BAB II
TINJAUAN PUSTAKA

II.1. Perangkat Lunak Dodo Kids Browser

Dodo kids browser adalah perangkat lunak untuk mengontrol dan mengawasi anak-anak dalam kegiatan menggunakan internet, dimana anak-anak dapat mengeksplorasi internet dengan aman dan orang tua tidak merasa khawatir tentang bahaya penggunaan internet yang dilakukan oleh anak.

. Dodo Kids Browser berbentuk Extension yang dipasang pada web browser desktop, hal tersebut digunakan sebagai fungsi tambahan pada aplikasi web browser untuk melakukan filtering dengan menggunakan algoritma Naïve bayes. Ketika anak melakukan pencarian dengan menggunakan keyword yang mengandung kata negative, maka aplikasi ini akan melakukan blocking website, sehingga memungkinkan mengamankan aktivitas browsing yang dilakukan anak.

II.1.1 Fitur Dodo Kids Browser

Dodo kids browser memiliki beberapa fitur utama, yaitu :

1. Surfior

   Fitur untuk menyaring penggunaan internet anak-anak.

2. Notifior

   Fitur untuk memberikan pemberitahuan ketika anak-anak terindikasi mengakses konten yang negatif.

3. Reportior

   Fitur reportior berpua laporan tentang penggunaan internet anak-anak, laporan tersebut berupa tabel.

II.2. Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam analisis peningkatan akurasi pada perangkat lunak dodo kids browser.

II.2.1 Web browser

Web browser adalah perangkat yang digunakan untuk mengakses halaman website dan menampilkan informasi yang disediakan server web. Informasi yang ditampilkan pada Web browser dapat berupa text, gambar, suara maupun video [5]. Web browser biasanya sudah tersedia di sistem operasi yang digunakan, seperti pada sistem operasi windows, sudah tersedia web browser internet explorer, namun menurut data statistik w3school pada tahun 2015, web browser yang sering digunakan untuk browsing adalah google chrome, firefox, dan opera [7].

II.2.2 Extension Browser

Extension browser adalah perangkat lunak kecil yang ditanamkan pada web browser untuk menambah fungsionalitas tertentu dari web browser. Extension browser berinteraksi dengan halaman web yang dibuka melalui skrip tertentu yang dapat digunakan untuk membaca address bar maupun isi dari halaman web. Beberapa Extension browser ditulis menggunakan teknologi web seperti html, javascript, dan css. Extension browser dapat mengubah antarmuka yang ditampilkan di web browser melalui proses yang tidak terlihat atau proses yang berjalan dibelakang layar [6].

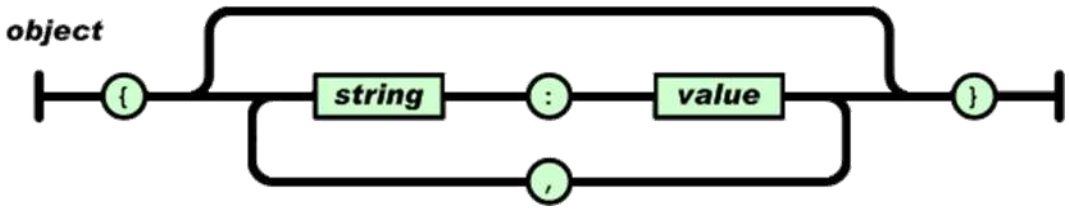
II.2.3 JSON

JSON kepanjangan dari (JavaScript Object Notation) adalah format data interchange-ringan yang sangat mudah dibaca dan ditulis. Hal ini didasarkan pada subset dari bahasa pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang benar-benar bahasa independen tetapi menggunakan konvensi yang akrab bagi programmer dari C, termasuk C , C ++, C #, Java, JavaScript, Perl, Python, dan banyak lainnya. Properti ini membuat JSON menjadi bahasa data interchange yang ideal.

JSON dibentuk oleh dua struktur, yaitu collection dari nama atau value pairs. Dalam berbagai bahasa, ini direalisasikan object, record, struct, dictionary, hash table, keyed list, atau associative array. Selain itu ordered list dari value.

