

# Aplikasi Kalkulator Tulisan Tangan Sederhana Menggunakan Optical Character Recognition (OCR)

Supriadi<sup>1\*</sup>

<sup>1</sup>Program Studi Teknik Informatika, Universitas Nurul Jadid, Probolinggo

*\*Corresponding Author*

*E-mail: supriadiadefa85@gmail.com*

## Abstrak

Kalkulator adalah alat hitung yang banyak digunakan berbagai bidangnya bisnis dan perdagangan. Penggunaan kalkulator memudahkan manusia dalam melakukan operasi hitung, tetapi terdapat kendala dalam proses penginputan angka jika ingin menghitung nilai dari angka-angka pada media tulis seperti kertas, papan tulis dan sebagainya. Pengguna harus terlebih dahulu melihat teks pada media tulis, kemudian membacanya dan mengingatkannya lalu menyetik tulisan tersebut pada alat atau aplikasi kalkulator. Kekurangan dari cara ini adalah ketika pengguna lupa dengan tulisan pada media tulis, maka pengguna akan melihat teks tulisan dan mengingatkannya lagi sehingga membutuhkan waktu yang lebih lama dalam melakukan perhitungan menggunakan kalkulator. Metode yang digunakan pada penelitian ini adalah Optical Character Recognition, metode ini dapat mengenali teks yang terdapat pada gambar atau citra tulisan tangan operasi bilangan matematika. Hasil dari pengenalan teks tersebut kemudian akan dilakukan perhitungan aritmatika untuk mendapatkan hasil hitungnya. Dari uji coba terhadap 20 citra tulisan tangan operasi bilangan matematika, diperoleh hasil akurasi ekstraksi sebesar 85% dan akurasi citra tulisan tangan yang dapat dihitung dan benar sebesar 85%.

**Kata kunci:** *Image Processing, Optical Character Recognition, Handwritten Recognition*

## Abstract

*The calculator is a calculation tool that is widely used in various specialized fields of business and commerce. The use of a calculator makes it easier for humans to perform arithmetic operations, but there are obstacles in the process of inputting numbers if you want to calculate the value of numbers on written media such as paper, whiteboards and so on. The user must first see the text on written media, then read it and remember it then type the writing on a calculator tool or application. The drawback of this method is that when the user forgets the writing on the written media, the user will see the written text and remember it again so that it takes longer to perform calculations using a calculator. The method used in this study is Optical Character Recognition, this method can recognize text contained in images or handwritten images of mathematical number operations. The results of the text recognition will then be carried out by arithmetic calculations to get the calculation results. From the trials on 20 handwritten images of mathematical number operations, the results obtained were 85% accuracy of extraction and accuracy of handwritten images that can be calculated and correct by 85%.*

**Keywords:** *Image Processing, Optical Character Recognition, Handwritten Recognition.*

## 1. PENDAHULUAN

Kalkulator merupakan alat hitung yang ditemukan oleh Blaise Pascal seorang matematikawan bangsa Prancis pada tahun 1642. Perkembangan kalkulator telah berevolusi mulai dari segi ukuran hingga fitur yang semakin canggih[1]. Penggunaan kalkulator banyak digunakan dalam berbagai

bidang salah satu nya bisnis atau transaksi penjualan. Kalkulator dapat memberikan hasil perhitungan yang tepat dan efisien daripada menghitung secara manual yang memerlukan waktu yang lebih lama dan masih dapat berpotensi terjadi kesalahan dalam perhitungan.

Penggunaan kalkulator memudahkan manusia dalam melakukan operasi hitung, tetapi terdapat kendala dalam proses penginputan angka jika ingin menghitung nilai dari angka-angka pada media tulis seperti kertas, papan tulis dan sebagainya. Pengguna harus terlebih dahulu melihat teks pada media tulis, kemudian membacanya dan mengingatnya lalu mengetik tulisan tersebut pada alat atau aplikasi kalkulator. Kekurangan dari cara ini adalah ketika pengguna lupa dengan tulisan pada media tulis, maka pengguna akan melihat teks tulisannya lagi dan mengingatnya lagi sehingga membutuhkan waktu yang lebih lama dalam melakukan perhitungan menggunakan kalkulator. Oleh karena itu dibutuhkan suatu metode agar komputer dapat membaca tulisan tangan khusus nya angka dan operator matematika serta dapat melakukan proses perhitungan secara otomatis tanpa harus melakukan pengetikan atau penginputan angka lagi. Untuk mengatasi permasalahan diatas, penelitian ini akan membuat Aplikasi Kalkulator Tulisan Tangan Sederhana Dengan Menggunakan Optical Character Recognition (OCR). Beberapa metode lain juga akan diterapkan dalam penelitian ini, salah satu nya dengan menggunakan metode image processing atau pengolahan citra.

