

2. LANDASAN TEORI

2.1. Sejarah Singkat aplikasi Telegram

Telegram adalah salah satu aplikasi *IM* yang dikembangkan oleh Nikolai Durov dan Pavel Durov. Aplikasi ini pertama kali diluncurkan versi *alpha*-nya pada bulan Agustus 2013. Telegram memiliki 6 aplikasi *native* untuk beberapa *platform*, diantaranya adalah; iOS, Android, Windows Phone, *Desktop* (Mac dan Windows), dan Telegram versi *web*. Pada bulan Desember 2015, tercatat ada 1.5 juta pengguna yang aktif.

Telegram merupakan aplikasi yang menggunakan konsep *open source*, yang berarti kode sumber mereka tidak dimiliki oleh siapapun atau lembaga manapun. Karena tidak dimiliki oleh siapapun maka kode sumbernya dapat dengan sangat mudah tersebar dan tersedia secara bebas. Sehingga siapapun dapat bekerja sama untuk merubah dan menggantinya.

2.2. Keunggulan utama Telegram

Keunggulan yang ditawarkan ke pengguna Telegram adalah sinkronisasi *real-time* pada beberapa perangkat Telegram, hal ini bisa tercapai karena sistem Telegram telah menggunakan sistem *cloud*, yang berarti semua pesan (baik berbentuk pesan atau file) semuanya langsung diunggah ke sistem dan perangkat dengan akun yang sama akan bisa melihat semua pesan yang dikirim di perangkat lainnya. Sehingga pengguna dapat menggunakan beberapa perangkat untuk menggunakan Telegram. Tidak hanya itu, Telegram juga menawarkan kecepatan dalam mengirimkan pesan. Secara *default* semua pesan dan *file* yang terkirim menggunakan Telegram akan langsung terenkripsi. Fitur *Secret Chat* pertama kali diluncurkan oleh Telegram sebelum diadopsi dan diimplementasikan pada aplikasi IM lainnya. Fitur ini adalah fitur untuk menghapus pesan obrolan yang telah terkirim oleh pengguna, sehingga pesan tersebut akan benar-benar hilang jika waktu pesan tersebut telah habis. Telegram

juga tidak membatasi ukuran file yang akan dikirim, juga tidak membatasi tipe file yang akan dikirim. Fitur *reply* juga menjadi salah satu keunggulan yang ditawarkan Telegram yang tidak terdapat di IM lainnya, dengan fitur ini pengguna dapat membalas pesan yang telah terkirim di *chat window* sehingga pengguna tidak perlu lagi menjelaskan konteks pembicaraan lagi. Sebuah grup pembicaraan di Telegram dapat menampung 5000 buah pengguna.

2.3. Telegram API

Telegram menyediakan API yang memiliki 2 bentuk, pertama adalah bentuk klien Telegram yang bisa dirubah sesuai keinginan. Pengembang Telegram menyediakan *Source Code* klien Telegram mereka, sehingga pembuat klien Telegram tidak harus membuat klien Telegram yang benar-benar baru. Pengembang pihak ketiga dapat menggunakan *platform* yang disediakan oleh Telegram untuk digunakan oleh klien Telegram yang telah dibuat. Pengembang Telegram juga menyediakan contoh-contoh kode Telegram yang telah siap digunakan. Namun pengembang klien Telegram harus mendaftarkan klien Telegram yang telah mereka buat ke *web* Telegram agar dapat bisa digunakan oleh masyarakat luas.

Fitur Telegram yang kedua adalah disediakannya API Telegram yang berbentuk *Bot*. *Bot* adalah akun Telegram yang tidak membutuhkan nomer telepon untuk membuat akun. *Bot* juga tidak membutuhkan manusia untuk membalas chat dari para pengguna IM Telegram. Hal ini memungkinkan pengguna Telegram untuk mendapatkan pelayanan setiap saat, dan tidak terbatas oleh ruang dan tempat selama ada koneksi internet. *Bot* merupakan salah satu fitur yang ditawarkan oleh *IM* Telegram karena belum ada *IM* yang menawarkan dan membiarkan pengguna pihak ketiga untuk secara bebas menggunakan API yang mereka sediakan untuk membuat *bot* yang bebas dirancang dan dibuat oleh siapapun. Beberapa *bot* yang telah selesai dirancang dan siap digunakan adalah rancangan dari penyedia konten di internet, seperti

contohnya YouTube, IMDB, dan sebagian ada *bot* yang dirancang dari Telegram sendiri.

```
<?php  
define('TOKEN', '200942287:AAGUSfMA7lSmmGsd9ltqdOKAjRqlobTZphM');
```

