

Implementasi Metode *Recurrent Neural Network* pada Pengklasifikasian Kualitas Telur Puyuh

Exel Defrisco Tarkus¹⁾, Sherwin R.U.A. Sompie²⁾, Agustinus Jacobus³⁾
 Teknik Elektro Universitas Sam Ratulangi Manado, Indonesia, Jl. Kampus Bahu-Unsrat manado, 95115
 Email : exeldtarkus@gmail.com¹⁾, aldo@unsrat.ac.id²⁾, a.jacobus@unsrat.ac.id³⁾

Abstrak — Telur merupakan salah satu sumber makanan yang banyak mengandung protein sekaligus disukai oleh banyak orang dan juga mempunyai berbagai macam mamfaat bagi manusia. *RNN* adalah jaringan saraf berulang atau jaringan saraf tiruan yang pemerosesannya di panggil secara berulang-ulang untuk memproses masukan yang biasa adalah data sekuensial. Setelah mengambil suara ketukan, pemerosesan akan memasuki tahap ekstraksi fitur, *training data* dan prediksi data. Hasil dari proses ekstraksi fitur berupa nilai statistika, *FFT* dan *MFCC* yang kemudian akan di *training* dengan metode *RNN* dan menghasilkan file *model.h5*. Untuk hasil prediksi *precision* diatas 75 %, *recall* diatas 81 % dan akurasi diatas 87 %.

Kata Kunci — Kualitas Telur, *Recurrent Neural Network*, *mfcc*

Abstract — *Eggs are one of the food sources that contain lots of protein and at the same time favored by many people and also have a variety of benefits for humans. RNN is a recurrent neural network or artificial neural network that the processing is called repeatedly to process input which is usually sequential data . After taking the sound of quail, processing will enter the feature extraction, training data and data prediction stages. The results of the feature extraction process are statistical values, FFT and MFCC which will then be trained with the RNN method and produce a model.h5 file. For precision prediction results above 75%, recall above 81% and accuracy above 87%.*

Keywords — *Egg Quality, Recurrent Neural Networks, mfcc*

I. PENDAHULUAN

Telur puyuh merupakan salah satu sumber protein hewani yang lebih baik disbanding dengan kandungan protein telur ayam dan telur itik, juga relatif murah dibandingkan dengan telur ayam, daging sapi, dan lain-lain. Dilihat dari nilai gizinya, telur puyuh mengandung 13,6% protein dan 8,2% lemak. [1] Struktur dari telur puyuh secara umum tidak berbeda dengan struktur telur ayam yang terdiri atas tiga bagian, yaitu putih telur (58%), kuning telur (31%), dan kerabang telur (11%).

Di Indonesia, telur puyuh sendiri adalah salah satu sumber makanan. Namun seperti bahan makanan yang lain telur puyuh juga dapat busuk dan juga retak. Kekuatan dan ketebalan kulit telur puyuh menjadi pelindung kualitas telur tersebut terhadap serangan-serangan bakteri maupun hal yang mengganggu kualitas telur puyuh itu sendiri. Dan terkadang masih terdapat telur-telur yang retak yang juga masih di jual dipasaran sehingga dapat merugikan konsumen dan juga produsen penyedia telur, maka dari itu teknik klasifikasi yang baik dapat menjawab hal tersebut. Dengan menggunakan salah satu metode deep learning yang adalah salah satu cabang dari ilmu pembelajaran machine learning yang terdiri algoritma pemodelan abstraksi tingkat

tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam Pembelajaran dalam dapat digunakan baik untuk kebutuhan pembelajaran terarah, pembelajaran tak terarah dan semi-terarah (semi-supervised learning) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. Teknik klasifikasi telur dengan menggunakan teknologi deep learning yang semakin berkembang menjadi pilihan yang tepat dan cepat dalam melakukan klasifikasi pada telur, salah satu teknik penyortiran yang ditawarkan adalah dengan menggunakan teknik pengolahan suara sehingga dapat mengklasifikasikan telur yang baik untuk konsumen.

Tujuan penelitian ini adalah untuk mengklasifikasi telur puyuh dengan pengolahan suara berdasarkan metode *Recurrent Neural Network* (RNN).

II. LANDASAN TEORI

A. Penelitian Terkait

Terdapat beberapa penelitian sebelumnya yang berhubungan dengan implementasi metode RNN pada penentuan kualitas telur dan dijabarkan sebagai berikut:

- 1) Klasifikasi Telur Ayam Berbasis Pengelolaan Suara, oleh Devi Sahati. [2] Penelitian yang menggunakan metode *Support Vector Machine*. Dengan pengolahan suara, telur yang diteliti akan diambil sampel suaranya dengan cara diketuk cangkangnya. Setelah sampel didapat, pengolahan suara akan dilakukan, yang terdiri dari pengekstrasian ciri telur yang bagus maupun ciri telur yang buruk, kemudian akan dilakukan pengklasifikasian antara telur yang bagus dan telur yang buruk, dengan hasil yang didapat yaitu akurasi dari klasifikasi pada penelitian ini adalah 95% pada pengujian dengan menggunakan data latih, dan 89,47% pada pengujian dengan menggunakan data baru.
- 2) Prediksi Kunjungan Wisatawan Dengan *Recurrent Neural Network Extended Kalman Filter* oleh Rizal, Ahmad Ashril. [3] Untuk prediksi kunjungan wisatawan penelitian ini memiliki beberapa tahap diantaranya pre-processing, training RNN, dan proses testing dengan hasil training RNN dengan *Extended Kalman Filter* untuk prediksi memiliki tingkat akurasi terbaik sebesar 64,37% dan hasil testing RNN-EKF pada kasus prediksi memberikan hasil akurasi terbaik sebesar 62,91%.
- 3) Aplikasi *Model Recurrent Neural Network dan Recurrent Neuro Fuzzy* Untuk Peramalan Banyaknya Penumpang Kereta Api Jabotabek oleh Hermawan. [4] Penelitian ini menggunakan dua metode untuk proses peramalan data, peramalan dilakukan dengan memilih input, memilih struktur model yang terbaik setelah itu memilih

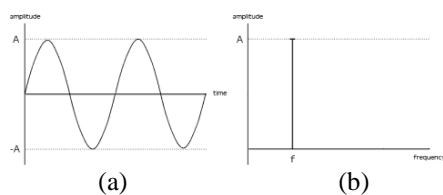
menyerderhanakan koefisien dan melakukan kesesuaian model dan didapatkan hasil MAPE training 1,2617% dan MSE training 29500000. Serta MAPE testing 3,785% dan MSE testing 2610000000.

