

Deteksi Cuaca Secara *Real-Time* Menggunakan NodeMCU ESP8266 dengan Algoritma Machine Learning

Ahmad Syahdi Alkhawarizmi¹, Debri Yanti Safitri Santoso², Faiza Latifah³, Galih Multi Lidzikri⁴, Muhammad Bilal Hamzah⁵, Muhammad Fariduddin Athar⁶, Muhammad Wildan Alfarizy⁷, Rafharum Fatimah⁸, Rizal Ramdhani⁹, Vinka Reviansa¹⁰, Vladio Sada Arihta Sembiring¹¹

1, 2, 3,4,5,6,7,8,9,10,11 Mekatronika dan Kecerdasan Buatan, Universitas Pendidikan Indonesia

rizmisyahdi@upi.edu , debriyanti@upi.edu , faizalatifah@upi.edu , galihmulti@upi.edu ,
bilalhamzah29@upi.edu , farid.udin25@upi.edu , muhammad.wildan@upi.edu , rafharumf@upi.edu ,
rizalramdhani03@upi.edu , vinkareviansa@upi.edu , vladioarihta@upi.edu

Abstract

I. PENDAHULUAN

Perkembangan teknologi kian hari semakin berkembang pesat. Salah satu dari perkembangan teknologi yaitu sistem pemantauan atau biasanya disebut sistem monitoring. Sistem monitoring bertujuan untuk mengetahui situasi dan kondisi tertentu untuk membantu pekerjaan manusia dalam melakukan tugas pemantauan, salah satunya yaitu sistem pemantauan cuaca. Dalam mendeteksi cuaca suatu daerah, pengumpul data atau *user* harus datang dan menetap dalam jangka waktu tertentu pada lokasi yang ingin di pantau. Alat yang digunakan untuk mendeteksi cuaca pun belum terintegrasi satu sama lain dan mencatat secara manual.

Teknologi Internet of Things (IoT) telah memungkinkan pengembangan solusi yang dapat mendeteksi dan memantau kondisi cuaca secara langsung dan efisien. Dalam konteks ini, penggunaan NodeMCU 8266 sebagai platform IoT memberikan kemudahan dalam mengintegrasikan sensor dan mengirimkan data cuaca secara real-time ke server.

Beberapa penelitian sebelumnya telah membuat sistem monitoring cuaca se“Implementasi Sistem Monitoring Deteksi Hujan dan Suhu Berbasis Sensor Secara Real Time” oleh M.Y.Mustar et.al [1], pada penelitian ini menyajikan perancangan sebuah sistem monitoring dengan sensor NTC Thermistor yang diinputkan pada Arduino NANO dengan antarmuka *graphical user interface*(GUI) berbasis JAVA. Kemudian pada penelitian “IOT LIVE WEATHER MONITORING SYSTEM USING NODEMCU ESP8266” oleh A.K.Singh, A.Mishra, et.al [2], pada penelitian ini menggunakan sensor DHT11, sensor BMP180, dan sensor Hujan yang di inputkan pada NodeMCU 8266 yang terintegrasi pada ThingSpeak dan IFTTT. Selanjutnya pada penelitian “Smart weather monitoring and real time alert system using IoT” oleh Y.Rahut, R.Afreen, et.al [3], pada penelitian ini membuat sistem pemantauan cuaca dengan sensor LDR,

sensor CO, sensor DHT11 yang terintegrasi pada NodeMCU 8266 yang terhubung pada web yang bisa diakses oleh seluruh dunia.

Dalam penelitian ini, peneliti menggunakan NodeMCU 8266 sebagai platform sensor untuk mengumpulkan data cuaca secara real-time. Data yang terkumpul kemudian akan dianalisis menggunakan algoritma K-Means Clustering untuk mengidentifikasi pola-pola yang mungkin tersembunyi dalam data tersebut. Setelah melakukan labelling, data tersebut kemudian diolah menggunakan algoritma Random Forest untuk melakukan prediksi cuaca. Hal ini diharapkan dapat memberikan pemahaman yang lebih baik terkait dengan perubahan cuaca yang terjadi secara real-time.

Algoritma K-Means merupakan metode unsupervised learning yang dapat mengelompokkan data kedalam kelompok-kelompok berdasarkan karakteristik yang serupa. Dalam konteks deteksi cuaca, penggunaan K-Means Clustering dapat membantu dalam mengidentifikasi pola-pola tertentu pada data sensor yang bervariasi, seperti suhu, kelembaban, tekanan udara, dan lainnya.

Random Forest merupakan algoritma ensemble learning yang bekerja dengan membangun sejumlah besar decision tree dari subset data latih secara acak dan kemudian menggabungkan output dari seluruh decision tree tersebut untuk prediksi akhirnya.

Dengan memadukan teknologi IoT dan algoritma analisis data, proyek ini bertujuan untuk menciptakan sistem deteksi cuaca yang dapat memberikan informasi yang berguna dan akurat dalam memantau perubahan cuaca secara real-time. diharapkan dengan implementasi teknologi ini, diharapkan kita dapat meningkatkan pemahaman dan kemampuan dalam mengantisipasi serta mengelola dampak dari perubahan cuaca yang terjadi di lingkungan sekitar.

II. KAJIAN PUSTAKA

A. Penelitian Terdahulu

No.	Nama,Tahun	Permasalahan	Data	Kontribusi
1.	M.Y.Mustar et.al,2017 [1].	Pada umumnya alat yang digunakan untuk mendeteksi hujan dan suhu belum terintegrasi menjadi satu.	NTC Thermistor Arduino NANO Radio Telemetry	. Sensor hujan dapat bekerja mendeteksi hujan, selain itu rata-rata selisih pembacaan sensor suhu dengan thermoter tergolong sekitar 0,39°C, sehingga alat yang di bangun dapat digunakan sebagai sistem yang mampu memberikan informasi mengenai pendeteksian hujan dan suhu secara

				real time dengan komunikasi nirkabel menggunakan telemetri.
2.	A.K.Singh, A.Mishra, et.al, 2022 [2].	Proyek ini akan fokus pada pengembangan platform IoT Thing Speak yang menampilkan data sensor.	Sensor BMP180 Sensor DHT11 Sensor Hujan ESP8266	Sistem pelaporan cuaca ini akan menampilkan data sensor ke ThingSpeak dan IFTTT untuk disimpan ke dalam google sheet.
3.	Y.Rahut, R.Afreen, et.al,2018 [3].	merancang dan mengimplementasikan sistem pemantauan yang cerdas dimana parameter yang diperlukan dipantau dari jarak jauh menggunakan internet dan data yang dikumpulkan dari perangkat disimpan di cloud dan untuk memproyeksikan tren yang dapat diprediksi di browser web.	Arduino UNO LDR Light-Dependent Resistor NodeMCU CO Sensor DHT11 ML8511 Anemometer	Sistem pemantauan kebisingan dan polusi udara dengan konsep Internet of Things (IoT) diuji secara eksperimental untuk memantau dua parameter. Itu juga mengirimkan parameter sensor ke cloud (Google Spread Sheets). Data ini akan berguna untuk analisis di masa mendatang dan dapat dengan mudah dibagikan kepada pengguna akhir lainnya.

