

# Pengenalan Angka Pada Sistem Operasi Android Dengan Menggunakan Metode *Template Matching*

Abdi Pandu Kusuma<sup>1</sup> dan Darmanto<sup>2</sup>

<sup>1,2</sup>Teknik Informatika Fakultas Teknologi Informasi Universitas Islam Balitar, Jl. Majapahit 4A, Blitar

E-mail: <sup>1</sup>[pans.uib1blitar@gmail.com](mailto:pans.uib1blitar@gmail.com), <sup>2</sup>[darmanto.bravo58@gmail.com](mailto:darmanto.bravo58@gmail.com)

## Abstrak

Usia dini merupakan usia yang efektif untuk mengembangkan berbagai potensi yang dimiliki oleh anak. Upaya pengembangan potensi dapat dilakukan melalui berbagai cara termasuk dengan cara bermain. Bermain bagi anak merupakan cara yang tepat untuk belajar. Berdasarkan fenomena tersebut, maka perlu dibuat sebuah aplikasi pengenalan angka yang interaktif dengan unsur edukasi. Aplikasi tersebut diharapkan dapat mengambil keputusan secara otomatis apa yang ditulis anak itu bernilai benar atau salah dan juga dapat membangkitkan semangat belajar anak dalam mengenal pola angka. Solusi yang sesuai agar aplikasi tersebut dapat memberikan jawaban salah atau benar digunakan satu metode yaitu *template matching*. Pengenalan angka dengan menggunakan metode *template matching* dilakukan dengan cara membandingkan citra masukan dengan citra *template*. Hasil *template matching* dihitung dari banyaknya titik pada citra masukan yang sesuai dengan citra *template*. *Template* disediakan pada *database* untuk memberikan contoh cara penulisan pola angka. Uji coba dilakukan pada aplikasi sebanyak 40 kali dengan pola yang berbeda. Dari hasil uji coba didapat prosentase keberhasilan aplikasi ini mencapai 75,75%.

**Kata kunci:** Belajar, bermain, *Template Matching*, dan pola.

## Abstract

*Early childhood is an effective age to develop the potential of the child. Potential development efforts can be done through various ways, including by playing. Playing for children is a great way to learn. Based on this phenomenon, it should be made an introduction to the numbers interactive application with elements of education. The application is expected to take decisions automatically what the child is written is true or false, and also can encourage a child's learning in recognizing number patterns. Appropriate solutions so that the app can give an answer right or wrong to use the methods that template matching. The introduction of the numbers by using template matching is done by comparing the input image with the image of the template. Template matching results are calculated from the number of points in the input image corresponding to the image of the template. Templates are provided in the database to provide an example of how to write a pattern of numbers. Tests performed on the application as much as 40 times with different patterns. From the test results obtained percentage of success of these applications reached 75.75%.*

**Key word:** *Early age, playing, study, Template Matching.*

## 1. Pendahuluan

Kemampuan untuk menulis menurut Hendrawan, dkk (2012), merupakan ketentuan utama untuk anak-anak agar dapat memahami pelajaran yang diberikan di sekolah. Kemampuan mengenal lambang bilangan pada anak sangat penting dikembangkan guna memperoleh kesiapan dalam mengikuti pembelajaran di tingkat yang lebih tinggi khususnya dalam penguasaan konsep matematika. Salah satu alternatif untuk membantu meningkatkan minat dan kemampuan anak-anak dalam belajar membaca dan menulis menggunakan media pembelajaran dalam bentuk perangkat lunak. Android adalah alternatif *software* yang dipilih, alasannya Android cenderung digunakan pada perangkat *mobile* seperti *Smartphone* dan *tablet*, *Personal Computer* (PC), selain kemudahan Android dalam pengembangan juga karena mudahnya untuk mendapatkan sumber dan tutorial serta menjadi salah satu alasan Android itu dipilih sebagai media untuk membangun aplikasi pengenalan angka ini (Pranata, 2016).

