

GAME EDUKASI PENGENALAN HURUF LATIN MENGUNAKAN TEMPLATE MATCHING UNTUK ANAK USIA DINI

Made Harry Dananjaya Adiartika^{a1}, I Gede Arta Wibawa.^{a2}, I Gusti Ngurah Anom Cahyadi
Putra^{a3}

^aFakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Denpasar, Indonesia

¹dananjaya.adiartika@gmail.com (Corresponding author)

²gede.arta@unud.ac.id

³anom.cp@unud.ac.id

Abstrak

Masa Kanak – kanak merupakan masa yang penting bagi perkembangan bagi seorang anak dimana pada masa tersebut anak – anak berada dalam fase dimana kemampuan motoric, sensorik maupun kognitif anak sedang mengalami perkembangan, tentunya ketiga kemampuan tersebut berkembang secara alami sejalan dengan usia anak, ketiga kemampuan tersebut juga dapat dikembangkan lebih lanjut lewat berbagai aktifitas baik fisik maupun mental, salah satu contoh aktifitas tersebut adalah menulis, menulis sendiri merupakan salah satu bentuk dari kemampuan kognitif yang penting dimiliki oleh seorang anak karena aktifitas menulis mendukung integrasi bahasa dan keterampilan literasi yang muncul meletakkan dasar bagi keterampilan membaca anak, terdapat berbagai metode atau cara yang biasanya digunakan untuk mengajarkan anak untuk menulis, salah satu metode yang sering digunakan saat ini adalah belajar menulis sambil bermain, penelitian ini memanfaatkan metode tersebut untuk mengembangkan sebuah media interaktif yang dapat digunakan untuk belajar menulis serta mengenalkan karakter huruf alfabet (A-Z), (a-z) dan angka (0-9) kepada anak yang dibuat dalam bentuk sebuah game edukasi, game edukasi ini menggunakan metode Template Matching untuk menilai hasil tulisan tangan anak dengan mencari tingkat kemiripan atau kecocokan tulisan anak tersebut dengan template karakter yang disediakan.

Berdasarkan penelitian yang dilakukan, pengenalan karakter dengan metode Template Matching dinyatakan berhasil dalam mengenali hasil tulisan karakter anak, dimana didapatkan akurasi dari pengenalan karakter menggunakan metode Template Matching terhadap hasil tulisan karakter huruf alfabet (A-Z), (a-z) dan angka (0-9) yang didapatkan, menghasilkan akurasi pengenalan karakter untuk karakter huruf alfabet (A-Z) sebesar 88%, karakter huruf alfabet (a-z) sebesar 86%, dan karakter angka (0-9) sebesar 87%.

Kata Kunci: Game, Game Edukasi, Template Matching, Character Recognition, Zhang-suen.

1. Pendahuluan

Masa Kanak – kanak merupakan masa yang penting bagi perkembangan bagi seorang anak dimana pada masa tersebut anak – anak berada dalam fase dimana kemampuan motoric, sensorik maupun kognitif anak sedang mengalami perkembangan, tentunya ketiga kemampuan tersebut berkembang secara alami sejalan dengan usia anak, ketiga kemampuan tersebut juga dapat dikembangkan lebih lanjut lewat berbagai aktifitas baik fisik maupun mental, salah satu contoh aktifitas tersebut adalah menulis, menulis sendiri merupakan salah satu bentuk dari kemampuan kognitif yang penting dimiliki oleh seorang anak, hal ini dikarenakan kegiatan

menulis mendukung integrasi bahasa penting dan keterampilan literasi yang muncul meletakkan keterampilan membaca anak – anak (Gerde, Bingham, & Wasik, 2012), terdapat berbagai metode atau cara yang biasanya digunakan untuk mengajarkan anak untuk menulis, salah satu metode yang sering digunakan saat ini adalah belajar menulis sambil bermain, Game terutama game edukasi merupakan salah satu media yang baik dalam menerapkan metode belajar sambil bermain, Game edukasi merupakan sebuah multimedia interaktif yang menerapkan materi edukasi kedalam permainannya sehingga game edukasi dapat digunakan untuk meningkatkan minat anak – anak dalam belajar mengenal dan menulis karakter dan karena sifatnya yang merupakan sebuah multimedia interaktif, Game edukasi tersebut dapat dikembangkan kedalam bentuk *multi-platform*, sehingga game edukasi tersebut dapat diakses dan digunakan dengan mudah oleh anak – anak

