

# APLIKASI PENGHITUNG KOLONI BAKTERI BERBASIS ANDROID

Dicky Pradana<sup>1</sup>, Putra Prima Arhandi<sup>2</sup>, Annisa Taufika Firdausi<sup>3</sup>

<sup>1,2,3</sup>Jurusan Teknologi Informasi, Program Studi Teknik Informatika, Politeknik Negeri Malang

<sup>1</sup>dickypradana390@gmail.com, <sup>2</sup>putraprima@polinema.com, <sup>3</sup>annisa.taufika@polinema.ac.id

---

## ABSTRAK

Bakteri adalah cabang dari ilmu mikro-biologi yang selalu berkelompok (berkoloni) untuk melakukan perkembangan biakan dan bertahan hidup karena ketergantungan satu sama lain. Koloni bakteri sering di temukan di media-media sumber kehidupan seperti makanan, air, dan media lainnya, namun dengan adanya alat-alat penelitian yang terdapat di laboratorium menghambat proses penelitian, analisa dan perhitungan terhadap koloni bakteri bagi para peneliti atau praktisi mahasiswa yang sedang melakukan praktikum. Dengan demikian sangat penting untuk melakukan analisa bagi seorang peneliti dalam melakukan penelitian dengan mikro biologi yang akan di teliti. Untuk mengatasi masalah ini dikembangkan sebuah aplikasi penghitung koloni bakteri yang akan membantu para peneliti untuk melakukan analisa perhitungan jumlah koloni bakteri yang sedang di teleti. Sistem ini berbasis android dan menggunakan pengolahan citra digital dengan pemotretan citra dari koloni bakteri yang akan di hitung dan openCV sebagai alat bantu yang digunakan untuk mendukung proses pengolahan citra pada android.

**Kata kunci** : Android, *OpenCV*, Pengolahan Citra

---

## 1. Pendahuluan

Dalam ilmu biologi banyak cabang-cabang ilmu yang di kembangkan dan di teliti oleh para peneliti dalam bidang tersebut, mulai dari biologi murni, micro biologi, organisme, makhluk hidup lain, bio-kimia dan lain sebagainya. Ilmu mikro biologi juga banyak pembahasan dalam bidang medis, sering berhubungan dengan bakteri dan virus, bisa untuk pengobatan, pencegahan, dan penelitian terhadap makhluk kecil yang banyak merugikan dan sedikit menguntungkannya.

Bakteri adalah cabang dari ilmu mikro-biologi yang selalu berkelompok (berkoloni) untuk melakukan perkembangan biakan dan bertahan hidup karena ketergantungan satu sama lain. Koloni bakteri sering di temukan di media-media sumber kehidupan seperti makanan, air, dan media lainnya, namun dengan adanya alat-alat penelitian yang terdapat di laboratorium menghambat proses penelitian, analisa dan perhitungan terhadap koloni bakteri bagi para peneliti atau praktisi mahasiswa yang sedang melakukan praktikum. Pada tahun 2012, seorang bernama Quentien Geissmann telah melakukan penelitian dan menyelesaikan masalah diatas dengan mengembangkan sebuah aplikasi desktop dengan menggunakan perangkat *open source* yaitu OpenCFU media untuk menghitung koloni bakteri dengan beberapa metode dan sistem pendukung seperti *Threshold*, segmentasi citra, *Watershed*, dan OpenCV. Dengan begitu aplikasi dapat menghitung dengan kecepatan yang tepat dan keakurasian yang baik, sehingga membantu para peneliti dalam menghitung jumlah koloni bakteri yang sedang diteliti.

Namun dari jurnal terdahulu yang di kembangkan oleh Quentinn Giesmann dan Meet Poladia dkk masih menggunakan media desktop untuk proses analisa data yang di dapat dari sebuah citra, dan juga menggunakan media ponsel untuk ala bantu menangkap citra dan diolah kembali ke sebuah aplikasi *desktop*. Dengan begitu peneliti mencoba mengembangkan aplikasi ini kedalam bentuk mobile app atau android, dengan metode yang diterapkan di jurnal sebelumnya yaitu OpenCFU, a New Free and *Open-Source Software to Count Cell Colonie and Other Circular Objects* dengan tujuan dan hasil yang sama dan memudahkan laboratorium dalam perhitungan jumlah koloni yang masih menggunakan perhitungan manual, dan memanfaatkan perkembangan teknologi yang ada yaitu ponsel pintar yang sudah umum di miliki oleh beberapa kalangan.