Pentingnya suatu media perangkat lunak berupa aplikasi yang memberikan pembelajaran menulis kepada anak dengan berbasis *mobile* dan media layar sentuh, yang digunakan sebagai sarana pembelajaran melalui interaksi sentuhan jari sehingga anak dapat merasa senang serta tidak bosan untuk belajar menulis dikarenakan mereka merasa dalam suatu permainan. Sedangkan perangkat *mobile*

nantinya dapat mempermudah pengajaran untuk penyampaian pembelajaran menulis tanpa terkait oleh target waktu dan tempat, karena dengan aplikasi yang berbasis *mobile* materi dapat dilakukan di mana saja.

Hasil penelitian ini adalah sebuah aplikasi yang menyajikan studi menulis dengan dikemas seperti bermain *game*. Metode *Template matching* adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template* (acuan) (Wardhana dan Prayudi, 2008). Metode *Template Matching* dalam penelitian ini digunakan sebagai alat untuk membantu melakukan validasi bagian-bagian gambar untuk diolah menghasilkan *output* yang akan diterima oleh anak pada perangkat *mobile*. Sementara dalam tahap pengujian aplikasi melalui pengujian kuantitatif dan kualitatif telah menunjukkan respon positif dari responden untuk menggunakan aplikasi ini sebagai media alternatif belajar menulis.

## 2. Metode Penelitian

Tahapan dalam metode penelitian ini meliputi analisis sistem, analisis data, dan perancangan sistem, diantaranya:

### 2.1. Analisis sistem

Sistem yang dibangun mempunyai spesifikasi kemampuan sebagai berikut:

1. Memberikan soal angka secara *random*.
2. Melakukan pembacaan karakter angka.
3. Mengubah masukkan *canvas* menjadi *bitmap*.
4. Mengkonversikan citra *Red Green Blue* (RBG) 24 bit menjadi citra biner 1 bit.
5. Mencocokkan citra masukkan sesuai dengan *template*.

Sistem ini ditujukan untuk digunakan oleh anak usia dini dalam proses pembelajaran pada pengenalan angka, dalam peningkatan belajar yang lebih efektif dengan memanfaatkan teknologi yang telah ada.

### 2.2. Analisis data

Data pengujian yang digunakan dalam penelitian ini dibagi menjadi tiga bagian utama, yaitu data masukan, data yang digunakan selama proses pencocokan *template matching*, dan data keluaran berupa hasil dari pengolahan *template matching*, digunakan *template* angka 0 sampai 9 untuk proses perhitungan agar didapatkan keluaran, seperti tertera pada Tabel 1.

Nomor	Angka	Pola Angka
1	0	<b>0</b>
2	1	<b>1</b>
3	2	<b>2</b>
4	3	<b>3</b>
5	4	<b>4</b>
6	5	<b>5</b>
7	6	<b>6</b>
8	7	<b>7</b>
9	8	<b>8</b>
10	9	<b>9</b>

Pada Tabel 2 ditunjukkan beberapa variasi *template* gambar angka yang menjadi citra masukan. Untuk data soal berupa gambar binatang dan gambar buah, digunakan agar tampilan lebih menarik dan memberikan nilai pembelajaran dalam proses berhitung ditunjukkan Tabel 3. Pada Tabel 4 terdapat data suara yang digunakan agar *user* lebih bersemangat dalam menggunakan aplikasi pengenalan pola angka. Data masukan yang pertama dari pengguna adalah arsip citra yang dipilih oleh pengguna.

### 2.3. Perancangan sistem

Sistem yang dibangun ini adalah pengenalan pola angka yang diakses melalui piranti bergerak yang di dalamnya diimplementasikan metode *template matching* untuk optimasi pencocokan karakter angka. Tujuannya adalah untuk mendapatkan hasil pencocokan antara *template* terhadap masukan gambar. Terdapat dua tahapan dalam pembuatan aplikasi, yaitu *preprocessing* dan analisa.







