

Implementasi Metode *K-Nearest Neighbor* pada Purwarupa Jemuran Otomatis berdasarkan Sensor Hujan dan Intensitas Cahaya

Pabela Purwa Wiyoga¹, Rizal Maulana², Agung Setia Budi³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹purwawiyoga@gmail.com, ²rizal_lana@ub.ac.id, ³agungsetiabudi@ub.ac.id

Abstrak

Indonesia yang beriklim tropis mendapatkan sinar matahari sepanjang tahun. Hal ini sangat bermanfaat bagi masyarakat untuk menjemur atau mengeringkan barang-barang rumah tangga maupun dalam kebutuhan industri. Contoh industri yang memanfaatkan sinar matahari adalah, produksi kerupuk, ikan asin, kopi, gabah dan yang lainnya. Tetapi, selain terus mendapatkan panas matahari, hujan juga datang tidak menentu. Sehingga membiarkan barang-barang diluar ruangan akan sangat berisiko, terutama ketika musim hujan. Karena itu peneliti membuat sebuah sistem jemuran yang secara otomatis dapat dikeluarkan ketika kondisi cerah dan tidak hujan, dan dimasukkan ketika kondisi gelap atau hujan. Kondisi tersebut akan diambil menggunakan *raindrop sensor* dan *light dependent resistor sensor*. Arduino Nano sebagai mikrokontroler yang digunakan untuk mengendalikan sensor dan aktuator dengan menggunakan metode *K-Nearest Neighbors*. Setelah dilakukan pengujian didapatkan hasil yang baik. Nilai akurasi dan waktu komputasi terbaik pada metode *K-Nearest Neighbors* adalah pada $k=3$ yaitu dengan akurasi terbesar 98% dan waktu komputasi sistem sebesar 37,173 mikrodetik. Ketika dibandingkan dengan metode *if-else*, metode *if-else* memiliki nilai akurasi 100% dan waktu komputasi 13,232 mikrodetik. Metode *K-Nearest Neighbor* layak diimplementasikan pada sistem ini namun *if-else* sederhana lebih menguntungkan dikarenakan akurasi yang lebih baik dan metode ini juga lebih sederhana dalam penerepan penulisan kode program.

Kata kunci: *k-nearest neighbor, kendali otomatis, jemuran otomatis*

Abstract

Indonesia, which has a tropical climate, gets sunshine all year round. It is very useful for people to dry household or industrial needs. Examples of industries that use sunlight are the production of crackers, salted fish, coffee, and unhulled rice and others. However, apart from getting the continuous heat of the sun, there is also rain that is present erratically. So leaving things outdoors will be risky, especially during the rainy season. Therefore, the researchers created a sun dry system that can be put outside automatically when the outdoor conditions are sunny and there is no rain, and inserted when the outdoor conditions are rainy or there is no sunlight. These conditions will be taken using a *raindrop sensor* and a *light dependent resistor sensor*. Arduino Nano as a microcontroller used to control sensors and actuators. To control the system using the *K-Nearest Neighbors* method. After testing, obtained good results. The best value of accuracy and computation time in using the *K-Nearest Neighbors* method is at $k=3$ with the average accuracy of 98% and the computation time is 37.173 microseconds. When compared with the *if-else* method, the *if-else* method has an accuracy value of 100% and a computation time of 13.232 microseconds. The *K-Nearest Neighbor* method is feasible to implement in this system, but a simple *if-else* method is more profitable due to better accuracy and simpler source code.

Keywords: *Nearest Neighbor, automatic control, automatic sun dry*

1. PENDAHULUAN

Indonesia adalah negara yang berada di garis katulistiwa, sehingga memiliki iklim tropik. Dengan demikian wilayah di daerah

tropik dalam setahun dilewati matahari sebanyak dua kali. Di daerah tropik selisih antara lama hari siang dan lama hari malam kecil (Wirjohamidjojo & Swarinoto, 2010). Yang berarti wilayah Indonesia mendapatkan panas

sinar matahari sepanjang tahun. Oleh sebab itu masyarakat Indonesia cukup bergantung pada panas matahari dalam kehidupan sehari-hari maupun dalam proses produksi perindustrian kecil dan menengah khususnya untuk menjemur atau mengeringkan barang.

