OPTIMALISASI SISTEM PENGENALAN WICARA DENGAN LEVENSTHEIN EDIT DISTANCE UNTUK TRADING FOREX ONLINE

Andi Iwan Nurhidayat
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Intitut Teknologi Sepuluh Nopember,andyl34k5@gmail.com

I Ketut Eddy Purnama
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Intitut Teknologi Sepuluh Nopember,ketut@ee.its.ac.id

Surya Sumpeno
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Intitut Teknologi Sepuluh Nopember, surya.its@gmail.com

Abstrak

Kecepatan, kenyamanan serta kemudahan merupakan hal penting dalam transaksi forex online. Sampai saat ini, software metatrader 4 belum mendukung pengenalan wicara untuk transaksinya. S\textit{oftware metatrader 4 memberikan fasilitas berupa expert advisor, script dan indicator yang dapat dibuat atau disesuaikan dengan kebutuhan trader. Fasilitas yang diberikan oleh metatrader 4 harus dibangun menggunakan bahasa MetaQuotes Languages(MQL). Bahasa MetaQuotes Languages(MQL) memiliki fungsi dan library yang terbatas sehingga memerlukan library dari platform} lain.

Beberapa penelitian sebelumnya[1] telah membuktikan bahwa, penerapan pengenalan wicara secara otomatis memberikan kemudahan dan kecepatan waktu yang lebih baik dengan pengoperasian baik secara manual maupun semi manual pada tools dan aplikasi yang digunakan.

Pada penelitian ini menggunakan socket untuk komunikasi dengan software metatrader 4 dan HTTP Request / Response untuk komunikasi dengan google speech engine. Berdasarkan uji coba yang telah dilakukan pengenalan wicara menggunakan Google Speech API memberikan hasil yang lebih cepat 4 dan tentunya kemudahan dibandingkan dengan pengoperasian semi-manual.

Kata Kunci: *Forex, Googe Speech Api, Expert Adisor, Script, Indicator, HTTP Request / Response*

1. PENDAHULUAN

FOREX (*Foreign Currency Exchange*) adalah pertukaran nilai mata uang asing yang dibandingkan satu dengan lainnya. Nilai tukar mata uang tersebut memberikan data secara signifikan yang diperlukan untuk perdagangan mata uang di pasar moneter internasional. Harga Forex dipengaruhi oleh beberapa factor antara lain kondisi ekonomi, politik suatu Negara, dan bahkan kondisi psikologis para trader dan investor. Faktor-faktor tersebut sangat berkorelasi dan berinteraksi antara satu dengan lainnya dengan cara yang sangat kompek.

Trader melakukan transaksi di valuta asing menggunakan alat (*software*) yang disediakan oleh pialang valuta asing (*broker*). *Metatrader 4* merupakan alat yang diberikan oleh pialang valuta asing untuk menghubungkan nasabah (trader) dengan pasar valuta asing. Dengan software tersebut, trader dapat mengamati pergerakan harga valuta asing dari time frame satu menit sampai satu bulan. Sampai saat ini software metatrader 4 belum mendukung pengenalan wicara untuk transaksinya.

Tidak dapat dipungkiri dengan adanya *Automatic Speech Recognition* (ASR) dapat mempermudah dan mempercepat pengguna dalam pengoperasian aplikasi dan komputer[1].*Speech recognition* atau proses pengenalan wicara telah banyak dikembangkan dengan berbagai metode serta diterapkan pada berbagai bidang. Kemampuan yang handal untuk memodelkan sesuatu yang kompleks dan algoritma yang efisien dalam menangani sejumlah data yang sangat besar merupakan kelebihan HMM yang digunakan untuk membangun *Automatic Speech Recognition* (ASR). Penerapan ASR dengan menggunakan Bahasa Indonesia masih memerlukan banyak penyesuaian dan perubahan yang harus dilakukan. Selain itu, pengembangan ASR untuk Bahasa Indonesia memerlukan dukungan berbagai informasi linguistik yang masih relatif terbatas.

