

Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab

by Edwin Febrywinata

Submission date: 21-Jun-2024 09:14PM (UTC-0500)

Submission ID: 2406541516

File name: MERKURIUS_-_VOLUME_2,_NO._4_JULI_2024_hal_173-181.docx (713.6K)

Word count: 1874

Character count: 12592

Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab

Edwin Febrywinata

Universitas Muhammadiyah Ponorogo

Alamat: Jln Budiutomo No.10,Ronowijayan, Kec. Ponorogo, Kab. Ponorogo, Jawa Timur 63471

Korespondensi penulis: edwin999@gmail.com

Abstract. This research discusses the implementation and evaluation of the Convolutional Neural Network (CNN) convolutional neural network model for classification of fruit types, specifically to differentiate between Banana and Papaya. The CNN model used consists of several convolutional, pooling, and fully connected (dense) layers designed to extract features and perform binary classification. Data augmentation is applied to the training set to increase data variation and prevent overfitting. The image data used is normalized to speed up training convergence. The model was trained using the Adam optimizer and the binary crossentropy loss function for 20 epochs. Performance evaluation was carried out using the validation set. The results show that the model is able to effectively classify fruit images with a high level of accuracy. Predictions are made by uploading images, resizing them, and normalizing them before using the model for predictions. The classification threshold was set at 0.4, where a predicted probability greater than or equal to 0.4 was classified as Banana and a probability less than 0.4 was classified as Papaya. This research shows that the CNN model can be used effectively for binary image classification tasks and can be extended to classify more types of fruit with appropriate data adjustments and model architecture.

Keywords: Image classification, tensorflow, CNN, google colab, machine learning.

Abstrak. Penelitian ini membahas implementasi dan evaluasi model jaringan saraf konvolusi Convolutional Neural Network (CNN) untuk klasifikasi jenis buah, khususnya untuk membedakan antara Pisang dan Pepaya. Model CNN yang digunakan terdiri dari beberapa lapisan konvolusi, pooling, dan fully connected (dense) yang dirancang untuk mengekstraksi fitur dan melakukan klasifikasi biner. Augmentasi data diterapkan pada set pelatihan untuk meningkatkan variasi data dan mencegah overfitting. Data gambar yang digunakan di-normalisasi untuk mempercepat konvergensi pelatihan. Model dilatih menggunakan optimizer Adam dan fungsi kerugian binary crossentropy selama 20 epoch. Evaluasi kinerja dilakukan menggunakan set validasi. Hasil menunjukkan bahwa model mampu secara efektif mengklasifikasikan gambar buah dengan tingkat akurasi yang tinggi. Prediksi dilakukan dengan mengunggah gambar, mengubah ukurannya, dan melakukan normalisasi sebelum menggunakan model untuk prediksi. Threshold klasifikasi ditetapkan pada 0.4, di mana probabilitas prediksi lebih besar atau sama dengan 0.4 diklasifikasikan sebagai Pisang dan probabilitas kurang dari 0.4 diklasifikasikan sebagai Pepaya. Penelitian ini menunjukkan bahwa model CNN dapat digunakan secara efektif untuk tugas klasifikasi gambar biner dan dapat diperluas untuk mengklasifikasikan lebih banyak jenis buah dengan penyesuaian data dan arsitektur model yang sesuai.

Kata kunci: Klasifikasi gambar, tensorflow, CNN, google colab, machine learning.

LATAR BELAKANG

Pengenalan jenis Buah merupakan faktor yang cukup penting bagi masyarakat yang baru mengenal beberapa jenis buah yang jarang ditemui tentunya membuat bertanya tanya. Pada kasus kali ini dibuatnya sistem ini bertujuan untuk mempermudah masyarakat untuk mengenal jenis buah. Pada sistem ini akan menggunakan metode CNN (Concolutional Netural Network) untuk mengklarifikasikan citra genus ikan yang ekonomis penting. menggunakan CNN berhasil mencapai akurasi sebesar 85,31%. Model yang dibangun juga memiliki nilai rata-rata antara presisi dan sensitivitas yang baik, yaitu 89,92% dan 86,49%.