Table 1 Penelitian Terdahulu

Dari beberapa penelitian diatas peneliti mengembangkan sistem untuk memonitoring cuaca secara realtime menggunakan sensor dengan integrasi NodeMCU 8266 yang di clustering dengan algoritma machine learning.

B. Internet of Things

Internet of Things (IoT) adalah konsep penggabungan dari sebuah sistem, perangkat, sensor, dan teknologi koneksi yang memungkinkan mereka saling berkomunikasi dan bertukar data melalui internet [4]. Tujuan utama dari IoT adalah untuk memberikan kecerdasan dan konektivitas yang lebih baik pada objek-objek tersebut, meningkatkan efisiensi, dan memberikan layanan yang lebih baik kepada pengguna. Selain itu, penerapan IoT dalam mendeteksi cuaca secara real-time bertujuan untuk memanfaatkan konektivitas internet yang terhubung ke dalam sistem dan mengamati keadaan cuaca pada tempat tinggal atau area lokal yang diinginkan [5].

Penerapan IoT tersebut menggunakan perangkat sensor yang saling terintegrasi. Dimana sensor sendiri merupakan perangkat yang dapat menangkap perubahan variabel seperti suhu, tekanan, kelembapan, dan lainnya. Perubahan tersebut mempengaruhi sifat fisik, kimia, atau elektromagnetik dari sensor yang kemudian hasilnya akan diproses hingga dapat dibaca oleh suatu rangkaian elektronik [6]. Sensor yang saling terintegrasi dalam perangkat IoT menjadi kunci utama dalam mendeteksi cuaca secara real-time. Sensor yang biasa digunakan dalam mendeteksi cuaca adalah sensor suhu dan sensor tekanan atmosfer.

C. NodeMCU ESP8266 dan *Firestore Integration*

NodeMCU ESP8266 merupakan modul mikrokontroler berbasis chip ESP8266 yang memiliki kemampuan untuk terhubung dengan jaringan internet (WiFi). Dalam penggunaannya, modul ini dikembangkan untuk mendukung pengembangan IoT. Sebagai perangkat penghubung, NodeMCU dapat diintegrasikan dengan berbagai sensor dan aktuator untuk memonitor dan mengendalikan lingkungan fisik [7].

NodeMCU dapat diintegrasikan dengan Firestore. Firestore merupakan platform pengembangan aplikasi berbasis cloud dari Google yang juga menyediakan berbagai macam layanan, termasuk penyimpanan data realtime, autentikasi pengguna, dan hosting aplikasi web [8]. Integrasi NodeMCU dengan Firestore memungkinkan transfer data secara langsung dan real-time antara perangkat IoT dan cloud, menciptakan solusi yang responsif dan terhubung.

Keterkaitan antara keduanya menghasilkan solusi IoT yang kuat dan terintegrasi. NodeMCU, dengan kemampuannya untuk mengumpulkan data dari sensor dan mengirimkannya ke Firestore, memungkinkan pengguna untuk dengan mudah mengakses dan memanfaatkan data tersebut dari mana saja. Data yang dikirim ke Firestore secara real-time dapat memberikan informasi yang akurat dan up-to-date kepada pengguna atau sistem untuk membantu memantau kondisi lingkungan yang terhubung. Dengan demikian, integrasi antara NodeMCU dan Firestore menciptakan solusi IoT yang memungkinkan monitoring dan pengendalian yang efisien, serta menyediakan basis untuk pengembangan aplikasi IoT yang lebih kompleks dan berkelas tinggi.

D. Algoritma K-Means Clustering

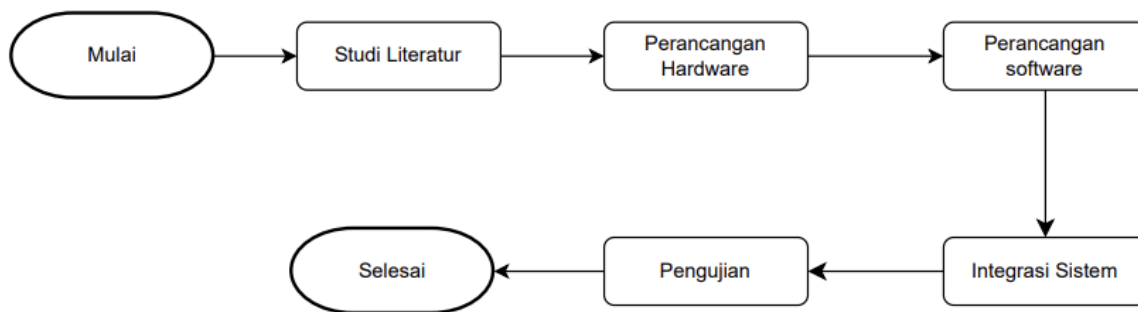
Algoritma K-Means Clustering merupakan suatu metode dalam analisis data yang digunakan untuk mengelompokkan atau mengklasifikasikan objek-objek ke dalam beberapa kelompok berdasarkan kesamaan atribut atau ciri-ciri tertentu. Algoritma ini bekerja dengan cara mengelompokkan data ke dalam k kelompok (kluster) sedemikian rupa sehingga objek-objek dalam satu kluster memiliki kesamaan yang tinggi dengan objek-objek lain dalam kluster yang sama. Tujuan utama dari algoritma ini adalah mengelompokkan objek-objek data sedemikian rupa sehingga objek dalam satu kluster memiliki kesamaan yang tinggi dan berbeda secara signifikan dengan objek dalam kluster lainnya [9].

