Jurnal Komtika (Komputasi dan Informatika),
Vol. 4 No. 1 | Mei 2020

Machine Learning Berbasis *Desktop* dan *Web* dengan Metode Jaringan Syaraf Tiruan Untuk Sistem Pendukung Keputusan

Rahmadya Trias Handayanto[1], Herlawati [2*]
[1] Fakultas Teknik, Universitas Islam 45
[2] Fakultas Teknik, Universitas Bhayangkara Jakarta Raya
*email: herlawati@ubharajaya.ac.id

DOI: https://doi.org/10.31603/komtika.v4i1.3698

Received: 26-05- 2020, Revised:03-06- 2020, Accepted:06-06- 2020

ABSTRACT

Machine learning application demand is increased massively because it provides good ability in the classification that is needed by decision makers. Machine learning application uses a programming language with strong characteristics in computing, usually the back-end programming language, such as Matlab, Python, R, etc. The obstacle faced by the decision support system developer is preparing an interface that makes it easy for the user. Some back-end programming languages have provided a good interface. Therefore, in this study they were compared by taking the case of a scholarship decision support system. The language used is Python with two web-based applications including Google Interactive Notebook and Flask framework. Both devices have their respective advantages and are worthy of being the first choice in the design of decision support systems. Python has advantages with framework Flask support and Matlab is easy in interface design.

Keywords: *Decision Support System, Flask, Jinja2, Technical Computing Language, Artificial Neural Network*

ABSTRAK

Machine learning makin diminati karena memiliki kemampuan yang baik dalam klasifikasi yang sangat dibutuhkan oleh pengambil keputusan. Dalam penerapannya machine learning menggunakan bahasa pemrograman dengan karakteristik yang tangguh dalam hal komputasi, biasanya bahasa pemrograman bertipe *back-end,* seperti Matlab, Python, R, dan sejenisnya. Kendala yang dihadapi oleh pengembang sistem pendukung keputusan adalah menyiapkan antarmuka yang mempermudah pengguna. Beberapa bahasa pemrograman *back-end* memberikan fasilitas antarmuka yang baik dan dalam penelitian ini akan dicoba dibandingkan beberapa piranti untuk pembuatan antarmuka dengan mengambil kasus sistem pendukung keputusan pemberian beasiswa. Bahasa yang digunakan adalah Python dengan aplikasi berbasis web yang digunakan adalah *Google Interactive Notebook* dan framework Flask. Kedua piranti memiliki keunggulan masing-masing sehingga layak menjadi pilihan utama dalam disain sistem pendukung keputusan. Selain itu juga dibandingkan dengan aplikasi berbasis desktop yaitu Matlab. Hasil penelitian diperoleh bahwa Python mempunyai keunggulan dengan dukungan framework Flask. Matlab memiliki kemudahan dalam disain antarmuka.

Kata-kata kunci: Sistem Pendukung Keputusan, Flask, Jinja2, Bahasa Komputasi Teknis, Jaringan Syaraf Tiruan

PENDAHULUAN

Perkembangan perangkat keras ikut andil dalam perkembangan *machine learning* (ML). Salah satunya adalah pemrosesan paralel menggunakan *Graphic Processing Unit* (GPU) dan *Tensor Processing Unit* (TPU) yang mempercepat kerja *Deep Learning* (DL), salah satu metode ML terkenal yang merupakan perkembangan lebih lanjut dari Jaringan Syaraf Tiruan

(JST). Metode ini menirukan prinsip kerja otak makhluk hidup berupa susunan sel syaraf bernama neuron dengan rangkaian sinaps yang menghubungkan satu neuron dengan neuron lainnya [1], [2].

Selain dari sisi perangkat keras, perkembangan ML juga didukung oleh perangkat lunak, salah satunya adalah bahasa pemrograman. Beberapa bahasa pemrograman yang memang ditujukan untuk mengelola ML terus menambah pustaka-pustaka (*library*) guna memudahkan pembuat program, salah satunya adalah bahasa Python. Bahasa ini telah mengadopsi pemrosesan paralel lewat teknologi *Computer Unified Device Architecture* (CUDA), khususnya untuk GPU-GPU keluaran NVIDIA [3]–[5]. Bahasa pemrograman lain yang turut berkembang adalah Matlab dengan perlengkapan (*toolbox*) untuk memudahkan penanganan ML. Tren penggunaan bahasa pemrograman Python saat ini sangat tinggi, bahkan perusahaan raksasa Google mengadopsi Python dalam bahasa pemrograman *online* (http://colab.research.google.com). Selain menyediakan fasilitas pemrograman *online*, Google juga menyediakan dua jenis perangkat kerasnya, yaitu GPU dan TPU, untuk menjalankan kode program yang telah ditulis di pemrograman *online* yang dikenal dengan istilah *Google Interactive Notebook* atau singkatnya Google Colab.

