

Sistem Pencocokan Aksara Bali Menggunakan Template Matching Dan Komputasi Paralel

I Gede Arta Wibawa¹, I Gede Tendy Ariyanto², Luh Arida Ayu Rahning Putri³, I Ketut Gede Suhartana⁴, I Gusti Agung Gede Arya Kadyanan⁵, Luh Gede Astuti⁶

Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana

Jalan Kampus Bukit Jimbaran, Badung, Bali

Email : gede.arta@unud.ac.id¹, tendy.ariyanto99@gmail.com², rahningputri@unud.ac.id³,
ikg.suhartana@unud.ac.id⁴, gungde@unud.ac.id⁵, lq.astuti@unud.ac.id⁶

ABSTRAK

Aksara bali merupakan aksara tradisional yang digunakan untuk menulis bahasa dan sastra bali. Aksara bali juga merupakan salah satu kekayaan budaya bali. Namun aksara bali memiliki keunikan dari bentuknya yang hampir sama satu dengan yang lainnya, dan beberapa tulisan hanya dibedakan oleh satu guratan garis. Hal tersebut dapat mempersulit dalam pembelajarannya juga dalam pengembangan aplikasinya di era yang modern ini. Jadi oleh sebab saya melakukan penelitian dalam pembuatan sistem pencocokan aksara bali menggunakan template matching dan komputasi paralel agar dapat mengetahui berapa nilai dari rata2 ketidakmiripan dari setiap huruf aksara bali agar dapat menjadi acuan dalam pengembangan aplikasi yang berhubungan dalam aksara bali. Dalam penelitian ini saya menggunakan metode template matching untuk mengetahui nilai eror rata2 dari setiap huruf aksara bali yang terdapat pada font bali simbar. Dari metode tersebut akan mencocokkan setiap nilai pixel-pixel dari huruf aksara bali yang sudah diolah dalam proses binerisasi untuk mendapatkan nilai eror. Berdasarkan penelitian yang dilakukan, hasil yang didapat nilai MSE terbesar yaitu 0,071354167 memiliki kemiripan 93%. Dimana ini menunjukkan bahwa kemiripan aksara bali dengan yang lainnya itu memiliki kemiripan diatas 90% dan hasil waktu proses data sistem yang menggunakan komputasi paralel adalah 4977 second dan sistem yang tidak menggunakan komputasi paralel adalah 4789 second.

Kata kunci : Template matching, komputasi paralel, pencocokan aksara bali, MSE, bali simbar

1. PENDAHULUAN

Aksara bali merupakan aksara tradisional yang digunakan untuk menulis bahasa dan sastra bali. Aksara bali juga merupakan salah satu kekayaan budaya bali. Namun aksara bali memiliki keunikan dari bentuknya yang hampir sama satu dengan yang lainnya, dan beberapa tulisan hanya dibedakan oleh satu guratan garis[4]. Jadi oleh sebab saya melakukan penelitian dalam pembuatan sistem pencocokan aksara bali menggunakan template matching dan komputasi paralel agar dapat mengetahui berapa nilai dari rata2 ketidakmiripan dari setiap huruf aksara bali agar dapat menjadi acuan dalam pengembangan aplikasi yang berhubungan dalam aksara bali.

Dalam penelitian ini saya menggunakan metode template matching salah satu metode pengenalan citra dengan cara melakukan proses operasi pengurangan antara matrik citra uji dengan matriks citra latih, dan hasilnya dimutlakkan[2] dari setiap huruf aksara bali yang terdapat pada font bali simbar. Dari metode tersebut akan mencocokkan setiap nilai pixel-pixel dari huruf aksara bali yang sudah diolah dalam proses binerisasi untuk mendapatkan nilai eror. Nilai eror tersebut akan di cari rata2nya dan akan di buat table agar mempermudah dalam mencari nilai MSE yang terbesar atau tulisan aksara bali yang memiliki nilai kemiripan yang kecil. Dalam proses template matching tersebut juga di sisipkan komputasi paralel sekaligus melakukan perbandingan antara aplikasi yang menggunakan komputasi paralel dan tidak menggunakan komputasi paralel agar dapat mengetahui terjadi perbedaan waktu dalam pemrosesan atau tidak.