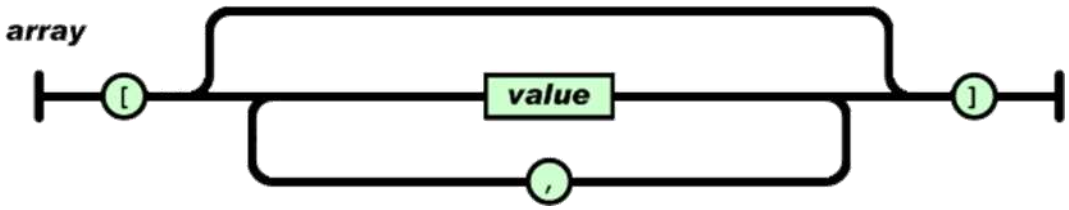
Dalam kebanyakan bahasa, ini dinyatakan sebagai array, vector, list, atau sequence. Berikut merupakan bentuk dari JSON.

Object merupakan unordered set dari nama atau value pairs dimulai dengan {(kurung kurawal kiri) dan diakhiri dengan} (kurung kurawal kanan). Setiap nama diikuti dengan: (titik dua) dan nama atau value pairs dipisahkan oleh, (koma).



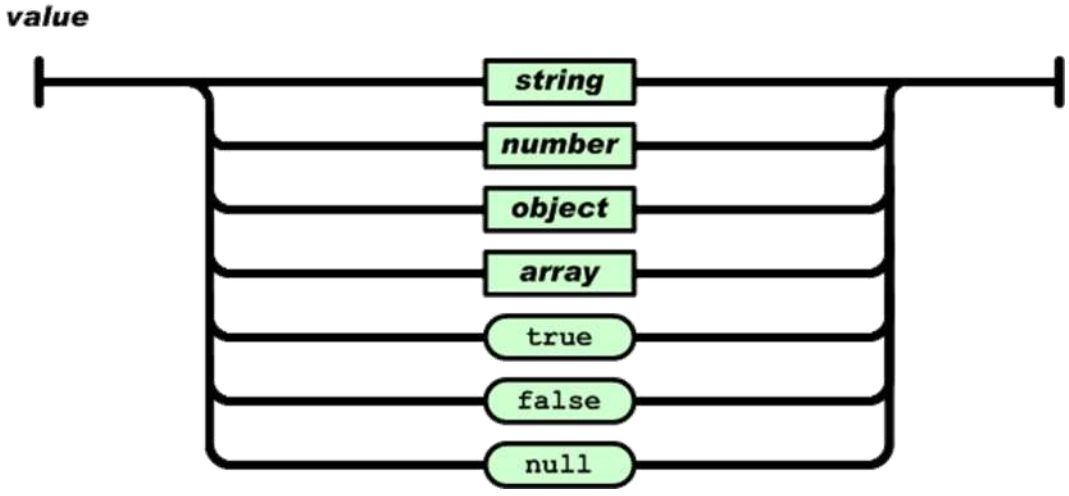
Gambar II.1 Object Pada JSON

Array merupakan ordered list dari value. Array dimulai dengan [(kurung siku kiri) dan diakhiri dengan] (kurung siku kanan). Value dipisahkan oleh, (koma).



Gambar II.2 Array Pada JSON

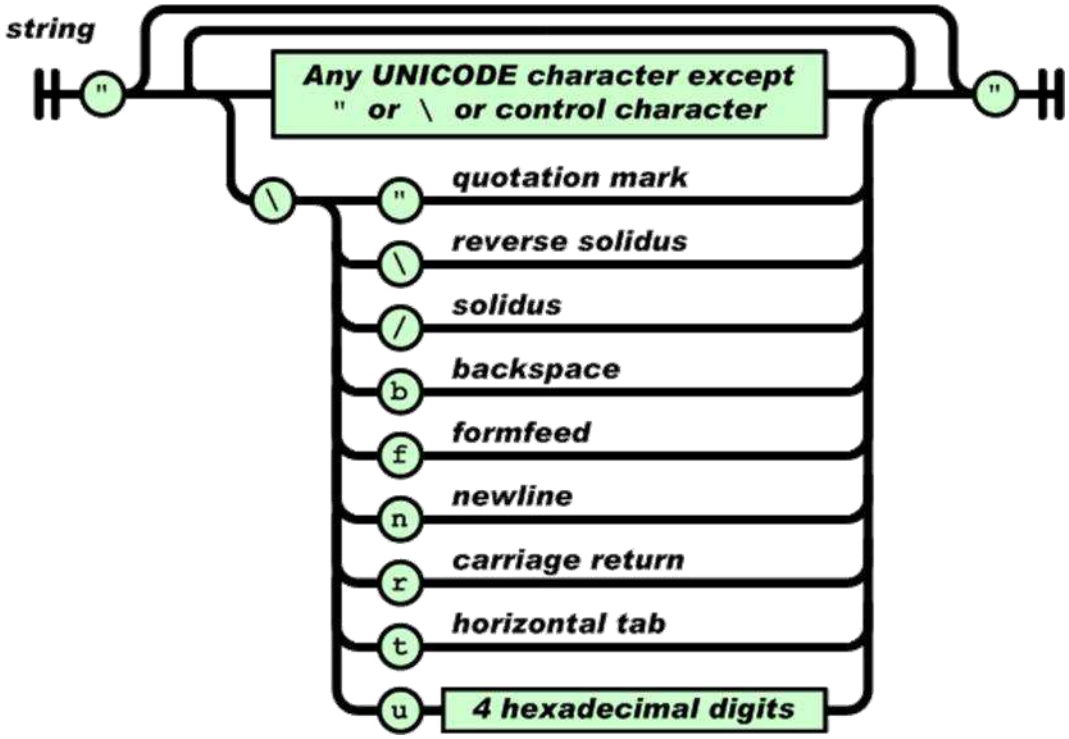
Value dapat menjadi string dalam tanda kutip ganda, number, true, false, null, object, ataupun array. Selain itu struktur ini dapat berupa nested.



