



Prototype Permainan Cerdas HTML 5 Berbasis Artificial Neural Network (ANN)

Pitrasaha Adytia¹, Wahyuni², Theresia Retno Widian³

¹Sistem Informasi, STMIK Widya Cipta Dharma

^{2,3}Teknik Informatika, STMIK Widya Cipta Dharma

pitra@wicida.ac.id

Abstract

Multimedia is currently widely used as a medium of learning, especially in kindergarten. As in educational games that are often found today, they have a passive nature, which only clicks every button on the game so there is no difference. In this study, the use of an artificial network is used as a knowledge center for artificial intelligence in the guessing fruit game. The right artificial network to become the center of artificial intelligence because the artificial intelligence artificial network that continues to evolve is getting smarter. To overcome this problem, an artificial neural network (ANN) algorithm is needed to make educational games smarter. The game uses HTML as the user interface. The test method uses the Confusion Matrix test which can provide precision information, recall, *fi-score*, and Black Box Testing to see that the functions of the buttons in the game can work properly. Describing the accuracy of how accurate the model is in classifying correctly. In this study, an accuracy value of 97% means that the model can classify fruit images well. During the image prediction process, it takes quite a long time, this is because the model recognizes the pattern/shape of each type of fruit. The more types of fruit used, the longer the model takes to predict.

Keywords: Artificial Neural Network, Smart Game, Fruit Classification

Abstrak

Multimedia pada saat ini banyak digunakan sebagai media pembelajaran terutama pada taman kanak-kanak. Seperti pada *game* edukasi yang sering dijumpai pada saat ini memiliki sifat yang pasif, yang hanya mengklik setiap tombol yang ada pada permainan sehingga tidak memiliki perbedaan. Pada penelitian ini pemanfaatan jaringan syaraf tiruan digunakan sebagai pusat pengetahuan bagi kecerdasan buatan pada permainan cerdas tebak buah. Jaringan syaraf tiruan dirasa tepat untuk menjadi pusat kecerdasan buatan karena jaringan syaraf tiruan menawarkan kecerdasan buatan yang terus berevolusi menjadi semakin pandai. Untuk mengatasi masalah tersebut maka diperlukan algoritma *artificial neural network* (ANN) untuk membuat *game* edukasi menjadi lebih cerdas. Permainan ini menggunakan HTML sebagai user interface. Metode pengujiannya menggunakan pengujian *Confusion Matrix* yang dapat memberikan informasi *precision*, recall, *fi-score* dan pengujian *Black Box* untuk melihat fungsi dari button-button yang ada pada permainan dapat berfungsi dengan baik. Akurasi menggambarkan seberapa akurat model dalam mengklasifikasi dengan benar. Pada penelitian ini model menghasilkan nilai akurasi sebesar 97% artinya model dapat mengklasifikasikan gambar buah dengan baik. Pada saat melakukan proses prediksi gambar membutuhkan waktu yang cukup lama hal tersebut disebabkan model melakukan pengenalan pola/bentuk setiap jenis buah. Semakin banyak jenis buah yang digunakan maka semakin lama model melakukan proses prediksi.

Kata kunci: Artificial Neural Network, Permainan Cerdas, Klasifikasi Buah

1. Pendahuluan

Multimedia merupakan salah satu media penyampaian pembelajaran dapat meningkatkan nilai pembelajaran bagi siswa, sehingga sangat dibutuhkan adanya perangkat multimedia yang mampu menarik minat belajar serta meningkatkan kemampuan dan kreativitas siswa. Terutama pada taman kanak-kanak seperti pada permainan *game* edukasi belajar alfabet dan *game* edukasi lainnya. Namun pada permainan *game* edukasi memiliki sifat pasif yang hanya mengenalkan huruf alfabet dan cara penyebutan

huruf alfabet dengan cara mengklik tombol yang ada dan setiap permainan *game* edukasi saat ini menggunakan pola yang sama sehingga tidak memiliki perbedaan.