## 2. Dasar Teori

### 2.1 Bakteri

Bakteri adalah salah satu golongan prokariotuk (tidak memiliki selubung inti). Bakteri sebagai makhluk hidup tentu memiliki informasi genetik berupa DNA, tapi tidak terlokalisasi dalam tempat khusus (nukleus) dan tidak ada membran inti. Bentuk DNA bakteri adalah sirkuler, panjang dan biasa disebut nukleoi, Jawetz, *et al* (2008).

Faktor-faktor yang mempengaruhi pertumbuhan bakteri adalah:

- Sumber energi, yang diperlukan untuk reaksi-reaksi sintesis yang membutuhkan energi dalam pertumbuhan dan restorasi, pemeliharaan keseimbangan cairan, gerak dan sebagainya.

- b. Sumber karbon.
- c. Sumber nitrogen, sebagian besar untuk sintesis protein dan asam-asam nukleat.
- d. Sumber garam-garam anorganik, khususnya folat dan sulfat sebagai anion, dan potasium, sodium magnesium, besi, mangan sebagai kation.
- e. Bakteri-bakteri tertentu membuahkan faktor-faktor tumbuh tambahan.

## 2.2 Koloni Bakteri

Koloni bakteri merupakan kumpulan bakteri sejenis, hasil reproduksi yang mengumpul pada satu tempat di medium kultur atau kumpulan bakteri yang berasal dari hasil pertumbuhan atau keturunan dari satu sel bakteri. Beberapa koloni bakteri menunjukkan ciri-ciri koloni yang saling berbeda, baik dilihat dari bentuknya, elevasi, maupun bentuk tepi koloni.

Memiliki sifat – sifat umum yaitu:

- a. Besar kecilnya koloni: berupa titik atau melebar sampai menutup permukaan medium.
- b. Bentuk : memanjang, tepi rata atau tidak merata.
- c. Kenaikan permukaan: rata dengan medium atau timbul menjulang dari permukaan medium.
- d. Hasil kasarnya permukaan: halus, kasar atau tidak rata.
- e. Wajah permukaan: mengkilat atau suram.
- f. Warna: keputihan atau kekuning-kuningan, merah muda, coklat, hijau, ungu atau biru.
- g. Kepekaan : lunak seperti lendir, seperti mentega, kering atau keras.

## 2.3 Threshold

*Thresholding* memainkan peranan penting dalam proses segmentasi citra dokumen. Tujuan *thresholding* ini adalah menemukan nilai *threshold* yang tepat untuk memisahkan objek dari *background*. Metode ini digunakan untuk menganalisis gambar skala abu-abu menjadi citra biner dan metode ini mengasumsikan bahwa objek yang berbeda dalam sebuah gambar dapat dibedakan dengan nilai abu-abu dan kunci untuk metode ini adalah bagaimana menentukan ambang batas, Anas, A. (2017).

*Thresholding* digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Maka derajat keabuan bisa diubah sesuai keinginan, proses *thresholding* ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan *thresholding* dengan derajat keabuan dapat digunakan rumus, Wijaya, M. C, *et al* (2009).

$$x = \frac{w}{b} \quad (1)$$

Perhatikan bahwa persamaan diatur rata Dimana :  
 w adalah nilai derajat keabuan sebelum *thresholding*  
 b adalah jumlah derajat keabuan yang diinginkan  
 x adalah nilai derajat keabuan setelah *thresholding*