Gambar II.3 Value Pada JSON

Sebuah string adalah urutan nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda, menggunakan escape backslash. Sebuah

karakter yang direpresentasikan sebagai string karakter tunggal. Sebuah string sangat mirip seperti C atau Java string.



Gambar II.4 Unicode Pada JSON

Number sangat mirip C atau Java number, terkecuali format oktal, dan heksadesimal yang tidak digunakan[7].

II.2.4 Web Service

Web Service adalah teknologi yang diisi sekumpulan method. Web service terletak pada server yang terhubung ke internet dan dapat diakses oleh berbagai piranti dengan menggunakan perantara tertentu. Teknologi tersebut digunakan untuk memudahkan beberapa aplikasi atau komponennya untuk saling berhubungan dengan aplikasi lain, tanpa terikat bahasa pemrograman yang digunakan [8].

II.2.5 Microsoft Azure

Microsoft Azure merupakan platform cloud computing yang disediakan oleh Microsoft. Microsoft Azure menyediakan berbagai macam layanan yang dapat digunakan pengembang perangkat lunak untuk menyelesaikan tugas-tugas yang

mungkin tidak sesuai dalam lingkungan lokal. Misalkan komputasi, storage, termasuk web service.

Dengan menggunakan Windows Azure, pengembang dapat membuat aplikasi web, tempat penyimpanan data suatu aplikasi, jembatan koneksi antar platform, dan melakukan push notification yang disimpan dan diproses di cloud [9].

II.2.6 Preprocessing

Preprocessing adalah tahap yang dilakukan sebelum proses klasifikasi teks. Tujuan preprocessing tersebut adalah untuk lebih memudahkan dan mengefektifkan proses mining. Tahap yang dilakukan pada preprocessing adalah Cleansing, Case Folding, Tokenizing, Stopping dan Stemming. Berikut ini penjelasan tahap-tahap yang dilakukan pada preprocessing [11]:

1. Cleansing

Data cleansing adalah proses membersihkan data dari item data yang tidak memberikan informasi berguna dalam analisis selanjutnya. Selain simbol seperti slash (/), “http://www”, domain dan angka akan dihapus, karena hal-hal tersebut dianggap tidak memberikan informasi yang berguna.

2. Case Folding

Case Case Folding disini merupakan tahapan merubah semua inputan huruf menjadi lower case. Hal tersebut dilakukan agar mempermudah dalam proses langkah-langkah selanjutnya seperti stopping, dan stemming.

3. Stopping

Proses ini adalah proses menghilangkan kata yang tidak relevan, agar tidak menimbulkan kerancuan dalam proses pengklasifikasian. Proses penghilangan kata dilakukan dengan cara melakukan pencocokan kata yang dimasukan dengan stoplist. Jika ditemukan kata yang termasuk pada stoplist ,maka kata tersebut akan dihilangkan.

4. Tokenizing

Tokenizing merupakan tahapan melakukan pemecahan suatu gabungan dua kata atau lebih atau dapat disebut juga seperti suatu frasa atau kalimat sehingga menjadi persatu satu.