- 4) Identifikasi Judul Berita Clickbait Berbahasa Indonesia Dengan Algoritma *Long Short Term Memory* (LSTM) *Recurrent Neural Network* oleh Ibnu. [5] Penelitian ini akan mengidentifikasi berita dengan LSTM-RNN dengan tahap input data, normalisasi text, proses training, dan pengujian dengan hasil pelatihan dan validasi model dengan menambahkan Dropout Layer dan Callback Function mampu menghindari model dari overfitting dengan menghasilkan akurasi 82% dan loss 38%.
- 5) Dalam Publikasi oleh Suyudi, Esmeralda, Muspupah. [6] Prediksi Harga Saham Menggunakan Metode *Recurrent Neural Network*. Sistem prediksi harga saham terdiri dari tiga tahap. Tahap pertama yaitu pra-proses data yang terdiri dari ekstraksi fitur, normalisasi, ascending dan segmentasi. Tahap kedua yaitu proses pelatihan data menggunakan RNN, kemudian tahap ketiga yaitu proses pengujian. Hasil dari penelitian ini menunjukkan bahwa prediksi harga saham menggunakan RNN mendapatkan hasil kurang baik. Menggunakan optimasi Adam dengan nilai learning rate 0.001 dan epoch 200 memperoleh akurasi untuk data latih sebesar 94.16% dan data uji sebesar 55.26%.

B. Gelombang Suara

Gelombang suara merupakan gelombang longitudinal yang merambat melalui medium seperti medium padat, cair, atau gas. [7] Gelombang suara merupakan gelombang analog yang apabila diolah menggunakan peralatan elektronik, gelombang tersebut harus melalui tahap digitalisasi sehingga gelombang tersebut berupa data *digital*. [8] Pendengaran manusia terbatas pada daerah frekuensi 16 Hz sampai 20 kHz. Namun, pendengaran akan lebih sensitif pada suara yang frekuensinya antara 1 kHz dan 4 kHz. Suara di atas 20 kHz tidak akan bisa didengar, sedangkan di bawah 16 Hz sinyal suara akan terdengar lambat dan berderak-derak. Antara 0 dB sampai 130 dB adalah level suara yang aman didengar. Di atas level 130 dB, suara tersebut akan menyakitkan telinga, dan dapat pula mengakibatkan selaput telinga pecah.

Sinyal bisa dilihat dari dua sudut pandang, yaitu *time domain* dan *frequency domain*. Gambar 1 (a), *time-domain* menunjukkan bagaimana bentuk sinyal dalam rentang waktu tertentu. Dan untuk Gambar 1 (b), *frequency-domain* menunjukkan seberapa besar sinyal yang ada dalam setiap *frequency band* dalam rentang beberapa frekuensi.



Gambar 1 Sinyal dalam *Time-domain* (a), dan *Frequency-domain* (b)

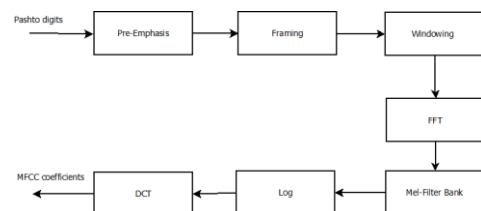
Sinyal suara dibentuk dari kombinasi berbagai frekuensi pada berbagai amplitudo dan fasa. Untuk dapat melakukan proses pengolahan sinyal suara, dilakukan proses yang

disebut penyampling atau sampling. Sampling adalah mengukur masukan sinyal suara pada sela waktu tertentu, kemudian mengkonversinya ke dalam skala tersendiri, dan menyimpannya.

C. Mel Frequency Cepstrum Coefficient

MFCC adalah salah satu metode ekstraksi fitur dalam bidang *speech technology* yang banyak digunakan untuk pengenalan suara atau *audio recognition* maupun *speech recognition*. MFCC merupakan proses *feature extraction* yang mengkonversi sinyal suara menjadi beberapa parameter. Keunggulan dari metode ini adalah:

1. Mampu untuk menangkap karakteristik suara yang sangat penting bagi pengenalan suara. Atau dengan kata lain mampu menangkap informasi-informasi penting yang terkandung dalam sinyal suara.
2. Menghasilkan data seminimal mungkin, tanpa menghilangkan informasi-informasi penting yang ada.
3. Dapat mengadaptasi organ pendengaran manusia dalam melakukan persepsi terhadap signal suara.



Gambar 2 Blok Diagram MFCC

Perhitungan yang dilakukan dalam MFCC menggunakan dasar perhitungan *short-term analysis*. Hal ini dilakukan mengingat sinyal suara bersifat *stationary*. [9] MFCC *features extraction* sebenarnya merupakan adaptasi dari sistem pendengaran manusia, dimana sinyal suara akan di *filter* secara linear dengan frekuensi rendah (kurang dari 1000Hz) dan secara logaritmik untuk frekuensi yang tinggi (diatas 1000Hz).

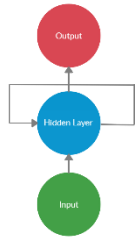
D. Recurrent Neural Network

RNN termasuk dalam jenis jaringan pemodelan *Neural Network*. Selain RNN, *Multilayer Perceptron* (MPL) dan *Feedforward Neural Network* (FNN) termasuk dalam jenis pemodelan *Neural Network*. [10] FNN telah banyak diaplikasikan untuk melakukan prediksi atau klasifikasi data selama bertahun-tahun yang lalu. Salah satu tipe dari *neural network* yang dikembangkan dari dasar pemikiran FNN adalah RNN.

RNN yang juga disebut jaringan umpan balik adalah jenis jaringan pada *neural network* dimana terdapat *loop* sebagai koneksi umpan balik dalam jaringan. [11] Jaringan RNN adalah jaringan yang mengakomodasi *output* jaringan untuk menjadi *input* pada jaringan tersebut yang kemudian digunakan untuk menghasilkan *output* yang baru. RNN merupakan jaringan yang dinamis karena perilaku jaringan tidak tergantung pada input saat ini saja melainkan pada operasi jaringan sebelumnya. Selain itu, RNN biasanya digunakan untuk menyelesaikan tugas terkait dengan data *time series*.

