Pengenalan *Viseme* Dinamis Bahasa Indonesia Menggunakan *Convolutional Neural Network*

Aris Nasuha[1,5], Tri Arief Sardjono[2,4], Mauridhi Hery Purnomo[2,3]

***Abstract*—There has been very little researches on automatic lip reading in Indonesian language, especially the ones based on dynamic visemes. To improve the accuracy of a recognition process, for certain problems, choosing suitable classifiers or combining of some methods may be required. This study aims to classify five dynamic visemes of Indonesian language using a CNN (Convolutional Neural Network) and to compare the results with an MLP (Multi Layer Perceptron). Varying some parameters theoretically improving the recognition accuracy was attempted to obtain the best result. The data includes videos on pronunciation of daily words in Indonesian language by 28 subjects recorded in frontal view. The best recognition result gives 96.44% of validation accuracy using the CNN classifier with three convolution layers.**

***Intisari*— Penelitian tentang pembacaan bibir otomatis dalam bahasa Indonesia masih sangat sedikit dilakukan, apalagi yang meneliti dengan berbasis kepada pengenalan *viseme* dinamis. Di sisi lain, untuk meningkatkan akurasi suatu proses pengenalan, diperlukan suatu pengklasifikasi yang tepat untuk masalah yang dihadapi atau dengan menggunakan gabungan beberapa metode. Makalah ini bertujuan mengenali lima *viseme* dinamis dalam bahasa Indonesia dengan menggunakan pengklasifikasi *Convolutional Neural Network* (CNN) dan sebagai pembanding digunakan *Multi Layer Perceptron* (MLP). Beberapa parameter yang secara teoretis berpengaruh pada akurasi telah divariasi untuk mendapatkan hasil terbaik. Data penelitian adalah video pengucapan kata-kata sehari-hari dalam bahasa Indonesia yang diucapkan oleh 28 subjek dan direkam dari arah depan. Hasil terbaik pengenalan adalah akurasi validasi 96,44% untuk pengklasifikasi CNN dengan tiga lapisan konvolusi.**

***Kata Kunci*—*viseme* dinamis, bahasa Indonesia, *Convolution Neural Network*.**

I. PENDAHULUAN

Pembacaan bibir (*lip reading*) adalah proses mengenali pembicaraan seseorang hanya dengan mengamati sinyal visual [1], dalam hal ini adalah bentuk dan gerak bibir, juga lidah dan gigi. Pembacaan bibir dilakukan oleh manusia maupun mesin. Pembacaan bibir secara otomatis oleh mesin telah digunakan untuk berbagai macam keperluan, seperti untuk meningkatkan pengenalan suara otomatis [2], sebagai kunci pengaman [3], alat bantu pembelajaran wicara bagi tunarungu [4], dan alat bantu orang yang tidak bisa berbicara [5].

Pendekatan dalam proses pembacaan bibir otomatis dapat dibagi menjadi dua, yaitu pendekatan per kata atau per *viseme*. Pendekatan per kata biasa dilakukan pada penggunaan yang hanya dibatasi pada pengenalan kata-kata tunggal, misalnya untuk alat bantu pembelajaran awal bagi tunarungu, untuk semacam kunci pengaman, ataupun untuk mengenali perintah-perintah tertentu yang berupa satu kata saja, seperti mengenali perintah sederhana pada *smartphone* [6]. Pendekatan per *viseme* biasanya digunakan untuk pengenalan frasa atau kalimat.

Penelitian tentang pembacaan bibir otomatis dalam bahasa Indonesia masih sangat sedikit. Sebuah penelitian membahas pengenalan tiga kata sederhana dalam bahasa Indonesia dengan ekstraksi fitur gabungan *frame difference* dan *image projection* [7]. Penelitian tersebut dilanjutkan untuk lima kata sederhana, dengan ekstraksi fitur gabungan *double difference* dan *image projection* [8]. Penelitian lain meneliti pengenalan pola gerak bibir dalam pengucapan fonem vokal [9], dan pengenalan pola gerak bibir dalam pengucapan suku kata, dengan menggunakan *Hidden Markov Model* (HMM) [10].

Penelitian lain yang tidak terkait langsung dengan pembacaan bibir otomatis dalam bahasa Indonesia, tetapi penting untuk mendukung pembacaan bibir otomatis, misalnya adalah tentang pemetaan fonem ke *viseme* dalam bahasa Indonesia [11], [12]. Sedangkan penelitian lain meneliti tentang pemodelan *viseme* dinamis [13]. Pada penelitian-penelitian tersebut, ada yang menemukan bahwa jumlah *viseme* dalam bahasa Indonesia adalah dua belas buah [11] dan ada pula yang menemukan sebanyak empat belas buah [12]. Perbedaan ini bisa terjadi karena perbedaan penentuan jumlah fonem maupun perbedaan pengertian *viseme*, sebagaimana disebutkan dalam beberapa literatur. Penelitian lainnya menemukan bahwa jumlah viseme dinamis dalam bahasa Indonesia adalah 38 buah [13].

[1] *Mahasiswa, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember, Jl. Teknik Mesin Gedung B, C, dan AJ Kampus ITS Sukolilo Surabaya, Jawa Timur 60111 Indonesia (tlp: 031-5947302; fax. 031-5931237; e-mail: aris12@mhs.ee@its.ac.id)*

[2] *Dosen, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember, Jl. Teknik Mesin Gedung B, C, dan AJ Kampus ITS Sukolilo Surabaya, Jawa Timur 60111 Indonesia (tlp: 031-5947302; fax. 031-5931237; e-mail: sardjono@elect-eng.its.ac.id, hery@ee.its.ac.id)*

[3] *Dosen, Departemen Teknik Komputer, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember, Jl. Teknik Mesin Gedung B, C, dan AJ Kampus ITS Sukolilo Surabaya, Jawa Timur 60111 Indonesia (tlp: 031-5922936; e-mail: hery@ee.its.ac.id)*

[4] *Dosen, Departemen Teknik Biomedik, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember, Jl. Teknik Mesin Gedung B, C, dan AJ Kampus ITS Sukolilo Surabaya, Jawa Timur 60111 Indonesia (tlp: 031-5923644; e-mail: sardjono@elect-eng.its.ac.id)*

[5] *Dosen, Jurusan Pendidikan Teknik Elektronika dan Informatika, Fakultas Teknik, Universitas Negeri Yogyakarta, Jl. Colombo No. 1 Karangmalang Yogyakarta, 55281 Indonesia (tlp: 0274-554686; e-mail: arisnasuha@uny.ac.id)*

Di sisi lain, dalam suatu proses pengenalan (*recognition*) diperlukan pengklasifikasi (*classifier*), misalnya jaringan saraf tiruan (*artificial neural network*), *Support Vector Machine* (SVM), *Hidden Markov Model* (HMM), atau yang lain. Untuk meningkatkan akurasi pengklasikasi, biasanya dipilih metode spesifik yang sesuai dengan masalah, misalnya *Convolutional Neural Network* (CNN) untuk pengenalan citra [14] atau dengan menggabungkan beberapa pengklasifikasi seperti *Bootstrap aggregating* [15] dan *Adaptive Boosting* [16]. Biasanya metode yang secara spesifik diketahui cocok untuk suatu lingkup penelitian akan diprioritaskan.

