

BAB II

LANDASAN TEORI

Pada bab ini akan menerangkan teori-teori penunjang untuk merancang sebuah sistem navigasi menggunakan *sensor fusion* dengan kalman filter untuk meningkatkan akurasi dan keandalan data GPS pada *unmanned surface vehicle* (USV), adapun teori-teori tersebut akan dipaparkan sebagai berikut.

2.1 Sensor Fusion

Sensor fusion merupakan metode menggabungkan data-data sensorik atau data yang berasal dari sumber yang berbeda sehingga informasi yang dihasilkan memiliki ketidakpastian yang lebih kecil daripada yang mungkin terjadi ketika sumber data digunakan secara individual. Istilah pengurangan ketidakpastian dalam hal ini dapat diartikan lebih akurat, lebih lengkap, atau lebih dapat diandalkan, atau merujuk pada hasil dari suatu pandangan yang muncul, seperti penglihatan *stereoskopis* (penghitungan informasi kedalaman dengan menggabungkan gambar dua dimensi dari dua kamera dengan sedikit perbedaan sudut pandang).[8]

Sumber data untuk proses penggabungan tidak ditentukan berasal dari sensor yang identik. Seseorang dapat membedakan penggabungan langsung, penggabungan tidak langsung, dan penggabungan dari dua keluaran sebelumnya. Penggabungan langsung adalah fusi data sensor dari sekumpulan sensor *heterogen* atau *homogen*, sensor lunak, dan nilai riwayat data sensor, sedangkan penggabungan tidak langsung menggunakan sumber informasi seperti pengetahuan apriori tentang lingkungan dan masukan manusia. *Sensor fusion* juga dikenal sebagai *data fusion* (multi-sensor) dan merupakan bagian dari *information fusion*.

Tetapi, dengan lebih banyak data yang didapatkan dari lebih banyak sensor, menerjemahkannya secara matematis pun akan menjadi lebih sulit. Lantas, ada beberapa agregasi data yang dapat dilakukan, salah satunya menggunakan filter Kalman [9].

2.2 Kalman Filter

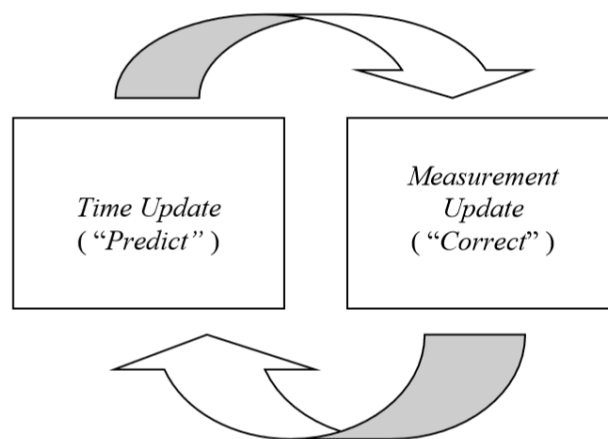
Teknik filter ini dinamakan berdasarkan penemunya, Rudolf E. Kalman. Kalman filter sangat berguna terutama dalam suatu sistem *sensor fusion* diantaranya sistem navigasi dan lingkungan dengan *gaussian noise*. Kalman filter merupakan sebuah *recursive* filter yang efisien, yang mengestimasi state pada *linear dynamic system* dari rentetan pengukuran noise. Disebut *recursive* sebab untuk menghitung state estimasi saat ini, hanya membutuhkan data state estimasi satu waktu sebelumnya dan data pengukuran saat ini [6].