Pada penelitian ini pemanfaatan jaringan syaraf tiruan/*Artificial neural network* (ANN) digunakan sebagai pusat pengetahuan bagi kecerdasan buatan pada permainan cerdas tebak buah. Jaringan syaraf tiruan dirasa tepat untuk menjadi pusat kecerdasan buatan karena jaringan syaraf tiruan menawarkan kecerdasan buatan yang terus berevolusi. Beberapa penelitian telah

menunjukkan penggunaan ANN sebagai agen cerdas dalam permainan [1]–[3].

Artificial Neural Network adalah sebuah bentuk perhitungan yang meniru bentuk jaringan syaraf pada makhluk hidup [4]. Sama seperti pada jaringan syaraf asli, jaringan syaraf tiruan memiliki neuron untuk memproses *input/output*. Karena itulah, pada sebagian besar kasus, ANN merupakan sistem yang adaptif, karena struktur sistemnya dapat berubah seiring dengan berubahnya informasi internal ataupun eksternal yang diproses pada saat *learning phase* (tahap pembelajaran).

ANN telah digunakan untuk mengklasifikasikan jenis batu bara dan menghasilkan tingkat akurasi sebesar 98% diujikan sebanyak 200 data [4]. ANN juga berhasil mengklasifikasikan jenis buah mangga dengan tingkat akurasi 95.31% menggunakan sebanyak 192 data [5]. Selain itu model ANN juga telah digunakan untuk pengujian penyakit bacterial spot sebanyak 50 data, penyakit yellow leaf curl virus sebanyak 50 data citra dan daun sehat sebanyak 50 yang menghasilkan akurasi sebesar 78%. Memerlukan waktu selama 319,77 detik untuk memproses data [6].

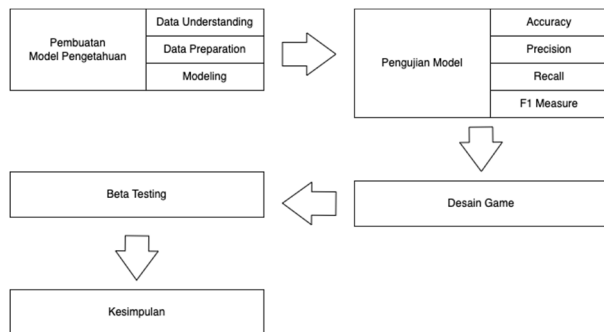
Penelitian ini ingin menerapkan ANN untuk klasifikasi jenis buah, dimana model akhir diharapkan dapat diterapkan dan diujikan pada sebuah game tebak gambar buah. Dengan model yang terbentuk, permainan tebak buah akan lebih dinamis dalam menyajikan pilihan gambar buah bagi pengguna. Program tidak lagi harus secara manual untuk menebak klasifikasi buah. Dengan menerapkan model klasifikasi diharapkan kode program permainan menjadi lebih efisien dengan mengurangi pernyataan bercabang dan kondisional yang sangat besar. Dalam penelitian ini prototype permainan difokuskan pada : (1) Permainan ini diperuntukkan untuk anak usia. (2) Permainan ini bersifat *offline* dan *single player* yang dimainkan di pc atau laptop. (3) Permainan ini mengenalkan 17 jenis buah dan hanya menampilkan bagian luar buah saja. (4) Permainan ini hanya dapat mengklasifikasikan jenis buah yang digunakan pada penelitian ini.

Tujuan penelitian ini adalah untuk menguji model ANN untuk diterapkan dalam sebuah permainan. Model yang dibentuk kemudian diimplementasikan dalam sebuah prototype permainan tebak buah bagi taman kanak-kanak. Arsitektur dan desain pemanfaatan model dalam permainan cerdas juga dibahas dalam penelitian ini.