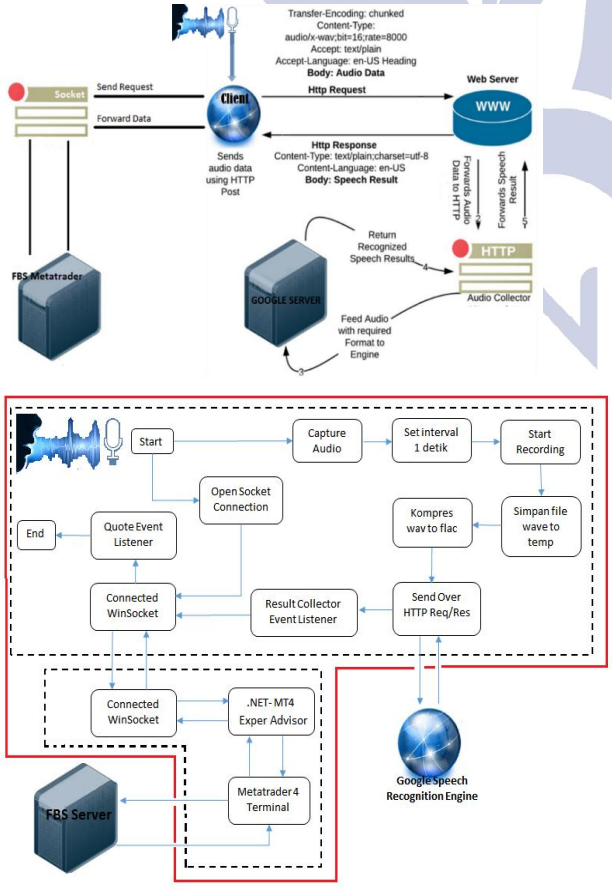
Terdapat dua pendekatan yang dilakukan pada implementasi *Automatic Speech Recognition* (ASR) yaitu *embeded system* dan *server system*. Kelebihan server system memiliki tingkat akurasi yang lebih baik dibandingkan dengan embeded system, karena *server*

system memiliki data yang lebih banyak sebagai bahan pelatihan. Sebagaimana diketahui *Google Speech Api* termasuk dalam kategori *server system* yang menggunakan *HTTP Request / Response* untuk berkomunikasi dengan *client*. *Google Speech Api* telah diaplikasikan pada *web base (HTML 5) application* dan *chrome engine* serta hanya mendukung audio file yang bertipe *Flac* dan *Speex*, sehingga pada *chrome browser* terdapat *engine* yang mengkompresi *wav file* menjadi *flac file*.

Pada penelitian ini *Speech recognition* atau proses pengenalan wicara menggunakan *Google Speech Api* akan diterapkan untuk melakukan transaksi Forex, sehingga diharapkan para trader Forex dapat mengurangi kesalahan yang sering terjadi dalam transaiksi dan mempercepat proses transaksi.

2. METODE

Pada penelitian ini dibuat sebuah *Speech Recognition system* untuk Trading Forex seperti yang terlihat pada pada Gambar 1. Secara umum system tersebut terdiri daritigaapilikasiutama,aplikasiyangmenanganipengolahan suara(aplikasidesktop), aplikasi *web service* untuk pengiriman dan penerimaan dari internet, serta aplikasi*Gateway*yangmenjembataniantaraaplikasidesktopd enganaplikasi*metatrader 4*



Gambar 1. Design Sistem Aplikasi

3. DATA

Data sinyal suara yang digunakan pada penelitian ini diperoleh dengan melakukan perekaman suara sendiri sebanyak 15 x yang disesuaikan dengan kata atau prase yang terjadi pada transaksi forex menggunakan *software metatrader 4*. Sebagai contoh perintah "*sell 0.3 take profit 6*" akan dipadanankan dengan pengucapan wicara "beli nol titik tiga profit enam". Untuk lebih lengkap dapat dilihat pada tabel 1.

Table 1. Tabel Suara Yang Direkam sebanyak 15x

No	Perintah Metatrader 4	Kalimat Uji Coba
1	buy/jual	Beli
2	sell 0.3 tp 6	jual 0.3 profit 6
3	buy 0.1	beli 0.1
4	buy 0.2 tp 5	beli 0.2 profit 5
5	jual 0.2	jual 0.2
6	buy tp 5	beli profit 5
7	buy tp 5 sl 5	beli profit 5 rugi 5
8	buy 0.1 tp 5	beli 0.1 profit 5
9	buy sl 9	beli rugi 9
10	buy 0.2 tp 5 sl 5	jual 0.2 profit 5 rugi 5
11	jual 0.2 limit tp 5 sl 5	jual 0.2 limit 5 profit 5 rugi 5
12	buy 3 limit 7 tp 5 sl 9	beli 3 limit 7 profit 5 rugi 9
13	buy 0.3 tp 5 sl 9	beli 0.3 profit 5 rugi 9
14	sell stop tp 3	jual tunda 3
15	buy stop 3 tp 5 sl 9	jual tunda 3 profit 5 rugi 9
16	sell stop tp 10	jual tunda 10
17	sell stop 10 tp 5 rugi 9	jual tunda 10 profit 5 rugi 9
18	buy 0.2 stop 10 tp 5 rugi 9	beli 0.2 tunda 10 profit 5 rugi 9
19	buy 0.2 stop 10	beli 0.2 tunda 10
20	sell 0.3 limit 10 tp 5 rugi 5	jual 0.3 limit 10 profit 5 rugi 5

3.1 PROSES *PEREKAMAN*

Dalam proses perekaman suara secara digital, ada 2 hal yang menentukan kualitas dari format sebuah file audio digital, yaitu :

- *Sample Rate*, yaitu tingkatan dimana sample ditangkap atau dimainkan. Satuannya adalah *Hertz* atau *Sample per Second*.
- *Sample Format* atau *sample size*, yaitu jumlah angka pada saat representasi digital dari setiap sampel.

