PEMBANGUNAN BACKEND UNTUK APLIKASI PENGAWASAN PENGGUNAAN INTERNET ANAK "DODO KIDS BROWSER" DENGAN TEKNOLOGI MICROSOFT AZURE

Adi Gumelar[1], Adam Mukharil Bachtiar[2]

[1,2]Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur No. 102-116, Bandung 40132

E-mail : gumelar.adi@gmail.com[1], adammb@outlook.com[2]

ABSTRAK

Tidak semua informasi di internet bermanfaat terutama bagi anak sehingga hal ini sangat mengkhawatirkan untuk orang tua. Dodo Kids Browser membantu orang tua dalam memonitoring aktivitas anak ketika menggunakan internet. Dalam penggunaannya, aplikasi ini harus dapat mengelola data seperti data orang tua, data pencarian anak, sampai harus dapat mengelola koneksi dan transfer data. Backend Dodo Kids Browser adalah sebuah perangkat lunak yang dibangun dengan menggunakan teknologi Microsoft Azure dan algoritma Boyer-Moore untuk mengatasi hal tersebut. Perangkat lunak ini akan mengelola data yang diterima dari setiap *platform* aplikasi Dodo Kids Browser. Selain mengelola data, perangkat lunak ini juga menjadi jembatan interaksi yang dilakukan di sisi orang tua ke anak dengan membantu mengirimkan data notifikasi pada *platform* aplikasi Dodo Kids Browser.

Berdasarkan hasil pengujian fungsional dapat disimpulkan bahwa perangkat lunak Backend Dodo Kids Browser sudah dapat membantu aplikasi Dodo Kids Browser dalam pengelolaan data dan menentukan kata masukkan yang baik bagi anak ketika menggunakan aplikasi ini untuk mengakses internet. Saran yang dapat diberikan adalah penambahan *storage* sehingga perangkat lunak ini dapat menampung data masukan dari pengguna lebih banyak dan pengoptimalan sistem yang meliputi notifikasi, transfer data, dan penyajian data.

Kata kunci : Backend, Microsoft Azure, Boyer-Moore, Dodo Kids Browser.

1. PENDAHULUAN

Pada bagian ini akan dipaparkan mengenai latar belakang masalah, maksud dan tujuan, dan landasan teori.

1.1 Latar Belakang Masalah

Perkembangan teknologi khususnya yang mencakup pencarian informasi di internet sudah sangat memudahkan orang-orang di semua profesi dan umur sehingga secara tidak langsung informasi di internet pun ikut berkembang. Dilansir dari situs berita *online* BBC menyebutkan bahwa pada akhir tahun 2015 ini diperkirakan sekitar 3,2 miliar penduduk akan terhubung dengan internet dan sekitar 2 miliar di antaranya tinggal di negara-negara berkembang. [1]

Indonesia adalah negara berkembang yang saat ini berpenduduk kurang lebih 250 juta jiwa dan hampir semua penduduk negara Indonesia sudah mengenal apa itu teknologi, khususnya internet. Hal ini menyebabkan keseharian setiap penduduk tidak lepas dari internet termasuk kalangan anak-anak dan remaja, hal ini berdasarkan pada hasil studi Kemenkominfo dan UNICEF di tahun 2014 yang menyatakan sebanyak 98% dari anak dan remaja tahu tentang internet dan 79,5% diantaranya adalah pengguna internet [2], dan sebanyak 52% dari 43,5 juta anak-anak dan remaja usia 10 sampai 19 tahun menggunakan *smartphone* untuk mengakses internet baik untuk mencari data dan informasi, khususnya untuk tugas-tugas sekolah, berkomunikasi dengan teman melalui media sosial, mencari konten musik dan video, dan juga situs lainnya dimana konten-konten tersebut belum tentu sesuai dan aman dikonsumsi oleh anak sementara orang tua tidak bisa memantau sepenuhnya aktivitas internet anak pada *smartphone* tersebut sehingga potensi resiko negatif dari konten tersebut menjadi tinggi [3]. Dari pemaparan beberapa fakta tersebut maka didapat masalah utama yaitu bagaimana orang tua dapat mengontrol dan mengetahui aktivitas penggunaan internet anaknya.