Selain mendapat panas sepanjang tahun, Indonesia mendapatkan variasi musim. Variasi musim tersebut dapat jelas terlihat pada curah hujan. Kedua musim tersebut dibedakan dari banyaknya curah hujan (Wirjohamidjojo & Swarinoto, 2010). Variasi musim tersebut oleh masyarakat biasa disebut dengan musim kemarau dan musim penghujan. Sehingga ketika musim penghujan dan terjadi hujan maka barang-barang yang dijemur harus segera ditutup atau dimasukkan kedalam ruangan.

Algoritma K-Nearest Neighbor (KNN) adalah algoritma yang digunakan untuk melakukan klasifikasi terhadap suatu objek, berdasarkan k buah data latih yang jaraknya paling dekat dengan objek tersebut (Rivki & Bachtiar, 2017). Jika dibandingkan dengan pengklasifikasian secara if-else maka metode ini lebih rumit, baru dan cukup populer. Sehingga diharapkan bahwa dalam sistem yang menggunakan metode K-Nearest neighbor ini dapat memiliki hasil yang lebih baik dari pada menggunakan metode if-else. Sehingga dari uraian diatas perlu dirancang sebuah sistem otomatis yang dapat memasukkan jemuran ke tempat teduh dari hujan maupun gelap. Dimana dalam perancangan akan menggunakan sensor hujan dan sensor cahaya dan menggunakan metode K-Nearest Neighbor.

2. LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Tabel 1 Penelitian terkait

Judul	Keterkaitan	Hasil
Sistem Notifikasi Kondisi Cuaca Untuk Keselamatan <i>Take Off</i> Paralayang Menggunakan Metode Naïve Bayes (Studi Kasus: Paralayang Gunung Banyak, Batu)	Penggunaan Sensor hujan	Sensor hujan menunjukkan hasil baik
Alat Pengendali Atap Jemuran Otomatis Dengan Sensor Cahaya Dan Sensor Air Berbasiskan	Kesamaan sistem, menggunakan sensor <i>LDR</i> dan <i>Raindrop</i> ,	Sesuai dengan yang diharapkan.

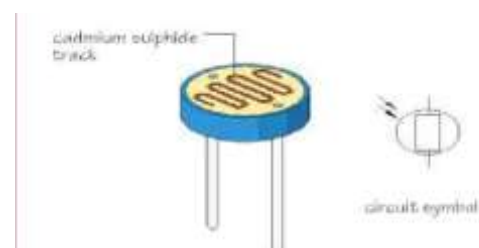
Mikrokontroler ATmega16	tidak menggunakan metode KNN	
<i>A Special Event-Based K-Nearest Neighbor Model for Short-Term Traffic State Prediction</i>	Metode <i>special event</i> KNN	Memiliki prediksi lebih baik dari pada metode machine learning yang lebih tradisional

2.2 Sensor hujan

Sensor raindrop mendeteksi adanya air hujan pada lapisan tembaga yang telah terpasang pada PCB apakah sedang hujan atau tidak, jika hujan maka sensor raindrop akan mengirim data ke Arduino (Utama, Mulyanto, Fauzi, & Putri, 2018). Sensor ini memiliki dua jenis output yaitu output digital dan output analog.

2.3 Sensor Intensitas Cahaya

LDR merupakan sensor yang dapat digunakan untuk menangkap input berupa intensitas cahaya yang berupa resistansi yang dikirimkan dari sensor ke mikrokontroler. Artinya pada saat cahaya redup, LDR menjadi konduktor yang buruk, atau bisa disebut juga LDR memiliki resistansi yang besar pada saat gelap atau cahaya redup (Mufida & Abas, 2017) disitasi dalam Damastuti & Syafi'i, 2016). Sehingga nilai resistansi dari sensor tersebut dapat digunakan sebagai input dari mikrokontroler, nilai yang diterima dari sensor ini berupa nilai ADC.



