

Penentuan Jumlah Butir Gabah dengan Computer Vision

Tamrin¹, Junaidi Muhidong¹, dan Tahir Sapsal¹

Program Studi Teknik Pertanian, Universitas Hasanuddin Makassar

ABSTRAK

Perkiraan produksi padi, salah satunya dapat dilakukan dengan menghitung jumlah butir padi atau gabah yang dihasilkan. Metode perkiraan ini digunakan dalam berbagai penelitian yang berhubungan dengan peningkatan produksi padi, seperti pengaruh pemberian pupuk, perlakuan metode tanam dan lainnya. Menghitung jumlah butir padi tentu saja membutuhkan waktu, ketekunan dan ketelitian. Untuk itu diperlukan sebuah metode yang dapat digunakan untuk menghitung jumlah gabah yang dihasilkan dengan cepat dan akurat. Salah satunya dengan *Computer Vision*. Penelitian ini menggunakan *Raspberry Pi* tipe B yang dilengkapi dengan kamera 5 MP. Proses pengolahan citra meliputi penentuan *Region of Interest* (ROI) dan proses deteksi menggunakan *Contour Finding*. Tingkat akurasi dari hasil deteksi yaitu 99%. Posisi benih yang terdeteksi dengan tepat yaitu posisi benih yang terpisah dengan benih lain lebih dari 1 mm dan posisi benih yang berpeluang tidak terdeteksi dengan tepat yaitu posisi berhimpit dan bertumpuk serta tingkat kesalahan (*error rate*) pada proses pengujian yaitu pada benih berhimpit sebesar (0,23), pada benih bertumpuk sebesar (0,09) dan benih terpisah sebesar (0,01) serta tingkat kesalahan dalam proses perhitungan sebesar (0,03). Adapun ukuran gambar yang sesuai untuk proses pengolahan citra yaitu ukuran dengan *Resize* 75% (3,75 MP) dengan kecepatan deteksi 1.34 detik.

Kata kunci : Jumlah gabah, computer vision, contour finding

PENDAHULUAN

Latar Belakang

Jumlah gabah dapat digunakan untuk memprediksi produktivitas padi. Penelitian seperti pengaruh pemberian pupuk, perlakuan metode tanam dan hal lainnya, menjadikan produksi padi sebagai parameter pengamatan. Salah satu metode yang umum digunakan untuk memprediksi produktivitas padi adalah metode 4 faktor (BBPP Lembang, 2015), yaitu jarak tanam, jumlah anakan, jumlah butir permalai, dan berapa gram dalam 1000 butir gabah. Salah satu kendala dalam metode ini adalah dalam penentuan jumlah butir permalai, yang diambil dari jumlah gabah dalam satu rumpun dibagi dengan jumlah anakan yang memiliki malai. Jumlah gabah yang dihitung dapat mencapai ribuan. Tentu saja hal tersebut membutuhkan waktu, ketekunan, dan ketelitian. Sehingga diperlukan sebuah metode yang dapat digunakan untuk menghitung jumlah gabah yang dihasilkan dengan cepat dan akurat.

Metode yang dapat digunakan untuk menghitung jumlah gabah dengan menggunakan sensor visual. Sensor visual dapat digunakan untuk mendeteksi objek dengan menggunakan kamera. Untuk menghitung jumlah objek pada gambar yang tertangkap kamera dapat dilakukan dengan memanfaatkan *Computer Vision* (CV). CV adalah kemampuan mesin mengekstrak informasi dari sebuah gambar. CV dapat mengambil keputusan, melakukan aksi, dan mengenali suatu objek. Sehingga objek yang serupa bisa dikenali seperti kemampuan manusia dalam melihat dan mengenali objek serta mampu mengolah informasi dari gambar objek (Muljono, 2015).

Berdasarkan uraian di atas, maka perlu dikembangkan metode penghitungan jumlah gabah yang cepat dan akurat, dengan memanfaatkan *computer vision*. Sehingga dapat digunakan untuk mendukung penelitian-penelitian yang menggunakan parameter penghitungan jumlah gabah, dan juga untuk pengembangan alat deteksi objek.

Tujuan dan Kegunaan

Tujuan dari penelitian ini untuk membuat suatu program yang dapat dikembangkan untuk membuat alat yang mampu menentukan dan menghitung jumlah gabah dengan penerapan *Computer Vision*.