Gambar 2.1 PHP *define token bot* Telegram

Disetiap *bot* baru, *bot* bernama *BotFather* akan memberikan token untuk yang memperbolehkan skrip PHP dari perangkat pengembang untuk meminta server Telegram untuk menjalankan *script* PHP yang telah dirancang. Maka kerahasiaan *token* yang diberikan oleh *BotFather* menjadi salah satu hal penting. Karena setiap *bot* akan memiliki token yang berbeda. Jika token tidak sesuai maka *request* tidak akan terpenuhi. Token akan berbentuk seperti Gambar 2.1

2.4. Metode pengiriman yang Disediakan oleh Telegram *bot* API

Huruf kapital tidaklah mempengaruhi metode yang digunakan oleh Telegram *Bot* API. Semua objek yang dikirim menggunakan JSON, jika berhasil maka balasan dari server Telegram juga akan berupa objek JSON. Telegram *Bot* API juga mendukung penggunaan metode *GET* dan *POST* dari HTML Steam. Beberapa bentuk data yang dapat diterima oleh server Telegram adalah sebagai berikut:

- *sendMessage* – perintah untuk mengirim pesan dari server ke pengguna, atau dari pengguna ke server
- *forwardMessage* – fitur ini digunakan untuk meneruskan pesan dalam bentuk apapun ke siapapun
- *sendPhoto* – perintah untuk mengirimkan file foto ke pengguna atau ke server Telegram
- *sendAudio* – cara ini digunakan agar *Bot* dapat mengirimkan file audio ke server atau ke pengguna. Agar file bisa langsung dimainkan bentuk file audio harus dalam format mp3

- *sendDocument* – cara ini digunakan untuk mengirimkan file apapun ke server Telegram atau dari server ke pengguna. *Bot* dapat menyimpan file sebesar 50 megabytes
- *sendSticker* – Cara ini digunakan agar *bot* dapat mengirimkan gambar sticker ke pengguna, untuk pengiriman dari pengguna ke server Telegram juga menggunakan cara yang sama
- *sendVideo* – cara ini digunakan agar file video dapat dikirim ke *bot* baik dari *bot* ke pengguna. Format yang didukung oleh klien Telegram adalah video berbentuk mp4.
- *sendVoice* – berbeda dengan file audio cara ini digunakan untuk mengirim rekaman suara ke pengguna. *Bot* dapat mengirim file rekaman suara sebesar 50 megabyte (batas bisa berubah sewaktu-waktu).
- *sendLocation* – cara ini digunakan untuk mengirim lokasi di peta, jika berhasil maka pesan yang dikirim akan dibalas.
- *sendVenue* – berbeda dengan *sendLocation*, cara ini akan mengirimkan lokasi yang telah terdaftar di peta, *sendLocation* hanya akan mengirimkan koordinat di peta.
- *sendContact* – cara ini dapat digunakan agar baik pengguna atau server dapat mengirimkan kontak di perangkat ponsel pintar atau dari server.
- *sendChatAction* – cara ini digunakan untuk menunjukkan apa yang sedang dilakukan *bot* pada saat perintah diterima. Jika *Bot* akan membalas dengan file gambar, maka dibawah nama *bot* akan terlihat tulisan “Retrieving Image please wait...”.
- *getUserProfilePhotos* – untuk mengambil sejumlah profile foto pengguna Telegram cara ini bisa digunakan, balasan akan berupa file foto.
- *getFile* – Cara ini disediakan oleh Telegram untuk mengambil file dan disiapkan untuk didownload oleh pengguna, file yang bisa diambil hanya memiliki batas ukuran 20 megabyte.

- *kickChatMember* – cara ini dapat digunakan untuk menghapus dan mengeluarkan anggota dari suatu kelompok chat. Agar dapat berfungsi dengan baik maka *bot* harus menjadi admin kelompok chat tersebut.
- *leaveChat* – cara ini digunakan untuk agar *bot* dapat meninggalkan kelompok chat tersebut, jika berhasil maka balasan *True* akan dikirim.
- *unbanChatMember* – cara ini digunakan untuk memperbolehkan suatu user untuk kembali masuk kedalam kelompok chat. Jika memberikan perintah ini anggota yang telah di keluarkan tidak langsung masuk ke kelompok *chat*, anggota yang sudah keluar harus masuk melalui *join link* lagi.
- *getChat* – cara ini digunakan untuk mendapatkan perbaruan informasi yang paling baru untuk chat, contohnya username pengguna, nama kelompok atau channel, dll.
- *getChatAdministrator* – cara ini digunakan untuk menampilkan semua admin yang berada di suatu kelompok chat, jika berhasil maka *bot* akan membalas dengan *array* yang berisi anggota *chat*.
- *getChatMember* – cara ini digunakan untuk mendapatkan informasi tentang pengguna yang berada di dalam kelompok chat, jika berhasil *bot* akan membalas dengan objek *ChatMember*.
- *answerCallbackQuery* – cara ini digunakan untuk mengirimkan perintah yang berada di *inline keyboard*, jika berhasil maka *bot* akan mengirimkan *true*.