Makalah ini membahas penelitian tentang pengenalan lima *viseme* dinamis dalam bahasa Indonesia dengan pengklasifikasi CNN, dan sebagai pembanding adalah *Multi Layer Perceptron* (MLP). CNN telah dikenal berhasil sebagai pengklasifikasi citra, sedangkan MLP telah dikenal sebagai *universal approximator* [17]. Penelitian ini diharapkan dapat terus dikembangkan untuk mengenali frasa atau kalimat dalam bahasa Indonesia.

Pengaturan berikutnya dari makalah ini adalah sebagai berikut. Bagian II membahas tentang Metodologi dan Data, bagian III tentang Hasil dan Pembahasan, sedangkan Kesimpulan ditulis pada bagian V.

II. METODOLOGI DAN DATA

A. Viseme Dinamis

Viseme, yang merupakan singkatan dari *visual phoneme*, adalah representasi visual dari suatu fonem [18], sedangkan *viseme* dinamis mewakili koartikulasi. Koartikulasi atau artikulasi sekunder adalah gejala interaksi antara satu suara dengan suara lain ketika artikulasi utama menghasilkan suara pertama, organ-organ wicara membuat persiapan untuk menghasilkan suara berikutnya. Misalnya, bunyi /m/ dalam kata ‘makan’ dengan bunyi /m/ dalam kata ‘minum’, diucapkan berbeda, meskipun titik artikulasinya sama, yaitu bilabial. Ada dua jenis koartikulasi, yaitu koartikulasi antisipatif dan koartikulasi pengawet. Yang pertama terjadi ketika fitur wicara diantisipasi selama produksi bunyi ujaran sebelumnya, sedangkan yang kedua terjadi ketika efek suara terlihat selama produksi suara yang mengikuti.

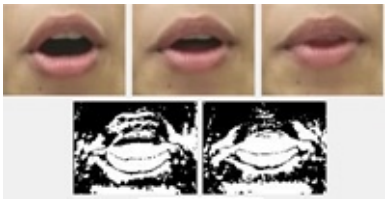
Hasil sebuah penelitian menunjukkan bahwa animasi wicara yang dihasilkan dari *viseme* dinamis lebih alami dan wajar dibandingkan dengan *viseme* statis [19]. Hasil penelitian eksperimental dalam bahasa Cina juga menunjukkan bahwa penggunaan *viseme* dinamis dapat meningkatkan kealamian animasi bibir, dibandingkan dengan penggunaan *triphones* [20]. Dari penelitian-penelitian di atas, diharapkan penggunan *viseme* dinamis dapat meningkatkan akurasi pembacaan bibir otomatis dalam bahasa Indonesia.

B. Frame Difference

Frame difference adalah salah satu metode pada pengolahan citra digital untuk mendeteksi gerak. Prinsip dari metode ini adalah menghitung selisih mutlak intensitas suatu piksel dalam bingkai saat ini dan bingkai sebelumnya, dinyatakan dalam (1).

$$dI(x,y,k)=\begin{cases}1, \left|I(x,y,k)-I(x,y,k-1)\right| \geq T\\ 0, \left|I(x,y,k)-I(x,y,k-1)\right| < T\end{cases} \quad (1)$$

dengan x dan y adalah absis dan ordinat suatu bingkai (*frame*), k = bingkai sekarang, k-1 = bingkai sebelumnya, I = intensitas piksel, dI = selisih intensitas piksel, dan T = *threshold* (nilai ambang). Ilustrasi dari *frame difference* diperlihatkan pada Gbr. 1. Gambar pada baris atas menunjukkan urutan bingkai dalam pengucapan suatu kata, sedangkan baris bawah menunjukkan selisih dua bingkai yang ada di atasnya.



Gbr. 1 Ilustrasi *frame difference.*

Penentuan besarnya nilai ambang dalam metode *frame difference* sangatlah penting. Jika nilai ambang terlalu rendah, akan dihasilkan banyak derau, sebaliknya jika terlalu tinggi, akan terlalu sedikit fitur yang diperoleh. Sementara itu, untuk menemukan nilai yang tepat bagi banyak data sekaligus juga tidak mudah, bahkan mungkin memerlukan penelitian khusus tentang masalah ini.

C. Convolutional Neural Network (CNN)

CNN adalah salah satu jenis jaringan saraf tiruan yang merupakan pengembangan dari MLP. Nama CNN diambil dari adanya operator konvolusi. Tujuan utama dari operator konvolusi di sini adalah untuk mengekstraksi fitur dari data masukan.

CNN pertama kali dikenalkan oleh Yann LeCun, dkk. dengan arsitektur yang diberi nama LeNet-5, yang terdiri atas tujuh lapis, yaitu empat lapis ekstraksi fitur dan tiga lapis terhubung penuh [21]. Jaringan ini berhasil mengenali citra angka tulisan tangan pada basis data *Modified National Institute of Standards and Technology* (MNIST) yang berisi 60.000 data dan sepuluh kelas. Proses pembelajaran LeNet-5 dilakukan dengan 50.000 data dan 10.000 data sebagai data uji. Setelah itu, perkembangan CNN terus berlangsung, apalagi sejak tahun 2010 dimulai kompetisi *ImageNet Large-Scale Visual Recognition Chalenge* (ILSVRC). Kompetisi ini bertujuan untuk mengenali 1.000 kelas dari 1,2 juta data pembelajaran. Beberapa arsitektur CNN yang kemudian populer antara lain Alexnet dengan delapan lapisan [22], VGGNet dengan sembilan belas lapisan [23], GoogLeNet dengan 22 lapisan [24], dan Resnet dengan 152 lapisan [25].

CNN memiliki setidaknya tiga lapisan pokok, yaitu lapisan konvolusi, *pooling*, dan terhubung penuh (*fully connected layer*). Dua lapisan pertama disebut juga lapisan ekstraksi fitur dan lapisan terakhir adalah lapisan klasifikasi. Dalam aplikasinya, masing-masing jenis lapisan jumlahnya bisa lebih dari satu. Urutan lapisan pun bisa bervariasi.

Pada masing-masing lapisan ada beberapa parameter yang perlu diatur dalam membangun suatu CNN. Pada lapisan konvolusi ada parameter ukuran filter, ukuran *kernel*, ukuran

stride, dan *zero padding*. Pada lapisan *pooling* atau juga disebut lapisan *subsampling*, ada parameter ukuran *pool* dan tipe *pooling*. Pada tiap lapisan terhubung penuh ada parameter jumlah neuron. Pada tiap lapisan ada fungsi aktivasi dan dapat ditambahkan fungsi *dropout*. Pada proses pembelajaran, ada parameter yang juga harus diatur, yaitu jumlah *epoch*, ukuran *batch*, dan tipe *optimizer*. Parameter-parameter tersebut harus dipilih dan diatur secara cermat untuk mendapatkan hasil klasifikasi yang terbaik.