Adapun kegunaan penelitian ini sebagai tambahan informasi dalam pengembangan pendeteksian objek.

METODOLOGI PENELITIAN

Waktu dan Tempat

Penelitian ini dilaksanakan pada bulan Maret, hingga April 2017, di Laboratorium Alat dan Mesin Pertanian Program Studi Teknik Pertanian

Departemen Teknologi Pertanian, Fakultas Pertanian, Universitas Hasanuddin Tamalanrea Makassar.

Alat dan Bahan

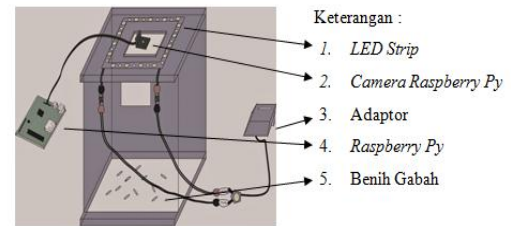
Alat yang digunakan dalam penelitian ini adalah Mini Komputer *Raspberry Pi Tipe B*, *Camera Module Raspberry Pi* dengan resolusi 5 MP, timbangan, mistar, *lux* meter, dan Bahasa pemrograman *Python 2.7* yang telah dilengkapi dengan *library Open CV* dan penyusunan program menggunakan *Sublime text* versi 3.

Bahan yang digunakan pada saat penelitian terdiri papan tripleks, *LED Strip* dengan arus 1,5 A, cat hitam, kertas HVS. Bahan untuk pengujian yaitu gabah jenis Cigeulis.

Prosedur Penelitian

1. Persiapan Alat

Sampel citra gabah diambil menggunakan alat yang dibuat agar mengurangi pengaruh perubahan cahaya lingkungan terhadap pengambilan citra. Alat ini berbentuk kotak tertutup, dengan dinding berwarna hitam, bagian dasar dilapisi kertas HVS berwarna putih dan dilengkapi dengan *led strip* sebagai sumber cahaya. Adapun luas permukaan kotak yaitu 18x18 cm .



Gambar 1. Kotak deteksi gambar gabah

2. Pengambilan Gambar Sampel Gabah

Pengambilan gambar sampel gabah bertujuan untuk mendapatkan contoh gambar gabah yang akan diolah. Adapun tahap dalam pengambilan gambar sampel gabah yaitu:

- Mengukur intensitas cahaya pada alat pengambilan gambar
- Memasukkan gabah dalam wadah pada alat yang disusun bertumpuk, berdekatan (jarak < 1mm) dan terpisah (jarak > 10 mm)
- Mengambil gambar gabah menggunakan modul kamera *Raspberry*

3. Penyusunan Program Penghitung Jumlah Gabah

Program deteksi gabah dikembangkan menggunakan Bahasa Python. Pengembangan dilakukan dengan editor berupa *Sublime text* Adapun tahap pembuatan program deteksi gambar gabah adalah sebagai berikut:

- Menginput gambar sampel gabah
- Melakukan proses *Resize*, yaitu untuk mengubah ukuran gambar yang ukurannya yang dibagi dalam 3 kelompok, yaitu 5% (0,25 MP), 50 %, (2.5 MP), 75% (3,75MP) dan 100% (tidak mengalami *resize*).
- Melakukan proses *Masking* untuk menutupi bagian gambar yang dapat mengganggu dalam proses pendeteksian kontur objek gabah pada gambar
- Melakukan proses *Thresholding*, proses ini akan membedakan gambar gabah dan *background* dengan mengubah gambar dalam warna abu-abu
- Melakukan proses dilasi dan erosi,

proses ini memiliki tujuan untuk membersihkan bintik-bintik pada gambar dengan menggunakan perintah erosi dan membuat gambar lebih solid dengan menggunakan perintah dilasi.

- f. Melakukan proses *Contour Finding* untuk mendeteksi bagian kontur dari gambar objek gabah yang ada pada gambar sampel yang sudah diolah dari proses sebelumnya.
- g. Mengulang poin c sampai f untuk ukuran gambar selanjutnya.
- h. Membandingkan hasil pendeteksian dan waktu eksekusi yang dibutuhkan dari 3 bentuk ukuran *resize*. Ukuran yang digunakan adalah yang mampu mendeteksi objek dengan tingkat kebenaran tertinggi dan waktu eksekusi tercepat.