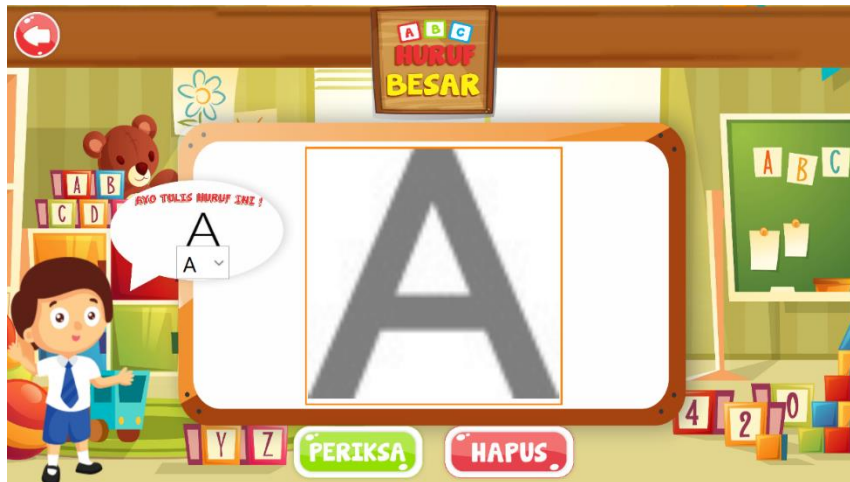
Game edukasi yang akan dikembangkan dalam penelitian ini merupakan sebuah game menulis yang dimana anak akan menulis karakter yang diinginkan pada sebuah kanvas dan kemudian akan periksa apakah karakter yang dibuat tersebut merupakan karakter yang dimaksud, proses ini disebut sebagai *Character Recognition*, *Character Recognition* merupakan sebuah bentuk image processing yang sudah tidak asing saat ini dan sudah banyak metode yang telah digunakan pada kasus pengenalan karakter terutama karakter huruf alfabet latin (A-Z), (a-z), maupun angka (0-9), salah satu metode yang sering digunakan adalah *template matching*, *template matching* merupakan sebuah metode pengenalan citra dengan yang secara singkat bekerja dengan mencocokkan (*matching*) tiap – tiap bagian dari suatu citra dengan citra yang dijadikan sebagai *template* (acuan), namun metode ini memiliki kelemahan yang terletak pada terbatasnya jumlah model yang dapat digunakan sebagai *template* pembanding pada basis data seperti ukuran, bentuk serta orientasi. (Solih, Ginting, & Sirati, 2019), hal ini dapat diatasi dengan pengolahan awal (*preprocessing*) citra sebelum digunakan dalam *template matching*.

Penelitian ini membahas tentang pengaplikasian metode *template matching* dalam pengenalan karakter (*Character Recognition*) untuk huruf latin baik alfabet (A-Z), (a-z) maupun angka (0-9) pada kasus game edukasi yang telah dikembangkan sebelumnya untuk penelitian ini dan melihat tingkat akurasi dari pengenalan citra karakter huruf latin yang dibuat oleh anak pada aplikasi game untuk karakter alfabet (A-Z), (a-z), dan angka (0-9).

1.1. Ayo Menulis

Ayo menulis merupakan sebuah *prototype* aplikasi game edukasi berbasis *progressive web application* (PWA) yang dibuat untuk mengenalkan karakter huruf alfabet latin (A-Z), (a-z) dan angka (0-9) untuk anak usia dini yang dibuat khusus untuk penelitian ini.

Aplikasi ini ditunjukkan kepada pengguna anak – anak, terutama anak – anak yang baru belajar mengenal karakter baik karakter huruf maupun angka, dimana aplikasi ini dapat digunakan untuk mempermudah anak dalam mengenal karakter huruf alfabet latin (A-Z), (a-z) maupun angka (0-9) melalui media latihan menulis secara langsung serta lewat video edukasi yang disediakan oleh aplikasi.

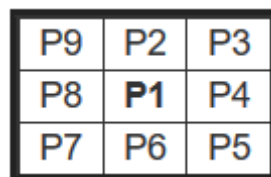


Gambar 1. Tampilan Game Ayo Menulis

1.2. Zhang Suen Thinning Algorithm

Algoritma *Thinning* Zhang-Suen disebut juga sebagai algoritma 2-pass, yang berarti bahwa untuk setiap iterasi melakukan dua set pemeriksaan untuk menghilangkan piksel dari citra. Pemeriksaan dirancang sehingga set pertama menghilangkan dari sudut tenggara citra (kanan bawah), dan set kedua menghapus dari sudut barat laut (kiri atas). Metode ini digunakan untuk ekstraksi kerangka citra yang terdiri dari menghapus semua poin kontur citra kecuali poin milik kerangka. (Rahmadwati & Wijono, 2016).

Secara umum algoritma ini beroperasi pada semua piksel hitam (P_1) yang mempunyai delapan tetangga, dengan urutan tetangga sebagai berikut



Gambar 2. Ilustrasi Tetangga P_1

Tentunya, batas piksel pada citra tidak dapat memiliki delapan tetangga, sehingga perlu mendefinisikan dua buah parameter, pertama definisikan parameter $A(P_1)$, jumlah transisi dari piksel putih ke hitam sesuai urutan tetangga ($P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_2$) dan kedua definisikan parameter $B(P_1)$, jumlah piksel hitam tetangga dari P_1 ($P_2 + P_3 + \dots + P_9$), setelah kedua parameter tersebut terdefinisi baru dapat masuk ke proses *thinning*, dimana algoritma ini terdiri dari dua tahap *thinning*.

a. Tahap 1

Semua piksel diuji dan piksel harus memenuhi semua kondisi yang tertera pada tahap ini.

