

Rancangan Binarisasi Citra dan Pengenalan Karakter Teks Dengan Raspberry Pi

Ida Bagus Putu Widja

Program Studi Sistem Komputer STMIK STIKOM Bali
Jalan Raya Puputan No.86 Renon Denpasar, (0361) 244445
e-mail: ibpwidja@gmail.com

Abstrak

Penelitian ini bertujuan untuk mengkaji dan merancang sistem yang dapat melakukan konversi citra teks yang dihasilkan dari foto kamera Raspberry Pi menjadi karakter teks. Pengenalan karakter umumnya dilakukan pada citra dokumen hasil pemindaian oleh alat pindai karena hasilnya lebih baik dan lebih murah dibandingkan dengan menggunakan foto kamera. Akan tetapi dewasa ini banyak diproduksi kamera beresolusi tinggi berharga murah dengan kualitas yang sangat baik. Untuk itu peneliti mengusulkan menggunakan modul kamera Raspberry Pi (5 Megapixels) untuk menangkap citra dokumen. Jarak kamera dengan dokumen menjadi penentu awal agar memperoleh citra yang jelas dengan fokus yang seragam. Komputer berukuran kecil Raspberry Pi (CPU ARMv8 quad-core 1.2GHz) akan memproses citra awal menjadi citra biner (binarisasi) untuk memisahkan bagian konten teks dengan latarnya, memakai program python dengan library opencv. Pencahayaan yang tidak seragam juga akan menurunkan kualitas proses pengenalan karakter (Tesseract-OCR), sehingga diterapkan filterisasi homomorphic untuk menanggulangi masalah tersebut. Hasil uji coba protipe ini menunjukkan bahwa rancangan dapat bekerja sesuai harapan.

Kata kunci: Raspberry Pi, Python, Opencv, Tesseract-ocr

1. Pendahuluan

Tulisan analog atau teks yang tercetak pada dokumen merupakan obyek media yang akan ditangkap sekaligus direkam ke dalam bentuk citra digital melalui kamera atau dengan mesin pemindai. Citra digital hasil tangkapan kamera tersebut akan mengalami proses binarisasi sedemikian rupa agar aplikasi OCR (*Optical Character Recognition*) berhasil mengenali dengan baik konten teks tersebut untuk dirubah menjadi karakter digital.

Konversi citra ke teks dengan *Raspberry Pi* sebelumnya pernah dilakukan dengan citra dokumen yang relatif kecil, yang hanya mengandung 7 kata saja [1]. Kemudian oleh *Nirmala* dan *Meghana* juga berhasil mengkonversi citra ke teks memakai *Raspberry Pi* dengan camera *webcam logitech 3MP*, akan tetapi fokus *webcam* harus dilakukan secara manual setiap pengambilan gambar [2]. Penulis pada penelitian ini bermaksud untuk mewujudkan prototipe yang dapat mengkonversi citra dokumen ke teks dengan tulisan yang relatif lebih banyak, dengan fokus yang lebih stabil (tidak manual).

Penelitian ini mewujudkan konstruksi prototipe alat yang dapat menangkap citra teks melalui modul kamera *Raspberry Pi* yang memiliki sensor *Omniroy 5647 5MP* (2592×1944 piksel) dalam fokus tetap. Memakai sistem operasi *Raspbian* berbasis *Debian GNU Linux* dilengkapi dengan program *Python* dan pustaka untuk keperluan computer vision seperti *Opencv*. Unsur fokus dan pencahayaan menjadi sangat penting, jarak kamera dan obyek dokumen harus disesuaikan untuk mendapatkan fokus yang mendekati seragam pada tiap sisi dokumen dengan cahaya yang merata. Konstruksinya harus mampu mengadopsi keperluan *adjustment* jarak fokus dan *alignment* kamera untuk menemukan posisi yang optimum.

Pencahayaan pada citra teks pada kenyataannya cenderung tidak merata. Keadaan tersebut memerlukan proses filterisasi untuk menyelaraskan unsur pencahayaan. Prototipe ini menggunakan filter *homomorphic* yang menggunakan sebuah nilai *threshold*. Nilai *threshold* ini digunakan sebagai nilai ambang batas piksel mana yang masuk kategori *foreground* dan piksel mana yang masuk kategori *background*. Pencahayaan yang tidak merata seringkali menyebabkan adanya bagian teks yang hilang, dan bagian teks yang hilang dari proses binarisasi ini otomatis tidak muncul di hasil OCR [3].