Prosedur Pengujian

Aplikasi yang telah dibuat diuji dengan mengamati tingkat kebenaran pendeteksian gabah. Hal ini dilakukan dengan membandingkan hasil yang diperoleh oleh program yang telah dibuat dengan pengamatan langsung. Adapun rumus yang digunakan untuk menyatakan tingkat akurasi yaitu:

$$Akurasi = \frac{\text{beni h terdeteksi}}{\text{beni h aktual}} \times 100\% \dots \dots \dots (1)$$

Pengujian ini dilakukan dengan cara :

- a. Menjatuhkan 25 - 50 butir gabah ke dalam alat pengambil gambar secara acak
- b. Melakukan pengambilan gambar
- c. Melakukan penghitungan jumlah gabah dengan program yang telah dibuat
- d. Membandingkan dengan hasil pengamatan langsung
- e. Menghitung akurasi dengan menggunakan persamaan 1
- f. Mengulang poin a – e sebanyak 100 kali.
- g. Menghitung tingkat kesalahan (*error rate*)

$$Error\ rate = 100\% - Akurasi$$

Keterangan :

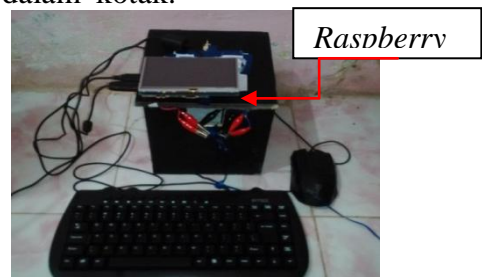
Error rate= Tingkat kesalahan (%)

Akurasi = Tingkat kebenaran hasil deteksi (%)

HASIL & PEMBAHASAN

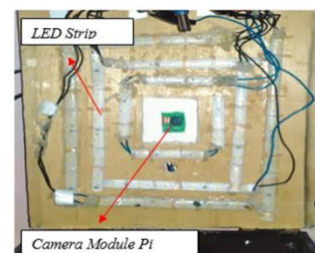
Pengambilan Gambar

Pengambilan gambar dilakukan dengan menggunakan *Camera Module Pi* dengan resolusi 5 MP yang terhubung dengan mini komputer *Raspberry Pi*. Untuk meningkatkan kualitas gambar, wadah dipasangkan dengan *LED strip* yang bertujuan untuk meningkatkan intensitas cahaya dalam kotak.



Gambar 2. Kotak Deteksi Gambar Gabah yang Terhubung dengan *Raspberry Pi*

Adapun intensitas cahaya dalam kotak yaitu sebesar 560 *lux*. Untuk meningkat tingkat akurasi pada saat mendeteksi, ukuran gambar diperkecil menjadi 75% (3,75 MP) karena pada ukuran tersebut dapat mengurangi rata-rata kesalahan pada saat mendeteksi dan menghitung gabah.



Gambar 3. LED Strip dan Camera Module Pi

Adapun tahap utama sebelum menentukan objek gabah yaitu tahap *Region Of Interest (ROI)*. Proses ini bertujuan untuk meningkatkan nilai piksel pada daerah gambar yang akan dilakukan proses deteksi. Selain itu proses *thresholding* pada tahap ROI bertujuan untuk

membedakan objek gabah yang akan dideteksi dengan latar gambar. Untuk mempermudah proses *thresholding* maka pada bagian alas permukaan kotak dipasangkan alas berwarna putih. Latar putih akan mudah dikonversi ke warna hitam pada proses *thresholding* dibandingkan warna lain, sehingga program akan lebih mudah membedakan antara latar dengan gambar objek yang berubah ke warna putih. Hal ini sesuai dengan dengan pernyataan Bradski dan Kaehler (2008), yang menyatakan hasil dari *threshold* adalah citra yang tampak nyata perbedaan intensitasnya, biasanya berwarna hitam dan putih.



