

## BAB IV PERENCANAAN

Pada bab ini, kita akan membahas implementasi dan metodologi yang digunakan dalam penelitian ini untuk mengembangkan model jaringan saraf tiruan (ANN) yang mampu mengklasifikasikan objek multi-kelas. Bab ini akan membahas secara rinci langkah-langkah yang diambil dalam mempersiapkan data, mengembangkan arsitektur jaringan, serta melatih dan menguji model ANN yang telah dibangun. Selain itu, kita akan membandingkan kinerja model ANN dengan satu dan dua lapisan tersembunyi, serta mengevaluasi hasil yang diperoleh melalui visualisasi *loss* dan akurasi.

### 4.1 Langkah-langkah Penelitian

- Menurut (Supriyanto et al., 2022, hal. 33) bahwa nilai lapisan tersembunyi mempengaruhi kecepatan pelatihan *Artificial Neural Network* (ANN) *backpropagation* dan *epoch*. *Artificial Neural Network* (ANN), merupakan salah satu jenis metode *machine learning*. Itu terdiri dari jaringan neuron yang saling berhubungan yang diatur berlapis-lapis. Setiap *neuron* mengambil informasi dari *neuron* lain, memprosesnya, dan kemudian mengirimkan kembali informasi ke *neuron* lain di lapisan di bawahnya.

Bobot dan bias *neuron* diubah selama fase pelatihan *Artificial Neural Network* (ANN) untuk mengurangi perbedaan antara keluaran yang diproyeksikan dan keluaran sebenarnya. *Backpropagation*, sebuah metode yang menentukan gradien dari fungsi kerugian sehubungan dengan bobot dan bias, sering digunakan untuk mencapai hal ini. Langkah-langkah penelitian ini meliputi:

- a. Melakukan studi pustaka tentang teori-teori terkait algoritma ANN 1-layer, seperti prinsip dasar, arsitektur, metode pelatihan, dan aplikasi.
- b. Melakukan studi pustaka tentang teori-teori terkait algoritma ANN 2-layer, seperti prinsip dasar, arsitektur, metode pelatihan, dan aplikasi.
- c. Melakukan studi pustaka tentang kasus pengenalan karakter, termasuk teknik pengolahan gambar, ekstraksi fitur, dan teknik klasifikasi yang umum digunakan.

- d. Membuat rancangan komputasi untuk mengukur akurasi antara algoritma ANN 1-layer dan algoritma ANN 2-layer pada kasus pengenalan karakter, termasuk pembuatan kode program, pemilihan data latih dan data uji, serta evaluasi performa.
- e. Mengumpulkan data hasil perbandingan akurasi antara algoritma ANN 1-layer dan algoritma ANN 2-layer pada kasus pengenalan karakter.
- f. Membahas hasil perbandingan akurasi, termasuk analisis terhadap faktor-faktor yang mempengaruhi kinerja kedua algoritma, serta identifikasi kelebihan dan kekurangan dari masing-masing algoritma.
- g. Membuat kesimpulan berdasarkan hasil penelitian, termasuk rekomendasi untuk penelitian lebih lanjut dan aplikasi praktis dari algoritma yang lebih unggul dalam penelitian ini.

Selain itu, analisis terdahulu juga akan menghubungkan hasil analisis dengan rencana penelitian yang akan dilakukan. Hal ini dilakukan untuk mengidentifikasi kebutuhan dan tujuan penelitian secara lebih spesifik. Menurut (Supriyanto et al., 2022, hal. 33), nilai lapisan tersembunyi memiliki pengaruh terhadap kecepatan pelatihan Artificial Neural Network (ANN) dengan metode backpropagation dan epoch. ANN sendiri merupakan salah satu metode machine learning yang terdiri dari jaringan neuron yang saling berhubungan dalam lapisan-lapisan. Setiap neuron menerima informasi dari neuron lain, memprosesnya, dan mengirimkan kembali informasi tersebut ke neuron lain di lapisan di bawahnya.

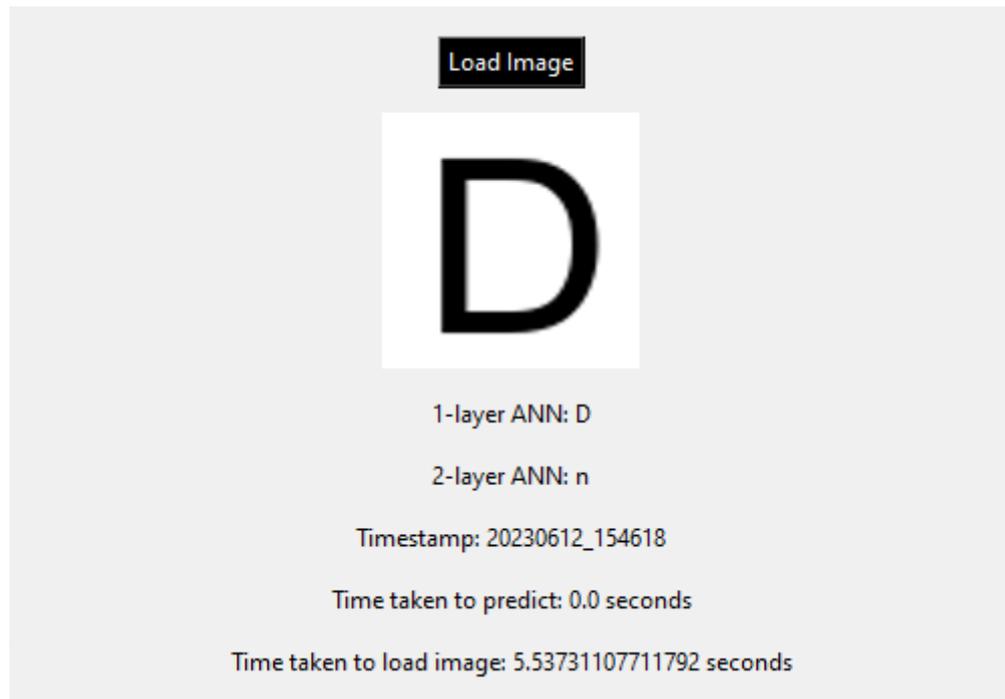
Langkah-langkah penelitian yang akan dilakukan setelah analisis terdahulu mencakup:

- a. Melakukan studi pustaka tentang teori-teori terkait algoritma ANN 1-layer, seperti prinsip dasar, arsitektur, metode pelatihan, dan aplikasi.
- b. Melakukan studi pustaka tentang teori-teori terkait algoritma ANN 2-layer, seperti prinsip dasar, arsitektur, metode pelatihan, dan aplikasi.
- c. Melakukan studi pustaka tentang kasus pengenalan karakter, termasuk teknik pengolahan gambar, ekstraksi fitur, dan teknik klasifikasi yang umum digunakan.

