

# Pengembangan Program Bantu Pembelajaran Aksara Jawa Untuk Anak Sekolah Dasar

Elizabeth Nurmiyati Tamatjita<sup>#1</sup>, Anggraini Kusumaningrum<sup>#2</sup>

<sup>#</sup>Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Adisutjipto

Jalan Janti Blok. R, Lanud Adisutjipto, Banguntapan, Bantul, Daerah Istimewa Yogyakarta 55198

Telp. (0274) 451262

<sup>1</sup>tamatjita@gmail.com

<sup>2</sup>informatika@stta.ac.id

**Abstract - Optical character recognition (OCR) is a widely used method to recognise and transliterate printed or handwritten texts. There are also many implementations applying OCR techniques to symbols. This research implements OCR to recognise Javanese texts, to help primary school pupils to learn Javanese. This research used OCRChie algorithm which includes zoning and binary feature extraction, and Nearest Centroid Classifier (NCC)-like for classification. The algorithm is then used to classify characters in Unicode Javanese Font against trained characters obtained from scanned Javanese books. The results shown 87% accuracy of the classification, with main errors generated from a style difference between certain characters in the two fonts, which is irresolvable using features inside the OCRChie algorithm.**

**Keywords:** OCR, Aksara Jawa

Dalam penelitian ini penulis memilih satu aspek budaya yang pembelajarannya dapat dibantu oleh teknologi informasi yaitu bidang bahasa, dan berdasarkan latar belakang budaya penulis, maka dipilihlah bahasa Jawa dengan fokus pada pembelajaran penulisan aksara Jawa bagi anak sekolah dasar.

Lingkup penelitian adalah pembuatan perangkat lunak program aplikasi untuk membantu pembelajaran aksara Jawa, serta mengujikan penerapannya pada anak-anak tingkat Sekolah Dasar untuk melihat efektifitas dari program bantu ini dalam mendukung metoda belajar anak.

Penelitian ini memiliki target luaran untuk dapat menjadi awal bagi pembuatan penelitian-penelitian serupa, guna memperkenalkan anak-anak pada taraf yang dini dengan kebudayaan daerahnya, sehingga apa yang menjadi ciri budaya dapat dipegang teguh dan dilestarikan, karena budaya adalah aset bangsa Indonesia.

## I. PENDAHULUAN

Perkembangan teknologi informasi di masa kini sudah mencapai taraf yang dapat dimanfaatkan untuk segala bidang dan segala usia, dari hiburan hingga kesehatan, mulai dari kanak-kanak hingga orang dewasa. Teknologi informasi juga sudah menjangkau sebagian besar masyarakat Indonesia, dan merupakan sarana yang bisa ditemui di hampir setiap lokasi dan juga sudah tersedia dengan harga yang terjangkau.

Penelitian ini berangkat dari keprihatinan akan kondisi masyarakat saat ini, di mana karena derasnya arus globalisasi dan modernisasi yang tidak terkendali, serta kurangnya minat akan pelestarian budaya sendiri, budaya-budaya kedaerahan termasuk bahasa daerah semakin rendah tingkat penyerapannya di masyarakat, sehingga dari generasi ke generasi semakin sedikit masyarakat yang mengenal lebih dalam dan dapat mempraktekkan dengan benar budaya ibu dari daerah masing-masing.

Budaya merupakan sesuatu yang tidak bisa diajarkan secara instan untuk dapat mengenal dengan dekat dan tepat sesuai dengan pakem, sehingga harus diajarkan sejak dini.

## II. LANDASAN TEORI

Program bantu pembelajaran Aksara Jawa memerlukan beberapa teori pendukung. Sesuai dengan gambaran umum penelitian yang sudah disampaikan pada butir latar belakang, maka program aplikasi yang dibuat akan menerapkan teori-teori berikut :

### 2.1 Optical Character Recognition

Optical Character Recognition (OCR) merupakan salah satu metode pengenalan pola yang dikembangkan khusus untuk mengenali bentuk-bentuk karakter. OCR melibatkan sebuah perangkat lunak yang didesain untuk menerjemahkan citra yang berisi teks terketik, yang pada umumnya adalah hasil pemindaian, menjadi teks yang dapat diedit secara digital, yaitu bentuk *standard encoding* seperti ASCII atau Unicode.

Proses OCR membutuhkan sekelompok contoh bentuk karakter yang disebut *training characters*, untuk digunakan sebagai dasar pengenalan terhadap karakter *input*. Proses pengenalan dilakukan dengan sebuah algoritma pencocokan pola. Hasil dari proses pengenalan ini adalah persentase kemiripan antara karakter input dengan *training character*. Sebuah nilai pembatas ditentukan untuk menentukan apakah

karakter input tersebut masih dapat diinterpretasikan sebagai karakter yang sama dengan *training character*. Besar nilai pembatas ini bervariasi sesuai dengan algoritma yang digunakan.

Menurut Schulz, terdapat tujuh tahapan yang diperlukan dalam proses pengenalan karakter [1]:

#### 1. Akuisisi Citra

Dengan pemindaian, didapatkan sebuah citra digital dari dokumen asli, berupa citra bitmap *grayscale*. Terhadap citra digital ini, pertama kali diterapkan akuisisi citra yang terdiri dari tiga bagian yaitu *sampling*, kuantisasi dan *coding*.

