

## Rancangan Aplikasi Android Penerjemah Wicara ke Wicara Dengan Komunikasi Dua Arah

Agung Santosa\*, Asril Jarin\*, Lyla Ruslana Aini, Fara Ayuningtyas, Gunarso, Made Gunawan, Mohammad Teduh Uliniansyah, Andi Djalal Latief, Gita Citra Puspita, Elvira Nurfadhilah, Harnum Annisa

Pusat Teknologi Informasi dan Komunikasi  
Badan Pengkajian dan Penerapan Teknologi (BPPT)  
Jakarta, Indonesia

\*e-mail: agung.santosa@bppt.go.id, asril.jarin@bppt.go.id

**Abstrak**— Dengan ketersediaan sumber daya kebahasaan dan sistem Pengolahan Bahasa Alami yang sudah dikembangkan sebelumnya, kegiatan-kegiatan rekayasa Teknologi Bahasa BPPT melakukan pengembangan sebuah aplikasi penerjemah wicara-ke-wicara untuk dua Bahasa (Bahasa Indonesia dan Bahasa Inggris) yang memanfaatkan layanan dari server pengenalan wicara, mesin penerjemah, dan sintesis wicara. Aplikasi ini dikenal sebagai *speech-to-speech translation* (S2ST). Di makalah ini, kami deskripsikan rancangan aplikasi S2ST tersebut dengan fokus pengembangan pada aplikasi mobile android yang dapat melayani percakapan antara dua pengguna. Teknik-teknik yang diterapkan antara lain adalah WebSocket, RESTful service, JSON, dan OkHttp3.

**Katakunci** – penerjemah wicara ke wicara; S2ST; NLP; ASR; MT; TTS; WebSocket; RESTful Service.

### I. PENDAHULUAN

Sejak tahun 1987, BPPT sudah mengembangkan sumber daya kebahasaan, diantaranya: korpus monolingual Bahasa Indonesia, korpus paralel Bahasa Indonesia-Bahasa Inggris, korpus wicara Bahasa Indonesia, kamus, leksikon Bahasa Indonesia, dan lain-lain (Gunarso et al., 2016). Beberapa sistem Pengolahan Bahasa Alami yang memanfaatkan sumber daya kebahasaan tersebut sudah berhasil dikembangkan sampai pada tahapan tertentu, seperti: pengenalan wicara otomatis (Gunawan et al., 2018; Uliniansyah, Riza, & Riandi, 2013), mesin penerjemah, dan sintesis suara (Santosa et al., 2018). Khususnya, pada tahun 2010 BPPT mengembangkan sebuah aplikasi risalah rapat otomatis yang didukung oleh sebuah model pengenalan wicara otomatis. Model dilatih dengan data wicara yang berasal dari 200 pembicara (100 pria dan 100 wanita) yang masing-masing mengucapkan 250 kalimat yang berbeda yang dikumpulkan dari berbagai sumber, seperti: surat kabar, majalah, novel, naskah, pidato politik, dan risalah sidang kementerian dan lembaga (Riandi, Jarin, Uliniansyah, Santosa, & Prasetyo, 2010).

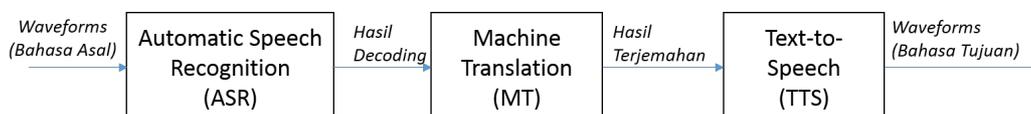
Sebagai bagian dari konsorsium U-STAR (Universal Speech Translation Advanced Research) pada tahun 2010, Indonesia berkontribusi dalam sebuah proyek pengembangan sistem penerjemah wicara-ke-wicara berbasis jaringan (U-STAR Consortium, 2016).

Selanjutnya, BPPT mengikuti tren pengembangan sistem penerjemah wicara-ke-wicara tersebut, atau yang disebut sistem *speech-to-speech translation* (S2ST), ke dalam kegiatan rekayasa Teknologi Bahasa dari tahun 2013 hingga sekarang.