Dropout adalah metode yang sederhana tetapi efektif untuk menghindari *overfitting* pada jaringan saraf tiruan maupun model *deep learning*. Metode ini mulai dikenalkan oleh Srivastava dan menjadi populer ketika dipakai pada Alexnet yang menjuarai ILSVRC-2012 [26]. Penetapan nilai *dropout* memang masih bersifat ekperimental. Seorang peneliti menganjurkan agar nilainya di antara 0,2 dan 0,5 [27].

Optimizer adalah algoritme untuk memperbarui bobot dan *bias* pada proses pembelajaran jaringan saraf tiruan, dalam rangka memperkecil *error* atau selisih antara keluaran jaringan dan target. Algoritme dasar yang umum dipakai adalah *gradient descent*. Algoritme ini mempunyai kelemahan yaitu lambat atau kadang sangat lambat untuk mencapai nilai *error* yang cukup kecil. Untuk mengatasi kelemahan tersebut, telah banyak dikembangkan berbagai algoritme, sebagian di antaranya adalah RMSProp, Adadelta [28], AdaGrad [29], Adam [30], dan Nadam [31].

Optimizer Adam adalah gabungan dari *optimizer* RMSProp dan momentum, yang memang memiliki beberapa keunggulan, antara lain efisien dalam komputasi, hemat memori dan cocok untuk berbagai masalah pengoptimalan *non-convex* di bidang pembelajaran mesin. Beberapa peneliti merekomendasikan pemakaian *optimizer* jenis ini [32]-[34]. *Optimizer* Nadam adalah gabungan Adam dan momentum Nesterov. Dalam beberapa kasus, Nadam melampaui kinerja Adam, mengingat secara teoretis dan kadang juga empiris, momentum Nesterov lebih baik dari momentum konvensional.

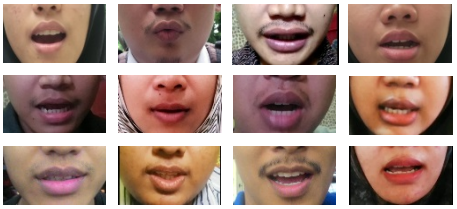
D. Data

Data dalam makalah ini berupa rekaman video tampak depan dari 28 relawan dengan fokus setengah bagian bawah wajah. Contoh citra untuk masing-masing relawan yang diambil datanya ditunjukkan pada Gbr. 2.

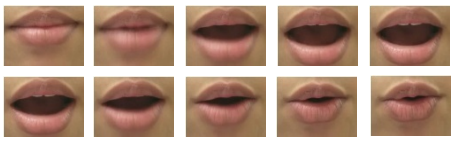
Masing-masing relawan mengucapkan sepuluh kata sehari-hari dalam bahasa Indonesia, yaitu: /saya/, /mau/, /makan/, /minum/, /mandi/, /tolong/, /sapu/, /meja/, /kursi/, dan /sudah/. Gbr. 3 adalah contoh urutan bingkai salah satu relawan yang mengucapkan kata /mau/, setelah mengalami *cropping* sehingga didapatkan citra sekitar bibir saja.

Dari sepuluh kata tersebut lalu dipilih lima *viseme* dinamis saja, yaitu /ma/, /me/, /mi/, /pu/, dan /sa/. Pemilihan dilakukan dengan cara memotong video secara manual. Lima viseme ini dipilih dengan alasan bahwa empat dari lima viseme tersebut termasuk fonem *bilabial* yang mudah dilihat dari luar dan satu termasuk fonem *lamino alveolar* sebagai pembanding.

Dari data yang ada, dipilih data yang cukup baik. Didapat enam puluh data per kelas *viseme*, atau totalnya 300 data. Berdasarkan asumsi bahwa separuh bibir kiri dan kanan simetris, maka data yang dipakai adalah separuh bibir kiri (dilihat dari kamera), sedangkan separuh bibir kanan "dilipat" secara horizontal sehingga terlihat menjadi separuh bibir kiri. Kelebihan dari prinsip ini adalah fitur yang diproses menjadi setengah dari sebelumnya dan jumlah datanya menjadi dua kali lipat, yaitu enam ratus data. Kelebihan lain dari prinsip ini adalah dapat mengurangi pengaruh perbedaan pencahayaan antara bagian kiri dan kanan bibir. Kelebihan ini telah diteliti dan pada pembacaan bibir otomatis dengan menggunakan *smartphone* [35], [36].



Gbr. 2 Contoh bingkai masing-masing relawan.



Gbr. 3 Contoh urutan bingkai salah satu relawan.

Data yang sudah ada tersebut masih tergolong sedikit untuk digunakan dalam pembelajaran CNN. Oleh karena itu, perlu ditambah dengan data buatan yang dikenal sebagai proses *data augmentation*, yang telah terbukti memperbaiki akurasi pada beberapa penelitian [37], [38]. Untuk data penelitian ini, dari tiap kelas dipilih enam data, dengan format augmentasi adalah sebagai berikut:

1. geser horisontal, yaitu secara acak geser kanan atau kiri maksimal 10% dari lebar citra,
2. geser vertikal, yaitu secara acak naik atau turun maksimal 10% tinggi citra,
3. rotasi, yaitu secara acak putar searah atau berlawanan arah jarum jam maksimal 10º, dan
4. *zoom*, yaitu secara acak diperkecil atau diperbesar maksimal 10% dari luas citra.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} z_x & 0 & 0 \\ 0 & z_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4}$$

Persamaan (2) adalah matriks untuk menghitung pergeseran horizontal dan vertikal, dengan t_x adalah besarnya pergeseran horisontal dan t_y adalah besarnya pergeseran vertikal. Variabel x dan y adalah posisi piksel sebelum augmentasi, sedangkan x'

dan *y'* adalah posisi piksel setelah augmentasi. Persamaan (3) adalah matriks untuk menghitung rotasi, dengan θ adalah besarnya sudut rotasi dalam derajat. Persamaan (4) adalah matriks untuk menghitung hasil proses *zoom*, dengan z_x adalah besarnya *zoom* searah sumbu x dan z_y adalah besarnya *zoom* searah sumbu *y*. Contoh hasil augmentasi diperlihatkan pada Gbr. 4.

Gbr. 4 Contoh hasil augmentasi data.

Pada proses augmentasi ini, dari masing-masing data, yaitu enam buah untuk tiap kelas, dibuat menjadi 21 data, sehingga total menjadi 630 data. Dari data tersebut lalu dipilih lagi sehingga tinggal 625 data. Data asli ditambah data hasil proses augmentasi menjadi total 1.225 data. Untuk keperluan pembelajaran dipakai 1.000 data, dan sisanya untuk validasi.