Kuantisasi dilakukan dengan melakukan *thresholding* terhadap piksel-piksel citra *grayscale*, menjadi piksel-piksel hitam dan putih saja. Kuantisasi dapat dilakukan secara otomatis (*automated thresholding*) melalui histogram atau *adaptive thresholding*, maupun dengan *uniform thresholding*, yaitu menentukan suatu nilai konstanta sebagai pembatas intensitas piksel yang akan dikonversikan sebagai hitam atau putih.

#### 2. Segmentasi Tata Letak

Pada tahap ini dilakukan pemisahan antara unsur-unsur dokumen seperti grafik, gambar, teks dan latar belakang yang terdapat pada citra. Tahap ini menghasilkan koordinat baris-baris teks yang nanti akan diproses pada tahap berikutnya. Unsur-unsur yang tidak diperlukan dalam proses pengenalan karakter tidak dipakai, namun ditandai untuk mengingat letaknya dalam dokumen.

#### 3. Persiapan dan Transformasi Citra

Setelah dilakukan segmentasi tata letak, mungkin didapat bahwa teks pada citra ternyata dipindai dalam keadaan miring. Untuk itu tahap ini melakukan koreksi berupa transformasi geometris yang diperlukan terhadap baris-baris teks. Pada tahap ini juga dapat dilakukan *filtering* untuk menyederhanakan piksel-piksel karakter, seperti penerapan metode *thinning*, *closing* dan *opening*. Semua proses ini dilakukan untuk mempersiapkan teks agar lebih mudah saat dilakukan ekstraksi karakteristiknya.

#### 4. Ekstraksi Karakteristik

Tahap ekstraksi karakteristik adalah tahap di mana karakteristik - disebut juga properti atau *feature* - sebuah karakter diambil sebagai *identifier* karakter tersebut dalam proses pengenalan. Pada tahap inilah biasanya terdapat perbedaan pada metode yang digunakan antara satu algoritma dengan algoritma lainnya. Secara umum metode ekstraksi karakteristik ini dibagi dalam tiga golongan berdasarkan cara memperoleh *feature* karakter:

- Feature* yang didasarkan distribusi statistik piksel pada area karakter. Ini adalah teknik yang paling sederhana dan paling efisien.
- Feature* yang dihasilkan dengan melakukan transformasi terhadap karakter, seperti transformasi Fourier, Karhunen-Loeve dan Hough.
- Feature* yang didapat dari analisis struktural karakter secara geometris dan topologis, seperti banyaknya garis, kurva, ujung karakter dan sebagainya.

#### 5. Klasifikasi Simbol

Pada tahap ini, hasil ekstraksi karakteristik yang didapatkan pada tahap sebelumnya, digunakan untuk mengklasifikasikan apakah karakter yang sedang diproses, termasuk sebagai sebuah alfabet atau simbol tertentu.

#### 6. Post-correction

Tahap ini melakukan proses koreksi klasifikasi terhadap karakter-karakter yang salah diklasifikasikan.

#### 7. Document Understanding

Tahap ini melakukan analisis terhadap isi dokumen, mengenali serta menandai bagian-bagian dokumen seperti judul, pengarang, abstraksi, bab dan sebagainya, sehingga secara umum didapatkan sebuah representasi mengenai isi dokumen tersebut. Proses OCR yang digunakan pada penelitian ini hanya sampai pada tahap ke-5 yaitu klasifikasi simbol, dan dengan asumsi bahwa citra *input* yang digunakan dalam eksperimen tidak perlu mengalami transformasi untuk mengoreksi kesalahan geometris seperti pada tahap ke-3. Keberhasilan mengenali karakter merupakan kunci keakuratan sebuah algoritma OCR (Tanner, 2004:4). Hasil eksperimen menunjukkan bahwa untuk setiap algoritma pengenalan pola, dapat dipetakan sekelompok karakter yang memiliki tingkat kesalahan interpretasi tinggi. Karakter-karakter yang sering saling disalah artikan dapat disusun menjadi sebuah *confusion matrix*, sebagai petunjuk kehandalan kinerja sebuah algoritma pengenalan.

### 2.2 Confusion Matrix

*Confusion matrix* adalah sebuah alat visualisasi yang biasa digunakan dalam pembelajaran terarah (*supervised learning*), seperti pada pengenalan pola [2]. Setiap kolom matriks mewakili entitas-entitas yang berada dalam kelas terprediksi, sedangkan tiap baris mewakili entitas-entitas yang berada dalam kelas aktual.

### 2.3 Pengenalan Pola Karakter

Tamatjita dan Mahastama (2007) meneliti pengenalan pola karakter cetak latin menggunakan algoritma OCRChie menurut tiga buah *font*: Arial, Times New Roman dan Courier New, yang mana metode yang sama akan dipergunakan dalam penelitian ini. Ciri dari algoritma OCRChie adalah sebagai berikut[3]:

#### A. Learning Characters

Algoritma OCRChie bergantung pada sekumpulan *learning characters* atau sebuah kamus karakter yang akan dipergunakan sebagai acuan dalam mengenali karakter-karakter dalam citra input, *learning characters* ini terangkum dalam sebuah *learning set* [4].

Setelah *learning set* selesai dibaca dan dikenali, hasil pengenalan tersebut disimpan pada sebuah *file* terpisah dalam bentuk data numerik yang lebih ringkas tanpa harus mempertahankan citra asli yang terdapat dalam *learning set*. Bentuk penyimpanan seperti ini juga mengurangi waktu pembacaan kembali *learning set*.