## 2.4 Gray-scale

*Grayscale* adalah citra keabuan dengan nilai intensitas kedalaman pixel 8 bit. *Grayscale* biasanya dipakai untuk mempresentasikan insitas cahaya pada jalur tunggal dari spektrum elektromagnetik (inframerah, cahaya, ultraviolet, dsb). Beberapa metode *graysacling* yang sering dipakai adalah *Averaging*, *Lightness*, dan *Luminance*. Dimana *Averaing* ditentukan dengan merata-rata nilai dari tiap *channel* R (*Red*), G (*Green*), dan B (*Blue*). *Lightness* ditentukan dengan merata-rata nilai maksimum dan nilai minimum seluruh *channel* R, G, dan B. *Luminance* ditentukan dengan memberikan pembobotan yang disesuaikan dengan spektrum *visible* yang paling banyak pada citra alami.

$$Grayscale_{avg} = \frac{(R+B+G)}{3} \quad (2)$$

$$Grayscale_{Lightness} = \frac{\max[R,G,B]+\min[R,G,B]}{2} \quad (3)$$

$$Grayscale_{Luminance} = 0.21R + 0.72G + 0.07B(4)$$

## 2.5 OpenCV (Computer Vision)

*Open Source Computer Vision Library* (OpenCV) adalah library yang dikembangkan oleh pengembang *source code* di tahun 99' untuk menggabungkan pemrosesan gambar kedalam variasi dalam bahasa pemrograman dengan menerapkan metode *thresholding* dan segmentasi dapat memilah gambar piksel yang akan diolah utnuk mempermudah representais gambar tersebut.

Dalam *library* ini dapat di ambil dari berbagai platform seperti Windows, Linux, IOs, dan Android. Untuk memudahkan, menggunakan alat bantu seperti kamera untuk menangkap gambar. Namun pemotretan pertama memerlukan inialisasi pada gambar yang di tangkap dan pada pemrosesan selanjutnya bisa di munculkan *output* yang diinginkan dari gambar, Wijaya, M. C, *et al* (2009).

## 2.6 Canny Edge Detection

Pendeteksi tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tujuan dari pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Yang dimaksud tepi adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Metode Canny merupakan salah satu algoritma dekteksi tepi yang ditemukan oleh Marr dan Hildertth yang meneliti pemodelan persepsi visual manusia. Pendekatan algortima Canny dilakukan dengan kovolusi fungsi citra dengan *operator* Gaussian dan turunan-turunannya, Jawetz, *et al* (2008).

Ada beberapa kriteria pendeteksi tepian yang dapat dipenuhi oleh algoritma Canny:

- a. Mendeteksi dengan baik, kemampuan untuk meletakkan dan menandai semua tepi yang ada

sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan.

- b. Mengalokasi dengan baik, dengan Canny dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.
- c. Respon yang jelas, dengan Canny dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.

**2.7 Contour**

Kontur atau *contours* dapat diartikan sebagai kurva yang menggabungkan semua titik kontinyu (sepanjang batas), yang memiliki warna atau intensitas yang sama. Kontur berguna untuk analisis bentuk dan deteksi objek dan pengenalan. Fungsi yang di gunakan adalah `cv.findContours()`, `cv2.drawContours()`. Pada `cv.findContours()` menggunakan beberapa parameter sebagai inputan untuk mendapatkan hasil analisa suatu objek, beberapa parameter di peroleh dari metode-metode pengolahan seperti *threshold* melakukan perubahan warna pada gambar asli ke skala keabuan untuk membedakan antara objek dengna *background*, melakukan perubahan warna ke tingkat hitam pudah menggunakan *threshold* untuk memeperjelas objek dengan *bakcground* agar bisa di bedakan lagi, selanjutnya dilakukan deteksi tepi (*canny edge*) untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra dan menandai objek tersebut dengan radius inputan *threshold* yang akan memperjelas gambar tepian objek, membuat sumber gambar baru untuk di jadikan tolak ukur antra gambar olah citra dengan gambar asli yang mana akan memberikan inputan tersendiri sebagai objek yang sudah di pisahkan dari *background*.

