

PEMANFAATAN WEB SERVICES PADA APLIKASI TELEMEDICINE

Tigor H. Nasution, F. Fahmi

Program Magister Teknik Elektro, Fakultas Teknik Universitas Sumatera Utara
E-mail: tigor.nasution@usu.ac.id, fahmimn@usu.ac.id

ABSTRAK

Kondisi geografis Indonesia yang amat luas menuntut dikembangkannya aplikasi telemedicine dalam peningkatan pelayanan kesehatan. Topik yang diangkat adalah perancangan pengolahan citra medis berbasis web services. Berbeda dengan aplikasi standalone, sistem yang dirancang dapat digunakan oleh banyak pengguna sehingga dapat mengurangi biaya pengadaan komputer berkemampuan tinggi yang dibutuhkan dalam pengolahan citra medis. Sistem yang dirancang bekerja di berbagai platform dengan antar muka berbasis web dan dibangun menggunakan teknologi web services XML-RPC. Sistem terdiri dari aplikasi server dan aplikasi klien berbasis Java dan ITK. Pengujian dilakukan dengan membandingkan kinerja sistem yang dibangun dengan aplikasi standalone untuk pengolahan segmentasi Otsu citra otak. Diperoleh hasil pengolahan citra yang identik sama, dengan waktu yang dibutuhkan untuk proses lebih cepat 15.34 detik dari proses yang dilakukan secara standalone. Sistem ini dapat diterapkan untuk aplikasi telemedicine dalam pengolahan citra medis.

Kata-kata kunci: web services, XML-RPC, telemedicine pengolahan citra medis

1. Pendahuluan

1.1 Latar Belakang dan Tujuan

Dewasa ini penggunaan sistem berbasis jaringan semakin banyak dikembangkan untuk aplikasi telemedicine. Terutama di Indonesia yang memiliki daerah yang sangat luas, perkembangan di bidang ini dituntut untuk terus dilakukan dalam rangka peningkatan pelayanan kesehatan.

Salah satu pemanfaatan jaringan adalah teknologi web services. Web services menggunakan standar yang tidak terikat pada platform (*platform-neutral*) dan tidak terikat pada bahasa pemrograman yang digunakan (*language-neutral*) [1] [2]. Dengan demikian, web services memudahkan beberapa aplikasi atau komponennya untuk saling berhubungan dengan aplikasi lain dalam sebuah organisasi maupun di luar organisasi.

Di lain pihak, seiring dengan perkembangan teknologi kedokteran, perkembangan pengolahan citra medis juga semakin pesat. Hal ini dapat dilihat dengan banyaknya aplikasi yang dikembangkan untuk pemrosesan hasil citra medis dari peralatan kesehatan seperti *Computed Tomography* (CT), *Magnetic Resonance Imaging* (MRI), *Positron Emission Tomography* (PET) dan sistem *x-ray* [3]. Dengan aplikasi pengolahan citra medis, data citra medis dapat divisualisasikan dan dianalisis untuk keperluan diagnostik. Namun untuk melakukan hal tersebut, dibutuhkan infrastruktur peralatan komputer standalone dengan kemampuan dan kualifikasi yang tinggi yang mampu melakukan pemrosesan citra dengan cepat.

Topik yang diangkat pada penelitian ini adalah perancangan pengolahan citra medis berbasis web service. Berbeda dengan aplikasi standalone, sistem yang dirancang dapat digunakan secara jarak jauh, dapat digunakan oleh banyak pengguna sehingga dapat mengurangi biaya dalam pengadaan

infrastruktur hardware berkemampuan tinggi yang dibutuhkan dalam pengolahan citra medis.

Sistem yang dirancang dapat bekerja di berbagai macam platform dengan grafis antar muka berbasis web dan dibangun dengan menggunakan teknologi web services.

1.2 Penelitian Terkini

Beberapa penelitian terkait yang telah dilakukan dalam bidang ini antara lain penelitian yang dilakukan oleh Spiros Koulouzis, dkk [4] yang mencoba transportasi data medis menggunakan web services. Dalam penelitiannya, mereka menggunakan *Simple Object Access Protocol* (SOAP) yang merupakan salah satu protokol berbasis *Extensible Markup Language* (XML) [5]. Pada penelitian lain yang dilakukan oleh Sayed Ehsan Mahmodi, dkk [6] telah dirancang suatu sistem visualisasi citra medis yang berbasis web menggunakan bahasa pemrograman murni, tanpa menggunakan *applet*. Java Applet yang merupakan aplikasi menggunakan bahasa pemrograman Java yang dapat berjalan pada sebuah browser [7].