**B. Ekstraksi Karakter**

Proses ekstraksi karakter meliputi beberapa tahap yaitu:

1. Pendeteksian garis pemisah dan teks

Proses ini membedakan garis pemisah (line break) dan karakter pada input, dengan tujuan menentukan posisi karakter pada baris-baris teks output dan mengambil satu baris teks untuk diproses pada langkah selanjutnya. Caranya adalah dengan menghitung jumlah piksel hitam yang terdapat pada sebaris piksel horizontal. Beberapa baris piksel yang berdekatan dan sama-sama memiliki jumlah piksel minimum akan dianggap sebuah garis pemisah. Derau mungkin akan mengacaukan proses ini, namun dapat diatasi dengan pemberian nilai toleransi jumlah piksel hitam minimum.

```
objects. For example,
rectangles; also, on a
appears to be a square
coverings appears to b
chap. 4.4-4.6, p. 105
```

```
PROBLEM S7: Analyze gi
1 captures must be mad
although you can cho
2 pawns must be promot
3 king is just another
```

Gbr. 1 Proses pendeteksian garis pemisah dan baris teks

2. Pendeteksian dan pengisolasian komponen karakter. Secara prinsip, proses ini mendeteksi secara individual setiap karakter dalam sebuah baris teks, kemudian mengisolasi tiap-tiap karakter menjadi sebuah bagian yang berbentuk segi empat. Cara yang paling sederhana adalah menerapkan cara untuk mendeteksi garis pemisah, namun dilakukan secara vertikal, mulai dari ujung paling kiri sebuah baris teks. Setelah terbagi dalam tiap-tiap karakter, piksel putih yang *redundant* dihilangkan sehingga terbentuk isolasi berbentuk segi empat yang melekat pada ujung-ujung karakter.

Gbr. 2 Hasil ekstraksi karakter

**C. Ekstraksi Properti Karakter**

Setelah proses *isolasi* dilakukan, maka dilakukan *property extraction* atau perhitungan nilai-nilai properti yang melekat pada sebuah karakter. Secara keseluruhan terdapat 29 sifat atau properti yang didapatkan dari setiap karakter. Langkah pertama, karakter dibagi menjadi dua puluh lima segmen yang sama besar, dengan penentuan posisi segmen:

```
0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
```

Gbr. 3 Segmentasi karakter ke dalam 25 segmen

Kemudian dihitung tiap-tiap properti dengan rincian sebagai berikut:

- Properti 0-24 adalah nilai rasio jumlah piksel hitam dan jumlah piksel putih pada tiap segmen karakter dalam skala keabuan (grayscale). Sebagai contoh segmen kiri atas (0) terdiri dari 4 piksel putih dan 5 piksel hitam, maka properti  $0 = (255+255+255+255+0+0+0+0)/9 = 113,333 = 113$ . Nilai pecahan yang didapat dibulatkan.
- Properti 25 adalah rasio jumlah piksel hitam dan jumlah piksel putih dalam skala keabuan, untuk separuh bagian atas karakter.
- Properti 26 adalah rasio jumlah piksel hitam dan jumlah piksel putih dalam skala keabuan, untuk separuh bagian bawah karakter.
- Properti 27 adalah rasio lebar dan tinggi karakter dalam skala 0-255.
- Properti 28 adalah indikator karakter yang terpisah secara vertikal seperti i dan j. Jika tidak terpisah, indikator bernilai 0. Jika terpisah bernilai 255.

Rangkaian kedua puluh sembilan properti di atas disebut *property set*.

**D. Perbandingan Karakter Input dan Learning Characters**

Seperti telah dikemukakan di awal, sebelum menerima input, algoritma ini terlebih dahulu mengenali karakter-karakter yang terdapat *learning set*. Proses yang dijalankan untuk mengenali *learning characters* sama dengan pengenalan karakter input, sehingga akan dihasilkan pula sebuah *property set* untuk setiap *learning character*.

Dengan demikian pada tahap ini dilakukan perbandingan nilai masing-masing properti antara sebuah karakter input dengan setiap karakter yang tersimpan dalam *learning set*.

Perhitungan yang dipakai dalam perbandingan ini didefinisikan dalam *pseudo-code* berikut [4]:

```
Distance Component::distance(Component * comp)
{
    Property * a = fproperty;
    Property * b = comp->properties();
    Distance dist=0;
    int dif=0;

    for(int i= 0; i < numProperties; i++)
    {
        dif = a[i] - b[i];
    }
}
```

```

dist += dif * dif;
}
return dist;}
    
```

*Distance* adalah jumlah dari kuadrat selisih nilai masing-masing properti, di mana *distance* sama dengan nol menunjukkan kemiripan yang sempurna sedangkan *distance* yang besar menunjukkan bahwa karakter input tidak sama dengan *learning character* yang dijadikan acuan perbandingan. Untuk *mempercepat* proses pengenalan, Marsden mendefinisikan *confidence* yang merupakan jumlah *distance* dari properti 0 sampai 24 saja. Keempat properti lainnya dipergunakan untuk menggolongkan karakter berdasarkan jenis karakter, yaitu karakter tinggi, karakter pendek, dan karakter yang memiliki titik terpisah seperti i dan j [5].

$$Confidence = \sum_{n=0}^{24} (P_{learned} - P_{extracted})^2 \dots\dots\dots 1$$

Keterangan :

$P_{learned}$  = properti karakter *learning*

$P_{extracted}$  = properti karakter *input*

n = indeks properti.