Persyaratan frekuensi sampling menurut teorema Shannon harus sama dengan atau melebihi 2 kali frekuensi sinyal yang di sample

$$fs \geq 2xf1,$$

Jikasinyalinformasiyang disamplememilikikomponenfrekuensi beragam,misalnya untuk sinyal wicara memungkinkan untuk memiliki frekuensi dari 20 sampai 4000 Hz, maka sinyal informasi tersebut bisa dituliskan sebagai

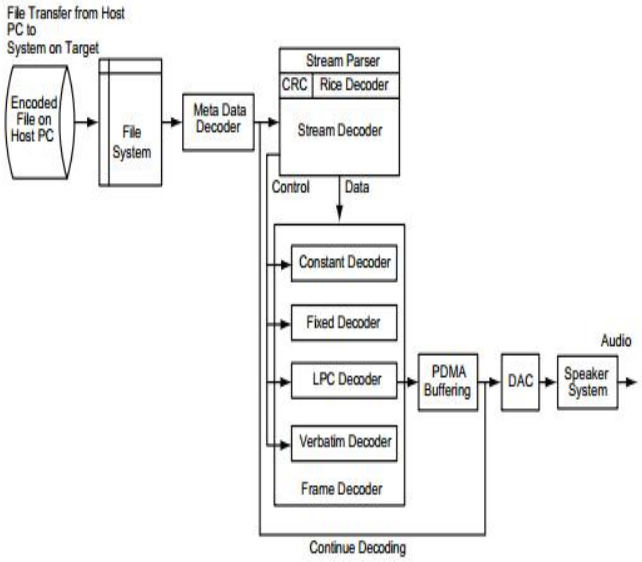
$$x(t) = \sum_{i=1}^{i=max} \sin(2\pi f_i t,$$

Dan persyaratan untuk frekuensi sampling menjadi :$fs \geq 2xf1max,$

Frekuensi sampling seringkali dikatakan dengan terminology *sampling rate*, yaitu jumlah sample yang diambil setiap detik, fs = 1Ts yang juga dikenal sebagai *Nyquist rate*.

3.2 KOMPRESI FILE AUDIO

Pada tahap ini file hasil rekaman yang berforamat *wav* dirubah menjadi format *flac* melalui *flac encoder* dengan tahapan sebagai berikut[2] :



Gambar 2. Tahapan Kompresi FLAC

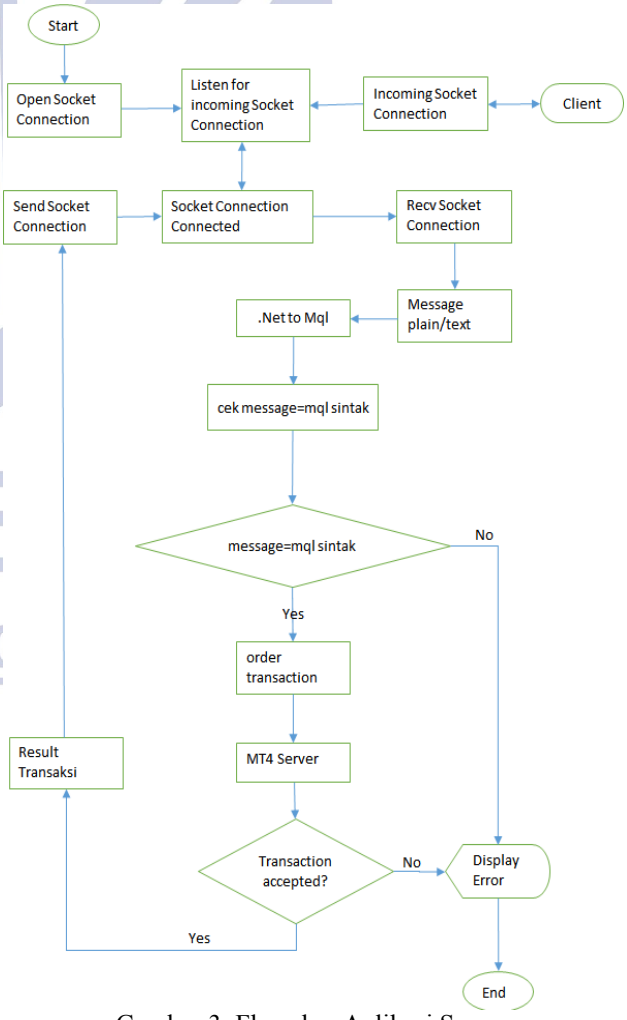
- *Blocking*,
  Input bentuk gelombang dipecah menjadi beberapa blok yang berdekatan. Blok dalam FLAC mengacu pada deretan sample pada beberapa channel. Ukuran blok dapat berbeda-beda, bergantung pada beberapa faktor termasuk sample rate. Ukuran blok ini mempengaruhi rasio kompresi secara lang- sung. Jikaukuranblokterlalukecil,makadibutuhkanbanyakframesehingga banyak bit akan terbuang untuk menyimpan frame header. Jika terlalu besar, karakteristik sinyal audio akan terlalu bervariasi sehingga sulit menemukan predictor yang optimal. FLAC membatasi ukuran blok antara 16 hingga 65535 sample per blok
- *Interchannel Decorrelation*,
  Pada tahap ini encoder akan membuat sinyal Mid-Side berdasarkan rata-rata dan perbedaan (masing-masing) dari saluran kiri dan kanan. Encoder kemudian akan memberikan bentuk terbaik dari sinyal ke tahap prediksi.
- *Prediction*
  Untuk menghilangkan *redudancy* yang terjadi digunakan metode *Linier Predictive Coding(LPC).*