Metode ini berfungsi untuk mengambil gambar atau citra tulisan tangan angka dan operator matematika yang kemudian mengubahnya menjadi informasi yang bisa diproses. Informasi yang didapat dari citra tulisan tangan tersebut kemudian akan diterjemahkan oleh OCR untuk mendapatkan teks angka dan operator matematika yang kemudian dilakukan operasi aritmatika agar mendapatkan hasil perhitungan.

Tujuan penelitian ini adalah untuk membuat aplikasi kalkulator tulisan tangan sederhana menggunakan Optical Character Recognition (OCR) dengan menambahkan metode segmentasi proyeksi histogram. Dengan adanya penelitian ini, diharapkan dapat menerapkan metode Image Processing dan Optical Character Recognition untuk menerjemahkan dan menghitung citra atau gambar tulisan tangan yang mengandung operasi bilangan matematika dengan akurat.

Penelitian tentang Optical Character Recognition juga pernah dilakukan oleh beberapa penelitian sebelumnya. Sebuah penelitian dilakukan oleh M Maskuri dalam melakukan pengenalan karakter dari citra huruf alphabet dengan metode template machine yang mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi acuan nya. Citra masukan akan dibandingkan dengan citra acuan yang terdapat pada basis data dan menghasilkan tingkat kesamaan yang menentukan citra tersebut dikenali sebagai salah satu citra acuan. Penelitian ini menghasilkan akurasi dibawah 80% dari 312 data uji dan data training sebanyak 104 buah[2].

Kemudian penelitian selanjutnya dilakukan oleh Ali Mahmudi dkk dengan objek citra tulisan tangan menggunakan metode Image Processing yang terdiri dari grayscale dan thresholding serta segmentasi dan cropping untuk menghasilkan citra yang baik untuk proses pengenalan karakter. Adapun metode pengenalan karakter pada penelitian ini menggunakan metode Jaringan Syaraf Tiruan Backpropagation. Penelitian ini menghasilkan akurasi sebesar 70% dari pengujian terhadap 7 buah citra tulisan tangan [3].

Penelitian berikutnya dilakukan oleh Febri Liantoni dengan objek citra karakter angka dengan menggunakan metode Integral Proyeksi atau Proyeksi Histogram yang berfungsi untuk menghitung jumlah piksel per baris dan jumlah piksel per kolom. Kelebihan dari metode ini yaitu pemrosesan yang sederhana dan cepat dalam mengidentifikasi suatu citra digital. Sedangkan

untuk pengenalan angka pada penelitian ini menggunakan metode ekstraksi fitur yang digunakan untuk mencocokkan karakter angka dari fitur integral proyeksi vertikal dan horizontal yang telah didapatkan sebelumnya. Pada pengujian terhadap 20 gambar uji, menghasilkan nilai akurasi sebesar 65% hasil ini membuktikan bahwa metode integral proyeksi mampu melakukan pengenalan karakter angka. Hasil metode integral proyeksi bergantung dari pre-processing citra. Jika hasil pre-processing citra menghasilkan gambar yang baik, maka hasil dari integral proyeksi akan menjadi sempurna [4].

## 2. METODOLOGI

### 2.1. Landasan Teori

#### 2.1.1. Aritmatika

Aritmatika berasal dari bahasa Yunani arithmos yang berarti angka. Aritmatika adalah cabang tertua dan paling dasar dari matematika yang digunakan oleh semua orang, Aritmatika biasa digunakan untuk melakukan perhitungan biasa. yang mencakup penjumlahan, pengurangan, perkalian, dan pembagian [5].

#### 2.1.2. Citra Digital

Citra adalah suatu gambaran yang merupakan kemiripan atau imitasi dari sebuah objek. Citra terbagi menjadi dua jenis, yaitu citra analog dan citra digital. Citra analog adalah citra yang tidak dapat diolah oleh komputer. Contohnya adalah lukisan, foto yang tercetak di kertas, hasil CT scan, dan lain sebagainya. Sedangkan citra digital merupakan citra yang dapat disimpan kedalam memori dan dapat diolah oleh komputer. Adapun beberapa jenis citra digital yang biasa digunakan yaitu Citra Biner, Citra Grayscale dan Citra Berwarna [6].

#### 2.1.3. Thresholding

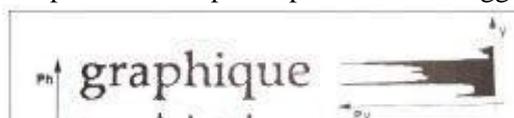
*Thresholding* merupakan proses mengubah citra *grayscale* menjadi citra hitam putih atau biner yang mempunyai nilai piksel 0 dan 1. *Thresholding* bertujuan untuk mengetahui region yang termasuk objek citra dan latar belakang citra [7]. *Thresholding* dapat didefinisikan pada persamaan 1.

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) \geq T \\ 0, & \text{if } f(x, y) < T \end{cases} \quad (1)$$

Dimana  $g(x,y)$  merupakan citra biner hasil dari citra *grayscale*  $f(x,y)$  dan  $T$  merupakan nilai ambang (*threshold*). Jika nilai piksel pada citra melewati nilai ambang, maka intensitas warna pada citra akan diubah menjadi 1. Sebaliknya jika nilai piksel pada citra kurang dari nilai ambang, intensitas warna pada citra akan diubah menjadi 0 [8].