Sejak diperkenalkannya JST sederhana di tahun 40-an yang dikenal dengan nama McCulloch-Pitts, metode ini terus berkembang di era 90-an ketika metode terkenal perambatan balik (*backpropagation*) mampu menyelesaikan problem model JST McCulloch-Pitts [1]. Dengan perambatan balik, sebuah JST lapis banyak (*multi-layer perceptron*) dapat dengan mudah disetel baik bobot maupun biasnya, namun untuk jumlah *layer* yang besar tetap kesulitan karena butuh kerja *central processing unit* (CPU) yang terbatas karena memang sifatnya yang serial. Dengan bantuan GPU, sebuah prosesor yang sebelumnya difungsikan untuk mengelola grafis (dalam bentuk kartu grafis), JST berlapis banyak dapat dikelola. Karena penanganan yang sangat berbeda dengan JST, model baru JST dikenal dengan istilah *Deep Learning* (DL) yang bercirikan jumlah lapis (termasuk neuronnya) yang berukuran besar serta data latih yang juga besar seukuran Bigdata (namun biasanya tetap berjenis data terstruktur) [6].

Penelitian ini bermaksud membandingkan dua jenis ML berbasis web yaitu Google Colab dan framework Flask yang menggunakan bahasa Python. Selain itu sebuah ML berbasis desktop dengan bahasa Matlab juga dilibatkan dalam penelitian ini sebagai pembanding. Sebuah kasus sederhana akan diselesaikan lewat sebuah sistem pendukung keputusan (*decision support system*) yang dibuat menggunakan ketiga jenis metode tersebut sebelum dibandingkan agar diketahui keunggulan dan kelemahan masing-masing piranti pembuat ML tersebut.

METODE

Metode yang digunakan dalam penelitian ini adalah jaringan syaraf tiruan perambatan balik. Secara umum, algoritma perambatan balik dapat dijelaskan sebagai berikut [10]:

Tahap 0: Pembobotan Awal (Set ke nilai random serendah mungkin), set harga *error* minimal.
Tahap 1: Ketika kondisi *stop*, *False*, lakukan tahap 2 – 9.
Tahap 2: Untuk setiap pasangan *training*, lakukan tahap 3 – 8.

Feedforward:

Tahap 3: Tiap unit masukan *(X_i, $i = 1, ..., n$)*, menerima sinyal x_i dan menyebarkan sinyal ke seluruh lapis tersembunyi (*hidden layer*).

Tahap 4: Tiap unit tersembunyi *(Z_j, $j = 1, ... ,p$)*, jumlahkan bobot sinyal inputnya,

$z\_in_j = v_{0j} + \sum_{i=1}^{p} x_i v_{ij}$,

Terapkan fungsi aktivasi untuk menghitung sinyal keluarannya,

$z_j = f(z\_in_j)$, dan mengirimkan sinyal ini ke seluruh unit lapis di atasnya (lapis output).

Tahap 5: Tiap unit keluaran *(Y_k, $k = 1, ... , m$)*, jumlahkan bobot sinyal keluarannya,

$y\_in_k = w_{0k} + \sum_{j=1}^{m} x_j v_{jk}$, Sedangkan untuk Cascade: $y\_in_k = w_{0k} + \sum_{j=1}^{m} x_j v_{jk} + \sum_{i=1}^{p} x_i v_{ij}$)

Terapkan fungsi aktivasi untuk menghitung sinyal keluarannya,

$y_k = f(z\_in_k)$.

Backpropagation Error:

Tahap 6: Tiap unit keluaran *(Y_k, $k = 1, ... , m$)*, menerima pola target dan mengacu ke target masukan, hitung kesalahannya

$\delta_k = (t_k - y_k)f'(y\_in_k)$,

Hitung koreksi bobot

$\Delta w_{jk} = \alpha\delta_k z_j$,

Hitung koreksi terhadap bias

$\Delta w_{0k} = \alpha\delta_k$,

dan mengirimkan sinyal tersebut ke lapis sebelumnya (mundur).

Tahap 7: Tiap unit tersembunyi *(Z_j, $j = 1, ... ,p$)*, menjumlah delta masukan dari lapis diatasnya.

$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$

Kalikan dengan turunan dari fungsi aktivasi untuk mencari kesalahan,

$\delta_j = \delta\_in_j f'(z\_in_j)$,

dan hitung bobot koreksinya

$\Delta v_{ij} = \alpha\delta_j x_i$,

Dan hitung koreksi biasnya,

$\Delta v_{0j} = \alpha\delta_j$,

Update bobot dan bias:

Tahap 8: Tiap unit keluaran *(Y_k, $k = 1, ... , m$)*, update bias dan bobot-bobotnya *($j = 0, ... , p$):*