Penelitian lainnya, oleh Danzhou Liu, dkk [8] dirancang sebuah *framework* untuk pengolahan citra medis dalam jaringan internet yang difokuskan pada rekonstruksi citra medis 3D dengan resolusi tinggi. Sedangkan pada penelitian yang dilakukan oleh Jianguo Zhang, dkk [9] turut membahas masalah keamanan komunikasi citra medis pada jaringan internet.

2. Metode dan Bahan

2.1 Proses Pengolahan Citra Medis

Ada beberapa tahapan yang dilakukan pada pengolahan citra medis. Proses yang paling sering digunakan adalah segmentasi citra medis yang

digunakan untuk membagi citra medis menjadi beberapa *region of interest* sesuai dengan kebutuhan. Berbagai macam metode segmentasi yang telah berkembang saat ini. Pengolahan citra medis yang akan dikembangkan pada penelitian ini adalah segmentasi citra medis dengan metode Otsu. Hal ini dilakukan berdasarkan penelitian yang telah dilakukan sebelumnya oleh W. Mimi Diyana W. Zaki, dkk [10]. Penggunaan metode Otsu dilakukan untuk multi level segmentasi pada citra otak dari CT-Scan. Metode Otsu merupakan salah satu metode segmentasi dengan mengadopsi algoritma *region growing*. Algoritma *region growing* digunakan karena algoritma ini merupakan pendekatan yang efektif untuk segmentasi citra dengan homogenitas tertentu [11]. Pemilihan metode Otsu dikarenakan metode ini termasuk metode sederhana yang baik dan cepat dalam melakukan segmentasi citra [12], sehingga dianggap sesuai dengan sistem pengolahan citra berbasis jaringan yang dikembangkan.

Metode Otsu bekerja berdasarkan data histogram dari suatu citra. Histogram pada suatu citra menunjukkan keseluruhan nilai intensitas dari setiap *pixel* pada citra dalam satu dimensi. Pada histogram, sumbu x menyatakan level intensitas yang berbeda sedangkan sumbu y menyatakan jumlah akumulatif *pixel* yang memiliki nilai intensitas tersebut. Dengan menggunakan histogram dapat dilakukan pengelompokan *pixel-pixel* dalam suatu citra medis. Pengelompokan ini didasarkan pada nilai ambang batas atau *threshold*. Nilai *threshold* ini menjadi parameter utama dari metode Otsu.

2.2 Data Citra Medis

Data citra medis yang digunakan dalam penelitian ini berupa citra medis 2D dari otak manusia yang merupakan hasil *scan* MRI. Data didapat dari kumpulan librari citra medis yang terdapat pada situs <http://www.osirix-viewer.com/datasets/>. Hasil *scan* MRI tersebut berupa citra dengan format DICOM sebagai format standar citra medis. Data citra medis tersebut nantinya akan disegmentasi untuk mendapatkan nilai *threshold* yang optimal yang dapat digunakan dalam analisa kondisi pasien. Pengiriman data citra medis terlebih dahulu dilakukan dengan membuang data identitas pasien yang secara hukum tidak boleh didistribusikan di luar kewenangan dan persetujuan pasien yang bersangkutan.

2.3 Komunikasi Data Citra Medis

Dalam melakukan distribusi pengolahan data citra medis maka diperlukan suatu mekanisme komunikasi data citra medis pada sistem berbasis jaringan. Beberapa penelitian sebelumnya telah menjelaskan komunikasi data citra medis pada sistem berbasis jaringan. Spiros Koulouzis, dkk [4] telah mendemonstrasikan penggunaan *web services*

dalam komunikasi data citra medis pada sistem berbasis jaringan.