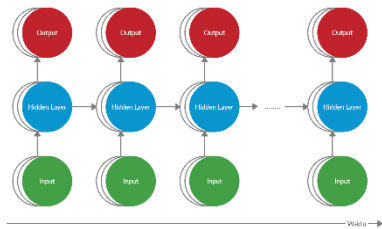
Pembuatan RNN sebenarnya dibuat untuk data-data yang bersifat *sequential* atau bertahap. Penggunaan *neural network* [12] biasanya semua *input* dan *output* tidak bergantung satu sama lain, maka akan terjadi penumpukan tugas pada *neural network* sangat banyak dan bertumpuk, sedangkan jika ingin membangun sebuah aplikasi yang

mengelola data ilmiah dengan data sampel berbentuk *time series* diperlukan algoritma yang dapat *handle* data tersebut dengan cepat dan tepat, maka RNN dapat melakukan hal itu karena RNN melakukan tugas yang sama pada setiap elemen data di sebuah urutan, lalu memproses *outputnya* yang mengacu pada hasil komputasi sebelumnya. Maka secara teori RNN mampu menggunakan informasi yang telah direkam sebelumnya yang panjang urutannya beragam-
 ragam. [13] Jaringan *node* pada RNN dimasukkan ke dalam *layer* yang berurutan. Setiap *node* dalam lapisan tertentu terhubung dengan koneksi terarah ke setiap *node* lain dilapisan *layer* berikutnya secara berturut-turut, setiap *node* juga memiliki aktivasi yang bernilai nyata yang bervariasi tergantung waktu.



Gambar 3 Arsitektur RNN

Diagram arsitektur pada pemrosesan RNN, terlihat bahwa lingkaran hijau merupakan *input layer* atau masuknya data yang akan dilakukan komputasi pada RNN, setelah itu masuk ke lingkaran biru yang merupakan *hidden layer* terlihat ada garis *looping* yang kembali pada lingkaran biru tersebut, pola *looping* ini memungkinkan RNN untuk menyimpan ingatan sementara yang nantinya akan dipakai dalam pemrosesan data berikutnya, dan juga yang terakhir ada *output layer* yang nantinya adalah hasil dari proses komputasi RNN.



Gambar 4 Diagram Arsitektur RNN Banyak Layer

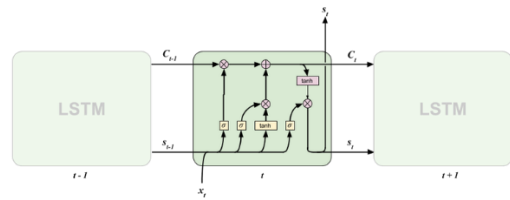
Arsitektur RNN menunjukkan bagaimana RNN yang bekerja dengan banyak layer dengan perulangan hidden state sebanyak data waktu yang disediakan $x_t = \text{input}$, $s_t = \text{hidden layer}$, $O_t = \text{Output}$ Maka dalam *hidden layer* dapat menyimpan memori dari kalkulasi dengan rumus:

$$s_t = f(Ux_t + Ws_{t-1}) \tag{1}$$

Dimana fungsi f biasanya adalah non-linearitas *tanh* atau *ReLU*, s_{t-1} digunakan untuk menghitung nilai *hidden layer* yang pertama, biasanya pada inialisasi selalu diawali dari 0 (nol), sedangkan u adalah data vektor *input* pada RNN dan w adalah data vektor *output* pada RNN.

E. Long Short Term Memory

LSTM merupakan salah satu arsitektur *recurrent neural network*, LSTM diciptakan oleh Hochreiter & Schmidhuber dan kemudian dikembangkan dan dipopulerkan oleh banyak periset. Seperti RNN, jaringan LSTM (*LSTM network*) juga terdiri dari modul-modul dengan pemrosesan berulang. Ringkasnya, [14] LSTM menambahkan sebuah proses seleksi di dalam *cell* (kotak kontrol) nya sehingga bisa menyeleksi informasi mana yang layak untuk diteruskan, sekaligus menjadi solusi bagi permasalahan *vanishing gradient*.



Gambar 5 Diagram Arsitektur LSTM

Gambar arsitektur dari RNN menjadi LSTM maka *hidden layer* pada RNN akan berbentuk seperti *cell* yang berfungsi untuk mengingat peristiwa masa lalu. Dengan memperhatikan lebih detail apa yang terjadi di dalam *hidden layer*, maka perlu membedah isi dari perceptron X_t . Juga terdapat notasi baru I_t (dibaca *input* saat waktu t) yang merupakan input untuk perceptron X_t . Fungsi aktivasi di dalamnya yang digunakan bisa *sigmoid* atau *hyperbolic (tanh)*. [15] LSTM dibagi menjadi beberapa gerbang diantaranya *Forget Gate*, *Input Gate*, *Cell Gate* dan *Output Gate* yang mempunyai tugasnya masing-masing. Pada *forget gate* informasi pada setiap data masukan akan diolah dan dipilih data mana saja yang akan disimpan atau dibuang pada *memory cell*. Fungsi aktivasi yang digunakan pada *forget gate* adalah fungsi *sigmoid* dimana keluaran dari *forget gate* antara 0 dan 1. Jika keluarannya 0 maka data tidak akan disimpan dan sebaliknya jika keluarannya 1 maka data akan disimpan. Dengan rumus:

$$F_{gate} = \sigma(W_f \cdot [S_{t-1}, X_t] + b_f) \tag{2}$$

Pada *input gate* terdapat dua gerbang yang dilaksanakan, pertama akan diputuskan nilai mana yang akan diperbaharui menggunakan fungsi *sigmoid*. Kedua mengubah nilai dengan fungsi *tanh* yang akan membuat vektor dengan nilai yang baru dan yang akan disimpan pada *memory cell*. Dengan rumus sebagai berikut:

$$i_t = \sigma(W_i \cdot [S_{t-1}, X_t] + b_i) \tag{3}$$

$$l_c = \tanh(W_c \cdot [S_{t-1}, X_t] + b_c) \tag{4}$$

$$I_{gate} = i_t * l_c \tag{5}$$

Pada *cell gate* akan mengganti nilai pada *memory cell* sebelumnya dengan nilai *memory cell* yang baru. Dimana nilai ini didapatkan dari menggabungkan nilai yang terdapat pada *forget gate* dan *input gate*. Dengan rumus sebagai berikut:

$$C_{Memory} = f_{gate} * C_{t-1} + i_{gate} \tag{6}$$

Pada *output gates* terdapat dua *gates* yang akan dilaksanakan, pertama akan diputuskan nilai pada bagian *memory cell* mana yang akan dilakukan dengan menggunakan fungsi aktivasi *sigmoid*. Selanjutnya akan ditempatkan nilai pada *memory cell* dengan menggunakan fungsi aktivasi *tanh*. Terakhir kedua *gate* dikalikan sehingga menghasilkan nilai yang akan dikeluarkan dengan rumus:

$$O_{gate} = \sigma(W_o \cdot [S_{t-1}, X_t] + b_o) * \tanh(C_{memory}) \tag{7}$$

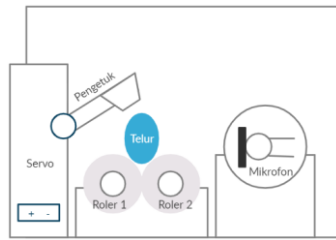
Dengan mendapatkan nilai C_{Memory} dan O_{gate} maka hasil *outputnya* akan diteruskan ke *layer LSTM* selanjutnya sampai data yang diinput habis.

III. METODOLOGI PENELITIAN

Tahapan penelitian yang dilalui dimulai dari desain dan pembuatan alat perekam bunyi ketukan, pengumpulan data suara, pengolahan data suara, pengujian dan evaluasi. Tahapan tersebut diuraikan lebih rinci sebagai berikut

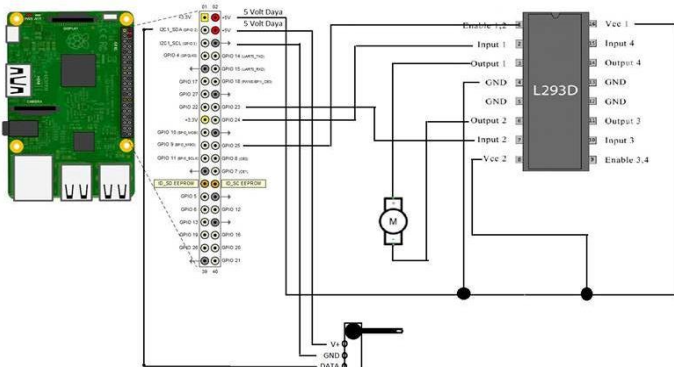
A. Perancangan Alat

Tahap ini merupakan tahap perancangan suster bagaimana cara kerja agar alat dapat merekam telur pad proses klasifikasi telur puyuh.



Gambar 6 Desain Alat

Gambar alat tampak depan dimana dalam alat tersebut terdapat servo yang sudah dipasangi pengetuk untuk mengetuk telur puyuh. Roler1 dan roler2 adalah penampang tempat. Terdapat juga mikrofone kondensor yang akan merekam ketukan telur pada saat proses perekaman telur. Dan juga terdapat rangkaian alat dimana alat akan dikoneksikan dengan *raspberrypi*



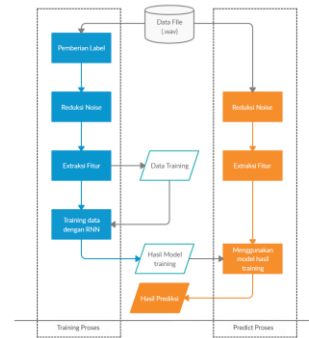
Gambar 7 Rangkaian Alat

Pada Gambar 7 terlihat pada raspberry untuk menggerakkan servo digunakan pin 3 (GPIO 2) untuk mentransfer data ke servo, pin 2 (power 5 v) sebagai daya 5 volt pada servo dan pin 6 (GND) sebagai ground pada servo. Sedangkan untuk menggerakkan motor diperlukan IC (Intergrate Circuit) dengan menggunakan pin 4 (power 5v) yang dihubungkan dengan pin 14 pada IC sebagai sumber daya pada motor, pin 16(GPIO23) yang terhubung dengan pin pada IC sebagai input negatif yang juga terhubung dengan kabel negatif motor DC, pin 18 (GPIO24) yang terhubung dengan pin 2 IC sebagai input positif dan juga terhubung dengan kabel positif motor DC, pin 22 (GPIO 25) yang terhubung dengan pin 1 IC sebagai input enable 1,2 dan pin 14 (GND) yang terhubung dengan pin 4 IC sebagai ground data.

B. Pengumpulan Data

Pada tahap pengumpulan data merupakan tahap pengambilan dataset yang kemudian akan di *training* dengan menggunakan agloritma *recurrent neural network*. Dimana telur yang diambil terdiri dari sampel telur dengan kualitas cangkang baik dan sampel telur dengan kualitas cangkang buruk, proses pengambilan sampel telur dilakukan dengan merekam suara hasil ketukan telur dan disimpan dengan format (.wav)

C. Prinsip Kerja Sistem



Gambar 8 Prinsip Kerja Sistem

Prinsip kerja sistem yang memiliki dua proses alur data yang berbeda dimana proses dengan warna biru adalah proses melakukan *training* dan warna *orange* adalah proses prediksi data dengan data yang baru. Proses training data terdapat satu proses yaitu data *training* yang merupakan proses untuk membuat data-data untuk pemerosesan model, dengan meng-*input* kumpulan data audio yang belum diolah dan belum terstruktur. Kemudian memberikan label pada data yang dikumpulkan tersebut, selanjutnya data tersebut akan mengalami reduksi *noise* yang dapat mengganggu tingkat akurasi. Setelah menghilangkan *noise* data tersebut akan diextraksi fitur-fiturnya dengan menggunakan FFT dan MFCC sehingga fitur-fitur tersebut akan di *training* dengan menggunakan metode *recurrent neural network* dan menghasilkan model yang akan disimpan untuk proses prediksi. Berbeda dengan proses training data. Proses prediksi dimulai dengan meng-*input* data suara telur yang baru yang kemudian akan di hilangkan *noise*-nya, kemudian fitur dari suara baru tersebut akan di extraksi dan akan dibandingkan dengan model yang didapat pada proses *training data*. Setelah perbandingan model selesai maka akan didapatkan *output* berupa klasifikasi telur “good” atau “bad”