- d. Membuat rancangan komputasi untuk mengukur akurasi antara algoritma ANN 1-layer dan algoritma ANN 2-layer pada kasus pengenalan karakter, termasuk pembuatan kode program, pemilihan data latih dan data uji, serta evaluasi performa.
- e. Mengumpulkan data hasil perbandingan akurasi antara algoritma ANN 1-layer dan algoritma ANN 2-layer pada kasus pengenalan karakter.
- f. Membahas hasil perbandingan akurasi, termasuk analisis terhadap faktor-faktor yang mempengaruhi kinerja kedua algoritma, serta identifikasi kelebihan dan kekurangan dari masing-masing algoritma.
- g. Membuat kesimpulan berdasarkan hasil penelitian, termasuk rekomendasi untuk penelitian lebih lanjut dan aplikasi praktis dari algoritma yang lebih unggul dalam penelitian ini.

#### **4.2 Rancangan Pengujian**

Pada rancangan ini akan mengimplementasikan model jaringan saraf untuk klasifikasi multi-kelas. Kode ini mencakup fungsi untuk pengkodean satu-hot, standarisasi data input, fungsi aktivasi *softmax*, dan inisialisasi parameter. Kode ini juga mencakup fungsi untuk membagi data menjadi set pelatihan dan pengujian serta mencetak distribusi label mereka. Kode ini menggunakan *library Pandas* untuk membaca dataset dari file CSV dan mengembalikan data pelatihan dan pengujian beserta label-labelnya. Model jaringan saraf dapat dilatih dan diuji pada masalah klasifikasi multi-kelas apa pun dengan memanggil fungsi yang sesuai dengan parameter yang diperlukan. *Library Numpy* digunakan untuk menghasilkan fitur dalam kode yang akan dibuat. Gambar dimuat dan diubah menjadi skala abu-abu, ukurannya diubah menggunakan *library Pillow*. Kemudian model ANN yang sudah dilatih, kami digunakan untuk memprediksi gambar.



Gambar 4. 1 Desain Antarmuka Grafis Uji Gambar Huruf

Program ini menggunakan antarmuka grafis (GUI) dengan library Tkinter di Python. Ketika program dijalankan, GUI window akan muncul dengan tombol "Load Image" untuk memuat gambar. Setelah gambar dimuat, program akan melakukan prediksi karakter menggunakan model artificial neural network (ANN) dengan 1 layer dan 2 layer. Hasil prediksi karakter ditampilkan pada GUI. Program juga mencatat timestamp waktu saat gambar dimuat, serta menghitung waktu yang dibutuhkan untuk memuat gambar dan melakukan prediksi. Pengguna dapat memuat gambar baru dengan menekan tombol "Load Image" untuk mengulangi proses.

#### 4.2.1 Spesifikasi Kebutuhan Sistem Baru

Pengembangan sistem baru merupakan sebuah upaya yang penting dalam memenuhi kebutuhan bisnis atau organisasi. Untuk mencapai tujuan tersebut, diperlukan pemahaman yang jelas dan komprehensif mengenai spesifikasi kebutuhan sistem baru. Pada sejumlah sub-bab berikut ini akan menjelaskan secara rinci spesifikasi dalam enam hal yang meliputi proses, data, perangkat lunak, perangkat keras, pengguna, dan dokumen pendukung.

Dengan memahami dan menguraikan setiap aspek ini dengan detail, diharapkan sistem baru yang akan dikembangkan dapat berjalan sesuai dengan kebutuhan dan tujuan yang telah ditetapkan.

#### **4.2.1.1 Spesifikasi Proses**

Spesifikasi proses pada program ini mencakup serangkaian langkah yang dilakukan dalam pengenalan karakter pada gambar. Dalam pengenalan karakter, program menggunakan pendekatan berbasis jaringan saraf tiruan (ANN) untuk melakukan klasifikasi huruf berdasarkan gambar yang dimuat oleh pengguna.

1. Program memberikan pengguna opsi untuk memuat gambar melalui jendela dialog file. Setelah gambar dimuat, langkah pertama dalam proses ini adalah mengubah gambar menjadi vektor fitur menggunakan fungsi "image\_to\_feature\_vector". Fungsi ini mengambil gambar sebagai input dan menghasilkan representasi numerik dari gambar tersebut. Hal ini penting untuk mengubah gambar menjadi bentuk yang dapat diproses lebih lanjut oleh jaringan saraf.
2. Setelah vektor fitur diperoleh, langkah selanjutnya adalah melakukan standarisasi vektor fitur menggunakan fungsi "Standardize". Standarisasi ini bertujuan untuk memastikan bahwa nilai-nilai dalam vektor fitur memiliki skala yang seragam, sehingga tidak ada atribut yang mendominasi dalam proses pengenalan karakter.
3. Setelah proses standarisasi, program melanjutkan dengan propagasi maju (forward propagation) melalui jaringan saraf tiruan. Propagasi maju dilakukan dengan menggunakan fungsi "forwardProp", yang mengambil vektor fitur sebagai input, serta parameter-parameter jaringan yang telah diinisialisasi sebelumnya. Dalam tahap ini, jaringan saraf menghitung nilai-nilai aktivasi pada setiap lapisan, mulai dari lapisan input hingga lapisan output. Hasil prediksi huruf kemudian dihasilkan menggunakan fungsi aktivasi softmax pada lapisan output, yang menghasilkan distribusi probabilitas untuk setiap kelas huruf.