III. DESAIN EKSPERIMEN

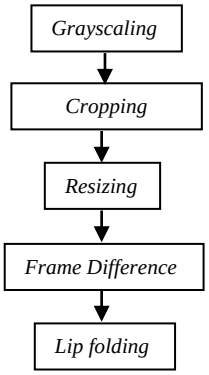
Penelitian dalam makalah ini terbagi menjadi dua tahap, yaitu tahap penyiapan data dan tahap eksperimen. Tahap-tahap penyiapan data ditunjukkan pada Gbr. 5. Penjelasan masing-masing tahap adalah sebagai berikut.

1. *Grayscaling* adalah pengubahan citra warna dalam ruang warna RGB ke citra *grayscale.*
2. *Manual cropping* yaitu *cropping* secara manual untuk mendapatkan citra sekitar bagian bibir saja, sehingga ukuran citra berubah dari 360x240 piksel menjadi 160x120 piksel.
3. *Resizing* yaitu memperkecil ukuran citra menjadi 80x60 piksel atau 50% dari sebelumnya.
4. *Frame difference* yang dimodifikasi yaitu selisih mutlak nilai intensitas piksel pada suatu posisi piksel pada bingkai sekarang dengan nilai intensitas piksel pada posisi yang sama pada bingkai sebelumnya, tanpa menggunakan nilai ambang tertentu. Persamaan untuk menghitung metode ini mengacu pada (5).

$$dI(x,y,k) = \left| I(x,y,k) - I(x,y,k-1) \right| \quad (5)$$

   dengan *x* dan *y* adalah absis dan ordinat suatu bingkai, *k* = bingkai sekarang, *k*-1 = bingkai sebelumnya, *I* = intensitas piksel, dan *dI* = selisih intensitas piksel.
5. *Lip folding* adalah mengambil separuh bibir, dalam hal ini bagian kiri saja (dilihat dari kamera), dan bagian kanan dilipat sehingga seperti bagian kiri. Ukuran citra berubah menjadi 40x60 piksel atau tinggal setengahnya, sekaligus jumlah data menjadi dua kali lipat.

Tahap eksperimen terdiri atas dua bagian, yaitu: pertama, membandingkan hasil pengenalan *viseme* dinamis antara pengklasifikasi MLP dan CNN; dan kedua, variasi parameter lapisan konvolusi pada CNN untuk mencari hasil pengenalan *viseme* yang lebih baik. Perangkat keras yang digunakan adalah komputer pribadi dengan pemroses Intel Core i7 2,7 GHz, RAM 8 GB tanpa *Graphics Processor Unit* (GPU), dengan sistem operasi Windows 7 Ultimate 64 bit. Perangkat lunak yang dipakai adalah Python 3.6.4 dengan berbagai *library*, khususnya TensorFlow 1.8 dan Keras 2.1.4, serta menggunakan *editor* PyCharm 2018.1.3 (*Community Edition*).



Gbr. 5 Tahap penyiapan data.

Eksperimen pertama ini sekilas kurang layak, mengingat bagian akhir dari CNN adalah lapisan terhubung penuh yang isinya adalah MLP. Akan tetapi, pembandingan ini tetap relevan karena data yang dipakai sebagai masukan sudah dikondisikan, sehingga keunggulan CNN, yaitu dapat melakukan ekstraksi fitur untuk data yang bervariasi, tidak lagi dominan. Pengkondisian tersebut antara lain video direkam hanya dari tampak depan dan dilakukan *cropping* manual sehingga citra yang ada hanya daerah sekitar bibir.

Dalam eksperimen pertama, dilakukan variasi berbagai parameter yang secara teoretis berpengaruh pada hasil pengenalan, yaitu ukuran *batch*, jumlah *epoch*, nilai *dropout*, tipe *optimizer*, jumlah lapisan tersembunyi, dan jumlah unit tersembunyi pada lapisan terhubung penuh. Untuk setiap eksperimen, yang divariasi hanya satu parameter dan parameter lainnya diisi dengan suatu nilai asal. Nilai asal untuk masing-masing parameter adalah *epoch* = 10, *batch* = 32, *dropout* = 0,5, tipe *optimizer* = Adam, dan jumlah lapisan tersembunyi adalah 1 dengan jumlah unit tersembunyi = 1.024. Nilai parameter lain tidak divariasi, untuk CNN maupun MLP, yaitu semua fungsi aktivasi adalah *Rectifed Linier Unit* (ReLU), kecuali bagian akhir lapisan terhubung penuh fungsi aktivasinya adalah *Softmax*. Khusus untuk CNN, parameter-nya memiliki nilai asal sebagai berikut: dua lapisan konvolusi, berturut-turut ukuran filternya adalah 32 dan 64, ukuran *stride* = 1x1, ukuran *kernel* = 3x3, ukuran *pool* = 2x2, tipe *pool* = Max, dan ada *zero padding*. Pada masing-masing parameter yang divariasi, ekperimen diulangi sebanyak tiga kali, dengan bobot awal acak. Ringkasan arsitektur jaringan MLP dan CNN ditunjukkan pada Tabel I.

Eksperimen kedua khusus meneliti pengaruh variasi lapisan konvolusi pada CNN, dengan parameter lain tetap. Variasi tersebut adalah variasi jumlah lapisan konvolusi, yaitu dua dan tiga lapisan, serta penyisipan lapisan *Max-pooling* di antara lapisan konvolusi, khusus untuk penggunaan tiga lapisan konvolusi. Ringkasan arsitektur jaringan pada eksperimen ini ditunjukkan pada Tabel II. Pada salah satu hasil eksperimen dengan tiga lapisan konvolusi, diteliti tentang *confusion matrix* dari keluarannya untuk mengamati hasil pengenalan dari masing-masing *viseme* dinamis.

