

BAB 2

LANDASAN TEORI

2.1 Tempat Penelitian

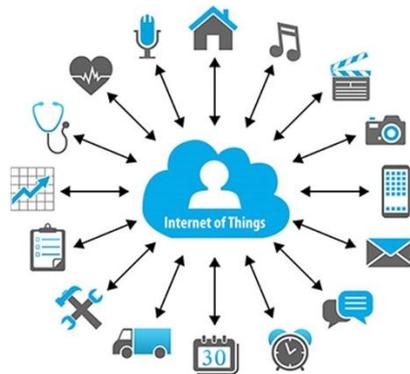
PT. OTTO Pharmaceutical Industries adalah bagian dari Mensa Grup, itu adalah perusahaan manufaktur farmasi memproduksi berbagai obat-obatan yang berkualitas dengan tujuan untuk melayani rakyat secara nasional. Otto didirikan pada 8 April 1963 di Bandung, Jawa Barat. Pada tahun 1981, kepemilikan perusahaan diambil alih oleh Mensa Group, selama waktu manajemen re-organisasi terjadi termasuk perbaikan dalam sistem akuntansi perusahaan dan penambahan fasilitas produksi di dalam pabrik. Sejak itu Otto terus berkembang mengembangkan kedua pabrik dan sumber daya manusia perusahaan. Sejalan dengan keyakinan yang diperoleh oleh dokter dan masyarakat terhadap kualitas produk Otto, tersedia Otto kesempatan untuk meningkatkan penjualan dan pangsa pasar.

PT. Otto Pharmaceutical Industries dalam proses produksinya lebih banyak memproduksi obat-obat ethical daripada OTC. Dengan pertimbangan produk-produk tersebut lebih banyak dipakai dan diminati daripada produk OTC. Jenis-jenis produk yang dihasilkan antara lain: untuk produksi non beta laktam yaitu bentuk sediaan sirup, tablet, kaplet, kapsul, suspensi, tablet salut, injeksi, *cream*, infuse dan sirup kering. Untuk produk β laktam yaitu sirup kering, tablet dan kaplet.

2.2 *Internet Of Things (IOT)*

Internet of Things (IOT) merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, *remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan *global* melalui sensor yang tertanam dan selalu aktif [4]. Pada dasarnya, *Internet of Things* mengacu pada benda yang dapat diidentifikasi.

mulai terkenal melalui *Auto-ID Center* di MIT. Gambar 2.1 menunjukkan konsep *internet of things*.



Sumber Gambar: <https://pixabay.com/images/search/internet%20of%20things/>

Gambar 2.1 Konsep *Internet of Things*

Metode yang digunakan oleh Internet of Things adalah nirkabel atau pengendalian secara otomatis tanpa mengenal jarak [4]. Pengimplementasian Internet of Things sendiri biasanya selalu mengikuti keinginan developer dalam mengembangkan sebuah aplikasi yang ia ciptakan, apabila aplikasinya itu diciptakan guna membantu monitoring sebuah ruangan maka pengimplementasian Internet of Things itu sendiri harus mengikuti alur diagram pemrograman mengenai sensor dalam sebuah rumah, berapa jauh jarak agar ruangan dapat dikontrol, dan kecepatan jaringan internet yang digunakan. Perkembangan teknologi jaringan dan Internet seperti hadirnya IPv6, 4G, dan Wimax, dapat membantu pengimplementasian Internet of Things menjadi lebih optimal, dan memungkinkan jarak yang dapat di lewati menjadi semakin jauh, sehingga semakin memudahkan kita dalam mengontrol sesuatu.

2.3 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah sebuah Bahasa pemodelan visual yang dirancang khusus untuk pengembangan dan analisis sistem berorientasi objek

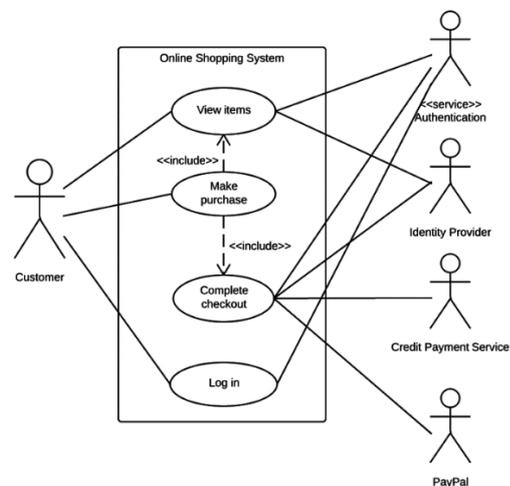
dan desain. UML pertama kali dikembangkan oleh Grady Booch, Jim Rumbaugh, dan Ivars Jacobson pada pertengahan tahun 1990 [5].

2.3.1 Diagram UML

UML menyediakan 4 macam diagram untuk memodelkan aplikasi perangkat lunak berorientasi objek. Yaitu:

1. Use Case Diagram

Use case diagram adalah sebuah diagram yang menunjukkan hubungan antara *actors* dan *use cases*. Digunakan untuk analisis dan desain sebuah sistem. Gambar 2.7 menunjukkan *Use Case Diagram* dalam UML [5].