4. Selanjutnya, program melakukan perhitungan gradien parameter-parameter jaringan. Fungsi "gradient" digunakan untuk menghitung gradien berdasarkan perbedaan antara prediksi yang dihasilkan dengan label yang benar. Gradien ini penting dalam proses pelatihan jaringan, di mana parameter-parameter jaringan diperbarui secara iteratif untuk meminimalkan selisih antara prediksi dan label yang benar.
5. Terakhir, program menampilkan hasil prediksi huruf yang dihasilkan oleh jaringan saraf pada antarmuka pengguna (GUI). Selain itu, program juga menyediakan fitur untuk menyimpan gambar yang dimuat bersama dengan timestamp, serta melakukan tangkapan layar GUI dengan timestamp yang sesuai. Hal ini memungkinkan pengguna untuk melacak hasil prediksi dan mengelola data gambar yang telah diolah.

Dengan demikian, spesifikasi proses pada program ini mencakup langkah-langkah seperti pemuatan gambar, pengubahan gambar menjadi vektor fitur, standardisasi vektor fitur, propagasi maju dalam jaringan saraf tiruan, perhitungan gradien, dan tampilan hasil prediksi serta fitur-fitur tambahan untuk menyimpan gambar dan tangkapan layar.

#### 4.2.1.2 Spesifikasi Data

Dalam spesifikasi data, penjelasan akan fokus pada data yang dibutuhkan oleh program untuk melakukan pengenalan karakter melalui pendekatan berbasis jaringan saraf tiruan (ANN). Berikut ini adalah spesifikasi data yang dibutuhkan:

Tabel 4. 1 Data Analisis

No.	Spesifikasi Data	Deskripsi
1	Format Gambar	Program dapat memuat gambar dalam format yang umum digunakan seperti PNG.
2	Resolusi Gambar	Gambar harus memiliki resolusi yang spesifik sesuai dengan dataset yang dibuat.
3	Tipe Karakter	Program ini dapat mengenali karakter huruf kapital (A-Z) dan huruf kecil (a-z). Data yang digunakan harus terdiri dari karakter-karakter ini.
4	Jumlah Karakter	Program ini telah diatur untuk mengenali 52 karakter (26 huruf kapital dan 26 huruf kecil). Oleh karena itu, data yang digunakan harus mencakup setidaknya satu contoh dari setiap karakter.

5	Variasi Karakter	Data harus mencakup variasi karakter untuk setiap huruf. Misalnya, huruf 'A' harus muncul dalam berbagai gaya tulisan atau font.
6	Dataset Training	Program ini memerlukan dataset training yang berisi contoh-contoh gambar dengan label yang sesuai. Setiap gambar harus memiliki label yang menunjukkan karakter yang terkandung di dalamnya.
7	Dataset Testing	Program ini juga memerlukan dataset testing yang terpisah dari dataset training. Dataset ini digunakan untuk menguji kinerja model setelah proses training.
8	Data Preprocessing	Sebelum proses training, data perlu melalui tahap preprocessing. Ini termasuk perubahan gambar menjadi vektor fitur, standarisasi data, dan pemisahan dataset training dan testing.

#### 4.2.1.3 Spesifikasi Perangkat Lunak

Dalam spesifikasi perangkat lunak, program ini dikembangkan menggunakan bahasa pemrograman Python dengan menggunakan berbagai tools dan library untuk mencapai fungsionalitas yang diinginkan. Berikut adalah spesifikasi perangkat lunak yang digunakan:

1. Bahasa Pemrograman: Program ini ditulis menggunakan bahasa pemrograman *Python*. *Python* dipilih karena bahasa ini memiliki sintaks yang sederhana dan mudah dipahami, serta memiliki dukungan yang luas dalam pengembangan aplikasi berbasis data dan pengolahan gambar.
2. *Tools PyCharm*: Pengembangan program ini menggunakan *PyCharm*, yang merupakan sebuah *Integrated Development Environment (IDE)* khusus untuk bahasa pemrograman *Python*. *PyCharm* menyediakan berbagai fitur pengembangan yang mempermudah proses penulisan, *debugging*, dan pengujian program.
3. *Library dan Modul*:
  - a. ``pickle``: *Library* ini digunakan untuk melakukan serialisasi dan deserialisasi objek *Python*. Dalam program ini, ``pickle`` digunakan untuk menyimpan dan memuat model *Artificial Neural Network (ANN)* yang telah dilatih.
  - b. ``numpy``: *Library* ini menyediakan dukungan untuk operasi numerik dan manipulasi array multidimensi. Dalam program ini, ``numpy`` digunakan untuk

melakukan operasi matematika pada vektor fitur dan matriks parameter dalam ANN.

- c. ``tkinter``: *Library* ini digunakan untuk membuat antarmuka grafis pengguna (GUI) dalam program. Dalam program ini, ``tkinter`` digunakan untuk membuat jendela aplikasi, tombol, label, dan elemen GUI lainnya.
- d. ``filedialog`` dari modul ``tkinter``: Modul ini menyediakan fungsi untuk dialog pemilihan file. Dalam program ini, ``filedialog`` digunakan untuk memungkinkan pengguna memilih gambar yang akan dikenali karakternya.
- e. ``os``: Modul ini menyediakan fungsi-fungsi untuk berinteraksi dengan sistem operasi. Dalam program ini, ``os`` digunakan untuk membuat direktori dan melakukan operasi pada sistem file.
- f. ``PIL`` (*Python Imaging Library*): *Library* ini menyediakan berbagai fungsi untuk memanipulasi gambar. Dalam program ini, ``PIL`` digunakan untuk memuat, *resize*, menyimpan, dan memanipulasi gambar-gambar yang akan dikenali karakternya.
- g. ``time``: Modul ini menyediakan fungsi untuk mengukur waktu dalam program. Dalam program ini, ``time`` digunakan untuk menghitung waktu yang dibutuhkan dalam proses pengenalan karakter.

Dengan menggunakan bahasa pemrograman *Python* dan berbagai *tools* serta *library* yang disebutkan di atas, program ini dapat dikembangkan dengan lebih efisien dan dapat mengakses fungsionalitas yang diperlukan untuk pengenalan karakter menggunakan jaringan saraf tiruan.