TABEL I
RINGKASAN ARSITEKTUR CNN DAN MLP PADA EKSPERIMEN PERBANDINGAN AKURASI

CNN	MLP
Masukan citra	
Lapisan Konvolusi1: 32	Lapisan tersembunyi1
Lapisan Konvolusi2: 64	
Max pooling	
Dropout	
Lapisan tersembunyi1	
Dropout	*Dropout*
Lapisan tersembunyi2	Lapisan tersembunyi2
Dropout	*Dropout*

TABEL II
RINGKASAN ARSITEKTUR CNN DENGAN DUA LAPISAN KONVOLUSI, TIGA LAPISAN KONVOLUSI DAN TIGA LAPISAN KONVOLUSI DITAMBAH *MAX-POOLING*

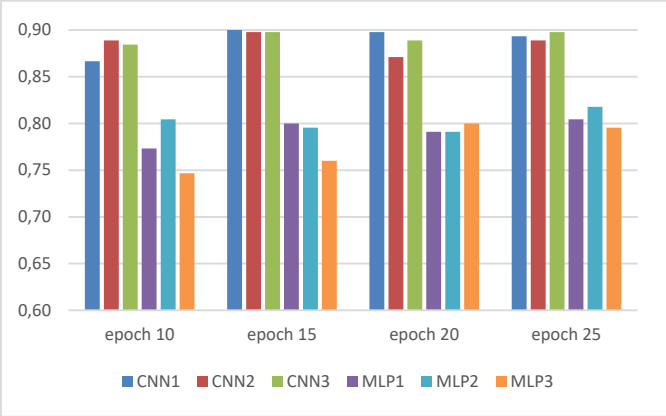
Citra masukan		
Lapisan Konvolusi1	Lapisan Konvolusi1	Lapisan Konvolusi1
		Max pooling
Lapisan Konvolusi2	Lapisan Konvolusi2	Lapisan Konvolusi2
		Max pooling
	Lapisan Konvolusi3	Lapisan Konvolusi3
Max pooling	*Max pooling*	*Max pooling*
Dropout	*Dropout*	*Dropout*
Lapisan tersembunyi	Lapisan tersembunyi	Lapisan tersembunyi
Dropout	*Dropout*	*Dropout*

IV. HASIL DAN PEMBAHASAN

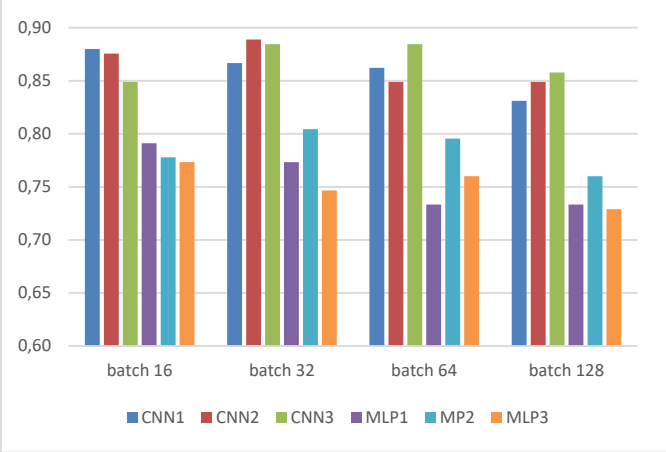
Hasil eksperimen, yaitu perbandingan hasil validasi dengan pengklasifikasi CNN dan MLP, ditunjukkan pada Gbr. 6 sampai Gbr. 11. Hasil eksperimen untuk variasi parameter CNN disajikan pada Tabel III dan Tabel IV.

Gbr. 6 menunjukkan perbandingan akurasi validasi untuk jumlah *epoch* 10, 15, 20, dan 25. Gbr. 7 menunjukkan perbandingan akurasi validasi untuk ukuran *batch* 16, 32, 64, dan 128. Gbr. 8 menunjukkan perbandingan akurasi validasi untuk nilai *dropout* 0,1 sampai 0,7. Perbandingan akurasi validasi untuk jumlah unit tersembunyi 1.024, 512, dan 256, diperlihatkan pada Gbr. 9. Perbandingan akurasi validasi untuk penggunaan dua lapisan tersembunyi ditunjukkan pada Gbr. 10. Sedangkan Gbr. 11 menunjukkan perbandingan akurasi validasi untuk tipe-tipe *optimizer*, yaitu Adadelta, Adam, Nadam, RMSProp, Adamax, dan Adagrad.

Tabel III menunjukkan perbandingan akurasi validasi pada penggunaan CNN untuk dua dan tiga lapisan konvolusi. Baris ke-1 sampai dengan ke-5 adalah dua lapisan konvolusi, baris ke-6 sampai ke-13 adalah tiga lapisan konvolusi tanpa penyisipan lapisan *max-pooling*, dan baris ke-14 sampai dengan ke-21 adalah tiga lapisan konvolusi dengan penyisipan lapisan *max-pooling*. Kolom eks1, eks2, dan eks3 adalah berturut-turut eksperimen pertama, kedua, dan ketiga. Sedangkan rerata adalah nilai rerata dari tiga kali eksperimen. Tabel IV adalah salah satu contoh rincian keluaran salah satu arsitektur jaringan, dalam bentuk *confusion matrix*, untuk mengetahui akurasi masing-masing *viseme*. Dalam hal ini, dipilih keluaran CNN untuk arsitektur tiga lapisan konvolusi dengan penyisipan lapisan *max-pooling*. Nilai pada baris menunjukkan nilai aktual, sedangkan nilai pada kolom adalah nilai prediksi.



Gbr. 6 Perbandingan akurasi validasi untuk jumlah *epoch* bervariasi.

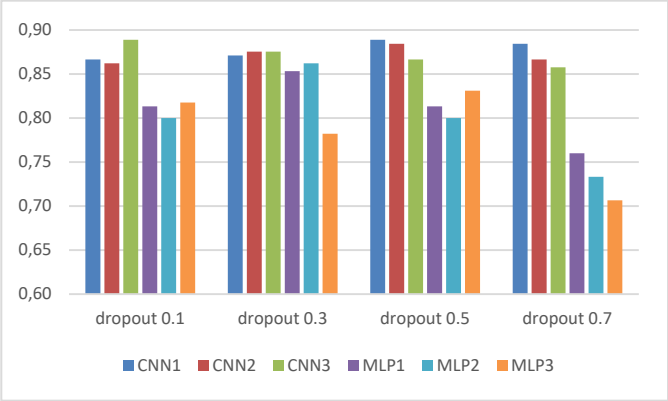


Gbr. 7 Perbandingan akurasi validasi untuk nilai *batch* bervariasi.

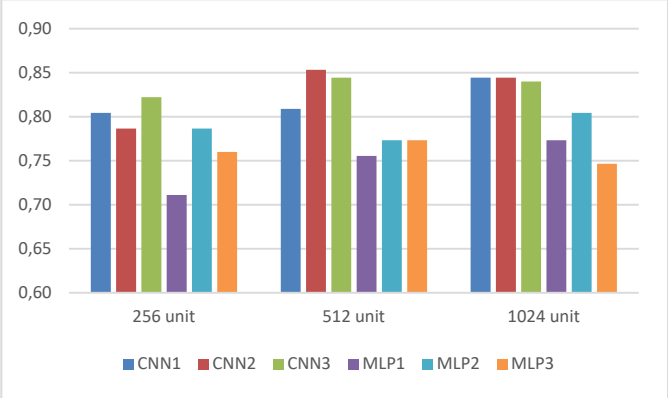
Pada eksperimen dengan variasi pada parameter jumlah *epoch*, terlihat pada Gbr. 6 bahwa dengan CNN, jumlah *epoch* 15 sudah mendapatkan hasil akurasi terbaik, sementara dengan MLP hasil terbaik untuk jumlah *epoch* 25. Meskipun demikian, hasil akurasi terbaik dengan CNN, yaitu 90,22%, masih jauh lebih tinggi dari akurasi terbaik dengan MLP, yaitu 81,78%.