D. Pengujian dan Evaluasi

Pengujian dan evaluasi sistem data akan dihitung dengan menggunakan Confusion Matrix. Pada confusion matrix terdapat dua pengukuran rasio pada setiap kelas yaitu positive rate adalah nilai yang diprediksi sebagai label good pada kedua label sedangkan negative rate adalah nilai yang diprediksi sebagai label bad dikedua label, yang kemudian dibagi lagi menjadi empat bagian nilai diantaranya adalah true positive rate (TPR), true negative rate (TNR), false positive rate (FPR) dan False negative rate (FNR) .Tabel confusion matrix berdasarkan positive rate dan negative rate dari hasil prediksi data. True positive rate (TPR) merupakan data label good yang terdeteksi benar dengan perhitungan membagi semua nilai hasil prediksi label good yang benar dengan semua label good

$$TPR = \frac{\text{jumlah prediksi good yang diprediksikan benar}}{\text{jumlah semua prediksi good}} \quad (6)$$

True negative rate merupakan data label bad yang terdeteksi salah sehingga diprediksi menjadi label bad, dengan perhitungan membagi semua nilai hasil prediksi label good yang salah diprediksi dengan semua label good

$$TNR = \frac{\text{jumlah prediksi good yang diprediksikan salah}}{\text{jumlah semua prediksi good}} \quad (7)$$

False positive rate merupakan data label bad yang terdeteksi benar, dengan perhitungan membagi semua nilai hasil prediksi label bad yang benar dengan semua label bad

$$FPR = \frac{\text{jumlah prediksi bad yang diprediksikan benar}}{\text{jumlah semua prediksi bad}} \quad (8)$$

False negative rate (TPR) merupakan data label bad yang terdeteksi benar dengan perhitungan membagi semua nilai hasil prediksi label bad yang salah diprediksi menjadi label good dengan semua label bad

$$FNR = \frac{\text{jumlah prediksi bad yang diprediksikan salah}}{\text{jumlah semua prediksi bad}} \quad (9)$$

Selain itu menggunakan confusion matrix akan didapatkan hasil berupa tingkat akurasi, presisi, dan recall. Hasil akurasi dari confusion matrix dapat menggunakan perhitungan persentase seperti rumus 4.1, hasil akurasi diperoleh dari pembagian jumlah prediksi yang tepat dibagi dengan jumlah seluruh data

$$\text{akurasi} = \frac{\text{jumlah prediksi yang tepat}}{\text{jumlah seluruh data}} \quad (10)$$

Hasil precision dari confusion matrix dapat menggunakan perhitungan persentase seperti rumus 4.2, hasil precision diperoleh dari perbandingan rasio prediksi tepat positif dengan keseluruhan hasil positif

$$\text{precision}_{\text{label}} = \frac{\text{jumlah data yang diprediksi dengan tepat}}{\text{jumlah data per label}} \quad (11)$$

Hasil recall dari confusion matrix dapat menggunakan perhitungan persentase seperti rumus 4.3, hasil recall diperoleh dari pembagian jumlah data yang benar true positive dibagi dengan jumlah data yang dipisahkan

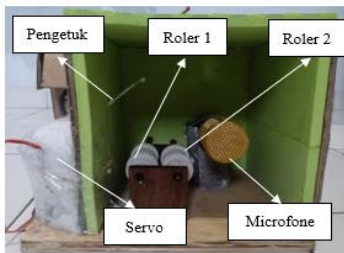
$$\text{recall}_{\text{label}} = \frac{\text{jumlah data yang diprediksi dengan tepat}}{\text{jumlah data prediksi per label}} \quad (12)$$

Dalam melakukan prediksi data menggunakan dataset yang telah dikumpulkan. Hasil percobaan dari proses klasifikasi telur dengan menggunakan dataset yang telah diambil dan direkam pada saat pengumpulan data

IV. HASIL DAN PEMBAHASAN

A. Implementasi Perancangan Alat

Alat yang digunakan berbentuk kubus dengan alas memiliki luas sekitar 20 cm x 20 cm dimana untuk bagian dalam terdapat mikrofon untuk mengambil suara telur puyuh dan penampang untuk meletakkan telur puyuh



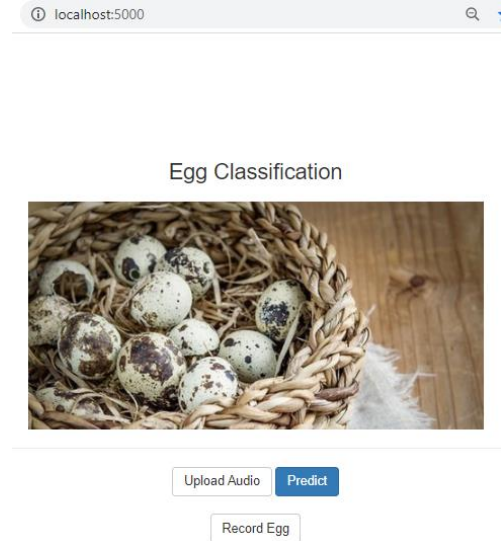
Gambar 9 Alat Perekam Telur

Pada gambar 9 adalah gambar tampak depan pada alat dimana terdapat pengetuk yang berfungsi untuk mengetuk telur, roler1 dan roler2 yang berfungsi untuk memutar telur dalam proses perekaman suara, selain itu terdapat servo yang akan menggerakkan pegetuk dalam proses perekaman suara dan terakhir terdapat mikrofone yang akan menangkap suara hasil ketukan telur.