2. Metode Penelitian

Usulan metode penelitian dapat dilihat tahapannya pada gambar 1. Dimana ada 4 langkah utama di mulai dari melakukan pembuatan model pengetahuan, dilanjutkan dengan pengujian model, desain game, serta dilakukan

black box testing dan terakhir adalah menarik kesimpulan penelitian



Gambar 1. Alur penelitian

Langkah awal penelitian ini adalah dengan mengembangkan dan membuat model pengetahuan. Dimana Langkah yang dilakukan mengadopsi dari SKKNI yang terdiri dari beberapa tahapan dimulai dari *data understanding*, *data preparation* dilanjutkan dengan pembuatan dan evaluasi model [7].

Model yang telah dibentuk, kemudian dilakukan pengujian untuk dapat melihat performansinya. Adapun metrik yang digunakan dalam pengujian model ada 4 yaitu *accuracy*, *precision*, *recall* dan *f1-score*.

Model yang telah dibentuk kemudian akan digunakan didalam aplikasi game. Untuk itu perlu dibuat desain dan arsitektur aplikasi game untuk dapat menggunakan model yang telah dibuat. Model digunakan untuk melakukan prediksi tebakan gambar buah didalam game.

Hasil dari game yang telah dibuat kemudian perlu diujikan. Salah satu metode pengujian yang dapat dilakukan adalah dengan menggunakan metode black box testing.

Hasil dari tahapan tahapan mulai dari pembuatan model pengetahuan hingga diimplementasikan dalam bentuk game, akan ditarik kesimpulan yang dapat digunakan bagi penelitian selanjutnya

3. Hasil dan Pembahasan

3.1. Permodelan

Langkah awal mengambil data dari kaggle.com, dan menyiapkan *tools* dan *library* yang dibutuhkan, kemudian melakukan proses augmentasi data gambar. Selanjutnya melakukan *pra training* model densenet201 dan mendefinisikan *layer-layer* pada model *Artificial Neural Network*. Setelah proses *training* selesai, model di simpan dengan format h5. Masuk ketahap evaluasi model dengan melakukan testing pada dataset uji. Dan tahap terakhir yaitu melakukan *deployment*.

Data untuk pembelajaran mesin diambil dari Kaggle.com data asli berisi 120 jenis buah, tetapi di sini hanya 17 jenis buah yang akan digunakan jadi ada 17 kelas yang berbeda. Total data yaitu sebanyak 2.768 gambar yang terbagi menjadi 3 kategori yaitu 2.220 sebagai training data lalu 850 gambar sebagai data testing dan 548 gambar digunakan sebagai data validasi. Jumlah *dataset* gambar setiap buah yang dapat dilihat dalam tabel 1

Tabel 1

No	Nama Buah	Jumlah Gbr Training	Jumlah Gbr Testing
1	Alpukat	187	50
2	Anggur	175	50
3	Apel	180	50
4	Buah Naga	144	50
5	Cherry	167	50
6	Durian	156	50
7	Jeruk	183	50
8	Kelengkeng	197	50
9	Kiwi	119	50
10	Mangga	95	50
11	Manggis	180	50
12	Nanas	110	50
13	Pisang	189	50
14	Rambutan	188	50
15	Salak	132	50
16	Semangka	173	50
17	Strawberry	185	50

Proses augmentasi data dilakukan dimana gambar akan diperbarui dalam kumpulan data secara artificial untuk meningkatkan ukuran himpunan data training agar lebih bervariasi.

Program Augmentasi Gambar

```
train_generator =
    ImageDataGenerator(
        rescale = 1./225,
        shear_range=0.15,
        zoom_range=0.15,
        rotation_range=30,
        width_shift_range=0.2,
        height_shift_range=0.2,
        horizontal_flip=True,
        validation_split=0.2,
        fill_mode='nearest'
    )
```

Membuat *flow* data dimana folder sumber data berasal dimensi dari gambar diubah menjadi 244 x 244 pixel dan banyaknya gambar yang dimasukkan kedalam setiap tahapan *training* sebanyak 32 gambar dengan menggunakan *categorical*.