5. Stemming

Kata-kata yang muncul pada dalam dokumen sering kali mengandung imbuhan. Oleh karena itu, setiap kata yang tersisa dari proses hasil tahapan stopword removal dibentuk ke dalam kata dasar dengan cara menghilangkan imbuhannya.

II.2.7 Pembobotan TF-IDF

Pada penelitian yang dilakukan oleh Bruno Trstenjak, Sasa Mikac dan Dzenana Donko pada penelitian "KNN with TF-IDF Based Framework for Text Categorization" menunjukan, dengan mengkombanasikan algoritma klasifikasi KNN dan menggunakan metode pembobotan kata TF-IDF, menunjukan hasil klasifikasi yang baik [13].

Metode TF-IDF adalah metode yang dapat digunakan untuk mendapatkan bobot dari data training yang akan menentukan pengklasifikasian pada data testing [10]. Formula yang digunakan untuk menghitung bobot adalah $W_{d,t}=Tf_{d,t}*Idf_t$ dengan,

W = bobot dokumen ke-d

Tf = frekuensi dari kemunculan sebuah term

Idf = Nilai invers document frequency

Rumus mencari nilai Idf adalah log(d/df) dengan,

Df = banyak term yang muncul pada dokumen ke-d

D = dokumen ke d.

Sebagai contoh, Tabel II-1 adalah dokumen-dokumen yang telah memiliki klasifikasi.

Tabel II-1 Data Training

Data Training	Teks	Term	Klasifikasi
d1	porn teen fuck sexy porn	porn teen fuck sexy	Bad
d2	milf fuck porn fuck learn fuck	milf fuck porn learn	Bad
d3	learn physics learn mathematics	learn physics mathematics	Good
d4	porn massage hardcore	porn massage sexy	Bad

Dari data training pada Tabel II-1 akan dilakukan klasifikasi dokumen baru, sebagai contoh data testing pada Tabel II-2 adalah dokumen yang akan diklasifikasikan.

Tabel II-2 Data Testing

Data Testing	Teks	Klasifikasi
t1	learn fuck milf	?

Tahap pertama yang dilakukan adalah menghitung Term frequency (tf) dan Document frequency (df). Term frekuensi adalah frekuensi dari kemunculan sebuah term dalam yang bersangkutan, dan Document frequency (df) adalah jumlah file yang mengandung term yang bersangkutan, dimana nilai df selanjutnya digunakan untuk menghitung Nilai invers document frequency (idf) dari sebuah term. Proses perhitungan nilai Tf dan Idf dapat dilihat pada Tabel II-3.

Tabel II-3 Proses Perhitungan TfIdf

Term	Df	Ida	test	tf1	tf2	tf3	tf4
Learn	2	0.3010	1	0	1	2	0
Fuck	2	0.3010	1	1	3	0	0
Milf	1	0.6021	1	0	1	0	0
Porn	3	0.1249	0	2	1	0	1
Teen	1	0.6021	0	1	0	0	0
Sexy	2	0.3010	0	1	0	0	1
Physics	1	0.6021	0	0	0	1	0
mathematics	1	0.6021	0	0	0	1	0
Massage	1	0.6021	0	0	0	0	1

Setelah nilai tf dan idf ditemukan, proses penghitungan bobot dilakukan dengan mengalikan setiap nilai dengan mengalikan masing-masing nilai tf terhadap nilai idf. Sehingga didapatkan bobot pada masing-masing dokumen seperti pada .

Tabel II-4 Hasil Pembobotan

Term	w(test)	w1	w2	w3	w4
Learn	0.3010	0	0.3010	0.6021	0
Fuck	0.3010	0.3010	0.9031	0	0
Milf	0.6021	0	0.6021	0	0
Porn	0	0.2499	0.1249	0	0.1249
Teen	0	0.6021	0	0	0
Sexy	0	0.3010	0	0	0.3010
Physics	0	0	0	0.6021	0

Mathematics	0	0	0	0.6021	0
Massage	0	0	0	0	0.6021

II.2.8 Algoritma K-Nearest Neighbor

Algoritma k-nearest neighbor (KNN) adalah algoritma yang digunakan untuk melakukan klasifikasi terhadap suatu objek, berdasarkan k buah data latih yang jaraknya paling dekat dengan objek tersebut. Syarat nilai k adalah tidak boleh lebih besar dari jumlah data latih, dan nila k harus ganjil dan lebih dari satu. Dekat atau jauhnya jarak data latih yang paling dekat dengan objek yang akan diklasifikasi, dapat dihitung dengan menggunkan metode cosine similiarity [11].