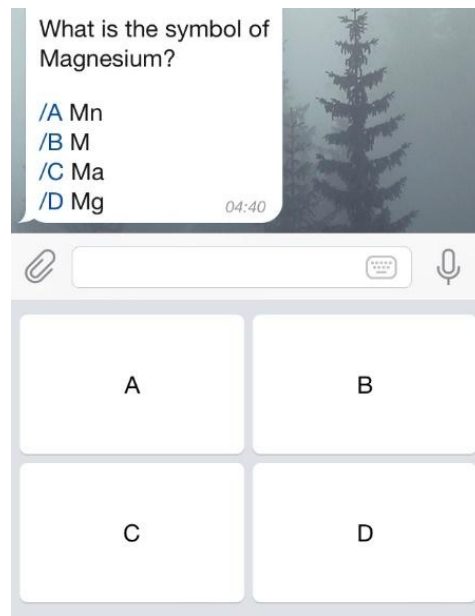
2.5. Fitur Bot Telegram

Pengembang aplikasi Telegram memberikan *API* pada pengguna yang ingin menggunakan fitur ini. *Developer* pihak ketiga hanya perlu mengunggah *script* yang telah mereka buat untuk digunakan oleh pengguna lainnya. *Bot* berbeda dengan akun Telegram karena *bot* tidak menggunakan manusia untuk mengoperasikannya, *Bot* dapat di kerjakan menggunakan bahasa pemrograman populer di pemrograman *web* seperti PHP dan JSON. Bahasa pemrograman

tidaklah terbatas pada PHP, karena pada intinya semua bahasa pemrograman bisa dieksekusi oleh server Telegram, bagian yang paling penting adalah cara *memparsing* data dari *Source Code* ke JSON dan PHP. *Bot* juga dapat diimplementasikan dengan berbagai *database*. Tidak terbatas *database* apa yang akan digunakan, selama data dari *database* dapat dibaca dan *database* bisa digunakan dengan bahasa PHP, maka *database* layak digunakan.

2.5.1. Custom Keyboard

Bot Telegram juga mendapatkan keunggulan untuk menggunakan keyboard yang dapat dirancang khusus untuk penggunaannya. Berbeda dengan jika pengguna menggunakan Telegram untuk bertukar pesan. *Keyboard* yang digunakan oleh *bot* bisa dirancang oleh perancang *bot* sehingga pengguna tidak harus mengetikkan perintah yang ingin dikirimkan ke *bot* Telegram. *Custom keyboard* yang disediakan oleh Telegram akan mempermudah interaksi antara pengguna dan *bot*. Dapat dilihat pada Gambar 2.2.



Gambar 2.2 Contoh penggunaan *custom keyboard*

2.5.2. Perintah untuk *bot*

Perintah *bot* dapat diberikan dengan awalan “/”, dengan memberikan “/” maka *bot* akan langsung mengeluarkan beberapa opsi untuk menerima pesan yang akan dikirim. Dapat dilihat pada Gambar 2.3. Dengan memilih salah satu opsi tersebut kata yang dipilih akan langsung tersalin ke *textbox* di perangkat pengguna. Telegram juga akan menampilkan deskripsi singkat tentang opsi yang tertera pada layar perangkat. Secara umum perintah yang diberikan kepada *bot* akan memiliki struktur seperti berikut;

```
/[kata kunci perintah] [argument atau query]
```

Gambar 2.3 Contoh struktur perintah *Bot* Telegram

2.5.3. Perintah Global

Sebuah *bot* Telegram pasti memiliki 3 perintah umum, antara lain adalah:

- */start* – sebelum memulai menerima perintah *bot* perlu menerima perintah *start* begitu perintah ini diterima maka *bot* akan mengeluarkan pesan balasan. Biasanya pesan akan berupa sapaan untuk pengguna dan sedikit penjelasan apa saja yang bisa dilakukan *bot*, beserta perintah-perintah apa saja yang bisa diterima oleh *bot* tersebut.
- */help* – perintah ini digunakan untuk membantu pengguna menjelaskan apa saja yang bisa dilakukan *bot* secara lebih mendetail.
- */settings* – perintah ini bisa digunakan bila perancang *bot* mengizinkan pengguna *bot* untuk mengganti beberapa pengaturan yang telah diatur oleh perancang. Jika tidak diizinkan maka perintah ini tidak akan bisa digunakan oleh pengguna biasa.