Gambar 1 LDR Sensor

2.4 Arduino Nano

Arduino Nano merupakan papan mikrokontroler yang berbasis AT Mega328. Mempunyai 14 digital input / output pin (dimana 6 dapat digunakan sebagai output PWM), 6 input analog, osilator kristal 16MHz, koneksi USB dan tombol reset (Isfarizky, Fardian, & Mufti, 2017). Dalam penggunaannya Arduino nano diprogram melalui

aplikasi Arduino IDE pada komputer yang dihubungkan menggunakan kabel USB, dimana Arduino Nano merupakan mikrokontroler yang dapat langsung di program ulang jika diperlukan

2.5 Algoritma K-Nearest Neighbor

K-Nearest Neighbor merupakan metode dengan mengklasifikasikan suatu objek dengan mempertimbangkan kelas terdekat dari objek tersebut (Prahudaya & Harjoko, 2017). Dengan begitu kelas suatu objek baru tergantung objek-objek lama sejumlah k yang terbanyak. Dalam mengimplementasi Algoritma K-Nearest Neighbor ada beberapa langkah yang perlu dilakukan secara berurutan,

1. Menentukan nilai parameter k
2. Menghitung jarak Euclidean

$$dist(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

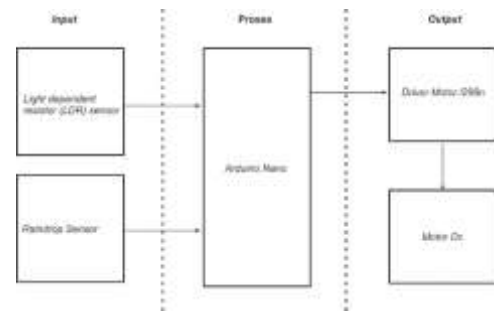
Dimana,
 dist (x, y) : jarak Euclidean
 n : jumlah data latih
 i : data latih ke-
 x : nilai fitur pertama
 y : nilai fitur kedua

3. Mengurutkan hasil jarak secara ascending.
4. Mengumpulkan jarak terdekat sejumlah k
5. Kategori dengan data mayoritas dari k data terdekat dapat diprediksi sebagai kategori objek.

3. PERANCANGAN DAN IMPLEMENTASI

3.1 Perancangan blok diagram

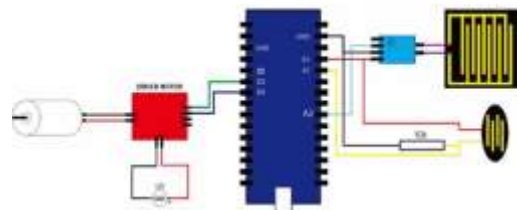
Blok diagram sistem menunjukkan blok-blok hubungan antar komponen. seperti pada gambar 1 menunjukkan terdapat input berupa raindrop dan ldr sensor yang akan mengirimkan data kepada arduino nano, sehingga hasil dari proses akan di lakukan oleh motor dc.



Gambar 2 Blok diagram sistem

3.2 Skematik Pemasangan Sistem

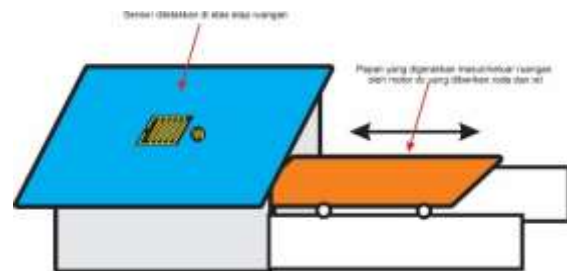
Setelah blok diagram maka dibuat detail dari pengkabelan dari masing masing komponen berbentuk skematik perangkat seperti pada gambar 2.