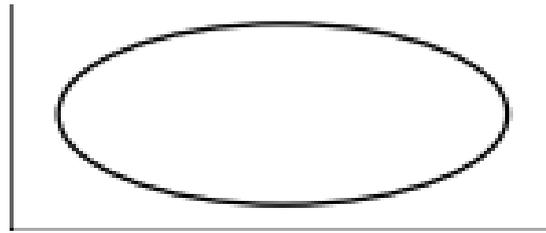


Gambar 2.2 Use Case Diagram

Berikut ini adalah bagian dari sebuah use case diagram :

a. Use Case

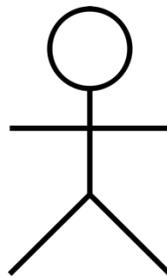
Use cases menjelaskan tentang tindakan/aksi yang dilakukan oleh *actors*. *Use case* digambarkan dalam bentuk elips yang *horizontal*. Gambar 2.3 menunjukkan bentuk *Use Case* dalam UML [5].



Gambar 2.3 Use Case

b. Actors

Actors adalah seorang peran yang berinteraksi dengan sistem. *Actors* meliputi baik manusia maupun organisasi yang saling bertukar informasi. Gambar 2.4 menunjukkan bentuk *actors* dalam UML.



Gambar 2.4 Actors

c. Relationship

Relationship adalah hubungan antara *use cases* dengan *actors*. *Relationship* dalam *use case* diagram meliputi:

1. Asosiasi antara *actor* dan *use case*.

Hubungan antara *actor* dan *use case* yang terjadi karena adanya interaksi antara kedua belah pihak. Asosiasi tipe ini menggunakan garis lurus dari *actor* menuju *use case* baik dengan menggunakan mata panah terbuka ataupun tidak [5].

2. Asosiasi antara 2 *use case*

Hubungan antara *use case* yang satu dan *use case* lainnya yang terjadi karena adanya interaksi antara kedua belah pihak. Asosiasi tipe

ini menggunakan garis putus-putus/garis lurus dengan mata panah terbuka di ujungnya [5].

3. Generalisasi antara 2 actor

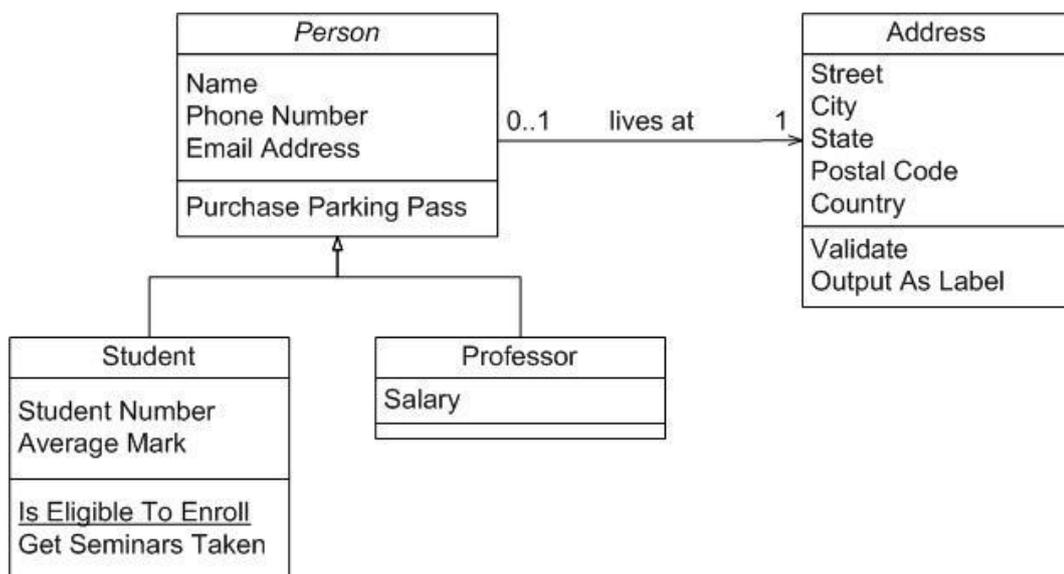
Hubungan *inheritance* (pewarisan) yang melibatkan actor yang satu (*the child*) dengan actor lainnya (*the parent*). Generalisasi tipe ini menggunakan garis lurus dengan mata panah tertutup di ujungnya [5].

4. Generalisasi antara 2 use case.

Hubungan *inheritance* (pewarisan) yang melibatkan use case yang satu (*the child*) dengan use case lainnya (*the parent*). Generalisasi tipe ini menggunakan garis lurus dengan mata panah tertutup di ujungnya [5].

a. Class Diagram

Class diagram adalah sebuah diagram yang menunjukkan hubungan antar *class* yang didalamnya terdapat atribut dan fungsi dari suatu objek. Gambar 2.5 menunjukkan *Class Diagram* dalam UML.



Gambar 2.5 Class Diagram

Class diagram mempunyai 3 relasi dalam penggunaanya, yaitu:

1 *Assosiation*

Assosiation adalah sebuah hubungan yang menunjukkan adanya interaksi antar *class*. Hubungan ini dapat ditunjukkan dengan garis dengan mata panah terbuka di ujungnya yang mengindikasikan adanya aliran pesan dalam satu arah [5].

2 *Generalization*

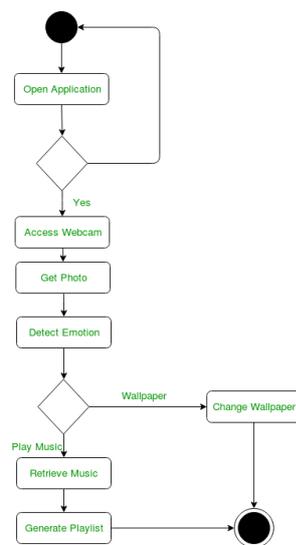
Generalization adalah sebuah hubungan antar *class* yang bersifat dari khusus ke umum [5].