Dodo Kids Browser adalah aplikasi *browser* internet yang dibangun untuk mengatasi permasalahan tersebut. Aplikasi ini membantu anak untuk mencari informasi di internet baik berdasarkan ketikan kata/kalimat atau berdasarkan ucapan kata/kalimat dan juga membantu orang tua untuk mengawasi anaknya ketika mengakses internet. Berdasarkan hasil observasi yang dilakukan dengan cara mencoba layanan yang disediakan pada aplikasi Dodo Kids Browser, aplikasi ini akan melakukan pengecekan kata/kalimat yang dimasukkan apakah memiliki konten baik atau buruk bagi anak sehingga

orang tua tidak perlu terlalu cemas lagi akan anaknya ketika melakukan *browsing* di internet karena aplikasi ini akan memberikan laporan (*report*) berupa riwayat akses internet kepada orang tua dan aplikasi ini akan memberitahu orang tua situs seperti apa yang sedang dikunjungi oleh anak sekaligus aksi seperti apa yang akan diberikan oleh orang tua kepada anak berdasarkan pada pengaksesan internet yang dilakukan oleh anak. Aplikasi ini dibangun pada dua *platform* yaitu aplikasi *mobile* yang saat ini diimplementasikan pada Windows Phone dan aplikasi *extension browser* yang diimplementasikan pada *browser* internet seperti Google Chrome. Agar dapat berjalan dengan baik, kedua aplikasi ini harus dapat mengelola data yang cukup besar seperti data pengguna atau orang tua dan data kata yang dimasukkan oleh anak ketika mengakses internet. Selain itu, aplikasi ini harus memiliki respons yang baik ketika digunakan sehingga dibutuhkan juga pengelolaan koneksi dan transfer data untuk *platform* aplikasi Dodo Kids Browser.

Berdasarkan permasalahan yang telah di paparkan, maka dibangunlah sebuah perangkat lunak *backend* yang dapat membantu kedua *platform* aplikasi tersebut dalam mengelola data masukkan berupa kata/kalimat, informasi pengguna, dan interaksi aksi ke perangkat *mobile* dan *extension browser*. Aplikasi *backend* ini berupa sebuah aplikasi *web service* yang dibangun dengan menggunakan teknologi Microsoft yaitu Microsoft Azure. Selain itu, untuk dapat lebih mengoptimalkan aplikasi ketika menentukan kata masukkan baik atau buruk bagi anak, maka pada perangkat lunak ini diterapkan sebuah algoritma pencocokan kata (*string*) yaitu algoritma Boyer-Moore.

1.2 Maksud dan Tujuan

Maksud pada penelitian ini untuk membangun *backend* aplikasi "Dodo Kids Browser" dan tujuan dari penelitian ini adalah sebagai berikut:

1. Membantu aplikasi dalam mengelola data masukkan kata/kalimat.
2. Membantu admin aplikasi dalam mengelola data pengguna, dalam hal ini data orang tua.
3. Membantu aplikasi dalam mengelola koneksi dan transfer data dari/ke perangkat *mobile* dan *extension browser*.

1.3 Microsoft Azure

Microsoft Windows Azure adalah sebuah teknologi untuk *cloud computing* yang di kembangkan oleh Microsoft. Dengan menggunakan Windows Azure, pengembang dapat membuat aplikasi web, sebagai tempat penyimpanan data suatu aplikasi, sebagai jembatan koneksi antar *platform*, dan sebagainya. Berikut ini adalah komponen yang terdapat dalam Windows Azure:

1. *Cloud Applications*
2. *Data Management*
3. *Networking*
4. *Identity and Access Control*
5. *Messaging and Integration*

Dengan menggunakan teknologi Microsoft Azure, didapat beberapa keuntungan yaitu:

1. Pengembang hanya membuat perangkat lunak/sistem karena Microsoft akan melindungi sistem tersebut termasuk jika akan dilakukan pembaharuan atau *backup* data.
2. Mudah dalam menambah jumlah kapasitas penyimpanan data.
3. Terintegrasi dengan aplikasi Microsoft Visual Studio.
4. Adanya manajemen seperti pada *Application Server* yang digunakan. [4]

