Penerapan *Convolutional Neural Network* untuk Melakukan Estimasi *Pitch* pada Rekaman Suara Penyanyi

Dionisius Pratama[#1], Hery Heryanto[#2], Hans Christian Kurniawan[#3]

[#]*Program Studi Informatika, Institut Teknologi Harapan Bangsa*
Jalan Dipatiukur No. 80-84, Bandung, Indonesia 40132
[1]dionisiuspr@gmail.com
[2]hery_heryanto@ithb.ac.id
[3]hans_christian@ithb.ac.id

Abstract*— A musical performance is determined by the intonation accuracy, which is the pitch accuracy of a musician or musical instrument, whether a tone is played 'in tune' or not. Therefore, to determine the intonation quality of a musical performance, it is necessary to estimate the pitch. In this research, a one-dimensional Convolutional Neural Network (CNN) is used to estimate the pitch from singing voice recording. After pitch estimation, Dynamic Time Warping (DTW) method is used to calculate the similarity (measured in distance) of pitch estimation results with the recording template from the dataset to determine intonation accuracy. This research uses several preprocessing methods, such as quantization pitch label, spectrogram generation, scaling, and spectrogram recoloring. The CNN method for performing pitch estimation is tested using five songs from the MIR-QBSH dataset. CNN testing is done by applying four architectural designs by combining epoch values, learning rate, number of filters in each convolutional layer, and number of convolutions to find the best combination that produces the highest accuracy. Based on the test results, the model built can produce the highest average accuracy of 97.425% with a difference between the average accuracy and the average validation accuracy of 14.383%. The optimal threshold value for distance is in the range of 1000-1500.

***Keywords*— Singing voice, pitch estimation, fundamental frequency, similarity matching, Convolutional Neural Network (CNN), Dynamic Time Warping (DTW)**

Abstrak*— Pembawaan karya musik yang baik ditentukan dari ketepatan intonasi yang merupakan akurasi *pitch* dari sebuah nada yang dikeluarkan oleh seorang musisi atau instrumen musik, diproduksi dengan tepat atau tidak. Maka dari itu, untuk menentukan kualitas intonasi penampilan suatu karya musik, estimasi *pitch* perlu dilakukan. Pada penelitian ini, sebuah *Convolutional Neural Network* (CNN) satu dimensi digunakan untuk melakukan estimasi *pitch* dari rekaman suara nyanyian. Setelah estimasi *pitch* dilakukan, maka digunakan metode *Dynamic Time Warping* (DTW) untuk melakukan pengujian kemiripan (dalam *distance*) hasil estimasi *pitch* dengan *template* rekaman dari *dataset*. Pengujian tersebut dilakukan untuk menentukan ketepatan intonasi. Beberapa metode *preprocessing* yang dilakukan adalah pembulatan *pitch* label, pembuatan spektogram, *scaling*, dan pewarnaan ulang spektogram. Metode CNN untuk melakukan estimasi *pitch* diuji dengan menggunakan lima lagu dari *dataset* MIR-QBSH. Pengujian CNN dilakukan dengan menerapkan empat rancangan arsitektur dengan mengombinasikan nilai *epoch*, *learning rate*, jumlah filter pada setiap *convolutional layer*, dan jumlah konvolusi untuk mencari kombinasi terbaik yang menghasilkan akurasi tertinggi. Berdasarkan hasil pengujian, model yang dibangun dapat menghasilkan rata-rata akurasi tertinggi sebesar 97,425% dengan selisih antara rata-rata akurasi dan rata-rata akurasi validasi sebesar 14,383%. Nilai *threshold* yang optimal untuk *distance* berada pada rentang 1000-1500.

***Kata Kunci*— Suara nyanyian, estimasi *pitch*, pengujian kemiripan (*similarity matching*), *Convolutional Neural Network* (CNN), *Dynamic Time Warping* (DTW)**

I. PENDAHULUAN

Menyanyi adalah salah satu media hiburan dan yang populer. Menyanyi juga merupakan salah satu keterampilan yang sering dikembangkan oleh seseorang. Namun, pembelajaran menyanyi saat ini masih sangat bergantung pada ahli musik manusia yang jumlahnya sedikit dan tidak tersedia dengan mudah untuk orang awam yang ingin belajar menyanyi. Selain itu, evaluasi nyanyian seseorang masih bergantung pada penilaian ahli yang subjektif. Maka dari itu, dibutuhkan sebuah sistem evaluasi nyanyian yang otomatis dan andal yang dapat berfungsi sebagai bantuan untuk pembelajaran menyanyi, kompetisi menyanyi, pengecekan rekaman nyanyian dalam acara *virtual* (misalnya: *virtual choir*), dan sistem karaoke, yang dapat membuat pelatihan menyanyi dan evaluasinya lebih mudah diakses oleh banyak orang [1].

Revolusi digital dalam distribusi dan penyimpanan data audio telah memicu minat dan perhatian yang besar tentang bagaimana cara teknologi informasi dapat dimanfaatkan untuk mengelola data audio [2]. Salah satu pemanfaatan teknologi tersebut adalah pemrosesan suara nyanyian [3]. Suara nyanyian telah menjadi instrumen yang menantang untuk dianalisis karena karakter suara yang ekspresif, *timbre* yang bervariasi, dan sumber daya (manusia) yang secara ekspresif mencirikan gaya yang khas. Berbeda dengan teknik pengolahan *speech*, variasi pada suara nyanyian lebih luas, cepat mengalami perubahan, mempunyai dinamika yang beragam, dan memiliki suara yang lebih panjang [4].

Dalam menganalisis nyanyian, salah satu atribut penting yang perlu diperhatikan adalah *pitch* yang merupakan interpretasi dari frekuensi dasar (*fundamental frequency*), f_0, dalam aliran audio akustik. Dari sudut pandang bidang keilmuan *Musical Information Retrieval*/MIR, *pitch* merupakan atribut utama musik yang menentukan ketepatan pembawaan karya musik. Ketepatan tersebut diberi istilah intonasi yang merupakan akurasi *pitch* dari sebuah nada yang dikeluarkan oleh seorang musisi atau instrumen musik, apakah diproduksi dengan tepat atau tidak. Untuk menentukan ketepatan intonasi, metode yang paling umum digunakan adalah estimasi *pitch*. Ketepatan intonasi, atau yang disebut juga dengan *singing 'in tune'*, secara langsung terkait dengan ketepatan *pitch* yang dihasilkan seorang musisi [1]. Maka dari itu, ketepatan *pitch* penting karena menentukan ketepatan pembawaan karya musik dari seorang musisi, dalam penelitian ini: penyanyi. Hasil dari estimasi *pitch* dapat menentukan apakah seorang musisi (penyanyi) telah bernyanyi dengan intonasi yang tepat atau tidak.

