BAB II
TINJAUAN PUSTAKA

2.1 Protokol

Protokol dalam jaringan komputer merujuk pada serangkaian aturan atau standar yang memungkinkan perangkat-perangkat dalam jaringan untuk berkomunikasi. Protokol mengatur bagaimana data dibungkus, dikirim, diterima, dan diinterpretasikan, serta bagaimana perangkat dapat mendeteksi dan memperbaiki kesalahan dalam proses transmisi data. Protokol diperlukan untuk memastikan bahwa perangkat yang berbeda dapat berkomunikasi dengan baik, meskipun mereka mungkin menggunakan perangkat keras atau perangkat lunak yang berbeda. Protokol biasanya diorganisir dalam beberapa lapisan, yang dikenal sebagai stack protokol**.** Setiap lapisan dalam stack memiliki tugas spesifik dan bekerja secara independen, tetapi tetap terkoordinasi dengan lapisan lain untuk memastikan komunikasi yang efektif. Contoh stack protokol yang umum digunakan adalah model OSI (Open Systems Interconnection) yang terdiri dari tujuh lapisan, dan model TCP/IP yang lebih sederhana dengan empat lapisan. Beberapa protokol yang umum digunakan dalam komunikasi data antara lain HTTP (Hypertext Transfer Protocol) untuk pertukaran data di web; FTP (File Transfer Protocol) untuk transfer file; dan SMTP (Simple Mail Transfer Protocol**)** untuk pengiriman email.

PROTOCOL

Gambar 2.1 Protocol

2.2 Internet Of Things (IoT)

IoT adalah sistem embedded yang bertujuan untuk memperluas pemanfaatan dari konektivitas internet yang tersambung secara terus-menerus. Kemampuan seperti berbagi data, remote control, dan sebagainya, termasuk juga pada benda di dunia nyata contohnya seperti bahan pangan,elektronik, peralatan yang terhubung dengan sensor dan terhubung dengan jaringan.Keterkaitan objek dengan koneksi internet sebagai dasar pengembangan semua layanan[9]. Benda-benda fisik diintegrasikan ke dalam jaringan informasi secara berkesinambungan,dan di mana benda-benda fisik tersebut berperan secara aktif dalam proses bisnis. Tersedia layanan pintar yang saling terkonesi, mencari dan mengubah statu s mereka sesuai dengan setiap informasi yang dikaitkan, disamping memperhatikan masalah privasi dan keamanan.Tapahan proses kerja dari Internet of Things dengan memanfaatkan pemrograman di setiap perintah untuk sebuah instruksi kepada mesin tanpa bantuan manusia . Dengan menggunakan sambungan atau koneksi internet. Seperti bagaimana mengolah data yang diperoleh dari peralatan eletronik melalui sebuah interface antara pengguna dan peralatan itu. Penggunaan sensor secara real time mengkonversikan ke dalam mesin format yang dimengerti sehingga akan mudah dipertukarkan antara berbagai bentuk format data (Thing). Kecerdasan intelejensi dan kontrol automatisasi merupakan bagian dari konsep asli Internet of Things. Perlu dilakukan lagi penelitian mendalam konsep Internet of Things dan kontrol automatisasi agar pada masa depan Internet of Thing sakan menjadi jaringan yang terbuka dan semua perintah dilakukan secara auto, terkelompok atau cerdas, objek virtual (avatar) dan dapat dioperasikan dengan mudah, bertindak secara independen sesuai dengan konteks, situasi atau lingkunganyang dihadapi .Di dalam membangun Internet Of Things para engineer harus memperhatikan ketiga aspek yaitu Ukuran, ruang, dan waktu. Dalam melakukan pengembangan IoT faktor waktu yang biasanya menjadi kendala. Biasanya dibutuhkan waktu yang lama karena menyusun sebuah jaringan kompleks di dalam IoT tidak lah mudah dan tidak dapat dilakukan oleh sembarang orang[10].