Di makalah ini, kami deskripsikan sebuah rancangan aplikasi mobile android S2ST yang dapat memfasilitasi komunikasi dua arah (percakapan antara dua pengguna). Aplikasi S2ST akan memanfaatkan layanan server S2ST yang berada di *back-end*. Layanan server S2ST tersebut adalah layanan yang menggabungkan tiga proses pengolahan bahasa alami secara *pipeline*, yaitu: pengenalan wicara, mesin penerjemah, dan sintesis wicara. Seluruh proses aplikasi dirancang ke dalam enam bagian: penanganan fungsi percakapan antara pengguna; pembentukan percakapan antar dua pengguna; *multiplexing* menggunakan WebSocket; rancangan format pertukaran data; rancangan proses pertukaran pesan; dan rancangan pengolahan pesan pada android dan server S2ST.

Sistem aplikasi S2ST yang dirancang ini merupakan sebuah upaya awal BPPT untuk menghasilkan inovasi serupa dengan teknologi-teknologi yang dikembangkan oleh pemain besar, seperti; Google Translate (Google, 2015), Microsoft Translator (Microsoft, 2018), dan iTranslate (iTranslate, 2018). Akan tetapi jika dibandingkan dengan aplikasi uniTRANS (U-STAR, 2013) yang dihasilkan oleh konsorsium U-STAR, sistem ini memiliki kelebihan pada arsitektur dengan komunikasi dua arah antara dua perangkat mobile user. Sistem ini juga berbeda dengan sistem aplikasi mobile android S2ST yang dimanfaatkan oleh pasukan amerika melalui hasil riset TransTac dari DARPA (Meermeier, Colbath, & Of, 2018), dimana hanya menggunakan satu perangkat mobile android sekalipun secara otomatis mampu mendeteksi bahasa yang direkamkan dari sebuah percakapan. Sedangkan aplikasi *mobile phone* S2ST yang pertama adalah VerbMobil (Wahlster, 2000), namun belum menerapkan sistem Android.

Selanjutnya, makalah ini diorganisir ke dalam empat Bab. Bab II kami kaji beberapa landasan teori untuk memahami desain aplikasi. Bab III mendeskripsikan rancangan aplikasi S2ST yang kami kembangkan. Bab IV menyimpulkan makalah ini.



Gambar 1. Model konseptual penerjemah wicara-ke-wicara

II. LANDASAN TEORI

A. *Speech-to-Speech Translation (S2ST)*

Aplikasi S2ST adalah aplikasi yang menerjemahkan kalimat-kalimat yang diucapkan dari sebuah Bahasa dan kemudian diujarkan ke dalam Bahasa lainnya dengan bantuan sebuah perangkat elektronik. Aplikasi memanfaatkan tiga layanan pengolahan bahasa alami, yakni: pengenalan wicara, mesin penerjemah, dan sintesis wicara atau text-to-speech (lihat Gambar 1). Berkas pesatnya kemajuan teknologi multimedia dan jaringan serta teknologi *smart phone*, aplikasi S2ST dapat diintegrasikan dan memberi salah satu fitur kenyamanan bagi pengguna dalam hal melakukan perjalanan wisata dan pertemuan-pertemuan internasional, seperti: konferensi, pertandingan, pameran, dan lain sebagainya. Dengan aplikasi ini, pengguna dapat mengurangi penggunaan kamus dan pemanfaatan jasa penerjemah atau pemandu perjalanan di luar negeri. Beberapa contoh aplikasi S2ST yang merupakan hasil riset bersama adalah uniTRANS dari U-STAR (U-STAR, 2016), JANUS-III (Lavie et al., 1997), dan NESPOLE! (Lavie et al., 2001). Sedangkan yang saat ini populer dan diproduksi oleh perusahaan besar adalah *Google Translate* (Google LLC, 2018) dan *Microsoft Translator* (Microsoft, 2018b), *iTranslate Voice* (iTranslate, 2018b), *TripLingo* (TripLingo, 2018), dan *Baidu Translate* (Baidu, 2018). Aplikasi-aplikasi populer ini sudah memiliki hasil akurasi yang bagus karena diperoleh dari hasil pemodelan yang menggunakan teknik-teknik *deep learning* (Lecun, Bengio, & Hinton, 2015).

B. *WebSocket*

WebSocket adalah protokol berbasis frame yang mengelola transaksi data teks dan biner antara klien dan server secara dua arah (*bidirectional*) dan *full-duplex*. WebSocket memiliki API yang paling mirip dengan *raw network socket* yang ditempatkan di browser, seperti: *open*, *read*, *write*, dan *close*. WebSocket API yang disediakan oleh browser sangat kecil dan sederhana. Semua hal detail di tingkat bawah, mulai dari pengaturan koneksi sampai pada pengelolaan *message* dikerjakan oleh browser. Sementara untuk menginisiasi koneksi baru dibutuhkan URL dari satu *resource* WebSocket dan beberapa *callback* aplikasi (Grigorik, 2013).