Persamaan 1 menunjukkan rumus *confidence*, dan nilai *confidence* maksimum yang disarankan dari hasil eksperimen Marsden adalah 65025. Jika nilai *confidence* melewati batas maksimum maka karakter *input* dianggap tidak sama dengan *learning character* yang sedang dijadikan acuan. Jika nilai *confidence* sama dengan nol, maka karakter *input* disebut memiliki *perfect confidence*, atau memiliki kemiripan 100% dengan *learning character*.

Untuk properti 25 hingga 28 dilakukan perhitungan tersendiri sebagai syarat apakah sebuah karakter *input* dapat diklasifikasikan yaitu:

- *Distance* properti 25 dan 26 dijumlah, hasilnya harus kurang dari akar *confidence* maksimum.
- *Distance* properti 27 harus kurang dari sepersepuluh akar *confidence* maksimum.
- *Distance* properti 28 harus sama dengan nol.

Jika sebuah karakter *input* tidak dapat memenuhi batas *confidence* maksimum dan ketiga syarat klasifikasi di atas terhadap minimum satu *learning character*, maka karakter *input* tersebut dinyatakan tidak dikenali. Sebaliknya jika sebuah karakter *input* telah memenuhi syarat klasifikasi dan memenuhi batas *confidence* maksimum terhadap beberapa *learning character*, maka diambil nilai *confidence* terkecil untuk menentukan representasi *learning character* yang paling mendekati.

### III. METODE PENELITIAN

*Tahap* pertama dari penelitian yang akan dilaksanakan adalah mengembangkan sebuah program aplikasi sebagai alat bantu pembelajaran aksara Jawa bagi anak dengan usia

setingkat Sekolah Dasar. Program aplikasi tersebut harus mencakup pembelajaran mengenai hal-hal berikut:

#### 1. Mengetahui Aksara Jawa

Program dapat menampilkan seluruh aksara Jawa dalam bentuk dasar (*nglegena*) beserta pasangannya. Bentuk dasar akan ditampilkan seluruhnya (20 karakter hanacaraka) beserta pasangannya, serta beberapa tanda baca, angka dan aksara Murda yang sesuai dengan kurikulum yang diajarkan di tingkat sekolah dasar. Untuk setiap aksara, program juga akan menampilkan beberapa kata-kata contoh secara acak untuk penggunaan aksara tersebut.

#### 2. Mengetahui Cara Penulisan Aksara Jawa

Untuk mengetahui cara penulisan aksara Jawa, maka anak akan menuliskan seluruh suku kata (*syllables*) dari kata yang ingin diketahui penulisannya, menggunakan aksara latin melalui *keyboard*. Program akan menampilkan penulisan aksara Jawa yang benar untuk kata yang telah dimasukkan. Untuk taraf pembelajaran setingkat Sekolah Dasar, maka kata yang dapat dimasukkan dibatasi menjadi 3 kata dalam satu kalimat.

#### 3. Menulis Aksara Jawa Dengan Benar

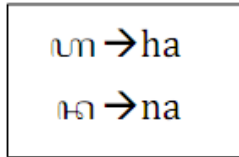
Program juga dapat digunakan untuk berlatih menulis aksara Jawa dengan benar, melalui soal yang diberikan di layar berupa kata bahasa Jawa dengan huruf latin. Anak kemudian akan menuliskan aksara Jawa yang benar menggunakan *mouse* pada bidang yang telah disediakan di layar, kemudian program akan memeriksa penulisan tersebut, apakah sudah benar atau belum. Jika belum maka program akan menampilkan penulisan aksara Jawa yang seharusnya. Program dibuat sebagai sebuah aplikasi *desktop* dengan bahasa pemrograman Visual Basic. Apa yang menjadi pokok penelitian adalah ketepatan program dalam menampilkan aksara Jawa sesuai dengan input teks dari pengguna, serta ketepatan pengenalan aksara Jawa yang diinputkan oleh pengguna. Hasil penelitian berupa statistika ketepatan dan matriks kebingungan (*confusion matrix*) untuk membantu analisis penyempurnaan program.

*Secara* ringkas terdapat beberapa bagian yang menjadi unsur pokok dalam pengembangan program aplikasi tersebut:

#### 1. Pelatihan Pola

Agar dapat melakukan pengenalan pola dan mempermudah alur pemrograman, setiap karakter aksara Jawa dan ekuivalensi suku katanya dalam aksara latin akan terlebih dahulu dilatihkan ke dalam program, untuk membentuk basis data awal. Setiap aksara Jawa akan diinputkan dalam bentuk citra *bitmap* keabuan 8-bit, sedang ekuivalensi latinnya dalam bentuk karakter ASCII.

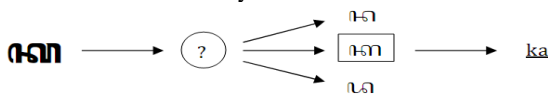
Citra aksara Jawa kemudian akan diambil fiturnya. Fitur tersebut akan disimpan dalam bentuk relasi pemetaan, satu fitur aksara Jawa terhadap satu ekuivalensi suku kata latinnya.