Gambar 3 Skematik pemasangan sistem

3.3 Perancangan Penempatan Sistem

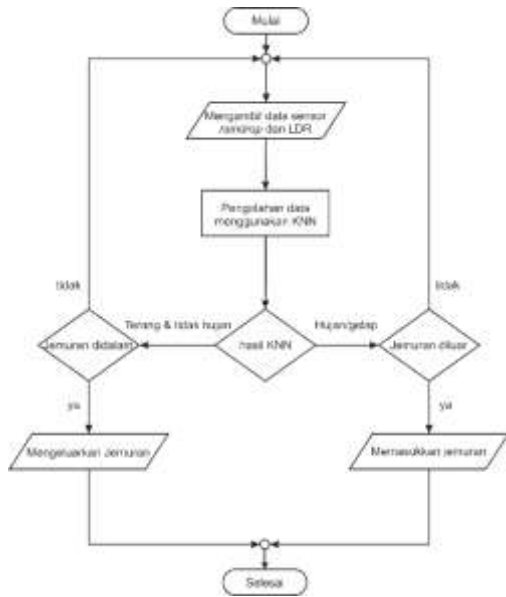
Selain perangkat elektronik maka diperlukan perancangan untuk maket sebagai media miniatur sistem seperti pada gambar 3.



Gambar 4 Desain Maket

3.4 Perancangan Program Sistem

Perancangan kinerja sistem yang akan diimplementasikan dalam kode program dibuat dalam bentuk flowchart seperti pada gambar 5 dimana setelah sistem dimulai maka sistem akan mengambil data sensor kemudian data masuk kedalam proses dengan menggunakan metode KNN dengan dibandingkan terhadap data latih yang telah diambil sebelumnya. Kemudian hasil akan berupa aksi yang dilakukan oleh aktuator.



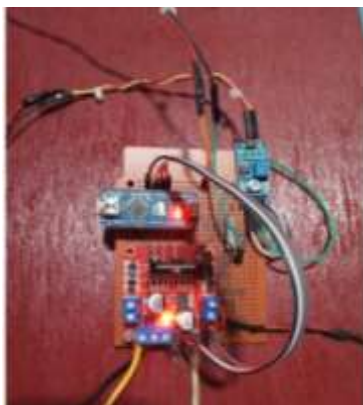
Gambar 5 Flowchart Metode KNN

3.5 Implementasi Sistem

Pada bagian ini berisi implementasi sistem berdasarkan dari perancangan yang telah dibuat sebelumnya dari perangkat keras, maket miniatur maupun kode program.



Gambar 6 Implementasi Perangkat Keras



Gambar 7 Implementasi Rangkaian Perangkat

Pada Gambar 5 dan Gambar 6 ditunjukkan hasil dari implementasi perangkat keras sistem yang dibuat.

Implementasi kode program dibuat menggunakan Arduino IDE dimana dibuat dua

jenis kode sumber yaitu menggunakan metode KNN dan metode *if-else* sebagai pembanding keakuratan metode KNN dalam sistem ini.

4. PENGUJIAN

4.1 Pengujian Sensor dan Aktuator

Sensor dan aktuator dilakukan pengujian, dengan mengambil data sensor, dan memberikan perintah aksi aktuator dengan menggunakan Arduino nano yang telah di berikan *source code* yang sesuai dengan sensor/aktuator tersebut. Sehingga mendapatkan hasil seperti pada tabel 2 berikut.

Tabel 2 Pengujian Sensor dan Aktuator

No.	Sensor/Aktuator	Hasil
1	LDR	Berjalan
2	Raindrop sensor/sensor hujan	Berjalan
3	Motor dc dengan driver motor	Berjalan

Dari tabel tersebut diketahui bahwa seluruh sensor dapat berjalan sesuai dengan yang diharapkan.