#### 4.2.1.4 Spesifikasi Perangkat Keras

Dalam pengembangan perangkat lunak, spesifikasi perangkat keras merupakan informasi penting untuk memastikan bahwa program dapat berjalan dengan baik dan optimal. Spesifikasi perangkat keras mencakup komponen fisik yang digunakan dalam sebuah sistem komputer, seperti prosesor, *Random Access Memory* (RAM), sistem operasi, dan sebagainya.

Berikut adalah tabel dengan spesifikasi perangkat keras yang lebih lengkap:

Tabel 4. 2 Spesifikasi Perangkat Keras

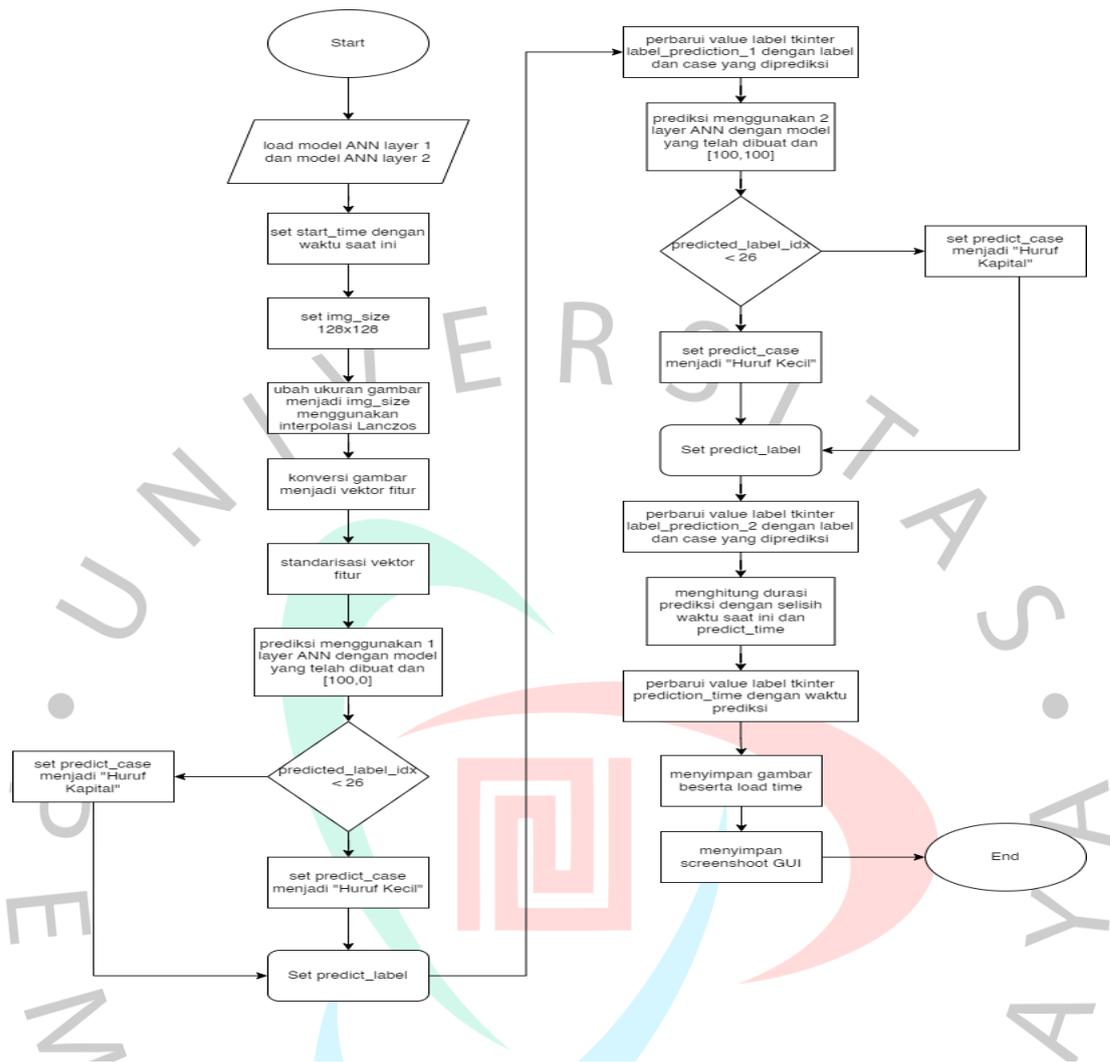
Komponen	Spesifikasi
Processor	Intel(R) Core (TM) i5-3360M CPU @ 2.80GHz
Kecepatan:	2.80 GHz
Installed RAM	16.0 GB
System Type	64-bit operating system, x64-based processor
Edition	Windows 10 Pro Version 21H2

#### 4.2.1.5 Spesifikasi Pengguna

Dalam pengembangan perangkat lunak, penting untuk memahami spesifikasi pengguna atau pengguna target yang akan menggunakan program. Spesifikasi pengguna mencakup informasi tentang karakteristik pengguna, kebutuhan, preferensi, dan batasan yang perlu dipertimbangkan dalam desain dan pengembangan perangkat lunak. Dalam konteks ini, spesifikasi pengguna program ini adalah pengguna yang dapat memanfaatkan antarmuka grafis yang dibuat dalam penelitian ini. Penelitian ini mengklasifikasikan pengguna sebagai pengguna akhir (end user). Pengguna akhir memiliki hak akses ke antarmuka grafis untuk melakukan prediksi gambar huruf sesuai dengan model-model yang telah dilatih sebelumnya.