1.4 Web Service

Web Service adalah sebuah *software* yang dirancang untuk mendukung interoperabilitas interaksi antar mesin melalui sebuah jaringan baik berupa agregasi (pengumpulan) maupun sindikasi (penyatuan). *Web Service* melayani integrasi data dan kolaborasi informasi yang bisa diakses melalui internet oleh pengguna dengan teknologi yang dimilikinya. Pemanggilan *web service* dapat dilakukan oleh bahasa pemrograman apa saja dan dalam *platform* apa saja. Berikut ini adalah kelebihan *web Service*:

1. Lintas *platform*.
2. *Language Independent*.
3. Jembatan penghubung dengan *database* tanpa perlu *driver database* dan tidak harus mengetahui jenis DBMS yang digunakan.
4. Mempermudah proses pertukaran data.
5. Penggunaan kembali komponen aplikasi.

Dalam perkembangannya, *web service* mengenal dua metode yang berorientasi pada layanan dan sumber daya informasi, yaitu metode SOAP (*Simple Object Access Protocol*) dengan proses layanannya memiliki tiga komponen utama, yaitu:

1. *Service Provider*
2. *Service Requester*
3. *Service Broker*

Serta komponen pendukung yaitu:

1. XML.
2. SOAP-XML (terdiri atas *header* dan *body*).
3. WSDL.
4. UDDI.

Dan metode REST (*REpresentational State Transfer*) yang didasari oleh empat prinsip utama teknologi, yaitu:

1. *Resource identifier through Uniform Resource Identifier* (URI).
2. *Uniform Interface*, yaitu sumber daya CRUD menggunakan operasi PUT, GET, POST, dan DELETE.
3. *Self-descriptive messages*, yaitu sumber daya tidak terikat sehingga dapat mengakses konten

HTML, XML, PDF, JPEG, *plain text*, *meta data*, dan lainnya.

4. *Stateful interactions through hyperlinks*.

Metode REST ini lebih sederhana karena menggunakan format standar (HTTP, HTML, XML, URI, MIME), dan jika diperlukan untuk proses pertukaran data, maka konten hasil eksekusi *web service* dapat diolah dalam format teks (XML atau HTML) dengan menggunakan utilitas komunikasi data berupa koneksi *socket* protokol HTTP yang tersedia pada pustaka komunikasi bahasa pemrograman. [5]

1.5 JSON (JavaScript Object Notation)

JavaScript Object Notation adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 – Desember 1999.

Struktur dan bentuk JSON terdiri dari:

1. **Objek/*Object*** (pasangan nama/nilai yang tidak terurutkan). Dalam objek tersebut terdapat kurung kurawal buka untuk mengawali dan diakhiri oleh kurung kurawal tutup ({ }), dalam kurung kurawal tersebut terdapat nama yang diikuti dengan titik dua (:) dan setiap pasangan nama/nilai dipisahkan oleh koma (,).
2. **Larik/*Array***, kumpulan nilai yang terurutkan. Larik dimulai dengan kurung kotak buka dan diakiri dengan kurung kotak tutup ([ ]) dan setiap nilai dipisahkan oleh koma (,).
3. **Nilai/*Value***, dapat berupa sebuah *string* dalam tanda kutip ganda (“ “) atau angka, atau nilai *true*, *false*, dan *null*, atau objek atau larik. Struktur-struktur tersebut dapat disusun bertingkat.
4. **String**, adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Dalam string dapat digunakan *backslash escapes* “\” untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string* dan *string* sangat mirip dengan *string* C atau Java.
5. **Angka**, nilai yang sangat mirip dengan angka di C atau Java, kecuali format Oktal dan Heksadesimal tidak digunakan. Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detail *encoding* yang secara lengkap dipaparkan oleh bahasa pemrograman yang digunakan. [6]

1.6 API (Application Programming Interface)

Application Programming Interface adalah sekumpulan perintah, fungsi, komponen dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemrograman tertentu yang dapat digunakan oleh pengembang/*programmer* saat membangun perangkat lunak. Pada API terdapat fungsi-fungsi/perintah untuk menggantikan bahasa yang digunakan dalam *system call* dengan bahasa yang lebih mudah dimengerti oleh *programmer*.