2.6. Kelemahan *Bot* Telegram

Kelemahan *Bot* Telegram terletak pada masalah kecepatan. Telegram menyediakan integrasi dengan *web* hosting bernama *Webhooks*. Telegram mendukung 2 cara untuk mendapatkan update untuk *bot*, yang pertama adalah

menggunakan fungsi API *getUpdate* dan yang kedua menggunakan *Webhooks*. Kelemahan menggunakan *getUpdate* yang menggunakan metode *Long-polling* adalah perbedaan kecepatan yang cukup jauh jika dibandingkan dengan menggunakan layanan *Webhooks*. Tidak hanya itu metode *Long-polling* juga mempunyai kelemahan yang cukup krusial yaitu, jika komputer yang berisi skrip PHP tidak terhubung ke server Telegram maka *bot* yang bersangkutan tidak dapat berfungsi. Namun masalah ini dapat diatasi dengan menyimpan *script* PHP pada sebuah server yang memiliki *downtime* yang kecil.

2.7. *Hypertext Preprocessor*

PHP adalah singkatan dari *Hypertext preprocessor*. Bahasa yang populer digunakan didalam dunia *web*. PHP sangat populer digunakan karena PHP dapat digunakan seiringan dengan *Hypertext Mark Up Language* (HTML) sehingga *script* PHP dapat digunakan dalam dokumen HTML. Penggunaannya juga cukup singkat, pengguna hanya butuh mengetikkan “<?php” (seperti yang terlihat pada Gambar 2.1 untuk mendeklarasikan bahwa konten setelahnya adalah konten bahasa PHP, *script* PHP diakhiri dengan tanda “?>”.

Fungsi utama bahasa PHP jika digabungkan dengan HTML adalah *script* dijalankan di server, bukan di komputer klien. Maka klien akan menerima hasil dari *script* PHP tersebut tanpa mengetahui apa yang dilakukan *script* tersebut. Bahasa yang cukup sederhana juga menjadi salah satu keunggulan bahasa PHP, banyak implementasi PHP populer digunakan.

Penggunaan *script* PHP juga banyak digabungkan dengan beberapa pemrograman database, contohnya XAMPP, MySQL, dll. Penggunaan ini disebut dengan *Server Style Scripting*. Penggunaan ini memungkinkan pengguna untuk mengakses suatu database menggunakan *web* browser tanpa menggunakan aplikasi tambahan. PHP juga dapat dijalankan tanpa menggunakan *web* browser, pemanfaatan ini digunakan oleh beberapa *operating system* (OS) untuk menjalankan beberapa tugas sederhana. Karena bahasa PHP adalah bahasa yang bisa digabungkan dengan HTML dan beberapa database server maka bahasa PHP

tidak terbatas oleh OS, semua OS yang banyak digunakan dapat menerima bahasa PHP untuk dijalankan.

PHP juga mendukung pemakaian untuk *service* lainnya, contohnya untuk protokol *Lightweight Directory Access Protocol* (LDAP), *Internet Messaging Access Protocol* (IMAP), *Hyper Text Transfer Protocol* (HTTP), dan masih banyak lagi. PHP juga mendukung pengoperasian dengan bahasa pemrograman *web* lainnya. Contoh penggunaan PHP dapat dilihat pada Gambar 2.4,

```
<?php  
echo "My first PHP script!";  
?>
```

Gambar 2.4 Contoh penggunaan PHP

Sumber: www.w3schools.com

2.8. JSON

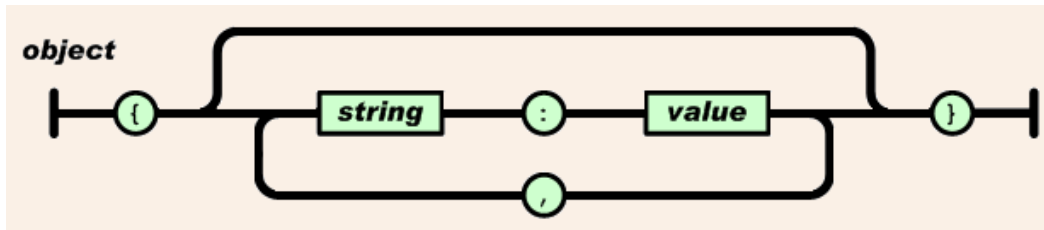
JSON adalah singkatan dari *JavaScript Object Notation*. JSON banyak digunakan untuk menukarkan data pada beberapa bahasa pemrograman *web*. Banyak digunakan untuk *parsing* data. Kemudahan untuk dibaca dan dimengerti juga menjadi salah satu alasan mengapa JSON sangat banyak digunakan. JSON juga menjadi salah satu derivatif dari bahasa Java yang sangat populer. Pemanfaatan JSON banyak digunakan dengan bahasa pemrograman seperti; C, C++, C#, Java, *JavaScript*, Perl, Python, dll.