#### 4.2.1.6 Flowchart

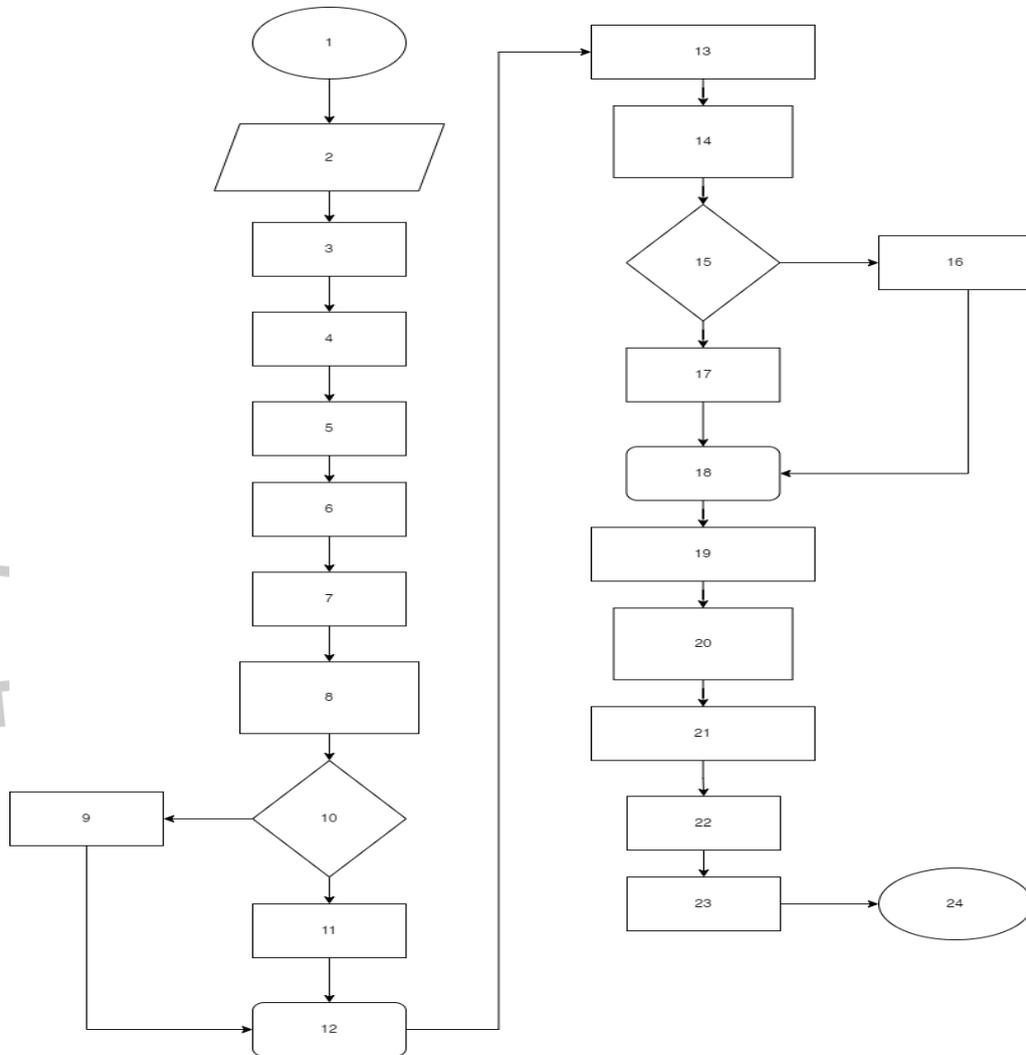
Flowchart membantu untuk memahami dan menjelaskan logika dan urutan eksekusi dari suatu program atau proses. Flowchart sering digunakan dalam pemrograman komputer untuk merancang, memecahkan masalah, dan mendokumentasikan algoritma.



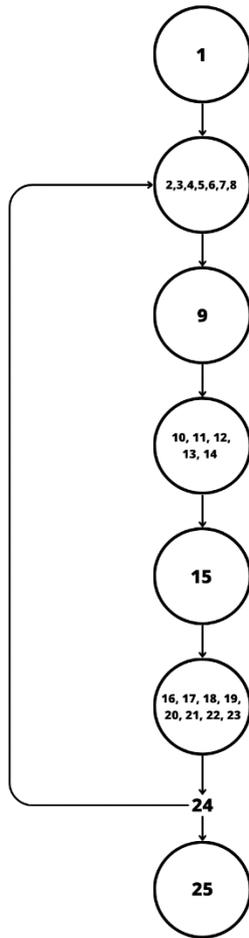
Gambar 4. 2 Flowchart Algoritma Prediksi Gambar Huruf

Flowchart ini menggambarkan alur operasi suatu program atau sistem yang melibatkan penggunaan model jaringan saraf tiruan (ANN) dengan dua layer. Langkah awal adalah memuat model ANN untuk kedua layer tersebut. Kemudian, dilakukan pengaturan waktu awal menggunakan variabel `start_time`. Gambar yang diinputkan akan diubah ukurannya sesuai dengan `img_size` menggunakan interpolasi Lanczos. Selanjutnya, dilakukan standarisasi vektor fitur gambar. Prediksi pertama dilakukan menggunakan model ANN dengan satu layer dan input `[100,0]`. Jika hasil prediksi menunjukkan nilai `predicted_label_idx` kurang dari 26, maka kasus prediksi akan ditetapkan sebagai "Huruf Kapital", dan jika lebih besar atau sama dengan 26, maka akan ditetapkan sebagai "Huruf Kecil". Setelah itu, dilakukan prediksi kedua menggunakan model ANN dengan dua layer dan input

[100,100]. Hasil prediksi ini juga akan menentukan kasus prediksi berdasarkan nilai `predicted_label_idx`. Durasi prediksi dihitung dengan mengurangi waktu saat ini dengan `predict_time`, dan hasilnya akan ditampilkan. Terakhir, dilakukan penyimpanan tangkapan layar GUI sebelum program berakhir.