Kelebihan dari API adalah:

1. Profitabilitas, *programmer* yang menggunakan API dapat menjalankan programnya dengan sistem operasi mana saja asalkan sudah ter-*install* API tersebut.
2. Mudah Dimengerti, bahasa yang digunakan lebih terstruktur dan mudah dimengerti oleh *programmer* daripada bahasa *system call*. [7]

1.7 Algoritma Boyer-Moore

Algoritma Boyer-Moore diciptakan oleh R.M Boyer dan J.S Moore. Algoritma ini terkenal karena banyak diterapkan pada kasus pencocokan string yang cepat dalam kinerjanya. Algoritma Boyer-Moore melakukan pencocokan karakter dimulai dari kanan ke kiri. Karakter paling kanan pada pola merupakan karakter pertama yang akan dicocokkan dengan teks. Algoritma ini mempunyai dua fase, yaitu fase preprocessing yang terdapat dua buah fungsi untuk menggeser pola ke arah kanan yaitu:

1. *Good-Suffix-Shift*, fungsi ini disimpan ke dalam tabel bmGs (boyer-moore Good suffix) berukuran m+1.
2. *Bad-Character-Shift*, fungsi ini disimpan ke dalam sebuah tabel bmBc (boyer-moore Bad character) yang berukuran n.

Dan fase pencarian string tersebut.

Pembentukan tabel bmBc dan bmGs mempunyai kompleksitas waktu O(m+n) dan kompleksitas ruang O(m+n). Kompleksitas waktu untuk fase pencarian adalah O(mn). [8]

2. ISI PENELITIAN

Pada bagian ini dipaparkan penjelasan mengenai analisis masalah, analisis penerapan algoritma Boyer-Moore, lingkungan implementasi, dan pengujian.

2.1 Analisis Masalah

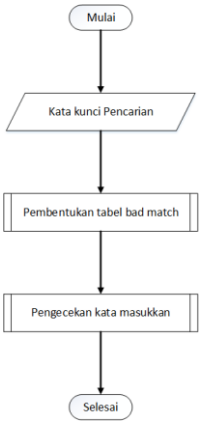
Permasalahan yang didapat ketika penggunaan aplikasi Dodo Kids Browser adalah sebagai berikut:

1. Belum ada tempat pengelolaan data masukkan berupa kata/kalimat untuk aplikasi Dodo Kids Browser.

2. Belum ada tempat pengelolaan data pengguna, dalam hal ini data orang tua untuk aplikasi Dodo Kids Browser.
3. Belum ada tempat pengelolaan koneksi dan transfer data antar *platform* Dodo Kids Browser.

2.2 Analisis Penerapan Algoritma Boyer-Moore

Penggunaan algoritma Boyer-Moore dilakukan untuk menentukan apakah kata kunci yang digunakan oleh anak ketika mengakses internet merupakan kata yang baik atau buruk bagi anak. Tahapan penggunaan algoritma Boyer-Moore dapat dilihat pada Gambar 1:

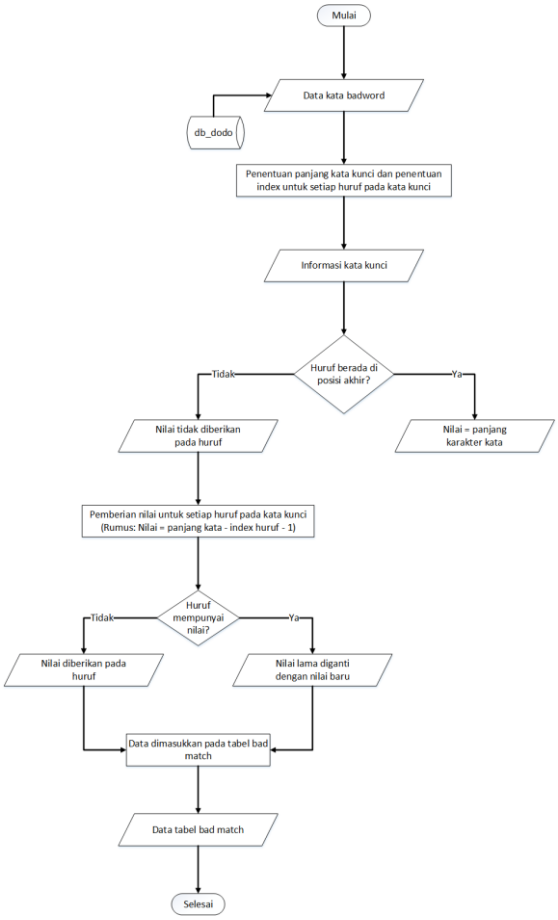


Gambar 1. Flowchart Algoritma Boyer-Moore

Pada Gambar 1 dapat dilihat bahwa proses yang digunakan untuk melakukan pencocokan kata kunci pencarian anak dengan kumpulan kata buruk dalam *database* harus melalui dua prosedur yaitu:

1. Prosedur pembentukan tabel *bad match*

Pada prosedur ini dilakukan proses pembentukan tabel *bad match* atau tabel *pattern* untuk setiap kata yang tersimpan di *database*. Tabel *bad match* digunakan untuk mencari kata buruk yang didapat pada suatu kata kunci pencarian. Proses pada prosedur ini dapat dilihat pada Gambar 2.



Gambar 2. Flowchart prosedur pembentukan tabel bad match

Sebagai contoh, misalkan terdapat kata "BASTARD" pada *database* yang merupakan kata buruk. Dari kata tersebut cari panjang karakter dan bentuk *index* untuk setiap hurufnya seperti yang terlihat pada Tabel 1.

Tabel 1. Informasi kata kunci

Kata Kunci	Keterangan							
BASTARD	Length = 7							
	Index =	B	A	S	T	A	R	D
		0	1	2	3	4	5	6

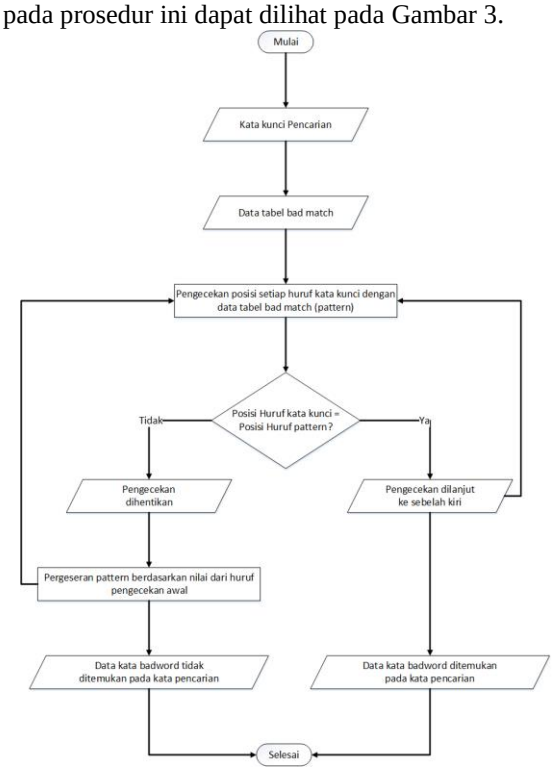
Setelah diketahui informasi kata kunci maka bangun tabel *bad match* dengan proses sebagai berikut pada Tabel 2.