Berdasarkan penelitian yang dilakukan, hasil yang didapat nilai MSE terbesar yaitu 0,071354167 memiliki kemiripan 93%. Dimana ini menunjukkan bahwa kemiripan aksara bali

dengan yang lainnya itu memiliki kemiripan diatas 90% dan hasil waktu proses data sistem yang menggunakan komputasi paralel adalah 4977 second dan sistem yang tidak menggunakan komputasi paralel adalah 4789 second

2. METODE PENELITIAN

2.1 Aksara bali

Suara khususnya wicara merupakan cara yang natural bahkan paling penting dalam melakukan proses komunikasi. Dalam kehidupan sehari-hari, manusia melakukan berbagai jenis komunikasi dengan sesama manusia, misalnya: body language, berbicara (*speech*) dan lain-lain. Diantara banyak komunikasi yang dilakukan oleh manusia, berbicara (*speech*) memberikan paling banyak informasi penting dan paling penting dalam komunikasi.

Sejarah Aksara Bali tidak dapat terlepas dari perkembangan aksara dari India, dimana perkembangan agama Hindu dan Buddha di Negara Indonesia terutama Bali diiringi juga dengan bahasa dan aksara. India merupakan suatu negara yang dahulu memiliki aksara yang disebut dengan aksara Karosti. Aksara Karosti tersebut mengalami perkembangan menjadi aksara Brahmi. Aksara Brahmi tersebut kembali mengalami perkembangan menjadi aksara Dewanegari dan aksara Pallawa. Aksara Dewanegari digunakan di India Utara yakni untuk menuliskan bahasa Sanskerta. Aksara Pallawa digunakan di India Selatan yakni untuk menulis bahasa Pallawa. Perkembangan aksara Dewanegari dan Pallawa di Indonesia mengikuti perkembangan agama Hindu dan Buddha. Dari perkembangan agama Hindu dan Buddha tersebut maka Aksara Kawi atau Aksara Indonesia Kuna lahir. Aksara Kawi tersebut menjadi tonggak terciptanya aksara Jawi dan Aksara Bali, serta aksara-aksara lainnya yang sekarang ada di Indonesia. Salah satu contohnya adalah pada kerajaan Kutai (Kalimantan Timur) pada *yupa*, terdapat penulisan aksara Dewanegari.

Aksara Bali saat ini sebenarnya merupakan gabungan dari aksara Wresastra dan aksara Swalalita. Aksara Wresastra (Wreastra) merupakan aksara yang dipakai untuk menuliskan bahasa Bali umum, misalnya: surat, urak, pipil, pangling-eling dan lain-lainnya. Anggota dari aksara Wresastra (Wreastra) diantaranya adalah aksara suara, aksara wianjana, Pengangge Suara, Pengangge Tengenan, aksara suara h wanda, dan angka Bali dari Aksara Bali. Bagian-bagian dari aksara Wresastra (Wreastra) adalah sebagai berikut[5]:

- a. Aksara Wianjana merupakan aksara dengan anggota ha, na, ca, ra, ka, da, ta, sa, wa, la, ma, ga, ba, nga, pa, ja, ya, nya, seperti pada Tabel 1 di bawah :

Tabel 1 Tabel Aksara Bali

h	n	c	r	k	d
ha	na	ca	ra	ka	da
t	s	w	l	m	g
ta	sa	wa	la	ma	ga
b	\	p	j	y	z
ba	nga	pa	ja	ya	nya

2.2 Template Matching

Algoritma template matching merupakan metode yang sederhana dan banyak digunakan untuk mengenali pola Metode template matching ini merupakan metode yang sederhana. Suatu citra masukan yang mengandung template tertentu dibandingkan dengan template pada basis data. Template ditempatkan pada pusat bagian citra yang akan dibandingkan dan dihitung seberapa banyak titik yang paling sesuai dengan template. Langkah ini diulangi terhadap keseluruhan citra masukan yang akan dibandingkan. Nilai kesesuaian titik yang paling mendekati atau error minimum yang paling kecil antara citra masukan dan citra template menandakan bahwa template tersebut merupakan citra template yang paling sesuai dengan citra masukan.

