

Prediksi Pertumbuhan Kangkung Hidroponik Dengan *Smart Farming* Menggunakan Metode *Supervised Learning*

1st Dimas Fadilah Permana R
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

dimasfadilah@student.telkomuniversity.ac.id

2nd Favian Dewanta
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

favian@telkomuniversity.ac.id

3rd Sri Astuti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

sriastuti@telkomuniversity.ac.id

Abstrak—Metode penanaman tumbuhan kangkung secara hidroponik sangat efektif dilakukan pada lahan padat penduduk seperti perkotaan. Pada kangkung hidroponik membutuhkan nutrisi yang ditambahkan pada air. Pemberian nutrisi jika dilakukan secara berlebih akan menyebabkan tumbuhan kangkung menjadi jenuh dan akhirnya mati. *Sonic Bloom* dilakukan pada penelitian ini untuk mengurangi pemberian nutrisi yang berlebih. Pada percobaan *Sonic Bloom* yang digunakan adalah musik bergenre Dangdut, Jazz, Murattal, dan Tanpa Musik. Kemudian pertumbuhan di monitoring menggunakan sensor suhu, suhu air, kamera yang dibuat dengan sistem *Internet of Things*. Data dari sensor kemudian dikumpulkan menjadi dataset untuk digunakan prediksi pertumbuhannya. Pertumbuhan Kangkung hidroponik dengan perlakuan *Sonic Bloom* kemudian di prediksi menggunakan model *Machine Learning*. Model dibuat dengan menggunakan algoritma *Decision Tree*, *Random Forest*, dan *K-Nearest Neighbour*. Dan didapatkan hasil pengujian model terbaik yaitu *Random Forest* pada rasio *splitting* 40% data tes dan 60% data training dengan nilai akurasi, presisi, recall, dan *f1-score* 99% dengan waktu komputasi 1,4 detik.

Kata kunci—Kangkung Hidroponik, *Sonic Bloom*, *Machine Learning*, *Smart Farming*.

I. PENDAHULUAN

Kangkung merupakan tanaman dengan masa panen yang tergolong cepat dengan waktu sekitar 25-30 hari. Kangkung banyak dijumpai pada masakan di Indonesia. Pada lahan perkotaan yang padat penduduk hanya memiliki lahan yang sempit. Menanam dengan cara hidroponik sangat cocok diterapkan pada lahan sempit[1]. Akan tetapi sering dijumpai pemberian nutrisi yang berlebih pada hidroponik dapat menyebabkan tumbuhan menjadi jenuh dan akhirnya mati. Pengurangan pemberian nutrisi dapat diatasi dengan teknologi *Sonic Bloom*. *Sonic Bloom* merupakan pancaran gelombang frekuensi tinggi[2]. Kangkung hidroponik yang diberi pancaran *Sonic Bloom* kemudian dapat dipantau dari jarak jauh menggunakan *Internet of Things*. Data hasil pemantauan kemudian dapat digunakan untuk pembuatan model *Machine Learning* untuk memprediksi pertumbuhan kangkung hidroponik berteknologi *Sonic Bloom* dimasa mendatang.

II. KAJIAN TEORI

Kajian teori akan membahas mengenai *Sonic Bloom*, *Internet of Things*, *Decision Tree*, *Random Forest*, dan *K-Nearest Neighbour*.

A. Sonic Bloom

Sonic Bloom merupakan gelombang frekuensi tinggi dan suara termasuk kedalamnya. Pada penelitian sebelumnya, suara dengan frekuensi 3.500 – 5000 Hz dapat merangsang pembukaan stomata pada daun. Musik dapat digunakan untuk *Sonic Bloom* dengan memutar pada tingkat kebisingan dan genre musik yang berbeda [3]. Pemaparan musik pada tumbuhan bisa membuka stomata lebih lebar sehingga CO₂, H₂O dan unsur hara lainnya dapat diserap lebih banyak. Dan pada akhirnya akan meningkatkan produktivitas dari tanaman [4].