Gambar 4. Bagian Alas Permukaan Pada Kotak Deteksi Gambar Gabah

Pada Gambar 8. pengujian, posisi gabah ditentukan dalam beberapa posisi yaitu posisi gabah yang terpisah lebih dari (>1mm), posisi gabah yang saling berhimpit dan posisi gabah yang saling bertumpuk. Adapun hasil yang didapat dari pengujian awal, sesuai dengan Gambar 9. gabah terdeteksi semua pada posisi yang terpisah, akan tetapi gabah yang berhimpit dan bertumpuk tidak terdeteksi dengan tepat karena gabah yang berhimpit akan terhitung satu sedangkan gabah yang bertumpuk hasil perhitungannya tidak menentu akibat program mendeteksi gabah bertumpuk dalam beberapa bagian. Adapun data hasil pengujian deteksi gabah berdasarkan posisi dari gabah tersebut yaitu :

Tabel 2. Hasil pengujian deteksi gabah berdasarkan posisinya

Posisi	Terdeteksi	Aktual
Berhimpit	110	148
Bertumpuk	17	32
Terpisah	2940	2942

Berdasarkan Tabel 4, bisa disimpulkan bahwa gabah yang dapat terdeteksi dengan tepat yaitu gabah yang posisinya terpisah dengan jarak lebih dari (>1mm). Maka dalam pengujian untuk mengurangi terjadinya kesalahan deteksi dan perhitungan jumlah gabah maka posisi gabah diusahakan tidak berhimpit dan bertumpuk.



Gambar 5. Hasil Deteksi Gabah

Perintah *Contour Finding*

Perintah *Contour Finding* merupakan perintah yang digunakan untuk mendeteksi objek gabah. Adapun cara kerja *Contour Finding* mendeteksi area kontur pada gambar objek yang ada pada gambar dengan cara menghitung luasan area dan perimeter kontur dari objek gabah. Hal ini sesuai dengan pernyataan Bradski dan Kaehler (2008), yang menyatakan bahwa fungsi *cv Contours Finding* untuk menghitung kontur dari gambar biner. Gambar kontur bisa didapat dengan dibuat menggunakan perintah (*cv Canny*), yang mana mempunyai kemampuan untuk mendeteksi sudut piksel dari kontur atau gambar. Adapun ukuran area dan parameter yang dideteksi sebagai gambar gabah yaitu dengan perimeter lebih dari atau sama dengan ≥ 50 dan area kurang dari < 500 , nilai yang didapatkan merupakan hasil dari metode *Trial Error* yang dilakukan beberapa kali untuk menemukan nilai yang tepat untuk mendeteksi kontur area objek gabah. Adapun Algoritma perintah *Contours Finding* adalah sebagai berikut :



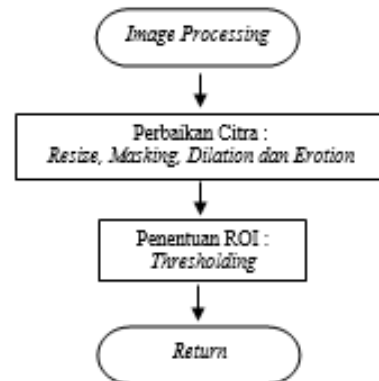
Gambar 6. Diagram Alir Contour Finding

Berdasarkan hasil pengujian adapun parameter dan area yang digunakan dalam menghitung jumlah gabah yaitu dengan perimeter >50 pixel dan area <500 pixel. Perimeter menunjukkan ukuran keliling piksel pada suatu objek dalam gambar sedangkan area merupakan nilai luasan kontur pada objek. Rata-rata nilai perimeter gabah berkisar antara 50 sampai 250 dengan luasan kontur berkisar 50-350. Nilai ini didapat dengan mengamati nilai dari perimeter dan area yang muncul dari 100 kali pengujian. Berdasarkan perimeter dan area kontur, jumlah gabah yang berhimpit akan bisa diketahui dengan melihat nilai yang didapat oleh program. Nilai gabah yang berhimpit akan memiliki nilai dua kali lebih besar dibanding dengan rata-rata nilai gabah yang terdeteksi satu. Sedangkan gabah yang bertumpuk nilai perimeter dan areanya tidak terlalu berbeda dengan nilai gabah yang tidak bertumpuk dan berhimpit akan tetapi akan mempengaruhi jumlah perhitungan gabah.