Rancangan ini menggunakan *Tesseract* sebagai mesin *OCR* yang dikelola oleh *Google* karena *Tesseract* sudah mengalami perkembangan cukup lama sejak pertama kali oleh *Hawlett Packard* (HP) pada tahun 1984 sebelum akhirnya menjadi *Open Source* dan mulai dikelola *Google* sejak tahun 2006.[4]

2. Metode Penelitian

Model penelitian dilakukan secara bertahap sesuai dengan model penelitian rancang bangun suatu sistem atau alat. Tahapan analisisnya adalah sebagai berikut:

- Studi Literatur: Mempelajari materi yang berhubungan dengan perangkat *Raspberry Pi*, sistem operasi linux *Rasbian*, pemrograman *Python*, *library Opencv* dan *Tesseract-OCR*.
- Analisa Kebutuhan Rancangan: Pada tahap ini peneliti menentukan pokok rancangan berupa blok diagram dan alur program yang harus ada pada sistem. Konstruksi alat harus cukup stabil untuk menyangga komputer, kamera dan *power supply*.
- Rancangan dan Realisasi: Berisi pembahasan detail dari blok diagram, perakitan konstruksi modul rangkaian, serta rancangan *Bash Shell* dan *Python* untuk mengatur alur kerja perangkat keras sehingga mencakup keseluruhan sistem secara lengkap.
- Pengujian dan Pembahasan: Tahapan ini untuk menemukan *setting* kisaran jarak optimum antara kamera dengan obyek dokumen, *alignment* kamera dan menguji unjuk kerja fungsional alat.

2.1 Penelitian Sebelumnya

Penelitian serupa oleh Nagaraja, Nagarjun (2015), merekayasa alat yang ditujukan untuk dapat membantu orang buta dalam membacakan teks tertulis yang merupakan alternatif tulisan yang biasanya menggunakan teks yang dipindai oleh *scanner*. Penelitian tersebut telah berhasil mengkonversi teks yang difoto melalui webcam beresolusi tinggi, akan tetapi penelitian tersebut berfokus obyek dengan areal yang tidak luas (hanya beberapa kata saja bukan kalimat) dan tidak disebutkan apakah alat tersebut dapat mengkonversi teks pada dokumen dengan *footprint* yang lebih luas.

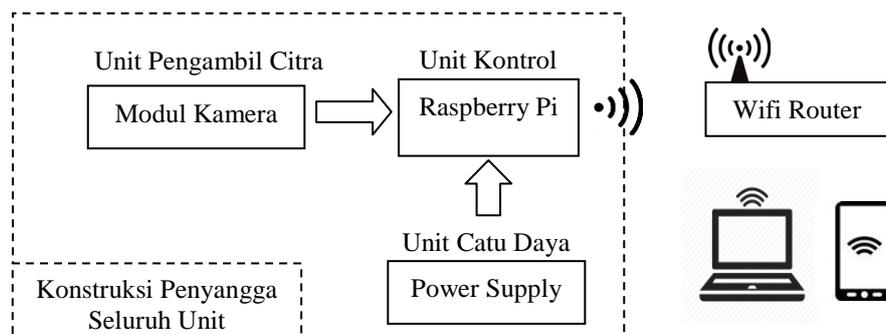
Penelitian selanjutnya dilakukan oleh Nirmala dan Meghana (2016) yang berjudul “*Image Text to Speech Conversion Using OCR Technique in Raspberry Pi*” mengangkat masalah yang sama yaitu mencoba merekayasa alat yang dapat mengolah teks yang tercetak menjadi suara dengan menggunakan *webcam logitech 3MP* untuk mengambil citra teks. Disebutkan bahwa sistem sering mengalami disfokus pada kamera sehingga perlu bantuan untuk memfokuskan kamera secara manual untuk dapat memperoleh citra yang diharapkan.

Penelitian oleh Naser Jawas (2015) mengangkat masalah citra dokumen yang cenderung mengalami pencahayaan yang tidak merata di beberapa bagian citra yang diambil menggunakan kamera handphone. Oleh karena itu, pada penelitian ini diusulkan metode binarisasi citra dokumen menggunakan filterisasi *homomorphic* untuk menanggulangi tingkat pencahayaan yang tidak merata. Hasilnya didapatkan tingkat *error* yang rendah dari pengenalan *OCR* pada citra hasil filterisasi *homomorphic*.