1. (0) piksel merupakan piksel hitam dan memiliki delapan tetangga.
2. (1) $2 \leq B(P_1) \leq 6$
3. (2) $A(P_1) = 1$
4. (3) setidaknya salah satu dari P_2, P_4 , dan P_6 merupakan piksel putih.

5. (4) setidaknya salah satu dari P4, P6, dan P8 merupakan piksel putih.

Semua piksel hitam yang memenuhi kondisi tahap 1 akan diset menjadi piksel putih.

b. Tahap 2

Semua piksel akan kembali diuji dan piksel harus memenuhi semua kondisi yang tertera pada tahap ini

1. (0) piksel merupakan piksel hitam dan memiliki delapan tetangga.
2. (1) $2 \leq B(P1) \leq 6$
3. (2) $A(P1) = 1$
4. (3) setidaknya salah satu dari P2, P4, dan P6 merupakan piksel putih.
5. (4) setidaknya salah satu dari P4, P6, dan P8 merupakan piksel putih.

Semua piksel hitam yang memenuhi kondisi tahap 2 akan kembali diset menjadi piksel putih. Proses ini akan terus berulang sampai tidak ada piksel hitam yang memenuhi kondisi baik tahap 1 maupun tahap 2.

1.3. Template Matching

Menurut (Solin et al., 2019) *Template Matching* adalah salah satu teknik yang umum digunakan dalam pengolahan citra digital yang berfungsi mencocokkan (*matching*) tiap – tiap bagian dari suatu citra dengan citra yang dijadikan sebagai *template* (acuan), Metode *Template Matching* seringkali digunakan untuk mengidentifikasi atau mengenali (*recognition*) pola seperti citra karakter baik huruf, angka, sidik jari, dll, namun metode ini memiliki kelemahan yang terletak pada terbatasnya jumlah model yang dapat digunakan sebagai *template* pembandingan pada basis data seperti ukuran, bentuk serta orientasi.

Untuk algoritma *Template Matching* yang digunakan pada penelitian ini adalah algoritma *Template Matching* dengan pendekatan *Cross Correlation*, *Cross Correlation* menggunakan *correlation coefficient* sebagai ukuran kemiripan antara citra *template* untuk setiap lokasi x,y pada citra target (karakter), dimana akan didapatkan sebuah *matching point* antara citra *template* dengan citra target (karakter) dengan mencari *maximum value* dari x,y citra *template* pada lokasi x,y citra target (karakter), *template matching correlation coefficient* dapat dirumuskan sebagai berikut :

$$R(x, y) = \sum_{x', y'} (T(x', y') * I(x + x', y + y')) \quad (1)$$

Dimana T(x,y) merupakan nilai dari citra *piksel* template dan I(x,y) adalah nilai dari *piksel* masukan (Baek & Kim, 2009), dimana secara singkatnya untuk mendapatkan kecocokan dilakukan dengan menghitung nilai kecocokan terbaik (*best matches*) dengan mengkalikan nilai piksel citra *template* dengan nilai citra piksel masukan sehingga menghasilkan nilai *maximum similarity*.

1.4. Pengujian Akurasi

Pengujian aplikasi berfungsi untuk mengetahui kinerja dari aplikasi dalam melakukan tugas yaitu mengenali karakter yang ditulis oleh anak dengan mencocokkannya dengan citra *template*.

Untuk mengetahui tingkat akurasi keberhasilan aplikasi dalam mencocokkan karakter tulisan anak, dilakukan dengan cara membandingkan jumlah data karakter yang berhasil dicocokkan secara benar dengan total jumlah data uji, jika di implementasikan ke dalam rumus maka akan menjadi seperti pada rumus berikut :

$$Akurasi = \frac{\text{Jumlah pencocokan yang benar}}{\text{Total data testing}} \times 100\% \quad (2)$$

2. Metode Penelitian

2.1. Studi Pendahuluan

Sebelum penelitian dilaksanakan terlebih dahulu, dilakukan sebuah kajian pustaka mengenai hal – hal terkait yang diperlukan untuk melakukan pengenalan karakter dengan menggunakan *template matching* untuk kasus aplikasi game edukasi anak usia dini yang dikembangkan pada penelitian ini, baik dari artikel, jurnal maupun pustaka lainnya baik nasional maupun internasional.

2.2. Pengumpulan Data

Jenis data yang digunakan pada penelitian ini adalah data *primer* yang terbagi menjadi dua yaitu data latih dan data uji. Data latih pada penelitian ini adalah seluruh karakter yang terdapat pada huruf alfabet (A-Z), (a-z) dan angka (0-9). Sedangkan data uji adalah 25 citra karakter untuk masing – masing karakter alfabet (A-Z), (a-z) dan angka (0-9).

Data uji didapatkan secara langsung ketika pengguna menggunakan aplikasi yang dikembangkan pada penelitian ini. Data latih yang digunakan pada penelitian ini berjumlah 1 buah *template* per karakter (A-Z), (a-z), dan angka (0-9), sedangkan untuk data uji berjumlah 650 citra untuk karakter huruf alfabet (A-Z), 650 untuk karakter huruf alfabet (a-z) dan 250 untuk karakter huruf angka (0-9).