Tabel 2 Beberapa variasi *template* gambar angka

Nomor	Angka	Template Gambar	
1	38	<b>3</b>	<b>8</b>
2	15	<b>1</b>	<b>5</b>
3	42	<b>4</b>	<b>2</b>
4	20	<b>2</b>	<b>0</b>
5	16	<b>1</b>	<b>6</b>
6	27	<b>2</b>	<b>7</b>
7	44	<b>4</b>	<b>4</b>
8	34	<b>3</b>	<b>4</b>
9	35	<b>3</b>	<b>5</b>
10	19	<b>1</b>	<b>9</b>
11	48	<b>4</b>	<b>8</b>
12	52	<b>5</b>	<b>2</b>
13	17	<b>1</b>	<b>7</b>
14	24	<b>2</b>	<b>4</b>
15	7	<b>0</b>	<b>7</b>
16	59	<b>5</b>	<b>9</b>

### 2.4. Perancangan sistem

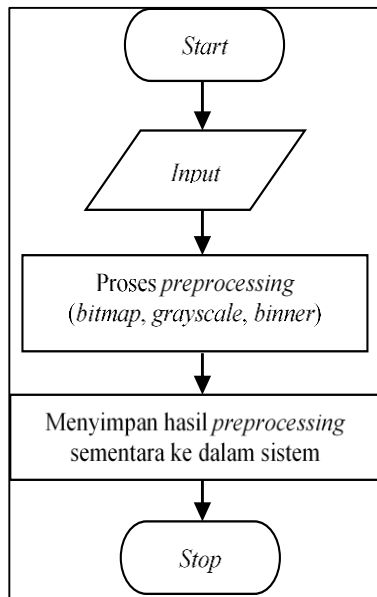
Sistem yang dibangun ini adalah pengenalan pola angka yang diakses melalui piranti bergerak yang di dalamnya diimplementasikan metode *template matching* untuk optimasi pencocokan karakter angka. Tujuannya adalah untuk mendapatkan hasil pencocokan antara *template* terhadap masukan gambar. Terdapat dua tahapan dalam pembuatan aplikasi, yaitu *preprocessing* dan analisa.

Tabel 3 Data uji soal

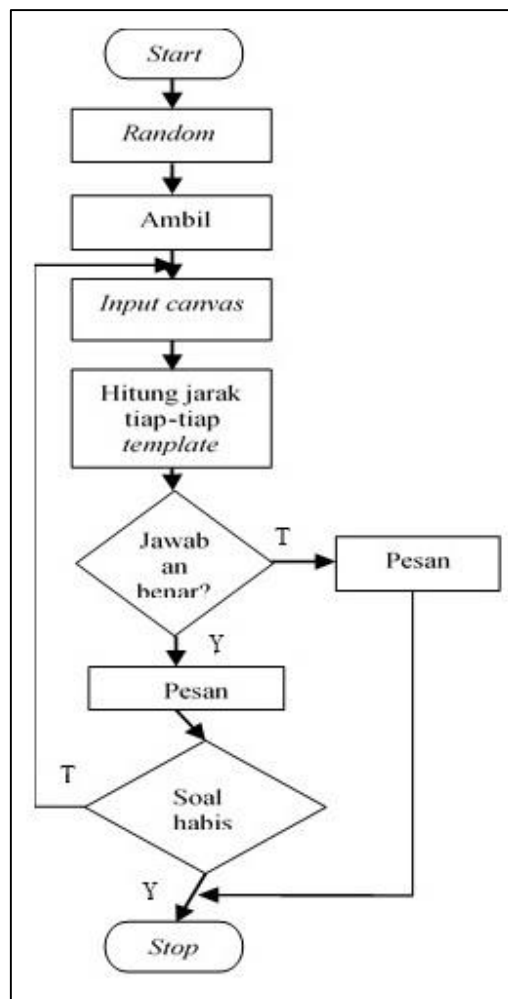
Nomor	Gambar	Keterangan	Sumber Gambar
1		Pisang	(Zanana, 2015)
2		Kupu-Kupu	(Fammy, 2012)
3		Jeruk	(Nugraha, 2012)
4		Kelinci	(Auliani, 2011)
5		Apel	(Yana, 2014)
6		Kucing	(Indonesia, 2013)

Tabel 4 Data suara

Nomor	Data	Tipe Data	Ukuran Data	Keterangan
1	<i>Play</i>	.mp3	947 MB	Data digunakan ketika aplikasi pertama kali dijalankan
2	<i>Main</i>	.mp3	0.98 MB	Data digunakan pada saat proses pembelajaran
3	<i>Applause</i>	.mp3	194 KB	Data dijalankan apabila jawaban dari soal bernilai benar
4	<i>Stop</i>	.mp3	27.5 KB	Data dipanggil pada saat jawaban soal merupakan jawaban yang salah
5	<i>Play</i>	.mp3	947 MB	Data digunakan ketika aplikasi pertama kali dijalankan
6	<i>Main</i>	.mp3	0.98 MB	Data digunakan pada saat proses pembelajaran