Terdapat berbagai macam teknologi *web services* yang berkembang saat ini, SOAP dan XML-RPC merupakan teknologi *web services* yang populer. Dalam penelitian ini digunakan teknologi *web services* XML-RPC karena dianggap lebih efektif dibandingkan dengan teknologi SOAP dalam pemanggilan prosedur pada jaringan. Hal ini disebabkan karena XML-RPC membungkus data yang dikirim ke dalam format XML serta dapat mendefinisikan kembali tipe data yang dibungkus sesuai dengan bahasa pemrograman yang digunakan. Berbeda dengan SOAP yang memerlukan pembuatan WSDL terlebih dahulu untuk pembungkusan data ke dalam format XML.

XML-RPC dapat dirancang dengan menggunakan berbagai macam bahasa pemrograman seperti C++, Java, Python, dan lain-lain. Dalam penelitian ini dilakukan perbandingan antara bahasa pemrograman C++ dan bahasa pemrograman Java untuk merancang XML-RPC. Perbandingan dilakukan untuk melihat efisiensi dan efektifitas dari bahasa pemrograman untuk merancang suatu aplikasi XML-RPC. Dari pengujian yang dilakukan, untuk pengembangan sistem ini bahasa pemrograman Java lebih unggul dibandingkan dengan bahasa pemrograman C++. Hal ini karena dengan menggunakan bahasa pemrograman C++ diperlukan suatu *framework* tambahan untuk membangun XML-RPC dan terdapat beberapa konfigurasi yang harus dilakukan dalam menjalankan *framework* tersebut. Hal ini membuat penulisan kode menggunakan bahasa pemrograman C++ jauh lebih kompleks dibandingkan dengan bahasa pemrograman Java yang hanya memerlukan librari tambahan yang dikembangkan oleh Apache untuk membangun teknologi XML-RPC (Apache XML-RPC).

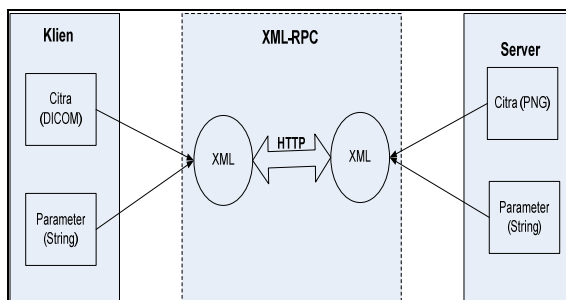
Paket Apache XML-RPC tersebut akan di-*import* kedalam aplikasi klien dan aplikasi server yang nantinya akan melakukan proses komunikasi. Berdasarkan pertimbangan tersebut maka dalam penelitian ini dipilih bahasa pemrograman Java untuk membangun teknologi *web services*.

Pada Apache XML-RPC, aplikasi klien dan aplikasi server berkomunikasi dengan menggunakan protokol HTTP. Protokol ini menggunakan nomor *port* yang berbeda dari nomor *port* yang standar digunakan. Hal ini dilakukan untuk mengakomodasi masalah keamanan pada jaringan komputer yang merupakan salah satu persyaratan mutlak dalam pengolahan citra medis.

Selain itu penggunaan nomor *port* yang berbeda juga bertujuan untuk membedakan jalur komunikasi dengan jalur yang biasa digunakan. Untuk menggunakan protokol HTTP, pada sisi server dapat digunakan librari untuk menggunakan *framework* server web tertentu atau menggunakan HTTP server yang di-*built-in* pada aplikasi server.

Komunikasi data yang terjadi pada sistem ini adalah pengiriman citra medis beserta parameternya. Baik aplikasi klien dan server sama-sama melakukan pengiriman citra medis. Aplikasi Klien akan mengirimkan citra medis yang akan diolah dalam format DICOM beserta parameter yang diinginkan. Aplikasi server selanjutnya akan mengirimkan hasil pengolahan tersebut, akan tetapi tidak dalam format DICOM. Hal ini disebabkan karena ukuran data yang cukup besar yang akan mempengaruhi lamanya komunikasi data yang terjadi. Untuk mengatasi masalah tersebut, maka data yang dikirim oleh server telah dipadatkan dan diubah formatnya menjadi citra dengan format PNG (*Portable Network Graphics*). Pemadatan ini tidak akan menghilangkan bagian inti dari data citra yang ada di dalamnya. Data citra inilah yang sebenarnya dibutuhkan untuk proses pengolahan lebih lanjut.