E. Algoritma Random Forest

Algoritma Random Forest adalah sebuah algoritma dalam machine learning yang digunakan untuk pengklasifikasian data set dalam jumlah besar. Algoritma ini memanfaatkan teknik interpolasi spasial yang banyak digunakan dalam bidang geografi, geologi, meteorologi, dan klimatologi [10]. Algoritma ini merupakan kumpulan dari decision tree atau pohon keputusan, yang kemudian digabungkan. Random Forest dapat digunakan untuk masalah klasifikasi dan regresi, dan biasanya dilatih dengan metode "bagging" yang menggabungkan model-model pembelajaran untuk meningkatkan hasil keseluruhan. Algoritma ini dianggap fleksibel, mudah digunakan, dan salah satu algoritma machine learning terbaik karena kesederhanaan dan keragamannya. Namun, kelemahan dari algoritma ini antara lain cenderung bias saat berhadapan dengan variabel kategorikal, waktu komputasi pada dataset berskala besar relatif lambat, dan tidak cocok untuk metode linier dengan banyak fitur sparse. Untuk memahami lebih lanjut tentang algoritma Random Forest, penting untuk memiliki pengetahuan mendasar tentang jenis pohon yang digunakan sebagai base learner, serta tentang konsep-konsep seperti ensemble learning dan metode "bagging" yang digunakan dalam pembentukan model ini.

III. METODE

Metode yang digunakan pada penelitian ini dapat direpresentasikan melalui diagram alur dibawah ini :



Gambar 1 Flowchart Metode Penelitian

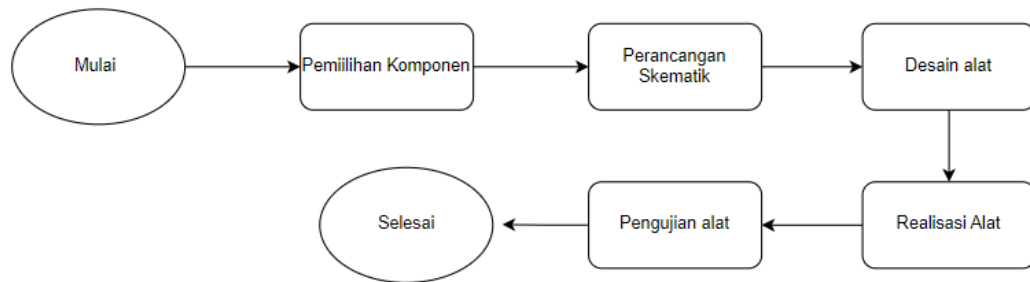
A. Studi Literatur

Seluruh tahapan dilakukan secara berurutan, diawali dengan studi literatur penelitian terdahulu yang relevan. Literatur sejenis yang digunakan seperti skripsi, jurnal, dan produk sejenis untuk kemudian dilakukan perbandingan dengan penelitian yang sedang dilakukan sehingga penelitian ini dapat menjadi pelengkap dan penyempurnaan dari penelitian-penelitian yang sudah dilakukan sebelumnya. Hal ini bertujuan untuk mencari keterbaruan penelitian, teknis perancangan sistem hardware, teknis perancangan

software, dan integrasi sistem pengujian software. Literatur sejenis yang menjadi referensi dapat dilihat pada tabel 1.

B. Perancangan Hardware

Flowchart perancangan hardware ditunjukkan pada flowchart pada gambar dibawah ini.

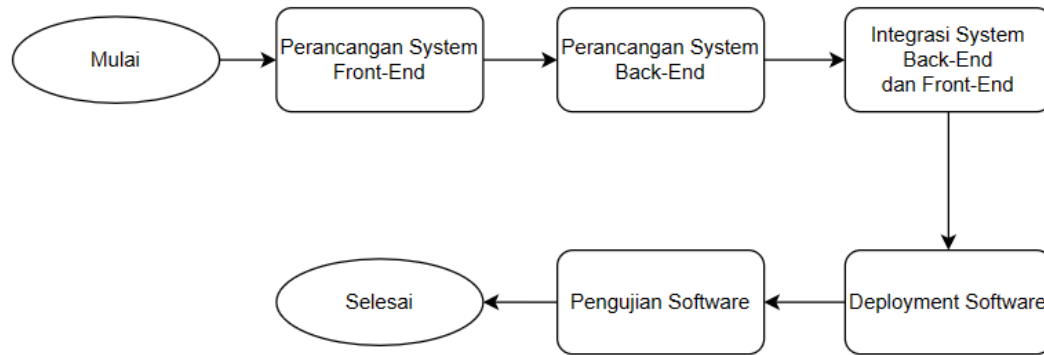


Gambar 2 Flowchart Perancangan Hardware

Tahap pertama adalah memilih komponen yang digunakan sesuai dengan alat yang akan dibuat. Tahap kedua adalah perancangan skematik diagram, yang merupakan representasi visual yang memberikan gambaran jelas mengenai hubungan dan interaksi antar komponen atau elemen dalam suatu sistem. Proses perancangan ini dimulai dengan identifikasi kebutuhan dan spesifikasi proyek, diikuti dengan pemilihan komponen-komponen yang sesuai. Tahap ketiga adalah desain alat, yang merupakan proses merancang dan mengembangkan alat, metode, dan teknik untuk memperbaiki efisiensi dan produktivitas manufaktur. Desain alat dibutuhkan untuk menentukan arsitektur desain. Tahap keempat yaitu realisasi alat, merujuk pada implementasi atau pembuatan fisik dari desain alat yang telah direncanakan sebelumnya. Setelah melalui tahap perancangan, langkah selanjutnya adalah mewujudkan konsep tersebut menjadi bentuk nyata yang fungsional. Selanjutnya, tahap terakhir adalah pengujian alat yang merupakan tahapan penting dalam pengembangan dan perancangan alat dengan tujuan untuk mengetahui kinerja alat baik dari masing-masing bagian maupun keseluruhan alat.

C. Perancangan Software

Flowchart perancangan software ditunjukkan pada flowchart pada gambar dibawah ini.