Struktur utama JSON memiliki 2 bentuk (www.json.org, 2016, paragraph 2) yaitu, beberapa koleksi dari beberapa nama seperti *object*, *members*, *pair*. Bentuk kedua adalah nilai-nilai yang tersusun secara berurutan yang bisa berwujud *array*, *elements*, dan *value*. Manfaat utama menggunakan JSON dalam *bot* Telegram adalah untuk *memparsing* data dari bahasa PHP ke server

Telegram, ini memungkinkan *bot* Telegram untuk ditulis dengan bahasa pemrograman apapun.

JSON memiliki keunggulan karena format penulisan yang ditawarkan. Keunggulan yang ditawarkan bahasa JSON adalah kebebasan terhadap suatu bahasa pemrograman. Hal ini memberikan *web developer* untuk memberikan suatu tipe data yang bisa langsung diterima oleh bahasa pemrograman yang mereka gunakan. Bahasa JSON juga sangat mudah untuk ditulis dan dimengerti oleh manusia. Contoh beberapa format data JSON adalah sebagai berikut:

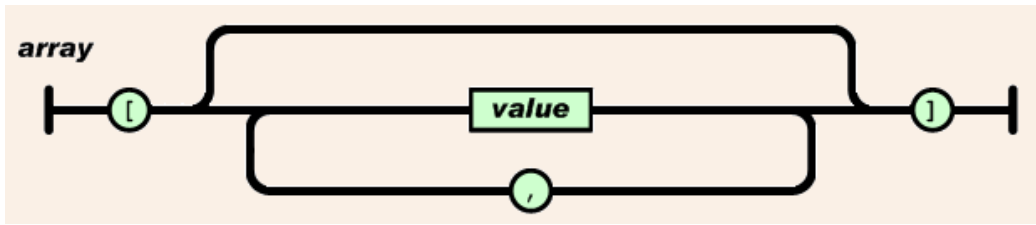
- *Object* dari suatu nama/nalai yang tidak beraturan. Penulisan objek dimulai dengan tanda “{” dan diakhiri dengan “}”. Setiap nama disertai dengan tanda “:”, dan setiap nama/nalai dipisahkan dengan tanda “,”. Ilustrasi bisa dilihat pada Gambar 2.5.



Gambar 2.5 Ilustrasi *Object* JSON

Sumber: www.json.org

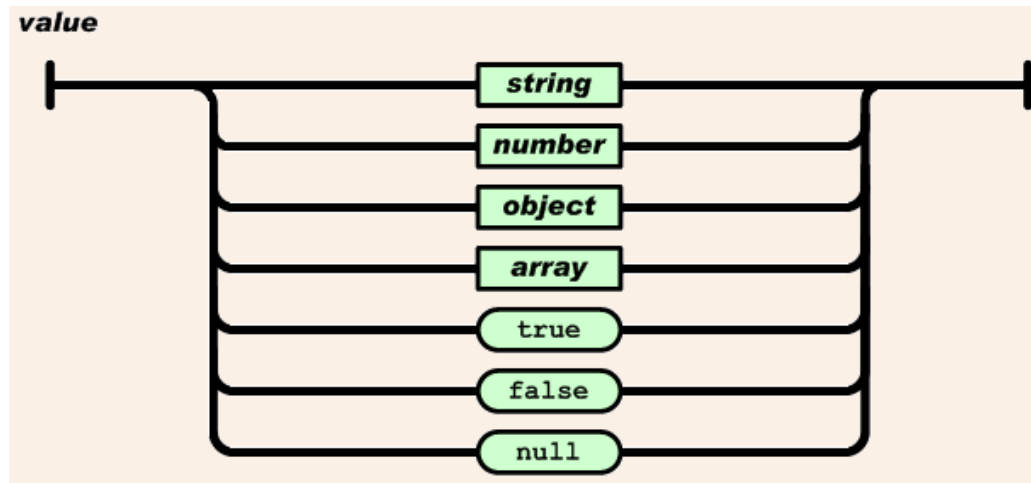
- *Array* adalah kumpulan nilai yang teratur dan berurutan. Penulisan sebuah *array* bisa dimulai dengan tanda “[” dan diakhiri dengan tanda “]”, semua nilai dipisahkan dengan menggunakan tanda “,”. Ilustrasi dapat dilihat pada Gambar 2.5.