Pada teori kontrol, kalman filter merupakan sebuah algoritma atau kumpulan persamaan matematika yang menghasilkan sebuah perhitungan yang efisien untuk mengestimasi state dari proses, dengan tujuan meminimalkan nilai *noise* atau variansi terhadap referensi lain. Filter ini sangat bagus dalam beberapa aspek misalnya mendukung estimasi state sebelumnya, saat ini dan berikutnya. Bahkan hal ini tetap dapat dilakukan meskipun model sistem yang sebenarnya tidak diketahui. [10]

Kalman filter akan mengestimasi proses dengan menggunakan bentuk pengendali *feedback*. Filter mengestimasi state proses pada beberapa waktu dan kemudian mendapatkan umpan balik (*feedback*) dalam bentuk pengukuran (*noise*). Oleh karena itu, persamaan kalman filter umumnya dibagi menjadi dua kelompok :

1. *Time update* atau yang biasa disebut juga sebagai *proses predict*, yaitu menggunakan estimasi state dari satu waktu sebelumnya untuk mendapatkan sebuah estimasi state pada saat ini.
2. *Measurement update* disebut juga sebagai *proses correct*, yaitu informasi pengukuran pada saat ini digunakan untuk memperbaiki prediksi, dengan harapan akan didapatkan state estimasi yang lebih akurat.

Sehingga dalam aplikasinya, algoritma kalman filter akan menggunakan proses berulang dari *predict* dan *correct*. [11]



Gambar 2.1 Proses Kalman Filter

Menurut Grewal dan Andrews (2001), berikut ini tahapan perhitungan kalman filter :

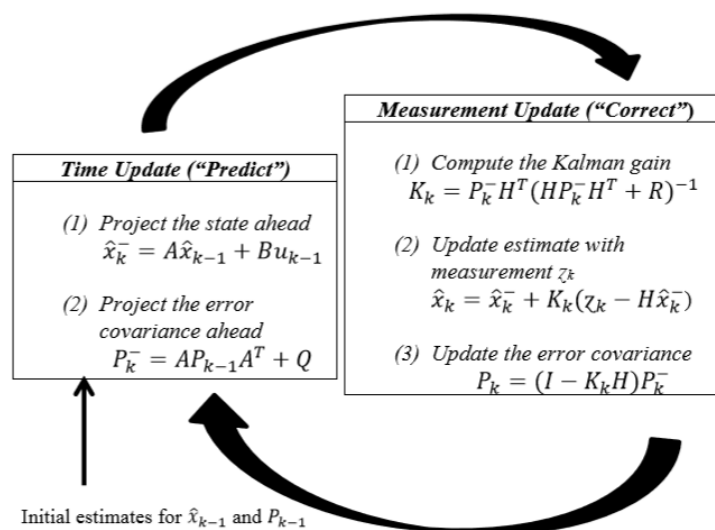
1. Tahap pertama "*Built a Model*" Tahap pertama ini adalah tahapan yang paling penting dalam proses kalman filter karena pada tahap ini menunjukkan apakah kalman filter bisa memproses dan menyelesaikan permasalahan yang ada. Berikut ini adalah persamaan dari proses awal kalman filtering.

$$x_k = Ax_{k-1} + Buk \quad (2.1)$$

$$z_k = Hx_k \quad (2.2)$$

x_k adalah nilai koordinat yang akan dihitung dengan menggunakan *linear stochastic equation* seperti persamaan 2.1. Persamaan 2.2 adalah persamaan yang menghitung nilai pengukuran dari model (Z_k). A, B, dan H adalah matriks dari model yang dibuat.

2. Tahap kedua “*Start the Process*” Melakukan proses perhitungan. Jika model bisa diproses dan diselesaikan dengan kalman filter maka di tahap kedua akan ditentukan beberapa parameter dari model. Dalam proses perhitungan kalman filter ada 2 persamaan utama yaitu persamaan *Time Update (Predict)* dan persamaan *Measurement Update (Correct)* yang ditunjukkan pada gambar 2.2 dibawah.