Received Mei 27, 2024; Accepted Juni 22, 2024; Published Juli 31, 2024

* Nama Penulis Pertama, edwin999@gmail.com

⁶ Metode Convolutional Neural Network (CNN) merupakan salah satu metode AI yang diterapkan khusus untuk data berbentuk visual atau citra. Metode CNN dengan arsitektur U-Net digunakan dalam penelitian ini untuk mengklasifikasi dan menentukan posisi patahan secara otomatis di dalam data seismik 3D. (⁶ *PENGGUNAAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDELINEASI PATAHAN PADA DATA SEISMIK 3D*, n.d.)

⁸ Dalam bidang deep learning, Convolutional Neural Network (CNN/ConvNet) merupakan bagian dari deep neural network, yakni jenis jaringan saraf tiruan yang umumnya digunakan dalam pengenalan dan pemrosesan gambar. Algoritma ini dirancang khusus untuk memproses data piksel dan citra visual. (⁸ *Pengertian dan Cara Kerja Algoritma Convolutional Neural Network (CNN)*, 2022)

¹ Google Collaboratory atau Google Colab adalah platform berbasis cloud untuk menulis, menjalankan, dan berbagi kode Python melalui web browser. Platform ini dirancang bagi analyst, developer, peneliti, dan pendidik yang bekerja di bidang data science dan machine learning dengan menyediakan environment komputasi yang fleksibel dan mudah diakses tanpa biaya. Google Colab juga menawarkan kemampuan untuk menjalankan Jupyter Notebook (web app open-source untuk kombinasi kode, teks terformat, dan visualisasi data) langsung dari web browser tanpa perlu konfigurasi apa pun. (*Apa Itu Google Colab?*, n.d.)

⁵ Google Drive adalah layanan penyimpanan file Google berbasis cloud. Untuk diketahui, cloud sendiri merupakan istilah yang mengacu pada teknologi server milik sebuah layanan di internet. Jadi, seandainya terdapat file yang disimpan melalui cloud, itu berarti file tersebut berada di server internet, alih-alih di memori penyimpanan fisik (hard drive, SSD, dan sejenisnya) pada komputer atau ponsel. (Media, 2023)

⁴ Tensorflow merupakan framework open-source yang dikembangkan oleh Google. Tensorflow ini digunakan untuk mengembangkan dan melatih berbagai model yang ada di machine learning, deep learning, serta pekerjaan yang berkaitan dengan analisis statistik lainnya. Tensorflow ini juga merupakan salah satu library yang paling populer serta banyak digunakan saat ini. Tensorflow ini digunakan untuk mengembangkan dan menerapkan machine learning dan algoritma lain yang memiliki banyak operasi matematika untuk dilakukan. (“Apa Itu Tensorflow?”, n.d.)

² KAJIAN TEORITIS

Bagian ini menguraikan teori-teori relevan yang mendasari topik penelitian dan memberikan ulasan tentang beberapa penelitian sebelumnya yang relevan dan memberikan acuan serta landasan bagi penelitian ini dilakukan. Jika ada hipotesis, bisa dinyatakan tidak tersurat dan tidak harus dalam kalimat tanya.

Pengenalan dan klasifikasi objek menggunakan kecerdasan buatan telah menjadi penelitian yang penting. Salah satunya yaitu pengenalan dan klasifikasi jenis buah menggunakan ³Convolutional Neural Network (CNN). Cnvolutional Neural Network(CNN) merupakan salah satu metode dari Deep Learningyang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan mempelajari data data yang ada. Metode Convolutional Neural Network memiliki hasil yang paling signifikan dalam pengenalan citra digital. (W. S. Eka Putra)

Pengenalan pola dan klasifikasi gambar merupakan identifikasi penting dari data input. Klasifikasi gambar menempatkan gambar ke dalam kategori yang telah ditentukan berdasarkan yang diidentifikasi. ⁷CNN memiliki akurasi tinggi karena memiliki jumlah ekstraksi fitur yang dihasilkan oleh konvolusi dan jumlah neuron, serta penggabungan setiap neuron dengan menggunakan bobot yang diperbarui pada iterasi tertentu. Kombinasi terbaik akan menghasilkan akurasi tinggi. (05111850010020-Master_Thesis.Pdf, n.d.)