#### 2.1.4. Histogram Proyeksi

Histogram proyeksi adalah sebuah metode untuk mencari daerah atau lokasi dari objek. Metode ini menjumlahkan piksel per baris dan piksel per kolom sehingga dapat diketahui batas-



Gambar 1. Histogram Proyeksi Horizontal dan Vertikal

batas dari objek yang berbeda [4]. Histogram proyeksi terdiri dari dua bagian yaitu histogram proyeksi horizontal dan histogram proyeksi vertikal [9].

Histogram proyeksi vertikal digambarkan dengan suatu vector  $P_v$  berukuran  $N$  dari sebuah citra biner  $S(N,M)$ , dimana  $N$  menyatakan banyaknya baris pada citra. Histogram proyeksi vertikal pada baris ke- $i$ , yaitu  $P_v[i]$ , histogram proyeksi vertikal dapat didefinisikan pada persamaan 2.

$$P_v[i] = \sum_{j=1}^M S[i, j] \quad (2)$$

Histogram proyeksi horizontal digambarkan dengan suatu vector  $P_h$  berukuran  $M$ , dimana  $M$  menyatakan banyaknya kolom pada citra. Histogram proyeksi horizontal pada kolom ke- $j$ , yaitu  $P_h[j]$ , Histogram proyeksi horizontal dapat didefinisikan dengan persamaan 3.

$$P_h[j] = \sum_{i=1}^N S[i, j] \quad (3)$$

#### 2.1.5. OpenCV

OpenCV merupakan sebuah library untuk visi komputer yang bersifat open source yang dibuat oleh Intel dan dapat dioperasikan pada komputer yang mempunyai sistem operasi Windows ataupun Linux. OpenCV dapat digunakan untuk penerapan interaksi manusia- komputer, identifikasi objek, segmentasi, serta pengenalan objek [10].

#### 2.1.6. OCR (Optical Character Recognition)

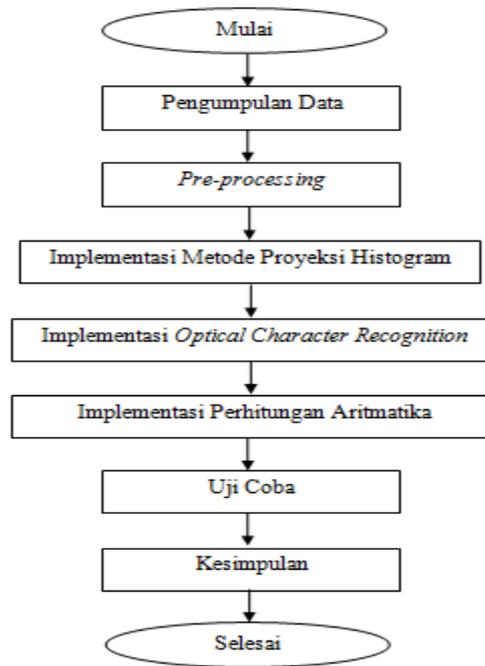
Optical Character Recognition merupakan sebuah sistem pada komputer yang mampu membaca karakter (huruf, angka maupun simbol tertentu) yang terdapat pada citra atau foto. Foto dapat berupa tulisan cetak maupun tulisan tangan. Hasil ekstraksi foto tersebut menghasilkan teks dengan menyesuaikan pola karakter per baris dengan pola yang terdapat pada database OCR. Tingkat akurasi pada OCR sangat bergantung pada kualitas citra dan metode yang digunakan pada pengolahan citra [11]. Kelebihan dari sistem OCR yaitu pengguna dapat memasukkan data tanpa harus menggunakan papan ketik, melainkan dapat menggunakan pena elektronik untuk menulis layaknya menulis di sebuah kertas [12].

#### 2.1.7. Tesseract OCR

Tesseract OCR merupakan sebuah mesin OCR yang dikembangkan oleh HP (Hewlett-Packard) pada tahun 1984 sampai 1994. Pada akhir 2005, Google mengambil alih proyek tersebut setelah HP merilis Tesseract untuk open source. Tesseract merupakan mesin OCR terbaik yang paling akurat dan bersifat open source [13].

## 2.2. Metode

Metodologi yang digunakan dalam penelitian ini meliputi pengumpulan data citra tulisan tangan operasi bilangan matematika, pre-processing, segmentasi citra menggunakan metode proyeksi histogram, implementasi optical character recognition, implementasi perhitungan aritmatika, uji coba, serta kesimpulan dari uji coba yang telah dilakukan. Adapun kerangka penelitian dapat dilihat pada Gambar 2.