Aplikasi WebSocket klien dan server berkomunikasi melalui API yang berorientasi *message*. Pengirim menyediakan *payload*, dan penerima dinotifikasi oleh pengantaran WebSocket ketika seluruh *message* tersedia. Untuk mengaktifkannya, WebSocket menggunakan format *framing biner* yang membagi *message* aplikasi ke dalam satu atau beberapa *frame*, mengangkutnya ke tujuan, merangkai kembali, dan akhirnya memberitahu penerima ketika seluruh *message* sudah diterima.

Selain sebagai *socket*, WebSocket menyediakan sejumlah layanan tambahan:

- Negosiasi koneksi dan pelaksanaan *policy origin* yang sama
- Interoperabilitas dengan infrastruktur HTTP yang ada
- Komunikasi berorientasi *message* dan pembuatan *frame message* yang efisien

- Negosiasi subprotokol dan kemungkinan bisa ditambah dengan fungsi-fungsi yang baru (*extensibility*)

Spesifikasi secara detil tentang WebSocket ditemukan di RFC 6455 (Bott, 2011) yang dipublikasikan secara resmi oleh *Internet Engineering Task Force (IETF)* pada tahun 2011.

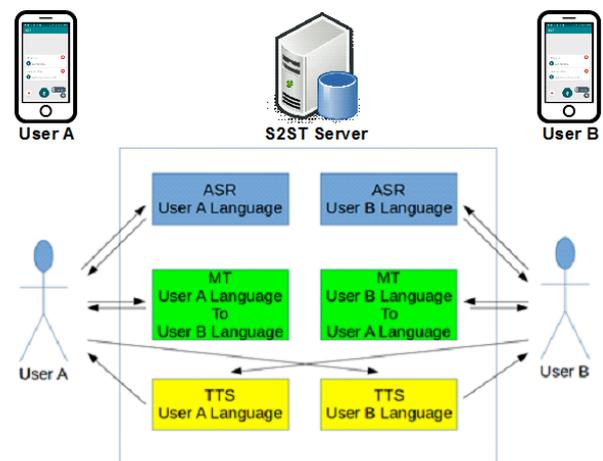
C. *RESTful Service*

*REpresentational State Transfer (REST)*, atau *RESTful web services* menyediakan interoperabilitas antara sistem komputer di Internet (Richardson & Ruby, 2007). *Web service* adalah sebuah layanan yang ditawarkan oleh sebuah perangkat elektronik ke perangkat elektronik lainnya, berkomunikasi satu sama lain melalui World Wide Web (World Wide Web Consortium (W3C), 2018). Dalam sebuah web service, teknologi Web seperti HTTP – awalnya dirancang untuk komunikasi antara manusia dengan mesin – digunakan untuk komunikasi mesin dengan mesin, secara lebih spesifik untuk mentransfer format-format berkas seperti JSON. JSON adalah format berkas standar yang terbuka yang menggunakan teks yang dapat dibaca manusia untuk mentransmisi objek-objek data yang berisi pasangan ‘atribut’ dengan ‘nilai’ dan tipe-tipe data array (Bray, 2014). JSON merupakan sebuah format data yang sangat umum digunakan untuk komunikasi asinkron antara browser dan server.

III. RANCANGAN APLIKASI MOBILE ANDROID S2ST DENGAN KOMUNIKASI DUA ARAH ANTAR DUA PENGGUNA

A. *Rancangan Global Penanganan Fungsi Percakapan Antar Pengguna S2ST*

Dalam rancangan penanganan percakapan antara pengguna S2ST, kendali alur dari pengolahan data percakapan banyak dilakukan oleh aplikasi Android, sedangkan Server S2ST sendiri bertindak sebagai penyedia tiga layanan NLP yaitu: ASR, MT dan TTS. Server S2ST juga menghubungkan dua pengguna kedalam sebuah sesi percakapan. Gambar 2 mengilustrasikan interaksi pengguna S2ST dan Server S2ST dalam suatu sesi percakapan.



Gambar 2. Arsitektur global aplikasi S2ST

Dalam ilustrasi terlihat *User A* melakukan percakapan dengan *User B* melalui Server S2ST. Server S2ST menyediakan layanan ASR, MT dan TTS. Yang membedakan dengan rancangan komunikasi satu arah adalah saat layanan TTS digunakan, dalam sebuah percakapan hasil dari TTS dikirimkan kepada lawan bicara dari pengguna layanan tersebut (seperti terlihat dalam Gambar 2).