$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$,

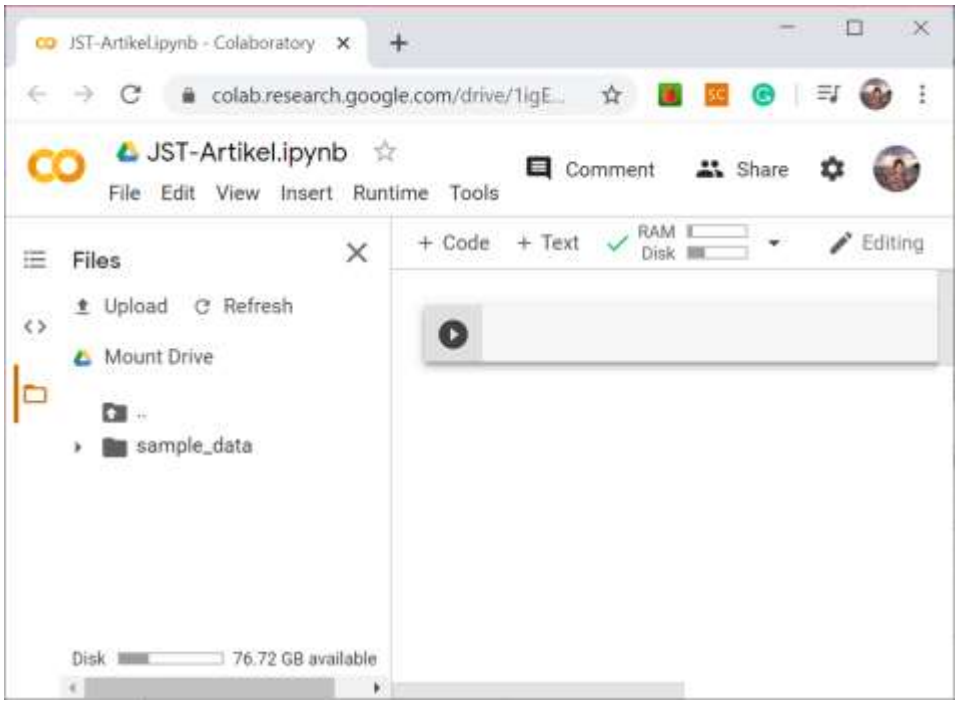
Tiap unit tersembunyi *(Z_j, $j = 1, ... ,p$)*, update bias dan bobotnya *($i = 0, ... , n$):*

$v_{ij}(new) = v_{ij}(old) + \Delta w_{ij}$,

Tahap 9: Uji kondisi berhenti.

If $\delta_k <$ harga error set awal Then "Stop Training"

Google menyediakan fasilitas pemrograman *online* yang dapat diakses di http://colab.research.google.com dengan *Integrated Environment Development* (IDE) merujuk Jupyter Notebook dengan ekstensi *.ipynb. Struktur IDE Google Colab tampak pada Gambar 1 berupa jendela kode berupa kumpulan sel (*cell*), jendela *file* yang secara *default* tertutup untuk keperluan unggah dan unduh *file* pendukung program, dan menu.



Sumber: Hasil Penelitian (2020)

Gambar 1. Google Interactive Notebook

Untuk memulai Google Colab, masuk ke menu dan pilih "new notebook". Karakter berupa sel pada Google Colab berfungsi untuk memisahkan antara satu grup kode, misalnya impor pustaka, dengan grup lainnya, misalnya proses *training*. Keunggulan IDE ini adalah fasilitas yang disediakan oleh Google untuk menggunakan GPU dan TPU milik mesin pencari ini yang terkenal tangguh. Sementara jika menggunakan Jupyter Notebook harus menyediakan kartu grafis GPU sendiri sebagai *accelerator*. Untuk memanfaatkan GPU/TPU pengguna dapat masuk ke menu *edit* lalu memilih GPU atau TPU setelah masuk ke *notebook settings*.

Framework Flask

Selain sebagai aplikasi *back-end*, Python juga telah lama digunaan oleh *front-end developer* untuk membuat aplikasi dengan *Graphic User Interface* (GUI) berbasis *web*. Beberapa *framework web* berbasis Python adalah Django, Flask, Bottle, Tornado, CubicWeb, Dash, Pyramid, Turbo Gears, dan *framework* lain yang masih dalam pengembangan [11]. Dalam artikel ini, framework Flask digunakan sebagai pembanding. Flask masuk dalam kategori *microframework* yang ditujukan untuk aplikasi skala kecil yang digunakan untuk divisi kecil. Jika ingin membuat aplikasi *enterprise* dapat menggunakan *framework* Django. Selain itu fasilitas lain juga diperlukan untuk menyisipkan kode Python dalam *file* html yaitu Jinja2.

Sebelum menggunakan *framework* ini, pengguna perlu mengunduh pustakan Flask dan Jinja2 menggunakan Pip Install Python (PIP) jika menggunakan versi terminal atau dengan mengunduh lewat paket Anaconda. Flask lebih sederhana dibanding Django dan framework skala besar lainnya. Satu file python (berekstensi *.py) bekerja di sisi server dengan port tertentu baik *training* maupun *testing* kemudian sebuah *file* html menampilkan antarmuka

yang dapat dilihat oleh pengguna. Lokasi *file* ditempatkan dalam folder tertentu, misalnya folder *templates* yang terletak di bawah file utama Python. Untuk menjalankan server cukup dengan mengarahkan terminal pada direktori tempat *file* utama Python berada dan mengetik instruksi "python <file>.py". Tunggu beberapa saat hingga server flask hidup. Secara *default* Flask menggunakan port 5000.

Matlab yang merupakan kepanjangan dari Matrix Laboratory merupakan kelanjutan dari bahasa komputasi teknis Fortran [12], [13]. Ketangguhan bahasa ini adalah kesederhanaannya dalam menangani matriks yang sangat dibutuhkan dalam perhitungan numerik. Matlab menyediakan *toolbox* yang mempercepat pemrograman ML, namun tetap menyediakan kode sumber yang dapat dimanipulasi untuk keperluan riset baik dalam peningkatan performa maupun hibridisasi dengan metode-metode ML yang beragam.