Gambar 3 Flowchart Perancangan Software

Tahap pertama adalah perancangan Front-End yaitu perancangan user interface dengan tujuan untuk menciptakan antarmuka yang intuitif, efisien, dan menyenangkan untuk digunakan. Tahap kedua adalah perancangan Back-End, yang merupakan tahap untuk menyiapkan database, menyiapkan API, dan menyiapkan model machine learning. Tahap ketiga adalah Integrasi sistem Back-End dan Front-End yaitu proses menyatukan dua bagian utama dari suatu sistem Back-End dan Front-End agar dapat beroperasi secara bersinergi. Tahap terakhir adalah pengujian software yaitu proses yang dilakukan sistematis untuk mengevaluasi dan memastikan kualitas, kehandalan, dan kinerja suatu aplikasi atau sistem perangkat lunak sepanjang siklus pengembangan.

D. Integrasi System

Integrasi sistem antara dengan Machine Learning memiliki beberapa aspek penting yang perlu diperhatikan. Pertama, Machine Learning dapat digunakan untuk memprediksi cuaca dengan menggunakan data dari stasiun cuaca. Kedua, stasiun cuaca yang terintegrasi dengan Machine Learning dapat mengumpulkan data cuaca secara terus berperiode, seperti suhu, kelembapan, dan angin. Ketiga, model residual machine learning dapat digunakan untuk membuat prediksi cuaca yang lebih akurat dan efisien. Keempat, stasiun cuaca yang terintegrasi dengan Machine Learning dapat menggunakan teknologi IoT, seperti sensor dan perangkat lunak yang terhubung dengan internet, untuk mengumpulkan data cuaca dan mengirimkan data tersebut ke perangkat pemantauan untuk analisis dan pengambilan keputusan. Kelima, Machine Learning dapat digunakan untuk mengoptimalkan pemilihan parameter yang digunakan dalam model cuaca, seperti mengurangi jumlah parameter yang berkontribusi pada prediksi cuaca. Integrasi sistem antara stasiun cuaca dengan Machine Learning dapat membantu mengoptimalkan proses pengumpulan data cuaca, memprediksi pola cuaca, dan membuat analisis cuaca yang lebih akurat dan efisien.

E. Pengujian

Pengujian yang dilakukan dalam penelitian ini dilakukan dalam dua tahap, tahap pertama yaitu proses pengambilan data dengan menguji sistem, sensor dan integrasi antar komponen, dalam pengujian pertama didapatkan data set yang terdiri dari suhu,

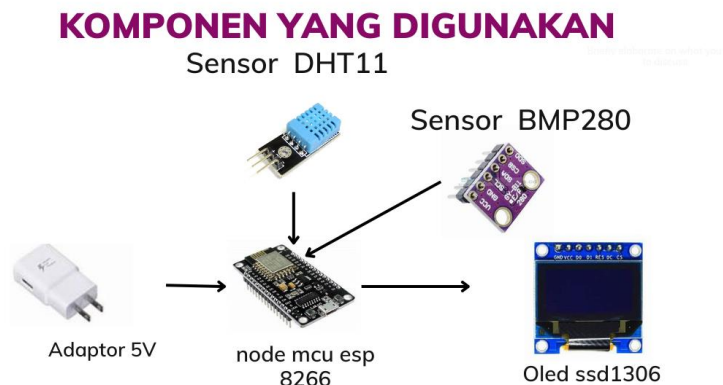
kelembaban, altitude dan pressure. Pengambilan data pada pengujian pertama dilakukan dengan rentang waktu 12 jam, dimulai dari siang hari hingga malam hari, ketika proses pengujian tersebut data terekam dalam firebase untuk penyimpanan database secara realtime yang nantinya akan digunakan untuk proses training model machine learning.

Pada percobaan pengujian kedua merupakan pengujian model setelah melewati proses training untuk memprediksi suatu keadaan cuaca, dimana hasil dari prediksi model akan dibandingkan dengan keadaan yang sebenarnya. Hasil prediksi dari proses pengujian ini ditampilkan pada Website HTML.

IV. HASIL DAN PEMBAHASAN

A. Hardware

1. Komponen yang digunakan



Gambar 4 Komponen yang digunakan

Komponen yang digunakan yaitu:

- Adaptor 5V

Adaptor 5V umumnya digunakan untuk menyediakan daya listrik stabil dengan tegangan 5 volt untuk perangkat-perangkat seperti NodeMCU, sensor-sensor, dan modul-modul lainnya. Daya yang cukup dan stabil sangat penting untuk menjaga agar komponen-komponen tersebut berfungsi dengan baik.

- Nodemcu esp8266

NodeMCU ESP8266 sering digunakan dalam proyek-proyek IoT karena kemampuannya untuk terhubung ke jaringan Wi-Fi. Dengan ini, dapat mengakses internet untuk mengirim dan menerima data, membuatnya sangat berguna untuk berbagai aplikasi IoT seperti pemantauan cuaca, pengendalian perangkat, dan lain sebagainya.

- Sensor DHT11

Sensor DHT11 memberikan informasi suhu dan kelembaban. Sensor ini relatif cukup mudah digunakan dalam proyek-proyek yang membutuhkan pemantauan kondisi suhu dan kelembaban.

- Sensor BMP280

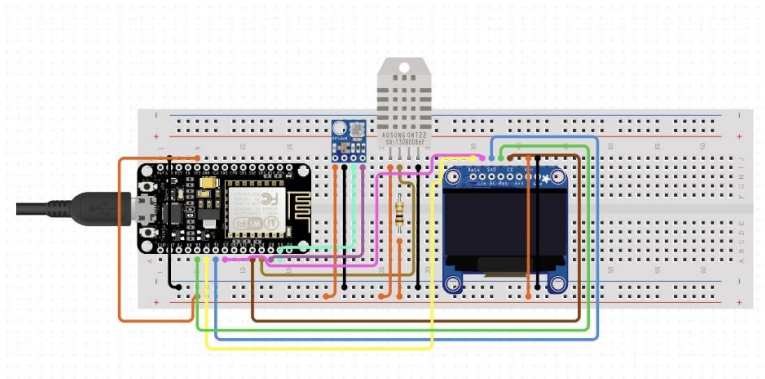
Sensor BMP280 memberikan informasi tentang tekanan atmosfer dan suhu. Ini bisa menjadi komponen penting dalam aplikasi yang membutuhkan informasi tentang ketinggian, perkiraan cuaca, atau pemantauan lingkungan.

- OLED SSD1306

Layar OLED SSD1306 memberikan tampilan visual yang jernih dan tajam dengan konsumsi daya yang rendah. Hal ini berguna untuk menampilkan data secara langsung dari perangkat atau memberikan informasi kepada pengguna dengan cara yang lebih interaktif.