**B. Rancangan Pembentukan Percakapan Antara Dua Pengguna S2ST**

Untuk dapat mendukung terjadinya percakapan antara dua pengguna S2ST perlu juga dirancang alur proses pembentukannya. Sebelum masuk kedalam langkah-langkah pembentukannya perlu diberikan kondisi awal dan beberapa definisi untuk mempermudah penjelasan prosesnya.

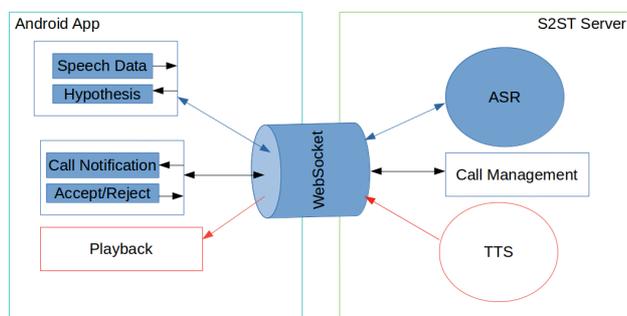
Prakondisi yang dipersyaratkan sebelum proses percakapan dapat dilakukan adalah: kedua pengguna yang melakukan percakapan sudah melakukan proses login kedalam Server S2ST. Dengan demikian Server S2ST nantinya dapat mengirimkan pesan notifikasi ke pengguna yang menerima panggilan.

Dalam proses ini pemanggil disebut sebagai “*Caller*” sedangkan pengguna yang dipanggil disebut sebagai “*Callee*”. Berikut adalah rancangan proses pembentukan percakapan melalui Server S2ST:

- *Caller* mengirimkan permintaan pemanggilan *Callee* kepada Server S2ST melalui WebSocket.
- Server S2ST memastikan ketersediaan sumberdaya untuk menangani permintaan tersebut jika sumberdaya tidak tersedia maka permintaan ditolak.
- Jika sumberdaya tersedia, Server S2ST mengirimkan notifikasi kepada *Callee* yang didalamnya terdapat informasi tentang *Caller* dan nomor identifikasi dari sesi percakapan. Notifikasi pemanggilan dikirimkan melalui WebSocket.

**C. Rancangan Multiplexing Menggunakan WebSocket**

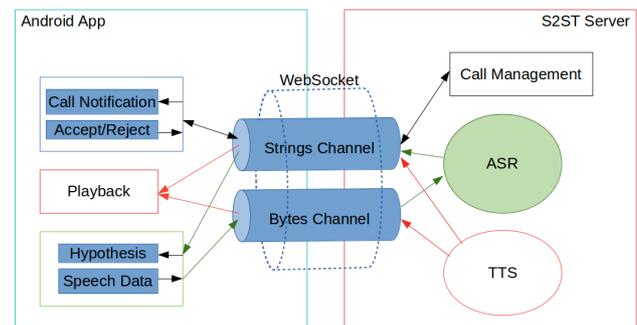
Dalam rancangan ini WebSocket digunakan untuk pertukaran data baik dalam bentuk data suara maupun perintah dan response nya, oleh karena itu dibutuhkan mekanisme yang dapat digunakan untuk mengakomodasi hal ini. Gambar 3 mengilustrasikan penggunaan WebSocket pada rancangan S2ST yang dibuat.



Gambar 3. Model Pemanfaatan WebSocket di Sistem S2ST

Untuk layanan ASR, WebSocket digunakan untuk mengirimkan data suara dari aplikasi android dan

mengirimkan hasil transkripsi berupa *hypothesis* dari Server S2ST. Sedangkan untuk *Call Management* WebSocket digunakan untuk menerima *request/reply* dan mengirimkan notifikasi panggilan. Selain dari itu WebSocket juga digunakan untuk mengirimkan WAV file yang merupakan hasil dari TTS. Data suara dan WAV file dikirimkan sebagai aliran byte sedangkan data yang lain dipertukarkan sebagai text/string. Pembagian ini secara *native* sudah didukung oleh WebSocket dimana kanal untuk menerima dan mengirim data dalam bentuk *byte* sudah dibedakan dari data dalam bentuk string. Sehingga secara ilustrasi dapat digambarkan seperti Gambar 4.