3 *Constraint*

Constraint adalah sebuah hubungan yang digunakan dalam sistem untuk memberi batasan pada sistem sehingga didapat aspek yang tidak fungsional [5].

b. *Activity Diagram*

Activity diagram adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu sistem [5]. Gambar 2.6 menunjukkan *Activity Diagram* dalam UML.



Gambar 2.6 *Activity Diagram*

Berikut ini merupakan komponen dalam activity diagram, yaitu:

1 *Activity node*

Activity node menggambarkan bentuk notasi dari beberapa proses yang beroperasi dalam kontrol dan nilai data [5].

2 *Activity edge*

Activity edge menggambarkan bentuk *edge* yang menghubungkan aliran aksi secara langsung, dimana menghubungkan *input* dan *output* dari aksi tersebut [5].

3 *Initial state*

Bentuk lingkaran berisi penuh melambangkan awal dari suatu proses.

4 *Decision*

Bentuk wajib dengan suatu *flow* yang masuk beserta dua atau lebih *activity node* yang keluar. *Activity node* yang keluar ditandai untuk mengindikasikan beberapa kondisi [5].

5 *Fork*

Satu bar hitam dengan satu *activity node* yang masuk beserta dua atau lebih *activity node* yang keluar [5].

6 *Join*

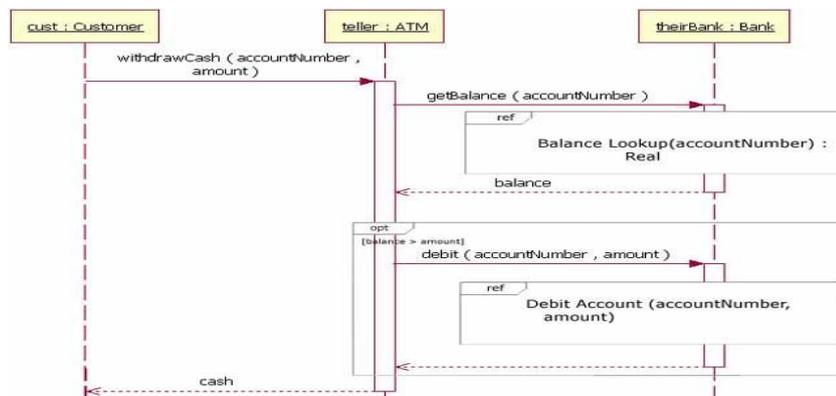
Satu bar hitam dengan dua atau lebih *activity node* yang masuk beserta satu *activity node* yang keluar, tercatat pada akhir dari proses secara bersamaan. Semua *actions* yang menuju *join* harus lengkap sebelum proses dapat berlanjut.

7 *Final state*

Bentuk lingkaran berisi penuh yang berada di dalam lingkaran kosong, menunjukkan akhir dari suatu proses.

2. Sequence Diagram

Sequence diagram adalah sebuah diagram yang menggambarkan kolaborasi dari objek-objek yang saling berinteraksi antar elemen dari suatu class. Gambar 2.7 menunjukkan *Sequence Diagram* dalam UML.



Gambar 2.7 Sequence Diagram

Berikut ini merupakan komponen dalam *sequence diagram*:

a. *Activations*

Activations menjelaskan tentang eksekusi dari fungsi yang dimiliki oleh suatu objek.

b. *Actor*

Actor menjelaskan tentang peran yang melakukan serangkaian aksi dalam suatu proses.

c. *Collaboration boundary*

Collaboration boundary menjelaskan tentang tempat untuk lingkungan percobaan dan digunakan untuk memonitor objek.

d. *Parallel vertical lines*

Parallel vertical lines menjelaskan tentang suatu garis proses yang menunjuk pada suatu *state*.

e. *Processes*

Processes menjelaskan tentang tindakan/aksi yang dilakukan oleh aktor dalam suatu waktu.

f. *Window*

Window menjelaskan tentang halaman yang sedang ditampilkan dalam suatu proses.

g. *Loop*

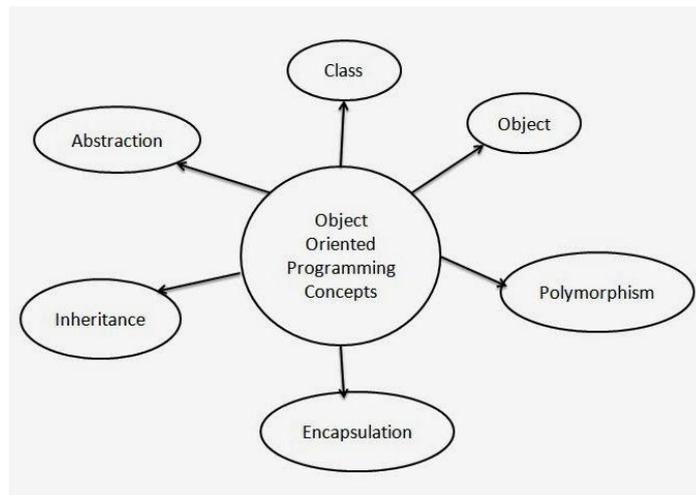
Loop menjelaskan tentang model logika yang berpotensi untuk diulang beberapa kali.