Secara alamiah XML-RPC tidak mendukung tipe data citra dalam proses komunikasi. Untuk mengatasi masalah tersebut maka sebelum dilakukan pengiriman data citra terlebih dahulu dilakukan konversi menjadi data *binary*. Dalam bahasa pemrograman Java data *binary* didefinisikan menjadi tipe data *byte[]*. Akan tetapi dalam XML-RPC tipe data *byte[]* tersebut akan dikonversikan menjadi tipe data *base64*. Untuk parameter data yang dikirimkan berupa data dengan tipe data *string*. Proses transportasi data yang terjadi dapat dilihat pada Gambar 1.



Gambar 1. Transportasi Data Dengan XML-RPC

Menggunakan Apache XML-RPC, komunikasi antara klien dan server dapat beroperasi dalam mode *streaming*. Dengan menggunakan mode *streaming* pemeliharaan sumber daya jauh lebih baik daripada mode standar yang berdasarkan pada besarnya *array byte* internal. Pembagian sumber daya bersama ketika lebih dari satu klien berkomunikasi dengan server akan dilakukan secara paralel dengan menggunakan eksekusi *multiple threads*. Hal ini memungkinkan penggunaan server oleh banyak klien secara bersamaan dengan kompensasi waktu proses pengolahan citra medis yang lebih lama.

3. Perancangan Sistem

Sistem yang dibangun pada topik ini terdiri dari dua buah aplikasi yaitu aplikasi server dan aplikasi klien. Pembuatan kedua aplikasi tersebut sama-sama menggunakan bahasa pemrograman Java. Kedua aplikasi menggunakan librari *web services XML-RPC* untuk berkomunikasi antara klien dan server.

3.1 Pembuatan Aplikasi Server

Pembuatan aplikasi server dimulai dengan pembuatan prosedur untuk melakukan segmentasi. Metode segmentasi yang dilakukan adalah metode segmentasi Otsu. Algoritma segmentasi Otsu menggunakan algoritma yang telah tersedia pada librari ITK [11]. Prosedur segmentasi akan menerima empat parameter yang dilewatkan, yaitu citra medis, nama *file* hasil segmentasi, jumlah iterasi segmentasi dan tipe *file* hasil segmentasi.

Prosedur segmentasi ini akan mengembalikan citra medis dan nilai *threshold* hasil segmentasi yang dipaketkan dalam variabel objek *Vector* pada bahasa pemrograman Java.

Selanjutnya dilakukan pembuatan prosedur XML-RPC *Listener* yang akan menerima *request* dari aplikasi klien dan meng-*import* paket *org.apache.xmlrpc.XmlRpcServer* untuk menggunakan objek *WebServer*. Objek ini akan mendefinisikan aplikasi sebagai server ketika dipanggil dengan melewati parameter berupa nomor *port* (tipe data *int*) yang akan digunakan. Selanjutnya objek *WebServer* akan memanggil *method addHandler()* untuk mendefinisikan nama objek server XML-RPC.

Untuk mulai menjalankan server objek *WebServer* akan memanggil *method start()*;

XML-RPC *Listener* menerima paket data XML dari klien yang berisi *method* yang akan dipanggil beserta parameter yang dilewatkan. XML-RPC *Listener* akan mem-*parsing* parameter-parameter yang ada dan mendefinisikan tipe data pada paket XML menjadi tipe data pada bahasa pemrograman Java.

3.2 Pembuatan Aplikasi Klien

Aplikasi klien dibuat dengan menggunakan Java Applet. Hal ini dilakukan agar aplikasi dapat berjalan pada sebuah *browser* sehingga tidak memerlukan instalasi dan dapat berjalan pada lintas *platform*. Pembuatan aplikasi klien terdiri dari pembuatan grafis antarmuka serta pembuatan prosedur pemanggilan *method web services* pada server.