Tingkat kesesuaian antara citra masukan dan citra template bisa dihitung berdasarkan nilai error terkecil dengan menggunakan persamaan 1.

$$\text{Min } e = \sum (I_{x,y} - T_{x,y})^2 \quad (1)$$

e = hasil template matching

x = kolom pixel

y = baris pixel

I adalah pola pixel citra masukan yang akan dibandingkan. T adalah pola pixel citra template.

Template dengan nilai error paling kecil adalah template yang paling sesuai dengan citra masukan yang akan dibandingkan.[1]

2.3 Komputasi Paralel

Teknologi komputasi paralel sudah berkembang lebih dari dua dekade, penggunaannya semakin beragam mulai dari kebutuhan perhitungan di laboratorium fisika nuklir, simulasi pesawat luar angkasa, hingga prakiraan cuaca. Komputasi paralel didefinisikan sebagai penggunaan sekumpulan sumberdaya komputer secara simultan untuk menyelesaikan permasalahan komputasi. Secara prinsip komputer paralel membagi permasalahan sehingga menjadi lebih kecil untuk dikerjakan oleh setiap prosesor (CPU) dalam waktu yang bersamaan/simultan (concurrent). Prinsip ini disebut paralelisme.

Paralelisme dalam komputasi paralel merupakan hal yang diciptakan dan dimanfaatkan. Sebenarnya prinsip paralelisme juga sudah diterapkan dalam komputer serial misal dengan pipelining dan superscalar-nya namun demikian tidak memberikan solusi terbaik dalam hal meningkatkan performansi dikarenakan terbatasnya kemampuan untuk menambah kecepatan prosesor dan fenomena memory bottleneck. Perkembangan penerapan paralelisme pada prosesor dari masa ke masa kita dapatkan bedakan ke beberapa tingkat paralelisme dalam komputasi khususnya pada prosesor.[3]

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Dataset

Data dalam penelitian ini adalah aksara bali yang ada pada font bali simbar yang proses menjadi sebuah image. Terdapat 142 aksara bali yang penulis gunakan sebagai data yang diambil dari font bali simbar dengan ukuran image 640 x 480 pixel. Berikut ada contoh gambar dari beberapa aksara bali yang di gunakan sebagai data penelitian.



Gambar 1. Contoh aksara bali



Gambar 2. Contoh aksara bali



Gambar 3. Contoh aksara bali

Data tersebut akan langsung di input kedalam sistem yang sudah siap dijalankan untuk mengetahui nilai MSE dari setiap aksara bali yang ada pada font bali simbar.

3.2 Implementasi Sistem

Terdapat beberapa proses dalam implementasi sistem pada penelitian ini. Proses proses dalam implementasi sistem ini adalah sebagai berikut.

- a. Sistem tanpa komputasi paralel
Berikut adalah proses – proses dalam implementasi sistem tanpa komputasi paralel.

1. Grayscale

Grayscale adalah proses pengubahan gambar masukan yang berupa gambar rgb menjadi gambar keabuan. Source code yang berfungsi untuk mengubah gambar rgb menjadi gambar keabuan terlihat pada tabel di bawah ini.