Gambar 4. 3 Flowchart Algoritma Prediksi Gambar Huruf



Gambar 4. 4 Algoritma Prediksi Gambar Huruf

Dengan total jumlah node sebanyak 8 dan edge sebanyak 9, kita dapat menghitung Cyclomatic complexity ( $V(G)$ ) menggunakan rumus

$$V(G) = E - N + 2 \quad (4.1)$$

$V(G)$  : Cyclomatic complexity

$E$  : Total Edge

$N$  : Total Node

Jadi perhitungan flowgraph diatas, adalah sebagai berikut ini:

$$V(G) = 9 \text{ Edge} - 8 \text{ Node} + 2$$

$$= 3$$

Hasil perhitungan menunjukkan bahwa Cyclomatic complexity dari flowgraph ini adalah 3. Dengan demikian, terdapat 3 *Independent Path* dalam flowgraph tersebut. Berikut adalah tiga *Independent Path* yang dapat diidentifikasi dari flowgraph:

a. *Independent Path 1*:

1 - 2 - 9 - 10 - 15 - 16 - 24 - 2 - 9 - 10 - 15 - 25

b. *Independent Path 2*:

1 - 2 - 9 - 10 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 2 - 9 - 10 - 15 - 25

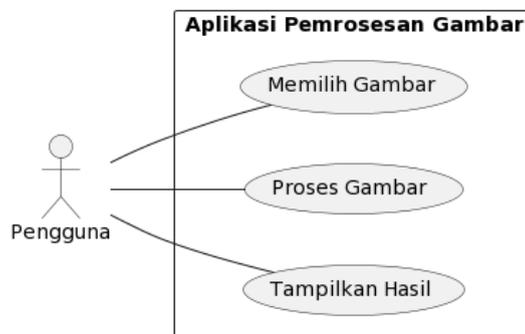
c. *Independent Path 3*:

1 - 2 - 9 - 10 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 2 - 9 - 10 - 15 - 16 - 24 - 2 - 9 - 10 - 15 - 25

Masing-masing jalur independen mengikuti urutan langkah-langkah dalam flowgraph dan mencakup semua node yang diperlukan.

#### 4.2.1.7 Use Case

Use case adalah deskripsi dari interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem yang sedang dibangun. Use case menjelaskan apa yang dapat dilakukan oleh aktor dan respons sistem terhadap interaksi tersebut. Use case juga dapat digambarkan dalam bentuk diagram Use Case yang menggambarkan aktor, use case, dan hubungan antara keduanya.



Gambar 4. 5 Use Case Antarmuka Grafis Prediksi Huruf

Tabel 4. 3 Skenario Use Case Prediksi Huruf

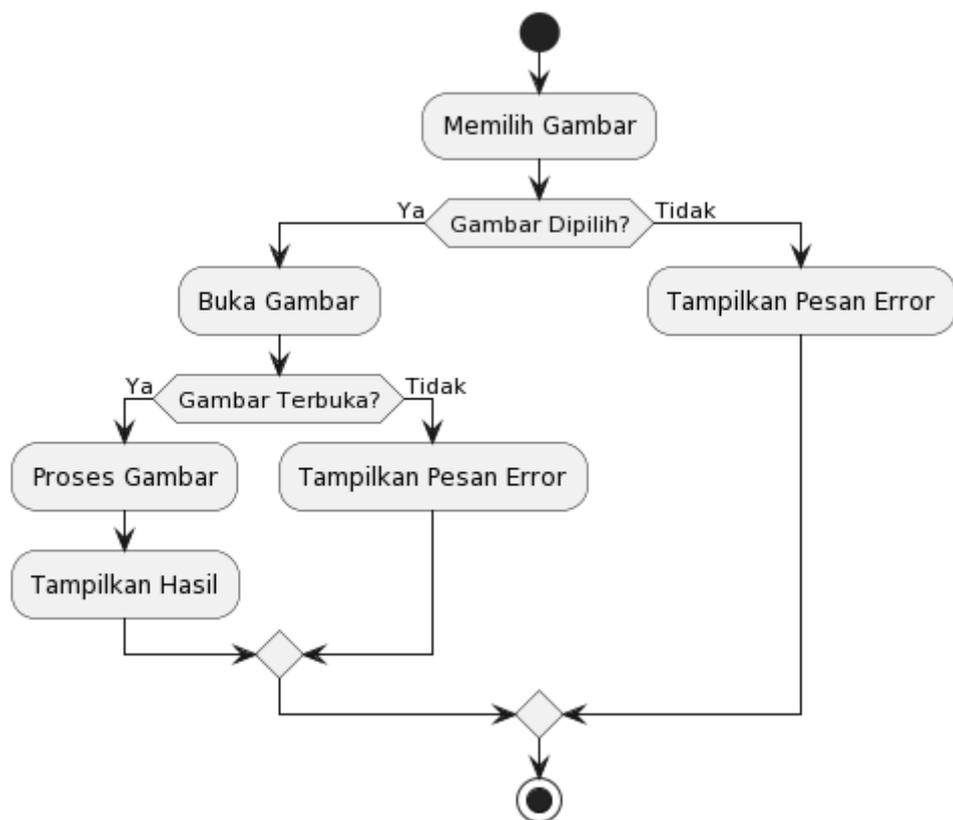
Nama Use Case	Antarmuka Grafis Prediksi huruf
Deskripsi	Use case ini menggambarkan interaksi pengguna dengan aplikasi pengenalan karakter melalui antarmuka grafis (GUI). Pengguna dapat memuat gambar yang mengandung karakter, melakukan pengenalan karakter, dan melihat hasil prediksi.
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Pengguna meluncurkan aplikasi pengenalan karakter.</li> <li>2. Pengguna memuat gambar dengan mengklik tombol "Load Image" di antarmuka pengguna.</li> <li>3. Sistem menampilkan jendela dialog pemilihan file, memungkinkan pengguna memilih gambar yang mengandung karakter.</li> <li>4. Pengguna memilih gambar yang diinginkan dan mengklik tombol "Open" di jendela dialog pemilihan file.</li> <li>5. Sistem memproses gambar yang dimuat dan menampilkan gambar tersebut di antarmuka pengguna.</li> <li>6. Sistem melakukan pengenalan karakter dengan menerapkan model jaringan saraf tiruan (ANN) yang telah terlatih pada gambar tersebut.</li> <li>7. Sistem menampilkan hasil prediksi dalam bentuk label karakter dan keterangan apakah karakter tersebut huruf kapital atau huruf kecil.</li> <li>8. Pengguna dapat melihat timestamp, yaitu waktu ketika gambar dimuat.</li> <li>9. Pengguna dapat melihat waktu yang diperlukan oleh sistem untuk memuat gambar dan melakukan prediksi karakter.</li> <li>10. Pengguna dapat menyimpan gambar hasil pengenalan karakter dengan mengeklik tombol "Save" di antarmuka pengguna.</li> <li>11. Sistem menyimpan gambar dengan menambahkan timestamp sebagai bagian dari nama file dan menampilkan pesan sukses.</li> <li>12. Pengguna dapat mengulangi proses dengan memuat gambar lain atau keluar dari aplikasi dengan menutup jendela antarmuka pengguna.</li> </ol>
Trigger	Pengguna mengklik tombol "Load Image" di antarmuka pengguna.