Pembuatan grafis antar muka dilakukan dengan meng-*import* paket *javax.swing* pada bahasa pemrograman Java. Komponen grafis antar muka terdiri dari *filechooser*, *radio button*, *text area*, *label*, dan *button*. *Filechooser* digunakan untuk mencari *file* citra medis yang akan disegmentasi

pada komputer lokal. Pada grafis antar muka terdapat tiga *radio button* untuk menentukan parameter nilai dari iterasi segmentasi. *Text area* digunakan untuk menampilkan nilai *threshold* dari segmentasi yang dilakukan. Selanjutnya terdapat dua komponen *button* digunakan untuk melakukan aksi proses pemanggilan *method* segmentasi pada server dan melakukan penyimpanan hasil proses segmentasi. Sedangkan komponen *label* digunakan untuk menampilkan citra yang disegmentasi. Komponen *label* juga diberikan aksi untuk memilih citra hasil segmentasi untuk dilakukan segmentasi lagi pada proses segmentasi multilevel.

Prosedur pemanggilan *method* segmentasi dilakukan dengan meng-*import* paket *org.apache.xmlrpc.XmlRpcKlien*. Dari paket tersebut digunakan objek *XmlRpcKlien* dengan melewati parameter URL dari server pada konstruktor objek tersebut. Pada prosedur ini juga digunakan objek *Vector* untuk memuat seluruh variabel yang akan dijadikan parameter. Selanjutnya objek *XmlRpcKlien* akan memanggil *method execute* dengan melewati parameter yang terdiri *method* yang akan dipanggil dan objek *Vector* yang menyimpan parameter yang akan dilewatkan pada objek tersebut. Parameter yang dikirim oleh klien akan dibungkus menjadi data XML.

Setelah server mengembalikan nilai *response* maka aplikasi pada klien akan mem-*parsing* data XML yang diterima. Data XML yang diterima akan dikonversi menjadi tipe data yang ada pada bahasa pemrograman Java. Dalam aplikasi ini nilai yang diterima berupa data citra dengan tipe data *byte[]* dan nilai *threshold*. Tipe data citra tersebut akan dikonversi menjadi tipe data *Image* agar bisa ditampilkan dan disimpan pada komputer lokal.

4. Pengujian Sistem

Pengujian dilakukan dengan menggunakan sebuah citra medis dari organ bagian otak. Pengujian dimulai dengan melakukan proses segmentasi dari aplikasi *standalone*. Kemudian dilakukan pengujian pada sistem berbasis jaringan untuk melihat hasil segmentasi yang didapat. Data citra medis yang didapat pada sistem berbasis jaringan dibandingkan dengan data hasil pengujian pada aplikasi *standalone*.

Selanjutnya dilakukan pengujian pada sistem berbasis jaringan dengan menggunakan multi klien: lima sampai dua puluh klien secara bersamaan dengan pertambahan lima klien pada tiap tahap pengujian. Hal ini dilakukan untuk melihat waktu proses segmentasi yang dilakukan untuk pengujian dengan banyak klien.

Pengujian dilakukan pada jaringan lokal dengan *bandwith* sebesar 100 MBps. Pengujian menggunakan dua jenis komputer yaitu komputer server untuk aplikasi server dan komputer klien untuk aplikasi klien.

Pengujian juga dilakukan untuk jumlah *threshold* bertingkat 2, 3, dan 4 level. Dari pengujian akan dilihat hasil segmentasi dan lamanya waktu proses segmentasi yang dibutuhkan untuk masing-masing jumlah *threshold*.

Untuk aplikasi *standalone* digunakan tiga jenis komputer yaitu komputer server, Klien 1 (Komputer PC) dan Klien 2 (Notebook). Spesifikasi komputer server yang digunakan adalah *IBM System X 3200 M3* dengan prosesor *Intel® Xeon® x3400 Series (quad-core) 2.4 GHz* dengan Memory sebesar 2GB. Sedangkan spesifikasi komputer klien yang digunakan adalah *Personal Lenovo ThinkCentre* dengan prosesor *Intel Core 2 Duo Prosesor 2.7 GHz* dengan memory sebesar 2GB DDR2.

Pada pengujian akan dilakukan perhitungan prediksi penambahan waktu untuk setiap penambahan klien. Prediksi rata-rata penambahan waktu dihitung dengan rumus berikut:

$$\bar{t} = \frac{\sum_{t=1}^n \Delta t}{n} \dots\dots\dots(1)$$

dimana:

- \bar{t} = rata-rata penambahan waktu.
- Δt = perubahan waktu/penambahan waktu.
- n = tingkat penambahan.