Program Membuat Flow Data

```
train_images =
    train_generator.flow_from_directory(
        data_dir,
        batch_size=32,
        target_size=(224, 224),
```

```
        class_mode='categorical',
        subset='training'
    )
val_images =
    train_generator.flow_from_directory(
        data_dir,
        batch_size=32,
        target_size=(224, 224),
        class_mode='categorical',
        subset='validation'
    )
```

Melakukan *pratraining* model Densenet201 sebagai transfer learning yaitu menggunakan data gambar sebagai inputan untuk melakukan prediksi atau klasifikasi

Program Pratraining Densenet201

```
pretrained_model=
    tf.keras.applications.DenseNet201(
        input_shape=(224, 224, 3),
        include_top=False,
        weights='imagenet',
        pooling='avg'
    )
```

Kemudian mendefinisikan *layer-layer artificial neural network* pada penelitian ini menggunakan algoritma *backpropagation* merupakan pelatihan yang terawasi dengan menggunakan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyi. *inputs* adalah model yang sudah didefinisikan sebelumnya sebagai *layer input*, di *layer* kedua yaitu *hidden layer* dengan 128 neuron, pada *layer* terakhir dengan 17 neuron jumlah dari kelas yang akan diklasifikasi. Metode optimasi menggunakan adam, *loss* adalah metode pengukuran nilai *loss* berdasarkan nilai menggunakan *categorical_crossentropy*, nilai matriks yang diukur menggunakan nilai akurasi sebagai nilai pengukurannya

Mendefinisikan layer-layer pada model ANN

```
inputs = pretrained_model.input
x=tf.keras.layers.Dense(
    128,activation='relu')(pretrained_model.output)

outputs=tf.keras.layers.Dense(17,
    activation='softmax')(x)

model=tf.keras.Model(inputs=inputs,
    outputs=outputs)

model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy'])
```

Melakukan *model.fit* pada *train_images* merupakan data gambar *training*, *val_images* adalah *validation* data yang dievaluasi pada setiap *epochs* berakhir, banyaknya proses iterasi atau pengulangan pada *training* yaitu 50 *epochs*. *EarlyStopping* adalah panggilan balik bawaan yang dirancang untuk penghentian awal untuk mencegah proses pelatihan terlalu lama.

Program Melatih Model

```
history = model.fit(train_images,
                    validation_data=val_images,
                    epochs=50,
                    callbacks=[
                        tf.keras.callbacks.EarlyStopping(
                            monitor='val_loss',patience=3)
                    ])
]
```

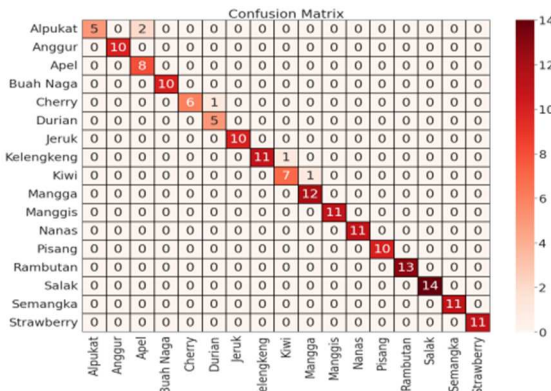
Kemudian, membagi data menjadi 80% data *training* dan data *testing* 20% untuk mempermudah proses *testing*,

Import StandardScaler yang merupakan *library* dari *sklearn.preprocessing* untuk menormalisasi data supaya hasil akurasi lebih akurat kemudian membuat variabel *scaler* yang akan memanggil kelas *StandardScaler()*, fungsi *scaler.fit(X_train)* dimana *fit* ini pada kelas *StandardScaler()* akan dipanggil untuk melatih algoritma pada data latih yang ada pada parameter *X_train*.

Lalu import estimator *MLPClassifier* pada *library* *sklearn.neural_network* untuk melatih model, menggunakan *hidden_layer_sizes* untuk menentukan jumlah *layer* dan jumlah *node* yang ingin dimiliki *dineural network*, *max_iter* untuk menunjukkan jumlah *epochs*, lalu *activation* sebagai fungsi aktivasi untuk lapisan tersembunyi, disini *solver* parameter yang menentukan algoritma untuk optimasi bobot pada *node*

3.2. Pengujian Model

Mendefinisikan variabel *y_pred*, untuk memprediksi nilai kelas berdasarkan model yang di latih. Kemudian mencoba melakukan prediksi pada *X_test*..