Pada `cv2.drawContours()` melakukan pewarnaan atau pelabelan pada keluaran `cv.findContours()` yang sudah dilakukan pengolahan citra dan memeperjelas area yang telah di tandai dan di olah citra agar lebih detail dalam melakukan analisa objek yang di teliti.

**2.8 Mean Absolute Percentage Error**

*Mean Absolute Percentage Error* (MAPE) dihitung dengan menggunakan kesalahan absolut pada tiap periode dibagi dengan nilai observasi yang nyata untuk periode itu. Kemudian, merata-rata kesalahan persentase absolut tersebut. MAPE merupakan pengukuran kesalahan yang menghitung ukuran presentase penyimpangan antara data aktual dengan data peramalan, Pakaja (2012). Interpretasi dari nilai MAPE sebagai berikut, M. E. Ervina, *et al* (2018).

- a. < 10 % = peramalan sangat akurat.
- b. 10 % - 20 % = peramalan akurat.
- c. 20 % - 50 % = peramalan cukup akurat.
- d. > 50 % = peramalan tidak akurat.

Dalam menghitung MAPE menunjukkan akurasi peramalan dalam bentuk persentase dengan menggunakan *Percentage Error* (PE), pada perhitungan PE digunakan menentukan jumlah persentase *error* pada peramalan (The OpenCV Tutorials (2019)). Berikut rumus menghitung PE dan MAPE.

Rumus PE

$$PE = \left( \frac{X_t - F_t}{X_t} \right) \times 100 \tag{5}$$

Rumus MAPE

$$MAPE = \sum_{t=1}^n = 1 \frac{PE_t}{n} \tag{6}$$

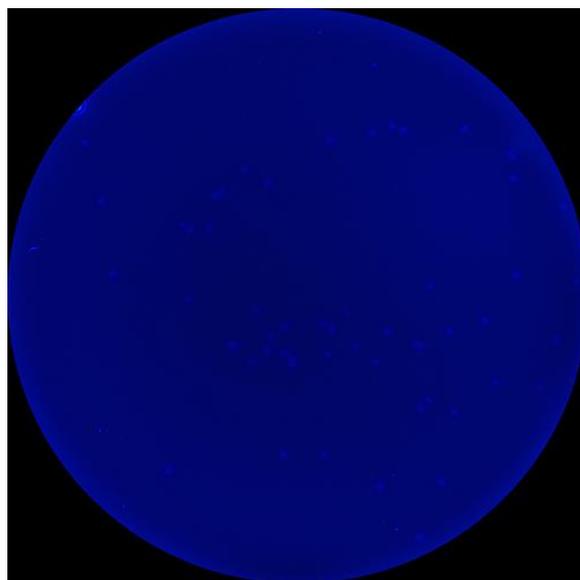
Dimana :

- n = jumlah data uji
- X<sub>t</sub> = nilai sebenarnya
- F<sub>t</sub> = nilai perkiraan

Hasil yang didapat setelah menghitung MAPE jika semakin rendah nilai MAPE, maka dapat dikatakan peramalan memiliki kemampuan yang baik dan sebaliknya, jika nilai MAPE menunjukkan nilai yang tinggi maka dapat dikatakan peramalan tidak akurat.

**3. Metodologi**

**3.1 Metode Penelitian Data**



Gambar 1. Koloni Bakteri

Metode pengambilan data yang peneliti gunakan adalah metode observasi. Observasi adalah teknik pengumpulan data yang tidak hanya mengukur sikap dari responden (wawancara dan angket) namun juga dapat digunakan untuk medapatkan berbagai fenomena yang terjadi. Teknik ini dapat digunakan apabila penelitian ditujukan untuk mempelajari perilaku manusia dan dilakukan pada responden yang tidak terlalu besar.