Gbr. 4 Pemetaan karakter aksara Jawa

2. Pengenalan Pola

Pada bagian belajar menulis aksara Jawa, aksara Jawa yang diinputkan oleh pengguna akan dikenali oleh program dan dievaluasi ketepatannya, sehingga dapat dinyatakan benar atau salah. Pengenalan dilakukan dengan mengubah input dari pengguna ke dalam bentuk citra *bitmap* keabuan 8-bit, melakukan ekstraksi fitur terhadap citra tersebut dan membandingkan fitur yang didapat dengan fitur aksara-aksara yang sudah disimpan untuk mendapatkan derajat kemiripan. Jika fitur mendekati tepat seratus persen maka aksara dianggap sama. Hasil identifikasi aksara kemudian dipetakan ke ekuivalensi suku kata latinnya untuk kemudian dicocokkan.



Gbr. 5 Pengenalan pola karakter aksara Jawa

3. Struktur Data

Struktur penyimpanan data bukan berupa basis data tetapi menggunakan struktur direktori, guna memudahkan distribusi dan instalasi program serta mengurangi ketergantungan terhadap sebuah sistem basis data tertentu. Data yang dibutuhkan untuk disimpan sebagai penunjang program di antaranya adalah:

- a. Data citra aksara Jawa untuk menampilkan aksara Jawa secara visual
- b. Data teks ekuivalensi suku kata aksara Jawa
- c. Data fitur aksara Jawa yang dimasukkan sebagai data pelatihan

Data teks dan fitur akan disimpan dalam bentuk *file* teks, data citra akan disimpan dalam bentuk *file* bitmap. Asumsi awal adalah bahwa data yang perlu disimpan tidak berukuran besar karena kumpulan aksara Jawa yang akan digunakan dalam program hanya yang sesuai dengan kurikulum pengajaran bahasa Jawa untuk Sekolah Dasar. Data citra diperlukan supaya tidak perlu melakukan instalasi *font* untuk menampilkan aksara Jawa.

IV. ANALISIS DAN PERANCANGAN SISTEM

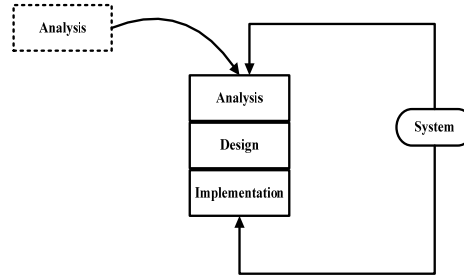
4.1 Tahapan Penelitian

Penelitian akan dibagi menjadi dua tahap yaitu tahapan pengembangan program aplikasi pembelajaran aksara Jawa. Selanjutnya penelitian melalui uji efektifitas program aplikasi tersebut melalui pengujian algoritma.

4.2 Analisis dan Perancangan Sistem

SDLC (*System Development Life Cycle*) yang digunakan sebagai metodologi adalah *The Extreme Programming*. Berikut

ini pemodelan berorientasi obyek untuk menggambarkan proses bisnis berupa aliran data dan proses ke dalam dan luar sistem dilihat dari sisi *user* menggunakan standar UML 2.0 seperti Gambar 6.



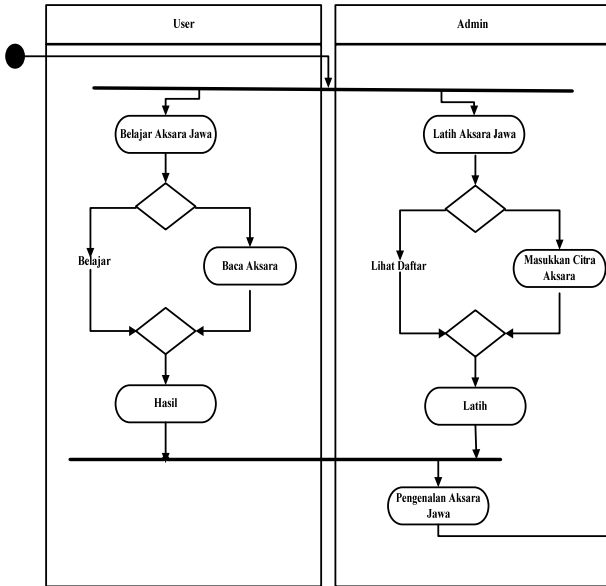
Gbr. 6 Proses Bisnis (Sumber : Dennis et all : hal 14, 2006)

4.2.2 Analisis Data dan Proses pada Sistem

Tahapan Analisis berorientasi obyek ini menggunakan *activity diagram*, *use-case diagram* untuk menggambarkan aktifitas yang dilakukan oleh *user* dan *admin* pada *functional modeling*. Analisis Sistem Pengembangan Program Bantu Pembelajaran Aksara Jawa untuk Anak Sekolah Dasar ini, dibuat sesuai dengan SDLC berorientasi obyek.