2.2.1. Data Latih

Data Latih berjumlah 62 data citra karakter *template* yang terdiri dari 26 citra karakter huruf alfabet latin (A-Z), 26 citra karakter huruf alfabet latin (a-z), dan 10 citra karakter angka arab (0-9), dimana masing – masing karakter memiliki 1 citra *template* yang digunakan sebagai data latih, citra *template* yang digunakan sebagai data latih merupakan citra karakter ketik (*typewritten*) yang dibuat secara digital, gambar 3. merupakan contoh dari data latih yang digunakan.



Gambar 3. Contoh Data Latih Template Karakter

2.2.2. Data Uji

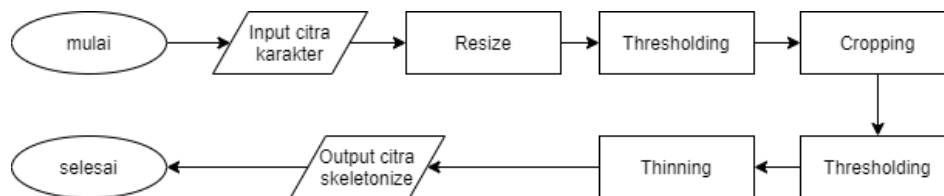
Data Uji berjumlah 1550 data citra karakter yang terdiri dari 650 citra karakter huruf alfabet latin (A-Z), 650 citra karakter huruf alfabet latin (a-z), dan 250 citra karakter angka arab (0-9), dimana masing – masing karakter memiliki 25 citra karakter yang digunakan sebagai data uji, citra karakter yang digunakan sebagai data uji merupakan citra karakter tulisan tangan (*handwritten*) yang dibuat oleh pengguna (responden) secara langsung pada kanvas yang disediakan pada aplikasi “Ayo Menulis” dan sudah melalui tahap *preprocessing* pada aplikasi, gambar 4.2. merupakan contoh dari data latih yang digunakan.



Gambar 4. Contoh Data Uji Karakter

2.3. Pengolahan Data Awal.

Data awal berupa citra diolah terlebih dahulu dalam proses *preprocessing* untuk mendapat citra yang dibutuhkan. Alur kerja pengolahan data awal yang dilakukan pada aplikasi terdapat pada gambar 5.

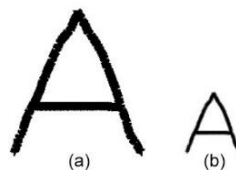


Gambar 5. Flowchart Alur Preprocessing

2.3.1. Preprocessing

a. Resize

Proses *Resize* atau *resizing* merupakan sebuah proses normalisasi citra ke ukuran normal sesuai dengan ukuran yang telah ditentukan sebelumnya, dimana ukuran yang digunakan pada penelitian ini adalah 64x64 piksel, dalam proses ini resolusi citra karakter akan di *scaling* (penskalaan) agar sama dengan resolusi *horizontal* dan *vertical* (panjang x lebar) citra *template* karakter tersebut.

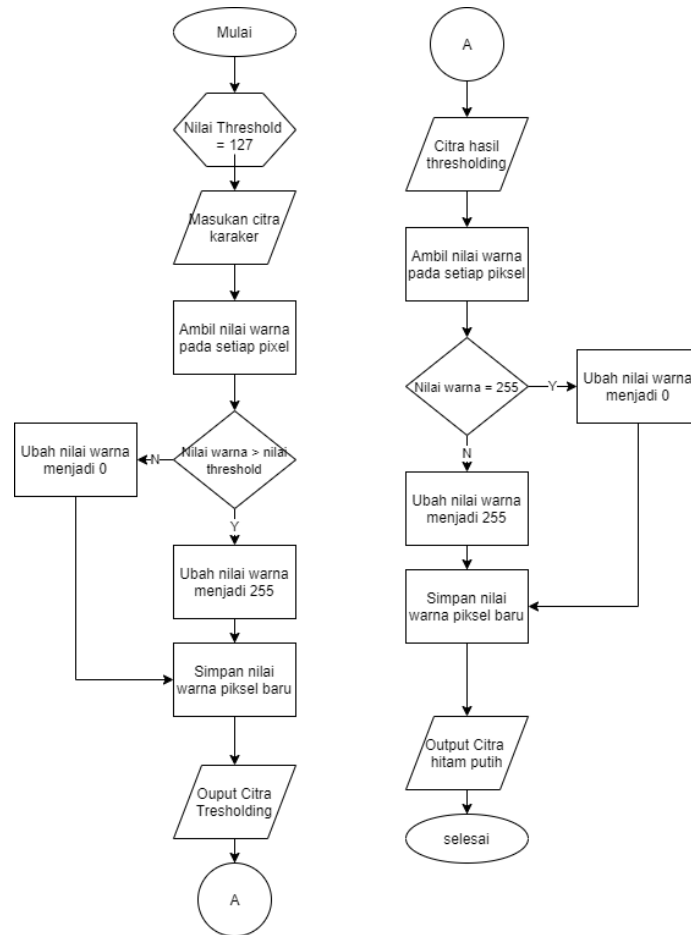


Gambar 6. Contoh Output Resize

Gambar 6. merupakan contoh *output* proses *resize* dimana citra (a) merupakan citra masukan dengan resolusi 192x192 yang kemudian akan diubah menjadi citra (b) dengan resolusi 64x64.