2. Skematik Diagram

Skematik diagram adalah representasi visual yang menunjukkan hubungan dan interaksi antar komponen dalam suatu sistem.

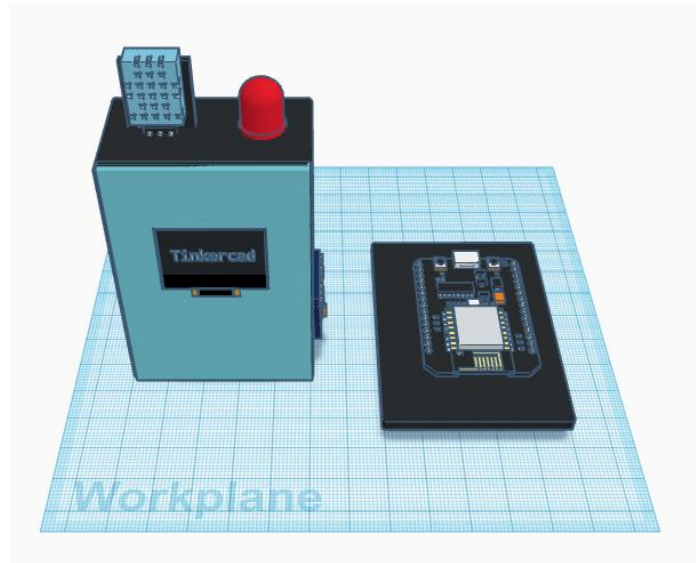


Gambar 5 Skematik Diagram Perancangan Hardware

- Pin VCC 3,3 Volt pada nodemcu esp8266 dihubungkan dengan VCC Sensor DHT11, Sensor BMP280, dan Oled ssd1306
- Pin D1 pada nodemcu esp8266 di dihubungkan dengan SDA pada sensor bmp280 dan oled ssd1306
- Pin D2 pada nodemcu esp8266 di dihubungkan dengan SCL pada sensor bmp280 dan oled ssd1306
- Pin GND pada nodemcu esp8266 dihubungkan dengan GND Sensor DHT11, Sensor BMP280, dan Oled ssd1306.
- Pin D5 pada nodemcu esp8266 dihubungkan dengan beban pin data pada DHT11.

3. Desain Alat

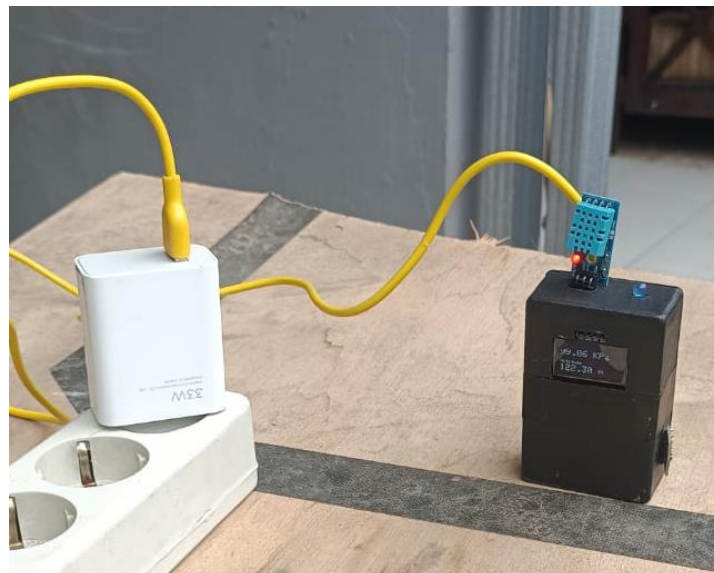
Desain ini merupakan kerangka dasar yang dapat disesuaikan dengan kebutuhan spesifik proyek dan integrasi dengan platform komunikasi atau layanan monitoring cuaca yang akan kami buat.



Gambar 6 Simulasi Desain Sistem

4. Implementasi Alat

Pada gambar 7 ini merupakan implementasi dari desain yang telah kita rancang sebelumnya.



Gambar 7 Implementasi Desain Sistem

Desain alat dibuat secara potret dimana pada tampak depan akan ada layar LCD yang menampilkan cuaca secara realtime dan tampak belakang akan ada Node MCU 8266

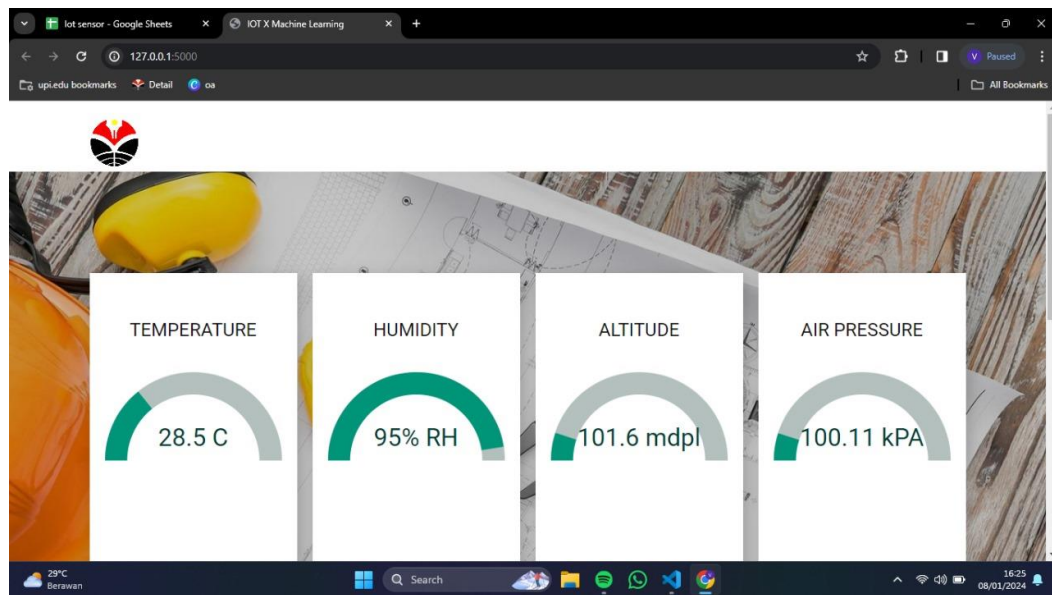
yang terhubung dengan sumber daya, selain sensor DHT11 disimpan pada bagian atas desain dan sensor BMP280 disimpan pada sebelah kanan dari layar LCD.