$$\hat{s}(t) = \sum_{i=1}^{p} a_i s(t-1)$$

Dimana :
$\hat{s}(t)$ *= estimasi linier predictor*
a_i *= koofesien predictor*
s(t) = speech signal

- *Residual coding* Jika pada tahap prediksi terdapat perbedaan dengan sinyal asli(disebut resid ual signal) maka dilakukan pengkodean dengan menggunakan algoritma Rice Code.

3.3 SERVER APPLICATION



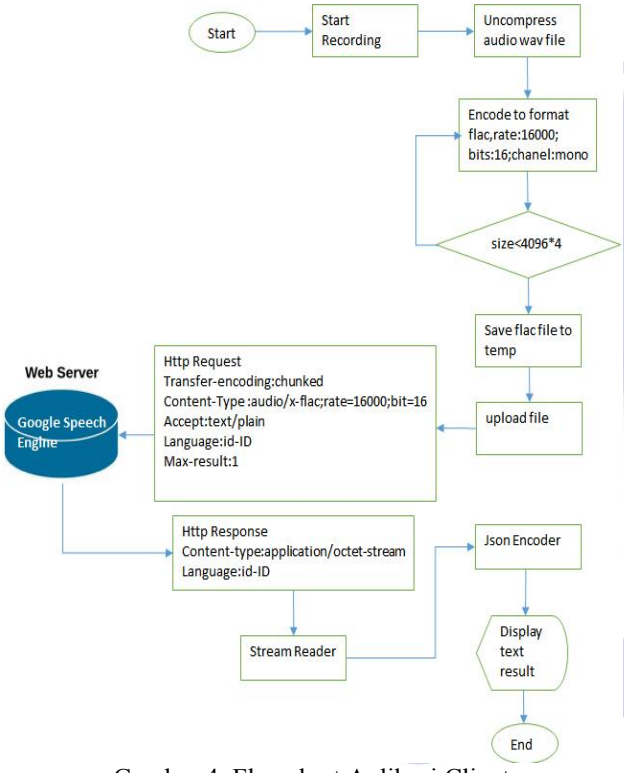
Gambar 3. Flowchar Aplikasi Server

Gambar 3 menunjukan aplikasi server melayani permintaan dari aplikasi asr ke aplikasi metatrader untuk

melakukan order transaction. Aplikasi metatrader menerima data dari aplikasi service melalui TCP/IP berupa message(plain/text). Aplikasi metatrader menjalankan perintah order transation melalui script yang dibuat dengan bahasa MQL.

Message yang diterima harus dikonversi menggunakan .Net-to-MQL libraryyang bertujuan merubah message (plain/text) menjadi perintah yang sesuai dengan format bahasa MQL. Jika keluaran yang dihasilkan sesuai dengan sintak MQL, maka perintah tersebut akan dieksekusi oleh metatrader dengan mengirimkan ke broker server.

3.4 CLIENT APPLICATION



Gambar 4. Flowchart Aplikasi Client

Gambar 4 memperlihatkan proses selanjutnya berupa pengiriman/upload file flackegooglespeechenginemelaluiHTTPRequest. GoogleServer mengirimhasil request berupa teks dengan format JSON dan diterima oleh aplikasi client melalui HTTP Response.

HTTP Request/Response merupakan teknik pengunduhan/pengunggaha dengan membenamkan data audio/teks dalam HTTP Post message.

3.5 WEB SPEECH

Web Speech API dikembangkan oleh *W3C Speech API Community Group* yang bersifat *open source* atau terbuka untuk semua orang/vendr[3], tetapi perkembangannya yang pesat didorong oleh dua perusahaan besar : Google dan Openstream.

3.3.1 Google Speech API

Perusahan Google memberikan*Google Speech API* secara gratis untuk digunakan oleh *browser* yang sudah mendukung *HTML5*.Secara garis besar cara kerja *speech recognition server based(google)* adalah data Audio dikumpulkan dari mikrofon, dan kemudian dikirim ke *Google server (Googel web service HTTPS POST)*, yang mengembalikan hasilnya dalam bentuk objek *JSON*.

Untuk dapat menggunakan *Google Speech API*, para *developer* harus mendapatkan *key* dengan menjadi anggota *Chromium Development Group*[4]. *API key* yang didapat hanya bisa melakukan request sebanyak 50 kali per hari.