4.2.3 Analisis menggunakan Activity Diagram

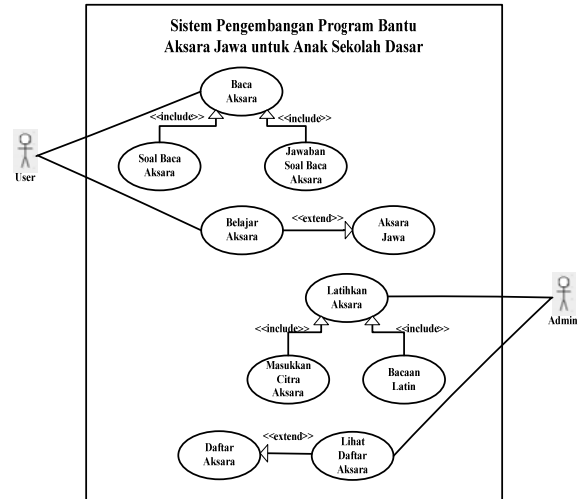
*Activity diagram* dibawah ini memberikan gambaran 2 (dua) obyek yaitu user dan admin yang digambarkan melalui *swimlane*. Adapun user memiliki aktifitas berupa Belajar Aksara Jawa dengan *decision* Belajar dan Baca Aksara. Jika memilih Baca Aksara, maka aktifitas selanjutnya memberikan hasil atas pembelajaran, sedangkan aktifitas Belajar juga memberikan hasil seperti terlihat pada Gambar 7. Obyek admin memiliki aktifitas Latih Aksara Jawa, setelah itu memiliki *decision* Lihat Daftar dan Memasukkan Citra Aksar. Melalui aktifitas Memasukkan Citra Aksara menghasilkan Latih. Baik obyek user dan admin, keduanya memiliki hasil akhir yang sama berupa aktifitas Pengenalan Aksara Jawa.



Gbr. 7 Activity Diagram Sistem Pengembangan Program Bantu Pembelajaran Aksasa Jawa untuk Anak Sekolah Dasar

4.2.4 Analisis menggunakan Use-case Diagram

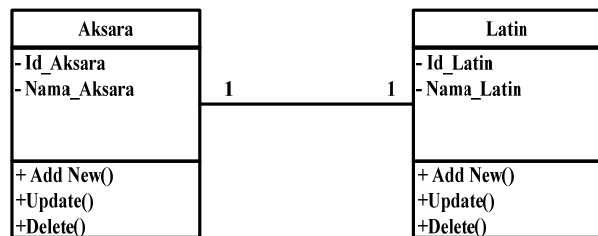
Use-case diagram pada Gambar 8 dibawah ini memiliki 2 (dua) actor yaitu user dan admin yang digambarkan diluar batasan dari Sistem Pengembangan Program Bantu Aksasa Jawa untuk Anak Sekolah Dasar. Actor Admin melakukan masukkan data untuk latihan pada use cara Latihkan Aksara dengan cara memasukkan Citra Aksara dan Bacaan Latin yang tergambar pada kedua use case <<include>> ke dalam Sistem, setelah itu proses berikutnya adalah memanggil latihan aksara yang telah dibuat dalam use case sebelumnya dengan use case Lihat Daftar Aksara yang dilanjutkan dengan use case Daftar Aksara untuk dicek kelengkapannya. Actor user memiliki satu use-case Baca Aksara, dalam hal ini melakukan proses yang sama serta bersamaan dengan use-case Soal Baca Aksara dan Jawaban Soal Baca Aksara. Adapun use-case lainnya pada actor user adalah use-case Belajar Aksara yang dilanjutkan dengan use-case Aksara Jawa ditampilkan.



Gbr. 8 Use-case Diagram Sistem Pengembangan Program Bantu Pembelajaran Aksasa Jawa untuk Anak Sekolah Dasar

4.2.5 Perancangan Sistem dengan Class Diagram

Class diagram digunakan pada tahapan perancangan berorientasi obyek yang digunakan pada structural modeling, untuk menggambarkan hubungan antar kelas pada tiap obyek. Tiap class memiliki an attribute, an operation dan an association yang saling berhubungan. Gambar 9 dibawah ini menggambarkan 2 (dua) buah class diagram yang memiliki an association 1 (satu) sama lainnya, sehingga multiple value, ketergantungan fungsional dan transitif tidak terjadi. Kedua class diagram sama-sama memiliki method yang sama. Adapaun kelas diagram dibawah ini memiliki hubungan 1 class Aksara, hanya dapat dikenali dengan 1 class Latin, demikian sebaliknya. Sehingga kedua class diagram tersebut memiliki an association one of one.

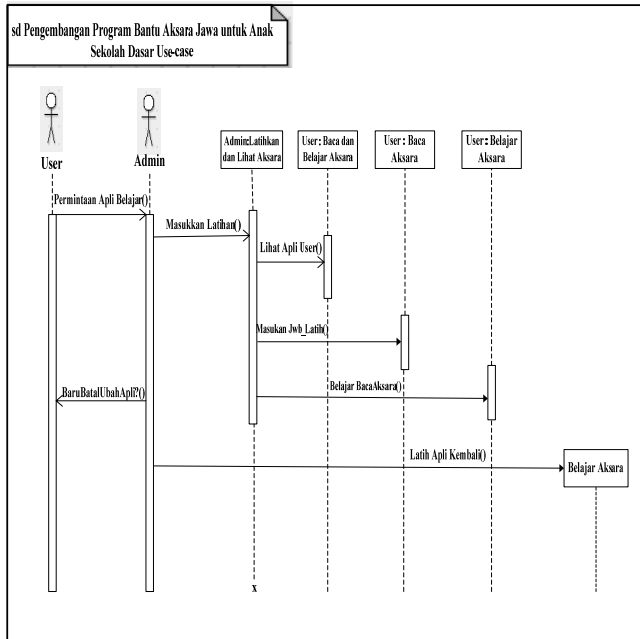


Gbr. 9 Class Diagram Sistem Pengembangan Program Bantu Pembelajaran Aksasa Jawa untuk Anak Sekolah Dasar