B. Software

1. Rancangan Font-End

- User Interface

Rancangan yang dibuat yaitu membuat antarmuka pengguna yang menarik dan responsif yang dapat menampilkan informasi cuaca secara real-time. Antarmuka ini menampilkan suhu, kelembaban, ketinggian, dan tekanan udara dengan tata letak yang intuitif dan tampilan yang estetik.



2. Rancangan Back-End

- Dataset

Selanjutnya untuk mengklasifikasikan suhu peneliti mengumpulkan dataset hingga nantinya dataset ini akan di klasifikasi menggunakan algoritma machine learning.

No.	Humidity	Temperature	Altitude	Air Pressure
1.	92	31.8	11,736,579	9,992,715,625
2.	92	31.3	12,106,192	9,987,934,375
3.	92	30.8	12,120,945	9,987,759,375
4.	91	30.2	12,126,337	9,987,695,312
5.	91	30.2	12,147,755	9,987,440,625
6.	93	29.8	12,098,816	9,988,021,875
7.	93	29.8	12,080,299	9,988,242,188
8.	93	29.4	12,114,891	998,783,125

9.	94	29.4	12,074,449	9,988,310,938
...
670.	87	29.4	11,558,456	9,994,439,062
671.	87	29.4	11,558,456	9,994,439,062
672.	87	29.4	11,558,456	9,994,439,062
673.	87	29.4	11,558,456	9,994,439,062
674.	96	27.7	8,932,879	1,002,566,563
675.	98	26.7	9,785,864	100,155,125

- API

Dalam pengembangan antarmuka pengguna untuk pemantauan cuaca, kami menggunakan API lokal yang dijalankan di dalam jaringan lokal atau perangkat sendiri. API lokal ini berfungsi sebagai titik akses untuk mengambil data cuaca yang dikumpulkan oleh perangkat sensor yang terhubung secara langsung ke sistem kami. Melalui penggunaan teknologi seperti Node.js atau Flask, kami mengakses API ini untuk mendapatkan informasi terbaru tentang suhu, kelembaban, ketinggian, dan tekanan udara yang diperoleh dari sensor-sensor yang ada. Integrasi API lokal memungkinkan kami untuk mengakses data secara cepat dan efisien tanpa ketergantungan pada sumber data eksternal, serta memberikan kontrol penuh atas informasi cuaca yang disajikan dalam antarmuka pengguna kami.

```

1 from flask import Flask, render_template, request
2 import pickle
3 import numpy as np
4
5 model = pickle.load(open('weather_model.pkl', 'rb'))
6
7 app = Flask(__name__)
8
9 prediction_mapping = {
10     'default': 1,
11     'Clear Sky': 2,
12     'Cloudy': 3
13 }
14 # tentukan nilai dan label yang sesuai sesuai dengan kebutuhan Anda
15
16 def get_prediction_output(pred):
17     return prediction_mapping[pred]
18
19 @app.route('/')
20 def main():
21     return render_template('web.html')
22
23 @app.route('/predict', methods=['POST']) # gets the values that were sent to /predict by 'index.html'
24 def predict():
25     data = request.get_json()
26     temp = data['temp']
27     press = data['press']
28     humi = data['humi']
29     alt = data['alt']
30     features = [[humi, temp, press]]
31
32     prediction = model.predict(features)
33     pred_output = get_prediction_output(prediction)
34     return pred_output
35
36 if __name__ == '__main__':
37     app.run(debug=True)

```

- Model Machine Learning

Pemodelan machine learning dibuat berdasarkan dataset hasil perekaman variabel-variabel dengan sensor. Variabel-variabel tersebut adalah Humidity, Temperature, Altitude, dan Air Pressure. Kemudian setelah data terekam, data dikumpulkan dalam bentuk csv dengan bantuan firebase. Setelah data sudah terkumpul dalam bentuk dataset, proses berlanjut ke proses tahap awal pemodelan

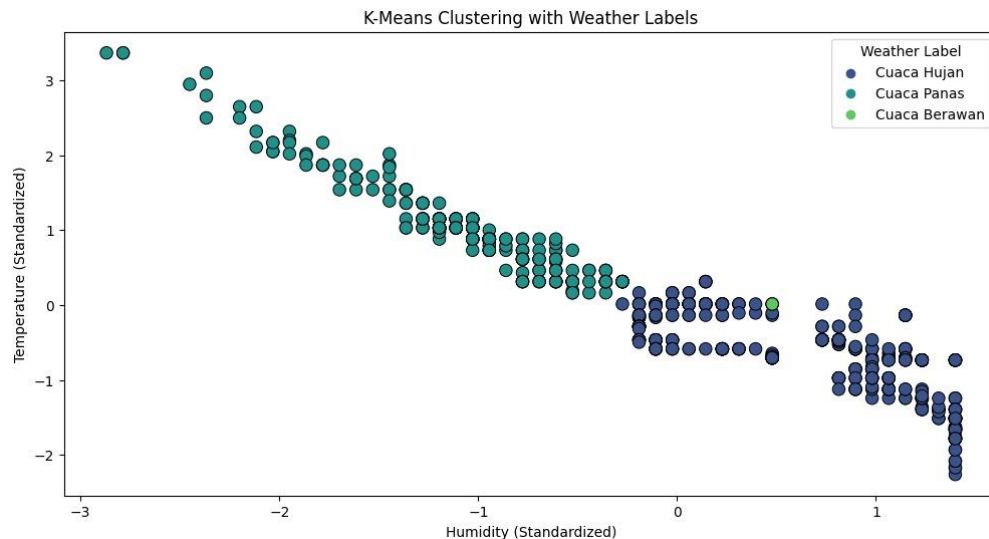
machine learning yaitu tahap preprocessing. Tahap ini mencakup proses cleaning data terhadap data yang memiliki nilai outlier dan imputasi nilai. Setelah itu proses selanjutnya adalah proses EDA (*Exploratory Data Analysis*), pada proses ini ditemukan bahwa fitur altitude memiliki pengaruh buruk terhadap hasil akhir model, maka pada tahap ini dilakukan drop pada kolom altitude.