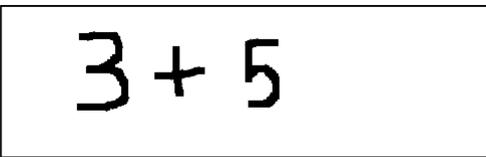
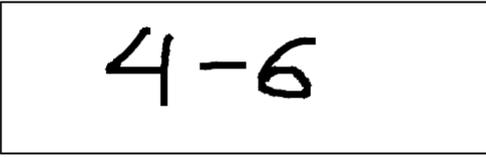
Gambar 2. Kerangka Penelitian

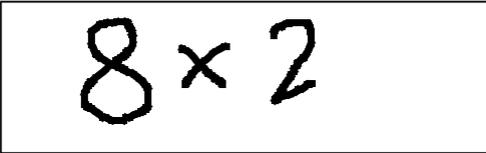
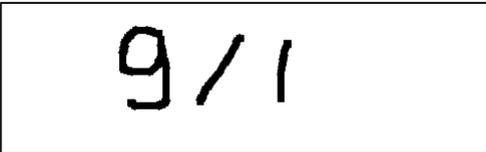
### 3. HASIL DAN PEMBAHASAN

#### 3.1 Penyajian Data Uji Coba

Pada penelitian ini *dataset* yang digunakan adalah citra tulisan tangan operasi bilangan matematika 20 citra yang terdiri dari 5 citra tulisan tangan operasi bilangan penambahan, 5 citra tulisan tangan operasi bilangan pengurangan, 5 citra tulisan tangan operasi bilangan perkalian dan 5 citra tulisan tangan operasi bilangan pembagian. Contoh *dataset* yang digunakan dapat dilihat pada Tabel 1.

Tabel 1. Contoh *Dataset*

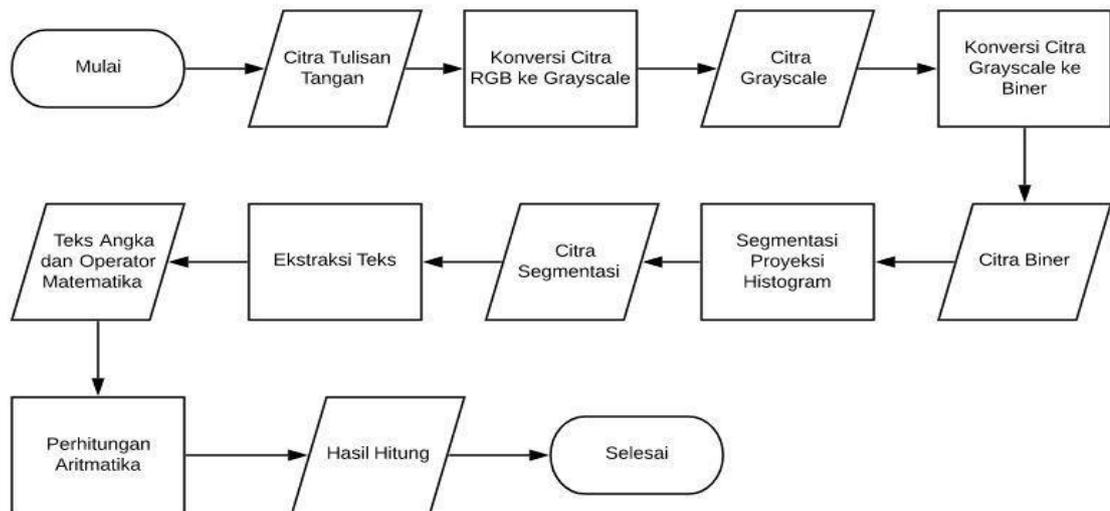
No	Nama Citra	Keterangan	Citra
1	penambahan1.png	Operasi bilangan penjumlahan	
2	pengurangan1.png	Operasi bilangan pengurangan	

3	perkalian1.png	Operasi bilangan pengurangan	
4	pembagian1.png	Operasi bilangan pembagian	

Pengujian dilakukan pada citra secara individu, dengan warna background canvas dan warna tulisan yang sama baik untuk citra tulisan tangan penambahan, pengurangan, perkalian dan pembagian.

### 3.2 Hasil Analisis

Pada bagian ini ditampilkan hasil uji coba alir sistem yang telah dilakukan mulai dari tahap *pre-processing*, segmentasi citra dengan menggunakan proyeksi histogram, implementasi OCR, dan perhitungan aritmatika hingga tahap uji coba.



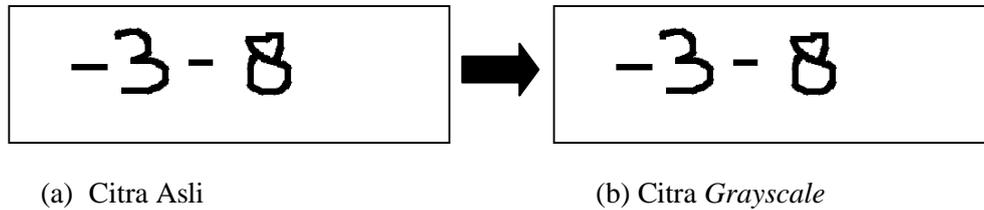
**Gambar 3.** Diagram Alir Sistem Kalkulator Tulisan Tangan

Cara kerja dari sistem ini adalah dengan menulis angka dan operator matematika pada media tulis, kemudian hasil dari tulisan tangan tersebut akan disimpan kedalam bentuk gambar dan kemudian menjadi masukkan untuk sistem yang selanjutnya diproses sesuai tahapan- tahapan pada Gambar 3.