Gambar 2.6 Ilustrasi *array* pada JSON

Sumber: www.json.org

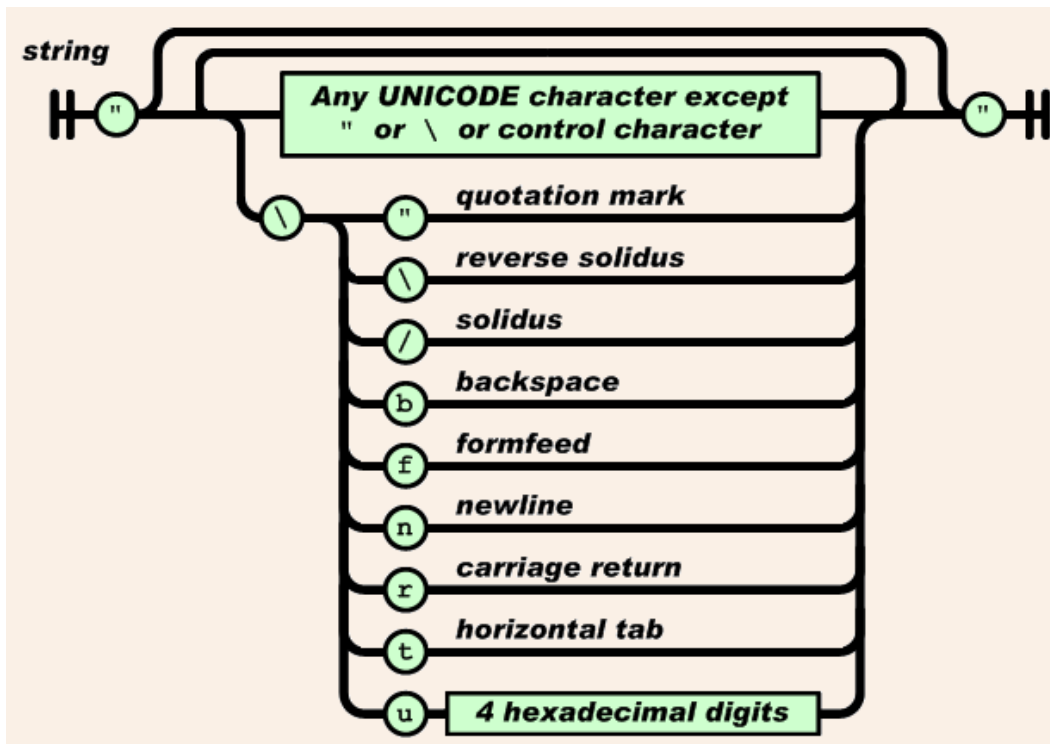
- *Value* (nilai) bisa berbentuk sebuah *string*, angka, *true*, *false*, *null*, *object*, atau sebuah *array*. Bentuk ini juga dapat ditulis dengan metode bersarang. Ilustrasi dapat dilihat pada Gambar 2.5.



Gambar 2.7 Ilustrasi *value* pada JSON

Sumber: www.json.org

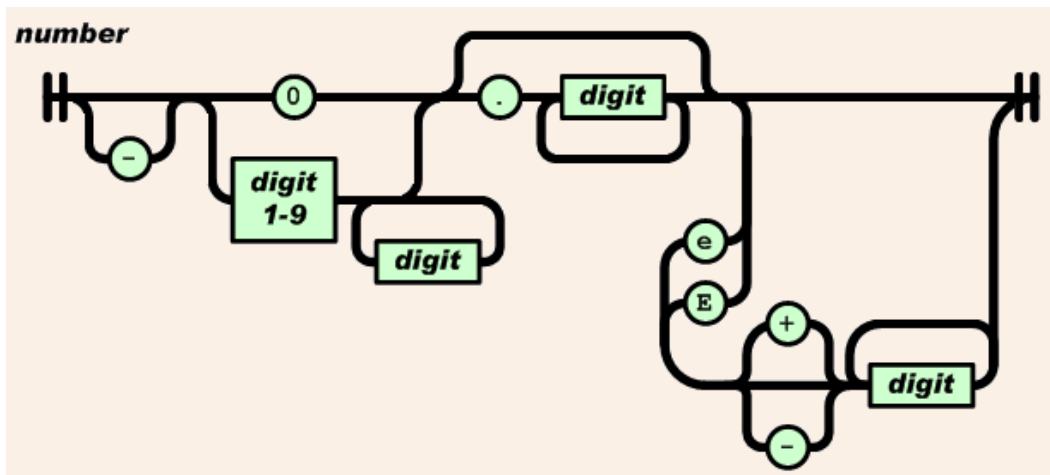
- *String* adalah suatu rangkaian urutan karakter *Unicode* yang berada di antara tanda kutip, *string* bisa menyimpan karakter kosong yang tidak terlihat. Karakter umumnya presentasikan sebagai karakter tunggal. Tipe data *string* juga ada di dalam bahasa C dan *Java*. Ilustrasi dapat dilihat pada Gambar 2.8.