Tabel 2. Source code grayscale

```
img_main=cv2.imread("/home/agasatya/Desktop/aksarabali/aksara%d.jpg")
%(a)
img_gray0 = cv2.cvtColor(img_main, cv2.COLOR_BGR2GRAY)
template=cv2.imread("/home/agasatya/Desktop/aksarabali/aksara%d.jpg")
%(b)
img_gray1 = cv2.cvtColor(template, cv2.COLOR_BGR2GRAY)
```

Pada source code diatas akan mengambil data yang sudah di simpan pada folder tertentu dan membaca image tersebut kemudian akan di olah menjadi gambar keabuan. Gambar yang sudah menjadi keabuan akan di simpan di sebuah variable dan akan memasukin proses selanjutnya.

2. Binerisasi

Binerisasi merupakan proses untuk mengubah gambar menjadi hitam putih. Gambar hasil dari proses ini hanya terdiri dari warna hitam dan putih. Source code untuk proses binerisasi terlihat pada tabel di bawah ini.

Tabel 3. Source code binerisasi

```
ret, img_biner = cv2.threshold(img_gray0, 125, 255, cv2.THRESH_BINARY)
ret, img_biner1 = cv2.threshold(img_gray1, 125, 255, cv2.THRESH_BINARY)
```

Pada source diatas gambar akan diubah nialinya setiap pixel dimana gambar tersebut nilainya kurang sama dengan dari 125 akan menjadi 1 atau warna hitam dan yang lebih dari 125 akan menjadi 0 atau warna putih

3. Template matching

Template matching merupakan proses untuk mencocokkan nilai setiap pixel pada data tersebut. Berikut source code untuk proses template matching.

Tabel 4 source code template matching

```
h , w = img_biner.shape
    i = 0
    j = 0
    nilai_kemiripan = 0
    while (i < h) :
        while (j < w) :
            if (img_biner[i][j] != img_biner1[i][j]):
                nilai_kemiripan = nilai_kemiripan + 1
            j = j + 1
        j = 0
        i = i + 1
    mse = float(nilai_kemiripan)/float(h*w)
    ws.cell(row=(row+a), column=(column+b), value=(mse))
    b = b + 1
column = 1
b = 1
a = a+1
print(a)
wb.save('asd.xlsx')
```

Pada source diatas akan dilakukan pencocok nilai yang terdapat dari setiap pixel yang ada akan dicari nilai yang tidak sama dari data pertama dan data kedua itu akan dilakukan secara berulang-ulang samapai sebuah data didapatkan total dari nilai yang tidak sama dan akan di bagi total pixel yang ada pada data tersebut agar mendapat nilai MSEnya. Setelah mendapatkan nilai MSE tersebut makan akan disimpan dalam sebuah document excel.

b. Sistem menggunakan komputasi paralel

Pada implementasi sistem menggunakan komputasi paralel tidak jauh berbeda dari implementasi sistem tanpa komputasi paralel yang membedakan hanya pada implementasi

sistem menggunakan komputasi paralel dibagi menjadi 2 fungsi karena data akan dibagi untuk di jalan di thread yang berbeda dan thread tersebut akan berjalan bersama. Berikut adalah source dari sistem menggunakan komputasi paralel.