Use case "Antarmuka Grafis Prediksi huruf" merupakan interaksi antara pengguna dengan aplikasi pengenalan karakter melalui antarmuka grafis (GUI). Pengguna dapat memuat gambar yang mengandung karakter, melakukan pengenalan karakter, dan melihat hasil prediksi. Langkah-langkahnya adalah sebagai berikut: pengguna meluncurkan aplikasi dan memuat gambar melalui tombol "Load Image". Sistem menampilkan jendela pemilihan file, pengguna memilih gambar yang diinginkan, dan sistem memproses gambar tersebut. Hasil

prediksi karakter ditampilkan dan pengguna dapat melihat timestamp serta waktu yang diperlukan untuk memuat gambar dan melakukan prediksi. Pengguna juga dapat menyimpan gambar hasil pengenalan karakter dan mengulangi proses dengan memuat gambar lain atau keluar dari aplikasi.

#### 4.2.1.8 Activity Diagram

*Activity* diagram menggambarkan alur aktivitas atau proses yang terjadi dalam konteks *Use Case* tersebut. Dalam *Use Case*, kita mengidentifikasi tujuan dari interaksi pengguna dengan sistem dan langkah-langkah yang harus dilakukan untuk mencapai tujuan tersebut. *Activity* diagram digunakan untuk memvisualisasikan langkah-langkah tersebut secara lebih rinci, termasuk bagaimana pengguna dan sistem berinteraksi, tindakan yang dilakukan, dan aliran informasi antara mereka. Berikut adalah *activity* diagram berdasarkan *use case* yang telah dibuat sebelumnya



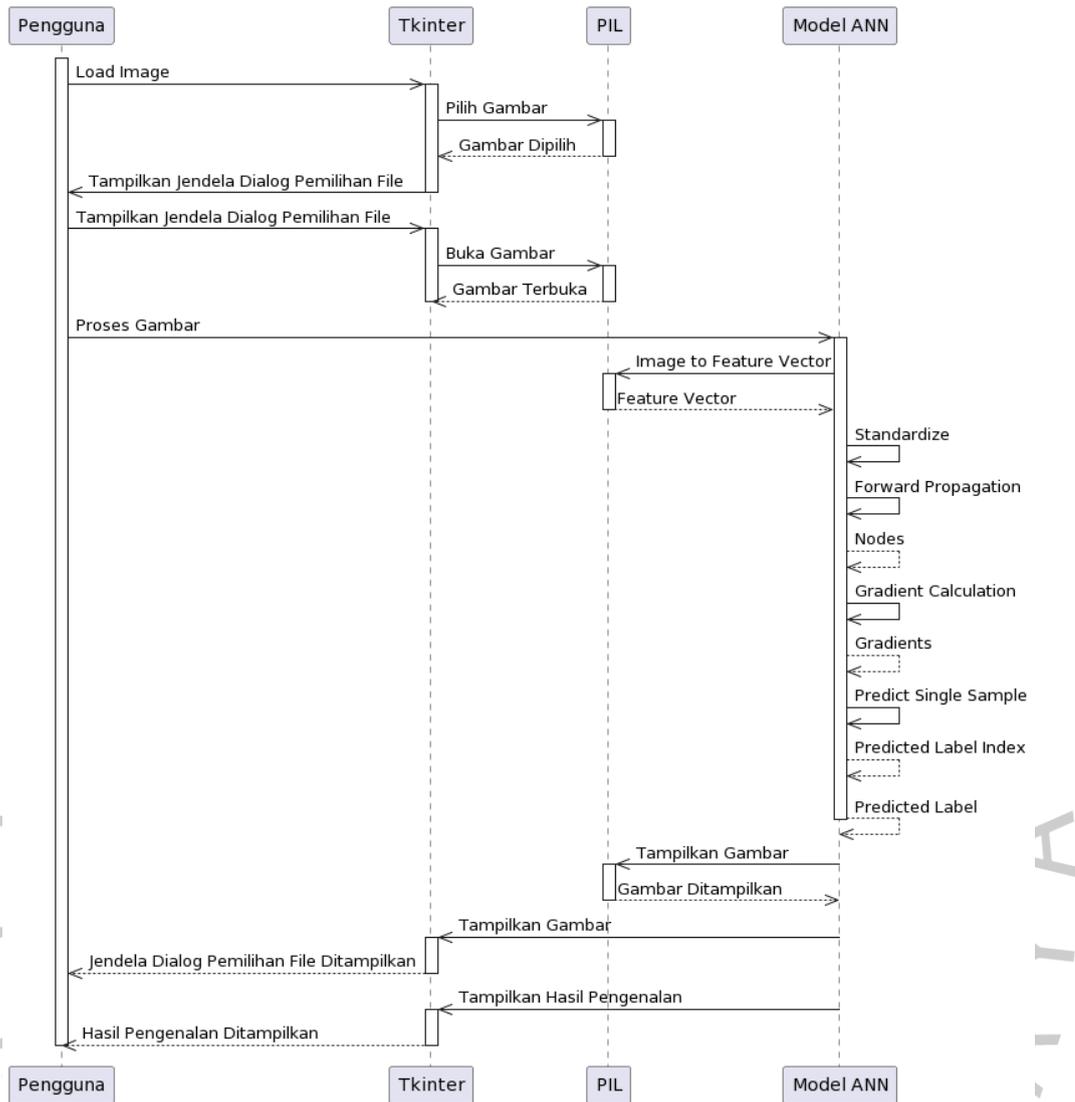
Gambar 4. 6 *Activity* Diagram Prediksi Huruf

Pada *activity* diagram ini, proses dimulai dengan pemilihan gambar oleh pengguna. Jika gambar dipilih (jawaban "Ya" pada pengujian kondisi), gambar tersebut dibuka untuk diproses. Jika gambar terbuka (jawaban "Ya" pada pengujian kondisi), proses pemrosesan gambar dilakukan, dan hasilnya ditampilkan kepada pengguna. Jika gambar tidak terbuka atau tidak dipilih (jawaban "Tidak" pada pengujian kondisi), pesan error ditampilkan.

