter *precision*.

## II.3. Analisis Sentimen

Analisis sentimen atau dapat disebut juga *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data teks tekstual secara otamatis untuk mendapatkan informasi sentiment yang terkandung dalam suatu kalimat opini [7]. Analisi sentimen bertujuan menentukan suatu isi dari dataset yang berbentuk tesktual atau kalimat apakah bernilai sentimen positif atau negatif [8]. *Opinion mining* dapat dianggap pula sebagai kombinasi antara *text mining* dan natural *language processing*. Metode klasifikasi merupakan metode yang dapat digunakan untuk menyelesaikan masalah pada *text mining*. Salah satunya yaitu dengan menggunakan algoritma *Naïve Bayes Classifier* (NBC). Sedangkan Natural *language processing* befungsi untuk memberikan kelas kata (*tag*) ke setiap kata dalam suatu kalimat.

## II.4. *Preprocessing*

*Preprocessing* merupakan tahapan sebelum proses pengklasifikasian yang diperlukan untuk membersihkan, menghilangkan, mengubah sumber data, baik itu berupa karakter non alfabet maupun kata-kata yang tidak diperlukan. Hal ini bertujuan agar data yang digunakan lebih optimal ketika digunakan pada proses pengklasifikasiannya.Tahapan preprocessing setiap kasus dapat berbeda-beda. Berikut ini merupakan tahapan preprocessing dan penjelasannya yang digunakan dalam penelitian ini.

1. *Cleansing*

*Cleansing* merupakan proses membersihkan data yang akan digunakan dari karakter-karakter bahkan kata-kata yang tidak diperlukan. Hal ini bertujuan untuk mengurangi noise yang dapat menimbulkan proses perhitungan dalam pengklasifikasian tidak optimal.

2. *Case Folding*

*Case Folding* merupakan proses pengubahan data menjadi format yang sesuai. Hal ini bertujuan mengurangi redudansi data yang akan digunakan dalam proses pengklasifikasian sehingga proses perhitungan pun menjadi optimal. Contohnya mengubah format data menjadi lowercase atau uppercase sesuai dengan kebutuhan yang dibutuhkan dalam proses pengklasifikasiannya.

3. *Tokenizing*

*Tokenizing* merupakan proses pemisahan atau memotong data baik berupa frasa, klausa, atau kalimat menjadi kata perkata berdasarkan delimeter yang digunakan yaitu space.

## II.5. *Naïve Bayes Classifier* (NBC)

*Naïve Bayes Classifier* merupakan suatu metode *classifier* yang mengacu pada teorema *bayes* yaitu teorema yang mengacu pada konsep probabilitas bersyarat. Pada metode ini diperlukan kombinasi pengetahuan sebelumnya dengan pengetahuan yang baru [9]. Dalam melakukan pengklasifikasiannya diperlukan *training* set sebagai data latih. Pada setiap *sample* dari data latih tersebut memiliki kelas label tersendiri. Berikut merupakan model matematis *naïve bayes classifier* yaitu:

$$p(H|X) = \frac{p(H)p(X|H)}{p(X)}$$ [2]

Di mana:

X = Data dengan kelas yang belum diketahui

H = Hipotesa data X merupakan suatu kelas spesifik

p(H|X) = Probabilitas hipotesis H berdasarkan kondisi X (*posterior probability*)

p(H) = Probabilitas hipotesis H (*prior probabilty*)

## II.6. *Object-Oriented Analysis and Design* (OOAD)

*Object-Oriented Analysis and Design* (OOAD) merupakan suatu metode analisis dalam pembangunan perangkat lunak yang berorientasi pada objek. Proses analisis yang dilakukan yaitu dengan memeriksa kebutuhan berdasarkan sudut pandang kelas dan objek yang terdapat pada ruang lingkup permasalahan. Proses analisis dilakukan menggunakan model menurut konsep sekitar dunia nyata yang dalam konsep tersebut ialah objek yang merupakan gabungan dari struktur data serta perilaku dalam suatu entitas. Dalam metode OOAD terdapat konsep umum salah satunya yaitu kelas. Kelas merupakan kumpulan dari objek-objek yang memiliki karakter serupa. Sebuah kelas memiliki atribut, operasi/metode, relasi, dan arti. Atribut dalam kelas merupakan suatu variable global berupa nilai atau elemen data suatu objek. Metode dalam kelas berguna untuk mengoperasikan suatu objek tertentu. Suatu kelas dapat mewarisi sebagian objek atau seluruhnya ke kelas yang baru di mana hal ini disebut dengan *inheritance* (pewarisan). Dalam suatu kelas terdapat objek yaitu suatu entitas yang dapat menyimpan informasi serta mampu melakukan operasi tertentu.