Tabel 5 source code menjalankan thread

```
import cv2
import numpy as np
from openpyxl import Workbook
from openpyxl import load_workbook
from threading import Thread
import time

start_time = time.time()
def job1():
    total_gambar = 142
    total_job = 71
    a=1
    b=1
    row = 1
    wb = load_workbook(filename = 'asd.xlsx')
    ws = wb.active
    column = 1
    while (a <= total_job) :
        while (b <= total_gambar) :
            img_main=
cv2.imread("/home/agasatya/Desktop/aksarabali/aksara%d.jpg" %(a))
            img_gray0=cv2.cvtColor(img_main,
cv2.COLOR_BGR2GRAY)
            template=
cv2.imread("/home/agasatya/Desktop/aksarabali/aksara%d.jpg" %(b))
            img_gray1=cv2.cvtColor(template, cv2.COLOR_BGR2GRAY)
            ret, img_biner = cv2.threshold(img_gray0, 125, 255,
cv2.THRESH_BINARY)
            ret, img_biner1 = cv2.threshold(img_gray1, 125, 255,
cv2.THRESH_BINARY)
                h , w = img_biner.shape
                i = 0
                j = 0
                nilai_eror = 0
                while (i < h) :
                    while (j < w) :
                        if (img_gray0[i][j] != img_gray1[i][j]):
                            nilai_eror = nilai_eror + 1
                                j = j + 1
                                    j = 0
                                        i = i + 1
                                            mse = float(nilai_eror)/float(h*w)
                                                ws.cell(row=(row+a), column=(column+b), value=(mse))
                                                    b = b + 1
                                                        column = 1
                                                            b = 1
                                                                a = a+1
                                                                    print 'thread1'
                                                                        wb.save('asd.xlsx')
                                                                            print("--- %s seconds ---" % (time.time() - start_time))

def job2():
    total_gambar = 142
    total_job = 142
```

```

a=72
b=1
row = 1
wb = load_workbook(filename = 'asd.xlsx')
ws = wb.active
column = 1
while (a <= total_job) :
    while (b <= total_gambar) :
        img_main=
cv2.imread("/home/agasatya/Desktop/aksarabali/aksara%d.jpg" %(a))
        img_gray0=cv2.cvtColor(img_main,
cv2.COLOR_BGR2GRAY)
        template=
cv2.imread("/home/agasatya/Desktop/aksarabali/aksara%d.jpg" %(b))
        img_gray1=cv2.cvtColor(template, cv2.COLOR_BGR2GRAY)
        ret, img_biner = cv2.threshold(img_gray0, 125, 255,
cv2.THRESH_BINARY)
        ret, img_biner1 = cv2.threshold(img_gray1, 125, 255,
cv2.THRESH_BINARY)
        h , w = img_biner.shape
        i = 0
        j = 0
        nilai_eror = 0
        while (i < h) :
            while (j < w) :
                if (img_gray0[i][j] != img_gray1[i][j]):
                    nilai_eror = nilai_eror + 1
                j = j + 1
            j = 0
            i = i + 1
        mse = float(nilai_eror)/float(h*w)
        ws.cell(row=(row+a), column=(column+b), value=(mse))
        b = b + 1
    column = 1
    b = 1
    a = a+1
print "thread2"
wb.save('asd.xlsx')
print("--- %s seconds ---" % (time.time() - start_time))

threads = []
threads1 = []
t = Thread(target=job1, args=())
threads.append(t)
t2 = Thread(target=job2, args=())
threads1.append(t2)
for t in threads :
    t.start()
for t in threads1 :
    t.start()

```

Pada source code diatas terdapat 2 fungsi yang sama hanya fungsi 1 (job1) dan fungsi 2 (job2) mengolah data yang berbeda agar dalam menjalankan thread yang berbeda pada waktu bersamaan akan mendapatkan waktu yang lebih cepat karena data yang di kerja di bagi menjadi 2.

3.3 Hasil dan Pengujian Sistem

Pada subbab ini akan dijelaskan mengenai hasil dari data yang sudah di olah oleh sistem dan juga perbandingan dari sistem yang menggunakan komputasi paralel dan yang tidak menggunakan komputasi paralel

a. Nilai MSE

Hasil dari semua nilai MSE dapat di lihat dari table yang berada pada bagian lampiran pada bagian data nilai MSE. Dari tabel pada lampiran bisa kita ketahui nilai MSE dari setiap data yang diolah dan juga kita bisa menemukan data yang nilai MSEnya 0 dikarena data tersebut di cocokan dengan dirinya sendiri jadi tidak terdapat nilai eror pada pencocokan tersebut. Juga dapat ditemukan ada beberapa data yang mendapatkan nilai eror yang besar. Berikut adalah tabel dari 10 nilai MSE terbesar yang didapatkan