Gambar 1 Alur proses *preprocessing*



Gambar 2 Diagram alir analisis konten

Seperti ditunjukkan pada Gambar 1 dalam tahap *preprocessing*, proses yang dilakukan adalah:

1. *User* memasukkan gambar ke dalam perangkat bergerak.

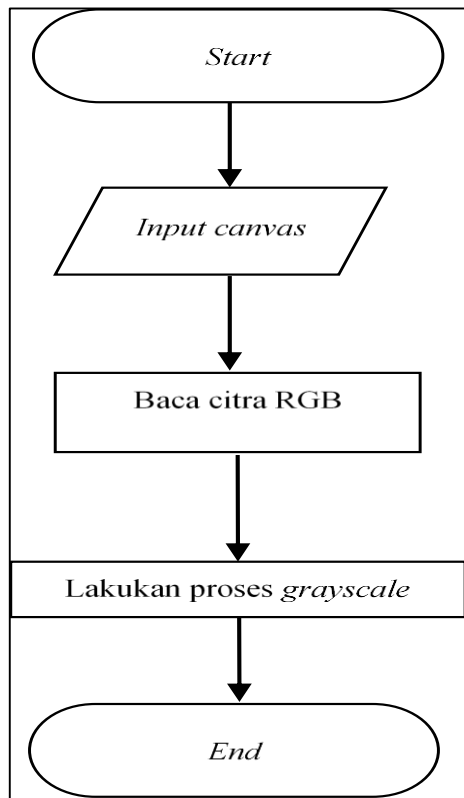
2. Sistem akan melakukan pemrosesan terhadap gambar tersebut, yaitu *bitmap*, *grayscale*, biner.
3. Hasil proses tersebut, kemudian disimpan sementara dalam sistem.

Pada Gambar 2, diagram alir analisis konten menggambarkan tentang proses berjalannya sistem yang akan memberikan soal pertanyaan kepada *user* secara acak, kemudian memunculkan *template* angka yang dilakukan secara transparan untuk mendapatkan jawaban sesuai dengan *template*. Proses selanjutnya, pada saat soal berjalan kemudian *user* memasukkan jawaban berdasarkan soal yang telah ditentukan, kemudian akan dilakukan proses perhitungan antara masukkan dengan *template* untuk mengetahui apakah jawaban tersebut benar atau salah.

Perancangan sistem pada pengenalan angka terdapat beberapa tahapan yang harus dilakukan agar aplikasi dapat memberikan hasil yang sesuai dengan metode *template matching*. Adapun tahapan-tahapan dalam perancangan sistem diantaranya:

1. Proses *input canvas*

Pada proses *input canvas* adalah merubah tulisan *user* ke dalam bentuk gambar atau *bitmap* agar bisa dibaca oleh sistem, kemudian setelah terbaca menjadi gambar dilakukan proses pembacaan RGB untuk membedakan antara tiga warna *red*, *green* dan *blue*. Kemudian akan diproses ke bentuk *grayscale* untuk mendapatkan hasil gambar yang berwarna abu-abu. Seperti yang ditunjukkan pada Gambar 3.



Gambar 3 Diagram alir proses *input canvas*

2. Proses *grayscale*

Pada pengubahan sebuah gambar menjadi *grayscale* dapat dilakukan dengan cara mengambil semua piksel pada gambar kemudian warna tiap piksel akan diambil informasi mengenai 3 warna dasar yaitu merah, biru dan hijau (melalui fungsi warna RGB), ketiga warna dasar ini akan dijumlahkan, dengan mengalikan tiap-tiap warna dengan angka yang berbeda, yaitu  $R \times 0,3$ ,  $G \times 0,59$  dan  $B \times 0,11$  sehingga didapat nilai rata-rata. Nilai rata-rata inilah yang akan dipakai untuk memberikan warna pada piksel gambar sehingga menjadi warna *grayscale*, tiga warna dasar dari sebuah piksel akan diset menjadi nilai rata-rata (melalui fungsi RBG warna). Proses *grayscale* ditunjukkan pada Gambar 4.