Dengan didasari oleh perkembangan teknik *deep learning* yang terbukti dapat meningkatkan kualitas serta akurasi teknik pengolahan suara nyanyian secara substansial dan metode *Convolutional Neural Network* (CNN) yang menghasilkan akurasi tinggi [3], [5]–[7], penelitian ini menggunakan metode CNN satu dimensi untuk melakukan estimasi *pitch*. Pergerakan satu dimensi ini digunakan untuk menganalisis spektogram yang merupakan data *time-series*. Objek yang dipilih adalah suara nyanyian, di mana akan dilakukan eksperimen untuk mengembangkan penelitian [4] yang belum menggunakan teknik *deep learning*. Hasil estimasi *pitch* pada penelitian ini adalah sebuah sekuens yang menunjukkan *pitch* yang terdeteksi dalam rekaman masukan. Penelitian ini juga melakukan eksperimen pengujian kemiripan untuk mengukur ketepatan intonasi sebuah lagu yang spesifik yang belum dilakukan pada penelitian sebelumnya [1], [3]–[7].

Pengujian kemiripan dillakukan sebagai berikut. Setelah hasil estimasi *pitch* didapatkan, pengujian kemiripan (*similarity matching*) dilakukan dengan cara membandingkan sekuens *pitch* yang terdeteksi pada rekaman dengan sekuens *pitch template* sebagai acuan pengujian kemiripan menggunakan metode *Dynamic Time Warping* (DTW). Jarak (*distance*) antara *pitch template* dan rekaman masukan menjadi indikator akurasi intonasi, di mana semakin kecil nilai *distance*, akurasi intonasi dianggap semakin baik [2].

Dengan adanya sistem estimasi *pitch* dan pengujian kemiripan yang diusulkan dalam penelitian ini, diharapkan sistem yang dibangun mampu dikembangkan untuk aplikasi dalam hidup sehari-hari. Apliaksi tersebut antara lain: aplikasi belajar bernyanyi secara mandiri, penilaian otomatis dari sebuah kompetisi bernyanyi melalui rekaman, pengecekan otomatis rekaman nyanyian yang banyak memiliki ketidaktepatan intonasi, dan sebagainya.

II. METODOLOGI

A. Notasi Musik dan Pitch

Setiap notasi (not) yang ditulis pada partitur musik menggambarkan durasi dan *pitch* nada yang akan dibawakan atau dimainkan oleh seorang musisi. Not yang tertulis pada partitur memberi tahu nada mana yang harus dimainkan dan berapa lama nada harus ditahan. Not pada suatu instrumen (termasuk suara manusia) dimainkan untuk menghasilkan bunyi periodik dengan frekuensi dasar (*fundamental frequency*). Hal tersebut berkaitan erat dengan penentuan *pitch* [2].

Dua nada dengan frekuensi dasar dalam rasio yang sama dengan pangkat dua akan terdengar sangat mirip. Oleh karena itu, semua nada dengan relasi seperti ini dapat dikelompokkan dalam *pitch class* yang sama. Hal ini mengarah pada pengertian dasar dari sebuah oktaf yang merupakan jarak (*interval*) antara satu *pitch* dengan *pitch* lainnya dengan setengah atau dua kali lipat frekuensi dasarnya [2].

Untuk mendeskripsikan musik dalam simbol yang terbatas, perlu ada suatu cara untuk memisahkan semua kemungkinan *pitch*. Hal ini mengarah kepada penggunaan tangga nada (*scale*) yang dapat dianggap sebagai satu himpunan perwakilan *pitch* yang terbatas. Karena hubungan yang dekat antara himpunan *pitch* tersebut, tangga nada umumnya dianggap menjangkau satu oktaf, dengan oktaf yang lebih tinggi atau lebih rendah hanya mengulangi pola [2]. Dalam *twelve-tone equal-tempered scale*, ada dua belas kelas *pitch* yang dilambangkan dengan menggabungkan huruf A, B, C, D, E, F, G, dan tanda aksidental berdasarkan notasi musik barat. Sebagai contoh, tujuh kelas *pitch* pada tangga nada C mayor dilambangkan dengan huruf C, D, E, F, G, A, B. Tujuh kelas *pitch* ini sesuai dengan tuts putih pada piano, sedangkan lima kelas *pitch* yang tersisa sesuai dengan tuts hitam pada piano. Lima kelas sisa tersebut dilambangkan dengan kombinasi huruf dan tanda aksidental yang ditulis setelah nama *pitch*. Tanda aksidental akan menaikkan atau menurunkan *pitch* sejumlah *semitone* (jarak setengah nada), tergantung tanda yang digunakan [2].

B. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah pengembangan *multilayer perceptron* yang berupa *neural network* khusus untuk memroses data dengan proses konvolusi. Data yang dapat diproses adalah data *time-domain* yang dapat dianggap sebagai *grid* satu dimensi dengan mengambil sampel pada *interval* waktu dan data gambar yang dapat dianggap sebagai *grid* piksel dua dimensi [8]. Proses pelatihan CNN terdiri atas *forward propagation* untuk mencari nilai probabilitas dari data yang diproses dari *layer* pertama hingga *layer* terakhir dan menentukan ke kelas mana suatu data akan dimasukkan, sedangkan *backward propagation* untuk melakukan optimasi arsitektur CNN dengan melakukan pembaharuan bobot (*weight*) *kernel*.

Pada penelitian ini, karena data berbentuk *time-series* yang membentuk sebuah sekuens, maka digunakan operasi konvolusi yang bersifat satu dimensi. Pergerakan *kernel* ke arah samping kanan (ke arah waktu). Gambar 1 menunjukkan ilustrasi operasi konvolusi satu dimensi, di mana dilakukan operasi konvolusi antara *kernel* dan masukan yang menghasilkan suatu fitur [9].