#### A. *Pre-processing* citra

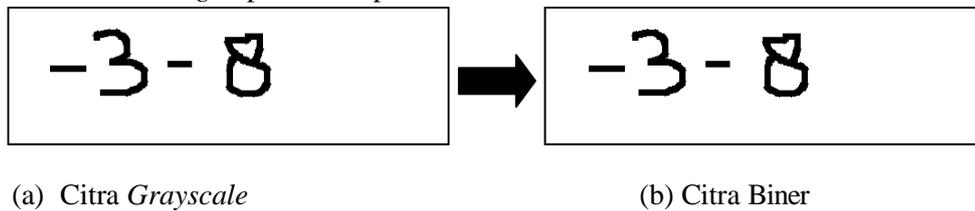
Pada tahap ini, citra tulisan tangan akan diproses menjadi citra *grayscale* yang bertujuan untuk memudahkan proses pengolahan citra, pada tahap *pre-processing* ini, dataset citra tulisan tangan

yang berjumlah 20 gambar diubah menjadi gambar berskala keabuan. Adapun proses konversi citra tulisan tangan menjadi citra *grayscale* dapat dilihat pada Gambar 4, sebagai berikut:



**Gambar 4.** Konversi Citra Asli Menjadi *Grayscale*

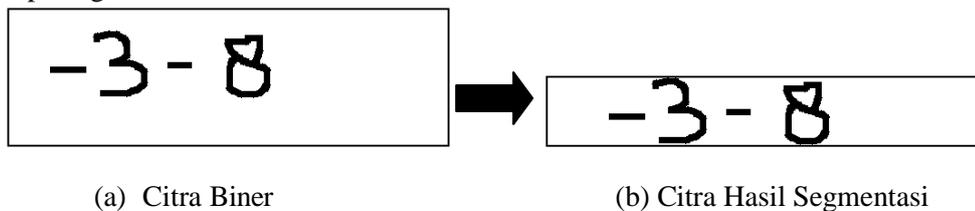
Setelah citra *grayscale* didapatkan, langkah selanjutnya yaitu dengan mengubah citra *grayscale* menjadi citra biner yang hanya mempunyai nilai intensitas 0 dan 1. Cara mengubah citra *grayscale* menjadi citra biner dapat dilakukan dengan menggunakan metode *thresholding*. Adapun proses *thresholding* dapat dilihat pada Gambar 5.



**Gambar 5.** Konversi Citra *Grayscale* Menjadi Citra Biner

#### B. Segmentasi Citra Menggunakan Proyeksi Histogram

Langkah selanjutnya yaitu melakukan segmentasi yang bertujuan untuk mengambil wilayah yang merupakan objek teks pada citra. Pada penelitian ini metode yang digunakan yaitu Proyeksi Histogram Horizontal untuk mendapatkan baris teks. Hasil dari segmentasi proyeksi histogram ditunjukkan pada gambar 6.



**Gambar 6.** Hasil Proyeksi Histogram

#### C. Implementasi *Optical Character Recognition (OCR)*

Pada tahap ini dilakukan ekstraksi teks dari citra hasil segmentasi yang telah dilakukan. Sebelum melakukan ekstraksi teks, langkah yang harus dilakukan yaitu dengan mengimport *library* Tesseract yang berfungsi untuk mengenali teks pada citra. Kemudian mengatur lokasi tesdata yang merupakan tempat *database* pengenalan karakter pada Tesseract. Pada penelitian ini output yang dikeluarkan adalah teks angka dan operator matematika, sehingga dilakukan pengaturan teks yang diinginkan dengan menggunakan fungsi *whitelist* dan pencocokkan karakter agar hasil *output* seperti yang diharapkan. Hasil *output* dari ekstraksi teks citra tulisan tangan operasi bilangan matematika dapat dilihat pada gambar 7.

```

run:
-3-8

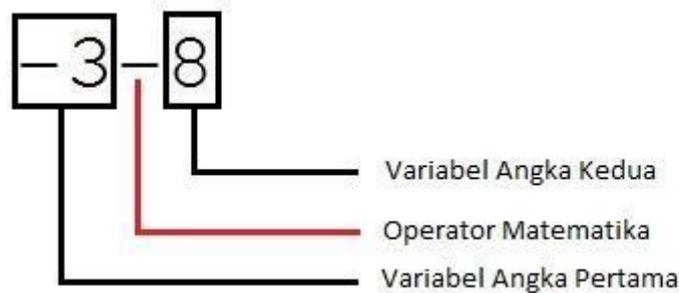
BUILD SUCCESSFUL (total time: 9 seconds)

```

Gambar 7. Hasil Ekstraksi Teks

#### D. Perhitungan Aritmatika

Setelah hasil ekstraksi teks didapatkan, terakhir dilakukan perhitungan aritmatika dengan melakukan operasi hitung antara variabel angka pertama dan variabel angka kedua dengan operator matematika yang terdapat pada citra tulisan tangan.