4.2 Pengujian Akurasi Metode

Pada pengujian ini menggunakan dataset sebanyak 100 data. Untuk menentukan validasi dataset maka digunakan metode K-Fold Cross Validation dimana data ini akan dibagi menjadi 10 bagian. Satu bagian akan digunakan sebagai data uji dan 9 bagian yang lain sebagai data latih, dan pada setiap bagian akan diberikan kesempatan sebagai data uji. Sehingga akan didapatkan rata-rata akurasi sistem.

Pada pengujian metode K-Nearest Neighbor akan diberikan variasi K yang terdiri dari tiga, lima, tujuh dan sembilan. Diambil nilai ganjil agar tidak terjadi kesamaan penentuan jumlah tetangga terdekat.

Tabel 3 Pengujian akurasi K-Nearest Neighbor K=3

Percobaan Ke	Data latih	Data uji	Benar	Akurasi
1	90	10	10	100%
2	90	10	10	100%
3	90	10	10	100%
4	90	10	10	100%

5	90	10	10	100%
6	90	10	9	90%
7	90	10	10	100%
8	90	10	9	90%
9	90	10	10	100%
10	90	10	10	100%
Rata Rata Akurasi				98%

Dari tabel 3 diatas dapat diketahui bahwa akurasi metode K-Nearest Neighbor dengan k=3 pada jemuran otomatis memiliki akurasi minimum 90% pada 2 percobaan dan akurasi maksimal sebesar 100% pada 8 percobaan sehingga rata-rata akurasi sistem menggunakan K-Nearest neighbor dengan K=3 adalah 98%

Tabel 4 Pengujian akurasi K-Nearest Neighbor K=5

Perco baan ke	Data latih	Data uji	Benar	Akur asi
1	90	10	10	100%
2	90	10	10	100%
3	90	10	10	100%
4	90	10	10	100%
5	90	10	10	100%
6	90	10	9	90%
7	90	10	10	100%
8	90	10	9	90%
9	90	10	10	100%
10	90	10	10	100
Persentase kebenaran				90%

Dari tabel 4 diatas dapat diketahui bahwa akurasi metode K-Nearest Neighbor dengan k=5 sama dengan akurasi pada k=3. Yaitu, memiliki akurasi minimum 90% pada 2 percobaan dan akurasi maksimal sebesar 100% pada 8 percobaan sehingga rata-rata akurasi sistem menggunakan K-Nearest neighbor dengan K=5 adalah 98%.

Tabel 5 Pengujian akurasi K-Nearest Neighbor K=7

Perco baan ke	Data latih	Data uji	Benar	Akur asi
1	90	10	10	100%
2	90	10	10	100%
3	90	10	10	100%
4	90	10	10	100%
5	90	10	10	100%
6	90	10	9	90%
7	90	10	10	100%
8	90	10	9	90%
9	90	10	10	100%
10	90	10	10	100
Persentase kebenaran				90%

Dari tabel 5 diatas dapat diketahui bahwa akurasi metode K-Nearest Neighbor dengan k=7

sama dengan akurasi pada k=3 dan k=5. Yaitu, memiliki akurasi minimum 90% pada 2 percobaan dan akurasi maksimal sebesar 100% pada 8 percobaan sehingga rata-rata akurasi sistem menggunakan K-Nearest neighbor dengan K=5 adalah 98%. Dimana akurasi tersebut terbilang hampir sempurna.

Tabel 6 Pengujian akurasi K-Nearest Neighbor K=9

Perco baan ke	Data latih	Data uji	Benar	Akurasi
1	90	10	9	90%
2	90	10	10	100%
3	90	10	9	90%
4	90	10	9	90%
5	90	10	9	90%
6	90	10	9	90%
7	90	10	10	100%
8	90	10	9	90%
9	90	10	9	90%
10	90	10	9	90%
Persentase kebenaran				92%

Dari tabel 6 diatas dapat diketahui bahwa metode K-Nearest Neighbor dengan k=9 pada jemuran otomatis memiliki akurasi yang cukup lain dari nilai k sebelumnya dimana akurasi minimal sebesar 90% pada 8 percobaan dan sisanya sebesar 100%. Sehingga rata-rata akurasi sistem menggunakan K-Nearest neighbor dengan K=9 adalah 92%.