Berbeda dengan python, Matlab memiliki lisensi dalam penggunaannya. Hal ini menyulitkan pengguna yang memiliki anggran terbatas. Namun komunitasnya sudah cukup banyak dan situs resminya, http://mathworks.com, menyediakan update beberapa fungsi yang belum tersedia di versi-versi sebelumnya. Dalam penulisan kode program, Matlab menyediakan editor. Selain *toolbox* fungs-fungsi terdahulu dalam format *.m dapat dengan mudah ditangani oleh editor yang dikenal dengan nama m-*file* editor.

Sejak Matlab versi 6, perkembangan antarmuka Matlab cukup masif dan dapat dikatakan menyamai antarmuka-antarmuka bahasa pemrograman *front-end* lainnya yang berbasis *dekstop* seperti Visual Basic, Delphi, Java Netbeans, dan lain-lain. Selain itu, dengan bantuan Matlab Runtime yang tidak berbayar (dikenal dengan nama Matlab Compile Runtime), sebuah aplikasi yang sudah dikompilasi dapat dijalankan pada komputer lain yang tidak terinstal Matlab di dalamnya.

Sebagai data untuk studi kasus pembuatan sistem pendukung keputusan, digunakan data sederhana berupa pemberian beasiswa berdasarkan Indeks Prestasi Kumulatif (IPK) dan Tingkat Kemiskinan (TM). Mengikuti standar, diperlukan tiga data yaitu data pelatihan (*training*), data validasi dan data pengujian. Terkadang data validasi dan data pengujian digabung menjadi satu. Contoh data untuk pelatihan dapat dilihat dalam Tabel 1.

Tabel 1. Data *Training* Beasiswa

Mahasiswa	IPK	Tingkat Kemiskinan (TM)	Perolehan Beasiswa*
1.	0.8	0.8	0
2.	1.2	1	0
3.	1	1.2	0
4.	2	1.5	0
5.	0.7	1.5	0
6.	2.5	3	1
7.	3	2.5	1
8.	3	2	1
9.	3	3	1
10.	3.5	3.9	1
11.	2.5	3.5	1
12.	2.6	3.6	1
13.	0.5	0.5	0
14.	1.0	2	0

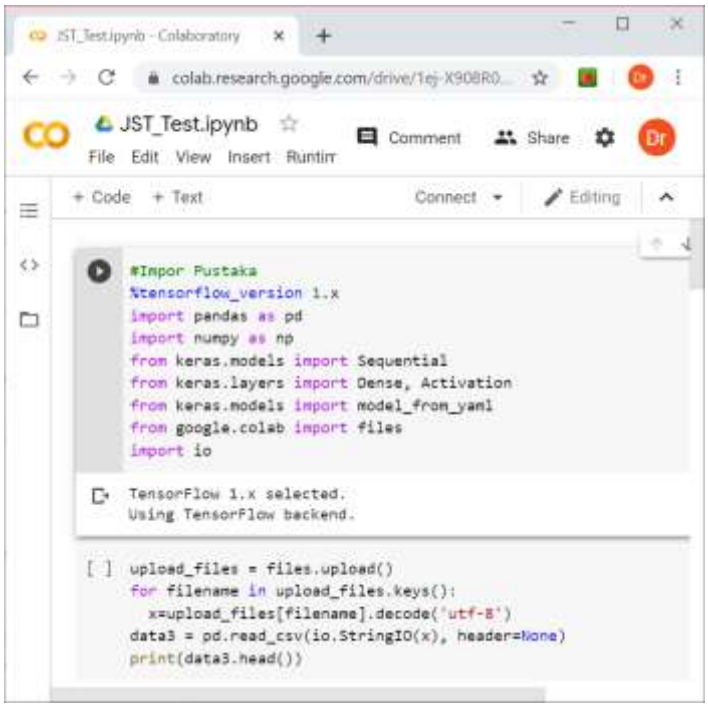
Keterangan: * nilai=1: Mendapat Beasiswa, nilai=0: Tidak Beasiswa.
Sumber: Hasil Penelitian (2020)

Untuk penggunaannya, tabel 1 tersebut disimpan dalam format Comma Seprated Value (CSV) dengan nama berturut-turut untuk data latih, data validasi dan data uji: beasiswa.csv, validasibeasiswa.csv, dan testbeasiswa.csv. Walaupun bisa menggunakan format Excel, format CSV lebih disarankan dan banyak digunakan oleh pengembang-pengembang ML. *File* tersebut dapat dibuat menggunakan spreadsheet seperti Microsoft Excel atau editor teks sederhana, misalnya Notepad, Wordpad, dan sejenisnya.

HASIL DAN PEMBAHASAN

Jika tiga buah data berformat CSV sudah tersedia dan tiga jenis media pemrograman berturut-turut Google Colab, framework Flask dan Matlab juga siap berikutnya adalah proses pembuatan aplikasi ML untuk sistem pendukung keputusan.