B. Internet of Things

Internet of Things merupakan konsep teknologi dimana kita dapat memantau sebuah obyek jarak jauh dengan menggunakan jaringan internet. *Internet of Things* terdiri dari sensor yang merupakan *hardware* yang digunakan untuk menangkap obyek yang akan dipantau. Kemudian data dari sensor terhubung dengan mikrokontroler dan diolah sesuai algoritmanya. Mikrokontroler yang digunakan mempunyai modul *WiFi* yang dapat terhubung ke internet. Data kemudian dikirim ke database dan bisa divisualisasikan melalui website maupun aplikasi gawai[5].

C. Decision Tree

Decision Tree merupakan salah satu algoritma klasifikasi *Supervised Learning*. Dengan sistem pemecahan kelas seperti pohon dimana terdapat akar, ranting, dan daun. Akar disini merupakan pembagian kelas yang pertama. Pada *Decision tree* penentuan kelas ditentukan berdasarkan kriteria *Entropy* dan *Gini*. *Entropy* merupakan tingkat ketidak-teraturan suatu data.

Data dengan tingkat ketidakteraturan yang masih tinggi akan terus di bagi sampai *entropy* yang rendah. Sedangkan dengan menggunakan kriteria penentuan kelas *gini* adalah dengan membagi menjadi dua dengan tingkat kesenjangan yang berbeda. Jika nilai kesenjangan masih tinggi maka akan

dibagi menjadi kelas-kelas sampai tingkat kesenjangan yang rendah[6].

D. Random Forest

Random Forest merupakan algoritma klasifikasi penyempurna dari kekurangan *Decision Tree*. *Decision Tree* mempunyai kelemahan pada saat klasifikasi data yang sangat banyak. Konsep dari *Random Forest* adalah dengan membandingkan jumlah voting dari hasil kelas *Decision Tree*. Dimana hasil penentuan dari *Decision Tree* terbanyak maka *Random Forest* menyimpulkan bahwa jumlah terbanyak merupakan kelas yang benar[7].

E. K-Nearest Neighbour

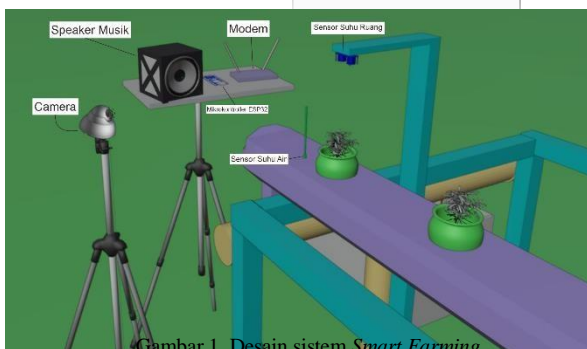
Konsep kerja dari algoritma *K-Nearest Neighbour* adalah dengan mencari kemiripan dari tetangga terdekatnya. Pada saat mencari kemiripan ini tentunya harus dibandingkan dengan beberapa tetangganya. Jumlah tetangga terdekatnya ditentukan dengan nilai K. Nilai K ini dapat ditentukan dengan mencari nilai error terendah tiap K. Kemudian dari nilai K ini dapat diketahui jumlah tetangganya dan dapat ditentukan jumlah kemiripan terbanyak. Jika ada data yang belum diketahui kelasnya, dan tetangga terdekatnya tiga buah kotak dan satu segitiga. Maka *K-Nearest Neighbour* dapat menyimpulkan bahwa data tersebut merupakan sebuah kotak [8].

III. METODE

Metode yang digunakan pada sistem pengambilan dan pengolahan data adalah sebagai berikut:

A. Model Pengambilan Data

Desain sistem *Internet of Things* yang digunakan untuk mengambil data dapat ditunjukkan pada gambar dibawah.



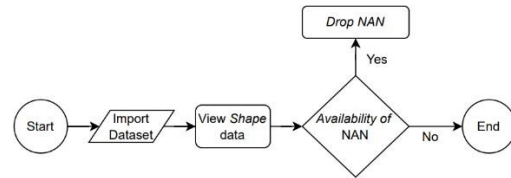
Gambar 1. Desain sistem *Smart Farming*.