Dengan membuka *source code Chromium*[5] diketahui terdapat 2 alamat *API endpoint* yaitu"https://www.google.com/speech-api/v2/recognize"dan"https://www.google.com/speech-api/full- duplex/v1/up". Dua *API endpoint* tersebut memiliki parameter yang sama antara lain:

- *key = API key* yang didapat pada anggota Chromium Development Group
- *lang*= bahasa yang diinginkan
- *output*=keluaran yang diinginkan "pb" untuk binary dan "json" untuk string json
- *maxresult*=jumlah pengenalan yang diinginkan
- *Content-Type: audio/x-flac; rate=16000*

Google Speech Server menerima data audio dengan format speex dan flac.

Untuk menggunakan *Google Speech API* sebagai mesin *Speech Recognition* dibutuhkan *Flac File* yang menyimpan suara yang direkam dan tool curl untuk menanganipermintaan*HTTP POST*,sehinggadapatdilakukanperintahsepertipada Gambar 5

```
curl -i -X POST -H "Content-Type:audio/x-flac; rate=12000" \
-T test.flac \
"https://www.google.com/speech-api/v1/recognize?xjerr=1\
&client=chromium&lang=en-EN&maxresults=10&pfilter=0"
```

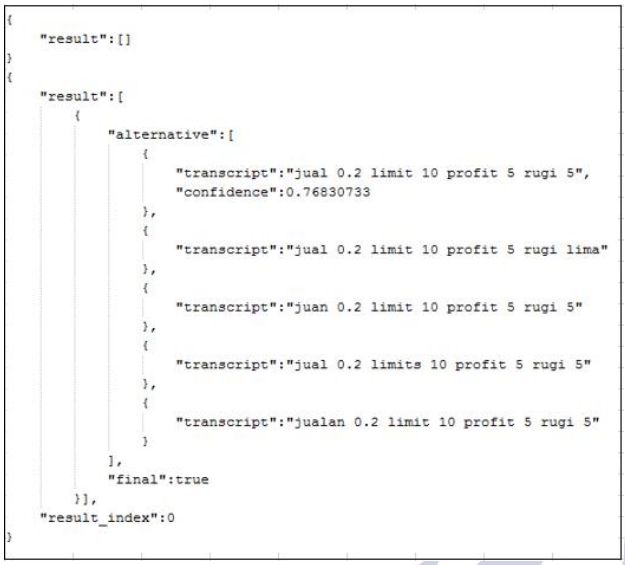
Gambar 5 Penggunaan *curl tool* untuk *Google Speech API*

Penjelasan parameter diatas adalah sebagai berikut :

- -H "Content-Type:audio/x-flac; rate=12000" Ini memberitahukan kepada Google Server, bahwa Flac File telah dikirim dengan bitrate 12000Hz
- -T test.flac File yang disertakan pada HTTP POST
- https://www.google.com/speech-api/v2/recognize?xjerr=1&client= chromium&lang=en-US&maxresults=10&pfilter=0

Merupakan alamat Google Speech Recognition Engine

ResponeyangdiberikandariGoogleserver berupadataJSON sepertipada Gambar 6.



```
{
    "result":[]
}
{
    "result":[
        {
            "alternative":[
                {
                    "transcript":"jual 0.2 limit 10 profit 5 rugi 5",
                    "confidence":0.76830733
                },
                {
                    "transcript":"jual 0.2 limit 10 profit 5 rugi lima"
                },
                {
                    "transcript":"juan 0.2 limit 10 profit 5 rugi 5"
                },
                {
                    "transcript":"jual 0.2 limits 10 profit 5 rugi 5"
                },
                {
                    "transcript":"jualan 0.2 limit 10 profit 5 rugi 5"
                }
            ],
            "final":true
        }],
    "result_index":0
}
```


Gambar 6. Response Google Speech API

3.6 LEVENSTHEIN EDIT DISTANCE

Berdasarkan beberapa penelitian terdahulu, akurasi pengenalan wicara oleh *Google Speech API* sebesar ±50%. Berdasarkan hasil tersebut, maka diperlukan sebuah algoritma untuk meningkatkan akurasi hasil tersebut. Salah satu algoritma yang digunakan untuk meningkatkan akurasi tersebut adalah *levensthein edit distance*.

Terdapatbeberapavariasiformulauntukal- goritma levensthein edit distance. Salah satu formula yang digunakan pada peneli- tian ini adalah sebagai berikut :

$$M_{i,j}\begin{cases} M_{i,j} \leftarrow 0 \\ M_{i-1,j}+1 \\ M_{i,j-1}+1 \\ M_{i-1,j-1}+\delta(x_i,y_j) \end{cases}$$

Dimana :

x_i dan y_j= karakter terakhir.

jika $x_i = y_j$ maka $x_{1\ldots i} = y_{1\ldots j}$ dengan edit distance sama dengan $M_{i-1,j-1}$.

jika $x_i \neq y_j$ maka $x_i = y_j$ dengannilaisubstitusisamadengan$M_{i-1,j-1}$+1,atau

x_i dapat dihapus dengan nilai $M_{i-1,j}$ +1, atau

y_j dapat disisipi ke x dengan nilai $M_{i,j}+1$

minimum edit distance=$M_{m+1,n+1}$

Tabel 2 adalah contoh matrik yang dihasilkan untuk menghitung edit distance antara string "DFGDGBDEGGAB" dan "DGGGDGBDEFGAB". Nilai edit dis- tance untuk matrik $M_{m+1,n+1}$ adalah 3.