b. Thresholding

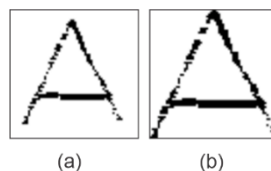
Tahap kedua dari proses *preprocessing* adalah *thresholding*. Merupakan proses untuk mengubah citra menjadi citra hitam putih atau biner, proses ini terdiri dari dua proses yaitu *thresholding* dan *invert*, *thresholding* menggunakan nilai *threshold* sebagai pembatas. Nilai *threshold* ditentukan diawal yaitu 127 dimana nilai tersebut didapat dari nilai tengah dari rentang nilai komponen warna 0-255, penggunaan nilai tengah sebagai nilai *threshold* sering juga disebut sebagai *global thresholding* dimana *global thresholding* terdiri dari pengaturan nilai intensitas (*threshold*) sedemikian rupa sehingga semua piksel yang memiliki nilai intensitas dibawah *threshold* memiliki satu nilai, dan yang berada diatasnya juga memiliki satu nilai. *Flowchart* dari proses ini ditunjukkan pada gambar 7.



Gambar 7. Flowchart Thresholding

c. Cropping

Cropping merupakan sebuah proses untuk menghilangkan area luar yang tidak diinginkan pada sebuah citra, proses ini terdiri dari penghapusan beberapa area peripheral dari suatu citra untuk menghilangkan area yang tidak diinginkan dari citra, proses ini bisanya digunakan untuk meningkatkan frame citra, mengubah aspect ratio, atau mengisolasi object dari backgroundnya.



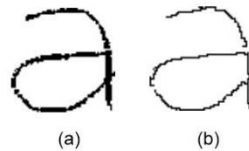
Gambar 8. Contoh Output Cropping

Gambar 8 merupakan contoh *output* dari proses *cropping* dimana citra (a) merupakan citra awal yang kemudian akan diproses sehingga menjadi citra (b) dengan cara menghilangkan area kosong pada setiap tepi citra (a).

d. Thinning

Berdasarkan pada kondisi yang telah tertera serta dijabarkan sebelumnya pada sub Bab 1.2, proses *thinning* menggunakan algoritma zhang-suen dibagi menjadi 2 tahap dimana tahap pertama akan mencari semua piksel hitam yang memenuhi semua atau kelima kondisi yang telah disebutkan dan kemudian mengubah piksel tersebut menjadi

piksel putih proses pada tahap ini akan terus berjalan sampai tidak ada piksel hitam yang memenuhi semua kondisi tersebut yang kemudian akan masuk ke tahap 2 dimana pada tahap ini citra akan di scan kembali untuk menemukan piksel hitam yang memenuhi semua atau kelima kondisi tahap 2 dan mengubah piksel yang memenuhi semua kondisi tersebut menjadi piksel putih lagi, proses ini akan terus berlangsung sampai tidak ada piksel hitam yang memenuhi kondisi tahap 2, setelah proses *thinning* tahap 2 berhenti maka akan didapatkan *output* berupa kerangka atau hasil citra masukan yang telah di tipiskan.

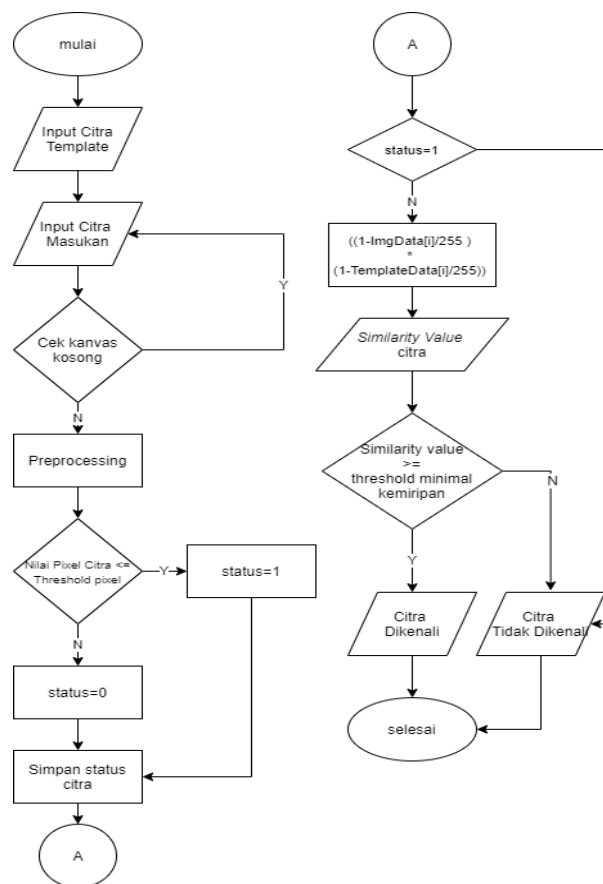


Gambar 9. Contoh Output Thinning

Gambar 9 merupakan contoh output dari proses *thinning* dimana citra (a) merupakan citra awal yang kemudian akan diproses sehingga menjadi citra (b) yang merupakan kerangka dari citra (a).