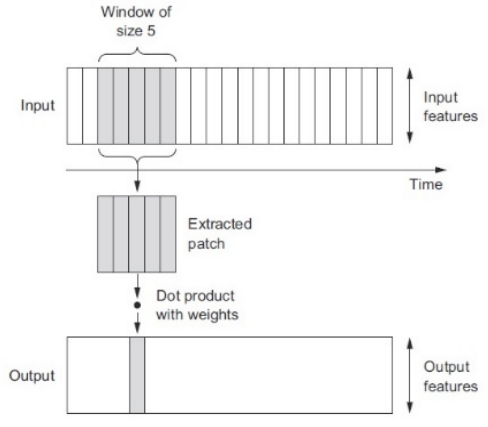
C. Masalah pada Perancangan Model

Faktor yang menentukan baik atau buruk performa algoritma pembelajaran mesin adalah kemampuannya menghasilkan *training error* yang rendah serta selisih antara *training error* dan *test error* yang rendah. Kedua faktor ini berhubungan dengan dua masalah yang umum dijumpai saat merancang model algoritma pembelajaran mesin, yaitu *underfitting* yang terjadi ketika model tidak mampu menghasilkan nilai *training error* yang rendah dan *overfitting* yang terjadi ketika selisih antara *training error* dan *test error* terlalu besar [8]. Untuk mengatasinya, dapat menggunakan teknik-teknik regularisasi. Teknik regularisasi yang digunakan pada penelitian ini adalah: L2 untuk melakukan penalti kepada bobot dan bias yang bernilai besar dari model yang dibangun [10], *batch normalization* untuk menormalisasi hasil konvolusi pada setiap data dalam suatu *batch* [11], dan *dropout* untuk menonaktifkan beberapa fitur secara acak dengan parameter nilai probabilitas *p* agar arsitektur tidak terlalu banyak beradaptasi dengan data pelatihan, sehingga memungkinkan terjadinya *generalization* yang lebih baik [12].

D. Dynamic Tine Warping (DTW)

Dynamic Time Warping (DTW) adalah teknik menyelaraskan dua sekuens yang dependen pada waktu dengan suatu batasan tertentu. Secara intuitif, kedua sekuens dibengkokkan (*warped*) secara nonlinear untuk dapat dicocokkan satu sama lain. Sekuens ini dapat berupa sinyal diskrit (*time series*), atau sekuens fitur, yang diambil sampelnya pada titik-titik yang berjarak sama dalam suatu satuan waktu. DTW akan mengompensasi perbedaan sekuens dengan menemukan kemungkinan keselarasan nonlinear antara elemen-elemen kedua sekuens [2].

Untuk menentukan *warping path* yang optimal dari dua sekuens X dan Y, dihitung total *cost* dari semua *warping path* yang mungkin dan diambil *cost* yang paling minimal. Hasil komputasi *warping path* yang optimal dengan teknik *dynamic programming* adalah *accumulated cost matrix D*. Setiap nilai D(*n,m*) menentukan total *cost* (atau akumulasi) dari *optimal warping path*. Sel terakhir (*n,m*) menunjukkan *distance* (jarak) atau *similarity score* dari dua buah sekuens X dan Y [2].



Gambar 1 Ilustrasi cara kerja operasi konvolusi satu dimensi dengan pergerakan kernel ke arah waktu [9]

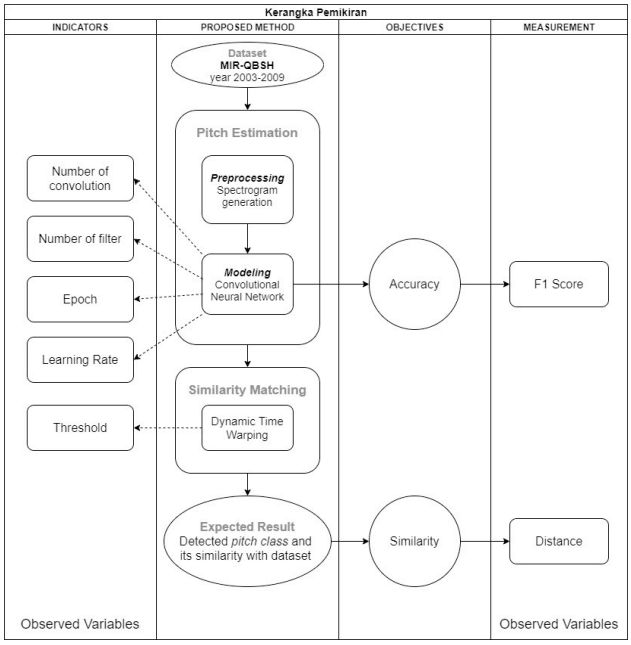
E. Perancangan Sistem

1) Kerangka Pemikiran

Gambar 2 adalah kerangka pemikiran dari metode yang diusulkan, yaitu untuk melakukan estimasi *pitch* dan menguji kemiripan (*similarity matching*). Dalam penelitian ini akan dicari akurasi dari estimasi sekuens *pitch* oleh CNN serta nilai kemiripan sekuens *template* dan sekuens hasil estimasi dari hasil komputasi DTW. Satuan ukur untuk mengukur hasil penelitian adalah metrik dari *confusion matrix* (*accuracy* dan F1 *Score*) untuk pengukuran akurasi serta nilai *distance* antara sekuens pitch dari *dataset* dan hasil estimasi. Beberapa variabel indikator yang perlu diuji adalah sebagai berikut. Variabel-variabel berikut ini diuji agar arsitektur CNN tidak terlalu banyak belajar yang dapat menyebabkan *overfitting*.

1. *Number of convolution* merupakan jumlah operasi konvolusi yang ditentukan dari banyaknya jumlah *convolutional layer*.
2. *Number of filter* menentukan banyak fitur yang akan dihasilkan dalam satu kali konvolusi pada *convolutional layer* serta menentukan banyak *kernel* untuk memroses fitur.
3. *Epoch* menentukan berapa kali algoritme akan memroses seluruh data pelatihan.