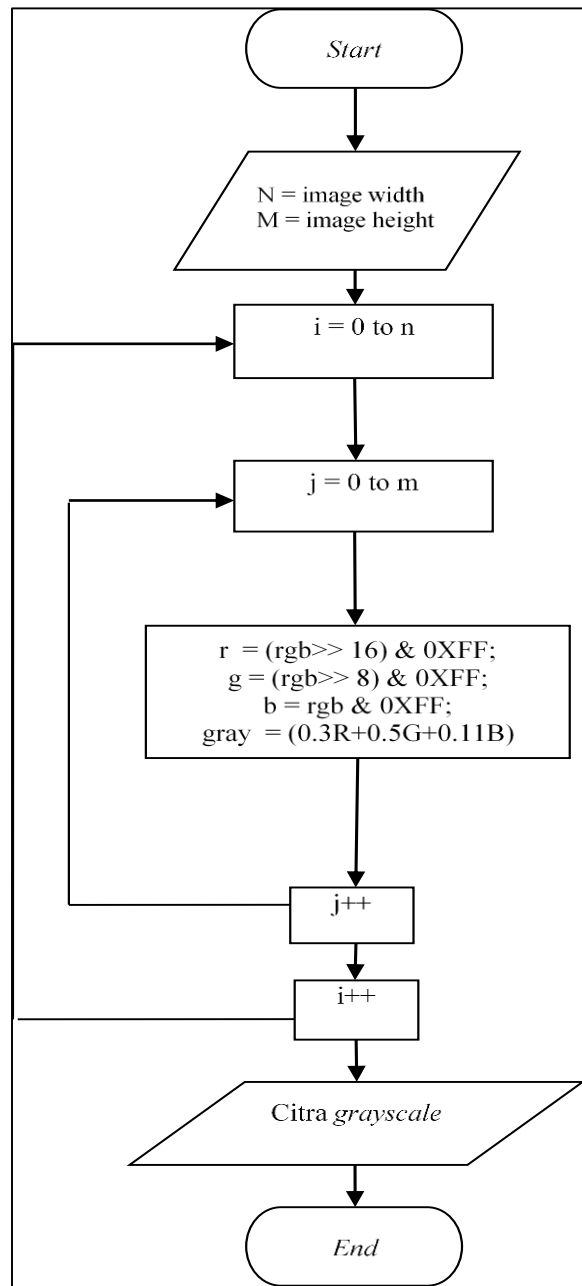
3. Proses binerisasi

Pengkonversian citra *true color* (24 bit) menjadi citra biner (1 bit) dilakukan dengan operasi pengambangan (*thresholding*). Operasi pengambangan mengelompokkan nilai derajat keabuan setiap piksel ke dalam dua kelas, hitam dan putih. Di mana hitam sebagai warna objek dan putih sebagai warna latar belakangnya.

Pemilihan nilai *threshold* yang dilakukan berpengaruh terhadap ketajaman suatu citra. Biasanya rentangan nilai *grayscale* yang sering dilakukan adalah antara 0 sampai 255. Pada penelitian ini, nilai *threshold* yang digunakan adalah 50. Proses pengubahan ini dilakukan dengan memeriksa piksel citra. Jika piksel kurang dari 50 maka warna piksel berubah menjadi hitam, begitu juga sebaliknya jika piksel lebih dari 50 maka warna piksel berubah menjadi putih. *Flowchart* diagram *grayscale* ditunjukkan pada Gambar 5.

4. Proses *template*

Pada tahap proses *template* terdapat perhitungan antara *input* gambar dengan *template* gambar, di mana dalam perhitungannya mencari perbedaan untuk mencocokkan satu persatu agar didapatkan nilai yang sesuai dengan *template* gambar. Proses ini ditunjukkan pada Gambar 6.