Gambar 2. Confusion Matrix

Dalam visualisasi ini, memiliki dua bagian yang telah diuraikan bagian klasifikasi yang diprediksi berisi 17 sub bagian, untuk masing-masing kelas yang ingin diklasifikasikan dan bagian klasifikasi aktual yang memiliki 17 sub bagian untuk tiap kelas. Dapat dilihat pada gambar 2 visualisasi *confusion matrix*.

Setelah memvisualisasikan *confusion matrix* peneliti dapat mengetahui keakuratan dari model yang sudah

dibuat menggunakan *performance metris* seperti berikut ini.

Precision ini mewakili jumlah prediksi model dimana gambar diklasifikasi positif yang sebenarnya adalah benar. *Recall* ini mewakili jumlah prediksi model dimana aktual positif diklasifikasi dengan benar. *F1-Score* adalah rata-rata *precision* dan *recall*. *Support* adalah jumlah sampel data uji yang digunakan untuk *testing*.

Macro avg adalah rata-rata *precision*, *recall*, *f1-score* perkelas tanpa mempertimbangkan proporsi. *Weighted avg* adalah rata-rata *precision*, *recall*, *f1-score* perkelas dengan mempertimbangkan proporsi.

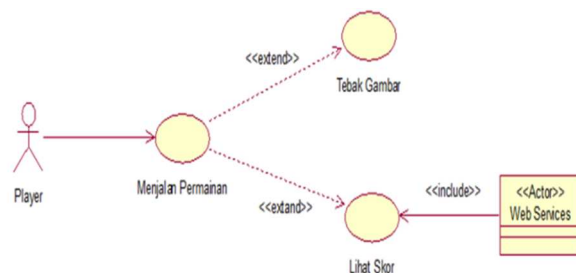
Accuracy adalah ukuran kinerja yang akan memberikan tingkat keakuratan dari keseluruhan model atau semua prediksi yang benar dari total jumlah data. Nilai akurasi yang dihasilkan pada penelitian ini sebesar 97% dari 850 gambar data *testing* dengan aktual kelas sebanyak 17 jenis buah. Dapat dilihat pada gambar *classification report*.

	precision	recall	f1-score	support
Alpukat	1.00	0.71	0.83	7
Anggur	1.00	1.00	1.00	10
Apel	0.80	1.00	0.89	8
Buah Naga	1.00	1.00	1.00	10
Cherry	1.00	0.86	0.92	7
Durian	0.83	1.00	0.91	5
Jeruk	1.00	1.00	1.00	10
Kelengkeng	1.00	0.92	0.96	12
Kiwi	0.88	0.88	0.88	8
Mangga	0.92	1.00	0.96	12
Manggis	1.00	1.00	1.00	11
Nanas	1.00	1.00	1.00	11
Pisang	1.00	1.00	1.00	10
Rambutan	1.00	1.00	1.00	13
Salak	1.00	1.00	1.00	14
Semangka	1.00	1.00	1.00	11
Strawberry	1.00	1.00	1.00	11
accuracy			0.97	170
macro avg	0.97	0.96	0.96	170
weighted avg	0.97	0.97	0.97	170

Gambar 3. Performance Metric

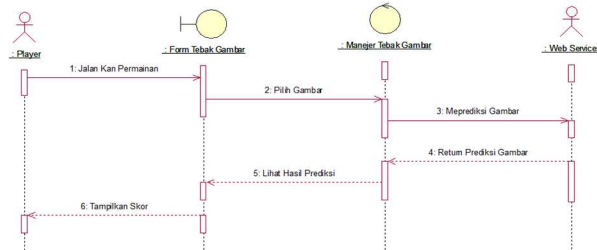
3.3. Desain Game

Pada saat player pertama kali menjalankan permainan maka akan muncul gambar buah pada layer dan sebuah perintah untuk menebak gambar setelah player selesai memilih gambar dapat mengklik button lihat skor maka gambar diprediksi. Dapat dilihat pada gambar 2 use case diagram permainan cerdas tebak buah