Pada tahap ini peneliti mengumpulkan data pada Mikro-Biologi Laboratorium di Jurusan Teknik Kimia Politeknik Negeri Malang dengan mewawancarai Ibu Yanti Maryanti dan meminta izin

kepada penanggung jawab laboratorium Ibu Rila Agustina untuk mendapatkan sampel data koloni bakteri. Dari hasil wawancara mendapatkan contoh sampel koloni bakteri sebagai berikut :

a. Konsep



Gambar 2. Alur Aplikasi

Pada Gambar 2 menjelaskan tentang pengambilan data pada cawan petri di potret menggunakan ponsel user dan akan secara otomatis langsung tampil di halaman awal aplikasi tersebut. Selanjutnya dilakukan proses pengolahan citra untuk mendapatkan hasil analisa dan hitung jumlah koloni bakteri.

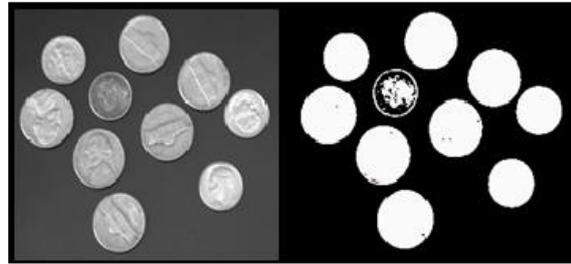
Adapun aturan-aturan yang harus dilakukan saat melakukan pengambilan foto sebagai berikut:

- a. Cawan petri harus berada di atas meja yang gelap atau meja bening dengan cahaya dari bawah untuk menjernihkan cawan petri dan memperjelas koloni bakteri.
- b. Intensitas cahaya pemotretan harus sedang tidak terlalu gelap dan tidak terlalu terang.
- c. Harus dengan pengawasan pihak laboratorium atau dosen penanggung jawab.

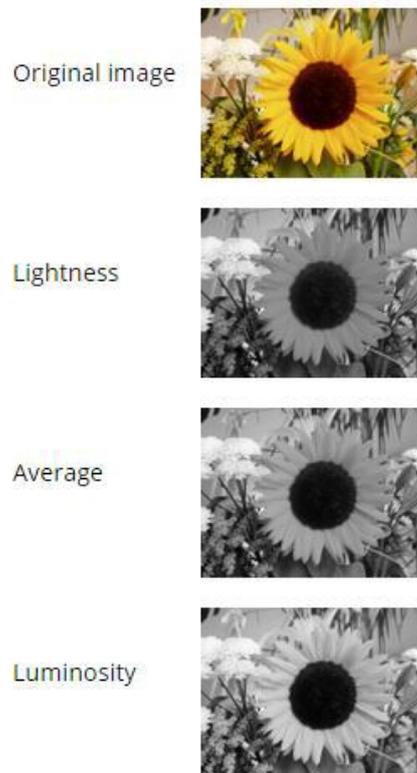
Tahap selanjutnya melakukan *processing* citra yang ditangkap oleh kamera ke dalam skala-keabuan untuk mendapatkan hasil yang berbeda dari objek yang tidak di inginkan.

$$Grayscale_{Lightness} = \frac{max[R + G + B] + min[R + G + B]}{2} \quad (7)$$

Selanjutnya nilai dari gray-scale atau skala-keabuan dimasukkan ke dalam proses *thresholding* dengan inputan nilai radius minimum dan maksimum yang kita inputkan untuk mencari keterkaitan dari objek-objek yang di *threshold* dan segmentasi. Pada pemrosesan *thresholding* dan segmentasi gambar melakukan pemrosesan dengan tingkat keabuan gambar. Yang terdiri dari tingkat warna piksel yaitu 0 dan 1 jika tingkat keabuan dihitung dan ditangkap sebagai 1-a maka objek tersebut dianggap sebagai background. Sebaliknya jika objek ditangkap sebagai nilai lebih dari 0 maka dianggap sebagai objek tertentu Langkah-langkah melakukan *threshold* adalah sebagai berikut



Gambar 3. Gray-scale dan Threshold



Gambar 4. Gray-Scale

Berikut langkah-langkah *Thresholding*:

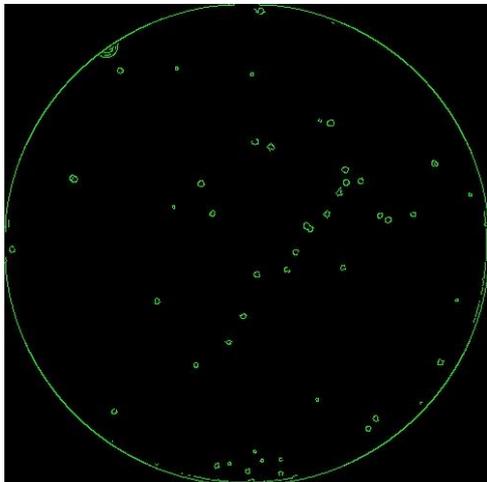
Langkah 1 : (1) *sortir* menurun seluruh nilai grade warna(w) dan bentuk (b) citra dalam basis data;(2) identifikasi citra dengan nilai *grade* warna dan bentuk tertinggi (urutan nomor1) dari daftar *grade* yang ada;(2) berhenti jika sudah mendapatkan sebanyak k citra dengan urutan nominasi.

Langkah 2: (1) identifikasi nilai grade ftur citra yang tidak lengkap (missing grade);(2) hitugn nilai representasi citra(A) dari daftar yang sudah ada dengan rumus:  $A_i = \min_{j \in \{0\}} [(w_i b_i)]$  dimana  $i = 1, 2, \dots, k$ ; (3) lakukan kembali terhadap citra selanjutnya dalam daftar k citra dengan nilai *grade* tertinggi.

Langkah 3: (1) tentukan nilai *threshold*(T) lebih kecil dibandingkan pada nilai representasi citra:  $T < A_k$ ; (2) ulang langkah 2; (3) berhenti jika sudah terdapat nilai representasi citra  $\geq$  nilai *threshold*.

Setelah mendapatkan nilai *threshold* yang di harapkan akan dilakukan proses pengolahan citra selanjutnya yaitu *contour* dengan data *threshold*

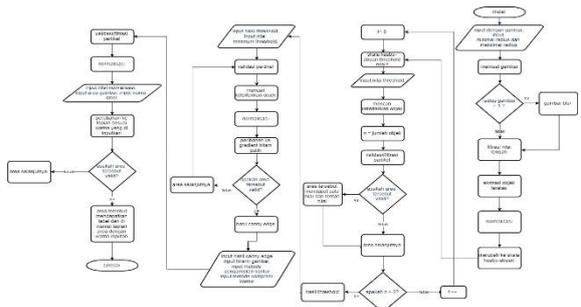
didapatkan perbedaan antara background dengan objek yang akan dihitung. Dengan melabeli objek dengan warna yang di inputkan maka objek yang akan dihitung akan terlabeli dan di analisa untuk perhitungan jumlah koloni.



Gambar 5. Proses Contour

b. Desain

Flowchart berikut menjelaskan tentang alur dari sistem aplikasi ini mulai dari kanan menuju ke kiri sebagai berikut. Alur pertama melakukan proses input *sample data* dengan pemotretan gambar atau gambar yang sudah ada di dalam galeri ponsel pengguna lalu dilakukan pengolahan citra pertama yaitu *gray-scale* dan mendapatkan hasil input olah citra, kedua dari nilai *gray-scale* dilakukan pengolahan citra selanjutnya yaitu *threshold* dengan inputan sebelumnya sudah memenuhi kriteria untuk melakukan proses *thresholding* dari tingkat keabuan menjadi hitam putih untuk pemisahan antara objek dengan *background* yang tidak diinginkan. Proses terakhir adalah pengirim data input *threshold* ke *server python* untuk dilakukan proses pengolahan citra kedua yaitu *contours* dimana dari hasil *threshold* dilakukan pelabelan terhadap objek yang sudah di pisahkan dengan *background* dan di warnai dengan inputan warna yang sudah di inputkan agar lebih mudah dipahami analisa oleh *user*. Dan terakhir dilakukan analisa dan perhitungan jumlah objek yang ada pada cawan petri tersebut.