Dari semua pengujian yang dilakukan dengan variasi K dan menggunakan metode validasi silang K-Fold Cross validation dengan k=10 didapatkan nilai setara tertinggi yaitu 98% pada k=3, k=5, dan k=7. Dan nilai terendah pada k=9 sebesar 92%. Namun secara keseluruhan akurasi metode k-nearest neighbor berada diatas 90%.

Tabel 7 Pengujian metode If-Else

Per cob aan ke	Kondisi sebenarnya	Kesimpulan Sistem	Benar /salah
1	Terang, tidak hujan	Jemuran diluar	benar
2	Gelap, tidak hujan	Jemuran didalam	benar
3	Terang, tidak hujan	Jemuran diluar	benar
4	Gelap, hujan	Jemuran didalam	benar
5	Gelap, hujan	Jemuran didalam	benar
6	Terang, hujan	Jemuran didalam	Benar
7	Gelap, hujan	Jemuran didalam	Benar
8	Terang, tidak hujan	Jemuran diluar	Benar
9	Terang, hujan	Jemuran didalam	Benar
10	Terang, tidak hujan	Jemuran diluar	benar
Persentase kebenaran			100%

Dibandingkan menggunakan metode *K-Nearest Neighbor* metode *if-else* untuk jemuran otomatis memiliki akurasi yang lebih sempurna dari 10 kali percobaan dihasilkan ketepatan sebesar 100%.

4.3 Pengujian Waktu Komputasi

Pengujian waktu komputasi merupakan pengujian kecepatan sistem dalam setiap satu kali sistem melakukan eksekusi *source code*. Penulis mengambil waktu eksekusi pada masing-masing kondisi satu nilai, hal ini dikarenakan ketika mengambil kondisi yang sama memiliki waktu komputasi yang sama pula.

Tabel 8 Waktu komputasi

Perco baan	Waktu Komputasi (mikrodetik)				
	K=3	K=5	K=7	K=9	If Else
1	37328	36976	37112	37192	13232
2	37000	37788	37312	37448	13232
3	37032	36932	37328	37412	13232
4	37332	37116	37288	37224	13232
Rata- Rata	37173	37203	37260	37319	13232

Dari tabel 8 diketahui bahwa waktu komputasi dari penggunaan metode *K-Nearest Neighbor* dan *if-else*. Dimana waktu komputasi untuk *if-else* seragam pada nilai 13.232 mikrodetik dan penggunaan kode program membutuhkan 2694 bytes. Sedangkan pada metode *K-Nearest Neighbor* menggunakan penyimpanan untuk kode program sebesar 4712 dimana waktu komputasi yang ditunjukkan semakin besar nilai *k* maka waktu komputasi semakin besar. Sehingga waktu komputasi untuk Metode *K-Nearest Neighbor* terendah pada *k=3* yaitu 37173 mikrodetik. namun waktu komputasi terkecil dari keseluruhan percobaan dimiliki oleh metode *if else*.

5. PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan oleh penulis mengenai implementasi metode *K-Nearest Neighbor* pada Purwarupa Jemuran otomatis yang berdasarkan sensor hujan dan intensitas cahaya, maka dapat diambil kesimpulan

1. Sensor LDR dalam sistem dapat mengambil data intensitas cahaya dengan baik.

2. Sensor hujan dalam sistem dapat mengambil data kondisi hujan maupun tidak hujan dengan baik.
3. Akurasi sistem jemuran otomatis berdasarkan sensor hujan dan intensitas cahaya menggunakan metode *K-Nearest Neighbors* adalah sebesar 90%.
4. Waktu komputasi rata-rata sistem jemuran otomatis berdasarkan sensor hujan dan intensitas cahaya menggunakan metode *K-Nearest Neighbors* ter rendah pada *k=3* yaitu sebesar 37.173 mikrodetik