Gambar 4. Ilustrasi Kanal Multiplexing Data Wicara dan Data Teks WebSocket Sistem S2ST

ASR menerima data wicara melalui kanal *bytes* dan mengirimkan hasil transkripsi melalui kanal string. TTS mengirimkan hasil generasi suaranya sebagai file WAV melalui kanal bytes yang sebelumnya didahului oleh pengiriman informasi melalui kanal text/string tentang file yang dikirimkan. Untuk call management interaksi hanya dilakukan melalui kanal text/string.

Untuk kanal byte tidak diperlukan multiplexing lebih lanjut sebab data yang dialirkan hanya satu jenis saja. Sedangkan untuk kanal text/string dibutuhkan pemrosesan lebih lanjut. Dan untuk dapat dilakukan proses tersebut dibutuhkan rancangan format data yang dipertukarkan melalui kanal text/string tersebut.

**D. Rancangan Format Pertukaran Data**

Untuk dapat melakukan pemilahan data yang mengalir pada kanal text/string dibutuhkan format data yang sesuai. Dalam rancangan ini dipilih JSON sebagai format data exchange yang akan digunakan. Struktur dari pesan yang dipertukarkan adalah sebagai berikut:

```
{ "cmd": CMDNAME, "params": OBJECT, "session": SESSION_ID }
```

, dimana CMDNAME berisi informasi tentang jenis dari pesan tersebut, OBJECT berisi parameter dari pesan yang bergantung pada jenisnya. Sedangkan SESSION\_ID berisi nomor id dari sesi login pengguna atau dari sesi percakapan tergantung pada jenis pesannya.

Berikut adalah jenis pesan dan parameters yang akan digunakan:

- “CALLREQ” parameters { “callee” : USERID }  
Pesan ini dikirimkan dari Caller ke Server S2ST untuk memulai sebuah percakapan. Parameter dari pesan ini adalah User ID dari Callee.

Contoh:

```
{ "cmd": "CALLREQ", "params": { "callee": "user1@bpt.go.id", "session": "7719529d-eee6-4141-b67b-572c774300bf" }
```

- "CALLNOTIF" parameters { "caller" : USERID }  
Pesan ini akan dikirimkan oleh Server S2ST ke Callee. Parameter dari pesan ini adalah User ID dari Caller.

Contoh:

```
{ "cmd": "CALLNOTIF", "params": { "caller": "user1@bpt.go.id", "session": "2ced9345-1664-4278-b35a-cc868aa1ebdb" }
```

- "CALLRESP" parameters { "response": "ACCEPT" | "REJECT" }  
Pesan ini dikirimkan oleh Callee yang merupakan jawaban dari pesan CALLNOTIF. Parameter dari pesan ini berupa indikasi ACCEPT atau REJECT.

Contoh:

```
{ "cmd": "CALLRESP", "params": { "response": "ACCEPT", "session": "2ced9345-1664-4278-b35a-cc868aa1ebdb" }
```

- "TTSRESULT" parameters { "format": "WAV", "text": text, "size": SIZE\_IN\_BYTES }  
Pesan ini dikirimkan oleh Server S2ST yang mengawali proses pengiriman WAV file melalui kanal byte. Parameter dari pesan ini adalah format dari file yang dikirimkan, ukuran dari file tersebut dan *text* yang digunakan untuk menghasilkan WAV file tersebut.

Contoh:

```
{ "cmd": "TTSRESULT", "params": { "format": "WAV", "text": "selamat pagi", "size": 34502, "session": "2ced9345-1664-4278-b35a-cc868aa1ebdb" }
```

- "ASRRESULT" parameters { "hypothesis" : text, "isfinal" : TRUE|FALSE }  
Pesan ini dikirimkan oleh Server S2ST yang berisi hasil transkripsi dari data suara yang dikirimkan melalui WebSocket kanal *byte*. Parameter dari pesan ini adalah *hypohotesis* dan indikator apakah *hypohotesis* tersebut final atau bukan.

Contoh:

```
{ "cmd": "ASRRESULT", "params": { "hypothesis": "selamat pagi", "isfinal": "TRUE", "session": "2ced9345-1664-4278-b35a-cc868aa1ebdb" }
```

- "CALLEND" parameters { "reason" : TEXT\_REASON }  
Pesan ini dikirimkan oleh *Caller* atau *Callee* untuk menghentikan sesi percakapan. Parameter dari pesan ini adalah alasan mengapa sesi diterminasi.

Contoh:

```
{ "cmd": "CALLEND", "params": { "reason": "Conversation Completed", "session": "2ced9345-1664-4278-b35a-cc868aa1ebdb" }
```

- "ERROR" parameters { "message" : ERROR\_MESSAGE }  
Pesan ini dikirimkan oleh Server S2ST untuk menginformasikan jika terjadinya kesalahan secara umum. Parameter dari pesan ini adalah deskripsi dari error yang terjadi.