B. Implementasi Rancangan Antarmuka

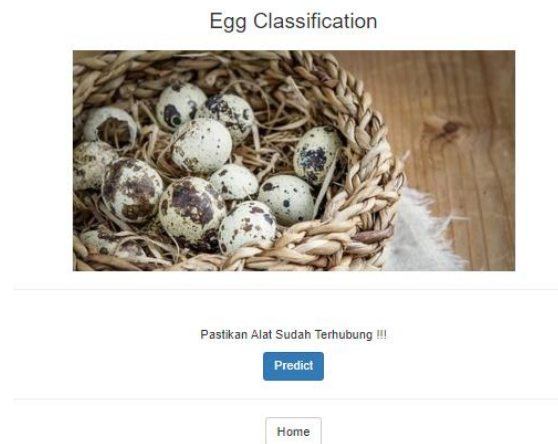
Pada tahap ini adalah tahap implementasi rancangan aplikasi yang digunakan untuk prediksi data dalam penelitian ini aplikasi ini berbasis website dan memiliki python sebagai pengelolah data atau backend dan HTML sebagai frontend, Dengan menggunakan python sebagai bahasa pemograman dalam pembuatan aplikasi akan memudahkan peneliti

menyambungkan aplikasi dengan mikroprosesor Raspberrypi yang nantinya akan sebagai komponen yang akan menggerakkan servo dan motor, dan mikrofon akan mengambil file suara yang akan diolah dalam aplikasi



Gambar 10 Tampilan Antarmuka Index.html

Tampilan Aplikasi pada gambar 10 adalah tampilan home dimana terdapat tombol *Upload Audio* berfungsi untuk mengambil data wav yang akan diuji dengan menggunakan model yang ada, dan tombol *Predict* berfungsi untuk melakukan prediksi pada *file wav* yang telah diupload dan hasilnya akan dipelihatkan.



Gambar 11 Tampilan Antarmuka Record Page

Antarmuka perekaman telur yang kemudian akan menjadi dataset baru yang akan diprediksi oleh user, user diharuskan untuk merekam suara terlebih dengan menekan tombol *predict* pada aplikasi, secara otomatis aplikasi akan menggerakkan alat yang telah dihubungkan terlebih dahulu, melakukan perekaman pada telur dan akhirnya melakukan prediksi tentang kualitas apa yang dimiliki oleh telur yang telah di prediksi



Gambar 12 Tampilan Antarmuka Predict

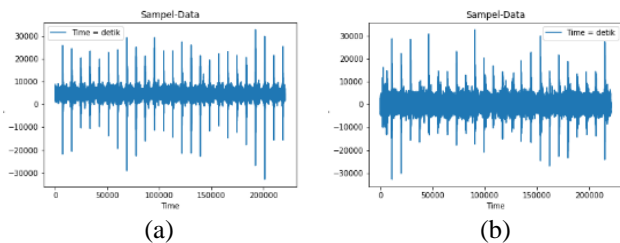
Pada bagian prediksi data user dapat memutar *audio* hasil perekaman telur dan melihat kategori telur hasil prediksi yang dilakukan dengan metode RNN

C. Implementasi Perekaman Suara

Perekaman suara meliputi proses pengambilan data yang dilakukan pada tanggal 12 September 2019 sampai tanggal 14 September 2019, dan dilakukan di Tomohon dan di Laboratorium Teknologi Basis Data Program Studi Teknik Informatika Fakultas Teknik Universitas Sam Ratulangi Manado dengan menggunakan alat yang telah dibuat

D. Implementasi Pelebelan Data

Pada implementasi pelebelan data adalah hasil proses pelabelan data yang dikategorikan menjadi dua label yaitu label *good* dan label *bad*

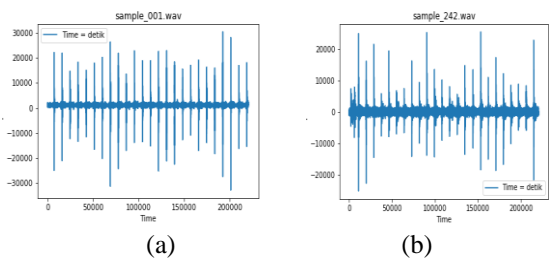


Gambar 13 Sampel Telur *Good* (a), dan Sampel Telur *Bad* (b)

Data yang sudah diambil dalam proses sebelumnya kemudian akan diberi label pada setiap data yang telah diambil berdasarkan kondisi cangkang telur yang dilihat pada proses pengambilan data. Pelabelan data mengkategorikan telur menjadi 2 kategori, pertama telur dengan kondisi cangkang masih utuh, tidak retak dan tidak terlihat cacat dikategorikan telur “*Good*” dan telur dengan kondisi cangkang rusak, pecah dan retak dikategorikan telur “*Bad*”.

E. Implementasi Reduksi *Noise*

Dari data yang telah di dapatkan dalam proses pengumpulan dan diberi label maka normalisasi pada data diperlukan untuk menghasilkan data yang bersih sehingga meningkatkan akurasi pada saat training data



Gambar 14 Reduksi Telur *Good* (a), dan Reduksi Telur *Bad* (b)

F. Implementasi Ekstraksi Fitur Data

Dalam mengelolah file suara atau audio sangat diperlukan proses ekstraksi fitur atau ekstraksi ciri dengan menggunakan fast fourier transform dan mel-frequency ceptral coefficient yang berguna untuk mengambil fitur-fitur yang terdapat dalam satu file audio(.wav) yang kemudian dari hasil ekstraksi tersebut nantinya akan digunakan untuk proses pengklasifikasian data suara dalam prosesnya akan mengambil beberapa fitur audio yang nantinya akan di simpan dalam format excel, fitur- fitur yang di ambil antara lain adalah fitur suara asli yang dibagi menjadi mean, median, max, min, range, standar deviation, median abs deviation, L1 norm, L2 norm dan maximum norm.