Tabel 2. Matrik edit distance untuk string "DFGDGBDEGGAB" dan "DGGGDGBDEFGAB"

		D	G	G	G	D	G	B	D	E	F	G	A	B
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
D	1	0	1	2	3	4	5	6	7	8	9	10	11	12
F	2	1	1	2	3	4	5	6	7	8	8	9	10	11
G	3	2	1	1	2	3	4	5	6	7	8	8	9	10
D	4	3	2	2	2	2	3	4	5	6	7	8	9	10
G	5	4	3	2	2	3	2	3	4	5	6	7	8	9
B	6	5	4	3	3	3	3	2	3	4	5	6	7	8
D	7	6	5	4	4	3	4	3	2	3	4	5	6	7
E	8	7	6	5	5	4	4	4	3	2	3	4	5	6
G	9	8	7	6	5	5	4	5	4	3	3	3	4	5
G	10	9	8	7	6	6	5	5	5	4	4	3	4	5
A	11	10	9	8	7	7	6	6	6	5	5	4	3	4
B	12	11	10	9	8	8	7	6	7	6	6	5	4	3

Pada Tabel 2 , elemen baris 1 kolom 1 ($M[1,1]$) adalah jumlah operasi yang diperlukan untuk mengubah substring dari kata "DFGDGBDEGGAB" yang diambilmulaidarikarakterawalsebanyak1(D)kesubstringdarikata "DGGGDGBDEFGAB" yang diambil mulai dari karakter awal sebanyak 1 (D). Sementara M[3,5] adalah jumlah operasi antara "DFG" (substring yang diambil mulai dari karakter awal sebanyak 3) dengan "DGGGD" (substring yang diambil mulai dari karakter awalsebanyak5).

HalinidisimpulkanbahwaelemenM[p,q]adalahjumlahoperasi antarasubstringkatapertamayangdiambilmulaidariawalsebanyakpdengansub- string kata kedua yang diambil dari awal sebanyak q. Dengan demikian, Elemen terakhir(kananbawah)adalahelemenyangnilainyamenyatakanjarakkeduastringyang dibandingkan. Lalu untuk menghitung nilai kemiripan menggunakan rumus:

$$Sim = 1 - (\frac{Dis}{Maxlength})$$

Dimana :

Sim = Similarity/nilaikemiripan

Dis = jarakLevenshtein

MaxLength = nilaistringterpanjang

Jika nilai similarity adalah 1, maka kedua string yang dibandingkan sama. Di lain hal, jika similarity 0, maka kedua string yang dibandingkan tidak sama.

4. HASIL DAN PEMBAHASAN

4.1 Google Speech API

Pengiriman data pada *Google Speech engine* dilakukan secara real time dengan berbicara pada mikropon dan mengirim langusng ke *Google Server*. Untuk setiap kata atau prasa yang dikirim dilakukan sebanyak 15 kali. Table 2 menunjukkan akurasi hasil dari

pengenalan wicara menggunakan *Google Speech API* sebesar 55%.

Disamping untuk menghitung nilai akurasi kata pada kalimat pengenalan Google, Algoritma *Levenshtein edit distance* digunakan untuk mencari kata-kata yang memiliki kesamaan dalam kamus kata

Kamus Kata	bel	0	.	1	lim	1	0	prof	1	0	rug	8
beli	1	4	4	4	3	4	4	4	4	4	4	4
jual	3	4	4	4	4	4	4	4	4	4	3	4
profit	6	6	6	6	5	6	6	2	6	6	5	6
rugi	4	4	4	4	4	4	4	4	4	4	1	4
limit	5	5	5	5	2	5	5	5	5	5	5	5
tunda	5	5	5	5	5	5	5	5	5	5	4	5
.	3	1	0	1	3	1	1	4	1	1	3	1
0	3	0	1	1	3	1	0	4	1	0	3	1
1	3	1	1	0	3	0	1	4	0	1	3	1
2	3	1	1	1	3	1	1	4	1	1	3	1
3	3	1	1	1	3	1	1	4	1	1	3	1
4	3	1	1	1	3	1	1	4	1	1	3	1
5	3	1	1	1	3	1	1	4	1	1	3	1
6	3	1	1	1	3	1	1	4	1	1	3	1
7	3	1	1	1	3	1	1	4	1	1	3	1
8	3	1	1	1	3	1	1	4	1	1	3	0
9	3	1	1	1	3	1	1	4	1	1	3	1