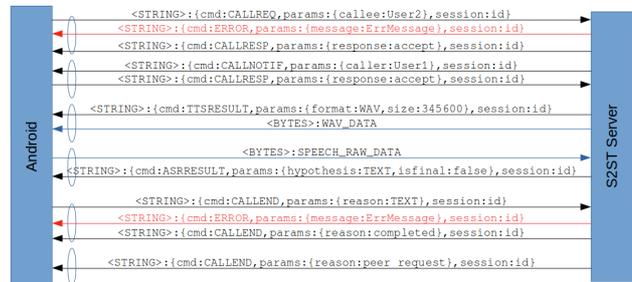
Contoh:

```
{ "cmd": "ERROR", "params": { "message": "UNKNOWN COMMAND", "session": "2ced9345-1664-4278-b35a-cc868aa1ebdb" }
```

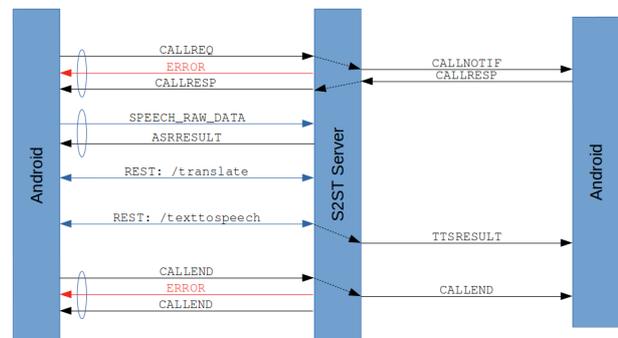
### E. Rancangan Proses Pertukaran Pesan

Dengan format pesan yang sudah ditentukan maka perlu ditetapkan penggunaan dari pesan-pesan tersebut. Gambar 5 adalah ilustrasi dari penggunaan pesan-pesan tersebut, dimana notasi <STRING> dan <BYTES> menunjukkan jenis kanal yang digunakan.

Seperti yang ditampilkan terdapat pasangan *request* dan *reply* dari jenis pesan tertentu misalkan: CALLREQ/CALLRESP dan CALLNOTIF/CALLRESP. Sedang pesan ERROR merupakan pesan umum yang dapat menjadi reply bagi beberapa jenis pesan. Pesan TTSRESULT merupakan pesan pembuka sebelum pengiriman data WAV file melalui kanal bytes. Sedangkan CALLEND mentrigger pengiriman pesan CALLEND kepada lawan bicara.



Gambar 5. Proses penggunaan pesan via WebSocket S2ST.



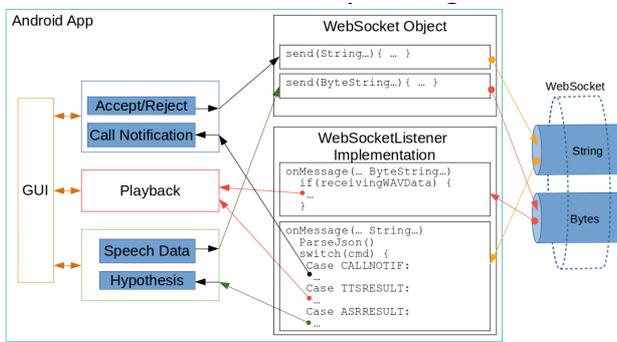
Gambar 6. Proses penggunaan pesan via WebSocket S2ST dan REST.

Proses percakapan diawali dengan pengiriman pesan CALLREQ oleh *Caller* yang kemudian mentrigger Server S2ST mengirimkan pesan CALLNOTIF ke *Callee*. *Callee* meresponse dengan pesan CALLRESP yang kemudian diteruskan ke *Caller*. *Caller* kemudian mulai berbicara dan ditranskripsikan oleh ASR dan hasil transkripsi dikirimkan melalui pesan ASRRESULT. *Caller* kemudian menggunakan layanan MT melalui RESTful API. Hasil dari MT kemudian dikirimkan ke TTS juga melalui RESTful API. Hasil dari TTS oleh Server S2ST dikirimkan ke *Callee* dengan pesan TTSRESULT yang kemudian diikuti dengan pengiriman data melalui kanal byte. *Caller* atau *Callee* dapat mengirimkan pesan CALLEND yang diteruskan kelawan bicara oleh Server S2ST yang mengindikasikan berakhirnya sesi percakapan.