Gambar 6. Flowchart Sistem

4. Hasil Pengujian

4.1 Pengujian Aplikasi

Pada bab ini menjelaskan hasil pengoprasian sistem. Berdasarkan hasil uji coba yang telah dilakukan akan dianalisa apakah perancangan ini dapat memahami tujuan yang telah dipaparkan sebelumnya pada Tabel 1.

Tabel 1. Pengujian Aplikasi

No	Skenario Pengujian	Hasil yang diharapkan	Kesimpulan
1	Membuka Aplikasi	Menampilkan <i>home</i>	Berhasil sesuai fungsi
2	Klik <i>icon</i> galeri	Menampilkan galeri dari ponsel	Berhasil sesuai fungsi
3	Klik <i>icon</i> kamera	Membuka kamera ponsel	Berhasil sesuai fungsi
4	Klik gambar yang dipilih	Kembali ke halaman awal dan menampilkan gambar yang dipilih	Berhasil sesuai fungsi
5	Klik <i>ok</i> pada hasil <i>capture</i> gambar	Kembali kehalaman awal dan menampilkan gambar hasil <i>capture</i>	Berhasil sesuai fungsi
6	Klik menu proses	Melakukan proses pengolahan citra dengan runtutan pengolahan <i>gray-scale, threshold, canny edge</i> dan <i>contour</i> untuk memperoleh hasil analisa dan hasil hitung jumlah	Berhasil sesuai fungsi

4.2 Pengujian Hasil

Pada sub bab ini menjelaskan tentang hasil akhir dari analisa dan perhitungan dari koloni bakteri menggunakan metode pengolahan citra yang di terapkan, dengan menggunakan metode perhitungan MAPE mendapatkan hasil tingkat akurasi perhitungan berupa prosentase tingkat keberhasilan aplikasi pada Tabel 2.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dalam penelitian kali ini memberikan informasi kepada para peneliti khususnya mahasiswa yang sedang melakukan penelitian terhadap koloni bakteri untuk melakukan analisa dan perhitungan secara akurat tanpa harus menggunakan alat yang mahal, *personal computer* atau menghitung secara manual untuk mendapatkan hasil perhitungan yang maksimal.

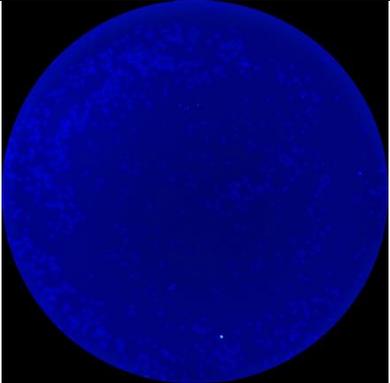
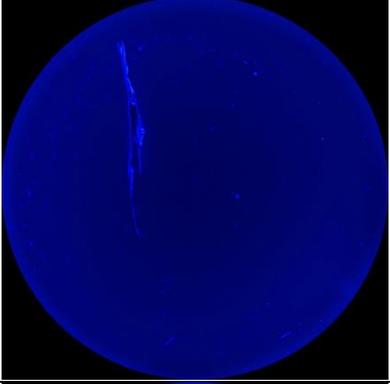
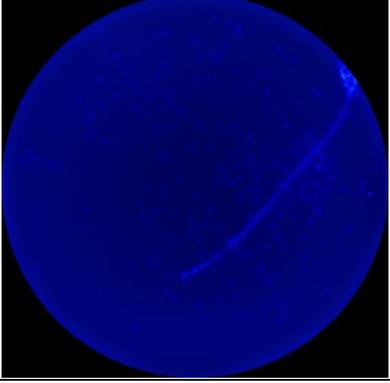
### 5.2 Saran

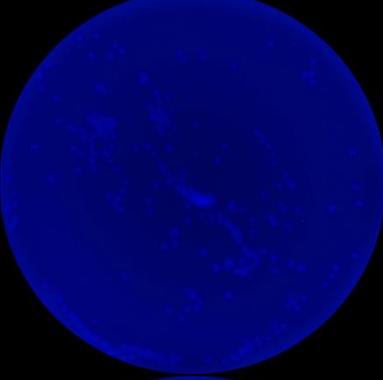
Demikian pembahasan dari penelitian yang saya paparkan, besar harapan saya dengan makalah ini bermanfaat bagi semua kalangan, karena keterbatasan pengetahuan dan referensi saya menyadari makalah ini jauh dari kata sempurna, oleh karena itu kritik dan saran yang membangun sangat diharapkan demi pengembangan dan penyusunan makalah menjadi lebih baik dimasa yang akan datang.