³Metode Convolutional Neural Network memiliki hasil yang paling signifikan dalam pengenalan citra digital. Hal tersebut dikarenakan CNN diimplementasikan berdasarkan sistem pengenalan citra pada visual cortex manusia. (Maulana, F. F., & Rochmawati, N. (2019). mengeksplorasi penggunaan berbagai arsitektur CNN untuk klasifikasi buah dan menemukan bahwa arsitektur yang lebih dalam cenderung memberikan akurasi yang lebih baik.

METODE PENELITIAN

Implementasi pengenalan dan klasifikasi jenis buah menggunakan Convolutional Neural Networks (CNN) memerlukan beberapa langkah utama, yaitu:

1. Pengumpulan Data

Mengumpulkan Dataset gambar buah-buahan yang berbeda. Dataset ini bisa diperoleh dari berbagai sumber atau dibuat sendiri dengan mengambil gambar buah-buahan.

2. Pra-Pemrosesan Data

Melakukan Pra-pemrosesan pada gambar seperti mengatur penglokasian data gambar. Seperti struktur dibawah ini :

```
dataset/  
  train/  
    pepaya/  
    pisang/  
  validation/  
    pepaya/  
    pisang/
```

3. Membangun Model CNN

Merancang arsitektur CNN untuk mengenali dan mengklasifikasi jenis buah.

4. Pelatihan Model

Melatih model CNN dengan menggunakan dataset yang telah disiapkan di Pra-Pemrosesan.

5. Evaluasi Model

Menguji kinerja model pada dataset uji untuk mengukur akurasi dan kinerja lainnya.

6. Implementasi Model

Menggunakan model yang telah dilatih untuk mengklasifikasikan gambar baru (bukan yang ada di dataset sebelumnya).

HASIL DAN PEMBAHASAN

1) Pelatihan Model

Pertama tama lakukan Pelatihan model untuk membuat pengenalan dan klasifikasi jenis buah :

```
from google.colab import drive  
drive.mount('/content/TugasPM')
```

Gambar 1 Import library drive

Import library agar google colab dapat tersambung pada data yang telah dibuat. Pada kode di atas menggunakan google drive untuk menghubungkan dataset nya.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Gambar 2 Import library Tensorflow

Import beberapa modul tensorflow dan modul keras yang diperlukan agar dapat membaca data berupa gambar yang telah dibuat pada dataset sebelumnya.

```
image_size = (150, 150)
train_path = "/content/TugasPM/MyDrive/Colab Notebooks/Pembelajaran Mesin/TugasPM/train"
validation_path = "/content/TugasPM/MyDrive/Colab Notebooks/Pembelajaran Mesin/TugasPM/val"
```

Gambar 3 Menentukan Ukuran dan jalur data

Atur ukuran gambar pada data dan jalur direktori untuk data pelatihan dan validasi. Pada train_path dan validation_path seuaikan jalur pada data yang telah disiapkan sebelumnya.

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)
validation_datagen = ImageDataGenerator(rescale=1./255)
```

Gambar 4 Data augmentation dan preprocessing

PENGENALAN DAN KLASIFIKASI JENIS BUAH MENGGUNAKAN METODE CNN SECARA SEDERHANA DENGAN MENGGUNAKAN GOOGLE COLAB

Train datagen digunakan untuk menyusun agumentasi data untuk set pilihan, untuk meningkatkannvariasi data pelatihan. Validation datagen digunakan untuk penskalaan ulang piksel untuk set validasi.

