

BAB II

LANDASAN TEORI

Pada Bab 2 ini menjelaskan teori dalam pemahaman dalam pembuatan suatu alat yang akan dibuat bertujuan mengenali apa saja inti dari teori yang akan dijelaskan berdasarkan pemahaman dalam suatu rancangan yang akan dibuat berdasarkan berisi referensi dari stidi pusakan, referensi buku, web dan informasi pendukung lainnya sebagai data pencapaian yang akan di rancang

2.1 Rancang Bangun

Pengertian Rancang Bangun Merancang sebuah adalah hal yang paling utama dalam membuat sesuatu, baik dalam membuat sistem atau membuat suatu karya dalam bentuk apapun. Karena rancangan akan menjadi fondasi utama dalam menjalankan langkah-langkah pembuatan sebuah karya. Perancangan atau rancang bangun merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dan sebuah sistem ke dalam Bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem di implementasikan. Sedangkan pengertian Bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2009) Sedangkan menurut (Thohari, 2016), rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak untuk kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada

2.2 Sistem informasi

Sistem informasi adalah kumpulan antara subsub sistem yang saling berhubungan yang membentuk suatu komponen yang didalamnya mencakup input-proses-output yang berhubungan dengan pengolahan data menjadi informasi sehingga lebih berguna bagi pengguna . Berdasarkan penjelasan di atas, maka dapat disimpulkan bahwa sistem informasi merupakan kumpulan

dari komponen-komponen yang saling berhubungan dan melakukan pengolahan data menjadi informasi sehingga informasi tersebut dapat digunakan untuk tujuan tertentu. (Kadir ,2014:9)

2.3 Umum

Menurut Peraturan Pemerintah Nomor 41 Tahun 1999 tentang pengendalian pencemaran udara, definisi pencemaran udara adalah masuknya atau dimasukkannya zat, energi, dan/atau komponen lain kedalam udara ambien oleh kegiatan manusia, sehingga mutu udara turun sampai ke tingkat tertentu yang menyebabkan udara ambien tidak dapat memenuhi fungsinya. Sedangkan menurut Nevers (2011), pencemaran udara adalah kehadiran materi yang tidak diinginkan di udara dalam jumlah tertentu sehingga dapat menghasilkan dampak yang merusak.

Pembangunan fisik kota dan berdirinya pusat-pusat industri disertai dengan melonjaknya produksi kendaraan bermotor, mengakibatkan peningkatan kepadatan lalu lintas dan hasil produksi sampingan, yang merupakan salah satu sumber pencemaran udara. Konsentrasi pencemaran udara di beberapa kota besar dan daerah industri Indonesia menyebabkan adanya gangguan pernafasan, iritasi pada mata dan telinga, serta timbulnya penyakit tertentu. Selain itu juga mengakibatkan gangguan jarak pandang (*visibilitas*) yang sering menimbulkan kecelakaan lalu lintas.

2.4 Sumber dan Jenis Pencemaran Udara

Sumber pencemaran di udara dapat digolongkan menjadi 2 (dua) yaitu kegiatan yang bersifat alami (natural) dan kegiatan akibat aktivitas manusia (antropogenik) yang semakin berkembang dari waktu ke waktu. Contoh sumber alami adalah akibat letusan gunung berapi, kebakaran hutan, dekomposisi biotik, debu, spora tumbuhan, dan lain sebagainya. Sedangkan secara kuantitatif pencemaran antropogenik sering terjadi dan banyak dihasilkan dari aktivitas transportasi, industri, rokok dan persampahan, baik akibat proses dekomposisi ataupun pembakaran, dan rumah tangga. (Soedomo, 2011).

Kualitas udara sangat dipengaruhi oleh besar dan jenis sumber pencemar yang ada seperti dari kegiatan industri, kegiatan transportasi dan lain-lain. Masing-masing sumber pencemar yang berbeda-beda baik jumlah, jenis, dan pengaruhnya bagi kehidupan. Pencemar udara yang terjadi sangat ditentukan oleh kualitas bahan bakar yang digunakan, teknologi serta pengawasan yang dilakukan. Peningkatan jumlah kendaraan bermotor akan meningkatkan pemakaian bahan bakar gas, dan hal itu akan membawa risiko pada penambahan gas beracun di udara terutama CO, HC, SO₂ (Santoso, 2011).