Google Colab sejatinya mirip Jupyter Notebook dengan tambahan fasilitas sharing agar bisa digunakan bersama. File tersimpan dalam Google Drive (http://drive.google.com). Jika diberi hak akses *edit*, beberapa pengembang dapat bersama-sama membuat program ML tanpa terkendala jarak dan waktu. Hasil pemrograman ML masih bersifat *text-based* yang mirip Jupyter Notebook maupun versi IDLE Python.



Sumber: Hasil Penelitian (2020)

Gambar 2. Pengkodean Pada Google Colab

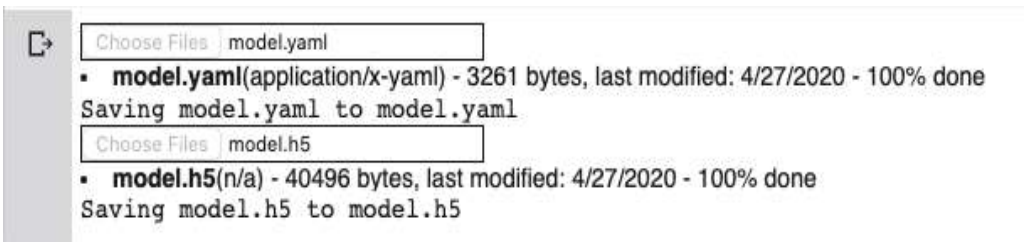
Gambar 2 menunjukan bagaimana pengkodean dengan Google Colab. Untuk kasus pelatihan JST, pustaka yang dibutuhkan adalah Keras dan Tensorflow versi 1.x. Data latih dapat diunggah lewat *file* (dengan membuka jendela tersembunyi bersimbol folder di bagian kiri) maupun lewat tombol unggah dengan memanfaatkan pustaka khusus Google Colab (pada kode di atas lewat pustaka IO). Secara umum *pseudocode* proses pemodelan JST adalah sebagai berikut.

Input: Data Latih, Data Validasi, Data Uji
Output: Model JST (file *.yaml dan *.h5)

Set layer dan neuron
Set parameter dan hyperparameter
While stop condision = False
    Set bobot dan bias
    Cek akurasi
End
Simpan model JST
Tampilkan akurasi

Dikenal dua jenis parameter yaitu parameter dan hyperparameter. Jika parameter menggambarkan struktur dari model JST, hyperparameter merupakan variabel yang digunakan dalam proses pelatihan (dalam *pseudocode* di atas dalam kalang While-End). Fungsi Python yang digunakan untuk mengklasifikasi sebuah input adalah fungsi "predict". Salah satu keunggulan Google Colab dibanding dengan Jupyter Notebook adalah pengguna tidak perlu mengunduh pustaka-pustaka yang diimpor karena semua telah disiapkan oleh Google Colab.

File yaml dan h5 yang tersimpan dapat digunakan baik dengan Google Colab maupun IDE lainnya asalkan berbasis bahasa Python. Untuk memastikan model JST hasil pelatihan berfungsi dengan baik, gunakan pengujian dengan cara mengunggah dua file hasil pelatihan yang diunduh sebelumnya, kemudian dengan data uji hasil pengujian dapat diketahui. Gambar 3 menunjukan proses pengunggahan model JST. Dua file model JST tersebut tidak serta-merta dapat digunakan, harus dikompilasi terlebih dahulu dengan fungsi "compile".



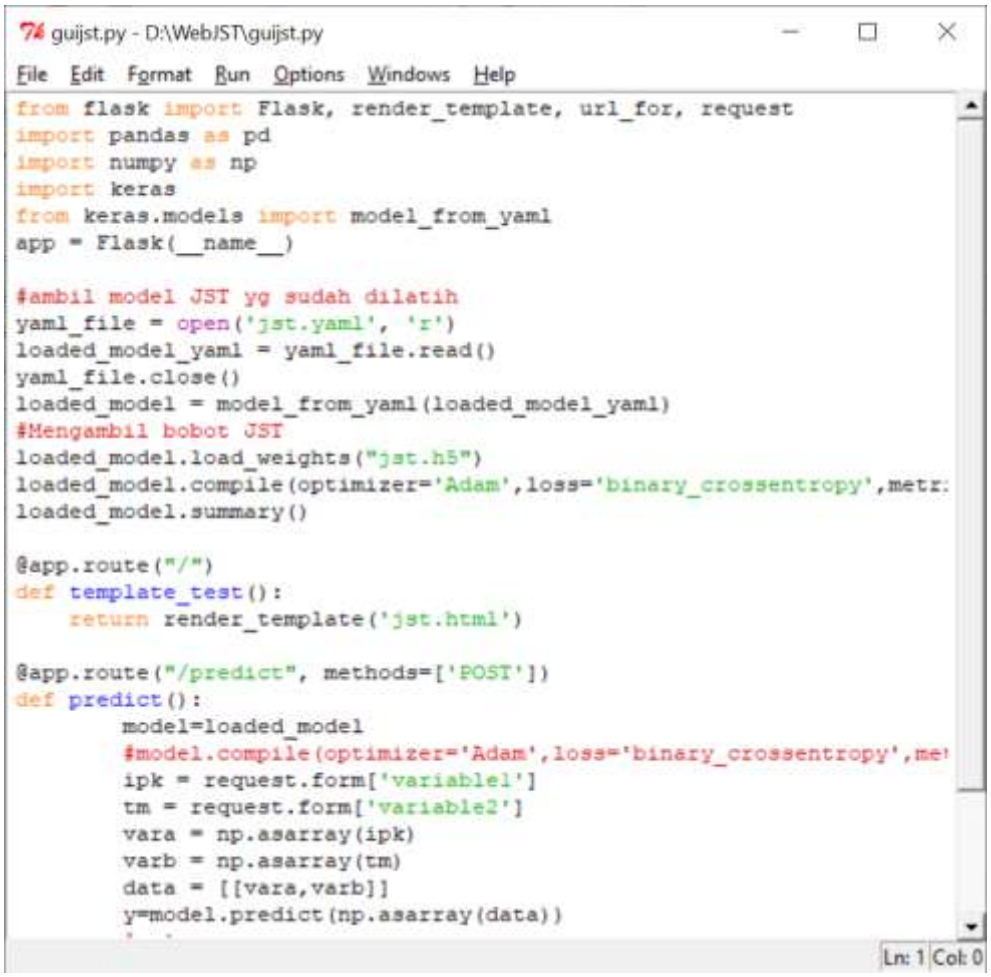
Sumber: Hasil Penelitian (2020)