Gambar 4. Usecase Game Tebak Gambar

Pada saat *player* menjalankan permainan maka akan diarahkan ke *user interface* berupa *form* tebak gambar, kemudian *player* memilih gambar lalu *form* tebak gambar akan mengirim untuk diproses, lalu di kirim ke *web services* untuk diprediksi kemudian *web services* mengirim kembali hasil prediksi berupa skor untuk ditampilkan dihalaman skor untuk dilihat *player*. Dapat dilihat pada gambar 3 sequence diagram permainan cerdas tebak buah



Gambar 5. Sequence Diagram Game Tebak Gambar

Design Interface Permainan Cerdas Tebak Buah. Pada Gambar 5. Pada saat *player* menjalankan permainan maka akan masuk ke halaman tebak gambar, terdapat tampilan perintah untuk menebak gambar dan pilihan untuk memilih gambar pada *layer*. Dapat dilihat pada gambar 5 *design interface* permainan cerdas tebak buah.



Gambar 6. Tampilan Game

4. Kesimpulan

Penelitian ini menunjukkan bahwa model yang dibentuk menggunakan ANN memiliki tingkat akurasi yang tinggi yaitu 97%. Model yang dibentuk sukses di implementasikan dalam bentuk game berbasis HTML5. Waktu yang diperlukan untuk 1 kali prediksi gambar memakan waktu cukup lama sehingga mengganggu pengguna ketika bermain game. Penelitian selanjutnya perlu dilakukan cara optimasi waktu prediksi agar lebih cepat atau menggunakan algoritma klasifikasi lainnya yang memakan waktu lebih sebentar.

Daftar Rujukan

- [1] S. Y. D. Tawas, G. S. Budhi, dan K. R. Purba, "Pembuatan Survival Action Game Dengan Non-Player Character Berbasis Neural Network," *Jurnal Infra*, vol. 4, no. 2, hlm. 251-255., 2016.
- [2] R. H. Kusumodestoni, A. Sucipto, S. N. Ismiati, dan M. N. Abid, "Penerapan Algoritma Backpropagation Pada Game Pengenalan Nahwu Di Mi Darul Falah Jepara," *POSITIF : Jurnal Sistem dan Teknologi Informasi*, vol. 5, no. 2, hlm. 103, Des 2019, doi: 10.31961/positif.v5i2.831.
- [3] H. Yudha Prasetya, M. Aminul Akbar, dan I. Arwani, "Penerapan Neural Network untuk NPC Braking Decision pada Racing Game," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 5, hlm. 4148-4154, Mei 2019.
- [4] D. F. Haryati, G. Abdillah, dan A. I. Hadiana, "Klasifikasi Jenis Batubara Menggunakan Jaringan syaraf Tiruan Dengan Algoritma Backpropagation," *Seminar Nasional Teknologi Informasi dan Komunikasi*, 2016.
- [5] J. Jamaludin, C. Rozikin, dan A. S. Y. Irawan, "Klasifikasi Jenis Buah Mangga dengan Metode Backpropagation," *Techné : Jurnal Ilmiah Elektroteknika*, vol. 20, no. 1, hlm. 1-12, Mar 2021, doi: 10.31358/techné.v20i1.231.
- [6] A. W. Putri, "Implementasi Artificial Neural Network (ANN) Backpropagation Untuk Klasifikasi Jenis Penyakit Pada Daun Tanaman Tomat," *MATHunesa: Jurnal Ilmiah Matematika*, vol. 9, no. 2, hlm. 344-350, Agu 2021, doi: 10.26740/mathunesa.v9n2.p344-350.
- [7] Menteri Ketenagakerjaan Republik Indonesia, *Penetapan Standar Kompetensi kerja Nasional Indonesia*.