Cosine similiarity merupakan salah satu cara atau metode yang dapat digunakan untuk melihat sejauh mana kemiripan isi antar dokumen. Dalam hal ini cosine similiarity berfungsi untuk menguji ukuran yang dapat digunakan sebagai interpretasi kedekatan jarak berdasarkan kemiripan dokumen.

Berikut ini adalah rumus untuk menghitung jarak pada algoritma KNN dengan metode cosine similiarity:

$$Cos(\Theta y) = \frac{\sum_k (d_{ik} d_{jk})}{\sqrt{\sum_k d_{ik}^2} \sqrt{\sum_k d_{jk}^2}}$$

Dengan,

$Cos(\Theta y)$ = jarak antara dokumen d_i dan d_j

d_k = nilai bobot pada dokumen

k = jumlah tetangga terdekat

Langkah pertama yang dilakukan untuk mengetahui jarak antar dokumen adalah dengan melakukan penghitungan perkalian skalar antara data testing dengan setiap data training. Setelah perkalian skalar dilakukan, hasil perkalian dari setiap data training dijumlahkan, dan panjang setiap dokumen dapat pun dapat ditentukan dengan cara menguadratkan bobot setiap term dalam setiap dokumen.

Hasil perhitungan perkalian skalar antara data testing dengan setiap data training dengan menggunakan data yang ada pada Tabel II-4, dapat dilihat pada

Tabel II-5. Hasil perhitungan cosine similiarity untuk menentukan kemiripan antar dokumen pada Tabel II-6.

Tabel II-5 Hasil Perkalian Skalar Data Testing Dengan Data Training

	t*di				panjang vektor				
	d1	d2	d3	d4	T	d1	d2	d3	d4
	0.0000	0.0906	0.1812	0.0000	0.0906	0.0000	0.0906	0.3625	0.0000
	0.0906	0.2718	0.0000	0.0000	0.0906	0.0906	0.8156	0.0000	0.0000
	0.0000	0.3625	0.0000	0.0000	0.3625	0.0000	0.3625	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0625	0.0156	0.0000	0.0156
	0.0000	0.0000	0.0000	0.0000	0.0000	0.3625	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0906	0.0000	0.0000	0.0906
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3625	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3625	0.0000
Σ	0.0906	0.7250	0.1812	0.0000	0.5437	0.6062	1.2843	0.3625	0.0156
					0.7374	0.7786	1.1333	1.0429	0.6864

Langkah selanjutnya adalah menerapkan rumus cosin similiarity. Rumus tersebut digunakan untuk mendapatkan tingkat kemiripan antara data testing dengan data training.

$$\text{Cos(t,d1)} = \frac{0.0906}{0.7374 * 0.7786} = 0.1578$$

$$\text{Cos(r,d2)} = \frac{0.7250}{0.7374 * 0.1333} = 0.8675$$

$$\text{Cos(r,d3)} = \frac{0.1812}{0.7374 * 0.6021} = 0.2357$$

$$\text{Cos(r,d4)} = \frac{0.0000}{0.7374 * 0.6864} = 0$$

Hasil perhitungan pengukuran tingkat kemiripan antara data testing dengan data training dapat dilihat pada tabel Tabel II-6.

Tabel II-6 Hasil Perhitungan Kemiripan Data Testing Dengan Data Training

Data training	Jarak	Klasifikasi
d1	0.1578	Bad
d2	0.8675	Bad
d3	0.2357	Good
d4	0	Bad

Setelah didapatkan hasil perhitungan pengukuran jarak antara data testing dengan data training, maka langkah selanjutnya adalah mengurutkan data hasil perhitungan dari yang terbesar hingga yang terkecil. Data yang telah diurutkan dapat dilihat pada Tabel II-7.

Tabel II-7 Hasil Pengurutan

Data training	Jarak	Klasifikasi
d2	0.8675	Bad
d3	0.2357	Bad
d1	0.1578	Good
d4	0	Bad

Langkah yang dilakukan setelah mengurutkan data hasil perhitungan dari yang terbesar hingga yang terkecil, adalah menentukan jumlah tetangga terdekat atau disimbolkan dengan k. pengambilan nilai k dilakukan secara bebas, asalkan nilai k yang diambil adalah angka ganjil, nilai k lebih dari satu dan nilai k tidak lebih besar dari data training. [11] sebagai contoh nilai k yang diambil disini adalah 3. Dengan k=3, maka didapat dokumen yang menjadi tetangga terdekat adalah D2 dengan klasifikasi bad, D3 dengan klasifikasi bad, dan D1 dengan klasifikasi good.