Gambar 3. Antarmuka Fasilitas Unggah Model JST Pada Google Colab

Peralatan yang dibutuhkan untuk ML berbasis web dengan Flask antara lain pustaka Flask dan Jinja2 disertai dengan pustaka pendukung lainnya seperti NumPy, Pandas, dan sejenisnya. Dibutuhkan juga browser untuk menjalankan aplikasi. Untuk pengetikan bahasa pemrograman dapat menggunakan beragam teks editor seperti notepad, sublimetext, wordpad, IDLE, dan lain-lain.

Untuk pengguna *Anaconda* perlu mengeset *environment* jika tidak menggunakan *environment default*-nya. Konsol berupa *command prompt* (jika menggunakan sistem operasi

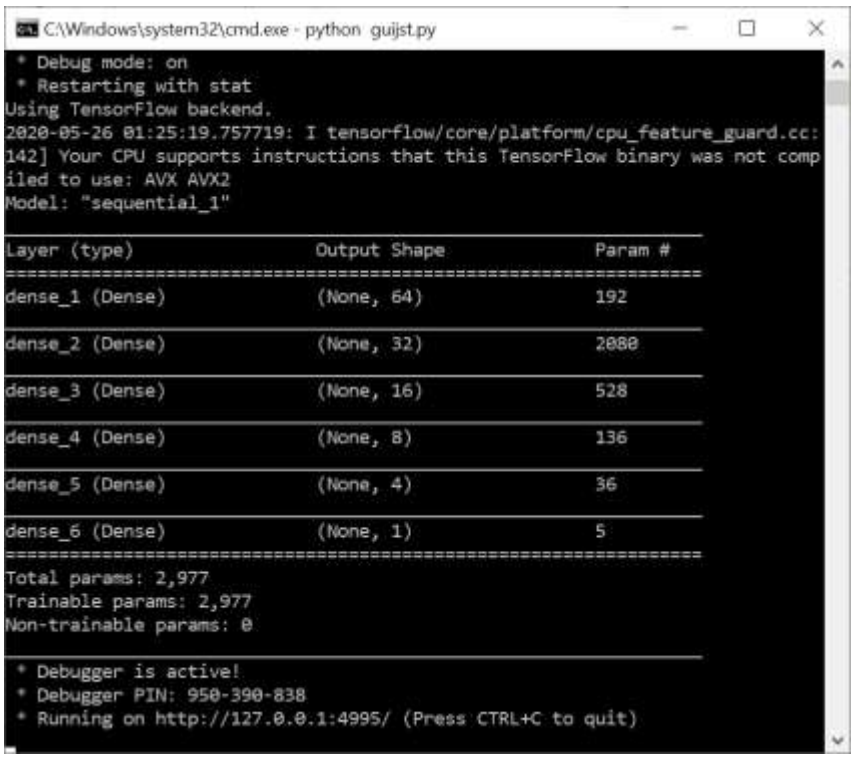
Windows) perlu dibuka untuk menjalankan server Flask. Program utama Python untuk menjalankan server Flask tampak pada Gambar 4.



Sumber: Hasil Penelitian (2020)