Hasil eksperimen yang membandingkan akurasi CNN dengan MLP pada variasi parameter ukuran *batch*, yaitu pada Gbr. 7, menunjukkan bahwa dengan CNN, nilai akurasi terbaik terjadi pada ukuran *batch* 32, yaitu nilai asal parameter ini, sedangkan dengan MLP pada ukuran *batch* 16. Sama seperti ekperimen sebelumnya, nilai akurasi terbaik CNN, yaitu sebesar 88,89%, jauh lebih baik dari akurasi terbaik dengan MLP, yaitu 80,44%.

Batch yang dimaksud di sini adalah jumlah sampel per pembaruan gradien dalam proses pembelajaran jaringan. Idealnya, seluruh sampel pembelajaran digunakan untuk menghitung perubahan gradien dalam setiap pembaruan. Namun, hal ini tidak efisien, bahkan dapat menghabiskan memori. Dengan menggunakan ukuran *batch* tertentu, atau menggunakan sejumlah sampel saja, akan lebih menghemat memori dan biasanya membuat proses pembelajaran berjalan lebih cepat. Hanya saja, ukuran *batch* yang terlalu kecil akan berakibat pada turunnya akurasi dalam estimasi gradien.



Gbr. 8 Perbandingan akurasi validasi untuk nilai *dropout* bervariasi.

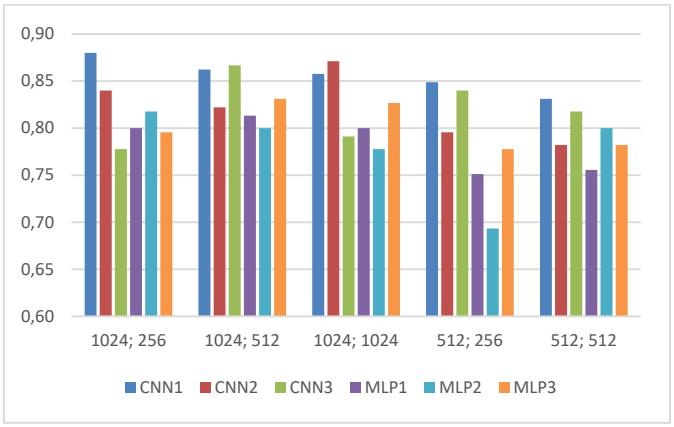


Gbr. 9 Perbandingan akurasi validasi untuk satu lapisan tersembunyi dengan jumlah unit tersembunyi bervariasi.

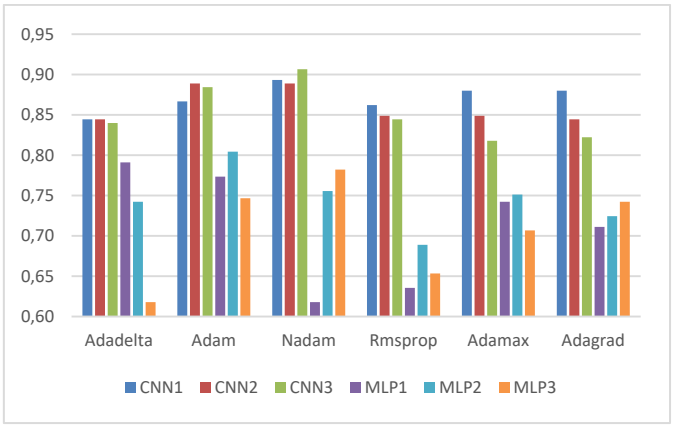
Perbandingan akurasi validasi dengan variasi pada parameter nilai *dropout*, yaitu pada Gbr. 8, menunjukkan bahwa nilai terbaik adalah 88,89%, yaitu pada penggunaan CNN untuk nilai *dropout* 0,5 dan 0,1. Ketika dilihat dari rerata tiga kali eksperimen, terlihat bahwa untuk *dropout* 0,5 rerata akurasinya lebih baik, yaitu 88%, sedikit di atas rerata akurasi untuk *dropout* 0,1, yaitu 87,26%. Sementara untuk penggunaan MLP, nilai akurasi terbaik adalah 86,22% untuk *dropout* 0,3, hanya sedikit lebih tinggi dari akurasi terendah untuk penggunaan CNN sebesar 85,78%.

Pada makalah ini, terbukti bahwa metode *dropout* cukup efektif digunakan pada MLP, dalam hal ini memperbaiki akurasi validasi. Namun, harus diakui bahwa penentuan besarnya *dropout* memang masih bersifat eksperimental. Jika nilai *dropout* terlalu kecil, efeknya tidak terlihat. Sebaliknya, jika terlalu besar akan mengakibatkan jaringan "kurang belajar" (*under-learning network*).

Perbandingan akurasi validasi untuk variasi parameter pada lapisan terhubung penuh, dalam hal ini variasi jumlah unit tersembunyi, ditunjukkan pada Gbr. 9. Hasil ekperimen ini menunjukkan bahwa baik untuk CNN maupun MLP, keduanya mencapai akurasi validasi terbaik pada jumlah unit tersembunyi 1.024, yang merupakan nilai asal parameter ini. Akan tetapi, dari sisi nilai akurasi, hasil CNN, yaitu 84,44%, masih lebih tinggi daripada hasil MLP, yaitu 80,44%. Apalagi jika dilihat dari rerata akurasi, hasil MLP hanya 77,48%, sedangkan hasil CNN adalah 84,30%.



Gbr. 10 Perbandingan akurasi validasi dengan dua lapisan tersembunyi, untuk jumlah unit tersembunyi bervariasi.



Gbr. 11 Perbandingan akurasi validasi untuk tipe *optimizer* bervariasi.

Perbandingan berikutnya masih pada variasi parameter unit tersembunyi, tetapi menggunakan dua lapisan tersembunyi, ditampilkan pada Gbr. 10. Hasil akurasi terbaik dengan CNN adalah 88%, yaitu untuk jumlah unit tersembunyi 1.024 dan 256. Hasil akurasi terbaik dengan MLP adalah 83,11% untuk jumlah unit tersembunyi 1.024 dan 512. Akan tetapi, jika dilihat dari rerata akurasi untuk tiga kali pengulangan eksperimen, hasil akurasi terbaik untuk keduanya, yaitu CNN dan MLP, adalah sama-sama pada jumlah unit tersembunyi 1.024 dan 512. Ketika dibandingkan antara penggunaan satu dan dua lapisan tersembunyi, terlihat baik untuk CNN maupun MLP, hasil akurasi terbaik untuk penggunaan dua lapisan tersembunyi lebih baik dari penggunaan satu lapisan. Hanya saja, penggunaan lapisan tersembunyi yang lebih banyak membutuhkan memori yang lebih banyak dan waktu proses pembelajaran yang lebih lama.