2.5 Pengendalian Pencemaran Udara

Upaya pengendalian pencemaran udara dapat dilakukan melalui 3 hal berikut (Soedomo, 2011) :

Penelitian dan Pemantauan

Pengendalian pengelolaan pencemaran udara perlu mempertimbangkan keserasian antara faktor-faktor sumber emisi, pengaruh/dampak, kondisi sosial, ekonomi dan politik serta melakukan pengukuran lapangan sesuai dengan kondisi.

Langkah pertama, dalam pengelolaan pencemaran udara adalah dengan melakukan pengkajian/identifikasi mengenal macam sumber, model dan pola penyebaran serta pengaruh/dampaknya. Model dan pola penyebaran dapat diperkirakan melalui studi mengenai kondisi fisik sumber, kondisi awal kualitas udara setempat, kondisi meteorology dan topografi. Studi pengaruh/dampak pencemaran udara, dilakukan terhadap kesehatan manusia, kehidupan hewan dan tumbuhan, material, estetika dan terhadap kemungkinan adanya perubahan iklim setempat maupun regional. Langkah selanjutnya adalah mengetahui dan mengkomunikasikan tentang pentingnya pengelolaan pencemaran udara dengan mempertimbangkan keadaan sosial lingkungannya, yang berhubungan dengan demografi, kondisi sosial ekonomi, sosial budaya dan psikologi serta imbalanced ekonomi. Juga perlunya dukungan politik baik dari segi hukum, peraturan, kebijakan maupun administrasi untuk melindungi pelaksanaan pemantauan, pengendalian, dan pengawasan.

Peraturan Perundangan

Upaya pengendalian pencemaran lingkungan khususnya udara saat ini masih bersifat sektoral, baik legislatif maupun institusinya, sesuai dengan kewenangan dari instansi/departemen dalam pengelolaan kegiatan yang bernaung di bawahnya. Adapun bentuk peraturan tersebut berupa peraturan, instruksi atau keputusan Menteri untuk tingkat pusat/departemen, sedangkan untuk tingkat daerah berupa peraturan daerah atau keputusan/instruksi Gubernur. Beberapa peraturan tentang upaya pengendalian pencemaran misalnya yang diterapkan untuk sektor industri, sektor pertambangan, dan sektor transportasi.

Teknologi Pengendalian Pencemaran

Upaya penanggulangan pencemaran udara dari segi teknologi pada prinsipnya mencakup dua masalah yaitu pengendalian pada sumbernya (pengendalian pencemaran debu/partikel, pengendalian gas, pengelolaan buangan kendaraan bermotor) dan pengendalian lingkungannya. Suatu teknologi pengendalian pencemaran umumnya terkait dengan peraturan tentang baku mutu pencemaran, sehingga pemilihan alternatif dari bentuk teknologi pengendalian pencemaran tergantung pula dari ketat atau tidaknya peraturan.

Pada umumnya teknologi pengendalian pencemaran akan mengacu kepada pembiayaan, sehingga hal tersebut akan terkait pula dengan keadaan ekonomi suatu Negara. Berdasarkan pengamatan yang telah dilakukan usaha pengendalian pada sumbernya merupakan usaha yang paling berhasil, namun lebih baik lagi apabila diikuti dengan pengendalian lingkungan.

2.6 Program Car Free Day (CFD)

Kanaf (2011) menyatakan bahwa Program *Car Free Day* merupakan salah satu program untuk mengurangi dan mengendalikan pencemaran udara.

Program

Car Free Day pertama kali dilakukan di negara Belanda dan Belgia dalam rangka mengurangi krisis energi pada 25 November 1956 hingga 20 Januari 1957. Pada 19 April 2001 program *Earth Car Free Day* pertama kali diadakan

dan serentak di seluruh penjuru dunia. Lebih dari 300.000 organisasi dan kota di seluruh dunia ikut berpartisipasi dalam kegiatan yang diadakan oleh *The Commons WC/FD program and Earth Day Network*. Pada tanggal 29 September 2009, *World Car Free Day* dirayakan di Washington, D.C. Kegiatan yang dilaksanakan di sana antara lain terdiri dari reparasi kendaraan bermotor gratis, senam yoga dan kegiatan-kegiatan lain yang dilakukan oleh berbagai Lembaga Swadaya Masyarakat (LSM).