2.4. Template Matching

Proses *Template Matching* dijalankan pada aplikasi ini ditambahkan beberapa modifikasi berupa pengecekan *threshold* yang harus terpenuhi oleh citra sebelum masuk kedalam tahap *Template Matching*. *Flowchart* dari proses ini ditunjukkan pada gambar 10.



Gambar 10. Flowchart Template Matching

Flowchart pada gambar 10 menunjukkan proses dimulai dari input citra masukan oleh pengguna, citra masukan tersebut kemudian akan masuk ke proses *preprocessing* yang kemudian akan dihitung tingkat kemiripan citra masukan dengan citra *template* dengan menggunakan *template matching* dengan pendekatan *cross correlation*, dimana pada penelitian ini untuk menyatakan apakah citra tersebut mirip, *similarity value* yang didapatkan dari hasil *template matching* citra masukan tersebut harus melewati *threshold* yang diberikan, dimana *threshold* tersebut merupakan *threshold* minimal kemiripan, dimana nilai *threshold* tersebut diambil dari persentase nilai piksel karakter tersebut, sebagai contoh nilai *threshold* minimal kemiripan untuk karakter alfabet A adalah 44%, jadi dapat dikatakan citra masukan karakter A harus memiliki tingkat kemiripan sebesar 44% untuk dinyatakan sebagai karakter A.

2.4.1. Pengujian Akurasi

Pengujian sistem yang dilakukan pada penelitian ini merupakan pengujian akurasi dari pengenalan tulisan karakter yang dilakukan menggunakan metode *template matching*, dimana pada penelitian ini menggunakan hasil masukan tulisan karakter dalam bentuk citra *digital* sebagai data uji yang diambil dari 5 orang responden anak – anak dengan kriteria sebagai berikut :

1. Anak berada di rentang umur 3-5 tahun atau anak yang masuk jenjang taman kanak – kanak (TK).
2. Anak belum atau baru belajar mengenal karakter baik huruf alfabet latin (A-Z), (a-z), maupun angka (0-9).

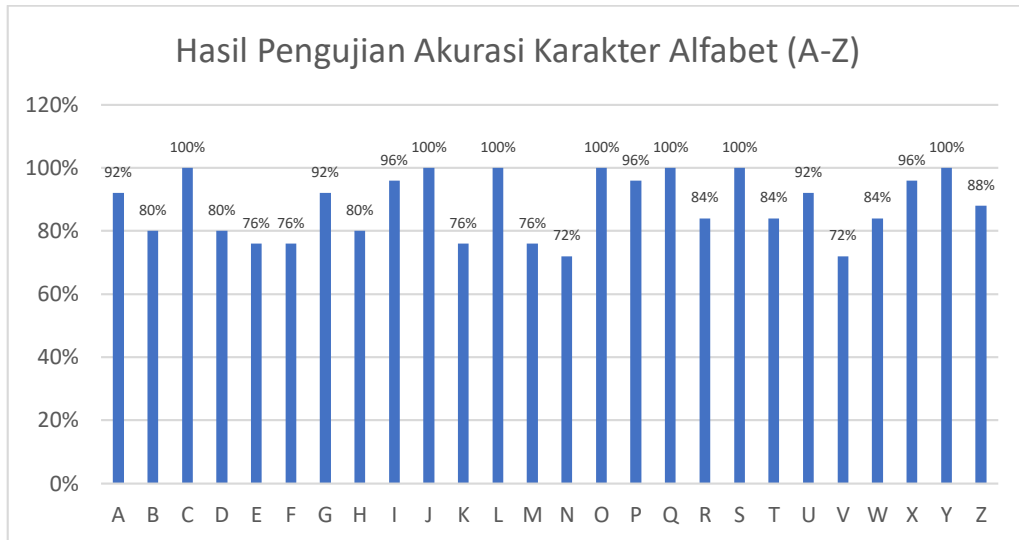
Masing – masing responden anak kemudian akan membuat 5 citra karakter untuk setiap karakter dalam kelompok huruf alfabet latin (A-Z), (a-z), serta angka (0-9) sehingga didapatkan sebanyak 25 citra karakter untuk setiap karakter baik huruf kapital (A-Z) maupun kecil (a-z) serta angka (0-9) sehingga didapatkan total 1550 data citra karakter, yang terdiri dari 650 data citra karakter huruf alfabet latin (A-Z) , 650 data citra karakter huruf alfabet latin (a-z) dan 250 data karakter karakter angka, data citra karakter yang didapatkan kemudian akan di cocokkan dengan citra *template*.