F. Rancangan Pengolahan Pesan Pada Android dan Server S2ST

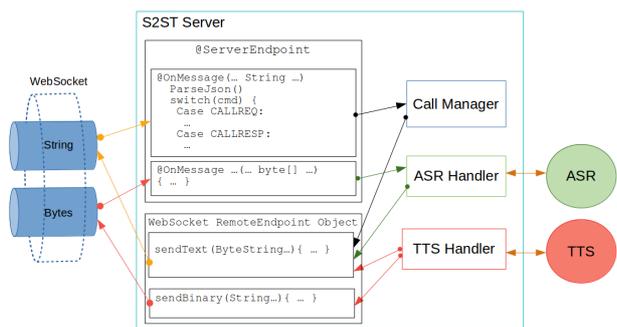
Pesan yang diterima dari WebSocket harus dapat diolah oleh aplikasi Android maupun Server S2ST secara benar. Gambar 7 adalah rancangan yang dapat dijadikan acuan untuk implementasi proses pengolahan pesan pada Android yang menggunakan library **OkHttp3** (Github, 2018) sebagai *WebSocket client*.

Dalam *OkHttp3* object *WebSocket* dapat digunakan sebagai *WebSocket client* dimana didalamnya terdapat metode *send* yang dapat digunakan untuk mengirim *byte* atau mengirim string. Sedangkan untuk proses penerimaan dilakukan dengan mendaftarkan *listener* yang dipanggil jika kanal string atau kanal *byte* menerima data. Untuk kanal *byte* tidak dibutuhkan proses pemilahan karena data yang masuk sudah bisa dipastikan adalah WAV file hasil dari TTS yang dikirimkan oleh Server S2ST. Untuk kanal string dibutuhkan proses pemilahan penanganan. Dengan memanfaatkan informasi jenis pesan yang diterima maka pesan dapat diolah sesuai dengan jenisnya.



Gambar 7. Proses pengolahan pesan S2ST menggunakan library *OkHttp3*

Untuk Server S2ST, Gambar 8 adalah rancangan proses pengolahan pesan yang masuk melalui WebSocket.



Gambar 8. Proses penggunaan pesan masuk via WebSocket

Pada Server S2ST yang diimplementasikan sebagai endpoint pada JEE Glashfish Server, maka dapat dibuat method yang menangani kanal byte dan kanal string secara terpisah. Untuk kanal byte tidak dibutuhkan pemilahan karena data yang masuk dapat dipastikan berupa data suara yang harus diteruskan ke penyedia layanan ASR. Sedangkan kanal string dapat dilakukan pemilahan menggunakan informasi jenis pesan yang diterima. Pada JEE Server proses pengiriman data dilakukan menggunakan object *RemoteEndpoint* yang dapat diambil

referensinya dari *WebSocket* object *Session*. Pengiriman ke kanal string dilakukan menggunakan metode *sendText* sedangkan pengiriman melalui kanal byte dilakukan dengan metode *sendBinary*.

IV. KESIMPULAN

Makalah ini mendeskripsikan hasil kerekayasaan Teknologi Bahasa BPPT yang berupa rancangan Aplikasi Mobile Android *Speech-to-Speech Translation (S2ST)*. Aplikasi ini memfasilitasi penerjemahan wicara dari komunikasi dua arah (percakapan antara dua pengguna). Aplikasi menerapkan beberapa teknik, seperti: *WebSocket*, *RESTful service*, *JSON*, dan *OkHttp3*.

Kelanjutan dari rancangan ini adalah kegiatan implementasi yang sedang dalam pengerjaan hingga tulisan ini dikeluarkan, sehingga tidak menutup kemungkinan hasil implementasi akan memberikan *feedback* terhadap revisi rancangan.

PENGAKUAN

Publikasi makalah ini didanai oleh DIPA BPPT dan Insentif Riset Pusat Unggulan Inovasi (PUI) PTIK-BPPT dari Kementerian RistekDikti.