Ketika dibandingkan dengan metode *if-else* yang lebih sederhana, didapatkan nilai, Akurasi sistem dalam menggunakan metode

1. *if-else* sederhana adalah sebesar 100%

2. Waktu komputasi rata-rata sistem yang menggunakan metode *if-else* sederhana adalah sebesar 13.232 mikrodetik

Sehingga penulis menyimpulkan Metode *K-Nearest Neighbor* cukup layak untuk diimplementasikan namun menggunakan metode *if-else* lebih baik. Dikarenakan akurasi *if-else* lebih baik, perbedaan waktu komputasi yang tidak signifikan hanya sebesar 0,023 detik, dan metode *if-else* lebih sederhana dalam implementasi *source code*.

5.2 Saran

Penulis menyadari penelitian ini memiliki bannya kekurangan, oleh sebab itu ada beberapa hal yang dapat dipertimbangkan untuk mengembangkan penelitian ini agar lebih baik yaitu,

1. Untuk dapat mengimplementasikan metode *K-Nearest Neighbor* pada sistem-sistem yang lain yang lebih kompleks sehingga dapat dikembangkan.
2. Menambahkan Sensor maupun aktuator lain, seperti sensor suhu, dan sensor kelembaban udara, dan dapat menambahkan blower atau penghangat ruangan.
3. Mengembangkan untuk dapat digunakan dalam prinsip *Internet of Thing* agar lebih dapat dipantau oleh pengguna ketika sistem sedang ditinggalkan.

6. DAFTAR PUSTAKA

- Akbar, M. F., Syauqy, D., & Setyawan, G. E. (2019, Agustus). Sistem Notifikasi Kondisi Cuaca Untuk Keselamatan Take Off Paralayang Menggunakan Metode Naïve Bayes (Studi Kasus: Paralayang Gunung Banyak, Batu). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(8), 7681-7687.
- Isfarizky, Z., Fardian, & Mufti, A. (2017). Rancang Bangun Sistem Kontrol Pemakaian Listrik Secara Multi Channel Berbasis Arduino (Studi Kasus Kantor LBH Banda Aceh). *KITEKTRO: Jurnal Online Teknik Elektro*, 2(2), 30-35.
- Mufida, E., & Abas, A. (2017). Alat Pengendali Atap Jemuran Otomatis Dengan Sensor Cahaya Dan Sensor Air Berbasis Mikrokontroler ATmega16. *INFORMATICS FOR EDUCATORS AND PROFESSIONALS*, 1, 163-172.
- Prahudaya, T. Y., & Harjoko, A. (2017). Metode Klasifikasi Mutu Jambu Biji Menggunakan KNN Berdasarkan Fitur Warna dan Tekstur. *Jurnal Teknosains*, VI(2), 59-138.
- Rivki, M., & Bachtiar, A. M. (2017). Implementasi Algoritma K-Nearest Neighbor dalam klasifikasi Follower Twitter yang Menggunakan Bahasa Indonesia. *Jurnal Sistem Informasi*, 13(1), 33.
- Utama, S., Mulyanto, A., Fauzi, M. A., & Putri, N. U. (2018). Implementasi Sensor Light Dependent Resistor (LDR) Dan LM35 Pada Prototipe Atap Otomatis Berbasis Arduino. *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*, 2(2), 83-89.
- Wirjohamidjojo, S., & Swarinoto, Y. (2010). *IKLIM KAWASAN INDONESIA (Dari Aspek Dinamik - Sinoptik)*. Jakarta : BADAN METEOROLOGI KLIMATOLOGI DAN GEOFISIKA .
- YU, H., JI, N., REN, Y., & YANG, C. (2019). A Special Event-Based K-Nearest Neighbor Model for Short-Term Traffic State Prediction. *IEEEAccess*, 7, 81717-81729.