Image Processing

Image processing terdiri dari rangkaian proses yaitu perbaikan citra, penentuan ROI (*Region of Interest*). Proses ini bertujuan untuk memproses gambar sesuai dengan kriteria yang diinginkan sebelum memasuki tahap pendeteksian. Pada tahap *image processing* yang ditulis dalam *subroutine* untuk memudahkan perintah yang digunakan secara berulang. Adapun

algoritma *subroutine* dapat digambarkan sebagai berikut:



Gambar 7. Diagram Alir Subroutine

Perbaikan Citra

Perbaikan citra bertujuan untuk memperbaiki gambar citra sebelum citra diproses ketahap selanjutnya. Perbaikan citra berguna untuk mengurangi gangguan pada gambar seperti objek kecil yang akan mengganggu dalam proses selanjutnya. Adapun proses perbaikan citra yaitu:

1. Resize

Tujuan *Resize* yaitu untuk melihat ukuran yang tepat dalam mendeteksi objek gabah pada gambar sehingga hasil deteksi lebih presisi dan lebih cepat. Adapun data hasil pengujian ukuran gambar dengan kecepatan deteksi gabah adalah sebagai berikut:

Tabel 3. Hasil pengujian ukuran gambar dengan kecepatan deteksi gabah

Resize Akurasi (%)	Waktu Deteksi (detik)
100% (5 MP)	1.45
98%	
75% (3.75 MP)	1.34
99%	
50% (2.5 MP)	1.14
45%	
5% (0.5 MP)	1.12
0	

Berdasarkan Tabel 3. Adapun hasil pengujian dari beberapa perubahan ukuran dan nilai piksel, ukuran gambar dan piksel yang tepat pada saat

mendeteksi objek gabah yaitu pada ukuran 75% (3.75 MP).

Ukuran tersebut dapat mendeteksi gabah dengan kecepatan deteksi 1,34 detik dan peluang gabah terdeteksi dengan tepat lebih besar dibandingkan dengan ukuran piksel dan gambar 5% (0,5MP), dan 50% (2,5MP). Pada ukuran 100 % (5MP) atau tanpa perubahan, hasil deteksi gabahnya sudah tepat namun proses deteksi cukup lama dibandingkan dengan ukuran gambar 75% (3.75 MP). Pada ukuran 50% (2,5MP) kecepatan deteksinya cukup cepat yaitu 1.14 detik, namun pada ukuran tersebut masih banyak gabah yang tidak terdeteksi. Berdasarkan hasil pengujian, ukuran 5% (0,5MP), kecepatan deteksi gabah sangat cepat namun gabah yang terdeteksi tidak ada, hal ini disebabkan oleh ukuran gambar yang terlalu kecil.

2. *Masking*

Pada perintah ini objek yang akan mengganggu proses deteksi dan tidak mampu ditangani dengan proses *erotion* akan ditutupi dengan metode *Rectangular Mask*.

Metode ini menutupi bagian di luar dari kotak putih dan menggelapkan bagian yang tidak ingin di deteksi. Pada tahap ini obyek yang ditutupi adalah bagian gambar yang akan digunakan dalam proses pendeteksian seperti sudut tepi gambar.

3. Tahap *Dilation* dan *Erotation*

Pada tahap ini dilakukan dua proses yang bertujuan untuk membersihkan gambar dari objek berukuran kecil yang berada didalam gambar yang akan mengganggu dalam proses deteksi dan membuat gambar objek gabah lebih solid.

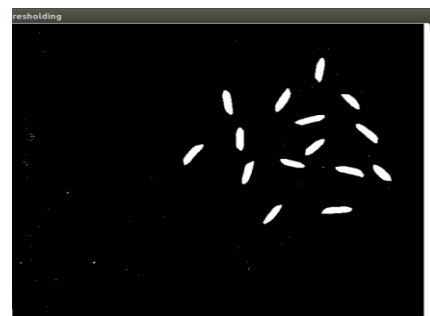
Pada proses *erotion* gambar dibersihkan dengan mengurangi area objek pada gambar dengan mengikis area yang menonjol, sehingga penampakan gambar akan lebih bersih. Kemudian dilakukan proses *dilation*. Pada proses

ini area diperluas dengan cara mengambil nilai tertinggi pada piksel dan memperluas bagian objek dan memperjelas objek yang cekung sehingga gambar tampak lebih jelas dan solid. Hal ini sesuai dengan pernyataan Bradski dan Kaehler (2008), yang menyatakan bahwa secara umum dilasi memperluas area dan erosi mengurangi area bahkan dilasi akan cenderung menghaluskan kecekungan dan erosi akan cenderung menghaluskan bagian yang menonjol.

Penentuan ROI (*Region of Interest*)

Proses *thresholding* bertujuan untuk mengubah gambar kedalam tampilan hitam putih, hal ini bertujuan untuk membedakan objek dengan *background* gambar.