5. Hasil

Hasil dari perancangan sistem berbasis jaringan berupa prototype-prototipe sistem yang terdiri dari aplikasi server dan aplikasi klien. Aplikasi server berupa aplikasi berbasis *command line* sedangkan aplikasi klien berupa aplikasi berbasis web dengan menggunakan java applet.

Pengujian pada sistem berbasis jaringan dilakukan dengan menggunakan data yang sama dengan pengujian pada aplikasi *standalone*, yaitu dengan citra medis dari otak manusia dan diuji untuk tiap jumlah *threshold* yang berbeda. Data perbandingan hasil pengujian waktu pemrosesan yang dilakukan dapat dilihat pada Tabel 1.

Tabel 1. Perbandingan Hasil Waktu Pengujian

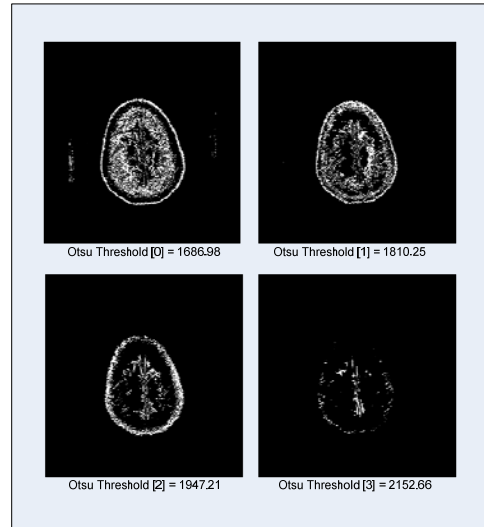
Jumlah <i>Threshold</i>	Waktu Prototipe Sistem	Waktu Server	Waktu Klien 1	Waktu Klien 2
2	0.547	0.269	0.593	0.812
3	0.797	0.460	1.125	2.11
4	5.781	5.410	15.86	21.125

Aplikasi *standalone* pada komputer server sedikit lebih cepat dari pada prototipe sistem. Sedangkan aplikasi *standalone* yang menggunakan Klien 1 ataupun Klien 2 waktu proses yang

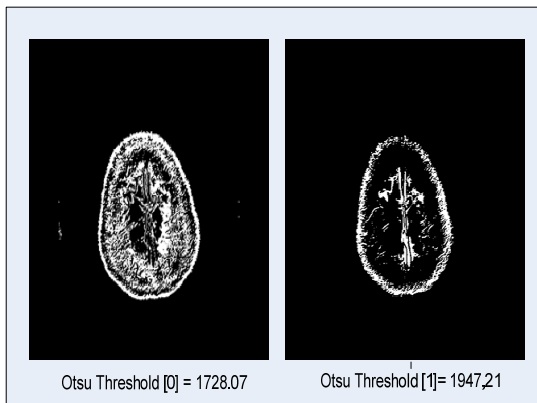
dibutuhkan lebih lama dari waktu proses pada prototipe sistem.

Untuk citra hasil segmentasi memiliki hasil yang sama untuk pada setiap percobaan yang dilakukan pada *platform* yang berbeda. Citra hasil segmentasi dengan metode Otsu untuk setiap jumlah *threshold* yang berbeda dapat dilihat pada Gambar 2,3, dan 4. Karena segmentasi menggunakan metode Otsu maka citra yang dihasilkan berupa citra hitam putih.

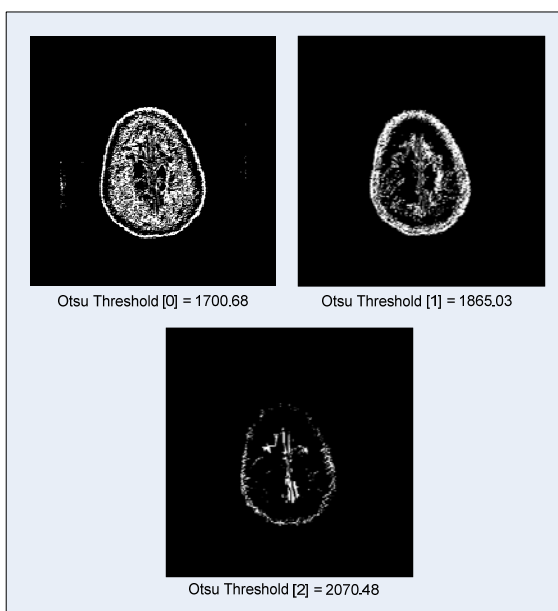
Pengujian multi klien dilakukan untuk melihat pengaruh jumlah aplikasi klien yang melakukan proses segmentasi pada waktu bersamaan. Pada pengujian ini dilihat waktu yang dibutuhkan untuk tiap jumlah klien yang berbeda. Pengujian dilakukan dengan lima klien sampai dengan dua puluh klien. Dengan jumlah klien tersebut pengujian ini dilakukan sebanyak empat tahap dengan penambahan lima klien.