Gambar 2.8 Ilustrasi *string* pada JSON

Sumber: www.json.org

- *Number* adalah tipe data numerik yang sangat mirip seperti yang terdapat di bahasa pemrograman lainnya, seperti bahasa C dan Java. Perbedaannya dalam JSON tipe bilangan *octal* dan *heksadesimal* tidak digunakan. Ilustrasi bisa dilihat pada Gambar 2.9



Gambar 2.9 Ilustrasi *number* pada JSON

Sumber: www.json.org

Contoh penggunaan JSON untuk mengisi data nama untuk pegawai bisa dilihat pada Gambar 2.10.

```

{"employees": [
  {"firstName": "John", "lastName": "Doe"},
  {"firstName": "Anna", "lastName": "Smith"},
  {"firstName": "Peter", "lastName": "Jones"}
]}

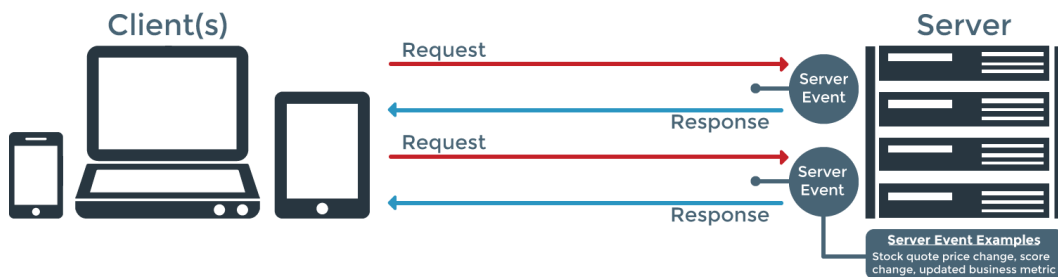
```

Gambar 2.10 Contoh penggunaan bahasa JSON

Sumber: www.w3schools.com

2.9. Metode Long polling

Ada 2 cara untuk berkomunikasi yang didukung oleh server Telegram ke *bot* mereka, diantaranya adalah metode *long polling* dan metode yang menggunakan *Webhook*. *Long Polling* adalah metode untuk menghubungkan antara server dan beberapa klien server. Cara ini populer digunakan untuk mengirimkan pembaruan dari server ke klien. Semua *browser* juga rata-rata bisa menerima protokol *long polling*. Secara singkat server membuat suatu koneksi ke klien yang bersangkutan yang secara terus menerus terhubung, maka jika ada perbaruan, server akan langsung mengirimkan data ke klien dan klien akan langsung menerima pembaruan tersebut. Proses ini akan terus diulang hingga klien selesai mendapatkan semua data dan pembaruan yang dibutuhkan. Kelemahan menggunakan metode ini adalah server harus tetap terhubung ke klien



Gambar 2.11 Ilustrasi HTML Long Polling

selama menunggu informasi yang akan dikirimkan, jika informasi belum tersedia maka koneksi antara klien dan server akan terus tetap terhubung. Metode long polling secara singkat dapat diilustrasikan pada Gambar 2.11. Jika diterapkan kepada *chat bot* maka *Source Code* harus sering menggunakan fungsi *update*. *Chat bot* juga akan memakan waktu lebih panjang untuk membaca pesan yang dikirimkan oleh pengguna. Secara *default bot* Telegram menggunakan metode long polling untuk mengirimkan pesan dan berinteraksi dengan server mereka. Penggunaan long polling juga tidak harus menggunakan *web hosting* yang berbayar sehingga pengguna tidak harus mendaftarkan pada penyedia hosting yang berbayar. Kekurangan dari menggunakan metode ini adalah jika perangkat server tidak *online* maka *bot* Telegram yang telah dirancang tidak akan jalan.

Selain itu *bot* yang telah dirancang harus sering menggunakan fungsi update untuk mendapatkan update.

2.10. MySQL

MySQL adalah sebuah *Database Management System* (DBMS). Sebuah database adalah beberapa sekumpulan data yang terstruktur dengan baik. Untuk mengakses, memanipulasi dan memproses data yang berada didalam sebuah database, diperlukan sebuah DBMS, maka dari itu penggunaan MySQL sangatlah populer. Bukan hanya sebuah DBMS, MySQL juga adalah sebuah *Relational Database Management System* (RDBMS). Perbedaan antara RDBMS dan DBMS terletak pada kemampuan suatu database untuk menyimpan data pada tabel yang berbeda. Dengan menggunakan beberapa tabel sekaligus maka kecepatan dan fleksibilitas suatu database pun bertambah.