TABEL I
Ekstraksi Ciri Audio

<i>Fname</i>	<i>OriMean</i>	<i>OriMedian</i>	<i>OriMax</i>	<i>OriMin</i>	<i>OriRange</i>
<i>sample_001.wav</i>	0.13962	0.13907	0.89371	-0.58851	1.4822
<i>sample_002.wav</i>	0.02101	0.02065	0.94055	-0.90486	1.8454
<i>sample_003.wav</i>	0.03889	0.03870	0.86517	-0.77941	1.6445
<i>sample_004.wav</i>	0.08123	0.08143	0.84413	-1.02902	1.8731
<i>sample_005.wav</i>	-0.00181	-0.0017	0.86699	-0.97957	1.8465
<i>sample_006.wav</i>	0.06677	0.06694	0.89413	-0.86549	1.7596
<i>sample_007.wav</i>	0.07822	0.07849	0.89931	-0.89039	1.7897
<i>sample_008.wav</i>	-0.07026	-0.0698	0.96958	-0.98938	1.9589
<i>sample_009.wav</i>	-0.18639	-0.1845	0.93671	-0.98665	1.9233
<i>sample_010.wav</i>	-0.58815	-0.5874	0.29296	-0.98454	1.2775
<i>sample_011.wav</i>	0.08681	0.08671	1.00297	-0.98606	1.9890
<i>sample_012.wav</i>	0.0445	0.0449	0.84143	-0.75824	1.5996

Hasil dari ekstraksi fitur dari beberapa audio berlabel “*Good*” dan “*Bad*” fitur-fitur yang diekstraksi berupa static value of original data yang berisi nilai-nilai statistic mean, median, maximum, minimum value dan range. *OriMean* adalah nilai dari mean dari original signal, *OriMedian* yang adalah median dari original signal, *OriMax* yang adalah nilai dari maximum original value dan *OriRange* yang adalah jangkauan dari original signal.

G. Evaluasi Sistem

Evaluasi sistem data akan dihitung dengan menggunakan Confusion Matrix. Pada confusion matrixs terdapat dua pengukuran rasio pada setiap kelas yaitu positive rate adalah nilai yang diprediksi sebagai label good pada kedua label sedangkan negative rate adalah nilai yang diprediksi sebagai label bad dikedua label. Yang kemudian dibagi lagi menjadi empat bagian nilai diantaranya adalah true positive rate (TPR), true negative rate (TNR), false positive rate (FPR) dan False negative rate (FNR) .Tabel

confusion matrixs berdasarkan positive rate dan negative rate dari hasil prediksi data. True positive rate (TPR) merupakan data label good yang terdeteksi benar dengan perhitungan membagi semua nilai hasil prediksi label good yang benar dengan semua label good

Pengujian dari model yang didapat, dengan data baru dan diambil 70 dataset baru untuk dihitung hasil akurasi, precision dan recall dengan confusion matrixs.

TABEL II

Hasil *Confusion Matrix* untuk Pengujian Dataset

		Predict Class	
		X	Y
True Class	Good	27	8
	Bad	2	33

Pada table II confusion matrix maka didapatkan hasil prediksi dari 70 data pengujian, 60 data yang berhasil diprediksi dengan tepat dan 10 data sisanya salah diprediksikan. Untuk kategori good dari 35 data, 27 data berhasil diprediksi dengan tepat, sedangkan 8 data salah diprediksikan menjadi kategori bad. Sedangkan untuk kelas bad dari 33 data berhasil diprediksikan dengan benar dan sisanya 2 data salah diprediksikan menjadi good

$$akurasi = \frac{27 + 33}{70} \times 100\% = 85\% \quad (13)$$

Hasil akurasi dari percobaan dengan menggunakan rumus 4.5 maka didapatkan dengan hasil akurasi 85%, menunjukan model yang dibuat untuk memprediksi data menghasilkan nilai tingkat kedekatan antara nilai prediksi dengan nilai yang sebenarnya dengan prentase yang tinggi.

$$precision_{good} = \frac{27}{27 + 8} \times 100\% = 78\% \quad (14)$$

Hasil $precision_{good}$ dari percobaan dengan menggunakan rumus 4.2 maka didapatkan hasil $precision_{good}$ sebesar 78% yang menunjukan model yang dibuat untuk memprediksi data menghasilkan nilai ketepatan antara label good yang diminta oleh pengguna dengan jawaban label good yang diberikan oleh sistem mempunyai persentase ketepatan 78 %

$$precision_{bad} = \frac{33}{33 + 2} \times 100\% = 95\% \quad (15)$$

Hasil $precision_{bad}$ dari percobaan dengan menggunakan rumus 4.2 maka didapatkan persamaan 4.5 dengan hasil $precision_{bad}$ sebesar 95 % yang menunjukan model yang dibuat untuk memprediksi data menghasilkan nilai ketepatan antara label bad yang diminta oleh pengguna dengan jawaban label bad yang diberikan oleh sistem mempunyai persentase ketepatan 95 %

$$recall_{good} = \frac{33}{33 + 8} \times 100\% = 80\% \quad (16)$$

Hasil $recall_{good}$ dari percobaan tabel 4.9 dengan menggunakan rumus 4.3 maka didapatkan hasil $recall_{good}$ sebesar 80 % yang menunjukan model yang dibuat untuk memprediksi data menghasilkan nilai tingkat keberhasilan sistem dalam menemukan kembali sebuah label good sebesar 80%.

$$recall_{bad} = \frac{27}{27 + 2} \times 100\% = 93\% \quad (17)$$

Hasil $recall_{baddari}$ percobaan tabel 4.9 dengan menggunakan rumus 4.3 maka didapatkan hasil recall 93%. yang menunjukan model yang dibuat untuk memprediksi data menghasilkan nilai tingkat keberhasilan sistem dalam menemukan kembali sebuah label good sebesar 93%.

TABEL III

Hasil Keseluruhan

Label	Positive rate	Negative rate	Precision	Recall	Akurasi
Good	78 %	22 %	78 %	80 %	85 %
Bad	95 %	5 %	95 %	93 %	

Pada tabel 4.11 adalah tabel hasil keseluruhan pengujian, pada pengujian pertama yang dilakukan dengan mengambil data suara dari dataset sebanyak 70 sampel dengan 35 sampel telur good dan 35 sampel telur bad. Dari sampel telur pada pengujian maka dipadatkan hasil positive rate yang merupakan nilai ketepatan sistem dalam memprediksi data dengan benar pada setiap kelas sedangkan negative rate adalah nilai kesalahan sistem dalam memprediksi data per kelas yang dimana pada pengujian ini didapatkan hasil positive rate untuk kategori label good adalah sebesar 78% sedangkan kategori label bad adalah sebesar 95 %, untuk negative rate kategori label good adalah sebesar 22% sedangkan kategori label bad adalah sebesar 5 %. Perhitungan precision dimaksudkan mencari ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem sehingga dari pengujian yang dilakukan didapatkan hasil untuk kategori label good adalah sebesar 78% sedangkan kategori label bad adalah sebesar 95 %, selanjutnya recall dimaksudkan untuk melakukan perhitungan nilai tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi yang dipengujian ini memiliki nilai persentase untuk kategori label good adalah sebesar 80% sedangkan kategori label bad adalah sebesar 93 % dan nilai akurasi dimaksudkan untuk mencari nilai tingkat kedekatan antara nilai prediksi dengan nilai yang sebenarnya, yang dalam pengujian ini didapatkan hasil 85 %. Sehingga model dengan metode recurrent neural network yang dibuat pada proses training sudah dapat mengklasifikasikan telur puyuh yang mempunyai cangkang bagus dan cangkang retak melalui suara dengan sangat baik dengan menunjukan nilai precision diatas 78 %, recall diatas 85 % dan akurasi 85 %.