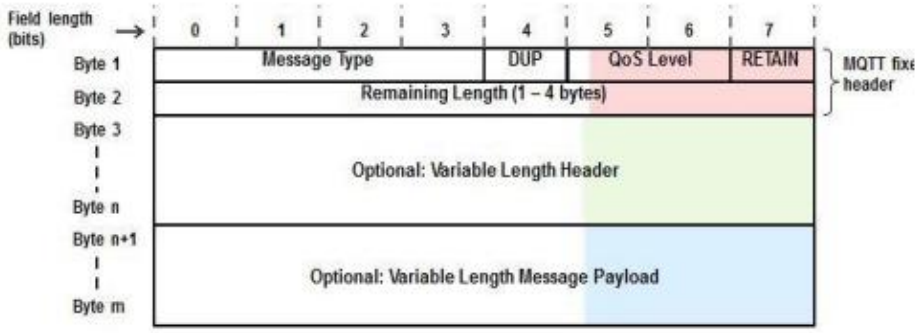
Gambar 2.2 Internet Of Things

2.3 MQTT

MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi yang dirancang untuk memberikan komunikasi yang ringan dan efisien antar perangkat, terutama dalam konteks aplikasi Internet of Things (IoT) yang sering kali beroperasi di lingkungan dengan keterbatasan sumber daya. MQTT menggunakan model komunikasi publish/subscribe, di mana perangkat dapat bertindak sebagai penerbit (publisher) yang mengirimkan pesan, atau sebagai pelanggan (subscriber) yang menerima pesan, atau keduanya. Dalam sistem ini, penerbit mengirim pesan ke topik tertentu pada broker MQTT, dan pelanggan yang telah berlangganan topik tersebut akan menerima pesan yang relevan. Metode ini memungkinkan komunikasi yang lebih efisien karena pelanggan hanya mendapatkan pesan yang terkait dengan topik yang mereka minati. MQTT mendukung beberapa tingkat QoS (Quality of Service) yang memastikan keandalan dalam pengiriman pesan, dari yang paling dasar tanpa konfirmasi hingga tingkat yang memastikan pesan dikirim dan diterima dengan pengakuan. Keunggulan utama dari MQTT termasuk penggunaan bandwidth yang sangat rendah, konsumsi daya yang efisien, dan kemampuan untuk berfungsi dengan baik pada jaringan yang mungkin tidak stabil. Oleh karena itu, MQTT sangat cocok untuk digunakan dalam aplikasi yang melibatkan banyak perangkat yang membutuhkan komunikasi yang real-time dan andal.

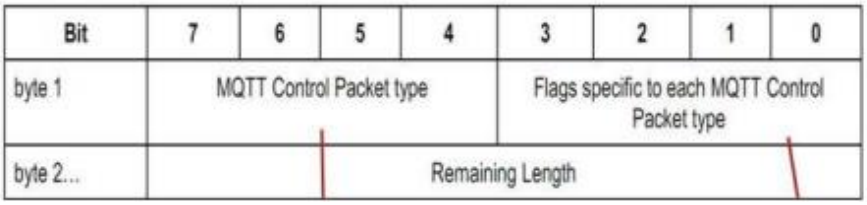
2.4 MQTT Frame Format

Pesan dalam protokol MQTT selalu memiliki header dengan panjang tetap yang terdiri dari 2 byte, serta dapat memiliki header variabel dan payload pesan yang sifatnya opsional. Walaupun field opsional biasanya menambah kompleksitas pemrosesan protokol, MQTT didesain khusus untuk jaringan yang memiliki keterbatasan bandwidth dan rentan terhadap gangguan (seperti jaringan nirkabel). Oleh karena itu, field opsional digunakan untuk mengurangi jumlah data yang perlu dikirimkan. MQTT menerapkan urutan byte dan bit sesuai dengan standar jaringan[11].



Gambar 2.3 MQTT Frame

Berikut ini adalah gambaran umum tentang field header tetap pada Gambar



Gambar 2.4 MQTT Header Field

RETAIN (menyimpan pesan terakhir): RETAIN=1 dalam pesan PUBLISH menginstruksikan server untuk menyimpan pesan untuk topik ini. Ketika ada klien baru yang berlangganan ke topik tersebut, server akan mengirimkan pesan yang disimpan. Dalam skenario aplikasi tipikal, klien hanya menerbitkan perubahan data, sehingga pelanggan menerima nilai terakhir yang diketahui.