2.10.1. Keuntungan menggunakan MySQL

Ada beberapa keuntungan dari menggunakan MySQL, diantaranya adalah:

- Mudah untuk digunakan. Bahasa yang digunakan untuk menulis didalam MySQL tidaklah jauh berbeda dengan bahasa SQL pada umumnya.
- Kecepatan yang cukup tinggi. Tujuan dari dibuatnya MySQL adalah untuk meningkatkan kecepatan.
- Gratis. Karena MySQL menggunakan lisensi *open source*, maka pengguna yang ingin menggunakannya tidak perlu membayar.
- Kompatibilitas tinggi. Banyak sistem operasi yang kompatibel dengan MySQL.

2.11. XAMPP

XAMPP merupakan salah satu distribusi *Apache*. *Apache* merupakan salah satu tipe *web server* yang menggunakan lisensi *open source*. *Apache* sangatlah kompatibel untuk beroperasi dengan sistem operasi yang menggunakan bahasa UNIX, contohnya Linux, Solaris, dan lain-lain. Tidak hanya bahasa UNIX, *web server* ini juga bisa digunakan oleh sistem operasi yang menjadi

derivatif dari sistem UNIX/POSIX, contohnya Rhapsody, BeOS, dan Windows. Dari sumber www.netcraft.com, 60% *web* server yang digunakan sekarang menggunakan *web* server Apache ini.

Filosofi nama XAMPP memiliki arti dari huruf X melambangkan *cross-platform* yaitu semua sistem operasi bisa menggunakan program ini. A berarti Apache, yaitu dasar dari program ini. M yang memiliki arti MySQL yaitu database utama yang menjadi bagian dari XAMPP. P yang pertama memiliki arti PHP, adalah suatu bahasa untuk berinteraksi dengan server. P yang kedua memiliki arti Perl, yaitu bahasa yang sering digunakan untuk membuat suatu aplikasi *web*.

2.12. *DOM Parser*

DOM Parser adalah sebuah PHP yang dapat digunakan untuk mempermudah *developer* untuk mengambil teks yang berada di suatu halaman HTML. Penggunaan *DOM Parser* dapat mengganti penggunaan *Regular Expression* karena penggunaannya yang jauh lebih mudah. *DOM Parser* memiliki kelebihan untuk mengambil data karena *developer* dapat langsung mengambil *tag-tag* di dalam sebuah halaman HTML, sehingga hasil yang dihasilkan oleh pengambilan akan langsung berbentuk teks.

```
// Create DOM from URL or file
$html = file_get_html('http://www.google.com/');

// Find all images
foreach($html->find('img') as $element)
    echo $element->src . '<br>';

// Find all links
foreach($html->find('a') as $element)
    echo $element->href . '<br>';
```

Gambar 2.12 Contoh penggunaan *DOM Parser*

Pada Gambar 2.12 dapat terlihat contoh penggunaan *DOM Parser*. Dalam gambar pengguna dapat menggunakan *DOM Parser* hanya dengan memanggil fungsi-fungsi yang disediakan oleh *DOM Parser*. Di Gambar 2.12 dicontohkan

bahwa *DOM Parser* dapat dengan mudah mencari semua gambar pada sebuah halaman HTML dan mencari semua *link* pada sebuah halaman HTML juga.

2.13. *Regular Expression*

Regular Expression biasa disingkat dengan singkatan *Regex*, adalah suatu pola yang mendeskripsikan beberapa karakter yang terbatas. *Regex* bisa digunakan langsung di dalam naskah PHP jika diperlukan selama library *Regex* sudah terpasang di dalam library PHP.

Bentuk dari *Regex* adalah sebagai berikut:

```
\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}\b
```

\b di depan dan di belakan pola *Regex* berarti menandakan *boundaries* yang akan menandakan batasan pola *Regex*. *Regex* bisa dipisahkan menjadi beberapa kelompok, suatu kelompok ditandai dengan tanda kurung, kurungan pertama berarti pola didalam kurung akan mengambil semua alphabet “a” sampai “z” angka “0” sampai dengan “9”, titik, *underscore*, lambang persen, dan tanda hubung. Setelah disambung dengan tanda “@”, kurung kedua akan mengambil semua karakter dari 26 alphabet dan semua digit angka. Group ketiga akan mengambil semua huruf dari abjad alphabet. Contoh pola ini akan mengambil semua alamat email dalam sebuah file teks.