```
train_generator = train_datagen.flow_from_directory(  
    train_path,  
    target_size=image_size,  
    batch_size=32,  
    class_mode='binary'  
)  
  
validation_generator = validation_datagen.flow_from_directory(  
    validation_path,  
    target_size=image_size,  
    batch_size=32,  
    class_mode='binary'  
)
```

Gambar 5 Menghasilkan Batch Data

Train generator dan validation generator menghasilkan batch dari data validasi tanpa augmentasi. Secara otomatis menghasilkan batch data yang telah di preproces sesuai kebutuhan termasuk mengubah ukuran gambar, normalisasi, dan augmentasi. Dengan menggunakan ini mudah mengelola data pelatihan dan validasi dalam bentuk batch yang efisien untuk melatih model CNN.

```
model = Sequential()  
  
model.add(Conv2D(32, (3, 3), input_shape=(image_size[0], image_size[1], 3), activation='relu'))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
  
model.add(Conv2D(64, (3, 3), activation='relu'))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
  
model.add(Flatten())  
model.add(Dense(128, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```

Gambar 6 Membangun Arsitektur Model

Mengatur beberapa model penggunaan arsitektue sequential, tambahan layer conv2D dan maxPooling2D untuk mengatur ukuran. Layer flatten digunakan untuk meratakan hasil konvolusi menjadi satu dimensi, dense layer dengan 128 neuron dan aktivitas ReLu, Dense output layer dengan satu neuron dan aktivasi sigmoid untuk klasifikasi biner.

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])  
  
model.fit(train_generator, epochs=20, validation_data=validation_generator)
```

Gambar 7 Kompilasi dan Pelatihan Model

Membuat program seperti model compile untuk mengkompilasi model dan model fit. Model dilatih menggunakan optimizer Adam dan fungsi kerugian binary crossentropy untuk melatih model menggunakan data pelatihan dan validasi selama selama 20 epoch.

```
Found 20 images belonging to 2 classes.
Found 20 images belonging to 2 classes.
Epoch 1/20
1/1 [-----] - 8s 8s/step - loss: 0.7105 - accuracy: 0.5000 - val_loss: 0.9644 - val_accuracy: 0.5000
Epoch 2/20
1/1 [-----] - 3s 3s/step - loss: 0.9992 - accuracy: 0.5000 - val_loss: 0.5810 - val_accuracy: 0.5500
Epoch 3/20
1/1 [-----] - 1s 1s/step - loss: 0.6004 - accuracy: 0.5500 - val_loss: 2.8412 - val_accuracy: 0.5000
Epoch 4/20
1/1 [-----] - 1s 1s/step - loss: 2.7667 - accuracy: 0.5000 - val_loss: 1.7849 - val_accuracy: 0.5000
Epoch 5/20
1/1 [-----] - 1s 1s/step - loss: 1.6723 - accuracy: 0.5000 - val_loss: 0.5752 - val_accuracy: 0.6000
Epoch 6/20
1/1 [-----] - 1s 1s/step - loss: 0.5640 - accuracy: 0.5500 - val_loss: 0.7867 - val_accuracy: 0.5000
Epoch 7/20
1/1 [-----] - 1s 1s/step - loss: 0.7814 - accuracy: 0.5000 - val_loss: 0.7787 - val_accuracy: 0.5000
Epoch 8/20
1/1 [-----] - 1s 1s/step - loss: 0.7930 - accuracy: 0.5000 - val_loss: 0.5752 - val_accuracy: 0.5500
Epoch 9/20
1/1 [-----] - 2s 2s/step - loss: 0.5869 - accuracy: 0.5000 - val_loss: 0.4107 - val_accuracy: 0.9000
Epoch 10/20
1/1 [-----] - 2s 2s/step - loss: 0.4476 - accuracy: 0.8500 - val_loss: 0.3404 - val_accuracy: 1.0000
Epoch 11/20
1/1 [-----] - 2s 2s/step - loss: 0.3742 - accuracy: 1.0000 - val_loss: 0.2843 - val_accuracy: 0.9500
Epoch 12/20
1/1 [-----] - 1s 1s/step - loss: 0.3223 - accuracy: 0.9500 - val_loss: 0.2284 - val_accuracy: 0.9500
Epoch 13/20
1/1 [-----] - 1s 1s/step - loss: 0.2694 - accuracy: 0.9500 - val_loss: 0.1614 - val_accuracy: 0.9500
Epoch 14/20
1/1 [-----] - 1s 1s/step - loss: 0.2281 - accuracy: 0.9500 - val_loss: 0.2067 - val_accuracy: 0.9000
Epoch 15/20
1/1 [-----] - 1s 1s/step - loss: 0.1975 - accuracy: 0.9500 - val_loss: 0.0998 - val_accuracy: 0.9500
Epoch 16/20
1/1 [-----] - 1s 1s/step - loss: 0.1195 - accuracy: 1.0000 - val_loss: 0.0699 - val_accuracy: 1.0000
Epoch 17/20
1/1 [-----] - 1s 1s/step - loss: 0.1006 - accuracy: 1.0000 - val_loss: 0.0552 - val_accuracy: 1.0000
Epoch 18/20
1/1 [-----] - 2s 2s/step - loss: 0.0656 - accuracy: 1.0000 - val_loss: 0.0600 - val_accuracy: 1.0000
Epoch 19/20
1/1 [-----] - 2s 2s/step - loss: 0.0489 - accuracy: 1.0000 - val_loss: 0.0339 - val_accuracy: 1.0000
Epoch 20/20
1/1 [-----] - 2s 2s/step - loss: 0.0296 - accuracy: 1.0000 - val_loss: 0.0180 - val_accuracy: 1.0000
<keras.src.callbacks.History at 0x7de4675b7e20>
```

Gambar 8 Hasil Pemrosesan Dataset

Hasil dari pemrosesan data yang telah dieksekusi (epoch) yaitu sesuai dengan jumlah seluruh data yang telah dibuat. Semakin banyak data yang digunakan pada dataset maka semakin akurat prediksinya.

2) Pengujian Model

Berikutnya merupakan kode untuk menguji model yang telah dilatih untuk mengklasifikasikan gambar :