Tabel 6 10 nilai MSE terbesar

Aksara pertama	Aksara kedua	Nilai MSE
ꦮ	ꦮ	0,071354167
ꦲ	ꦲ	0,066455078125
ꦲ	ꦲ	0.06605143229167
ꦲ	ꦲ	0.0653971354167
ꦲ	ꦲ	0.065384114583
ꦲ	ꦲ	0.06454752604167
ꦲ	ꦲ	0.063681640625
ꦲ	ꦲ	0.0619563802083
ꦲ	ꦲ	0.061829427083
ꦲ	ꦲ	0.0615657552083

b. Perbandingan waktu proses sistem

Setelah menjalankan sistem yang sudah selesai, output dari sistem tersebut ada yang berupa waktu lamanya proses sistem. Berikut adalah gambar jadi output waktu proses sistem menggunakan komputasi paralel dan sistem tanpa komputasi paralel

```
root@agasatya-G501JW:/home/agasatya/Desktop# python menakaithread.py
thread2
--- 4788.39378691 seconds ---
thread1
--- 4789.37117004 seconds ---
root@agasatya-G501JW:/home/agasatya/Desktop#
```

Gambar 4. Output waktu proses sistem menggunakan komputasi paralel

```
--- 4977.87736583 seconds ---
root@agasatya-G501JW:/home/agasatya/Desktop#
```

Gambar 5. Output waktu sistem tanpa menggunakan paralel

Dari gambar diatas dapatkan kita bandingkan bahwa waktu yang di butuhkan sistem tanpa menggunakan paralel untuk mennggolah data adalah 4977 second sedangkan pada sistem menggunakan paralel yang dimana thread 1 yang terakhir selesai dibandingkan thread 2 membutuhkan waktu 4789 seconds dalam memproses data.

4. KESIMPULAN

Pada penelitian yang dilakukan, diperoleh simpulan yaitu sebagai berikut.

1. Dari 10 nilai MSE terbesar nilai MSE yang paling besar adalah 0,071354167 yang merupakan pencocokan aksara 118 dengan aksara 130 dan juga 130 merupakan aksara yang sangat berbeda dengan aksara lainnya di karenakan 10 aksara terbesar kebanyakan pencocokan dari aksara 130.
2. Dari nilai MSE terbesar memiliki kemiripan 93%. Dimana ini menunjukkan bahwa kemiripan aksara bali dengan yang lainnya itu memiliki kemiripan diatas 92% dimana aksara bali satu dengan lainnya memiliki kemiripan yang sangar besar.
3. Hasil waktu proses data sistem yang menggunakan komputasi paralel adalah 4977 dan sistem yang tidak menggunakan komputasi paralel adalah 4789. Dapat disimpulkan bahwa sistem yang menggunakan komputasi paralel lebih cepat 3% dari pada sistem yang tidak menggunakan komputasi paralel.

REFRENSI

- [1] Dewi, M. S., Kesiman, M. W., & Sunarya, I. M. (2014). Aplikasi Pembelajaran Pengenal Aksara Bali Menggunakan Metode Template Matching. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 41-50.
- [2] Dwiadi, M. H., Hertiana, S. N., & Budiman, G. (2010). Pengenalan Bilangan Arab Menggunakan Tamplate Matching. *Telkom University*.
- [3] Ogi, D. (2010). Studi Kinerja Algoritma Paralel dengan MPICH2 dan Cilk++ pada Prosesor Multicore. *UNIVERSITAS INDONESIA*.
- [4] Pertama, P. G., Pande, Suyoto, & Suselo, T. (2015). Pengembangan Aplikasi Pengenalan Aksara Bali Kedalam Huruf Latin Dengan Augmented Reality. *Seminar Nasional Teknologi Informasi dan Komunikasi 2015*.
- [5] Setiyawan, P. A., Chayawan W., A. A., & Bayupati, I. P. (2014). Balinese Alphabet Sebagai Aplikasi Media Pembelajaran Aksara Bali Berbasis Android Mobile Platfrom. *Merpati Vol 2, No 2*.