Gambar 4. Citra Hasil Segmentasi dengan 4 Threshold



Gambar 2. Citra Hasil Segmentasi dengan 2 Threshold

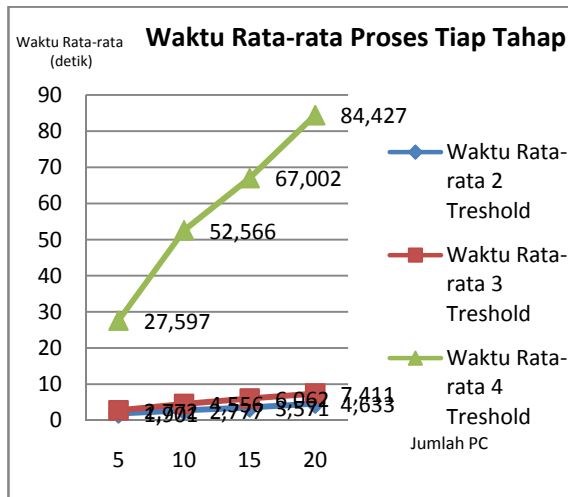


Gambar 3. Citra Hasil Segmentasi dengan 3 Threshold

Dari empat tahap pengujian yang dilakukan terdapat peningkatan rata-rata waktu yang dibutuhkan untuk setiap penambahan klien. Nilai rata-rata waktu proses pada tiap tahap dapat dilihat pada Tabel 2 dan digambarkan dengan grafik pada Gambar 5. Dari data yang didapat, pada dua nilai *threshold* dan tiga nilai *threshold* peningkatan waktu yang terjadi tidak terlalu signifikan. Akan tetapi pada empat nilai *threshold* penambahan waktu yang terjadi cukup besar.

Tabel 2. Waktu Rata-rata Proses Tiap Tahap

Jumlah Klien	Waktu Rata-rata 2 Threshold (detik)	Waktu Rata-rata 3 Threshold (detik)	Waktu Rata-rata 4 Threshold (detik)
5	1.901	2.772	27.597
10	2.777	4.556	52.566
15	3.571	6.062	67.002
20	4.633	7.411	84.427



Gambar 5. Grafik Waktu Rata-rata Proses Tiap Tahap

Dari grafik gambar 5 dapat dilihat pertambahan secara linear waktu rata-rata tiap tahap. Penambahan waktu yang terjadi setiap terjadi penambahan lima klien dapat diprediksi dari rata-rata selisih pertambahan waktu yang terjadi pada tiap tahap. Prediksi waktu penambahan per 5 klien untuk setiap jumlah *treshold* (2, 3, dan 4) berdasarkan rumus (1) adalah:

$$\bar{t}_2 = 0.911 \text{ detik}$$

$$\bar{t}_3 = 1.546 \text{ detik}$$

$$\bar{t}_4 = 18.943 \text{ detik}$$

6. Kesimpulan

Berdasarkan dari pengujian yang dilakukan, prototipe sistem dapat menghasilkan pengolahan citra medis yang sama dengan aplikasi *standalone*.

Untuk satu klien pada prototipe sistem, waktu yang dibutuhkan tidak terlalu berbeda dengan aplikasi *standalone* pada komputer server. Akan tetapi pada aplikasi *standalone* menggunakan PC atau *notebook* waktu proses yang dibutuhkan lebih lama sampai dengan 15.34 s untuk proses yang lebih kompleks dibandingkan dengan waktu proses yang dibutuhkan prototipe sistem.