Gambar 2.2 Iterasi Kalman Filter[12]

Pada tahap *measurement update*, kita akan menghitung *Kalman Gain* (K_k) terlebih dahulu. Kalman Gain merupakan suatu nilai yang akan membuat kita memilih untuk lebih mempercayai hasil estimasi atau pengukuran. Nilai Kalman gain umumnya berkisar antara 0 hingga 1. Jika nilai kalman gain

mendekati 0 maka hasil pengukuran tidak akurat namun terbilang cukup stabil, jika nilai Kalman gain mendekati 1 maka hasil pengukuran lebih akurat namun kurang stabil. Langkah berikutnya adalah proses perhitungan untuk mendapatkan nilai *actual measurement* (Z_k). Kemudian untuk menghasilkan estimasi (\hat{x}_k) dengan menggabungkan perhitungan menggunakan persamaan (2) pada gambar 2.2. Langkah terakhir adalah mendapatkan estimasi kovarian (P_k) menggunakan persamaan (3) pada gambar 2.2.

Setelah perhitungan pada proses *Measurement Update* selesai maka dilanjutkan dengan perhitungan pada proses *Time Update*. Proses diulangi dengan estimasi sebelumnya yang digunakan untuk membuat estimasi (\hat{x}_{k-1}) baru. Sifat *recursive* adalah satu fitur yang paling menarik dari kalman filter, sifat *recursive* membuat implementasi praktis menjadi lebih layak yang didesain untuk mengoperasikan data secara langsung untuk setiap estimasi [11]. Pada persamaan tabel 2.1 dan tabel 2.2 terdapat variabel Q yang merupakan *process noise covariance* dan R yang merupakan *measurement noise covariance*. Dalam implementasi sebenarnya dari filter ini, R dan Q biasanya diukur terlebih dahulu untuk operasi filter. Penentuan parameter ini sangat tergantung dari model sistem yang digunakan. Perlu digaris bawahi bahwa R dan Q sebaiknya adalah nilai yang konstan, dengan kondisi konstan *estimation error covariance* (P_k) dan *Kalman Gain* akan stabil lebih cepat dan akan tetap konstan.

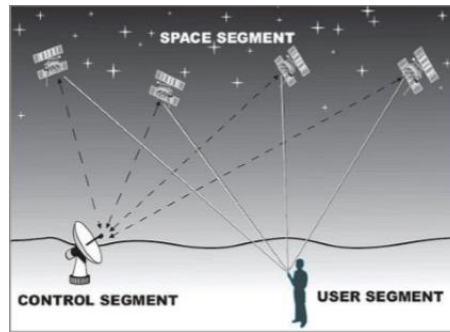
Selain itu ketika memulai proses kalman filter diperlukan *initial condition* dari P_{k-1} dan \hat{x}_{k-1} atau disebut juga P_0 dan \hat{x}_0 . Penentuan *initial condition* ini juga berdasarkan sistem yang digunakan. \hat{x}_0 biasanya didapatkan dengan memperkirakan state sistem pada kadaan awal, sedangkan P_0 sebaiknya bernilai

tidak sama dengan nol, sebab apabila $P_0 = 0$ akan menyebabkan filter menginisialisasi dan selalu percaya bahwa $\hat{x}_k = \hat{x}_0$ sedangkan penentuan P_0 sebenarnya tidak begitu penting karena filter akan menyesuaikan dengan sendirinya [11]. Setelah mengumpulkan semua informasi yang dibutuhkan dan memulai proses, tahap selanjutnya adalah melakukan iterasi melalui estimasi. Perlu diingat bahwa estimasi sebelumnya akan menjadi masukan untuk kondisi saat ini.

2.3 Global Navigation Satellites System (GNSS)

Global Navigation Satellite System (GNSS) merupakan suatu sistem yang dapat digunakan untuk mendapatkan data waktu dan posisi yang berasal dari hasil perhitungan sinyal yang secara terus menerus dipancarkan oleh konstelasi satelit. GNSS yang berfungsi saat ini dan mencakup area global adalah *Global Positioning System* (GPS) yang dimiliki oleh Amerika Serikat, *BeiDou Navigation Satellite System* (BDS) yang dimiliki oleh China, *Global Navigation Satellite System* (GLONASS) yang dimiliki oleh Rusia, dan Galileo yang dimiliki dan dikelola oleh Uni Eropa [7].