*Activity* diagram ini memberikan gambaran alur utama dari proses pemrosesan gambar dengan model ANN. Namun, Anda dapat menambahkan detail atau langkah-langkah tambahan sesuai dengan kebutuhan spesifik sistem yang sedang Anda buat.

#### **4.2.1.9 Sequence Diagram**

*Sequence* diagram sering digunakan untuk memodelkan logika bisnis, aliran kontrol, dan proses interaksi dalam sistem perangkat lunak. Diagram ini membantu dalam pemahaman yang lebih baik tentang hubungan antara objek-objek dalam sistem dan memvisualisasikan bagaimana pesan-pesan dikirim dan diterima di antara mereka.



Gambar 4. 7 Sequence Diagram Algoritma Prediksi Huruf

Sequence diagram diatas menggambarkan urutan interaksi antara beberapa partisipan yang terlibat dalam proses pengolahan gambar menggunakan model Artificial Neural Network (ANN). Berikut adalah penjelasan langkah demi langkah dari sequence diagram tersebut:

1. Pengguna memulai proses dengan melakukan pemilihan gambar melalui antarmuka pengguna (Tkinter) dengan mengirimkan pesan "Load Image" kepada Tkinter.
2. Tkinter menerima pesan "Load Image" dan mengirimkan pesan "Pilih Gambar" kepada PIL (Python Imaging Library) untuk meminta pengguna memilih gambar.

3. PIL menerima pesan "Pilih Gambar" dan menunggu aksi pengguna. Setelah pengguna memilih gambar, PIL mengirimkan pesan "Gambar Dipilih" kembali kepada Tkinter.
4. Tkinter menerima pesan "Gambar Dipilih" dari PIL dan melanjutkan dengan menampilkan jendela dialog pemilihan file kepada pengguna.
5. Setelah jendela dialog pemilihan file ditampilkan, pengguna kembali melakukan interaksi dengan Tkinter dengan memilih jendela dialog pemilihan file untuk membuka gambar.
6. Tkinter mengirim pesan "Buka Gambar" kepada PIL untuk membuka gambar yang dipilih oleh pengguna.
7. PIL menerima pesan "Buka Gambar" dan membuka gambar yang dimaksud. Setelah gambar terbuka, PIL mengirim pesan "Gambar Terbuka" kepada Tkinter.
8. Tkinter menerima pesan "Gambar Terbuka" dari PIL dan menampilkan gambar yang terbuka kepada pengguna.
9. Pengguna melakukan proses pemrosesan gambar dengan mengirim pesan "Proses Gambar" kepada Model ANN.
10. Model ANN menerima pesan "Proses Gambar" dan melakukan serangkaian operasi, seperti mengubah gambar menjadi vektor fitur dengan pesan "Image to Feature Vector" kepada PIL, melakukan standarisasi, melakukan propagasi maju (forward propagation), menghitung gradien, dan melakukan prediksi pada sampel tunggal dengan pesan-pesan yang sesuai kepada dirinya sendiri.
11. Setelah proses pemrosesan selesai, Model ANN mengirim pesan "Tampilkan Gambar" kepada PIL untuk menampilkan gambar yang telah diproses.
12. PIL menerima pesan "Tampilkan Gambar" dan menampilkan gambar tersebut.
13. Model ANN mengirim pesan "Tampilkan Gambar" kepada Tkinter untuk menampilkan gambar kepada pengguna.
14. Tkinter menerima pesan "Tampilkan Gambar" dan menampilkan gambar kepada pengguna melalui jendela dialog pemilihan file.

15. Model ANN mengirim pesan "Tampilkan Hasil Pengenalan" kepada Tkinter untuk menampilkan hasil pengenalan yang diperoleh dari pemrosesan gambar.
16. Tkinter menerima pesan "Tampilkan Hasil Pengenalan" dan menampilkan hasil pengenalan kepada pengguna.
17. Setelah semua interaksi selesai, partisipan Pengguna, Tkinter, dan Model ANN dinonaktifkan.

#### 4.2.1.10 Perancangan Pengujian Black Box

Pada pengujian black box, pengujian dilakukan dari sudut pandang pengguna atau pengamat eksternal yang tidak memiliki pengetahuan rinci tentang bagaimana sistem bekerja di dalamnya. Pengujian dilakukan dengan memasukkan input tertentu ke dalam sistem dan memeriksa output yang dihasilkan, serta membandingkannya dengan output yang diharapkan sesuai dengan persyaratan.

Tabel 4. 4 Tabel Pengujian Black Box

No.	Fungsi Pengujian	Hasil yang Diinginkan
1	Load Image	Gambar berhasil di-load dan ditampilkan dalam antarmuka grafis.
2	Predict Image	Prediksi huruf berhasil dilakukan berdasarkan gambar yang di-load.
3	Save Image	Gambar berhasil disimpan dengan timestamp sesuai waktu simpan.
4	Update Timestamp in Frame	Timestamp dan waktu load berhasil di-update pada antarmuka grafis.

#### 4.2.1.11 Perancangan Pengujian White Box

Tujuan utama dari pengujian white box adalah untuk memeriksa kebenaran implementasi sistem dan memastikan bahwa semua jalur kode telah diuji secara efektif. Teknik pengujian white box sering digunakan untuk memvalidasi struktur program, mengidentifikasi cacat logika atau kesalahan pemrograman, serta meningkatkan cakupan pengujian untuk mencakup semua bagian kode yang mungkin dieksekusi.

Tabel 4. 5 Tabel Pegujian White Box

NO	FUNGSI	HASIL YANG DIHARAPKAN
1	image_to_feature_vector	Mengembalikan vektor fitur gambar
2	Standardize	Mengembalikan matriks vektor fitur yang telah distandarisasi
3	SoftmaxFunction	Mengembalikan vektor output softmax
4	Sigmoid	Mengembalikan vektor output sigmoid
5	forwardProp	Mengembalikan array yang berisi nilai setiap node dalam jaringan