Gambar 1 menggambarkan desain sistem *smart farming* yang dibuat. Terdapat sensor suhu ruang untuk memantau suhu ruang disekitar tumbuhan. Terdapat juga sensor suhu air untuk memantau suhu air pada hidroponik, kemudian kamera digunakan untuk menataui tinggi tanaman. Digunakan juga speaker untuk memancarkan *Sonic Bloom* ke tanaman. Mikrokontroller dan *router* digunakan untuk mengirimkan data ke jaringan internet.

B. Pengolahan Data

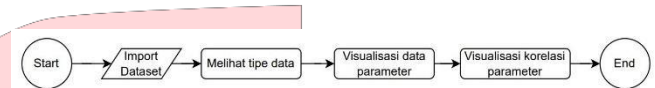
Hasil dari data dari sensor akan digunakan untuk membuat dataset. Dataset yang dibentuk kemudian digunakan untuk membuat model prediksi pertumbuhan Kangkung.

Sebelum dataset digunakan untuk melakukan *training model*, dataset akan dicek terlebih dahulu melalui proses *preprocessing* dengan *flowchart* sebagai berikut:



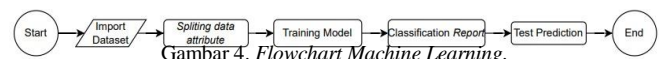
Gambar 2. *Flowchart preprocessing*.

Gambar 2 merupakan *Flowchart preprocessing* yang diawali dengan *import* dataset. Kemudian dilanjutkan dengan melihat jumlah kolom dan baris. Setelah itu pengecekan apakah terdapat data kosong atau tidak. Jika proses *preprocessing* berakhir dilanjutkan dengan *Exploratory Data Analysis* untuk menentukan *feature* dan label yang akan digunakan.



Gambar 3. *Flowchart Exploratory Data Analysis*.

Gambar 3 merupakan *Flowchart Exploratory Data Analysis* yang dimulai dengan *import* dataset kemudian dilanjutkan dengan melihat tipe data untuk label yang akan digunakan. Untuk melihat persebaran data dapat divisualisasikan dengan *library Matplotlib*. Terakhir adalah melihat korelasi untuk memilih *feature* yang akan digunakan.



Gambar 4. *Flowchart Machine Learning*.

Gambar 4 merupakan *Flowchart Machine Learning* yang akan digunakan untuk membuat model diawali dengan *import* dataset yang sudah melewati *preprocessing*. Kemudian dataset akan di bagi menjadi *data training* dan *data test* sesuai dengan rasio yang ditentukan. Kemudian model ditraining dengan parameter yang sudah ditentukan. Performa dari model dapat dilihat pada *classification report*. Jika performa model sudah bagus dapat dicoba untuk memprediksi data baru.

IV. HASIL DAN PEMBAHASAN

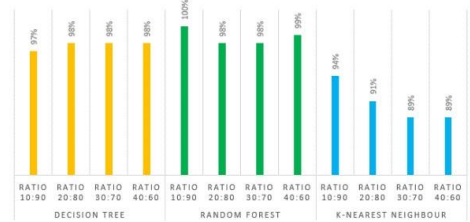
Pada saat *training model*, *feature* yang digunakan adalah tinggi, suhu runag, suhu air, jenis musik, dan label yang digunakan adalah hari.

Model	Parameter	Nilai
Decision Tree	<i>criterion</i>	<i>entropy</i>
	<i>max_depth</i>	20
	<i>min_samples_leaf</i>	5
	<i>random_state</i>	42
Random Forest	<i>n_estimators</i>	50
	<i>criterion</i>	<i>entropy</i>
	<i>min_samples_split</i>	6
	<i>min_samples_leaf</i>	1
	<i>max_depth</i>	40
<i>K-Nearest Neighbour</i>	<i>random_state</i>	42
	<i>n_neighbors</i>	3

Tabel 1. Hasil parameter *tunning* untuk *training model*.

Kemudian parameter *training* model harus di *tunning* terlebih dahulu. Hasil *tunning* parameter dapat dilihat pada Tabel 1 merupakan hasil parameter *tunning* yang kemudian digunakan untuk parameter *training* model pada setiap algoritma yang digunakan. Penentuan model terbaik yang akan digunakan dilakukan dengan membandingkan performa rasio *splitting* data, waktu komputasi antar model *Decision Tree*, *Random Forest*, dan *K-Nearest Neighbour*.