Setelah proses cleaning selesai, maka proses dapat berlanjut ke tahap pelatihan dan deployment model. Pada tahap ini, digunakan beberapa algoritma sebagai percobaan seperti DBSCAN, GMM, Gradient Boost, Hierarki Agglomerative dan K-Means. Pada penggunaan DBSCAN, algoritma dan dataset tidak kompatibel sehingga menimbulkan banyak kesalahan. Pada penggunaan gradient boost, terjadi ketidakefisienan dalam durasi proses pembuatan model. Pada penggunaan algoritma GMM, kami menemukan ketidakseimbangan pada jumlah antar kluster. Pada penggunaan Hierarki Agglomerative dan K-Means, dihasilkan skor koefisien silhouette sebesar masing-masing 50% dan 53%. Maka berdasarkan berbagai pertimbangan dan hasil skor koefisien silhouette, maka algoritma yang paling kompatibel adalah K-Means.

Setelah menentukan jenis algoritma yang akan dipakai, yaitu K-Means. Selanjutnya adalah proses model deployment yaitu dengan menyimpan model yang telah dibuat dalam bentuk file pickle (.pkl) kemudian menentukan lingkungan deployment, dalam hal ini kami menggunakan server cloud yaitu firebase untuk menyimpan data rekaman hasil sensor secara real time. Maka berdasarkan berbagai pertimbangan dan hasil skor koefisien silhouette, maka algoritma yang paling kompatibel untuk melakukan labelling data adalah K-Means. Setelah menggunakan K-Means, didapatkan 3 cluster yaitu panas, hujan, dan berawan.

Setelah menentukan jenis algoritma yang akan dipakai, selanjutnya adalah menentukan algoritma untuk melakukan prediksi. Setelah data sudah diberi label melalui proses clustering, pada eksperimen kali ini, kami menggunakan random forest sebagai algoritma untuk memprediksi hasil. Dengan menggunakan tiga fitur yaitu humidity, temperature, dan air pressure, selanjutnya data akan dibagi menjadi

data train dan test dengan metode train test split. Setelah data telah terbagi, maka proses dapat berlanjut ke pembuatan model prediksi, selanjutnya apabila model telah selesai dibuat, model dapat disimpan dalam bentuk file pickle (.pkl) agar dapat digunakan untuk memprediksi cuaca. Pada penggunaan algoritma random forest ini, kami mendapatkan akurasi 100% dengan metode accuracy score dari library sklearn.metrics.

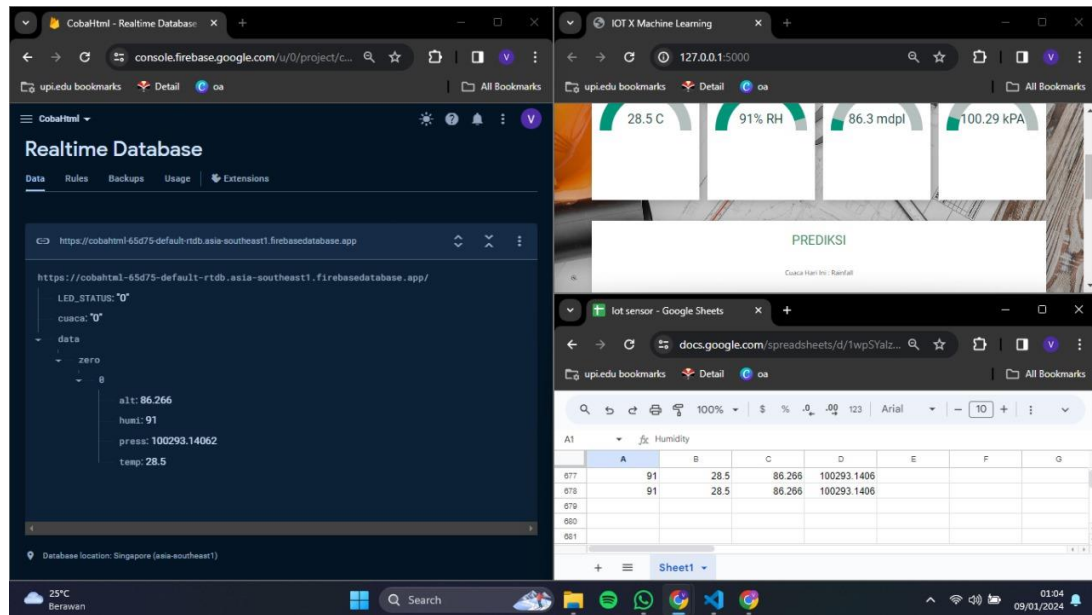


3. Pengujian system

Pengujian sistem disini peneliti ingin memastikan bahwa pembuatan antarmuka pengguna (UI) untuk pemantauan cuaca menggunakan HTML, CSS, dan JavaScript, API sudah tidak ada error dan dapat berfungsi dengan baik dan sudah berjalan seperti yang peneliti inginkan. Juga melihat hasil predik yang dilakukan ketika melakukan clustering dengan K-Means dan percobaan prediksi dengan random forest sudah berfungsi.

C. Integrasi dan Deployment system

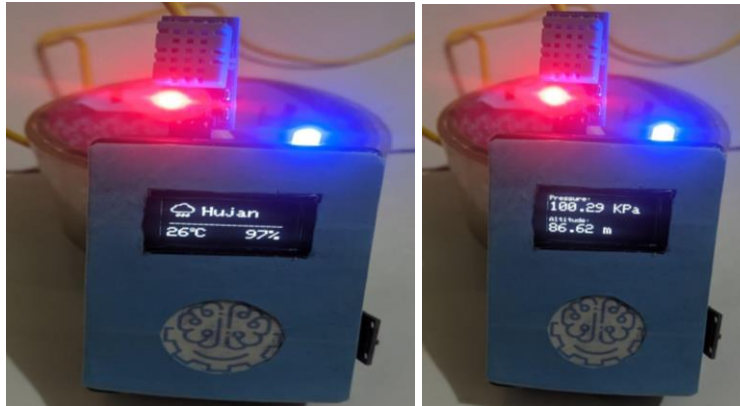
Integrasi dan sistem deployment digunakan untuk mengintegrasikan hardware dan software agar sistem dapat berjalan secara realtime dengan cara data yang dibaca oleh sistem dengan klasifikasi K-Means akan ditampilkan pada user interface.