Remaining length (RL): Field panjang yang tersisa mengkodekan jumlah panjang dari:

a. (Opsional) header variabel

b. (Opsional) payload

Untuk menghemat bit, panjang yang tersisa adalah field dengan panjang variabel antara 1 hingga 4 byte. Bit paling signifikan dari byte panjang memiliki arti sebagai "continuation bit" (CB). Jika ada byte lain yang mengikuti, bit ini diatur ke 1.

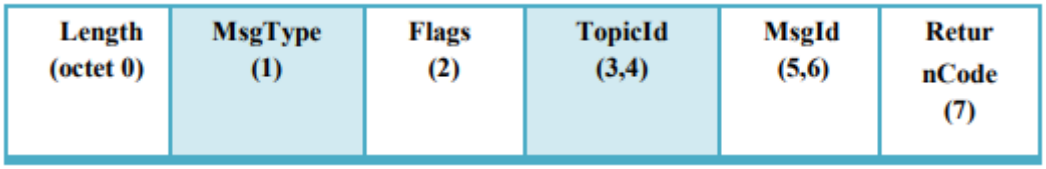
2.5 SUBSCRIBE

Subscribe adalah proses di mana suatu klien mendengarkan atau menerima pesan-pesan yang dikirim ke topik tertentu. Klien yang berlangganan ke topik akan menerima semua pesan yang dikirim ke topik tersebut oleh klien lain yang berperan sebagai publisher[11].

Pesan **SUBSCRIBE** digunakan oleh klien untuk berlangganan ke nama topik tertentu. Formatnya diilustrasikan dalam Gambar

Gambar 2.5 SUBSCRIBE

Dan untuk **SUBACK** Pesan ini dikirim oleh sebuah gateway ke klien sebagai tanda terima dan pemrosesan pesan **SUBSCRIBE**. Formatnya diilustrasikan dalam Gambar.



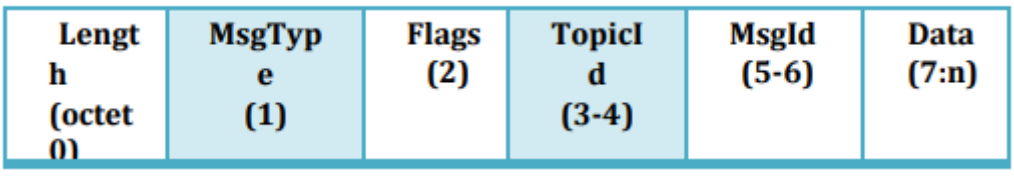
Gambar 2.6 SUBACK

UNSUBSCRIBE: Pesan UNSUBSCRIBE dikirim oleh klien ke GW untuk berhenti berlangganan dari topik yang disebutkan. Formatnya dijelaskan Gambar sebelumnya.

UNSUBACK: Pesan UNSUBACK dikirim oleh GW untuk mengonfirmasi penerimaan dan pemrosesan Pesan UnSubscribe

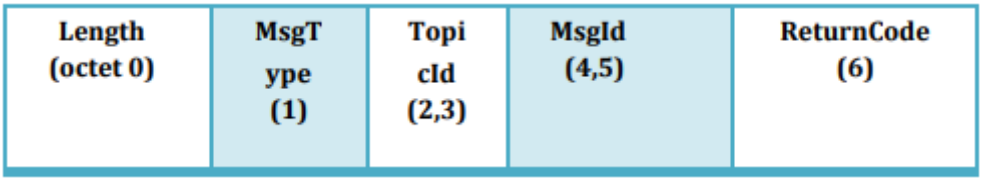
2.6 PUBLISH

Publish adalah proses pengiriman pesan oleh suatu perangkat atau aplikasi ke sebuah topik yang spesifik di broker MQTT. Topik ini berfungsi sebagai kategori yang mengorganisir pesan-pesan sehingga klien lain yang berlangganan pada topik tersebut dapat menerima pesan tersebut[11].