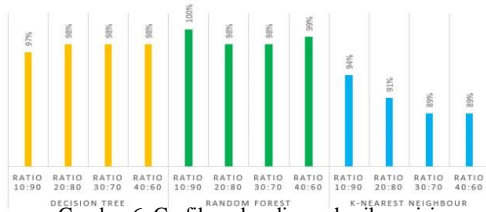
A. Performa Akurasi



Gambar 5. Grafik perbandingan hasil akurasi.

Gambar 5 menunjukkan bahwa hasil akurasi dari model *Decision Tree* meraih performa terbaik pada rasio *splitting* data 20:80, 30:70, dan 40:60. Kemudian model *Random Forest* meraih akurasi terbaik pada rasio *splitting* data 40:60. Dan model *K-Nearest Neighbour* meraih akurasi terbaik pada rasio *splitting* data 10:90.

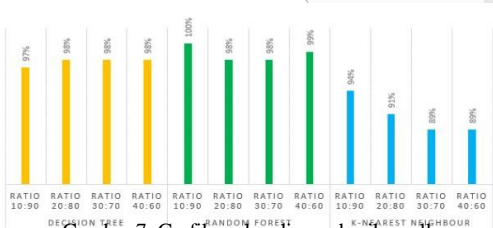
B. Performa Presisi



Gambar 6. Grafik perbandingan hasil presisi.

Gambar 6 menunjukkan bahwa hasil presisi dari model *Decision Tree* meraih presisi terbaik pada rasio *splitting* data 20:80, 30:70, dan 40:60. Kemudian model *Random Forest* meraih presisi terbaik pada rasio *splitting* data 40:60. Dan model *K-Nearest Neighbour* meraih presisi terbaik pada rasio *splitting* data 10:90.

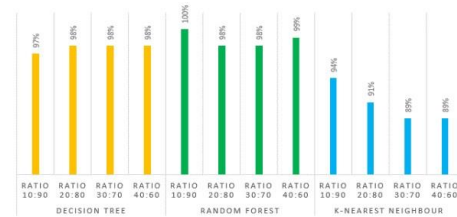
C. Performa Recall



Gambar 7. Grafik perbandingan hasil recall.

Gambar 7 menunjukkan bahwa hasil *recall* dari model *Decision Tree* meraih *recall* terbaik pada rasio *splitting* data 20:80, 30:70, dan 40:60. Kemudian model *Random Forest* meraih *recall* terbaik pada rasio *splitting* data 40:60. Dan model *K-Nearest Neighbour* meraih *recall* terbaik pada rasio *splitting* data 10:90.

D. Performa F1-Score



Gambar 8. Grafik perbandingan hasil *f1-score*.

Gambar 8 menunjukkan bahwa hasil *f1-score* dari model *Decision Tree* meraih *f1-score* terbaik pada rasio *splitting* data 20:80, 30:70, dan 40:60. Kemudian model *Random Forest* meraih *f1-score* terbaik pada rasio *splitting* data 40:60. Dan model *K-Nearest Neighbour* *f1-score* terbaik diraih pada rasio *splitting* data 10:90.

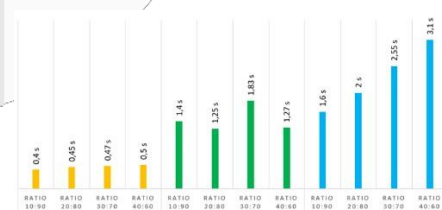
E. Perbandingan Akurasi Train dan Test



Gambar 9. Grafik perbandingan akurasi *train* dan *test*.