3. Hasil dan Pembahasan

3.1 Blok Diagram Rancangan

Rancangan utama Gambar 1. terdiri atas Unit Pengambil Citra, Unit Kontrol dan Unit Catu Daya, sedangkan unit pelengkap terdiri atas *Wifi Router* dan *Laptop/Smartphone* untuk kendali nirkabel. Ketiga unit utama tadi akan dipasang pada suatu konstruksi yang kokoh dan stabil terutama untuk menyangga kamera agar tidak mudah goyang/getar saat digunakan untuk foto dokumen. Unit kontrol sendiri tidak memungkinkan untuk dipasangi layar monitor, keyboard/mouse, karena koneksi kabelnya sangat rentan mengganggu kestabilan konstruksi. Untuk itu rancangan memerlukan kendali nirkabel



Gambar 1. Blok Diagram Rancangan

melalui laptop atau smartphone.

Unit Pengambil Citra adalah modul kamera yang dapat dihubungkan langsung ke konektor *Camera Serial Interface (CSI)* 15 pin pada *Raspberry Pi*. Kamera ini mampu menghadirkan gambar beresolusi 5MP yang jernih dan dapat merekam video HD 1080p pada 30fps. Kamera ini memiliki fokus tetap jarak jauh (*infinite*) sehingga sulit fokus pada jarak obyek yang sangat dekat (foto makro). Kamera dikendalikan melalui perintah teks pada *Shell* linux.

Unit Kontrol menggunakan Komputer *Raspberry Pi* 3 Model B adalah generasi ketiga yang dibangun dengan prosesor *Broadcom 2837 ARMv8* 64bit. Komputer di *boot* dari *MicroSD card* yang menjalankan sistem operasi linux *Raspbian* yang merupakan sistem turunan dari *Debian GNU Linux*. Program *Python*, *VNC Server*, *SSH Server* dan module *driver* kamera sudah terpasang secara *default* pada saat instalasi sistem operasi sehingga hanya perlu diaktifkan saja. Aplikasi tambahan yang dipasang melauai paket *Raspbian* adalah aplikasi *Imagemagick* untuk memanipulasi gambar, *Tesseract-ocr* untuk deteksi karakter dan *library opencv* yang merupakan pustaka untuk keperluan *vision computing*. *Opencv* harus dipasang secara manual pada sistem, karena paket instalasinya tidak tersedia, pemasangan dilakukan dengan kompilasi langsung melalui *source code*.

Raspberry Pi dan Kamera dicatu oleh sumber daya listrik 5Volt DC yang berasal dari perangkat *Powerbank* yang biasa digunakan untuk mengisi daya *smartphone*. Penulis memasang catu daya berkapasitas 11000 mAh untuk memastikan *supply* daya yang cukup karena selain komputer dan kamera, *wifi adapter* juga aktif untuk sistem kedali nirkabel.

3.1 Konstruksi

Rangkaian membutuhkan konstruksi yang kokoh untuk menyangga semua unit (Gambar 2) dan kamera harus dipasang menghadap kebawah untuk membuat foto dokumen. Peneliti menggunakan *Tripod* (Gambar 2A) sebagai penyangga utama sistem, sekaligus untuk mengadopsi keperluan *adjustment* fokus kamera melalui kaki *Tripod* yang dapat diatur.

Kamera yang terpasang secara manual pada *Tripod* tentunya tidak bisa betul-betul tegak lurus dengan lantai, jika kamera di tempel mati (*fixed*) ke *Tripod* maka pengaturan ulang posisi kamera tidak akan memungkinkan. Oleh karena itu peneliti memanfaatkan mainan anak *Brick Lego* berbentuk *ball joint* (Gambar 2B) sebagai adapter antara tiang *Tripod* dengan modul kamera sehingga posisi kamera dengan mudah diatur ulang dan bahkan dapat dilepas dan dipasang kembali (Gambar 2C). Gambar 2D memperlihatkan koneksi kabel pita pin 15 yang menghubungkan modul kamera dengan *Raspberry*, dan Gambar 2E menunjukkan posisi kerja unit secara keseluruhan dimana kamera menghadap kebawah.