Gambar 4 Proses *grayscale*

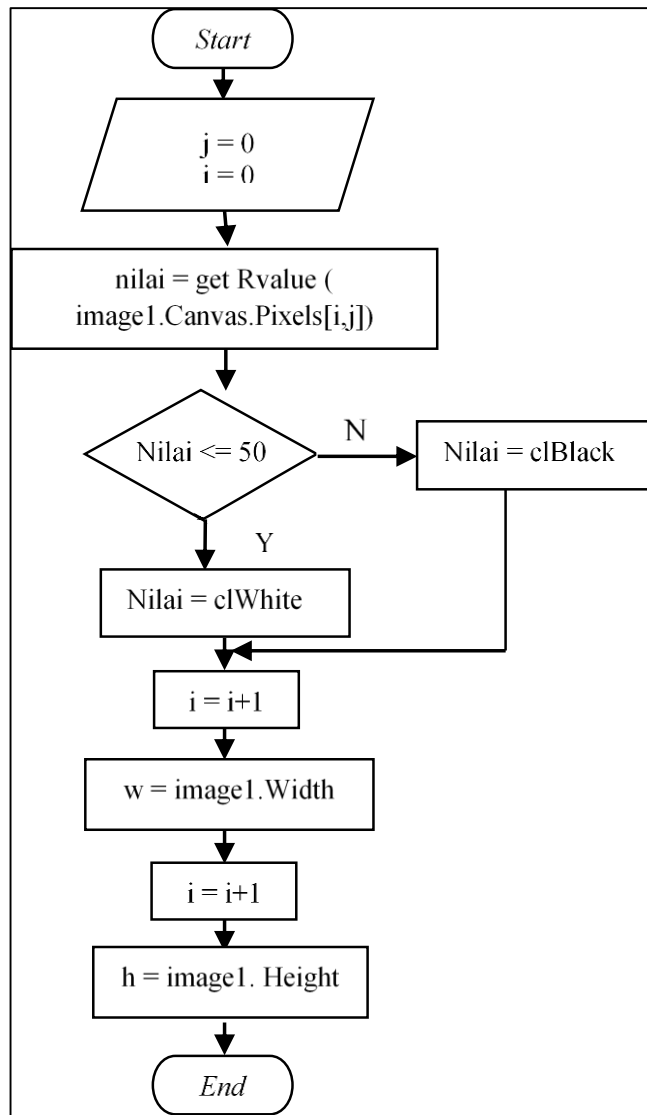
### 3. Hasil Penelitian dan Pembahasan

Perhitungan *Manual Template Matching* akan dijabarkan berdasarkan proses-proses dalam *template matching*, dengan menggunakan rumus SAD (*Sum Of Absolute Difference*) yaitu mencari jumlah perbedaan terendah dari penjumlahan, dengan melakukan perhitungan satu persatu untuk mendapatkan hasil mutlak. Pada contoh perhitungan akan menghitung nilai pada angka 1 antara matrik A (*template* angka) dengan matrik B (angka yang dituliskan), hasil yang diambil untuk memberikan kesimpulan yang menyatakan bahwa matrik A dan matrik B adalah sama dengan mencari nilai perbedaan terendah antara matik A dengan matrik B.

Pembuktian dari matrik A dengan matrik B memiliki nilai yang sama untuk matriks A berupa angka 0-9 dan matriks B berupa angka yang dituliskan yakni dapat dijelaskan dengan beberapa langkah berikut:

- Langkah pertama menghitung piksel di koordinat  $x, y$  pada matrik A (*pola 0*) dengan matrik B (angka 1) seperti ditunjukkan pada Gambar 7. Jumlah perhitungan antara matrik A (*pola 0*) dengan matrik B (angka 1), yakni:

$$(0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (1 - 0)^2 + (0 - 1)^2 + (1 - 0)^2 + (1 - 0)^2 + (0 - 1)^2 + (1 - 0)^2 + (1 - 0)^2 + (0 - 1)^2 + (1 - 0)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 = 9$$