4.2.6 Perancangan Perangkat Lunak dengan Sequence Diagram

Tahapan selanjutnya dari SDLC yang penulis gunakan adalah behavioral modeling yaitu sequence diagram. Sesuai dengan namanya, sequence diagram dapat dibaca dari arah kiri ke kanan secara berurutan. Sequence diagram Pengembangan Program Bantu Aksasa Jawa untuk Anak Sekolah Dasar Use-case memiliki 2 (dua) actor yaitu user dan admin. Actor user menampilkan 3 (tiga) obyek dan actor admin memiliki 1

(obyek) yang berpartisipasi, seperti terlihat pada Gambar 10. Obyek *admin* memasukkan Latihan pada Aksara dan Membuat Jawaban dari Latihan Aksara Jawa tersebut, dilanjutkan dengan hasil Latinnya. Obyek *user* meminta aplikasi belajar Aksara Jawa kepada Sistem, kemudian *user* mengerjakan soal-soal yang tertera didalam Sistem tersebut secara berkelanjutan. Setelah obyek *user* selesai mengerjakan soal-soal, maka, Sistem memberikan keluaran kepada obyek *user* berupa Skor dan Keterangan Hasil dari Aplikasi dengan Soal Latihan Aksara Jawa.

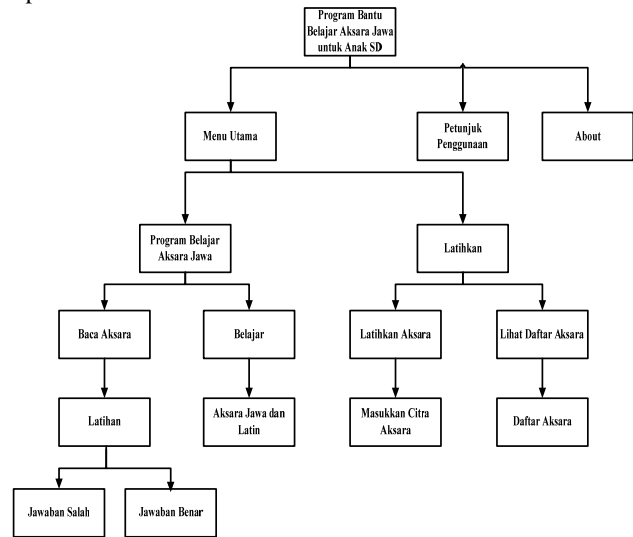


Gbr. 10 Sequence Diagram Sistem Pengembangan Program Bantu Pembelajaran Aksasa Jawa untuk Anak Sekolah Dasar

4.2.7 Rancangan Layar

Tahapan akhir dari analisis dan perancangan sistem adalah rancangan layar. Rancangan layar yang digunakan adalah *Hierarchy Layer Diagram*. Gambar 11 dibawah ini menggambarkan *top down* hirarki yang menunjukkan masing-masing *states*. *State* utama adalah Program Bantu Belajar Aksara Jawa untuk Anak SD, selanjutnya 3 (tiga) *states* yang memiliki kedudukan sama yaitu Menu Utama, Petunjuk Penggunaan dan About. Selanjutnya hanya *state* Menu Utama yang memiliki 2 (dua) *states* yaitu Program Belajar Aksara Jawa dan Latihkan. Tiap-tiap *states* yang diwakili keduanya memiliki fungsi masing-masing. *State* Baca Aksara, memiliki *state* Latihan, setelah *state* latihan dilakukan 2 (dua) *states* dibawahnya melakukan tugas masing-masing yaitu memberikan Jawaban Salah dan Jawaban Benar. Sedangkan *state* Belajar, hanya melakukan 1 (satu) dibawahnya yaitu memproses Aksara Jawa dan Latin. Semua *states* yang merupakan *state* menu utama dilakukan dari sisi *user*. Sedangkan *state* Latihkan, masing-masing yaitu *state* Latihkan Aksara memproses Masukkan Citra Aksara disisi Admin dan

*state* Lihat Daftar aksara melakukan proses Daftar aksara seperti Gambar 11.



Gbr. 11 Hierarchy Layer Diagram Sistem Pengembangan Program Bantu Pembelajaran Aksasa Jawa untuk Anak Sekolah Dasar

V. HASIL DAN PENGUJIAN

5.1 Pengujian Algoritma Pengenalan Aksara

Pengujian dilakukan terhadap bagian Belajar – Tulis Aksara, dengan membandingkan aksara Jawa hasil masukan dari pengguna yang menggunakan tipe font Javanese Unicode Font, dibandingkan dengan bentuk aksara Jawa yang telah dimasukkan melalui menu Admin – Latihkan Aksara yang berupa citra yang berasal dari pemindaian buku berbahasa Jawa.

Pengujian dilakukan dengan mengambil sampel berupa 30 soal secara acak dengan jawaban yang diberikan pengguna secara acak pula (belum tentu jawaban yang dimasukkan benar).

Dari 30 soal tersebut, masing-masing berisi dua aksara, sehingga total didapatkan 60 aksara Jawa. Hasil pengujian ditunjukkan oleh Tabel I.