Variabel indikator terakhir dalam pengujian arsitektur CNN adalah *learning rate*. *Learning rate* mengatur kecepatan CNN untuk mengenali karakteristik suatu objek. *Learning rate* diobservasi untuk mengontrol kecepatan arsitektur CNN mencapai konvergensi pada solusi yang optimal [8]. Selain keempat variabel tersebut di atas, variabel indikator juga digunakan untuk pengujian kemiripan dengan metode DTW. Variabel tersebut adalah *threshold*. *Threshold* merupakan nilai ambang batas untuk sebuah data (dalam penelitian ini: reka-



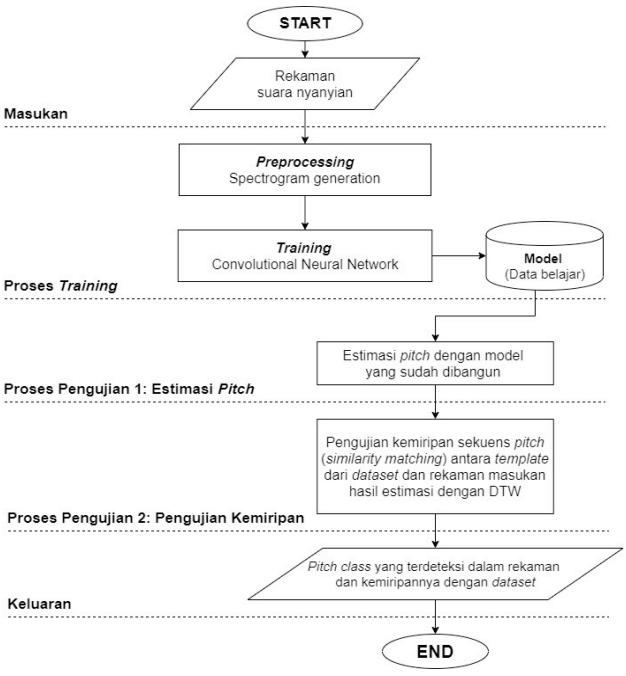
Gambar 2 Kerangka pemikiran

man lagu) dapat diterima tingkat kemiripannya atau tidak. Skor kemiripan (*distance*) yang melebihi *threshold* menyatakan rekaman tidak mirip dengan *template* dari *dataset*. Nilai *threshold* akan dicari dengan melakukan eksperimen setelah menemukan kombinasi antara *epoch*, *learning rate*, jumlah konvolusi, dan jumlah *filter* yang menghasilkan akurasi tertinggi dalam melakukan estimasi *pitch*.

2) Flowchart Global

Flowchart proses dalam penelitian ini ditunjukkan pada Gambar 3 dengan penjelasan sebagai berikut.

1. Pada awal proses, sebagai masukan, digunakan rekaman suara nyanyian yang diambil dari *dataset* MIR-QBSH.
2. Rekaman masukan tersebut kemudian melewati tahap *preprocessing*, di mana dilakukan pembuatan spektogram dari rekaman tersebut. Spektogram pada awalnya dibuat dengan *color space* RGB yang kemudian diubah menjadi *grayscale*. Karena *data label* dibutuhkan CNN untuk dapat melakukan pemrosesan data, maka pada tahap ini juga dilakukan *preprocessing* pada *data label. Data label* yang terdiri dari *pitch values* desimal dibulatkan ke bawah menjadi bilangan bulat tanpa angka desimal.
3. Spektogram beserta dengan *data label* kemudian dimasukkan ke dalam CNN. Hal ini dilakukan untuk melatih model CNN untuk mengenali *pitch* dalam spektogram. Hasil pelatihan tersebut disimpan dalam sebuah data (pada penelitian ini, data disimpan dalam model dengan ekstensi berkas .tf) yang akan digunakan untuk melakukan pengujian.
4. Proses pengujian dibagi ke dalam dua tahap besar, yaitu mencari sekuens *pitch* dari rekaman masukan menggunakan metode CNN dari model yang sudah dibangun dan menguji kemiripan antara rekaman.



Gambar 3 *Flowchart global* usulan sistem estimasi *pitch*

5. Masukan dan *pitch template* dari *dataset* menggunakan metode DTW. Sekuens *pitch* dikenali dengan mengamati pola dari spektogram per satuan waktu, sedangkan pengujian kemiripan ditentukan dari *distance* antara hasil pengujian dan *pitch template*.
6. Keluaran yang dihasilkan adalah perbandingan *pitch class* yang terdeteksi dalam rekaman dengan *pitch template* dan skor kemiripan (*distance*) pada kedua rekaman tersebut.

3) Dataset

Dataset yang digunakan dalam penelitian ini adalah lagu yang diambil dari *corpus* MIR-QBSH yang dibuat oleh Laboratorium MIR (Multimedia Information Retrieval) di Departemen CSIE (*Computer Science and Infomation Engineering*) National Taiwan University, Taiwan [13]. Data lagu dalam bentuk .wav dengan spesifikasi berkas 8 kHz, 8 bit, mono. Untuk setiap berkas .wav, ada berkas .pv (*pitch vector*) yang sesuai. Berkas .pv berisi *pitch label* (dalam nomor MIDI yang mengkodekan *pitch*), dengan *frame size* = 256 dan *overlap* = 0, dan *frame* pertama dimulai dari sampel pertama berkas audio. Proses membuat *pitch label* dilakukan secara manual oleh pembuat *dataset*. Lagu-lagu yang ada dalam *dataset* ini dinyanyikan dalam bahasa Mandarin.

Dalam penelitian ini, digunakan lima lagu dengan judul dan jumlah rekaman sebagai berikut (lihat Tabel I). *Dataset* dibagi dalam 80% dan 20%, masing-masing untuk data latih dan data uji. Dari 80% data latih, data tersebut dibagi lagi menjadi 80% dan 20% untuk data latih dan validasi. Data validasi digunakan untuk melihat seberapa baik performa model ketika mendapatkan data yang baru untuk diestimasi. Hal ini digunakan untuk melihat seberapa besar *overfitting* yang terjadi pada model.