Gambar 2. Konstruksi Rancangan

- A. Tripod kamera sebagai konstruksi penyangga utama.
- B. Brick/lego dengan *ball joint* dipilih untuk memudahkan *adjustment* kamera.
- C. Aplikasi brick/lego yang di lem ke Tripod dan alas Kamera.
- D. Rangkaian Raspberry Pi dan Modul Kamera
- E. Rangkaian lengkap yang terpasang pada Tripod

3.2 Kendali dan Referensi

Untuk menjaga kestabilan sistem, maka konstruksi unit dan koneksi seperti pada Gambar 2E bersifat tetap (*fixed*) tanpa penambahan koneksi kabel lain pada *Raspberry Pi*. Tambahan koneksi/kabel lain misalnya minitor, keyboard/mouse akan rentan merubah posisi *alignmet* kamera. Aplikasi *VNC* dan *SSH Client* pada Laptop atau Smartphone memungkinkan akses nirkabel ke *VNC Server* (Grafis) dan *SSH Server (Command Line)* pada *Raspberry Pi* melalui Akses Poin (Gambar 1). Akses Poin diatur sedemikian rupa untuk memberikan *IP* yang tetap pada *Raspberry Pi* sehingga kendali nirkabel dari laptop menjadi lebih stabil. Jika koneksi dari *VNC Client* ke *VNC Server* berhasil dilakukan maka *display* pada layar monitor laptop akan persis sama dengan tampilan di *Raspberry Pi*.

Sebelum mengambil citra gambar, yang pertama harus dilakukan adalah membuat referensi obyek terhadap kamera. Karena rancangan ini tidak memiliki sistem *preview* sebelum mengambil foto, maka foto akan cenderung tidak proporsional (*miring*) karena sulit meletakkan dokumen yang sesuai dengan *footprint* kamera tanpa adanya referensi yang pasti. Referensi dibuat menggunakan kertas putih yang diberi garis silang dan melintang di tengahnya (Gambar 3). Sebelum kertas dengan garis referensi tersebut di lem ke lantai, maka posisi kertas atau kamera diatur sedemikian rupa agar referensi tegak lurus dengan *grid* (garis kotak-kotak) kamera (Gambar 3). *Grid* pada kamera dibuat dengan mengadopsi shell script oleh *Fred Weinhaus* di situs aplikasi *Imagemagick*.



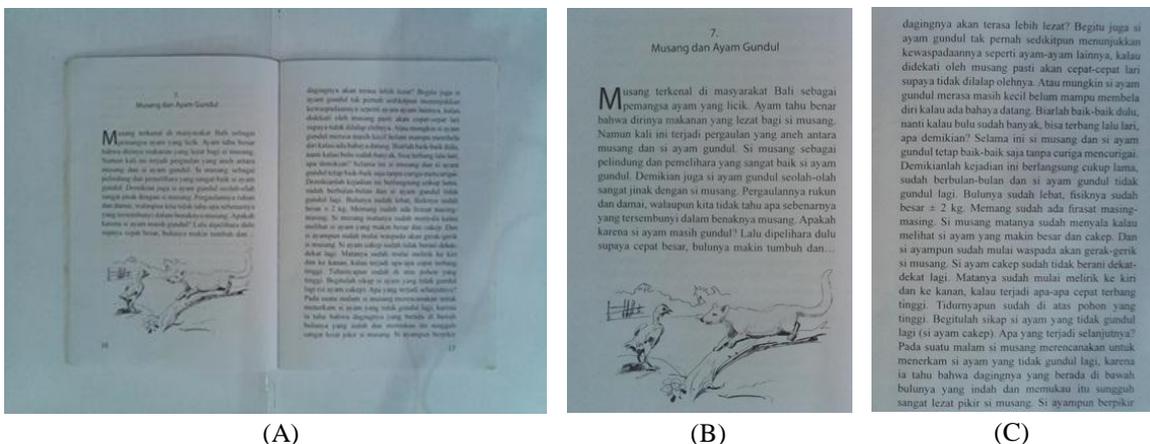
Gambar 3. Membuat referensi obyek terhadap kamera

3.3 Hasil

Rancangan ini menghasilkan sistem prototipe yang akan mengolah citra dokumen menjadi karakter teks dengan proses sebagai berikut:

Citra Dokumen → Crop → Binarisasi → OCR → Dokumen Dengan Karakter Teks

Rancangan menggunakan program *Python* dan *Bash script* untuk mengontrol kerja sistem yang diawali dengan mengambil citra dokumen seperti contoh Gambar 4A. Penulis memilih untuk meletakkan kamera pada jarak kurang lebih 36 cm dari lantai, hal ini dilakukan untuk mengurangi efek gelombang pada permukaan buku yang tidak rata jika jaraknya diperpendek. Tulisan dengan efek gelombang (bukan *miring*) pada titik tertentu akan diabaikan oleh proses pengenalan karakter sehingga akan memunculkan banyak kesalahan. Permukaan obyek foto harus dibuat setara mungkin dengan lantai.