TABEL 1  
HASIL PENGUJIAN

No.	Soal	Jawaban	Evaluasi sistem	Hasil Pengenalan	Tingkat Akurasi Pengenalan
1.	la wa	■	benar	la wa	100%
2.	la wa	■	salah	ha wa	100%
3.	sa da	■	salah (seharusnya benar)	sa -	50%
4.	sa da	■	salah (seharus-	sa -	50%

			nya benar)		
5.	sa da		salah	sa sa	100%
6.	ba ya		benar	ba ya	100%
7.	ca ra		benar	ca ra	100%
8.	ca ra		salah	sa ra	100%
9.	ca ra		salah	ra ca	100%
10.	ra na		salah	ra ka	100%
11.	ra na		benar	ra na	100%
12.	na ra		benar	na ra	100%
13.	na ra		benar	na ra	100%
14.	na ta		benar	na ta	100%
15.	da sa		benar	- sa	50%
16.	da sa		salah	na ca	0%
17.	ka la		benar	ka la	100%
18.	pa la		benar	pa la	100%
19.	pa la		benar (seharusnya salah)	pa la	50%
20.	ja wa		benar	ja wa	100%
21.	nya ta		salah (seharusnya benar)	nya ba	50%
22.	nya ta		salah	ba ta	100%
23.	dha dha		benar	dha dha	100%
24.	dha dha		benar	dha dha	100%
25.	dha dha		salah	dha pa	100%
26.	ma ta		benar	ma ta	100%
27.	ba ya		benar	ba ya	100%
28.	ba ta		benar	ba ta	100%
29.	ba ta		salah (seharusnya benar)	ba ba	50%
30.	sa nga		benar	sa nga	100%
Akurasi rata-rata					87%

Hasil pengujian dapat dirangkum dalam Tabel II berikut ini:

TABEL II  
RANGKUMAN HASIL PENGUJIAN

	Dikenali dengan benar	Tidak dikenali	Salah dikenali	Total
Aksara		da	da, wa, ta	
Jumlah	52	3	4	60
Persen	87	6	7	100

Aksara yang salah dikenali adalah aksara da, pa, ta, di mana kesalahan pengenalan ditunjukkan oleh *confusion matrix* teringkas seperti ditunjukkan oleh Tabel III.

TABEL III  
CONFUSION MATRIX

	da	wa	ta
na	1	-	-
pa	-	1	-
ba	-	-	2

## VI. KESIMPULAN DAN SARAN

### 6.1. Kesimpulan

1. Algoritma pengenalan karakter OCRchie dapat digunakan untuk mengenali aksara Jawa meskipun antara data uji dengan data pelatihan menggunakan font yang berbeda, dengan tingkat akurasi yang cukup baik (87%).
2. Karakter “da” selalu gagal dikenali dengan benar karena memiliki perbedaan bentuk yang cukup signifikan antara karakter “da” dengan font Unicode (memiliki tanduk di bagian depan), dengan karakter “da” dengan font standar percetakan Belanda yang digunakan sebagai data pelatihan.
3. Sejumlah karakter “da”, “wa” dan “ta” disalahkenali sebagai karakter “na”, “pa” dan “ba” karena masalah segmentasi region of interest (ROI) yang berbeda setiap kali algoritma pengenalan dijalankan. Matriks fitur yang dihasilkan tidak sama persis sehingga ketika diukur jarak terhadap matriks fitur data-data pelatihan menghasilkan urutan minimum yang berbeda, dan aksara yang terpilih sebagai ekuivalennya (memiliki nilai jarak minimum) juga berbeda.

### 6.2 Saran

1. Data pelatihan dimungkinkan untuk menerima berbagai macam font sebagai sampel, dan membangun *average face* untuk setiap fitur sampel yang mewakili aksara yang sama, agar mempercepat proses pengenalan.
2. Disediakan input penulisan berupa tulisan tangan menggunakan *mouse* agar dapat melatih pula keterampilan menulis aksara Jawa dengan tangan.
3. Program pengembangan selanjutnya dapat berupa web maupun mobile.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Kementerian Riset Teknologi dan Pendidikan Tinggi melalui Penelitian Dosen Pemula yang telah memberi dukungan financial terhadap penelitian ini.

## REFERENSI

- [1] Schulz, Klaus U: Korrektur-verfahren bei der Optischen Charaktererkennung und Dokumentenanalyse, Muenchen: CIS-LMU, 2003.



- [2] Hamilton, H.J: Confusion Matrix, [http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion\\_matrix/confusion\\_matrix.html](http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion_matrix/confusion_matrix.html), download pada 1 September 2006.
- [3] Tamatjita, E.N dan Mahastama, A. W: Optical Character Recognition dengan Algoritma OCRchie, Prosiding Konperensi Nasional Sistem Informasi, Yogyakarta: Universitas Sanata Dharma, 2008
- [4] Marsden, K: Character Recognition – OCRchie Character Recognition, <http://http.cs.berkeley.edu/~fateman/kathey/ocrchie.html>, download pada 18 Mei 2006.
- [5] Sheng, K: Optical Character Recognition (OCR) Based on OCRchie, <http://www.cs.wisc.edu/~dyer/cs766/hw/hw4/hw4-sheng/sheng.html>, download pada 18 Agustus 2006