III. Hasil dan Pembahasan

A. Hasil Pengujian

Pengujian dilakukan dengan mengombinasikan seluruh nilai *epoch* dan nilai *learning rate* pada empat buah arsitektur yang berbeda berdasarkan jumlah *convolutional layer* dan jumlah *filter* yang ada pada masing-masing *convolutional layer*. Jumlah kombinasi adalah 96 buah, maka pada peneliti-

Tabel I

Keterangan Penggunaan *Dataset*

No	Judul Lagu	Total Data	Pembagian Data	
			Pelatihan	Pengujian
1	Twinkle, Twinkle, Little Star	159	127	32
2	Old MacDonald Had a Farm	152	121	31
3	Happy Birthday	170	136	34
4	Brother John (Are you sleeping?)	173	138	35
5	London Bridge Is Falling Down	145	116	29

an ini akan diujikan 96 buah kombinasi dari parameter-parameter tersebut.

Pada masing-masing arsitektur (lihat Gambar 6 hingga Gambar 9) diujikan 24 kombinasi parameter nilai *epoch* dan nilai *learning rate*. Dari 96 buah kombinasi tersebut, akan dicari kombinasi yang menghasilkan akurasi terbaik. Arsitektur dengan kombinasi terpilih akan dijadikan arsitektur yang digunakan untuk melakukan pengujian kemiripan. Hal ini bertujuan untuk membuktikan apakah akurasi yang tinggi dari hasil estimasi *pitch* memiliki ketepatan intonasi yang baik. Tabel II merangkum parameter pengujian yang dilakukan pada penelitian ini.

Gambar 4 hingga Gambar 7 menunjukkan empat arsitektur yang digunakan untuk melakukan estimasi *pitch* dengan CNN. Pada setiap pengujian dengan masing-masing arsitektur tersebut, akan dicari beberapa metrik penilaian yang dijadikan sebagai tolak ukur penilaian model, yaitu: akurasi (menggunakan data latih), akurasi validasi (menggunakan data validasi dari data latih), dan F1 *Score*. Semua hasil metrik tersebut dinyatakan dalam nilai rata-rata dari hasil perhitungan setiap *epoch*. Rangkuman hasil pengujian kombinasi seluruh parameter pada Tabel II ditunjukkan pada Tabel III sampai Tabel VIII yang menunjukkan rata-rata hasil pengujian per arsitektur yang digunakan.

Dengan mempertimbangkan nilai akurasi, akurasi validasi, dan F1 *Score* dari seluruh hasil pengujian arsitektur CNN tersebut, keterkaitan jumlah konvolusi, jumlah *filter*, *epoch*, dan *learning rate* secara menyeluruh terhadap akurasi sistem dapat dijelaskan sebagai berikut.

Semakin banyak dilakukan konvolusi dengan jumlah *filter* yang optimal pada masing-masing *convolutional layer*, arsitektur yang dibangun dapat dinyatakan semakin baik untuk melakukan estimasi *pitch*. Jumlah *filter* menunjukkan jumlah fitur yang diekstrak dari suatu data. Maka dari itu, jika *filter* semakin banyak, fitur yang dapat diekstrak dari data juga semakin banyak. Sementara itu, jumlah konvolusi menentukan berapa kali pengambilan fitur yang diambil dari suatu data. Kombinasi yang tepat antara jumlah konvolusi dengan jumlah *filter* akan menghasilkan akurasi yang baik. Meskipun begitu, kedua parameter tersebut belum cukup untuk menentukan model mana yang memiliki hasil terbaik. Secara konsep, fitur akan lebih banyak diambil pada arsitektur pertama karena memiliki jumlah *filter* yang lebih banyak. Karena ada parameter lain yang perlu dipertimbangkan, yaitu *epoch* dan *learning rate*, maka belum dapat diketahui secara pasti apakah arsitektur pertama sudah pasti menghasilkan akurasi yang lebih baik. Ada sebuah kasus di mana arsitektur keempat dapat menghasilkan rata-rata akurasi yang lebih tinggi daripada arsitektur pertama (kombinasi nilai *epoch* = 100 dan *learning rate* = 0,001 (10^{-3})). Pada arsitektur keempat dihasilkan rata-rata akurasi sebesar 87,4% dengan selisih rata-rata akurasi dan rata-rata akurasi validasi sebesar 37,11%. Pada arsitektur pertama dihasilkan rata-rata akurasi sebesar 87,06% dengan selisih rata-rata akurasi dan rata-rata akurasi validasi sebesar 51,402%). Maka dari itu, perlu dilakukan pencarian nilai *epoch* dan *learning rate* yang optimal untuk dapat memastikan jumlah konvolusi dan *filter* pada setiap arsitektur dinyatakan menghasilkan akurasi yang baik. Misalnya, pada arsitektur keempat, dengan nilai *epoch* dan *learning rate* yang sama, dihasilkan akurasi yang lebih rendah (94,394%) dan selisih antara rata-rata akurasi dan rata-rata akurasi validasi yang lebih tinggi (14,572%).

Berdasarkan hasil pengujian pada arsitektur CNN, akan dipilih beberapa buah kombinasi *epoch* dan *learning rate* yang terbaik untuk dapat dilakukan percobaan pengambilan *threshold*. Pengambilan model terbaik bergantung dari sudut pandang hasil pengujian yang dilakukan. Keputusan pengambilan model terbaik, sebelum melakukan pengujian *threshold* dengan DTW, didasarkan kepada dua pertimbangan:

1) Model memiliki rata-rata akurasi yang tinggi. Selisih antara rata-rata akurasi dan akurasi validasi tidak diperhatikan sehingga model dibiarkan mengalami sedikit *overfitting*.

TABEL II

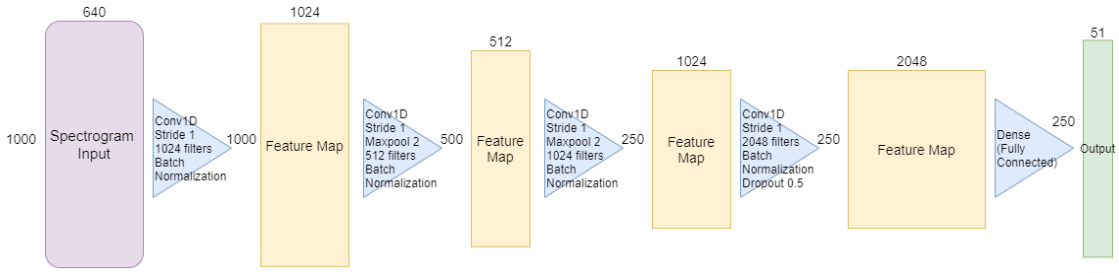
RANGKUMAN PARAMETER PENGUJIAN

Metode	Parameter Pengujian	Nilai yang Diujikan
CNN	*Epoch*	50, 100, 150, 200, 250, 300
	Learning rate	0,001; 0,0001; 0,0002; 0,00001
	Jumlah konvolusi dan *filter*	4 konvolusi - 4608 *filter* 4 konvolusi - 1920 *filter* 3 konvolusi - 3072 *filter* 3 konvolusi - 1536 *filter*
DTW	Threshold	500, 1000, 1500, 2000, 2500