```
import numpy as np
from google.colab import files
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
%matplotlib inline

uploaded = files.upload()

for fn in uploaded.keys():

    path = fn
    img = image.load_img(path, target_size=(150, 150))

    imgplot = plt.imshow(img)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    x /= 255.0

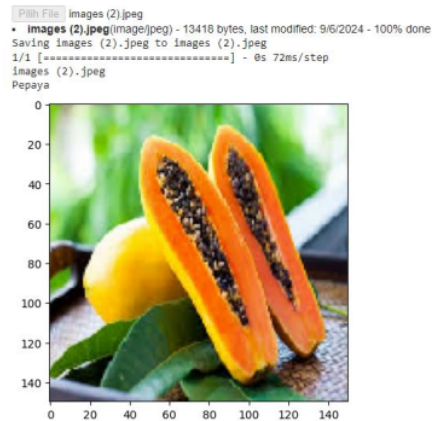
    classes = model.predict(x)

    print(fn)
    if classes[0][0] >= 0.4:
        print('Pisang')
    else:
        print('Pepaya')
```

Gambar 9 Melakukan pengujian model

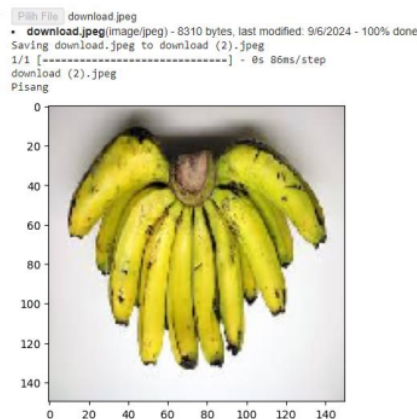
**PENGENALAN *DAN* KLASIFIKASI JENIS BUAH MENGGUNAKAN METODE CNN SECARA SEDERHANA
DENGAN MENGGUNAKAN GOOGLE COLAB**

Membuat program lagi untuk menguji model setelah pelatihan tahap selanjutnya yaitu import beberapa modul lagi sesuai yang dibutuhkan, tambahkan kode untuk upload file supaya dapat menguji dan pengkondisian Threshold klasifikasi ditetapkan pada 0.4, di mana probabilitas prediksi lebih besar atau sama dengan 0.4 diklasifikasikan sebagai 'Pisang' dan probabilitas kurang dari 0.4 diklasifikasikan sebagai 'Pepaya'. Dengan cara ini dapat menguji sistem pengenalan dan klasifikasi buah menggunakan data uji untuk mendapatkan tentang kinerja secara keseluruhan.



Gambar 10 Hasil prediksi dari input gambar

Hasil dari gambar yang telah diinputkan menunjukkan output sesuai seperti pada gambar yaitu pepaya.



Gambar 11 Hasil prediksi dari input gambar

Berikut merupakan hasil file yang telah diupload dan hasil klasifikasi sesuai dengan jenis gambar yang diinputkan yaitu pisang.

KESIMPULAN DAN SARAN

Pada penelitian yang telah dilakukan dapat disimpulkan bahwa program yang dilatih untuk memprediksi gambar berjalan sesuai dengan jenis buah yang telah diujikan. Model mampu membedakan antara pisang dan pepaya berdasarkan pelatihan yang telah dilatih. Threshold klasifikasi ditetapkan pada 0.4, di mana probabilitas prediksi lebih besar atau sama dengan 0.4 diklasifikasikan sebagai Pisang dan probabilitas kurang dari 0.4 diklasifikasikan sebagai Pepaya. Secara keseluruhan ini menunjukkan bahwa model CNN dapat digunakan secara efektif dan kinerja yang baik. Pada penelitian ini model yang dibuat masih sangat sederhana. Diharapkan kedepannya model klasifikasi ini bisa lebih lengkap lagi.