DAFTAR PUSTAKA

Baidu. (2018). Baidu Translator. Retrieved November 2, 2018, from <https://fanyi.baidu.com/>

Bott, R. (2011). RFC 6455 - The WebSocket Protocol, (1), 1–5. <https://doi.org/10.1007/s13398-014-0173-7.2>

Bray, T. (Ed.). (2014). *The JavaScript Object Notation (JSON) Data Interchange Format*. <https://doi.org/10.17487/rfc7159>

Github. (2018). *OkHttp-An HTTP & HTTP/2 client for Android and Java applications*. Retrieved October 4, 2018, from <https://square.github.io/okhttp/>

Google. (2015). Google Translate. Retrieved from <https://translate.google.com/>

Google LLC. (2018). Google Translate – Apps on Google Play. Retrieved November 2, 2018, from [https://play.google.com/store/apps/details?id=com.google.android.apps.translate&hl=en\\_IN](https://play.google.com/store/apps/details?id=com.google.android.apps.translate&hl=en_IN)

Grigorik, I. (2013). *High Performance Browser Networking*. O'Reilly Media. Retrieved from <https://hpbn.co/>

Gunarso, Riza, H., Nurfadhilah, E., Teduh Uliniansyah, M., Santosa, A., & Aini, L. R. (2016). An Overview of BPPT's Indonesian Language Resources, 73–77. Retrieved from <http://www.aclweb.org/anthology/W16-5409>

Gunawan, M., Nurfadhilah, E., Aini, L. R., Uliniansyah, M. T., -, G., Santosa, A., & Junde, J. (2018). Uji Coba Korpus Data Wicara BPPT sebagai Data Latih Sistem Pengenalan Wicara Bahasa Indonesia. *Jurnal Linguistik Komputasional (JLK)*, 1(2), 45. <https://doi.org/10.26418/jlk.v1i2.8>

iTranslate. (2018). iTranslate Converse - A Revolutionary New Speech Translation App. Retrieved December 4, 2018, from <https://www.itranslate.com/converse>

Lavie, A., Langley, C., Waibel, A., Pianesi, F., Lazzari, G., Coletti, P., ... Balducci, F. (2001). Architecture and Design Considerations

- in {NESPOLE!}: a Speech Translation System for {E}-commerce Applications. *HLT 2001 - Human Language Technology Conference*, 1–4. <https://doi.org/10.1002/dvdy.22350>. TOWARDS
- Lavie, A., Waibel, A., Levin, L., Finke, M., Gates, D., Gavalda, M., ... Zhan, P. (1997). Janus-III: Speech-to-Speech Translation in Multiple Languages. *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1*, 99–102. <https://doi.org/10.1109/ICASSP.1997.599557>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015, May 28). Deep learning. *Nature*. Nature Publishing Group. <https://doi.org/10.1038/nature14539>
- Meermeier, R., Colbath, S., & Of, M. L. (2018). Portable speech-to-speech translation on an Android smartphone: The MFLTS system. In *aclweb.org* (Vol. 2, pp. 303–308). Proceedings of the 13th Conference 2018.
- Microsoft. (2018). Conversations - Microsoft Translator. Retrieved December 4, 2018, from <https://translator.microsoft.com/>
- Riandi, O., Jarin, A., Uliniansyah, M. T., Santosa, A., & Prasetyo, B. (2010). BPPT Ciptakan Perisalah dan Sistem Notulen Otomatis. *Kompas.com*.
- Richardson, L., & Ruby, S. (2007). *RESTful web services*. O'Reilly Media.
- Santosa, A., Jarin, A., Gunawan, M., Uliniansyah, M. T., Gunarso, Nurfadhilah, E., ... Riza, H. (2018). Utilizing Indonesian Data Resources for Text-to-Speech Using End-to-End Method. In *Oriental COCOSDA 2018*. Miyazaki, Japan: naist.
- TripLingo. (2018). TripLingo Voice Translator. Retrieved November 2, 2018, from <http://www.triplingo.com/>
- U-STAR. (2016). uniTRANS. Retrieved from <http://www.ustar-consortium.com/qws/slot/u50227/app/app.html>
- U-STAR Consortium. (2016). U-STAR. Retrieved March 16, 2018, from <http://www.ustar-consortium.com/qws/slot/u50227/research.html>
- Uliniansyah, T., Riza, H., & Riandi, O. (2013). Developing corpus management system for Bahasa Indonesia the “Perisalah” project. In *2013 International Conference Oriental COCOSDA Held Jointly with 2013 Conference on Asian Spoken Language Research and Evaluation, O-COCOSDA/CASLRE 2013* (pp. 1–4). IEEE. <https://doi.org/10.1109/ICSODA.2013.6709887>
- Wahlster, W. (2000). Mobile Speech-to-Speech Translation of Spontaneous Dialogs: An Overview of the Final Verbmobil System. In *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 3–21). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-662-04230-4\\_1](https://doi.org/10.1007/978-3-662-04230-4_1)
- World Wide Web Consortium (W3C). (2018). Web Services Architecture.