Gambar 8. Proses Perhitungan Aritmatika

Pada contoh hasil *output* ekstraksi teks dari sebuah dataset yang terdapat pada Gambar 8. algoritma perhitungan aritmatika dimulai dengan cara mencari variabel angka pertama dan variabel angka kedua dari teks bilangan operasi matematika. Kemudian mencari operator matematika yang terdapat pada teks, Untuk bilangan negatif, jika terdapat tanda – (minus) diawal angka maka angka tersebut akan diubah menjadi bilangan negatif dan digabungkan menjadi 1 variabel. Jika operator ditemukan, maka akan dilakukan perhitungan aritmatika antara variabel angka pertama dan variabel angka kedua. Adapun implementasi dari perhitungan aritmatika pada Gambar 8. dapat dilihat pada persamaan 4.

Diketahui : Hasil ekstraksi teks yaitu -8-3

Variabel angka pertama : -8

Variabel angka kedua : 3 Operator matematika : -

Sehingga perhitungan aritmatika nya adalah :

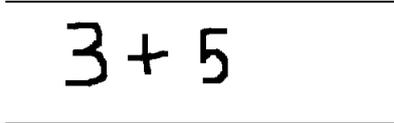
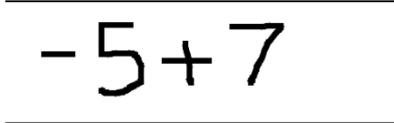
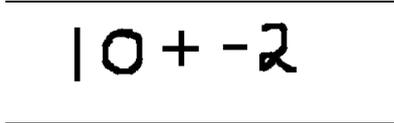
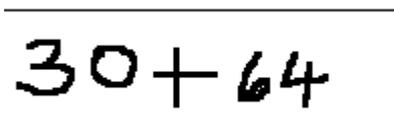
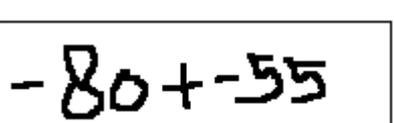
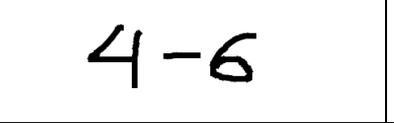
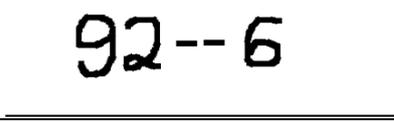
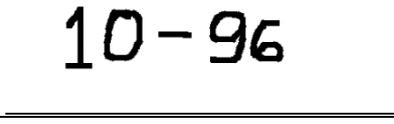
$$\begin{aligned}
 \text{Hasil Hitung} &= \text{Variabel angka pertama} - \text{Variabel angka kedua} && (4) \\
 &= -8 - 3 \\
 &= -11
 \end{aligned}$$

### 3.3 Hasil Analisis

Uji coba hasil ekstraksi teks dilakukan terhadap citra tulisan tangan operasi bilangan matematika. Jumlah dataset yang digunakan yaitu sebanyak 20 citra yang terdiri dari 5 citra tulisan tangan operasi bilangan penambahan, 5 citra tulisan tangan operasi bilangan pengurangan, 5 citra

tulisan tangan perkalian dan 5 citra tulisan tangan operasi bilangan pembagian. Berikut ini merupakan hasil ekstraksi teks dari citra tulisan tangan operasi bilangan matematika yang ditunjukkan pada Tabel 2.

**Tabel 2.** Hasil Ekstraksi Teks Citra Tulisan Tangan Operasi Bilangan Matematika

No	Nama Citra	Citra Asli	Output Teks	Pencocokan
1	penambahan1.png		3+5	Cocok
2	penambahan2.png		-5+7	Cocok
3	penambahan3.png		10+-2	Cocok
4	penambahan4.png		30+64	Cocok
5	penambahan5.png		-80+-55	Cocok
6	pengurangan1.png		4-6	Cocok
7	pengurangan2.png		-3-8	Cocok
8	pengurangan3.png		92--6	Cocok
9	pengurangan4.png		10-96	Cocok