Sistem GNSS didesain untuk memberikan posisi dan kecepatan tiga-dimensi serta informasi mengenai waktu, secara kontinyu di seluruh dunia tanpa bergantung waktu dan cuaca, bagi banyak orang secara simultan. Ketelitian posisi dari GNSS dapat bervariasi dari beberapa milimeter (orde nol) sampai dengan puluhan meter. Untuk ketelitian kecepataannya dapat beberapa cm/s dan beberapa nanodetik untuk ketelitian waktunya. Ketelitian posisi yang diperoleh akan bergantung pada beberapa faktor yaitu faktor medan, metode penentuan posisi, geometri satelit, tingkat ketelitian data, dan metode pengolahan datanya.



Gambar 2.3 Ilustrasi Penentuan Posisi GPS[13]

Pada dasarnya sistem pemosisian membutuhkan minimal tiga buah sinyal satelit untuk mendapat data posisi menggunakan metode *trilateration*. Pada pengukuran GPS, setiap epoknya memiliki empat paramater yang harus ditentukan : yaitu 3 paramater koordinat X,Y,Z atau L,B,h dan satu paramater kesalahan waktu akibat ketidaksinkronan jam osilator di satelit dengan jam di receiver GPS/GNSS. Oleh karena diperlukan minimal pengukuran jarak ke empat satelit [14].



Gambar 2.4 Modul GPS [15]

Modul GNSS biasanya berukuran kecil serta memiliki antena dan prosesor yang secara langsung menerima data yang dikirim oleh satelit dan menghitung posisi dan waktu dengan cepat. Modul penerima GNSS menggunakan konstelasi satelit dan stasiun bumi untuk menghitung posisi dan waktu hampir di mana saja di bumi.

Disetiap waktunya, setidaknya ada 24 satelit aktif mengorbit lebih dari 12.000 mil di atas bumi. Posisi satelit diatur sedemikian rupa agar langit di atas lokasi kita akan selalu berisi paling tidak 12 satelit. Tujuan utama dari 12 satelit adalah untuk mengirimkan informasi kembali ke bumi melalui frekuensi radio (mulai 1,1 - 1,5 GHz). Dengan informasi ini modul penerima GNSS dapat menghitung posisi dan waktu.

Sebuah penerima GNSS receiver harus berada pada lokasi *line-of sight* (LoS) terhadap ketiga satelit tersebut untuk menentukan posisi, maka dari itu GNSS hanya ideal untuk digunakan di lokasi *outdoor*. Meskipun masih memungkinkan jika mendapatkan posisi pada kurang dari 4 satelit, *error* dari posisi yang didapat bisa menjadi besar. Pembacaan paling akurat bisa didapat ketika langit cerah, jauh dari penghalang dan mengunci pada lebih dari 4 satelit.[15]

Data yang diberikan GNSS memiliki format khusus sebagai standar internasional. NMEA adalah singkatan dari *National Marine Electronics Association*. Saat ini format NMEA menjadi standar bagi semua jenis penerima GNSS. Penggunaan format data berbentuk NMEA memberikan pengguna memiliki keuntungan pada gabungan *hardware* dan *software*. Data NMEA diawali oleh karakter “\$” kemudian diikuti oleh 2 karakter *Talker ID* dan karakter identifikasi. *Talker ID* pada data NMEA ditunjukkan pada **Tabel 2.1**.

Tabel 2. 1 Kode *Talker ID* pada data NMEA

Talker ID	Keterangan
-----------	------------

GP	Digunakan ketika sistem satelit GPS yang terkunci
GL	Digunakan ketika sistem satelit GLONASS yang terkunci
GA	Digunakan ketika sistem satelit GALILEO yang terkunci
GN	Digunakan ketika lebih dari satu sistem satelit yang terkunci.