V. PENUTUP

A. Kesimpulan

Setelah melakukan pengumpulan data hingga menyelesaikan semua proses dalam penelitian ini maka dapat menarik kesimpulan sebagai berikut :

1. Ekstraksi fitur dengan menggunakan *Mel-frequency cepstral coefficients* dan ekstraksi fitur data statistic dari 252 data suara yang menghasilkan 8.316 data ekstraksi sudah mampu menghasilkan model yang baik dalam klasifikasi telur puyuh menggunakan *recurrent neural network*
2. Dengan Menggunakan *recurrent neural network*, klasifikasi telur puyuh berbasis pengolahan suara berhasil dilakukan dan memiliki hasil yang baik. Serta dapat mengklasifikasikan telur puyuh yang mempunyai cangkang bagus dan cangkang retak melalui suara, dan

menunjukkan nilai *precision* diatas 78 %, *recall* diatas 85% dan akurasi 85 %.

B. Saran

Dalam penelitian dari hasil pengumpulan, labeling, training, dan testing yang telah dilakukan masih ada hal – hal yang perlu dikaji agar dapat menjadi lebih baik. *Noise* perekaman suara dengan menggunakan alat masih sangat banyak dan belum tereduksi dengan baik, maka untuk penelitian selanjutnya diperlukan alat yang dapat meredam *noise* yang dihasilkan pada alat atau pada menemukan metode reduksi *noise* yang lebih baik sehingga proses klasifikasi boleh menghasilkan kualitas yang lebih baik

KUTIPAN

- [1] Sukma, Arie Widya, Antonius Hintono, and Bhakti Etza Setiani. "Perubahan Mutu Hedonik Telur Asin Sangrai Selama Penyimpanan." *Animal Agriculture Journal* 1.1 (2012): 585-598.
- [2] Devi Sahati. *Klasifikasi Telur Ayam Berbasis Pengelolaan Suara*. Diss. University of Sam ratulangi Manado, 2018.
- [3] RIZAL, Ahmad Ashril. 2017. *Prediksi Kunjungan Wisatawan Dengan Recurrent Neural Network Extended Kalman Filter*
- [4] Hermawan. 2014. *Aplikasi Model Recurent Neural Network dan Recurrent Neuro Fuzzy Untuk Peramalan Banyaknya Penumpang Kereta Api Jabotabek*
- [5] Ibnu. 2018 "Identifikasi Judul Berita *Clickbait* Berbahasa Indonesia Dengan Agloritma *Long Short Term Memory (LSTM) Reccurent Neural Network*."
- [6] Suyudi, Esmeralda, Muspupah, 2019. *Prediksi Harga Saham Menggunakan Metode Recurrent Neural Network*
- [7] FANSURI, Muhammad Ridwan. *Klasifikasi Genre Musik Menggunakan Learning Vector Quantization (LVQ)*. Institut Petanian Bogor, Bogor, 2011, 4.
- [8] Setiawan, Angga, Achmad Hidayatno, and R. Rizal Isnanto. *Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) untuk Mengoperasikan Cursor Komputer*. Diss. Diponegoro University, 2012.
- [9] Sariningsih, Ratna, and Indri Herdiman. "Mengembangkan kemampuan penalaran statistik dan berpikir kreatif matematis mahasiswa di Kota Cimahi melalui pendekatan open-ended." *Jurnal Riset Pendidikan Matematika* 4.2 (2017): 239-246.
- [10] Subagja, Muhammad Aldy. *Implementasi Recurrent Neural Network Pada Kasus Aplikasi Tanya Jawab*. Diss. Universitas Komputer Indonesia, 2019.

[11] Siti Soraya. "Multi Time Steps Prediction dengan Recurrent Neural Network Long Short Term Memory." *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer* 18.1 (2018): 115-124.

[12] Li, P., Wang, Q., Zhang, Q., dkk. 2012. 2012. Non-destructive Detection on the Egg Crack Based on Wavelet Transform. *IERI Procedia*, 2 (2012): 372-382.

[13] Li, P., Wang, Q., Zhang, Q., dkk. 2012. 2012. Non-destructive Detection on the Egg Crack Based on Wavelet Transform. *IERI Procedia*, 2 (2012): 372-382.

[14] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[15] Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999): 850-855.

TENTANG PENULIS



Exel Defrisco Tarkus Lahir di Palu, pada tanggal 8 Febuari 1998 dengan alamat tempat tinggal jalan trans sulawesi desa sumbersari, sulawesi tengah. Saya mulai menempuh Pendidikan SD di SD Negeri Sumbersari (2003-2009). Kemudian saya melanjutkan Pendidikan di tingkat pertama di SMP Negeri 1 Langowan (2009-2012).

Selanjutnya saya menempuh Pendidikan ke sekolah tingkat atas SMA Kristen 2 Binsus Tomohon (2012-2015). Setelah itu, di tahun 2015 saya melanjutkan Pendidikan S1 di Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sam Ratulangi Manado. Selama berada di bangku kuliah saya tergabung dalam organisasi kemahasiswaan intra kampus yaitu Himpunan Mahasiswa Elektro (HME), menjadi panitia di berbagai kegiatan himpunan, menjadi ketua HME-FT UNSRAT (2017-2018) unsrat, menjadi panitia dalam kegiatan *Bible Camp* UPK Kristen Fakultas Teknik UNSRAT, menjadi pengurus BEM-FT unsrat (2018-2019). Dan akhirnya, saya berhasil menyelesaikan studi di Program Studi Informatika UNSRAT