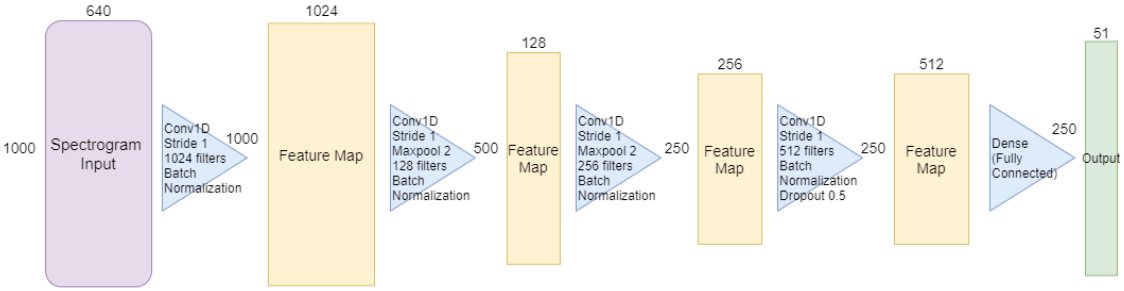
TABEL III

RATA-RATA HASIL PENGUJIAN – ARSITEKTUR PERTAMA

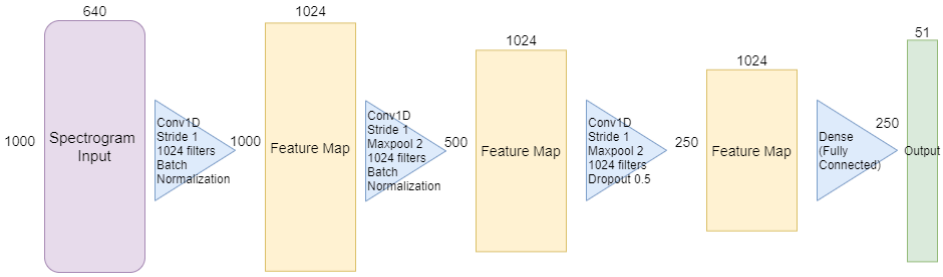
Learning Rate	*Epoch*	*Akurasi*	*F1 Score*	Akurasi Validasi
0,001	50	86,175%	86,190%	39,034%
	100	87,060%	87,063%	35,658%
	150	88,897%	88,908%	39,219%
	200	90,670%	90,675%	45,917%
	250	90,687%	90,688%	47,475%
	300	92,433%	92,434%	50,031%
0,0001	50	88,197%	88,079%	79,983%
	100	93,208%	93,143%	81,822%
	150	95,350%	95,305%	82,412%
	200	96,338%	96,307%	82,720%
	250	96,974%	96,947%	82,461%
	300	97,425%	97,400%	83,042%
0,0002	50	90,476%	90,436%	78,873%
	100	94,498%	94,476%	80,277%
	150	95,978%	95,953%	79,406%
	200	96,315%	96,301%	74,684%
	250	97,101%	97,088%	76,636%
	300	97,133%	97,122%	71,193%
0,00001	50	74,388%	73,682%	73,027%
	100	80,323%	79,959%	78,418%
	150	83,275%	83,097%	80,392%
	200	85,213%	85,068%	81,498%
	250	86,924%	86,815%	82,195%
	300	88,233%	88,140%	82,689%



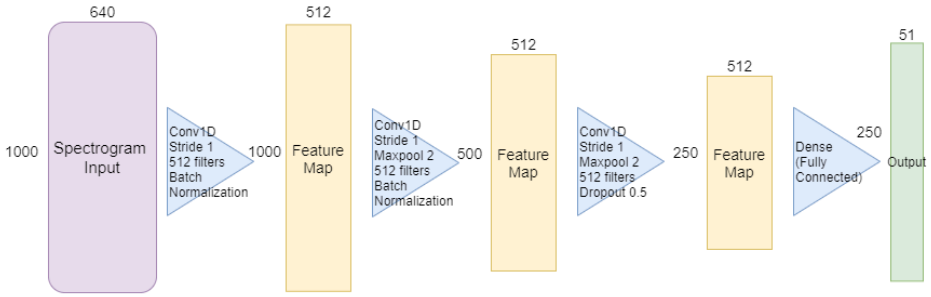
Gambar 4 Rancangan arsitektur CNN yang pertama



Gambar 5 Rancangan arsitektur CNN yang kedua



Gambar 6 Rancangan arsitektur CNN yang ketiga



Gambar 7 Rancangan arsitektur CNN yang keempat

2) Model memiliki rata-rata akurasi yang tidak terlalu tinggi, tetapi memiliki selisih rata-rata akurasi dan validasi yang kecil. Hal ini berarti mencegah penggunaan model yang *overfitting*.

Berdasarkan pertimbangan tersebut, dipilih dua buah model untuk pengujian *threshold* dengan DTW, yaitu:

1) Model CNN pada arsitektur pertama, *epoch* = 300 dan *learning rate* = 0,0001 (10^{-4}) dengan rata-rata akurasi 97,425%.
2) Model CNN pada arsitektur keempat, *epoch* = 300 dan *learning rate* = 0,00001 (10^{-5}) dengan selisih rata-rata akurasi dan akurasi validasi sebesar 0,449% dan rata-rata akurasi sebesar 81,077%.

Berdasarkan hasil pengujian *threshold* dengan metode DTW, dapat diambil beberapa kesimpulan:

1) Nilai *threshold* yang dapat digunakan untuk melakukan pengujian kemiripan dengan optimal berada di rentang 1000-1500.