Gambar 4. Program Utama Python Untuk Server Flask

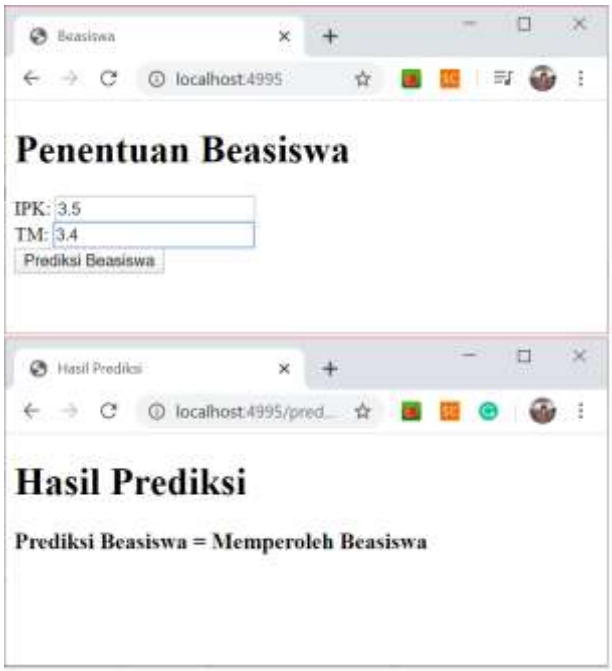
Setelah impor pustaka yang diperlukan (Flask, Pandas, NumPy, dan Keras) model JST yang telah dilatih sebelumnya, misalnya dengan Google Colab, dipanggil (contohnya adalah jst.yaml dan jst.h5). Atau bisa juga melatih model JST di sini. Tetapi biasanya pelatihan model JST dan penggunaannya terpisah. Gambar 5 menunjukkan proses server Flask yang telah running. Contoh port yang digunakan adalah 4995 dan program dapat diakses dengan alamat 127.0.0.1:4995 atau localhost:4995.



Sumber: Hasil Penelitian (2020)

Gambar 5. Aktivasi Server Flask dan Running Model

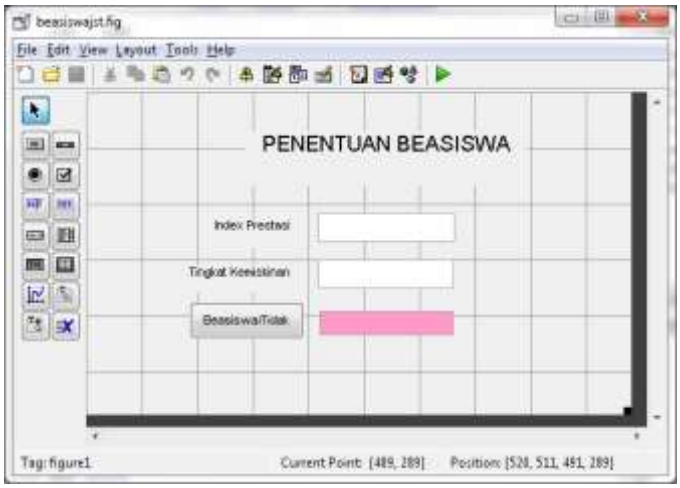
Tampak pada Gambar 5 enam buah layer JST berturut-turut memiliki neuron 64, 32, 16, 8, 4 dan 1. Untuk mengecek aplikasi tersebut, buka *browser* dan arahkan ke alamat sesuai dengan instruksi server Flask, termasuk juga alamat portnya. Gambar 6 menunjukan hasil *running web-based* ML model JST yang telah dilatih.



Sumber: Hasil Penelitian (2020)

Gambar 6. Tampilan Input-Output (atas) dan Hasil Prediksi (bawah)

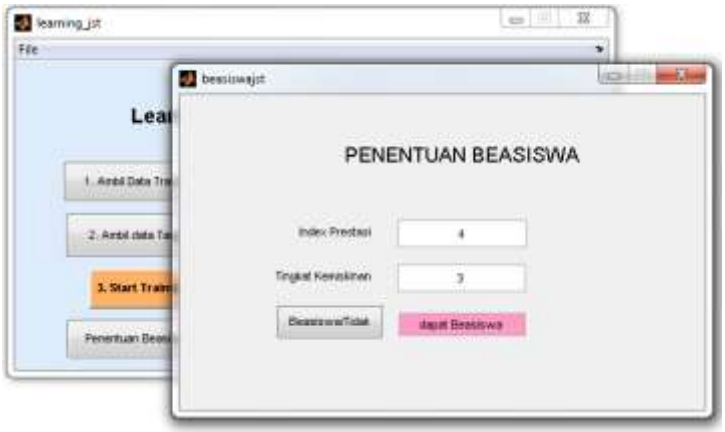
Matlab menyediakan fasilitas pembuatan antarmuka dengan fungsi "GUIDE". Fungsi ini bisa dipanggil lewat menu maupun dengan mengetik fungsi tersebut pada *command window*. Fasilitas yang terdapat dalam GUIDE antara lain: *input-output*, pembuatan label, grafik untuk citra, tombol, menu, dan fasilitas-fasilitas lain standar antarmuka (Gambar 7). Hanya saja Matlab tidak menyediakan aplikasi berbasis web dan memang bahasa ini lebih fokus ke komputasi (*back-end*).



Sumber: Hasil Penelitian (2020)

Gambar 7. GUIDE untuk Disain Antarmuka Matlab

Antarmuka yang telah dirancang secara otomatis terhubung dengan editor m-file. Berbeda dengan Python yang menggunakan dua *file* yaitu arsitektur (file yaml) dan bobot-bias (file h5), pada Matlab hasil pembelajaran model tersimpan dalam bentuk mat-file. Selain dijalankan langsung dengan menekan tombol "run" pengguna dapat menjalankan aplikasi pada komputer yang tidak memiliki Matlab dalam bentuk installer setelah dilakukan proses kompilasi. Gambar 8 menampilkan contoh dua jendela masing-masing untuk *training* dan penggunaan hasil training.