Aplikasi dapat dikatakan berhasil dalam mengenali karakter huruf (A-Z) kecil atau kapital maupun angka (0-9) jika hasil akurasi yang didapatkan sama dengan atau lebih dari 60% hal ini didasari dari hasil yang didapatkan pada penelitian yang dilakukan oleh (Hossain & Afrin, 2019) dimana didapatkan hasil dengan rentang dari 60% - 100% dari 7 font karakter yang digunakan untuk kasus 10 karakter *typewritten* serta hasil yang didapatkan pada penelitian yang dilakukan oleh (Kumar & Sharma, 2013) yang mendapatkan akurasi dengan rentang 50-60% untuk pencocokan karakter *handwritten* dengan *template* karakter *typewritten* dan pencocokan karakter *handwritten* dengan *template* karakter *handwritten*.

3. Hasil dan Pembahasan

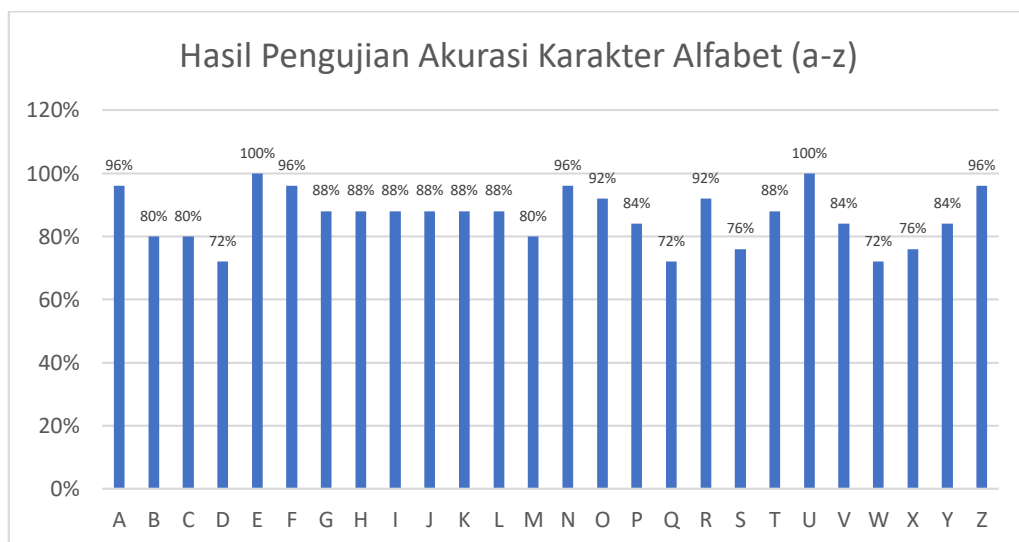
3.1. Hasil Pengujian Akurasi

Pengujian akan dilakukan untuk masing – masing karakter pada kelompok karakter huruf alfabet (A-Z), (a-z), dan angka (0-9), dimana dari ketiga kelompok tersebut akan didapatkan 1550 data *similarity value* yang didapatkan dari hasil *Template Matching* yang diambil, berikut merupakan hasil yang didapatkan untuk masing – masing kelompok karakter :



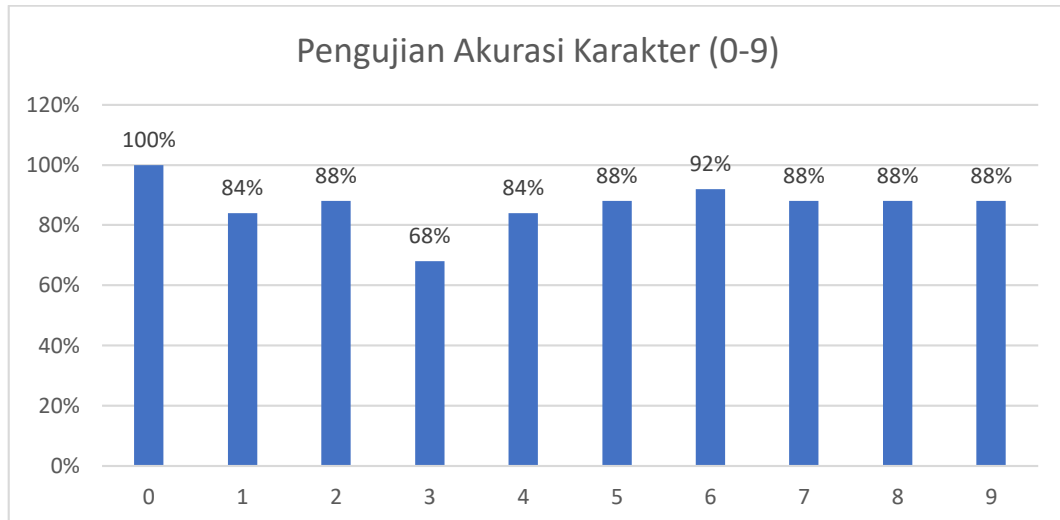
Gambar 11. Grafik Hasil Pengujian Akurasi Karakter Alfabet (A-Z)

Gambar 11 merupakan grafik detail hasil pengujian akurasi untuk karakter huruf alfabet (A-Z) dimana didapatkan dari 650 data *similarity value* (25 data *similarity value* untuk masing -masing karakter) diperoleh rata – rata akurasi ketepatan sebesar 88% dengan tingkat akurasi tertinggi didapatkan pada karakter C, J, L, O, Q, S dan Y sebesar 100% serta tingkat akurasi terendah yaitu karakter N dan V dengan tingkat akurasi 72%.