10	pengurangan5.png	$-69--30$	-69—30	Cocok
11	perkalian1.png	$8 \times 2$	8*2	Cocok
12	perkalian2.png	$-6 \times 4$	-6*4	Cocok
13	perkalian3.png	$71 * -3$	71*-3	Cocok
14	perkalian4.png	$22 * 44$	29*44	Tidak Cocok
15	perkalian5.png	$-10 * -10$	-10*-20	Tidak Cocok
16	pembagian1.png	$9/1$	9/1	Cocok
17	pembagian2.png	$-6/3$	-6/3	Cocok
18	pembagian3.png	$12 / -4$	12/-4	Cocok
19	pembagian4.png	$70 : -2$	70-2	Tidak Cocok
20	pembagian5.png	$-84 / -12$	-84/-12	Cocok
Total Ekstraksi Teks Yang Cocok :				17

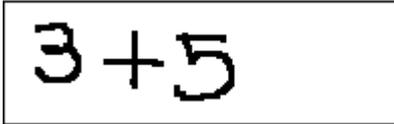
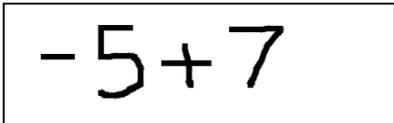
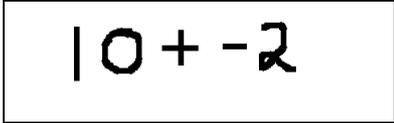
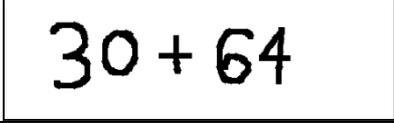
Dari tabel hasil ekstraksi teks diatas, terdapat beberapa hasil ekstraksi teks yang tidak cocok dengan citra tulisan tangan. Hal itu disebabkan karena Tesseract OCR mengklasifikasikan karakter tersebut menjadi karakter lain yang tidak sesuai dengan harapan seperti angka 2 yang terklasifikasi menjadi angka 9 pada gambar perkalian4.png dan juga dapat terklasifikasi menjadi angka 1 pada gambar perkalian5.png. Penyebab lainnya yaitu terdapat karakter yang tidak terdeteksi seperti yang ditunjukkan pada gambar pembagian4.png, Dimana Tesseract OCR tidak mampu mengenali karakter : (titik dua) yang merupakan simbol pembagian akan tetapi dapat membaca simbol pembagian yang lain nya yaitu / (garis miring). Untuk mengetahui akurasi dari ekstraksi teks dapat dilihat pada persamaan 5.

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah Hasil Ekstraksi Teks Cocok}}{\text{Jumlah Total Gambar}} \times 100\% \quad (5) \\
 &= \frac{17}{20} \times 100\%
 \end{aligned}$$

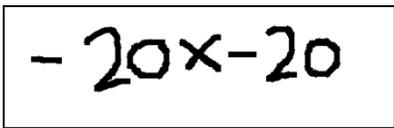
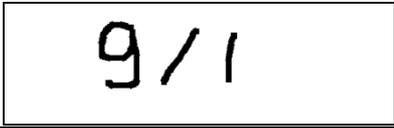
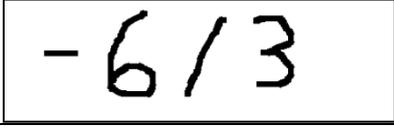
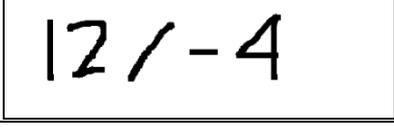
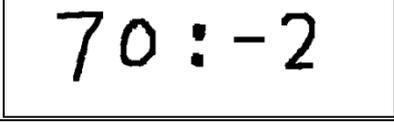
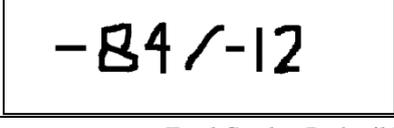
$$\text{Akurasi} = 85\%$$

Setelah pengujian ekstraksi teks terhadap citra tulisan tangan selesai dilakukan, pengujian berikutnya yaitu untuk mengetahui hasil perhitungan aritmatika dari teks angka dan huruf dari hasil ekstraksi teks. Indikator penilaian dari uji coba perhitungan aritmatika yaitu hasil perhitungan sesuai dengan tulisan tangan yang dibuat. Adapun hasil uji coba perhitungan aritmatika dari hasil ekstraksi teks yang dilakukan dapat dilihat pada Tabel 3.

Tabel 3. Hasil Uji Coba Perhitungan Aritmatika

No	Citra	Hasil Ekstraksi	Hasil Hitung	Keterangan
1		3+5	8	Benar
2		-5+7	2	Benar
3		10+-2	8	Benar
4		30+64	94	Benar

5	$-80 + -55$	$-80 + -55$	-135	Benar
6	$4 - 6$	$4 - 6$	-2	Benar
7	$-3 - 8$	$-3 - 8$	-11	Benar
8	$92 - -6$	$92 - -6$	98	Benar
9	$10 - 96$	$10 - 96$	-86	Benar
10	$-69 - 30$	$-69 - 30$	-39	Benar
11	$8 * 2$	$8 * 2$	16	Benar
12	$-6 * 4$	$-6 * 4$	-24	Benar
13	$71 * -3$	$71 * -3$	-213	Benar
14	$22 * 44$	$29 * 44$	1276	Salah