Perbandingan terakhir untuk ekperimen tahap pertama adalah pada variasi tipe *optimizer*. Pada Gbr. 11, terlihat bahwa hasil akurasi terbaik untuk CNN adalah pada penggunaan *optimizer* Nadam, yaitu sebesar 90,67%, sedangkan untuk MLP adalah pada penggunaan Adam, yaitu sebesar 80,44%. Jika dibandingkan dengan penggunaan tipe *optimizer* lain, Adam dan Nadam menempati rerata akurasi terbaik pertama dan kedua, baik untuk penggunaan CNN maupun MLP.

TABEL III
PERBANDINGAN AKURASI VALIDASI PADA CNN UNTUK LAPISAN KONVOLUSI YANG DIVARIASI

No	Ukuran Filter	eks1	eks2	eks3	rerata
1	32; 32	0,8444	0,8444	0,8756	0,8548
2	32; 64	0,8889	0,8844	0,8667	0,8800
3	64; 64	0,8222	0,8533	0,8622	0,8459
4	64; 128	0,8756	0,8844	0,8711	0,8770
5	128; 128	0,8622	0,8044	0,8533	0,8400
6	32; 32; 64	0,8889	0,8578	0,9022	0,8830
7	32; 64; 64	0,8933	0,9111	0,8889	0,8978
8	32; 64; 128	0,8800	0,8889	0,9067	0,8918
9	32; 128; 128	0,8978	0,8756	0,8578	0,8770
10	64; 64; 64	0.8933	0,8800	0,8978	0,8904
11	64; 64; 128	0,8978	0,9111	0,8978	0,9022
12	64; 128; 128	0,8933	0,9022	0,9022	0,8993
13	64; 128; 256	0,8844	0,8800	0,8533	0,8726
14	32; 32; 64	0,9200	0,9378	0,9289	0,9289
15	32; 64; 64	0,9422	0,9067	0,9422	0,9304
16	32; 64; 128	0,9244	0,9644	0,9289	0,9392
17	32; 128; 128	0,9289	0,9333	0,9200	0,9274
18	64; 64; 64	0,9244	0,9333	0,9111	0,9230
19	64; 64; 128	0,9200	0,9244	0,9244	0,9229
20	64; 128; 128	0,9422	0,9467	0,9200	0,9363
21	64; 128; 256	0,9556	0,9244	0,9333	0,9378

TABEL IV
CONFUSION MATRIX UNTUK PENGKLASIFIKASI CNN DENGAN TIGA LAPISAN KONVOLUSI

		Prediksi				
		ma	**mi**	**sa**	**me**	**pu**
Aktual	**ma**	**45**	0	0	0	0
	mi	0	**44**	0	1	0
	sa	2	0	**41**	2	0
	me	1	0	0	**43**	1
	pu	0	3	1	6	**35**

Dari hasil-hasil eksperimen perbandingan penggunaan CNN dan MLP, secara umum terlihat bahwa penggunaan CNN selalu mendapatkan akurasi yang lebih baik dibanding pada penggunaan MLP. Meskipun telah dilakukan pengkondisian terhadap data masukan, hasil CNN tetap lebih baik. Hal ini dikarenakan CNN secara internal melakukan ekstraksi fitur sebelum proses pengenalan dengan lapisan terhubung penuh yang berupa suatu MLP.

Khusus untuk eksperimen dengan CNN, pada Tabel III baris 1 sampai 13, terlihat bahwa semakin banyak lapisan konvolusi cenderung semakin akurat. Hal ini terlihat dari rerata ketiga eksperiman, bahwa rerata akurasi dari tiga lapisan selalu lebih tinggi dari dua lapisan. Sementara itu, hasil akurasi terbaik yang diperoleh adalah 91,11%, yaitu untuk penggunaan tiga lapisan konvolusi dengan ukuran filter berturut-turut 64, 64, dan 128. Pada penggunaan dua lapisan konvolusi, akurasi terbaik adalah 88,89% untuk ukuran lapisan konvolusi adalah 32 dan 64, yang merupakan nilai asal ukuran filter lapisan konvolusi.

Penyisipan lapisan *max-pooling* pada lapisan konvolusi dapat menaikkan akurasi validasi secara konsisten. Terlihat pada Tabel III baris 14 sampai 21, semua rerata akurasi validasi pada jaringan dengan penyisipan *max-pooling* selalu lebih tinggi dibandingkan tanpa penyisipan. Akurasi tertinggi pada arsitektur ini adalah 96,44%, sementara rerata akurasi tertinggi adalah 93,93%. Keduanya pada ukuran filter 32, 64, dan 128.

Pada Tabel IV, terlihat bahwa yang paling mudah dikenali berturut-turut adalah *viseme* /ma/, yaitu 100% dikenali, /mi/ (97,78%), /me/ (95,56%), dan /sa/ (91,11%). Yang paling sulit dikenali adalah /pu/ (77,78%), sebanyak enam data (13,33%) terbaca sebagai /me/, 6,67% terbaca sebagai /mi/, dan 2,22% terbaca sebagai /sa/.

V. KESIMPULAN

Pada eksperimen perbandingan akurasi validasi pengenalan lima *viseme* dinamis bahasa Indonesia untuk penggunaan pengklasifikasi MLP dan CNN, dengan berbagai variasi parameter, akurasi validasi CNN selalu lebih tinggi dibanding MLP, baik pada eksperimen individual maupun rerata tiga pengulangan eksperimen. Sementara itu, untuk ekpserimen pada variasi lapisan konvolusi CNN, rerata akurasi validasi dari penggunaan tiga lapisan konvolusi mengungguli dua lapisan konvolusi. Demikian pula penyisipan lapisan *max-pooling* pada lapisan konvolusi secara umum dapat meningkatkan akurasi validasi. Hasil terbaik untuk keseluruhan eksperimen adalah akurasi validasi sebesar 96,44%, sedangkan untuk masing-masing *viseme* dinamis, yang paling mudah dikenali adalah /ma/ dan yang paling sulit dikenali adalah /pu/.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Yosi Kristian, Willy Achmat Fauzi, dan Muhammad Afif Hendrawan, atas diskusi-diskusinya seputar *Convolutional Neural Network* dan Python, yang sangat membantu bagi penelitian ini.

REFERENSI

[1] S.H. Leung, A.W.C. Liew, W.H. Lau, dan S.L. Wang, "Automatic Lipreading with Limited Training Data," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006, Vol. 3, hal. 881–884.

[2] V. Estellers dan J.-P. Thiran, "Multi-pose Lipreading and Audio-Visual Speech Recognition," *EURASIP J. Adv. Signal Process.*, Vol. 51, hal. 1-23, 2012.

[3] A. B. Hassanat, "Visual Passwords Using Automatic Lip Reading," *Int. J. Sci. Basic Appl. Res.*, Vol. 13, No. 1, pp. 218–231, Sep. 2014.