Gambar 5 Proses binerisasi

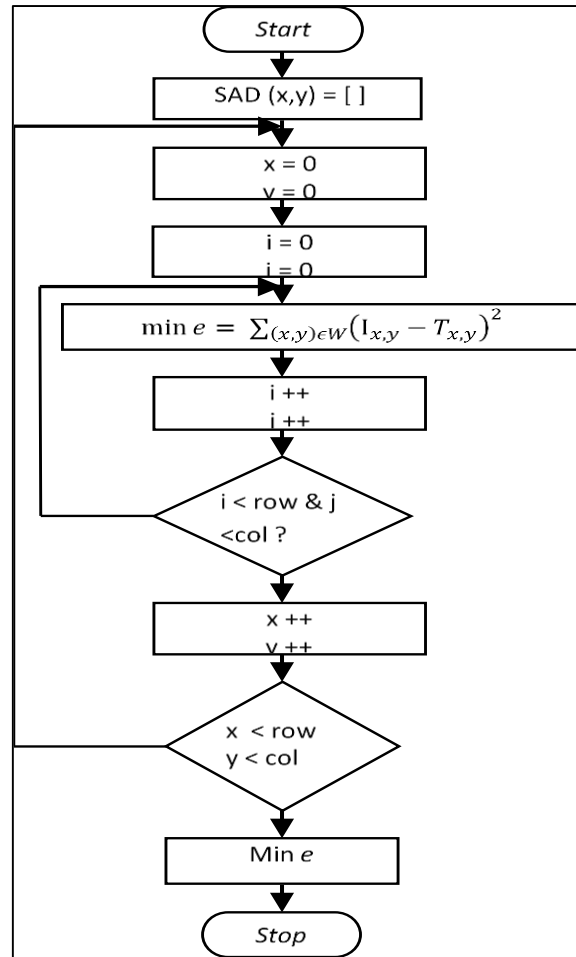
- Langkah kedua menghitung piksel di koordinat  $x, y$  pada matrik A (*pola 1*) dengan matrik B (angka 1) seperti ditunjukkan pada Gambar 8. Jumlah perhitungan antara matrik A (*pola 1*)



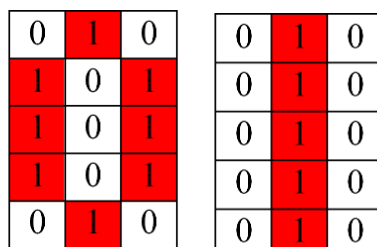
dengan matrik B (angka 1), yakni:

$$(0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 = 0$$

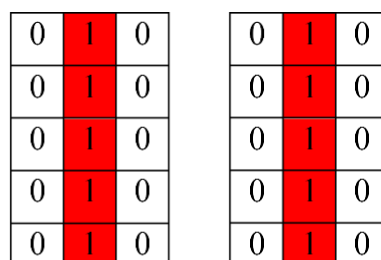
Pembuktian dilakukan hingga pada pola 9 untuk matriks A. Adapun tabel hasil pembuktian hingga pola ke 9 dapat ditunjukkan seperti Tabel 5.



Gambar 6 Diagram alir perhitungan *template*



Gambar 7 Pola 0 dan angka 1



Gambar 8 Pola 1 dan angka 1

Tabel 5 Data suara

Nomor	Pola Matriks A	Pola Matriks B	Jumlah Perhitungan
1	0	1	9
2	1	1	0
3	2	1	7
4	3	1	7
5	4	1	10
6	5	1	10
7	6	1	8
8	7	1	6
9	8	1	12
10	9	1	11



Gambar 9 Tampilan utama

### 3.1. Implementasi Tampilan Utama

Menu utama di dalam aplikasi pengenalan pola angka merupakan menu yang pertama kali terlihat pada saat aplikasi dijalankan. Menu utama pengenalan pola angka bisa dilihat pada Gambar 9.

### 3.2. Hasil Pengujian Sistem

Pada hasil pengujian dilakukan dengan membandingkan antara *input* angka terhadap *template* (Gonzales & Woods, 1992) angka sebanyak 40 *image*/gambar. Pengujian dilakukan untuk mengetahui keakuratan *output* program, dan untuk menghitung prosentase keberhasilan aplikasi yang telah dibangun. Untuk perhitungan nilai relevan menggunakan Persamaan 1:

$$Prosentase = \frac{\text{data keberhasilan}}{\text{Jumlah Data Pengujian}} \times 100\% \tag{1}$$