Selain hasil performa akurasi, presisi, *recall*, dan *f1-score*. Perbandingan akurasi prediksi model pada data *training* dan data *test* juga diperlukan untuk melihat model *overfitting* atau *underfitting*. Gambar 9 model *Decision Tree* tidak terjadi *overfitting* dan *underfitting*, dan meraih perbandingan terbaik pada rasio *splitting* data 20:80. Model *Random Forest* tidak mengalami *underfitting*, tetapi mengalami *overfitting* pada rasio 10:90. *Random Forest* meraih perbandingan terbaik pada rasio *splitting* data 40:60. *K-Nearest Neighbour* tidak mengalami *overfitting*. Tetapi dengan rasio *splitting* data yang semakin tinggi, hasil akurasi pada prediksi data *test* semakin rendah. Hal ini dapat disimpulkan semakin tinggi rasio *splitting* data pada model *K-Nearest Neighbour* maka model dapat terjadi *underfitting*. *K-Nearest Neighbour* meraih perbandingan terbaik pada rasio *splitting* data 10:90.

F. Perbandingan Waktu Komputasi



Gambar 10. Perbandingan waktu komputasi *training* model.

Hasil perbandingan waktu komputasi dapat dilihat pada Gambar 10. Perhitungan ini didapatkan dari menghitung waktu *training* antar model dan didapatkan hasil pada *training* model *Decision Tree* waktu komputasi tercepat diraih dengan waktu 0,4 detik pada rasio *splitting* data 10:90. Waktu komputasi tercepat pada *training* model *Random Forest* diraih dengan waktu 1,25 detik pada rasio *splitting* data 20:80. *K-Nearest Neighbour* meraih waktu komputasi tercepat dengan waktu 1,6 detik pada rasio *splitting* data 10:90.

V. KESIMPULAN

Dari hasil perbandingan performa dan waktu komputasi yang didapatkan dari masing – masing model. Dapat disimpulkan bahwa untuk memprediksi pertumbuhan Kangkung hidroponik dengan perlakuan *Sonic Bloom* dapat digunakan model *Machine Learning*. Training model terbaik yang dapat digunakan adalah algoritma *Random Forest* dengan rasio *splitting* data 40% untuk data test dan 60% untuk data training. Untuk waktu komputasi dapat diabaikan karena selisih waktu hanya 1 detik dari waktu komputasi tercepat.

VI. REFERENSI

- [1] I. S. Aminah, R. Rosmiah, H. Hawalid, L. Yuningsih, and H. Helmizuryani, "Penyuluhan budidaya tanaman sayur kangkung (*ipomoea reptans*) melalui sistem hidroponik di kelurahan alang-alang lebar kota Palembang," *Altifani Journal: International Journal of Community Engagement*, vol. 1, no. 1, pp. 46–52, 2020.
- [2] A. CHOLIK, "Pengaruh pemberian suara musik (klasik, hardcore, dan murottal) terhadap pertumbuhan vegetatif kangkung darat (*ipomoea reptans* poir)," *Fakultas Tarbiyah dan Keguruan Universitas Islam Negeri Raden Intan Lampung Tech. Rep.* 19361, 2022.
- [3] T. A. Yuwono, S. Sulistiadi, and D. Atmiasih, "Pengaruh teknologi ramah lingkungan sonic bloom menggunakan musik hard rock dan asmaul husna terhadap pertumbuhan kangkung (*ipomoea aquatic*)," *Mekanika*, vol. 2, no. 2, 2021.
- [4] S. A. Nio, J. A. Rumbay, P. S. Anggini, P. S. L. Supit, and D. P. M. Ludong, "Potensi metode sonic bloom untuk meningkatkan pertumbuhan tanaman," *Jurnal MIPA*, vol. 10, no. 2, pp. 76–80, 2021.
- [5] R. Piyare and S. R. Lee, "Smart home control and monitoring system using smart phone," *ICCA, ASTL*, vol. 24, pp. 83–86, 2013.
- [6] H. Sharma and S. Kumar, "A survey on decision tree algorithms of classification in data mining," *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 2094–2097, 2016.
- [7] J. Li, P.-a. Zhong, M. Yang, F. Zhu, J. Chen, W. Liu, and S. Xu, "Intelligent identification of effective reservoirs based on the random forest classification model," *Journal of Hydrology*, vol. 591, p. 125324, 2020.
- [8] A. Mucherino, P. J. Papajorgji, and P. M. Pardalos, "K-nearest neighbor classification," in *Data mining in agriculture*. Springer, 2009, pp. 83–106.