2.4 Object Oriented Programming (OOP)

Object Oriented Programming (OOP) merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedural. *Object Oriented Programming* (OOP) yang berarti pula Pemrograman Berorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1972.

Pemrograman Berorientasi Objek (*Object Oriented Programming/OOP*) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam *class-class* atau *object-object*. Setiap *object* dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya.

Masing-masing objek harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan objek yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman keyboard untuk mengerjakan banyak perintah atau *statement*. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh: python, java, php, perl, c#, cobol dan lainnya [6]. Gambar 2.8 menunjukkan konsep dasar OOP.



Gambar 2.8 Konsep Dasar OOP

Berikut ini adalah konsep-konsep dalam pemrograman berorientasi objek:

1. *Class*

Kelas (*Class*) merupakan penggambaran satu set objek yang memiliki atribut yang sama. Kelas mirip dengan tipe data ada pemrograman non objek, akan tetapi lebih komprehensif karena terdapat struktur sekaligus karakteristiknya. Kelas baru dapat dibentuk lebih spesifik dari kelas ada umumnya. kelas merupakan jantung dalam pemrograman berorientasi objek.

2. *Object*

Objek merupakan teknik dalam menyelesaikan masalah yang kerap muncul dalam pengembangan perangkat lunak. Teknik ini merupakan teknik yang efektif dalam menemukan cara yang tepat dalam membangun sistem dan menjadi metode yang paling banyak dipakai oleh para pengembang perangkat lunak. Orientasi objek merupakan teknik pemodelan sistem riil yang berbasis objek.

3. *Abstraction*

Kemampuan sebuah program untuk melewati aspek informasi yang diolah adalah kemampuan untuk fokus pada inti permasalahan. Setiap objek dalam sistem melayani berbagai model dari pelaku abstrak yang dapat melakukan kerja, laporan dan perubahan serta berkomunikasi dengan objek lain dalam sistem, tanpa harus menampakkan kelebihan diterapkan.

4. Encapsulation

Encapsulation adalah proses memastikan pengguna sebuah objek tidak dapat menggantikan keadaan dari sebuah objek dengan cara yang tidak sesuai prosedur. Artinya, hanya metode yang terdapat dalam objek tersebut yang diberi izin untuk mengakses keadaan yang diinginkan. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berintegrasi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut

5. Polimorfism

Polimorfism merupakan suatu fungsionalitas yang diimplikasikan dengan berbagai cara yang berbeda. Pada program berorientasi objek, pembuat program dapat memiliki berbagai implementasi untuk sebagian fungsi tertentu.

6. Inheritance

Inheritance mempunyai fungsi mengatur polimorfisme dan enkapsulasi dengan mengizinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Objek-objek ini dapat membagi dan memperluas perilaku mereka tanpa mengimplementasikan perilaku tersebut.

2.5 MySQL

MySQL adalah perangkat lunak database server atau sebut saja *Database Smart*. *Database* ini semakin lama semakin populer. Dengan menggunakan database ini, data semakin aman dan berdaya guna. Database ini juga banyak dipakai pada web database sehingga data semakin terintegrasi antara database dekstop dengan database web. Untuk menggunakan database *MySQL* harus menginstalasinya dahulu ke komputer. Sebagai *server database* dengan konsep *database modern*. Gambar 2.9 menunjukkan logo *MySQL*.



Sumber Gambar: <https://www.mysql.com/about/legal/logos.html>

Gambar 2.9 Logo MySQL

MySQL memiliki keistimewaan. Keistimewaan dimiliki MySQL sebagai berikut:

1. *Portability*

Database *MySQL* berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Mac OS X Server*, *Solaris*, *Amiga*, *HP-Unix*, dan lain-lain.

2. *Open Source*

MySQL merupakan *database open source* (gratis), di bawah lisensi *GPL* sehingga dapat memperoleh dan menggunakannya secara cuma-cuma tanpa membayar sepersen pun.

3. *Multuser*

MySQL merupakan *database* yang dapat digunakan untuk menangani beberapa *user* dalam waktu bersamaan tanpa mengalami masalah. Dan memungkinkan sebuah *database server MySQL* dapat diakses *client* secara bersamaan pula.

4. *Performace Tuning*

MySQL mempunyai kecepatan yang cukup baik dalam menangani *query-query* sederhana, serta mampu memproses lebih banyak *SQL* per satuan waktu.

5. *Column Type*

Database MySQL didukung dengan tipe data yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, data, time, datetime, timestamp, year, set* serta *enum*.

6. *Command dan Functions*

MySQL server memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

7. *Security*

Sistem *Security* pada *MySQL* mempunyai beberapa lapisan sekuritas seperti tingkatan *subnetmask, hostname*, dan izin akses *user* dengan sistem perizinan yang mendetil serta *password* terenkripsi.

2.6 Python

Python diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisi kesukaan Guido *Monty Python's Flying Circus*. Van Rossum mengembangkan *Python* sebagai hobi, kemudian *Python* menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintaks intuitif dan memiliki pustaka yang luas [16]. *Python* saat ini dikembangkan dan dikelola oleh tim relawan yang besar dan tersedia secara gratis dari Python Software Foundation. Gambar 2.10 menunjukkan logo *python*.