Berdasarkan hasil pengujian metode *template matching* pada aplikasi sebanyak 40 kali didapatkan hasil sebagai berikut:

- a) Nilai prosentase sebesar 60% terdapat pada angka 34, 48, 29 dan 18.
- b) Nilai prosentase sebesar 70% terdapat pada angka 42, 16, 9, 35, 19, 59, 28, 36, 43 dan 39.
- c) Nilai prosentase sebesar 80% terdapat pada angka 38, 15, 44, 6, 52, 13, 40, 4, 57, 23, 5, 22, 14, 3, 33 dan 8.
- d) Nilai prosentase sebesar 90% terdapat pada angka 1, 20, 27, 17, 24, 7, 12, 30 dan 21.
- e) Nilai prosentase tidak berhasil sebesar 24,25% dan nilai keberhasilan sebesar 75,75 %.

## 4. Kesimpulan

Adapun kesimpulan yang dapat penulis jelaskan pada penelitian ini diantaranya:

- a) Aplikasi pengenalan angka dengan sistem Android dapat dikatakan berhasil berdasarkan pengujian. Dari pengujian diperoleh bahwa 100% responden dapat menggunakan aplikasi ini

dengan baik.

- b) Metode *template matching* dapat diimplementasikan pada aplikasi pengenalan angka pada sistem operasi Android dengan hasil pengujian 75,75% *template matching* dapat dikenali dan 24,25% *template matching* tidak dikenali. Pengujian dilakukan sebanyak 40 kali dengan pola angka yang berbeda.
- c) Metode *template matching* membutuhkan *database template* angka yang banyak, semakin banyak *template* yang disediakan akan semakin tinggi prosentase keberhasilan mengidentifikasi pengenalan pola angka.

## 5. Referensi

- Auliani, I. (2011, Februari 19). Berternak Kelinci. Semarang, Jawa Tengah, Indonesia. Dipetik Mei 6, 2016, dari <http://rabbitica.blogspot.co.id/2011/02/kelinci.html>
- Fammy, A. A. (2012, Desember 22). Metamorfosis Kupu-Kupu. Kediri, Jawa Timur, Indonesia. Dipetik Mei 6, 2016, dari <http://fammy20.blogspot.co.id/2012/12/metamorfosis-kupu-kupu.html>
- Gonzales, R. C., & Woods, R. E. (1992). *Digital Image Processing*. Boston, US: Addison-Wesley.
- Hendrawan, H. S., Sunarto, M. J., & Sutanto, T. (2012). Aplikasi Belajar Menulis Untuk Anak Prasekolah Berbasis Android. *Jurnal JSIKA*, 1(1).
- Indonesia, N. G. (2013, Desember 17). Manusia Ternyata Sudah Pelihara Kucing Lebih dari 5.000 Tahun. Jakarta, Jakarta, Indonesia. Tratto il giorno Mei 6, 2016 da <http://nationalgeographic.co.id/berita/2013/12/manusia-ternyata-sudah-pelihara-kucing-lebih-dari-5000-tahun>
- Nugraha, A. A. (2012, September 15). JENIS-JENIS JERUK. Bandung, Jawa Barat, Indonesia. Dipetik Mei 6, 2016, dari <https://ayorange.wordpress.com/2012/09/15/jenis-jenis-jeruk-2/>
- Pranata, K. (2016). *Modul Pelatihan Qt Mobile Programming*. Malang.
- Wardhana, A. W., & Prayudi, Y. (2008). Penggunaan Metode Template Matching Untuk Identifikasi Kecacatan Pada PCB. *Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008)* (hal. J47-J50). Yogyakarta: Universitas Islam Indonesia.
- Yana, Y. (2014, Agustus 4). 33 Manfaat Apel Untuk Diet dan Untuk Kesehatan. -, -, -. Diambil kembali dari <http://manfaat.co.id/manfaat-apel>
- Zanana, K. P. (2015, Juni 7). Kandungan Gizi dan Manfaat Buah Pisang Bagi Kesehatan. Bandung, Jawa Barat, Indonesia. Dipetik Mei 6, 2016, dari <http://www.keripikpisangzanana.com/kandungan-gizi-dan-manfaat-buah-pisang-bagi-kesehatan/>