15		-10*-20	200	Salah
16		9/1	9	Benar
17		-6/3	-2	Benar
18		12/-4	-3	Benar
19		70-2	68	Salah
20		-84/-12	7	Benar
Total Gambar Berhasil Dihitung dan Benar				17

Dari tabel hasil perhitungan diatas, dari uji coba yang dilakukan terhadap 20 dataset citra tulisan tangan operasi bilangan matematika diperoleh total gambar yang berhasil dihitung dan benar sebanyak 17 citra, sedangkan 3 citra yang lainnya menghasilkan hasil perhitungan yang salah. Hal itu disebabkan oleh hasil ekstraksi teks yang tidak sesuai sehingga mempengaruhi hasil dari perhitungan. Untuk mengetahui akurasi dari uji coba perhitungan aritmatika dapat dihitung dengan rumus sebagai berikut:

$$Akurasi = \frac{\text{Jumlah Gambar Berhasil Dihitung \& Benar}}{\text{Jumlah Total Gambar}} \quad (1)$$

$$= \frac{17}{20} \times 100\%$$

$$Akurasi = 85\%$$

#### 4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa :

1. Aplikasi kalkulator tulisan tangan sederhana mampu melakukan perhitungan aritmatika dari citra tulisan tangan operasi bilangan matematika.

2. Tesseract OCR mampu melakukan ekstraksi teks pada citra tulisan tangan. Untuk menghasilkan output berupa karakter angka dan operator matematika dapat dilakukan dengan cara mendaftarkan karakter angka dan operator pada daftar whitelist.
3. Dari uji coba terhadap 20 citra tulisan tangan operasi bilangan matematika diperoleh hasil akurasi ekstraksi teks sebesar 85% dan akurasi citra tulisan tangan yang dapat dihitung dan benar sebesar 85%.

## 5. DAFTAR PUSTAKA

- [1] D. Hidayat, "Penggunaan kalkulator dalam pengajaran matematika sekolah dasar," *Cakrawala Pendidik.*, p. 87949, 1997.
- [2] M. Maskuri, "Implementasi Metode Template Matching Untuk Media Pembelajaran Pengenalan Karakter Huruf Alphabet," *Simki-Techsain*, vol. 01, no. 11, 2017.
- [3] A. Mahmudi, R. El Maghfiroh, and A. P. Sasmito, "Aplikasi MATLAB untuk Mengenali Karakter Tulisan Tangan." MATICS, 2017.
- [4] F. Liantoni, "Pengenalan karakter angka menggunakan metode Integral Proyeksi," *J. Ilm. Teknol. Sist.*, p. 57, 2017.
- [5] M. Rohmah, R. M. Akbar, and A. N. K. Sari, "Perancangan Aplikasi Game Aritmatika Dasar Berbasis Android," *J. Saintek*, vol. 13, no. 1, pp. 36–43, 2016.
- [6] T. d Sutoyo, E. Mulyanto, V. Suhartono, O. D. Nurhayati, and others, "Teori pengolahan citra digital," *Yogyakarta Andi*, 2009.
- [7] E. Maria, Y. Yulianto, Y. P. Arinda, J. Jumiaty, and P. Nobel, "Segmentasi Citra Digital Bentuk Daun Pada Tanaman Di Politani Samarinda Menggunakan Metode Thresholding," *J. Rekayasa Teknol. Inf.*, vol. 2, no. 1, pp. 37–46, 2018.
- [8] H. Masrani, I. R. Ilhamsyah, and others, "Aplikasi Pengenalan Pola pada Huruf Tulisan Tangan Menggunakan Jaringan Saraf Tiruan dengan Metode Ekstraksi Fitur Geometri," *Coding J. Komput. Dan Apl.*, vol. 6, no. 2, 2018.
- [9] A. Septiarini, "Segmentasi Karakter Menggunakan Profil Proyeksi," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 7, no. 2, pp. 66–69, 2016.
- [10] K. D. Irianto, "Pendeteksi Gerak Berbasis Kamera Menggunakan Opencv Pada Ruangan," 2010.
- [11] A. Setiawan, H. Sujaini, and A. B. P. Negara, "Implementasi Optical Character Recognition (OCR) pada Mesin Penerjemah Bahasa Indonesia ke Bahasa Inggris," *JUSTIN (Jurnal Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp. 135–141, 2017.
- [12] M. R. Fauzi, A. A. Zahra, and others, "Mengubah Tulisan Tangan Menjadi Text Digital Ocr (Optical Character Recognition) Dengan Menggunakan Metode Segmentasi Dan Korelasi," *Transient J. Ilm. Tek. Elektro*, vol. 2, no. 4, pp. 1013–1017, 2014.
- [13] P. Chakraborty and A. Mallik, "An open source tesseract based tool for extracting text from images with application in braille translation for the visually impaired," *Int. J. Comput. Appl.*, vol. 68, no. 16, 2013.