Tabel 2. Proses pembentukan tabel bad match

Proses 1								
Kata Kunci	**Keterangan Proses**							
BASTARD	B = 7-0-1=6							
	Hasil							
	Letter	B	A	S	T	R	D	*
	Value	6						
Proses 2								
Kata Kunci	**Keterangan Proses**							
BASTARD	A = 7-1-1=6							
	Hasil							
	Letter	B	A	S	T	R	D	*
	Value	6	5					
Proses 3								
Kata Kunci	**Keterangan Proses**							
BASTARD	S = 7-2-1=6							
	Hasil							
	Letter	B	A	S	T	R	D	*
	Value	6	5	4				
Proses 4								
Kata Kunci	**Keterangan Proses**							
BASTARD	T = 7-3-1=3							
	Hasil							
	Letter	B	A	S	T	R	D	*
	Value	6	5	4	3			
Proses 5								
Kata Kunci	**Keterangan Proses**							
BASTARD	A = 7-4-1=2, (Value 2 dimasukkan untuk menggantikan value A sebelumnya yaitu 5)							
	Hasil							
	Letter	B	A	S	T	R	D	*
	Value	6	2	4	3			
Proses 6								
Kata Kunci	**Keterangan Proses**							
BASTARD	R = 7-5-1=1							
	Hasil							
	Letter	B	A	S	T	R	D	*
	Value	6	2	4	3	1		
Proses 7								
Kata Kunci	**Keterangan Proses**							
BASTARD	D = 7, (D adalah huruf terakhir)							
	Hasil							
	Letter	B	A	S	T	R	D	*
	Value	6	2	4	3	1	7	7

2. Prosedur pengecekan kata masukkan

Pada prosedur ini dilakukan proses pengecekan kata masukkan dari *platform mobile* atau *extension browser* dengan data tabel *bad match* untuk mengetahui apakah kata masukkan tersebut termasuk kata yang buruk bagi anak atau baik bagi anak. Proses pada prosedur ini dapat dilihat pada Gambar 3.



Gambar 3. Flowchart prosedur pengecekan kata masukkan

Sebagai contoh, misalkan kata kunci yang dimasukkan adalah "THEBASTARDBOYS" akan dibandingkan dengan data tabel *bad match* kata "BASTARD" dalam *database*.

Berikut ini adalah proses pengecekan kata masukkan "THEBASTARDBOYS" dengan kata "BASTARD" yang sudah dibentuk tabel *bad match* pada Tabel 3.

Tabel 3. Proses pengecekan kata masukkan

Proses 1														
Kata Masukkan	T	H	E	B	A	S	**T**	A	R	D	B	O	Y	S
Pattern	B	A	S	T	A	R	**D**							
Keterangan	T ≠ D, maka *pattern* bergeser sebanyak 3 kali (Bobot nilai T = 3)													
Proses 2														
Kata Masukkan	T	H	E	**B**	**A**	**S**	**T**	**A**	**R**	**D**	B	O	Y	S
Pattern				**B**	**A**	**S**	**T**	**A**	**R**	**D**				
Keterangan	D = D R = R A = A T = T S = S A = A B = B													

	Karena posisi huruf pada kata masukkan = posisi huruf pada *pattern*, maka kata masukkan mengandung kata yang buruk bagi anak.

2.3 Lingkungan Implementasi

Sistem yang dibangun adalah sebuah *web service* yang bertugas untuk mengelola data yang dibutuhkan aplikasi Dodo Kids Browser termasuk mengelola koneksi dan transfer data antar *platform* aplikasi Dodo Kids Browser. Sistem diimplementasikan pada Microsoft Azure dengan paket *Standard*, berikut adalah spesifikasi dari paket *Standard* pada Microsoft Azure pada Tabel 4.

Tabel 4. Spesifikasi Paket Standard Microsoft Azure

Item	Spesifikasi
Processor	1 Core
Memory	1.75 GB RAM
Storage	50 GB
Custom Domains/SSL	5 SNI, 1 IP
Auto Scale	Up to 10 Instances
Backup Plan	Daily
Web App Staging	5 Slots
Geo Availability	Traffic Manager

2.4 Pengujian

Pada sistem yang dibangun akan dilakukan pengujian fungsional dengan menggunakan pengujian *black box* untuk mengetahui apakah fungsionalitas Backend Dodo Kids Browser sudah dapat membantu *platform* aplikasi Dodo Kids Browser. Dari pengujian yang sudah dilakukan, berikut ini adalah kesimpulan hasil pengujian dari Backend Dodo Kids Browser pada Tabel 5.