Pada gambar 10 bisa dilihat objek gabah dapat dibedakan dengan *background*. Fungsi *thresholding* dalam penentuan *Region of Interest* yaitu untuk menentukan nilai piksel objek yang akan digunakan dari perbedaan nilai piksel dari objek dan *background*. Hal ini sesuai dengan pernyataan Bradski dan Kaehler (2008), yang menyatakan *thresholding* adalah metode di mana penentuan suatu nilai, kemudian hanya mengambil nilai di atasnya saja atau di bawahnya saja, sedangkan nilai selainnya diabaikan. Hasil dari *threshold* adalah citra yang tampak nyata perbedaan intensitasnya, biasanya hitam dan putih. Metode ini biasanya digunakan untuk segmentasi atau pemisahan suatu objek dengan objek lainnya.



Gambar 12. Tahap *Thresholding*

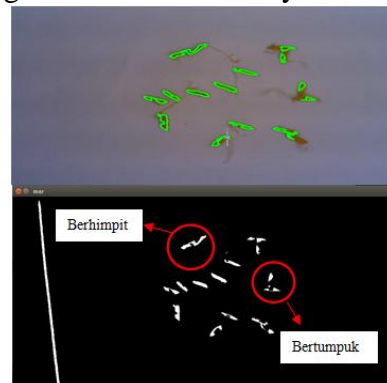
Akurasi Program Deteksi Gabah

1. Hasil Deteksi

Berdasarkan hasil pengolahan data dari pengujian program dengan cara mengambil gambar dari 100 sampel gabah, maka didapatkan tingkat akurasi dari deteksi program yaitu 99% dengan kecepatan deteksi yaitu 1.34 detik. Nilai ini dihasilkan dari menghitung jumlah gabah yang berhasil terdeteksi dari 100 sampel dan dibagi dari total keseluruhan gabah aktual dan kecepatan diperoleh dengan mengukur proses perhitungan jumlah gabah dengan *stopwatch*. Adapun 1 % merupakan tingkat kesalahan dari deteksi.

Data dari 100 kali pengujian menunjukkan bahwa posisi gabah yang dapat terdeteksi dengan tepat menggunakan *finding contour* yaitu posisi gabah yang terpisah dengan gabah lain lebih dari 1 mm, adapun posisi gabah yang berpeluang tidak terdeteksi dengan benar yaitu gabah yang berhimpit dan bertumpuk. Penyebab utamanya yaitu karena gabah yang berhimpit akan terbaca sebagai suatu objek yang sama sehingga objek akan dihitung sebagai satu objek atau keduanya terdeteksi. Sedangkan pada objek gabah yang bertumpuk akan lebih berpeluang terdeteksi sebagai dua objek karena adanya pengaruh efek bayangan akibat bertumpuk, akan tetapi proses pembacaan dalam program masih kurang tepat karena dalam tampilan *thresholding* gabah akan tampak terpisah beberapa bagian dan perintah *contour finding* akan menghitung nilai area dan perimeter kontur dari tiap bagian gabah. Hal ini sesuai dengan pernyataan Mulyana (2013), bahwa pada proses yang lebih khusus lagi, program *image analysis* digunakan untuk mencari tepian batas-batas objek dalam *image*. Sebuah tepian *edge* terbentuk antara objek dan latar belakangnya atau antara dua objek yang spesifik. Tepi ini akan terdeteksi sebagai akibat dari perbedaan level

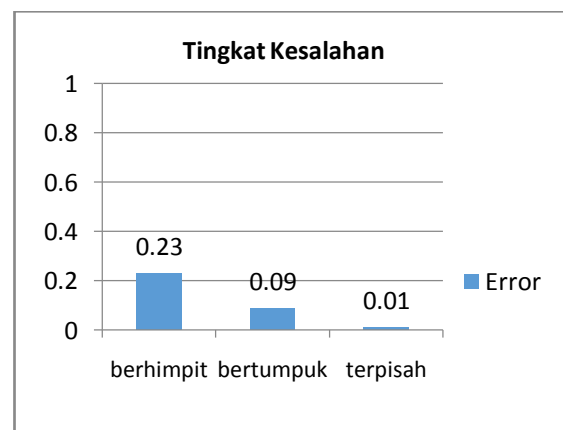
kecerahan pada sisi yang berbeda dengan salah satu batasnya.