(A) (B) (C)

Gambar 4. Contoh Citra Dokumen dan Hasil dari Proses Crop

Citra dokumen akan mengalami proses *crop* yang menghasilkan citra yang kiri dan yang kanan, dimana citra kiri (Gambar 4B) mengandung gambar binatang, dan yang disebelah kanan dengan teks penuh (Gambar 4C). Proses *crop* kedua citra tersebut terjadi sekaligus saat program berjalan dengan aplikasi *Imagemagick* yang meletakkan *fixel frame* pada argumen perintah *crop*.

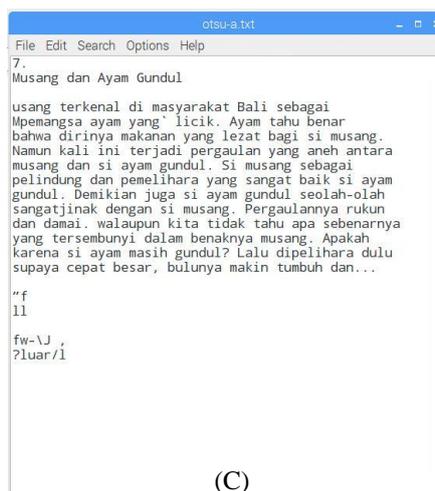
Pada proses selanjutnya, citra *crop* tadi akan mengalami proses filterisasi untuk menanggulangi dampak cahaya yang tidak merata (filter *homomorphic*) sebelum akhirnya masuk ke proses *Binarisasi Otsu* yang akan merubah citra berada pada dua sisi yaitu gelap dan terang dan tidak ada warna yang lain seperti pada gambar 5A-B. Hasil dari proses binarisasi akan masuk ke proses akhir yaitu pengenalan karakter (OCR) dengan aplikasi *Tesseract*, dan diperoleh hasil seperti pada Gambar 5C-D.



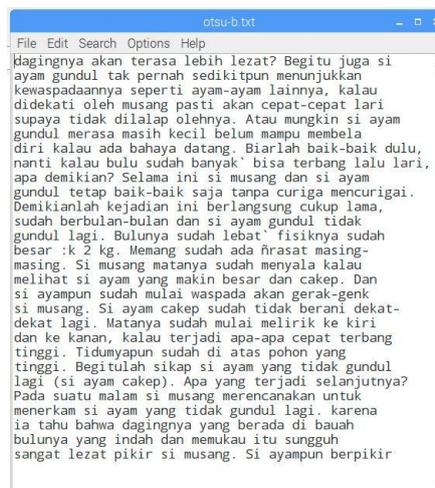
(A)

dagingnya akan terasa lebih lezat? Begitu juga si ayam gundul tak pernah sedikitpun menunjukkan kewaspadaannya seperti ayam-ayam lainnya, kalau didekati oleh musang pasti akan cepat-cepat lari supaya tidak dilalap olehnya. Atau mungkin si ayam gundul merasa masih kecil belum mampu membela diri kalau ada bahaya datang. Biarlah baik-baik dulu, nanti kalau bulu sudah banyak, bisa terbang lalu lari, apa demikian? Selama ini si musang dan si ayam gundul tetap baik-baik saja tanpa curiga mencurigai. Demikianlah kejadian ini berlangsung cukup lama, sudah berbulan-bulan dan si ayam gundul tidak gundul lagi. Bulunya sudah lebat, fisiknya sudah besar ± 2 kg. Memang sudah ada firasat masing-masing. Si musang matanya sudah menyala kalau melihat si ayam yang makin besar dan cakap. Dan si ayampun sudah mulai waspada akan gerak-gerik si musang. Si ayam cakap sudah tidak berani dekat-dekat lagi. Matanya sudah mulai melirik ke kiri dan ke kanan, kalau terjadi apa-apa cepat terbang tinggi. Tidurnyapun sudah di atas pohon yang tinggi. Begitulah sikap si ayam yang tidak gundul lagi (si ayam cakap). Apa yang terjadi selanjutnya? Pada suatu malam si musang merencanakan untuk menerkam si ayam yang tidak gundul lagi, karena ia tahu bahwa dagingnya yang berada di bawah bulunya yang indah dan memukau itu sungguh sangat lezat pikir si musang. Si ayampun berpikir