Format data NMEA selalu diawali karakter “\$” dan disetiap bagian data dipisahkan oleh karakter “,”. Berikut ini adalah penjelasan bagian data NMEA [16]

1. *Talker ID* (XX), Kode **GN** pada **GNGGA** merupakan data yang didapatkan tersebut merupakan data posisi pada GNSS yang menggunakan dua buah sistem navigasi satelit baik itu GPS+GLONASS atau GPS+BeiDou.
2. *SENTENCE ID* (XXX), kode **GGA** menyatakan bahwa data tersebut adalah data pemosisian secara global dengan informasi tambahan yang lengkap.
3. *UTC of Position*, (hhmmss.ss), merupakan informasi waktu saat data didapatkan.
4. *Latitude* (xxxx.xxxxxxx), menyatakan posisi pada koordinat latitude
5. *Hemisphere of latitude* (N/S), menyatakan posisi koordinat latitude pada bagian bumi Utara atau Selatan.
6. *Longitude* (xxxx.xxxxxxx), menyatakan posisi pada koordinat longitude
7. *Hemisphere of longitude* (E/W), menyatakan posisi koordinat latitude pada bagian bumi Timur atau Barat.

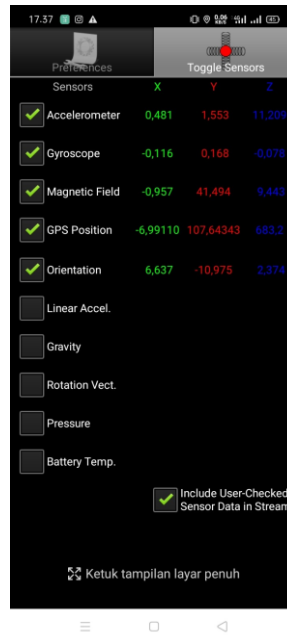
8. *GPS quality indicator* (X), 0: GNSS tidak tersedia; 1: GNSS valid; 4:RTK fix; 5:RTK float.
9. *Number of Satellites* (xx), menyatakan jumlah satelit yang digunakan untuk menentukan posisi.
10. *HDOP* (xx.x), menyatakan tingkat error kepresisian pada pemosisian horizontal.
11. *Altitude* (x.xx), menyatakan ketinggian penerima GNSS.
12. *Unit of altitude* (M), satuan hitung ketinggian.
13. *Geoidal separation* (x.xx), menyatakan ketinggian penerima GNSS.
14. *Unit of geoidal eparation* (M), satuan hitung *geoidal separation*.
15. *Age of differential data*, menyatakan perbedaan waktu
16. *Differential reference station ID*, menyatakan ID stasiun referensi diferensial.

2.4 Cara kerja Navigasi pada Smartphone

Prinsip kerja navigasi pada smartphone sejatinya memanfaatkan sistem GNSS yang pada umumnya menggunakan pembacaan koordinat dari *Global Positioning System* (GPS). Cara kerja pengambilan data koordinat dari smartphone sejatinya hamper sama sebagaimana smartphone yang terhubung dengan operator telekomunikasi melalui komunikasi dengan *Base Transceiver Station* (BTS) [17]. BTS merupakan suatu elemen paling luar yang berhubungan dengan suatu ponsel yang terhubung dengan suatu operator jaringan tertentu sehingga pada dasarnya ketika kita membawa *smartphone* kita bergerak, sejatinya smartphone kita menangkap sinyal dari BTS yang berbeda. Banyaknya menara-menara BTS dikarenakan hal tersebut merupakan syarat suatu operator telekomunikasi agar dapat memiliki sinyal yang baik di semua tempat sehingga operator telekomunikasi

dapat mengetahui keberadaan lokasi dari suatu *smartphone* yang menangkap sinyal dari menara BTS secara presisi.