TABEL IV

RATA-RATA HASIL PENGUJIAN – ARSITEKTUR KEDUA

Learning Rate	*Epoch*	Akurasi	F1 *Score*	Akurasi Validasi
0,001	50	85,649%	85,530%	46,317%
	100	87,710%	87,655%	44,447%
	150	89,187%	89,164%	50,576%
	200	90,674%	90,661%	51,947%
	250	91,718%	91,699%	55,974%
	300	92,511%	92,487%	59,435%
0,0001	50	81,314%	80,882%	76,666%
	100	85,657%	85,449%	80,237%
	150	88,277%	88,142%	81,683%
	200	90,236%	90,139%	82,149%
	250	91,801%	91,737%	82,184%
	300	93,019%	92,943%	82,666%
0,0002	50	84,068%	83,764%	77,591%
	100	87,863%	87,739%	80,060%
	150	90,338%	90,267%	79,447%
	200	91,869%	91,801%	79,937%
	250	93,204%	93,156%	79,019%
	300	94,079%	94,033%	78,470%
0,00001	50	60,389%	56,964%	60,451%
	100	69,913%	67,715%	70,820%
	150	74,243%	73,004%	74,423%
	200	77,048%	76,007%	76,634%
	250	78,806%	77,942%	78,048%
	300	80,301%	79,664%	79,485%

TABEL V

RATA-RATA HASIL PENGUJIAN – ARSITEKTUR KETIGA

Learning Rate	*Epoch*	Akurasi	F1 *Score*	Akurasi Validasi
0,001	50	85,553%	85,504%	41,951%
	100	86,969%	86,933%	40,973%
	150	87,847%	87,831%	41,014%
	200	88,518%	88,493%	41,851%
	250	89,405%	89,399%	44,592%
	300	89,769%	89,758%	45,173%
0,0001	50	85,483%	85,146%	81,582%
	100	89,590%	89,428%	82,881%
	150	92,556%	92,408%	83,511%
	200	94,032%	93,926%	83,489%
	250	95,266%	95,174%	82,998%
	300	95,933%	95,853%	82,584%
0,0002	50	87,565%	87,397%	79,082%
	100	91,558%	91,446%	78,117%
	150	93,686%	93,591%	74,133%
	200	94,897%	94,830%	73,539%
	250	95,777%	95,713%	71,385%
	300	96,298%	96,244%	70,019%
0,00001	50	73,683%	71,688%	73,396%
	100	78,605%	77,473%	77,866%
	150	81,882%	81,124%	80,690%
	200	83,200%	82,609%	81,510%
	250	84,527%	84,078%	82,310%
	300	85,670%	85,320%	82,899%

TABEL VI

RATA-RATA HASIL PENGUJIAN – ARSITEKTUR KEEMPAT

Learning Rate	*Epoch*	Akurasi	F1 *Score*	Akurasi Validasi
0,001	50	85,756%	85,591%	49,784%
	100	87,400%	87,304%	50,290%
	150	88,390%	88,333%	49,191%
	200	89,271%	89,220%	51,921%
	250	90,022%	89,988%	53,946%
	300	90,709%	90,682%	53,369%
0,0001	50	81,431%	80,773%	79,448%
	100	85,360%	85,055%	81,974%
	150	87,345%	87,119%	83,033%
	200	89,597%	89,427%	83,589%
	250	90,939%	90,801%	83,529%
	300	92,196%	92,074%	83,823%
0,0002	50	84,087%	83,647%	80,171%
	100	87,935%	87,699%	81,930%
	150	90,308%	90,146%	79,878%
	200	91,984%	91,868%	78,956%
	250	93,400%	93,299%	79,676%
	300	94,394%	94,305%	79,822%
0,00001	50	63,022%	57,498%	65,457%
	100	73,429%	70,409%	74,207%
	150	76,453%	74,335%	76,807%
	200	78,364%	76,785%	78,528%
	250	79,957%	78,572%	79,514%
	300	81,077%	79,873%	80,628%

2) Nilai tersebut dipilih karena dinilai dapat melakukan pengujian kemiripan secara baik. Data rekaman dengan kualitas *pitch* kurang baik dapat dikategorikan sebagai rekaman yang tidak mirip dengan *template* pada *dataset*.
3) Jika diambil nilai *threshold* yang terlalu tinggi, maka rekaman yang memiliki kualitas *pitch* kurang baik dapat dianggap mirip. Hal ini akan berdampak pada performa sistem yang kurang baik karena kualitas penyanyi yang baik tidak dapat ditentukan.
4) Jika diambil nilai *threshold* yang terlalu rendah, maka rekaman masukan cenderung harus identik dengan *template* pada *dataset*. Hal ini sangat sulit dicapai pada proses rekaman yang sesungguhnya.

Pengambilan nilai *threshold* pada rentang 1000-1500 dilakukan dengan cara melakukan perbandingan antara hasil estimasi *pitch* dengan data *pitch* dari *dataset*. Perbandingan ini dilakukan untuk melihat berapa banyak *pitch* yang terestimasi dengan tepat (dalam Tabel VII: kolom *match*). Selain itu, akan dilihat pula berapa banyak jumlah *pitch* yang tidak terestimasi (data yang kurang dalam Tabel VII: kolom *diff1*) dan yang seharusnya tidak terestimasi (data yang lebih dalam Tabel VII: kolom *diff2*). Tabel VII menunjukkan contoh hasil pengujian ketika menentukan *threshold* skor kemiripan dengan metode DTW menggunakan lagu *Twinkle-twinkle Little Star* dengan model CNN yang pertama.