Untuk jumlah klien lebih dari satu akan terjadi penambahan waktu proses seiring dengan penambahan klien. Proses penambahan waktu untuk tiap penambahan klien dipengaruhi oleh kemampuan dari komputer server. Untuk spesifikasi yang lebih tinggi diperkirakan penambahan waktu rata-rata akan lebih kecil. Hal ini belum dibuktikan pada penelitian ini karena keterbatasan waktu dan sumber daya yang ada.

Dari percobaan yang dilakukan disimpulkan prototipe sistem yang dikembangkan dapat menjadi alternatif untuk sistem pengolahan citra medis secara jarak jauh dengan kualitas hasil pengolahan citra yang sama. Kelebihan sistem ini adalah tidak memerlukan instalasi aplikasi pada tiap klien yang

ingin melakukan pengolahan citra medis. Selain itu komputer yang digunakan klien juga tidak harus memiliki kemampuan yang tinggi seperti pada komputer server. Karena menggunakan aplikasi berbasis web, aplikasi klien pada prototipe sistem dapat berjalan diberbagai jenis *platform* yang mendukung penggunaan web *browser* dengan *applet*.

Untuk pengembangan selanjutnya, prototipe sistem dapat dikembangkan untuk jaringan internet. Hal ini dapat dilakukan karena komunikasi yang terjadi pada prototipe sistem menggunakan protokol standar HTTP yang juga dapat digunakan dalam komunikasi pada jaringan internet. Selain itu dimungkinkan juga untuk pengembangan aplikasi klien pada perangkat bergerak (*mobile device*) tanpa mengganggu aplikasi pada server.

Referensi

- Doug Tidwell, James Snell, and Pavel Kulchenko, *Programming Web Services with SOAP*, First Edition ed. Sebastopol, Canada: O'Reilly, 2001.
- Ade Anom A. (2005) IlmuKomputer.Com. [Online]. <http://ikc.depsos.go.id/populer/adeanom/adeanom-axis.zip>
- John Stoitsis et al., "Computer aided diagnosis based on medical image processing and artificial intelligence methods," *Nuclear Instruments and Methods in Physics Research*, pp. 591-595, September 2006.
- Spiros Koulouzis, Elena Zudilova-Seinstra, and Adam Belloum, "Data Transport between Visualization Web Services for Medical Image Analysis," in *International Conference on Computational Science, ICCS 2010*, 2010.
- W3C. (2007, April) World Wide Web Consortium (W3C). [Online]. <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>
- Seyyed Ehsan Mahmoudi et al., "Web-based interactive 2D/3D medical image processing and visualization software," *Computer Methods and Programs in Biomedicine*, pp. 172-182, November 2010.
- E.S. Boese, *Java Applets (2nd edition) B&W: Interactive.*: Lulu.com, 2007.
- Danzhou Liu, Kien A. Hua, and Kiminobu Sugaya, "A Generic Framework for Internet-Based Interactive Applications of High-Resolution 3-D Medical Image Data," *IEEE Transactions On Information Technology In Biomedicine*, vol. 12, pp. 618-626, September 2008.
- Jianguo Zhang, Fenghai Yu, Jianyong Sun,

- Yuanyuan Yang, and Chenwen Liang, "DICOM Image Secure Communications With Internet Protocols IPv6 and IPv4," *IEEE Transactions On Information Technology In Biomedicine*, vol. 11, pp. 70-80, January 2007.
- W. Mimi Diyana W. Zaki, M. Faizal A. Fauzi, R. Besar, and W.S.H Munirah W. Ahmad, "Qualitative and Quantitative Comparisons of Haemorrhage Intracranial Segmentation in CT Brain Images," in *TENCON 2011 - 2011 IEEE Region 10 Conference*, Bali, 2011, pp. 369-373.
- Luis Ibanez, Will Schroeder, Lydia Ng, Josh Cates, and the Insight Software Consortium, *The ITK Software Guide*, Second Edition ed., 2005.
- Fengjie Sun, He Wang, and Jieqing Fan, "2D Otsu Segmentation Algorithm Based on Simulated Annealing Genetic Algorithm for Iced-Cable Images," in *Information Technology and Applications, 2009. IFITA '09. International Forum*, Chengdu, 2009, pp. 600-602.
- Simon St Laurent, Joe Johnstin, and Edd Dumbill, *Programming Web Services with XML-RPC*, First Edition ed. Sebastopol, Canada: O'Reilly, 2001.