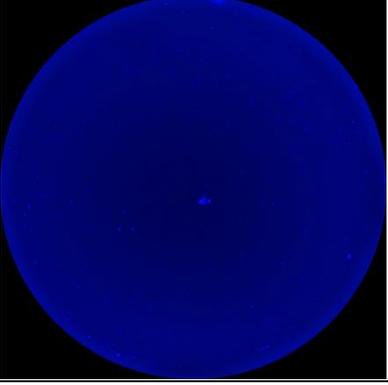
### Daftar Pustaka:

- Anas, A. (2017): *Penentuan Threshold Citra Mulut Dengan Metode Normal Probability Density Function (NPDF) Guna Mendeteksi Mulut Pemelajar*. ILKOM Jurnal Ilmiah, 9(2), 137-144.
- Anonim. (2019): *The OpenCV Tutorial*. Tersedia di : [https://docs.opencv.org/2.4/opencv\\_tutorial.pdf](https://docs.opencv.org/2.4/opencv_tutorial.pdf)
- Geissmann, Q. (2013): *OpenCFU, a new free and open-source software to count cell colonies and other circular objects*. PloS one, 8(2), e54072.
- Jawetz et al., (2008): *Medical Microbiology*. 24th ed. North America: Lange Medical boo.
- M. E. Ervina dan R.Silvi. (2018): *Peramalan Jumlah Penumpang Kereta Api Indonesia dengan Resilient Backpropagation Neural Network*, J. Mat "Mantik", vol. 4, no.02, pp.90-99.
- Pakaja. (2012): *Forecasting Single Exponential Smoothing Quantitative*. Jakarta Utara: Teknik Infomatika, Universitas Bunda Mulia.
- Poladia, M., Fakatkar, P., Hatture, S., Rathod, S. S., & Kuruwa, S. (2015, May): *Detection and analysis of waterborne bacterial colonies using image processing and smartphones*. In Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on (pp. 159-164). IEEE.
- Revan Asyari Sabir, Dian Novian, Abd. Aziz Bouty. (2015): *Penerapan Metode AHP dalam Menentukan Kawasan Peternakan*.
- Wijaya, M. C., & Tjiharjadi, S. (2009): *Mencari Nilai Threshold yang Tepat untuk Perancangan Pendeteksi Kanker Trofoblas*. In Seminar Nasional Aplikasi Teknologi Informasi (SNATI).

Tabel 2. Hasil Perhitungan MAPE

No	Data Sample	Metode Perhitungan	Hasil pengujian	Perhitungan Prosentase Error	Keterangan
1		Manual	830	2,8%	Kurang dari hitung manual
		Aplikasi	807		
2		Manual	143	1,4%	Kurang dari hitung manual
		Aplikasi	141		
3		Manual	75	117,3%	Lebih dari hitung manual
		Aplikasi	163		
4		Manual	299	0,7%	Lebih dari hitung manual
		Aplikasi	301		

No	Data Sample	Metode Perhitungan	Hasil pengujian	Perhitungan Prosentase Error	Keterangan
5		Manual	143	26,6%	Lebih dari hitung manual
		Aplikasi	181		
6		Manual	108	18,5%	Lebih dari hitung manual
		Aplikasi	128		
7		Manual	565	57,9%	Kurang dari hitung manual
		Aplikasi	238		
8		Manual	55	1,9%	lebih dari hitung manual
		Aplikasi	56		

9		Manual	68	36,8%	Lebih dari hitung manual
		Aplikasi	93		
10		Manual	178	34,9%	Lebih dari hitung manual
		Aplikasi	240		
Rata-rata Prosentase <i>Error</i>		26.39%			