Sumber Gambar: <https://www.python.org/community/logos/>

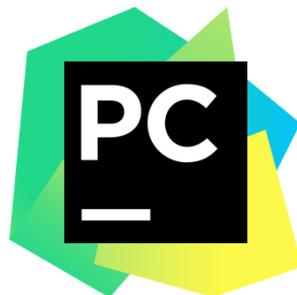
Gambar 2.10 Logo Python

2.7 *Virtual network computing (VNC)*

Virtual network computing (VNC) adalah software remote-control yang memungkinkan untuk mengontrol komputer lain melalui koneksi network. Pencetan keyboard dan mouse click dikirimkan dari satu komputer ke komputer lainnya sehingga seseorang dapat mengelola sebuah dekstop, server dan alat yang terhubung jaringan tanpa harus di lokasi yang sama [7]. VNC bekerja pada model *client/server*. Sebuah VNC Viewer (atau *client*) diinstall pada komputer lokal dan dihubungkan dengan server yang harus diinstall di komputer remote. Server mengirim duplikasi dari *display* komputer *remote* ke *viewer (client)*. Server juga menerjemahkan *command* dari *viewer* dan menerapkannya pada komputer *remote*. VNC adalah *platform independent* dan kompatibel dengan *operating system* apapun. Komputer harus berada di jaringan TCP/IP dan memiliki port yang terbuka untuk *traffic* dari IP *address* suatu alat yang akan mengontrol. VNC dikembangkan di AT&T Laboratories. *Source code* VNC aslinya adalah *open source* dibawah GNU General Public License dan jenis yang lainnya juga ada secara komersial [7].

2.8 PyCharm

PyCharm adalah *integrated development environment (IDE)* yang digunakan dalam pemrograman komputer, khusus untuk bahasa *Python*. Ini dikembangkan oleh perusahaan *JetBrains* *Ceko*. Ini menyediakan analisis kode, debugger grafis, unit tester terintegrasi, VCSes, dan dukungan pengembangan web dengan *flask*. Gambar 2.11 menunjukkan logo *PyCharm*.



Sumber Gambar : <https://www.jetbrains.com/help/pycharm-edu/viewing-images.html>

Gambar 2.11 Logo PyCharm

Fitur-fitur yang tersedia dalam *pycharm* adalah sebagai berikut:

1. *Coding assistance, analysis, code completion, error highlighting, linter integration, dan quick fixes.*
2. Navigasi proyek dan kode: tampilan proyek khusus, tampilan struktur file, dan lompatan cepat antara file, kelas, metode, dan penggunaan.
3. *Python refactoring*: termasuk mengganti nama, mengekstrak metode, memperkenalkan variabel, memperkenalkan konstan, *pull up, push down* dan lainnya.
4. Dukungan untuk *web framework*: *Django, web2py, dan Flask.*
5. *Integrated Python debugger.*

2.9 PHP

PHP adalah satu Bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam web server oleh interpreter PHP, dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke web browser. Karena pemrosesan program PHP dilakukan di lingkungan web server, PHP dikatakan sebagai sisi server (*server-side*). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “*View Source*” pada web browser yang mereka gunakan. Selain menggunakan PHP, aplikasi web juga dapat dibangun dengan Java (JSP – JavaServer Pages dan Servlet), Perl, Python, Ruby, maupun ASP (Active Server Pages).



Sumber Gambar: <https://www.php.net/download-logos.php>

Gambar 2.12 Logo PHP

Meskipun PHP 7 dapat digunakan untuk membuat aplikasi CLI (*Command Line Interview*) dan juga aplikasi *desktop* (seperti Perl, Python, dan Ruby), namun pada umumnya orang menggunakan PHP untuk tujuan pembuatan aplikasi *web*. PHP adalah Bahasa yang dirancang untuk mudah diletakkan di dalam kode HTML. Banyak dijumpai kode PHP yang menyatu dengan kode HTML. Kode PHP diawali dengan tag `<?php` dan diakhiri dengan tag `?>`.

2.10 Access Point

Access Point adalah sebuah perangkat jaringan yang berisi sebuah *transceiver* dan antena untuk mentransmisikan dan menerima sinyal ke dan dari *clients remote*. Dengan *access points* (AP) *clients wireless* bisa dengan cepat dan mudah untuk terhubung kepada jaringan LAN kabel secara *wireless*. Dengan kata lain *access point* merupakan sebuah alat yang digunakan untuk menghubungkan alat-alat dalam suatu jaringan, dari dan ke jaringan *wireless* [8].

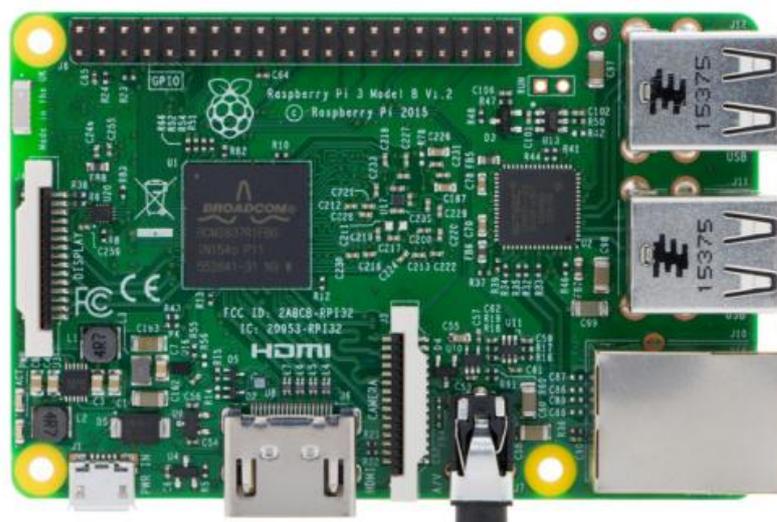
Secara garis besar, *access point* berfungsi sebagai pengatur lalu lintas data, sehingga memungkinkan banyak *client* dapat saling terhubung melalui jaringan (*network*). Sedangkan jika diperinci lebih jelas lagi fungsi *access point* adalah sebagai berikut:

1. Mengatur supaya *access point* dapat berfungsi sebagai *DHCP server*.
2. Menjalankan fitur *Wired Equivalent Privacy* (WEP) dan *Wi-Fi Protected Access* (WPA) untuk keamanan jaringan.
3. Mengatur akses berdasarkan *MAC address device* pengakses. Sebagai *Hub/Switch* yang bertindak untuk menghubungkan jaringan lokal dengan jaringan *wireless/nirkabel*.

2.11 Raspberry Pi 3 Model B+

Raspberry Pi 3 Model B+ merupakan generasi ketiga dari keluarga *raspberry pi*. *Raspberry pi 3* memiliki RAM 1GB dan grafis *Broadcom VideoCore IV* pada frekuensi *clock* yang lebih tinggi dari sebelumnya yang berjalan pada 250MHz. *Raspberry Pi 3* juga memiliki 4 *USB port*, 40 *pin GPIO*, *Full HDMI port*, *Port*

Ethernet, Combined 3.5mm audio jack and composite video, Camera interface (CSI), Display interface (DSI), slot kartu Micro SD (Sistem tekan-tarik, berbeda dari yang sebelumnya ditekan-tekan), dan VideoCore IV 3D graphics core. Gambar 2.13 adalah bentuk raspberry pi 3 model b [8].



Sumber Gambar: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Gambar 2.13 Raspberry Pi 3 Model B+

2.11.1 GPIO Raspberry Pi 3 Model B+

Raspberry pi 3 model B+ terdapat 40 buah pin. Pin pada raspberry pi 3 model B terdiri dari beberapa bagian yaitu bagian VCC, GND, dan GPIO (General Purpose Input/Output) terdapat 3 pin VCC dan 8 pin GND. Pin GPIO mulai dari GPIO 2 hingga GPIO 27, pada pin GPIO terdapat fungsi lain yang dapat dilihat pada Gambar 2.14.

Raspberry Pi 3 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power	Red	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	Blue	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Blue	Ground	06
07	GPIO04 (GPIO_GCLK)	Green	(TXD0) GPIO14	08
09	Ground	Black	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	Green	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Green	Ground	14
15	GPIO22 (GPIO_GEN3)	Green	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	Red	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Purple	Ground	20
21	GPIO09 (SPI_MISO)	Purple	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	Purple	(SPI_CE0_N) GPIO08	24
25	Ground	Black	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	Yellow	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Green	Ground	30
31	GPIO06	Green	GPIO12	32
33	GPIO13	Green	Ground	34
35	GPIO19	Green	GPIO16	36
37	GPIO26	Green	GPIO20	38
39	Ground	Black	GPIO21	40

Rev. 2
29/02/2016

www.element14.com/RaspberryPi

Sumber Gambar: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Gambar 2.14 Pin GPIO Raspberry Pi 3 Model B+

2.12 Sensor

Sensor adalah suatu perangkat yang mendeteksi perubahan energi yang berada di alam seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya. Di dalam sebuah sensor terdapat transduser yang berfungsi untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi besaran listrik berupa tegangan, resistansi dan arus listrik [6].

2.12.1 Sensor Suhu DS18B20 Waterproof

Sensor suhu DS18B20 Waterproof adalah versi sensor DS18B20 yang kedap air. Sensor ini dapat mengukur sesuatu yang berjarak jauh atau dalam kondisi yang basah. Sensornya dapat digunakan hingga 125°C, kabel dilapisi PVC dan dianjurkan untuk digunakan dibawah 100°C. DS18B20 menyediakan pembacaan suhu dari 9 sampai 12-bit (dapat dikonfigurasi) melalui antarmuka melalui 1 kawat, sehingga hanya satu kabel (dan arde) yang perlu dihubungkan dari mikroprosesor pusat. Dapat digunakan dengan sistem 3.0-5.5V [6].

Dalam penelitian ini, sensor suhu DS18B20 Waterproof digunakan untuk mengukur suhu air di dalam kolam. Suhu air diukur karena dapat mempengaruhi pertumbuhan benih ikan koi karena saat suhu air di dalam kolam tidak stabil maka pertumbuhan benih ikan koi pun menjadi tidak optimal.



Sumber Gambar: <https://randomnerdtutorials.com/guide-for-ds18b20-temperature-sensor-with-arduino/>

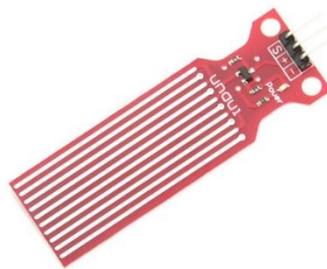
Gambar 2.15 Sensor Suhu DS18B20 Waterproof

Spesifikasi:

- a. Dapat digunakan dengan daya 3.0V hingga 5.5V
- b. Akurasi $\pm 5^{\circ}\text{C}$ dari -10°C hingga $+85^{\circ}\text{C}$
- c. Kisaran suhu yang dapat digunakan : -55°C hingga 125°C (-67°F hingga $+257^{\circ}\text{F}$)
- d. 9 hingga 12 bit resolusi yang dapat dipilih
- e. Menggunakan 1 kabel antarmuka- hanya memerlukan satu digital pin untuk komunikasi
- f. 64 bit ID unik dalam satu chip
- g. Banyak sensor dapat berbagi satu pin
- h. Sistem alarm batas suhu
- i. Waktu kueri kurang dari 750ms
- j. 3 antarmuka kabel :

2.12.2 Sensor Ketinggian air

Air Sensor sensor ketinggian air adalah yang mudah digunakan hemat biaya tinggi/drop pengakuan sensor yang diperoleh dengan memiliki serangkaian kabel paralel terkena jejak diukur tetesan/volume air untuk dengan cepat tingkat air. Mudah untuk menyelesaikan air ke sinyal analog konversi dan output analog nilai-nilai dapat langsung membaca Dewan Pengembangan untuk mencapai tingkat alarm efek.



Sumber Gambar : <https://www.cronyos.com/cara-mengukur-ketinggian-air-menggunakan-water-level-sensor-arduino/>

Gambar 2.16 Sensor Ketinggian air

2.12.3 Relay

Relay adalah saklar yang dioperasikan secara elektrik. *Relay* dipakai ketika sinyal berdaya rendah digunakan untuk mengontrol sebuah rangkaian (isolasi elektrik penuh terjadi antara rangkaian pengontrol dan rangkaian yang dikontrol) atau ketika beberapa sirkuit harus dikontrol oleh satu sinyal. Relay pada mulanya digunakan pada sirkuit telegram jarak jauh, mengulangi sinyal yang datang dari suatu sirkuit dan mentransmisikan kembali sinyal tersebut ke sirkuit yang lain. Relay digunakan secara luas dalam *switching* telepon dan juga pada komputer mula-mula untuk melakukan operasi logis [9]. Gambar 2.17 merupakan bentuk relay.



Sumber Gambar :

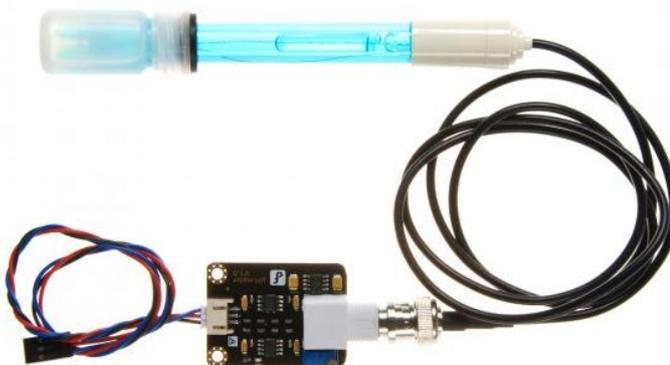
https://wiki.dfrobot.com/Gravity__Digital_Relay_Module__Arduino_&_Raspberr_y_Pi_Compatible__SKU__DFR0473

Gambar 2.17 Relay

2.13.4 Sensor pH Air

Sensor pH air adalah sensor yang memiliki konektor BNC, antarmuka PH2.0 dan LED yang berfungsi sebagai indikator daya. Untuk menggunakannya, cukup sambungkan sensor pH dengan konektor BNC, dan sambungkan antarmuka PH2.0 ke port input analog dari setiap pengontrol Arduino. Sensor ini dapat digunakan untuk pengujian kualitas air dan *aquaculture* [13].

Dalam penelitian ini, sensor pH air digunakan untuk mengukur keasamaan dan kebasaaan. PH adalah salah satu faktor penting kualitas air yang berpengaruh pada kesehatan ikan dan kelangsungan hidup ikan di kolam.



Sumber Gambar: https://wiki.dfrobot.com/PH_meter_SKU__SEN0161_

Gambar 2.18 Sensor pH Air

Spesifikasi:

- a. Modul Power :5.00V
- b. Ukuran Modul : 43mm x 32mm
- c. Rentang Pengukuran 0 – 14 pH
- d. Mengukur suhu : 0-60 °C
- e. Akurasi : $\pm 0,1$ pH (25°C)
- f. Waktu Respon : Kurang dari 1 menit
- g. Sensor pH dengan Konektor BNC
- h. Antarmuka PH2.0
- i. Potensiometer penyesuaian penguatan
- j. LED Indikator Daya

2.13.5 Sensor gas Alkohol dan Etanol MQ-3

MQ-3 merupakan modul sensor gas yang cocok digunakan untuk proses penentuan kadar alkohol yang terdapat dalam udara. Desain *hardware* modul telah dirancang dengan tujuan untuk memudahkan proses penggunaan serta implementasi sensor **MQ-3**, yaitu sensor untuk pendeteksian kadar alkohol dengan *range* 0.05 mg/L - 10 mg/L. Modul ini cocok digunakan pada aplikasi seperti *breathalyzer*, sistem keamanan berkendara, alarm, ataupun aplikasi sejenis lainnya. MQ-3 didesain menggunakan komponen SMD berkualitas agar dapat digunakan pada aplikasi yang memerlukan performa handal serta kebutuhan *space* yang minimal. Pada sisi antarmuka, tersedia dua buah pilihan untuk *user* yaitu UART TTL dengan baud rate 38400 bps atau I2C yang memungkinkan modul untuk di-*cascade* hingga 8 buah.