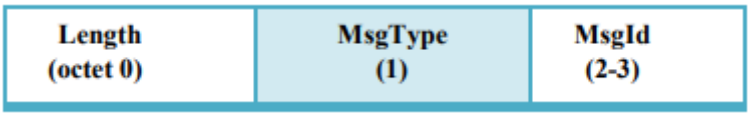
Gambar 2.7 PUBLISH

Untuk Pesan **PUBACK** digunakan oleh gateway atau klien untuk mengakui penerimaan dan pemrosesan pesan **PUBLISH**, terutama pada tingkat QoS 1 atau 2. Selain itu, pesan **PUBACK** bisa dikirim sebagai tanggapan jika ada kesalahan pada pesan **PUBLISH**, dengan alasan kesalahan tersebut ditampilkan dalam field ReturnCode. Struktur pesan ini dapat dilihat pada Gambar.



Gambar 2.8 PUBACK

Untuk Pesan **PUBREC**, **PUBREL**, dan **PUBCOMP**, sebagaimana digunakan dalam MQTT, berfungsi bersama dengan pesan **PUBLISH** yang memiliki tingkat QoS 2. Format pesan-pesan ini dapat ditemukan di Gambar berikut.



Gambar 2.9 PUBREC

2.7 Quality Of Service

Quality of Service adalah teknik untuk mengelola bandwidth, delay, dan packet loss untuk aliran dalam jaringan. Tujuan dari mekanisme QoS adalah mempengaruhi setidaknya satu diantara empat parameter dasar QoS yang telah ditentukan. QoS didesain untuk membantu end user (client) menjadi lebih produktif dengan memastikan bahwa user mendapatkan performansi yang handal dari aplikasi-aplikasi berbasis jaringan. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. QoS merupakan suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan[12].

Tabel 2. 1 Quality Of Service

NILAI	PERSENTASE	INDEKS
3,8 – 4	100 %	Sangat bagus
3 – 3,79	75 – 94,75 %	Bagus
2 – 2,99	50 – 74,75 %	Sedang

1 – 1,99	25 – 49,75 %	Buruk

Parameter QoS yang digunakan :

1. Throughput

   Throughput merupakan jumlah total kedatangan paket yang sukses yang diamati pada destination selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut. Throughput merupakan kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. Biasanya throughput selalu dikaitkan dengan bandwidth karena throughput memang bisa disebut juga dengan bandwidth dalam kondisi yang sebenarnya.

Tabel 2. 2 Throughput

Kategori Throughput	**Indeks**	**Throughput**
Sangat Bagus	76 – 100%	4
Bagus	51 – 75%	3
Sedang	26 – 50%	2
Buruk	25%	1

2. Packet Loss

   Packet loss didefinisikan sebagai kegagalan transmisi paket IP mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinkan, diantaranya yaitu :

   1.Terjadinya overload trafik didalam jaringan.

   2.Tabrakan (congestion) dalam jaringan.

   3.Error yang terjadi pada media fisik.

   4.Kegagalan yang terjadi pada sisi penerima antara lain bisa disebabkan karena overflow yang terjadi pada buffer.

Tabel 2. 3 Packet Loss

Kategori Degedrasi	Packet Loss	Indeks
Sangat Bagus	0 – 2%	4
Bagus	3 – 14%	3
Sedang	15 – 24%	2
Buruk	>25%	1

3. Delay

   Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. Delay di dalam jaringan dapat digolongkan sebagai berikut :

   - Packetization delay

     Delay yang disebabkan oleh waktu yang diperlukan untuk proses pembentukan paket IP dari informasi user. Delay ini hanya terjadi sekali saja, yaitu di sumber informasi.

   - Queuing delay

     Delay ini disebabkan oleh waktu proses yang diperlukan oleh router dalam menangani transmisi paket di jaringan. Umumnya delay ini sangat kecil, kurang lebih sekitar 100 micro second.

   - Delay propagasi

     Proses perjalanan informasi selama di dalam media transmisi, misalnya kabel SDH, coaxial atau tembaga, menyebabkan delay yang disebut dengan delay propagasi.