Gambar 12. Grafik Hasil Pengujian Akurasi Karakter Alfabet (a-z)

Gambar 12 merupakan grafik detail hasil pengujian akurasi untuk karakter huruf alfabet (a-z) dimana didapatkan dari 650 data *similarity value* yang didapatkan (25 *data similarity value* untuk masing -masing karakter) diperoleh rata – rata akurasi ketepatan sebesar 86% dengan tingkat akurasi tertinggi didapatkan pada karakter E dan U sebesar 100% serta tingkat akurasi terendah yaitu karakter D, Q, dan W dengan tingkat akurasi 72%



Gambar 13. Grafik Hasil Pengujian Akurasi Karakter angka (0-9)

Gambar 13 merupakan grafik detail hasil pengujian akurasi untuk karakter angka (0-9) dimana didapatkan dari 250 data *similarity value* yang didapatkan (25 data *similarity value* untuk masing -masing karakter) diperoleh rata – rata akurasi ketepatan sebesar 87% dengan tingkat akurasi tertinggi didapatkan pada karakter 0 sebesar 100% serta tingkat akurasi terendah yaitu karakter 3 dengan tingkat akurasi 68%.

Dari hasil yang didapatkan secara keseluruhan dari pengujian akurasi *Template Matching* pada kasus pengenalan karakter tulisan karakter pada game edukasi menulis “Ayo Menulis” mendapatkan tingkat akurasi rata – rata 88% untuk karakter huruf alfabet (A-Z), 86% untuk karakter huruf alfabet (a-z) dan 87% untuk karakter angka (0-9), dimana nilai *threshold* minimal kemiripan yang digunakan memiliki peran yang penting dalam menentukan tingkat akurasi dari pengenalan karakter yang didapatkan pada penelitian ini.

4. Kesimpulan

4.1. Kesimpulan

Adapun kesimpulan yang dapat diperoleh dari hasil perancangan yang telah dilakukan adalah :

1. Dari permasalahan yang telah diuraikan sebelumnya, dapat disimpulkan bahwa metode *template matching* dapat digunakan dalam pengembangan sebuah aplikasi game edukasi menulis untuk anak usia dini.
2. Akurasi pengenalan karakter tulisan karakter yang dibuat anak pada aplikasi game edukasi “Ayo Menulis” dengan metode *Template Matching*, secara keseluruhan telah melewati tingkat akurasi yang diharapkan yaitu 60%, dengan rata – rata tingkat akurasi pengenalan karakter untuk karakter huruf alfabet latin (A-Z) sebesar 88%, karakter huruf alfabet (a-z) sebesar 86%, dan karakter angka arab (0-9) sebesar 87%, dimana nilai parameter *threshold* minimal kemiripan memiliki peran penting dalam mendapatkan tingkat akurasi tersebut, sehingga pengenalan karakter dengan metode *Template Matching* dapat dinyatakan telah berhasil dan dapat digunakan dalam pengenalan karakter tulisan karakter yang dibuat oleh anak pada aplikasi game edukasi “Ayo Menulis”.

4.2. Saran

Saran yang dapat diberikan oleh penulis adalah dalam pengembangan selanjutnya adalah, untuk dapat meningkatkan akurasi dalam pengenalan karakter alfabet huruf (A-Z), (a-z) maupun angka (0-9) sehingga aplikasi dapat mengenali karakter yang ditulis oleh pengguna dengan lebih baik dan tepat.

Daftar Pustaka

- Baek, G., & Kim, S. (2009). Two step template matching method with correlation coefficient and genetic algorithm. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5755 LNAI, 85–90.
https://doi.org/10.1007/978-3-642-04020-7_10
- Gerde, H. K., Bingham, G. E., & Wasik, B. A. (2012). Writing in Early Childhood Classrooms: Guidance for Best Practices. *Early Childhood Education Journal*, 40(6), 351–359.
<https://doi.org/10.1007/s10643-012-0531-z>
- Hossain, M. A., & Afrin, S. (2019). Optical Character Recognition based on Template Matching. *Global Journal of Computer Science and Technology*, 19(2), 31–35.
<https://doi.org/10.34257/gjcstcvol19is2pg31>
- Kumar, S., & Sharma, P. (2013). Offline Handwritten & Typewritten Character Recognition using Template Matching. *Ijcset*, 4(06), 818–825. Retrieved from
<http://www.ijcset.com/docs/IJCSET13-04-06-085.pdf>
- Rahmadwati, R., & Wijono, W. (2016). Thinning Zhang-Suen dan Stentiford untuk Menentukan Ekstraksi Ciri (Minutiae) Sebagai Identifikasi Pola Sidik Jari. *Majalah Ilmiah Teknologi Elektro*, 15(2), 127–133. <https://doi.org/10.24843/mite.1502.19>
- Solin, M. C., Ginting, G., & Sirati, M. J. F. (2019). Penerapan Metode Template Matching Pada Citra Berwarna. 18, 49–51.