Selain itu *smartphone* pada generasi sekarang sejatinya telah dilengkapi oleh sebuah fitur yang dinamakan *Assisted GPS* (A-GPS). A-GPS merupakan sistem penyempurna dari *Global Positioning System* (GPS) sebagai satelit penentu posisi di belahan bumi [18]. A-GPS juga merupakan sebuah sistem yang mempercepat *Time To First Fix* (TTFF) atau kecepatan menentukan posisi pertama kali. Hal tersebut sangat dibutuhkan mengingat satelit-satelit GPS, GLONASS, BeiDou yang selalu bergerak dan tidak selalu berada pada posisi yang sama. Sistem navigasi pada *smartphone* harus dapat mengetahui satelit mana yang relevan untuk perhitungan posisinya sehingga untuk mengetahui mana yang relevan, perlu diketahui dari yang posisinya paling dekat untuk mempermudah perhitungan posisi. Saat pertama kali dihidupkan, sistem navigasi pada *smartphone* akan mencari ke semua satelit dan menunggu laporan posisi, sehingga waktu dalam mencari satelit yang paling relevan tersebut dinamakan TTFF. Dengan A-GPS yang memanfaatkan jaringan seluler, sistem navigasi dalam *smartphone* akan mendapatkan informasi mengenai satelit yang paling relevan untuk diperhitungkan. Dengan demikian, TTFF akan menjadi lebih cepat. A-GPS dapat bekerja dengan baik karena stasiun-stasiun pemancar sinyal seluler di darat atau menara-menara BTS yang telah memiliki kepastian lokasinya masing-masing sehingga dengan mengirimkan laporan lokasinya sendiri dan data satelit serta keberadaannya secara akurat ke *smartphone* akan membuat sistem navigasi di *smartphone* dapat melakukan kalkulasi posisi dengan cepat.



Gambar 2.5 GPS IMU Streaming

Selain dari faktor operator jaringan, nyatanya *smartphone* generasi sekarang sudah terintegrasi sistem inersia didalamnya sehingga dapat membantu penerima lokasi di *smartphone* untuk mendapatkan data posisi yang lebih akurat. Hal tersebut dapat dibuktikan dengan menggunakan *software* bernama *GPS IMU streaming* yang bisa diunduh untuk *smartphone*, sehingga kita dapat melihat bahkan mengolah data-data tersebut.

2.5 Google Geolocation API

Geolocation API berfungsi untuk membakukan sebuah antarmuka untuk mengambil informasi lokasi geografis untuk perangkat. Ini mendefinisikan satu set objek, *ECMA Script standar compliant*, yang mengeksekusi dalam aplikasi klien memberikan lokasi perangkat klien melalui konsultasi Server Informasi Lokasi, yang transparan untuk antarmuka pemrograman aplikasi (API). Sumber yang paling umum dari informasi lokasi yang alamat IP, Wi-Fi dan Bluetooth alamat MAC,

radio-frekuensi identifikasi (RFID), Wi-Fi lokasi koneksi, atau perangkat *Global Positioning System* (GPS) dan GSM / CDMA. Lokasi dikembalikan dengan akurasi yang diberikan tergantung pada sumber informasi lokasi terbaik yang tersedia [19].

Google *Geolocation* API merupakan salah satu fitur layanan yang disediakan oleh Google yang memungkinkan pengguna untuk memberikan lokasi mereka ke aplikasi web jika mereka menginginkannya. Untuk alasan privasi, pengguna dimintai izin untuk melaporkan informasi lokasi. Maka dari itu, fitur google geolocation tersebut dijadikan fitur untuk mendapatkan data lokasi dari *smartphone* yang nantinya akan *diffusion*kan dengan data koordinat dari pembacaan sensor GPS.