Mengintegrasikan atau menghubungkan berbagai macam sistem sehingga menjadi sebuah sistem. Terdapat beberapa tahapan Integrasi yaitu Integrasi Hardware dengan Firebase sehingga data yang dihasilkan oleh output sensor akan masuk kedalam firebase. Kemudian data dari Firebase akan di Integrasi dengan Spreadsheet dan API (HTML) sehingga data yang dihasilkan dari Firebase dapat ditampilkan di halaman HTML. Dimana sebelumnya pada API sudah dimasukan model Machine Learning yang sudah dilakukan training dengan menggunakan data yang berasal dari spreadsheet. Sehingga pada halaman HTML dapat menampilkan data realtime yang dihasilkan oleh Hardware dan menampilkan hasil prediksi cuaca dari model Machine Learning.

D. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan dengan cara setelah sistem hardware dan software terintegrasi maka sensor dapat membaca data cuaca saat alat tersebut dinyalakan dan User Interface dapat menampilkan suhu, tekanan, kelembaban dan altitude juga prediksi data yang masuk tersebut tergolong pada kategori hujan, panas atau berawan.



V. KESIMPULAN

Dalam rangka meningkatkan efisiensi dan akurasi pemantauan cuaca, penelitian ini memanfaatkan teknologi Internet of Things (IoT) dengan menggunakan NodeMCU 8266 sebagai platform sensor. Penggunaan algoritma K-Means Clustering untuk mengidentifikasi pola-pola dalam data sensor serta algoritma Random Forest untuk melakukan prediksi cuaca diharapkan dapat memberikan pemahaman yang lebih baik terkait perubahan cuaca secara real-time. Integrasi teknologi IoT dan analisis data ini bertujuan untuk menciptakan sistem deteksi cuaca yang informatif dan akurat, memungkinkan pengguna untuk mengantisipasi dan mengelola dampak dari perubahan cuaca di lingkungan sekitar. Kesimpulan proyek ini menunjukkan potensi besar dalam meningkatkan pemahaman kita terhadap dinamika cuaca dan memberikan solusi yang efisien dalam pemantauan kondisi atmosfer. Kedepannya proyek ini dapat dikembangkan dengan berbagai inovasi sistem pintar untuk menghindari dampak perubahan cuaca yang tidak tentu seperti sistem akan mematikan aliran listrik apabila cuaca buruk terjadi dan yang lainnya.

Dalam penelitian ini juga penggunaan algoritma K-Means Clustering untuk mengidentifikasi pola-pola dalam data sensor serta algoritma Random Forest untuk melakukan prediksi cuaca diharapkan dapat memberikan pemahaman yang lebih baik terkait perubahan cuaca secara real-time. Integrasi teknologi IoT dan analisis data ini bertujuan untuk menciptakan sistem deteksi cuaca yang informatif dan akurat, memungkinkan pengguna untuk mengantisipasi dan mengelola dampak dari perubahan cuaca di lingkungan sekitar. Kesimpulan proyek ini menunjukkan potensi besar dalam meningkatkan pemahaman kita terhadap dinamika cuaca dan memberikan solusi yang efisien dalam pemantauan kondisi atmosfer.

DAFTAR PUSTAKA

- [1] R. O. W. Muhamad Yusvin Mustar, "Implementasi Sistem Monitoring Deteksi Hujan dan Suhu Berbasis Sensor Secara Real Time (Implementation of Rain Detection and Temperature Monitoring System Based on Real Time Sensor)," *Semesta Tek.*, vol. 20, no. 1, hal. 20–28, 2017, [Daring]. Tersedia pada: <https://www.arduino.cc/en/Main/arduinoBoard>
- [2] A. K. Singh *et al.*, "Iot Live Weather Monitoring System Using Nodemcu Esp8266," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 4, no. 5, hal. 4926–4929, 2022, [Daring]. Tersedia pada: www.irjmets.com

- [3] Y. Rahut, R. Afreen, dan D. Kamini, "Smart weather monitoring and real time alert system using IoT," *Int. Res. J. Eng. Technol.*, vol. 848, hal. 848–854, 2018, [Daring]. Tersedia pada: www.irjet.net
- [4] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, dan A. A. Khan, "RETRACTED ARTICLE: A Review and State of Art of Internet of Things (IoT)," *Arch. Comput. Methods Eng.*, vol. 29, hal. 1395–1413, 2021, [Daring]. Tersedia pada: <https://link.springer.com/article/10.1007/s11831-021-09622-6>
- [5] R. Amaliani, A. J. Lubis, dan I. Lubis, "Rancang Bangun Miniatur Stasiun Cuaca Untuk Monitoring Curah Hujan, Suhu Dan Kelembaban Udara Area Lokal Menggunakan Berbasis IOT," *Semin. Nas. ...*, 2021, [Daring]. Tersedia pada: <http://prosiding.snastikom.com/index.php/SNASTIKOM2020/article/view/134%0Ahttp://prosiding.snastikom.com/index.php/SNASTIKOM2020/article/download/134/127>
- [6] Alifia Sekar Ratri, Vecky C. Poekoel, dan Arthur M. Rumagit, "Design Of Weather Condition Monitoring System Based On Internet Of Things," *J. Tek. Inform.*, vol. 17, no. 1, hal. 1–10, 2021, [Daring]. Tersedia pada: <https://ejournal.unsrat.ac.id/index.php/informatika>
- [7] B. Satria, "IoT Monitoring Suhu dan Kelembaban Udara dengan Node MCU ESP8266," *sudo J. Tek. Inform.*, vol. 1, no. 3, hal. 136–144, 2022, doi: 10.56211/sudo.v1i3.95.
- [8] M. Syahputra Novelan, "Monitoring System for Temperature and Humidity Measurement with DHT11 Sensor Using NodeMCU," *Int. J. Innov. Sci. Res. Technol.*, vol. 5, no. 10, hal. 124, 2020, [Daring]. Tersedia pada: www.ijisrt.com123
- [9] K. P. Sinaga dan M.-S. Yang, "Unsupervised K-Means Clustering Algorithm," *IEEE Access*, vol. 8, hal. 80716–80727, 2020, [Daring]. Tersedia pada: <https://ieeexplore.ieee.org/abstract/document/9072123>
- [10] N. Mladen dan B. Bajat, "Random Forest Spatial Interpolation," *MDPI*, vol. 12, no. 10, 2020, [Daring]. Tersedia pada: <https://www.mdpi.com/2072-4292/12/10/1687>