(B)



(C)



(D)

Gambar 5. Hasil Dari Proses Binarisasi dan OCR

3.4 Pembahasan

Hasil pengambilan citra sangat menentukan hasil akhir, jika terdapat distorsi pada dokumen asli maka hasil pengenalannya karakternya akan memunculkan kesalahan. Kamera yang berjarak terlalu dekat akan mudah menghasilkan fokus yang tidak merata dan apabila kamera berjarak terlalu jauh citra akan menjadi kurang jelas/tajam. Jadi jarak kamera 36 cm terhadap lantai yang diusulkan oleh peneliti bukannya jarak baku karena belum ada penelitian khusus untuk itu, jarak tersebut diambil karena alasan keseragaman cahaya yang diperoleh dan kejelasan citra teksnya.

Binarisasi Otsu menghasilkan citra hitam putih yang relatif baik (Gambar 5A-B), tulisan di buku cerita tersebut memang mengandung unsur putih dan hitam saja atau tidak mengandung warna yang lain,

citra asli (Gambar 4B-C) tertangkap agak gelap oleh kamera karena pengaruh pencahayaan, penulis melakukan test pada sistem menggunakan cahaya natural biasa tanpa penerangan yang spesifik.

Hasil OCR pada gambar 5C memunculkan kesalahan dibagian bawah dokumen karena sistem berusaha mengenali gambar binatang menjadi karakter teks, begitu juga di awal paragraf, seharusnya “Musang” bukan “usang” karena citra yang mengandung huruf “M” besar diletakkan pada pagragraf berikutnya (salah posisi). Berbeda dengan hasil OCR pada gambar 5D yang tidak mengandung citra gambar, maka pengenalan karakternya mendekati sempurna walaupun masih ada kesalahan minor, citra “firasat” menjadi “nrasat”, “fi” dikenali sebagai “n” karena karena hurufnya menempel.

4. Simpulan

Penelitian ini merancang sistem prototipe dengan menggunakan *Tripod* sebagai struktur utama untuk menyangga komputer *Raspberry Pi* dan modul kamera 5MP. Sistem yang dicatu oleh *powerbank* 5V / 11000 mAh ini telah berhasil menjalankan fungsinya untuk mengenali karakter (*OCR*) teks dari dokumen yang citranya diambil oleh kamera melalui algoritma yang dijalankan menggunakan *script Bash* dan *Python* dengan *library Opencv*.

Saran yang dapat diberikan untuk mengurangi kesalahan pengenalan karakter adalah memilih dokumen yang memiliki citra karakter yang homogen, tidak memiliki citra gambar dan tidak memiliki model huruf yang menempel satu sama lain. Permukaan dokumen harus dibuat rata untuk menghasilkan fokus yang homogen pada seluruh permukaan dokumen. Penambahan lampu LED pada pin *GPIO Raspberry pi* bisa dijadikan opsi untuk menambah pencahayaan.

Daftar Pustaka

- [1] NagarajaL, Nagarjun, Nishanth, Nithin & Veena. *Vison Based Text Recognition using Raspberry Pi*. International Journal of Computer Applications. National Conference on Power Systems & Industrial Automation (NCPSIA). 2015. www.ijcaonline.org. (p.975)
- [2] K Nirmala Kumari, Meghana Reddy J. *Image Text to Speech Conversion Using OCR Technique in Raspberry Pi*. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE). 2016. Vol-5: (p.3563)
- [3] Naser Jawas. *Binarisasi Citra Dokumen dengan Filterisasi Homomorphic*. SNIF.2015.
- [4] Ray Smith, Google Inc. *An Overview of the Tesseract OCR Engine*. ICDAR '07. Proceedings of the Ninth International Conference on Document Analysis and Recognition . 2007. Vol-2: (p.629-633)
- [5] Rafael C Gonzalez, Richard E Woods. *Digital Image Processing*. Second Edition. New Jersey 07458: Prentice Hall Upper Saddle River. 2008: (p.191-193)
- [6] Simon Monk. *Raspberry Pi Cook Book*. Second Edition. CA 95472. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol. 2016: (p-117)

Internet:

- [7] Improving the quality of the output. <https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality>. Diakses 5 Juli 2017.