Sumber: Hasil Penelitian (2020)

Gambar 8. Antarmuka Berbasis Desktop Pada Matlab

Masing-masing piranti perancangan sistem pendukung keputusan memiliki kelebihan. Peneliti sudah banyak menggunakan piranti-piranti tersebut baik untuk pengembangan maupun uji coba penelitian. Tabel 2 menampilkan ringkasan karakteristik bahasa pemrograman dan piranti pembuat aplikasi ML.

Tabel 2. Perbandingan Tiga Alat Pembuat Aplikasi ML

Tools	**Kelebihan**	**Kekurangan**
Google Colab	- Pustaka Lengkap - Dukungan GPU dan TPU Google yang tangguh - Kolaborasi	- Kerahasiaan Data - Tidak bisa utk Aplikasi Mandiri - Tergantung *bandwidth* jaringan
Flask	- Antarmuka Web Interaktif - Didukung bahasa tangguh Python - Bisa diakses di mana saja	- Lebih rumit - Butuh pengetahuan tambahan HTML, CSS, dan sejenisnya
Matlab	- Mudah - Toolbox lengkap - Antarmuka baik - Bisa untuk aplikasi standalone	- Berbayar - Butuh sumber daya komputer sendiri

Sumber: Hasil Penelitian (2020)

KESIMPULAN

Permintaan yang tinggi terhadap aplikasi ML memaksa pengembang mencari piranti yang tepat untuk pembuatan aplikasi tersebut. Tiga piranti pembuat aplikasi ML baik berbasis desktop maupun web diteliti dan diimplementasikan untuk mengetahui karakteristik masing-masing. Python memiliki keunggulan sebagai bahasa untuk membuat baik aplikasi *front-end* maupun *back-end.* Khusus untuk *deep* JST, Google Colab menawarkan kemudahan-kemudahan kepraktisan coding serta fasilitas GPU dan TPU yang cocok untuk pemrosesan paralel. Era industri 4.0 mengharuskan aplikasi-aplikasi dapat diakses dari berbagai platform melalui media internet, sehingga aplikasi-aplikasi khususnya sistem pendukung keputusan dituntut mampu berjalan lewat web. Di sini tampak keunggulan Python dengan dukungan framework web-nya yang dalam penelitian ini framework Flask. Matlab yang memiliki kemudahan dalam disain antarmuka dan perancangan ML layak juga digunakan khususnya untuk uji coba metode karena dapat dengan mudah diduplikasi dan diuji oleh peneliti lain yang membaca hasil publikasinya. Perbandingan dengan bahasa lain seperti C# dan Java akan dilakukan dalam penelitian berikutnya.

DAFTAR PUSTAKA

[1] S. Haykin, “Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7.,” *The Knowledge Engineering Review*, vol. 13, no. 4. pp. 409–412, 1999.
[2] M. Hagan, M. T., Demuth, H. B., & Beale, *Neural Network Design*. Boston: PWS Publishing Co., 1997.
[3] M. Koprawi, “Parallel Computation in Uncompressed Digital Images Using Computer Unified Device Architecture and Open Computing Language,” *PIKSEL*, vol. 8, no. 1,

pp. 31–38, 2020.
[4] NVIDIA, "NVIDIA on GPU Computing and the Difference Between GPUs and CPUs.," 2020.
[5] L. Pan, L. Gu, and J. Xu, "Implementation of medical image segmentation in CUDA," *5th Int. Conf. Inf. Technol. Appl. Biomed. ITAB 2008 conjunction with 2nd Int. Symp. Summer Sch. Biomed. Heal. Eng. IS3BHE 2008*, pp. 82–85, 2008.
[6] P. Kim, *MATLAB Deep Learning*. New York: Apress, 2017.
[7] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. USA: Prentice-Hall, Inc., 1994.
[8] R. N. Whidhiasih, "Identifikasi tingkat manis buah belimbing berdasarkan citra red green blue menggunakan fuzzy neural network," *PIKSEL Penelit. Ilmu Komput. Sist. Embed. Log.*, vol. 3, no. 2, pp. 109–120, 2015.
[9] R. T. Handayanto and H. Herlawati, "Prediksi Kelas Jamak dengan Deep Learning Berbasis Graphics Processing Units," *J. Kaji. Ilm.*, vol. 20, no. 1, pp. 67–76, 2020.
[10] L. Fausett, *Fundamentals of Neural Networks*, no. 1. 1994.
[11] M. Nehra, "Top 5 Python Frameworks for Web Development in 2020," 2020. [Online]. Available: https://medium.com/@mahipal.nehra/top-5-python-frameworks-for-web-development-in-2020-4abd6ee5f8ec. [Accessed: 26-May-2020].
[12] P. P. Widodo, R. T. Handayanto, and H. Herlawati, *Penerapan Data Mining dengan Matlab*. Bandung: Informatika, 2013.
[13] H. Herlawati and R. T. Handayanto, "Penggunaan Matlab dan Python dalam Klasterisasi Data," *J. Kaji. Ilm.*, vol. 20, no. 1, pp. 103–118, 2020.