Gambar 7. Penerapan Levensthein pada kamus kata

Gambar 7 menunjukkan untuk setiap kata pada kalimat "bel 0.1 lim 10 prof 10 rug 8" akan dibandingkan dengan kata-kata yang terdapat pada kamus dan dihitung nilai minimum edit distance-nya. Dalam hal ini kamus kata berisi ["beli","jual","profit","rugi","limit","tunda",".","0","1","2","3","4","5","6","7","8","9"]. Sebagai contoh kata "bel" akan dibandingkan dengan setiap kata yang terdapat dalam kamus, kemudian dihitung edit distance-nya dan selanjutnya mencari nilai minimumnya sebagai hasil akhir. Nilai minimum edit distance digunakan sebagai acuan untuk mengganti kata yang salah dengan kata yang terdapat pada kamus kata. Kata "bel" pada contoh ini memiliki nilai minimum edit distance paling kecil sebesar 1 dengan kata "beli", sehingga kata "bel" akan diganti menjadi kata "beli". Dan begitu seterusnya untuk kata-kata yang lain. Dengan melakukan perhitungan minimum edit distance setiap kalimat yang salah dari hasil pengenalan *Google Speech API* didapat hasil seperti pada Tabel 3

Berdasarkan Tabel 3 dapat diketahui bahwa penerapan algoritma Levensthein Edit Distance pada hasil pengenalan *Google Speech api* meningkatkan akurasi dari 55% menjadi 80%.

Table 3. Tabel Perbandingan Akurasi Google Speech API dan Optimalisasi dengan Lavensthein Edit Distance

No	Kalimat Uji Coba	Percobaan	Akurasi Prasa	Koreksi
1	beli	15	80%	100%
2	jual 0.3 profit 6	15	80%	100%
3	beli 0.1	15	80%	100%
4	beli 0.2 profit 5	15	87%	100%
5	jual 0.2	15	73%	87%
6	beli profit 5	15	73%	87%
7	beli profit 5 rugi 5	15	60%	80%
8	beli 0.1 profit 5	15	80%	93%
9	beli rugi 9	15	80%	93%
10	jual 0.2 profit 5 rugi 5	15	40%	67%
11	jual 0.2 limit 5 profit 5 rugi 5	15	33%	67%
12	beli 3 limit 7 profit 5 rugi 9	15	20%	53%
13	beli 0.3 profit 5 rugi 9	15	20%	60%
14	jual tunda 3	15	100%	80%
15	jual tunda 3 profit 5 rugi 9	15	27%	73%
16	jual tunda 10	15	80%	100%
17	jual tunda 10 profit 5 rugi 9	15	13%	67%
18	beli 0.2 tunda 10 profit 5 rugi 9	15	7%	53%
19	beli 0.2 tunda 10	15	53%	73%
20	jual 0.3 limit 10 profit 5 rugi 5	15	13%	60%
		Rata-rata	55%	80%

4.2 Transaksi Semi-manual dan Transaksi dengan Wicara

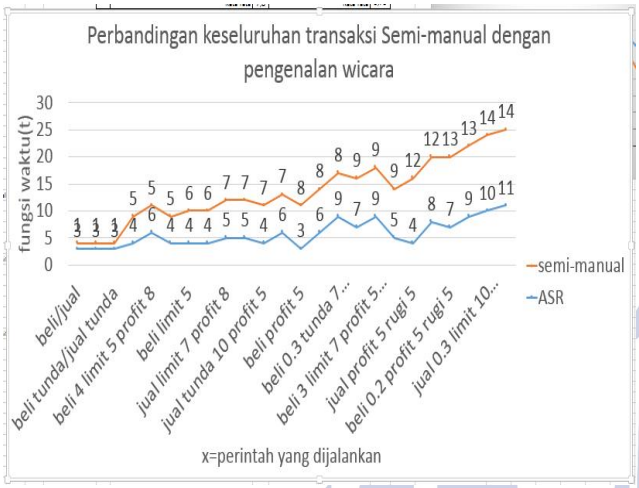
Setiap *order transaction*semi-manual dijalankan sebanyak 5 kali untuk mendapatkan nilai rata-rata waktu yang dibutuhkan oleh metatrader. Sebagai contoh, *order transaction"buy 0.1"* membutuhkan waktu rata-rata 4,2 detik. Tabel 2 menunjukkan waktu yang dibutuhkan untuk semua order transaction yang terdapat pada forex trading.

Untuk transaksi menggunakan wicara dilakukan dengan cara memilih hasil pengenalan wicara oleh Google dan telah diotimalisasi dengan *Lavensthein Edit Distance* yang mencapai akurasi 100% dan dilakukan sebanyak 15 kali. Sebagai contoh pengucapan "beli nol titik satu" akan diterjemahkan sebagai *"buy 0.1"* dalam perintah forex.

Table 4. Tabel Perbandingan semi-manual dengan pengenalan wicara.