DAFTAR REFERENSI

- "05111850010020-Master_Thesis.pdf." (n.d.). Retrieved June 19, 2024, from https://repository.its.ac.id/73567/1/05111850010020-Master_Thesis.pdf
- "Apa itu Google Colab? Pengertian dan contoh 2024 | RevoU." (n.d.). Retrieved June 10, 2024, from <https://revou.co/kosakata/google-colab>
- "Apa Itu Tensorflow?" (n.d.). School of Information Systems. Retrieved June 10, 2024, from <https://sis.binus.ac.id/2024/01/18/apa-itu-tensorflow/>
- "Google Colab: Definisi, Manfaat, Hingga Cara Menggunakan." (n.d.). Retrieved June 10, 2024, from <https://digitalskola.com/blog/home/google-colab>
- "Pengertian dan Cara Kerja Algoritma Convolutional Neural Network (CNN)." (2022, July 29). Retrieved from <https://www.trivusi.web.id/2022/04/algoritma-cnn.html>
- "PENGUNAAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDELINEASI PATAHAN PADA DATA SEISMIK 3D." (n.d.). Retrieved June 9, 2024, from <https://library.universitaspertamina.ac.id/xmlui/handle/123456789/1280>
- Maulana, F. F., & Rochmawati, N. (2019). "Klasifikasi citra buah menggunakan convolutional neural network." *Journal of Informatics and Computer Science (JINACS)*, 1(02), 104-108.
- Media, K. C. (2023, January 29). "Pengertian Google Drive dan Fungsinya, Layanan Penyimpanan File Google Berbasis Cloud." *KOMPAS.com*. Retrieved from <https://tekno.kompas.com/read/2023/01/29/10150057/pengertian-google-drive-dan-fungsinya-layanan-penyimpanan-file-google-berbasis>
- Putra, W. S. E. (2016). "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101." *J. Tek. ITS*, 5(1). doi: 10.12962/j23373539.v5i1.15696

Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab

ORIGINALITY REPORT

22%

SIMILARITY INDEX

22%

INTERNET SOURCES

2%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1

revou.co

Internet Source

4%

2

journal.appihi.or.id

Internet Source

3%

3

ejurnal.methodist.ac.id

Internet Source

3%

4

sis.binus.ac.id

Internet Source

3%

5

tekno.kompas.com

Internet Source

3%

6

library.universitaspertamina.ac.id

Internet Source

3%

7

digilib.unila.ac.id

Internet Source

2%

8

www.trivusi.web.id

Internet Source

2%

Exclude quotes On

Exclude matches < 2%

Exclude bibliography Off