Gambar 13. Hasil Deteksi dan Tampilan *Thresholding* Gabah Bertumpuk

2. Tingkat Kesalahan (*Error Rate*)

Data tingkat kesalahan deteksi didapatkan dari rata-rata 100 kali pengujian dengan menghitung jumlah gabah yang terdeteksi secara aktual pada posisi berhimpit, posisi bertumpuk dan posisi berpisah dan dibandingkan dengan hasil deteksi pada program. Adapun tingkat kesalahan hasil deteksi gabah berdasarkan posisinya dapat dilihat pada Gambar 14.



Gambar 14. Grafik Tingkat Kesalahan Deteksi Objek Gabah

Adapun hasil yang didapat pada Gambar. 1 yaitu tingkat kesalahan berhimpit (0,23), bertumpuk (0,09) dan kesalahan pembacaan gabah yang saling terpisah yaitu 0,01 dan nilai rata-rata tingkat kesalahan dari perhitungan yaitu 0.03 %, nilai rata-rata tingkat kesalahan dihasilkan dari selisih antara nilai dari data gabah perhitungan secara aktual

dengan nilai perhitungan program yang dirata-ratakan.

Pada proses pengambilan data, gabah yang berhimpit didapat tingkat kesalahan (*error rate*) paling tinggi yaitu (0,23). Hal ini disebabkan pada waktu dijatuhkan gabah sebagian akan saling berhimpit, posisi ini bisa terjadi sewaktu jumlah gabah yang dijatuhkan terlalu banyak sehingga peluang saling berhimpit akan lebih besar. Gabah saling berhimpit akan terdeteksi menjadi satu objek sehingga akan terjadi kesalahan perhitungan pada program atau program tidak mendeteksi sebagai suatu objek gabah. Sedangkan pada gabah yang bertumpuk, memiliki tingkat kesalahan (*error rate*) yaitu 0,09. Hal ini disebabkan karena peluang untuk mendapatkan gabah yang bertumpuk dalam pengujian cukup kecil.

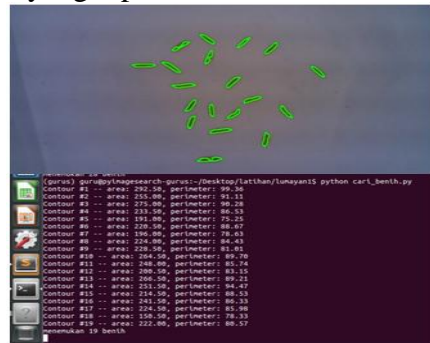
Gabah yang saling bertumpuk akan terdeteksi tetapi jumlah perhitungannya terkadang masih kurang tepat dan pada gabah yang saling terpisah tingkat kesalahannya yaitu (0,01) hal ini bisa terjadi karena posisi gabah berada pada daerah yang pencahayaannya kurang sehingga objek gabah akan tampak kurang jelas sehingga tidak akan terdeteksi oleh program.

Perbaikan Hasil Deteksi

Berdasarkan hasil yang diperoleh dapat disimpulkan bahwa penyebab terjadinya kesalahan deteksi yaitu posisi gabah yang saling berhimpit dan saling bertumpuk. Kesalahan yang dihasilkan dari posisi gabah yang terpisah disebabkan karena kurangnya pencahayaan. Untuk memperbaiki masalah pencahayaan maka pada saat pengambilan gambar dengan menggunakan kamera dengan intensitas cahaya pada ruangan minimal 560 lux. Dengan mempertahankan nilai intensitas cahaya 560 lux gambar gabah yang akan dihasilkan akan sesuai dengan sampel gambar yang dapat dideteksi oleh program deteksi gabah. Untuk memperbaiki hasil

deteksi dilakukan proses pendeteksian dengan cara posisi gabah harus terpisah lebih dari >1 mm. Gabah yang terpisah akan terdeteksi dengan tepat dan jumlah perhitungan gabahnya akan tepat karena sesuai dengan ukuran luas dan keliling yang telah ditentukan.

Berdasarkan Gambar 15, gabah yang berpisah akan terdeteksi dengan tepat karena objek dengan *background* akan jelas perbedaannya serta sudut tepi pada gabah terpisah akan lebih jelas dibandingkan dengan gabah yang berhimpit ataupun gabah yang bertumpuk. Adapun hasil dari perhitungan jumlah gabah akan terhitung dengan tepat hal ini disebabkan karena nilai perimeter dan nilai area dari gambar gabah akan sama sehingga akan terhitung dengan jumlah yang tepat.