No	Perintah yang dijalankan	waktu	Suara yang dijalankan	waktu
1	buy/sell	1	beli/jual	3
2	buy limit /sell limit	1	jual limit	3
3	buy stop/sell stop	1	beli tunda/jual tunda	3
4	buy N/sell N	5	beli 0.1	4
5	buy limit tp N/sell limit tp N	5	beli 4 limit 5 profit 8	6
6	buy stop tp N/sell stop tp N	5	jual tunda profit 5	4
7	buy limit N/sell limit N	6	beli limit 5	4
8	buy stop N/sell stop N	6	jual tunda 3	4
9	buy limit N tp N/sell limit N tp N	7	jual limit 7 profit 8	5
10	buy N limit N /sell N limit N	7	jual 4 limit 7	5
11	buy stop N tp N/sell stop N tp N	7	jual tunda 10 profit 5	4
12	buy N stop N/sell N stop N	7	beli 0.2 tunda 10	6
13	buy tp N/sell tp N	8	beli profit 5	3
14	buy N limit N tp N/sell N limit N tp N	8	jual 0.2 limit 10 profit 5	6
15	buy N stop N tp N/sell N stop N tp N	8	beli 0.3 tunda 7 profit 7	9
16	buy N tp N/sell N tp N	9	beli 0.1 profit 5	7
17	buy limit tp N sl N/sell limit tp N sl N	9	beli 3 limit 7 profit 5 rugi 9	9
18	buy stop tp N sl N/sell stop tp N sl N	9	jual tunda profit 5 rugi 9	5
19	buy tp N sl N/sell tp N sl N	12	jual profit 5 rugi 5	4
20	buy limit N tp N sl N/sell limit N tp N sl N	12	jual limit 5 profit 5 rugi 5	8
21	buy N tp N sl N/sell N tp N	13	beli 0.2 profit 5 rugi 5	7
22	buy stop N tp N sl N/sell stop N tp N sl N	13	jual tunda 3 profit 5 rugi 9	9
23	buy N limit N tp N sl N/sell N limit N tp N sl N	14	jual 0.3 limit 10 profit 5 rugi 5	10
24	buy N stop N tp N sl N/sell N stop N tp N sl N	14	beli 0.3 tunda 7 profit 7 rugi 10	11
	Rata-rata	7,79	Rata-rata	5,79

Dari percobaan yang telah dilakukan Gambar 8 menunjukkan bahwa transaksi forex online dengan menggunkan pengenalan wicara lebih cepat dibandingkan dengan transaksi secara semi-manual.



Gambar 8. Grafik Perbandingan Manual dengan ASR

5. Kesimpulan dan Saran

5.1 Kesimpulan :

1. Pengenalan wicara untuk transaksi forex online menggunakan *Google Speech API* memberikan hasil yang kurang memuaskan. Dari hasil uji coba, akurasi *google speech API* hanya 55%.
2. Dengan menggunakan Algoritma lavensthein edit distance dapat meningkatkan hasil dari Google speech API hingga mencapai 80%
3. Pengenalan wicara menggunakan *Google Speech API* memberikan hasil yang lebih cepat 4 detik dan tentunya kemudahan dibandingkan dengan pengoperasian semi-manual

5.2 Saran

Sistem aplikasi pengenalan suara pada penelitian ini menggunakan aplikasi yang berbeda platform(.NET) dengan *softwaremetatrader 4*(MQL4) sehingga memerlukan socket untuk dapat berkomunikasi. Sehingga terdapat jeda waktu ketika mengirimkan perintah transaksi. Untuk mengurangi jeda waktu tersebut dapat diselesaikan dengan membangun aplikasi pengenalan wicara menggunakan platform yang sama dengan software metatrader 4 dengan menggunakan bahasa MQL.

6. DAFTAR PUSTAKA

Zeljko Medenica and Andrew L. Kun, "COMPARING THE INFLUENCE OF TWO USER INTERFACES FOR MOBILE RADIOS ON DRIVING PERFORMANCE",PROCEEDINGS of the Fourth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, Hong Kong and Macau 978-1-4244-8376-1/10 IEEE 2010

SmartFusion cSoC: Implementation of FLAC Player Using Hardware and Software Partitioning,Application Note AC376

https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html, Glen Shires & Hans Wennborg, Google Inc., 19 October 2012, W3C

Chromium Development Group: https://groups.google.com/a/chromium.org/forum/#!forum/chromium-dev

http://src.chromium.org/viewvc/chrome/trunk/src/content/browser/speech/, Chromium Source Code for the Speech API

C. Southern B. Frey and M. Romero. Brailletouch: Mobile texting for the visually impaired. In International Conference on Human-Computer Interaction, 2011

Levenshtein. Binary codes capable of correcting deletions, insertions, and rever- sals. Soviet Physics-Doklady, 1966.

GonzaloNavarroandMathieuRaffinot. FlexiblePatternMatchinginStrings: Prac- tical On-line Search Algorithms for Texts and Biological Sequences. Cambridge University Press, 2002.