Tabel 2. 4 Delay

Kategori Latency	Delay	Indeks
Sangat Bagus	< 150 m/s	4
Bagus	150 s/d 300 m/s	3
Sedang	300 s/d 450 m/s	2
Buruk	> 450 m/s	1

2.8 Virtual Private Server (VPS)

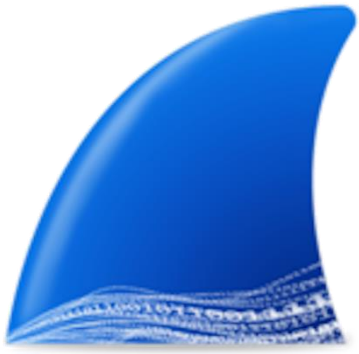
Virtualisasi adalah sebuah teknologi yang memungkinkan pembagian sebuah mesin fisik dengan kapasitas besar menjadi beberapa mesin virtual. Setiap mesin virtual ini berfungsi secara independen dengan sistem operasi dan perangkat lunaknya masing-masing. VPS (Virtual Private Server) merupakan metode yang mempartisi sumber daya server fisik menjadi beberapa server virtual. Setiap virtual machine (VM) memiliki kemampuan untuk menjalankan sistem operasi secara mandiri, seolah-olah berfungsi sebagai server fisik yang terpisah. Pengguna juga dapat mengelola VM secara independen dari hypervisor, yaitu sistem operasi utama dari server fisik tersebut. Untuk mengontrol VPS, pada sistem operasi Windows biasanya digunakan protokol RDP (Remote Desktop Protocol) melalui port TCP/UDP 3389, sementara untuk sistem operasi Linux digunakan SSH (Secure Shell) melalui port default TCP/UDP 22. Jika VPS dilengkapi dengan control panel, pengguna dapat mengelola berbagai aspek seperti script, pengguna, proses, file, backup, restore, dan fitur lainnya. VPS bertindak seperti server yang terpisah dengan proses, pengguna, dan file sistemnya sendiri, serta memberikan akses penuh ke root. Setiap VPS memiliki alamat IP, nomor port, tabel, aturan penyaringan, dan aturan routing yang unik. Pengguna juga bisa melakukan konfigurasi file untuk sistem dan layanan perangkat lunak. Dengan menggunakan VPS, pengguna tidak perlu melakukan perawatan server secara langsung, karena penyedia layanan VPS akan menangani pemeliharaan rutin seperti upgrade sistem operasi, backup sistem, dan sebagainya[13]

2.9 Paho MQTT

Paho MQTT adalah bagian dari proyek Eclipse Paho yang menyediakan implementasi untuk protokol MQTT (Message Queuing Telemetry Transport) di berbagai platform dan bahasa pemrograman. MQTT adalah protokol komunikasi yang ringan dan efisien, ideal untuk aplikasi Internet of Things (IoT) yang sering melibatkan perangkat yang beroperasi di jaringan dengan koneksi tidak stabil dan bandwidth terbatas. Proyek Paho menyediakan klien MQTT yang dapat digunakan untuk mengirim dan menerima pesan melalui broker MQTT. Ini mendukung banyak bahasa pemrograman seperti Python, Java, C, dan JavaScript, menjadikannya sangat fleksibel dan cocok untuk berbagai lingkungan pengembangan.Dengan memanfaatkan Paho MQTT, pengembang dapat dengan mudah menambahkan komunikasi berbasis MQTT ke dalam aplikasi mereka. Ini cocok untuk perangkat kecil seperti sensor dan aktuator, serta aplikasi yang lebih kompleks seperti sistem manajemen energi atau platform analitik untuk IoT. Selain itu, Paho MQTT menawarkan fitur-fitur seperti berbagai tingkat kualitas layanan (QoS), penyimpanan pesan, dan dukungan untuk Last Will and Testament (LWT), yang memberikan lebih banyak pilihan dalam pengelolaan komunikasi

2.10 Wireshark

Wireshark adalah alat analisis jaringan yang bersifat open-source dan sangat terkenal karena kemampuannya untuk menangkap dan menganalisis data yang mengalir melalui jaringan komputer. Dengan menggunakan Wireshark, pengguna dapat mengamati detail bagaimana paket data ditransmisikan dan diterima dalam jaringan, serta dapat mengevaluasi protokol yang terlibat pada setiap lapisan model OSI (Open Systems Interconnection). Wireshark memberikan kemampuan kepada pengguna untuk mendeteksi dan mendiagnosis masalah jaringan, seperti tingginya latensi, hilangnya paket, atau ancaman jaringan seperti sniffing atau serangan man-in-the-middle. Alat ini berguna dalam berbagai konteks, mulai dari troubleshooting jaringan, pengembangan protokol, hingga kegiatan pendidikan di bidang jaringan. Wireshark mendukung berbagai macam protokol jaringan dan dapat digunakan untuk menganalisis berbagai jenis lalu lintas, termasuk HTTP, TCP, UDP, DNS, dan banyak lagi. Selain itu, Wireshark dilengkapi dengan fitur filter yang kuat, yang memungkinkan pengguna untuk memfokuskan analisis pada jenis paket atau protokol tertentu, sehingga lebih mudah mengekstrak informasi yang relevan.