Gambar 15. Hasil deteksi setelah gabah yang terpisah lebih dari >1mm

PENUTUP

Kesimpulan

Adapun kesimpulan yang dapat diperoleh setelah melakukan penelitian ini yaitu :

1. Jumlah gabah dapat dihitung dengan *Computer Vision* yang memberikan hasil yang akurat dengan tingkat akurasi 99% dan kecepatan deteksi 1,34 detik.
2. Posisi gabah yang terdeteksi dengan tepat yaitu gabah yang terpisah lebih dari 1 mm, adapun posisi gabah yang berpeluang tidak terdeteksi dengan benar yaitu gabah yang berhimpit dan bertumpuk dengan tingkat kesalahan masing-masing (0,23) berhimpit dan (0,9) bertumpuk.

Saran

Untuk pengembangan lebih lanjut sebaiknya dilakukan proses klasifikasi nilai area dan perimeter pada perintah *Contour Finding* untuk menentukan nilai area dan perimeter yang tepat untuk objek gabah yang berhimpit dua atau lebih.

DAFTAR PUSTAKA

- Adhi, M. Z. K. 2011. *Pengembangan Metode Penghitungan Benih ikan Lele dengan Pengolahan Citra dan Metode Penimbangan Gabah*. Departemen Teknik Mesin dan Biosistem Fakultas Teknologi Pertanian Institut Pertanian Bogor. Bogor. Jawa barat
- Bradski., G, Kaehler, A. 2008. *Learning Open CV*. O'Reilly Media. Amerika Serikat.
- BBPP Lembang. 2015. Menghitung Produksi Padi. <http://www.bbpp-lembang.info/index.php/arsip/artikel/artikel-pertanian/867-menghitung-produksi-padi>. Diakses 4 April 2017.
- Christianti R.F, Puspa M.N, Kurnianto D. 2015. *Aplikasi Raspberry Pi pada Telerobot Pembersih Lantai*. Program Studi Teknik Telekomunikasi, Sekolah Tinggi Teknologi Telematika. Purwokerto.
- Hadi S, Yoesha R.S, 2011. *Deteksi Objek Kendaraan Pada Citra Dijital Jalan Raya Menggunakan Metode Visi Komputer*. Jurusan Matematika Bidang Minat Ilmu Komputer Fakultas MIPA Universitas Padjadjaran. Jatinangor.
- Kurniawan, W. 2015. *Dimmer Light Berbasis Web Menggunakan Mini Pc Raspberry Pi*. Fakultas Ilmu Komputer, Jurusan Teknik Informatika. Universitas Dian Nuswantoro. Semarang.
- Mulyana, Kurniawan, dan Ardy. 2013. *Perancangan Sistem Kendali Pada Robot Tanam Gabah Langsung (Tabela)*. Jurusan Teknik Komputer Unikom. Bandung.
- Prabowo, I. 2013. *Penggunaan Raspberry Pi Sebagai Web Server Pada Rumah Untuk Sistem Pengendali Lampu Jarak Jauh dan Pemantauan Suhu*. Program Studi Sistem Komputer Fakultas Teknik Elektronika dan Komputer Universitas Kristen Satya Wacana. Salatiga.
- Purwanto, P. Dirgantoro, B., & Jati, A. N. 2015. *Implementasi Face Identification Dan Face Recognition Pada Kamera Pengawas Sebagai Pendeteksi Bahaya*. Universitas Telkom. Jakarta.
- Rabe, U. 2013. *Analisis Karakteristik Fisik Padi (Oryza sativa) Ciliwung dan Ciharang*. Program Studi Keteknikan Pertanian, Jurusan Teknologi Pertanian, Fakultas Pertanian, Universitas Hasanuddin Makassar.
- Raspberry. org. 2015. *Camera Module Raspberry Pi*. <https://www.raspberrypi.org/products/camera-module/>. Diakses 1 Maret 2017.
- Raspberry. org. 2015. *Raspberry Pi Model B*. <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. Diakses 1 Maret 2017.
- Robbani, H.R. 2016. *Aplikasi Mobile Scotect: Aplikasi Deteksi Warna Tanah Dengan Teknologi Citra Digital Pada Android*. Program Studi Informatika. Universitas Brawijaya. Malang.
- Santoso, B. 2013. *Bahasa Pemrograman Python di Platform Gnu/Linux*. Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Komunikasi. Universitas Multimedia Nusantara. Gading Serpong, Tangerang