Setelah diketahui sejumlah k objek yang terdekat dengan data yang diuji. Nilai kelas yang paling banyak muncul pada sejumlah k objek yang termasuk tetangga terdekat dijadikan hasil klasifikasi pada data testing. Karena data tetangga terdekat "Dt" memiliki dua buah dokumen yang memiliki kelas bad dan satu buah yang memiliki kelas good, maka dari itu "Dt*"* diklasifikasikan sebagai dokumen dengan kelas bad.

II.2.9 Object Oriented Analysis and Design

Object-Oriented Analysis and Design (OOAD) adalah suatu metode analisis dalam pembangunan perangkat lunak yang berorientasi pada objek yang saling berinteraksi satu sama lain untuk menyelesaikan suatu tugas. Analisis dilakukan dengan memeriksa kebutuhan berdasarkan sudut pandang kelas dan objek yang terdapat pada ruang lingkup permasalahan. OOAD dirancang untuk menutup kelemahan dari pemrograman berorientasi prosedural yang sulit untuk dikelola dan

debug ketika kode yang dibuat telah mencapai beribu-ribu baris. Dengan pendekatan ini kode-kode program dipecah ke dalam beberapa segmen atau biasa disebut prosedur atau fungsi, sehingga memudahkan analis untuk melakukan analisis dalam membangun suatu perangkat lunak. Dalam memodelkan suatu analisis yang menggunakan metode OOAD, terdapat tools yang biasa digunakan untuk menggambarkan analisis yang dibuat, yaitu UML (Unified Modeling Languages).

II.2.10 UML

UML (Unified Modeling Language) merupakan salah satu tools untuk memvisualisasikan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OOP (Object Oriented Programming). Diagram yang umum dipakai untuk desain suatu sistem adalah sebagai berikut [12]:

1. Diagram Use Case.

   Diagram Use case dapat digunakan untuk menggambarkan fungsionalitas dari sebuah sistem. Sebuah Use case diagram dapat merepresentasikan sebuah interaksi antara aktor dengan sistem.
2. Diagram Sequence.

   Diagram Sequence menggambarkan interaksi pada sistem dan digambarkan terhadap waktu. Oleh karena itu sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan keluaran tertentu.
3. Diagram aktivitas.

   Diagram aktivitas menggambarkan berbagai alur aktivitas suatu sistem, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi, dan bagaimana akhir dari alur suatu aktivitas. Diagram aktivitas dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang melakukan aktivitas tertentu
4. Diagram kelas.

Diagram kelas adalah sebuah diagram yang menggambarkan objek, atribut, dan method pada suatu sistem. Diagram kelas juga digunakan untuk menggambarkan keseluruhan hubungan objek.

II.2.11 Confusion Matrix

Confusion Matrix berisi informasi tentang klasifikasi aktual dan klasifikasi dari diprediksi.

Metode ini menggunakan tabel matriks seperti pada Tabel II-8, jika data set hanya terdiri dari dua kelas, misalkan kelas yang satu dianggap sebagai good dan yang lainnya bad.

Tabel II-8 Confusion Matrix

		Diklasifikasikan sebagai (Predicted Class)	
		Good	Bad
Klasifikasi yang benar (Actual Class)	Good	True Good	False Bad
	Bad	False Good	True Bad

True Good adalah jumlah record Good yang diklasifikasikan sebagai Good, false Good adalah jumlah record Bad yang diklasifikasikan sebagai Good, false Bad adalah jumlah record Bad yang diklasifikasikan sebagai Good, true Bad adalah jumlah record Bad yang diklasifikasikan sebagai Bad. Setiap kolom dari confusion matrix merupakan contoh di kelas yang telah diprediksi, sedangkan setiap baris mewakili contoh di kelas yang sebenarnya. Setelah didapat true Good, false Good, true Bad dan false Bad, selanjutnya hitung untuk menghitung nilai akurasinya. Berikut persamaan untuk menghitung akurasi.

$$\text{Accuracy} = \frac{\text{TrueGood} + \text{TrueBad}}{\text{TrueGood} + \text{FalseGood} + \text{TrueBad} + \text{FalseBad}}$$