[4] S. Chen, D. M. Quintian, dan Y. L. Tian, "Towards a Visual Speech Learning System for the Deaf by Matching Dynamic Lip Shapes," *ICCHP'12*, 2012, hal. 1–9.

[5] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, dan J. S. Brumberg, "Silent Speech Interfaces," *Speech Commun.*, Vol. 52, No. 4, hal. 270–287, Apr. 2010.

[6] Y.-U. Kim, S.-K. Kang, dan S.-T. Jung, "Design and Implementation of a Lip Reading System in Smart Phone Environment," *2009 IEEE International Conference on Information Reuse Integration*, 2009, hal. 101–104.

[7] F. Arifin, A. Nasuha, dan H.D. Hermawan, "Lip Reading Based on Background Subtraction and Image Projection," *2015 International*

Conference on Information Technology Systems and Innovation (ICITSI), 2015, hal. 1–3.

[8] A. Nasuha, F. Arifin, T.A. Sardjono, H. Takahashi, dan M.H. Purnomo, "Automatic Lip Reading for Daily Indonesian Words Based on Frame Difference and Horizontal-Vertical Image Projection," *J. Theor. Appl. Inf. Technol.*, Vol. 95, No. 2, hal. 393-402, Jan. 2017.

[9] F. Faridah dan N. Effendy, "Pengenalan Pola Gerak Bibir dalam Pengucapan Fonem Vokal Bahasa Indonesia," *Teknofisika*, Vol. 1, No. 2, hal. 96–100, Sep. 2012.

[10] B. Achmad, F. Faridah, dan L. Fadillah, "Lip Motion Pattern Recognition for Indonesian Syllable Pronunciation Utilizing Hidden Markov Model Method," *TELKOMNIKA Telecommun. Comput. Electron. Control*, Vol. 13, No. 1, hal. 173–180, 2015.

[11] E. Setyati, S. Sumpeno, M.H. Purnomo, K. Mikami, M. Kakimoto, dan K. Kondo, "Phoneme-Viseme Mapping for Indonesian Language Based on Blend Shape Animation," *IAENG Int. J. Comput. Sci.*, Vol. 42, No. 3, 2015.

[12] M. Muljono, S. Sumpeno, A. Arifin, D. Arifianto, dan M.H. Purnomo, "Indonesian Text to Audio Visual Speech with Animated Talking Head," *Int. Rev. Comput. Softw. IRECOS*, Vol. 11, No. 3, hal. 261-269, Mar. 2016.

[13] Arifin, S. Sumpeno, M. Muljono, dan M. Hariadi, "A Model of Indonesian Dynamic Visemes from Facial Motion Capture Database Using a Clustering-Based Approach," *IAENG Int. J. Comput. Sci.*, Vol. 44, No. 1, hal. 41–51, Jan. 2017.

[14] W. Rawat dan Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Comput.*, Vol. 29, No. 9, hal. 2352–2449, Sep. 2017.

[15] L. Breiman, "Bagging Predictors," *Mach. Learn.*, Vol. 24, No. 2, hal. 123–140, Agt. 1996.

[16] Y. Yamasari, S.M.S. Nugroho, D.F. Suyatno, and M.H. Purnomo, "Meta-Algoritme Adaptive Boosting untuk Meningkatkan Kinerja Metode Klasifikasi pada Prestasi Belajar Mahasiswa," *J. Nas. Tek. Elektro Dan Teknol. Inf. JNTETI*, Vol. 6, No. 3, hal. 333-341, 2017.

[17] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Math. Control Signals Syst.*, Vol. 2, No. 4, hal. 303–314, Des. 1989.

[18] C.G. Fisher, "Confusions Among Visually Perceived Consonants," *J. Speech Lang. Hear. Res.*, Vol. 11, No. 4, hal. 796–804, Des. 1968.

[19] S.L. Taylor, M. Mahler, B.-J. Theobald, dan I. Matthews, "Dynamic Units of Visual Speech," *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2012, hal. 275–284.

[20] H. Li dan C. J. Tang, "Dynamic Chinese Viseme Model Based on Phones and Control Function," *Electron. Lett.*, Vol. 47, No. 2, hal. 144–145, Jan. 2011.

[21] Y. LeCun, L. Bottou, Y. Bengio, dan P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, hal. 2278-2324, 1998.

[22] A. Krizhevsky, I. Sutskever, dan G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, Vol. 60, No. 6, hal. 84–90, Mei 2017.

[23] K. Simonyan dan A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ArXiv14091556 Cs*, Sep. 2014.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke. A. Rabinovich, "Going Deeper with Convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, hal. 1–9.

[25] K. He, X. Zhang, S. Ren, dan J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, hal. 770–778.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, dan R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, Vol. 15, hal. 1929–1958, 2014.

[27] J. Brownlee (2016) Dropout Regularization in Deep Learning Models With Keras. [Online], https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/, tanggal akses: 20-Jun-2018.

[28] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *ArXiv12125701 Cs*, Dec. 2012.

[29] J. Duchi, E. Hazan, dan Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *J. Mach. Learn. Res.*, Vol. 12, hal. 2121–2159, 2011.

[30] D. P. Kingma dan J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv14126980 Cs*, Dec. 2014.

[31] T. Dozat, "Incorporating Nesterov Momentum into Adam," Stanford University, Technical Report, 054, 2015.

[32] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, dan Y. Bengio., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," *Proceedings of the 32nd International Conference on Machine Learning (PMLR 37)*, 2015, hal. 2048–2057.

[33] K. Gregor, I. Danihelka, A. Graves, D. Rezende, dan D. Wierstra, "DRAW: A Recurrent Neural Network For Image Generation," *Proceedings of the 32nd International Conference on Machine Learning (PMLR 37)*, 2015, hal. 1462–1471.

[34] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *ArXiv160904747 Cs*, Sep. 2016.

[35] M. G. Song, M. Tariquzzaman, J. Y. Kim, S. T. Hwang, dan S. H. Choi, "A Robust and Real-Time Visual Speech Recognition for Smartphone Application," *Int. J. Innov. Comput. Inf. Control*, Vol. 8, No. 4, hal. 2837-2853, Apr. 2012.

[36] Young-Un Kim, Sun-Kyung Kang, dan Sung-Tae Jung, "Design and Implementation of a Lip Reading System in Smart Phone Environment," *2009 IEEE International Conference on Information Reuse & Integration*, 2009, hal. 101–104.

[37] S. C. Wong, A. Gatt, V. Stamatescu, dan M. D. McDonnell, "Understanding Data Augmentation for Classification: When to Warp?," *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016, hal. 1–6.

[38] I. Rebai, Y. BenAyed, W. Mahdi, and J.-P. Lorré, "Improving speech recognition using data augmentation and acoustic model fusion," *Procedia Comput. Sci.*, vol. 112, pp. 316–322, 2017.