Tabel 5. Kesimpulan hasil pengujian fungsional

No	Fungsional	Hasil Pengujian
1	Login	Fungsional dapat berjalan dan dapat digunakan untuk setiap *platform* aplikasi Dodo Kids Browser.
2	Tambah Data User	
3	Ubah Data User	
4	Tambah Data Kata	
5	Pencocokan Kata	
6	Penyajian Data Report	Fungsional dapat berjalan dan dapat digunakan untuk *platform mobile* aplikasi Dodo Kids Browser.
7	Penyajian Data Chart	

3. PENUTUP

Pada bagian penutup ini akan dipaparkan mengenai kesimpulan dari hasil penelitian dan saran yang dipaparkan mengenai penelitian.

3.1 Kesimpulan

Berdasarkan hasil pengujian Backend Dodo Kids Browser maka dapat diperoleh kesimpulan sebagai berikut:

1. Backend Dodo Kids Browser sudah dapat mengelola data masukan yang didapat dari *platform mobile* dan *platform extension browser* aplikasi Dodo Kids Browser.
2. Backend Dodo Kids Browser sudah dapat mengelola data pengguna yang dalam hal ini data orang tua pengguna aplikasi Dodo Kids Browser.
3. Backend Dodo Kids Browser sudah dapat mengelola koneksi dan transfer data antar *platform* aplikasi Dodo Kids Browser.
4. Backend Dodo Kids Browser sudah dapat mengimplementasikan algoritma Boyer-Moore dengan baik ketika melakukan pencocokan kata masukkan dari *platform* aplikasi Dodo Kids Browser dengan kata yang terdapat di *database*.

3.2 Saran

Aplikasi Backend Dodo Kids Browser yang dibangun perlu dilakukan pemeliharaan dan pengembangan seiring dengan perkembangan *platform* aplikasi Dodo Kids Browser. Adapun saran-saran terhadap pemeliharaan dan pengembangan aplikasi Backend Dodo Kids Browser ke depan adalah sebagai berikut:

1. Penambahan *storage* untuk aplikasi Backend Dodo Kids Browser di Microsoft Azure agar dapat menampung data masukan dari pengguna lebih banyak.
2. Pengoptimalan notifikasi yang dikirimkan ke *platform* aplikasi Dodo Kids Browser.
3. Pengoptimalan transfer data dan penyajian data yang dibutuhkan *platform* aplikasi Dodo Kids Browser.

DAFTAR PUSTAKA

[1] BBC, “Hampir separuh warga dunia terhubung internet akhir 2015,” BBC Indonesia, 27 Mei 2015. [Online]. Available: http://www.bbc.com/indonesia/majalah/2015/05/150526_majalah_internet. [Diakses 30 Juni 2015].

[2] A. Panji, “Hasil Survei Pemakaian Internet Remaja Indonesia,” kompas.com, 19 Februari 2014. [Online]. Available: http://tekno.kompas.com/read/2014/02/19/1623250/Hasil.Survei.Pemakaian.Internet.Remaja.Indonesia. [Diakses 30 Juni 2015].

[3] U. Indonesia, “Studi Terakhir: Kebanyakan Anak Indonesia sudah online, namun masih banyak yang tidak menyadari potensi resikonya.,” UNICEF Indonesia, 18 Februari 2014. [Online]. Available: http://www.unicef.org/indonesia/id/. [Diakses 30 Juni 2015].

[4] V. A. Gandhi dan C. K. Kumbharana, "Compatarive Study of Amazon EC2 and Microsoft Azure Cloud Architecture," *International Journal of Advanced Networking Applications (IJANA),* pp. 117-123.

[5] E. Sutanta dan K. Mustofa, "Kebutuhan Web Service untuk Sinkronisasi Data antar Sistem Informasi dalam E-Gov di Pemkab Bantul Yogyakarta," *JURTIK - STMIK BANDUNG,* pp. 1-11, 2012.

[6] "JSON," [Online]. Available: http://json.org/. [Diakses 30 Juni 2015].

[7] OasisBiru, "=::.Hendry's Blog.::=^^[[>UNIVERSITAS GUNADARMA<]]=," 25 Januari 2013. [Online]. Available: http://oasisbiru.blogspot.com/2013/01/api-application-programming-interface.html. [Diakses 17 Agustus 2015].

[8] A. Minandar, A. Tanoto dan D. Tanadi, "Aplikasi Algoritma Pencarian String Boyer-Moore pada Pencocokan DNA," pp. 1-3.