Dalam memodelkan OOAD terdapat *tools* yang biasa digunakan yaitu UML (*Unified Modeling Languages*). UML merupakan sebuah bahasa standar dalam industri untuk proses visualsasi, perancangan, dan pendokumentasian perangkat lunak. Berikut adalah diagram yang umum digunakan dalam analisis dan desain.

1. *Use Case Diagram*

   Use Case Diagram menggambarkan fungsional berdasarkan sistem yang akan dibangun. Dalam sebuah usecase terdapat aktor yang menunjukan interaksi pengguna dengan fungsional dalam suatu sistem. [10]

2. *Use Case Scenario*

   Use case scenario digunakan untuk mendeskripsikan informasi dari interaksi aktor dan sistem secara detail.

3. *Activity Diagram*

*Activity diagram* menggambarkan alur dari aktifitas satu secara sekuensial use case mulai dari awal fungsional dijalankan hingga sampai tercapainya tujuan dari use case itu sendiri.

4. *Class Diagram*

   *Class diagram* menggambarkan hubungan antara class, interface, beserta objek pada suatu sistem.

5. *Sequence Diagram*

   *Sequence diagram* menggambarkan interaksi di antara *object* secara berurutan mulai dari objek diciptakan sampai berakhir aktifitas pada objek tersebut.

## II.7. JSON

JSON (JavaScript Object Notation) merupakan format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia serta mudah dibuat dan diterjemahkan oleh komputer [11]. JSON terdiri dari objek dan *array*. Objek merupakan sepasang nama dan nilainya yang tidak memiliki urutan. Objek menggunakan simbol { (kurung kurawal terbuka) dan } (kurung kurawal tertutup) sebagai pembuka dan penutup. Array merupakan kumpulan nilai yang memiliki urutan dengan [ (kurung siku buka) sebagai pembuka dan ](kurung siku tutup) sebagai penutup array. Setiap objek dan array dapat di definisikan oleh item dan nilainya dimana nama item dan nilainya dipisahkan dengan : (titik dua) dan diakhir dengan , (koma) bila memiliki item terurut lebih dari satu.

## II.8. XML

XML merupakan dasar terbentuknya web service yang digunakan untuk mendeskripsikan data. Web Service dibentuk diatas XML secara keseluruhan [12]. XML berfungsi sama halnya seperti JSON yaitu sebagai format pertukaran data antar sistem yang berbeda-beda dengan standarisasi yang serupa dengan penulisan web html yang mudah dibaca dan dipahami.

## II.9. Pengujian Black box

Pengujian *black box* merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian black box memungkinkan pengembang perangkat lunak untuk mengetahui berbagai kondisi masukan terhadap syarat-syarat fungsional suatu sistem. Pengujian black box ini merupakan upaya untuk menemukan kesalahan-kesalahan antara lain yaitu kesalahan fungsi atau fungsi yang hilang, kesalahan antarmuka, kesalahan struktur data, kesalahan performa, dan kesalahan dalam inisialisasi dan terminasi. Pada pengujian black box terdapat teknik pengujian yang membagi domain input dari suatu program menjadi beberapa kelas data dari kasus ujicoba yang dihasilkan. Metode pengujian black box tersebut dinamakan equivalence partioning. Equivalence Partioning berfokus pada kasus uji untuk menemukan sejumlah jenis kesalahan dan mengurangi jumlah kasus ujicoba yang harus dibuat. Kondisi input yang diuji pada metode Equivalence Partioning ini merepresentasikan apa yang dihasilkan oleh inputan tersebut valid atau tidak. Misalkan kondisi input yang diharapkan sebuah program yang diinginkan yaitu sebuah teks dengan format yang telah ditetapkan, sehingga jika kondisi tidak sesuai maka hasilnya menjadi tidak valid.

## II.10. Pengujian White Box

Pengujian *White Box* merupakan metode *test case* dengan menggunakan struktur control desain prosedural untuk memperoleh kasus-kasus uji. Tujuan dari pengujian white box ini, pengembang perangkat lunak dapat menghasilkan kasus-kasus uji sebagai berikut:

1. Menjamin bahwa seluruh *independent paths* dalam modul telah dilakukan sedikitnya satu kali.
2. Melakukan seluruh keputusan logika baik dari segi yang benar maupun yang salah.
3. Melakukan seluruh perulangan sesuai dengan batasan yang telah ditentukan.
4. Menguji struktur data internal untuk memastikan validitasnya.

Pengujian white box diperlukan karena adanya sifat kerusakan alami pada perangkat lunak itu sendiri antara lain yaitu:

1. Kesalahan logika dan asumsi.
2. Mengasumsikan bahwa alur logika tidak akan dieksekusi dalam dunia nyata.
3. Kesalahan typographical (cetakan) bersifat random. [13]