Sumber Gambar :

https://wiki.dfrobot.com/Analog_Gas_Sensor_MQ3__SKU_SEN0128

Gambar 2.19 Sensor MQ-3

Spesifikasi:

- Tegangan kerja: 5 VDC.
- Target gas: alkohol.
- *Range* deteksi: 0.05 mg/L - 10 mg/L.
- Antarmuka :
 - UART TTL: 38400 bps, 8-bit data, 1-bit stop, *no parity, no flow control*.
 - I2C: dapat di-*cascade* hingga 8 buah modul dalam satu jalur komunikasi.
- Menggunakan ADC 10-bit untuk konversi data analog dari sensor.
- Memiliki *output* berupa data digital dengan nilai 0 - 1023 (hasil konversi ADC).
- Terdapat 1 buah *variable resistor* untuk pengaturan nilai *threshold* secara manual.
- Disediakan beberapa *jumper* untuk konfigurasi *pull-up* I2C, resistor beban, serta *variable resistor threshold*.
- Memiliki fitur kendali *on/off* dengan 2 mode kerja pilihan yaitu *hysteresis* dan *window*.
- Pin I/O yang kompatibel dengan *level* tegangan TTL dan CMOS.
- Memiliki 2 buah LED sebagai indikator.

2.13.6 Sensor gas Amonia MQ-135

Sebuah modul sensor yang dapat digunakan untuk menentukan kadar konsentrasi gas-gas berbahaya dalam udara. Modul ini berbasis sensor MQ-135, yaitu sensor yang dapat mendeteksi gas amonia, bensol, alkohol, serta gas berbahaya lainnya. Idesain menggunakan komponen SMD berkualitas agar dapat digunakan pada aplikasi yang memerlukan performa handal serta kebutuhan *space* yang minimal. Pada sisi antarmuka, tersedia dua buah pilihan untuk *user* yaitu UART TTL dengan baud rate 38400 bps atau I2C yang memungkinkan modul untuk di-*cascade* hingga 8 buah



Sumber Gambar:

https://wiki.dfrobot.com/Analog_Gas_Sensor_MQ135__SKU_SEN0128

Gambar 2.20 Sensor MQ-135

Spesifikasi:

- Tegangan kerja: 5 VDC.
- Target gas: amonia (NH₃), nitrogen oksida (NO_x), alkohol, bensol, asap, karbon dioksida (CO₂), dll.
- *Range* deteksi: 10 ppm - 300 ppm amonia, 10 ppm - 1000 ppm bensol, 10 ppm - 300 ppm alkohol.
- Antarmuka :
 - UART TTL: 38400 bps, 8-bit data, 1-bit stop, *no parity, no flow control*.
 - I2C: dapat di-*cascade* hingga 8 buah modul dalam satu jalur komunikasi.
- Menggunakan ADC 10-bit untuk konversi data analog dari sensor.
- Memiliki *output* berupa data digital dengan nilai 0 - 1023 (hasil konversi ADC).

2.13.7 Arduino Uno



Sumber Gambar: <https://www.arduino.cc/en/Main/ArduinoBoardUnoSMD>

Gambar 2.21 Arduino uno

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (*data sheet*). Memiliki 14 pin *input* dari *output digital* dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang ke adaptor-DC atau baterai untuk menjalankannya. Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial [6].

Nama “Uno” berarti *satu* dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian board USB Arduino, dan sebagai model referensi untuk platform Arduino, untuk perbandingan dengan versi sebelumnya, lihat indeks board Arduino.

2.14 Instalasi Pengolahan Limbah Cair (IPAL)

IPAL adalah sebuah struktur teknik dan perangkat peralatan beserta perlengkapannya yang dirancang secara khusus untuk memproses atau mengolah cairan sisa proses, sehingga sisa proses tersebut menjadi layak dibuang ke lingkungan. Cairan sisa proses atau limbah bisa berasal dari proses industri, pabrik, pertanian, dan perkotaan yang tidak lain merupakan hasil limbah rumah tangga. Hasil dari pembuangan tersebut dapat membahayakan manusia maupun lingkungan, oleh karena itu diperlukan proses pengolahan lebih lanjut sebelum dibuang ke saluran pembuangan [9].

Menyaring dan membersihkan cairan yang sudah tercemar baik oleh pencemar organik atau kimia industri menjadi tujuan utama IPAL. Oleh sebab itu, IPAL memiliki urgensi untuk dilakukan. IPAL yang dikelola secara benar pun menjanjikan sejumlah manfaat atau kegunaan.

Ketentuan atau regulasi mengenai Pengolahan limbah diatur dalam Peraturan Pemerintah atau PP. Yakni PP No 19 Tahun 1999 tentang Pengendalian Pencemaran dan/atau Perusakan Laut. Tepatnya pasal 8, pasal 9, pasal 10, pasal 12, dan pasal 13. Dan di Indonesia masih ada sekitar 74% perusahaan kecil yang belum mengolah secara tepat air limbah mereka, Rendahnya pengawasan dari pemerintah menyebabkan perusahaan atau pelaku usaha dengan mudah mengabaikan ketentuan tersebut. Dan sekarang, oleh karena kurangnya pengendalian dari pemerintah dan kesadaran dari para pelaku usaha akan pentingnya pengolahan limbah tersebut menyebabkan banyaknya saluran air dan lingkungan yang terkontaminasi oleh limbah industri. Pencemaran tersebut menyebabkan timbulnya penyakit dari yang ringan hingga berat. Oleh karena diharapkan adanya pengawasan secara intensif yang dilakukan oleh pemerintah terkait akan hal ini [10].