Gambar 2.10 Wireshark

2.11 HiveMQ

HiveMQ adalah broker MQTT yang dirancang untuk menyediakan komunikasi data yang aman, andal, dan real-time antara perangkat Internet of Things (IoT) dan sistem backend. HiveMQ dirancang untuk mendukung skala besar, mampu menangani jutaan koneksi klien secara simultan, yang menjadikannya sangat cocok untuk aplikasi IoT berskala industri. HiveMQ dibangun dengan fokus pada kinerja tinggi, kemampuan untuk berkembang, dan fleksibilitas, serta dilengkapi dengan fitur-fitur keamanan seperti enkripsi TLS, autentikasi, dan pengendalian akses. HiveMQ mendukung berbagai protokol jaringan dan standar industri, sehingga memudahkan integrasi dengan infrastruktur dan sistem yang ada. Selain itu, HiveMQ menawarkan fitur monitoring dan manajemen, yang memungkinkan pengguna untuk memantau dan mengelola koneksi klien, penggunaan bandwidth, dan kinerja sistem secara langsung. Keunggulan utama HiveMQ termasuk kemampuannya untuk bekerja di lingkungan cloud dan hybrid, menyediakan koneksi yang andal dan aman, serta mendukung berbagai fitur yang diperlukan untuk manajemen perangkat IoT secara efisien. Dengan HiveMQ, organisasi dapat dengan mudah mengembangkan solusi IoT yang scalable, handal, dan aman[14].

Gambar 2.11 HiveMQ

2.12 Mosquitto

Mosquitto adalah broker MQTT yang bersifat open-source dan sangat populer, dirancang untuk menyediakan komunikasi yang efisien dan ringan antara perangkat melalui protokol MQTT (Message Queuing Telemetry Transport). Mosquitto mendukung protokol MQTT versi 3.1 dan 3.1.1, serta dirancang untuk memenuhi kebutuhan komunikasi dalam berbagai aplikasi Internet of Things (IoT), baik untuk perangkat dengan sumber daya terbatas maupun server skala besar. Sebagai broker MQTT, Mosquitto berfungsi sebagai perantara yang mengelola pengiriman pesan dari penerbit (publishers) dan mendistribusikannya ke pelanggan (subscribers) yang berlangganan topik tertentu. Mosquitto dikenal karena penggunaan sumber daya yang efisien, performa tinggi, dan kemampuannya untuk mengelola ribuan koneksi klien secara simultan. Mosquitto menyediakan fitur-fitur penting seperti berbagai tingkat Quality of Service (QoS) untuk memastikan pengiriman pesan yang dapat diandalkan sesuai dengan kebutuhan aplikasi. Selain itu, Mosquitto juga mendukung fitur Last Will and Testament (LWT), yang memungkinkan perangkat untuk mengirim pesan terakhir jika koneksi terputus, serta mendukung sistem keamanan seperti autentikasi dan enkripsi TLS untuk menjaga keamanan komunikasi.[15].

Gambar 2.12 Mosquitto

2.13 Python

Python merupakan bahasa pemrograman tingkat tin ggi yang yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991 Python juga merupakan bahasa yanng sangat populer belakangan ini. Selain itu python juga merupakan bahasa pemrograman yang multi fungsi contohnya python dapat digunakan untuk Machine Learning dan Deep Larning. Python dipilih sebagai penelitian karena python memiliki penulisan sintaksis yang mudah selain itu python juga memiliki library yang lengkap dan memiliki dukungan komunitas yang kuat karena python bersifat open source. Untuk menuliskan source code python anda dapat menggunakan IDE seperti vs code, sublime text, PyCharm atau anda juga dapat menggunakan IDE online seperti Jupyter notebook dan google colab[16].

Gambar 2.13 Python