TABEL VII

CONTOH HASIL PENGUJIAN *THRESHOLD* DENGAN DTW

Rekaman	*Distance*	*Match*	*Diff1*	*Diff2*	*Percentage*
1	2	14	0	0	100,00%
2	521	13	1	2	90,48%
3	550	12	2	1	88,10%
4	637	11	3	0	85,71%
5	646	12	2	0	90,48%
6	739	12	2	0	90,48%
7	860	10	4	2	76,19%
8	999	11	3	1	83,33%
9	1055	13	1	0	95,24%
10	1150	12	2	0	90,48%
11	1222	11	3	1	83,33%
12	1250	11	3	3	78,57%
13	1298	13	1	0	95,24%
14	1299	8	6	4	61,90%
15	1370	11	3	2	80,95%
16	1450	11	3	2	80,95%
17	1557	12	2	3	83,33%
18	1680	8	6	6	57,14%
19	1707	8	6	3	64,29%
20	1817	11	3	5	73,81%
21	1819	11	3	2	80.95%
22	1847	7	7	5	54,76%
23	1931	10	4	3	73,81%
24	2023	6	8	6	47,62%
25	2054	11	3	2	80,95%
26	2078	9	5	4	66,67%
27	2217	8	6	4	61,90%
28	2343	4	10	8	33,33%
29	2370	4	10	9	30,95%
30	2502	3	11	10	23,81%
31	2578	4	10	9	30,95%
32	2871	6	8	7	45,24%

IV. SIMPULAN

Metode CNN dapat melakukan estimasi *pitch* pada objek suara penyanyi dengan akurasi rata-rata tertinggi sebesar 97,425% pada arsitektur pertama, *epoch* = 300 dan *learning rate* = 0,0001 (10^{-4}) dengan selisih antara rata-rata akurasi dan rata-rata akurasi validasi sebesar 14,383%.

Nilai *threshold* yang optimal untuk menentukan kemiripan sebuah rekaman lagu dengan metode DTW berada dalam rentang 1000-1500. Nilai *threshold* tersebut dipilih agar kesalahan-kesalahan yang tidak terlalu besar dalam rekaman dapat ditoleransi. Dalam kondisi nyata untuk menghasilkan rekaman nyanyian dengan ketepatan intonasi yang sempurna tidak mudah. Penggunaan metode DTW dalam penelitian ini belum bisa menghubungkan *pitch* yang sama dalam oktaf yang berbeda dalam proses pengujian kemiripan. Nada yang sama dalam oktaf yang berbeda akan dikenali sebagai nada yang berbeda.

Dalam penelitian lanjutan, dapat digunakan *dataset* dengan rekaman audio yang lebih seragam, baik secara *pitch* maupun *tempo* dengan harapan dapat mengurangi kompleksitas arsitektur CNN. Selain itu, dapat ditambahkan proses *musical key estimation* untuk mengetahui tangga nada yang digunakan dalam rekaman sebelum dimasukkan ke proses estimasi *pitch*. Metode DTW juga dapat dikembangkan untuk dapat mengelompokkan *pitch* yang sama dalam oktaf yang berbeda. Metode CNN dapat dipertimbangkan untuk digabung dengan metode lain dalam proses estimasi *pitch*.

DAFTAR REFERENSI

[1] C. Gupta, H. Li, dan Ye Wang, "Perceptual evaluation of singing quality," dalam *2017 Proceedings of APSIPA Annual Summit and Conference*, Kuala Lumpur Sentral, Malaysia, 12-15 Desember 2017.

[2] M. Muller, *Fundamentals of Music Processing*, edisi ke-1, Erlangen: Springer International Publishing, 2015, hlm. i, 18-29, 57-68, 98-102, 131-140.

[3] D. Tatarenkov dan D. Podolsky, "Deep learning for singing processing: achievements, challenges and impact on singers and listeners," dalam *Proceedings of the* 35th *International Conference on Machine Learning*, Stockholm, Sweden, 10-15 Juli 2018.

[4] S. R. Kadiri dan B. Yegnanarayana, "Estimation of fundamental frequency from singing voice using harmonics of impulse-like excitation source," dalam *Interspeech 2018*, Hydreabad, 2-6 September 2018.

[5] Jong Wook Kim, J. Salamon, P. Li, dan J. Pablo Bello, "CREPE: a convolutional representation for pitch estimation," dalam *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP*), Calgary TELUS Convention Centre, Calgary, Canada, 15-20 April 2018, hlm.161-165.

[6] R. M. Bittner, B. McFee, J. Salamon, P. Li, dan J. Pablo Bello, "Deep salience representation for f_0 estimation in polyphonic music," dalam 18th *International Society for Music Information Retrieval Conference*, China, 23-27 Oktober 2017.

[7] H. Su, H. Zhang, X. Zhang, dan G. Gao, "Convolutional Neural Network for robust pitch determination," dalam *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 20-25 Maret 2016.

[8] Goodfellow, Y. Bengio, dan A. Courville, *Deep Learning, An MIT Press book*, MIT Press, 2016.

[9] F. Chollet, *Deep Learning with Python*, edisi ke-1. New York: Manning Publications, 2018, hlm. 225-226.

[10] H. Kinsley dan D. Kukieła, *"Neural Networks from Scratch in Python"*, edisi ke-1. Harrison Kinsley, 2020, hlm. 108, 333-358.

[11] S. Khan, H. Rahmani, Syed Afaq Ali Shah, dan M. Bennamoun, *"A Guide to Convolutional Neural Networks for Computer Vision"*, edisi ke-1, Gerard´ Medioni and Sven Dickinson, Ed. California: Morgan and Claypool, 2018, hlm. 45, 53, 56, 67-80.

[12] H. Habibi Aghdam dan E. Jahani Heravi, *Guide to Convolutional Neural Networks*, edisi ke-1. Switzerland: Springer International Publishing AG, 2017, hlm. 108-111, 118-120.

[13] J-S. R. Jang, "MIR-QBSH Corpus", 2003-2009. [Daring]. Tersedia: http://mirlab.org/dataset/public/MIR-QBSH-corpus.rar. [10 September 2020].

Dionisius Pratama, menerima gelar Sarjana Komputer dari Institut Teknologi Harapan Bangsa (ITHB) Bandung tahun 2021. Saat ini sedang menempuh studi pascasarjana jurusan Intelijen Bisnis di ITB dan juga sebagai dosen luar biasa di Prodi Informatika ITHB.

Hery Heryanto, menerima gelar Sarjana Komputer tahun 2004 dan Magister Sistem Informasi tahun 2010 dari STMIK LIKMI. Tahun 2016 menerima gelar Doktor dari STEI ITB. Saat ini aktif sebagai dosen di Prodi Informatika ITHB. Minat penelitian pada bidang ilmu *speech processing* dan basis data.

Hans Christian Kurniawan, menerima gelar Sarjana Teknik Informatika dari Institut Teknologi Harapan Bangsa (ITHB) Bandung tahun 2016 dan Magister Informatika dari ITB pada tahun 2019. Saat ini aktif sebagai pengajar di Prodi Informatika ITHB dan juga sebagai *Engineering Lead* di Bukalapak.