2.6 Sensor Kompas

Kompas merupakan alat navigasi untuk menentukan arah berupa sebuah jarum panah penunjuk magnetis yang bebas menyelaraskan dirinya dengan medan magnet bumi secara akurat. Kompas memberikan rujukan arah tertentu sehingga sangat membantu dalam bidang navigasi. Arah mata angin yang ditunjuk kompas adalah utara, timur, selatan dan barat [20]. Apabila digunakan bersama-sama dengan jam dan sectan, maka kompas akan lebih akurat dalam menunjukkan arah. Alat ini membantu perkembangan dunia perdagangan maritim dengan membuat perjalanan jauh lebih aman dan efisien dibandingkan saat manusia masih berpedoman pada kedudukan bintang untuk menentukan arah. Dalam kompas, medan magnet yang diukur adalah medan magnet bumi yang mengalir dari utara ke selatan.[15]



Gambar 2.6 Sensor Kompas

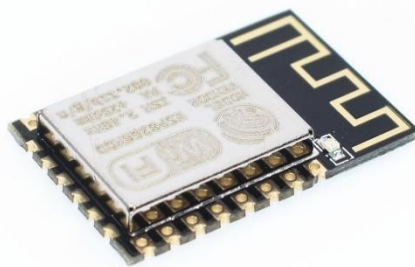
Penggunaan sensor kompas dalam penelitian ini sangat berpengaruh dalam sistem gerak rotasi dari wahana untuk mencapai *waypoint*. Dalam pengaplikasiannya penempatan sensor kompas perlu diperhatikan, karena sensor kompas yang membaca medan magnet bumi biasanya terinterferensi oleh putaran dari motor dc yang mengandung gelombang elektromagnetik [21] sehingga peletakan sensor kompas perlu diperhatikan untuk meningkatkan keandalan sistem wahana yang telah dirancang.

2.7 Mikrokontroler

Mikrokontroler adalah suatu chip berupa IC (*Integrated Circuit*) yang dapat menerima sinyal input, mengolahnya dan memberikan sinyal output sesuai dengan program yang diisikan ke dalamnya. Sinyal input mikrokontroler berasal dari sensor yang membaca informasi dari lingkungan atau parameter tertentu sedangkan sinyal output ditujukan kepada aktuator yang dapat memberikan efek ke lingkungan. Jadi secara sederhana mikrokontroler dapat diibaratkan sebagai otak dari suatu perangkat/produk yang mampu berinteraksi dengan lingkungan sekitarnya.[15]

Mikrokontroler pada dasarnya adalah komputer dalam satu chip, yang di dalamnya terdapat mikroprosesor, memori, jalur Input/Output (I/O) dan perangkat pelengkap lainnya. Kecepatan pengolahan data pada mikrokontroler lebih rendah jika dibandingkan dengan PC. Pada PC kecepatan mikroprosesor yang digunakan saat ini telah mencapai orde GHz, sedangkan kecepatan operasi mikrokontroler pada umumnya berkisar antara 1 – 16 MHz. Begitu juga kapasitas RAM dan ROM pada PC yang bisa mencapai orde Gbyte, dibandingkan dengan mikrokontroler yang hanya berkisar pada orde byte/Kbyte. [22]

Dalam penelitian ini, peneliti menggunakan dua jenis mikrokontroler yakni ESP8266 dan Atmega Series yang terintegrasi pada Arduino. Pemilihan ESP8266 dikarenakan penelitian ini membutuhkan komunikasi nirkabel seperti WiFi untuk mengirimkan data-data. ESP8266 merupakan platform yang sangat murah tetapi benar-benar efektif untuk digunakan berkomunikasi atau kontrol melalui internet baik digunakan secara *stand alone* (berdiri sendiri) maupun dengan menggunakan mikrokontroler tambahan dalam hal ini Arduino sebagai pengendalinya.



Gambar 2.7 ESP8266

ESP8266 merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti Arduino agar dapat terhubung langsung dengan

wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode wifi yaitu Station, Access Point dan Both (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler. [5] Firmware default yang digunakan oleh perangkat ini menggunakan AT Command, selain itu ada beberapa Firmware SDK yang digunakan oleh perangkat ini berbasis opensource yang diantaranya adalah NodeMCU dengan menggunakan *basic programming lua*, *MicroPython* dengan menggunakan *basic programming python* dan *AT Command* dengan menggunakan perintah perintah *AT command*.