Kanaf (2010) menambahkan bahwa Perserikatan Bangsa Bangsa (PBB) yaitu program *Car Free Day* ini merupakan sebuah proyek dunia dalam rangka mengurangi pencemaran udara. Hal ini termuat dalam proposal PBB mengenai *The United Nations Car Free Days Programme*, Di Indonesia sendiri, program *Car Free Day* pertama kali dikenal dengan program Hari Bebas Kendaraan Bermotor (HBKB). Pelaksanaanya pertama kali dilaksanakan di Jakarta pada tanggal 21 September 2004 di sepanjang ruas Jalan Sudirman - Thamrin. Pada hari itu seluruh kendaraan bermotor dilarang melintas di jalan yang telah ditentukan.

2.7 Partikulat Matter 10 μ m (PM10)

Partikulat Matter yang melayang di udara berisikan campuran yang heterogen yaitu padat dan cair yang bercampur di dalam udara, dan terus bervariasi di dalam ukuran dan komposisi kimia. Partikel termasuk pencemar primer karena diemisikan secara langsung ke dalam atmosfer, seperti asap dari mesin diesel, Sedangkan untuk partikel sekunder dihasilkan melalui transformasi psikokimia gas, seperti nitrat dan sulfat dari formasi dari asam nitrat dan Sulfur dioxide(SO₂) (Brook R.D et al,2014).

Berdasarkan kajian baku mutu udara ambien Kementerian Lingkungan Hidup (2011), salah satu jenis pencemar udara yang dihasilkan dari aktivitas transportasi adalah Particulate Matter 10 μ m (PM10). Partikulat didefinisikan sebagai partikel kecil yang berasal dari padatan maupun cairan yang tersuspensi dalam gas (udara). Ukuran partikel sangatlah penting untuk diketahui karena akan mempengaruhi dampak partikel tersebut terhadap

lingkungan dan manusia. PM10 merupakan partikel inhalable karena ukuran diameter dibawah $10\mu\text{m}$ dan

berdasarkan ukuran aerodinamisnya sehingga PM10 akan mudah terhisap masuk ke paru-paru dan bertahan di dalam tubuh dalam waktu yang lama. Salah satu parameter pencemaran udara yang memiliki dampak sangat berbahaya terhadap kesehatan manusia adalah PM10. Hasil penelitian yang dilakukan oleh pakar kesehatan masyarakat menunjukkan bahwa PM10 merupakan zat pemicu timbulnya infeksi saluran pernafasan karena partikel padat PM10 dapat mengendap pada bronki dan alveoli, sehingga PM10 memiliki daya rusak lebih besar bagi kesehatan manusia dibandingkan pencemar udara lain hingga dapat menyebabkan kematian (Fauzi et al. 2013). PM10 dihasilkan dari proses pengereman, abrasi ban dengan jalan, resuspensi debu jalan dan tanah, serta proses lainnya (Malina 2012).

Sifat fisis partikel yang penting adalah ukurannya, yang berkisar antara diameter 0,0002 mikron sampai sekitar 500 mikron. Pada kisaran tersebut partikel mempunyai umur dalam bentuk tersuspensi di udara antara beberapa detik sampai beberapa bulan. Umur partikel tersebut dipengaruhi oleh kecepatan pengendapan yang ditentukan dari ukuran dan densitas partikel serta aliran udara (Ratnani, 2008). Menurut Fardiaz (1992) Partikulat yang berdiameter lebih besar dari 10 mikrometer dihasilkan dari proses-proses mekanis seperti erosi angin, penghancuran dan penyemprotan, dan pelindasan benda-benda oleh kendaraan atau perjalanan kaki. Partikulat yang berukuran diameter diantara 1 – 10 mikrometer biasanya berasal dari tanah, debu dan produk-produk pembakaran dari industri lokal. Partikulat yang mempunyai diameter antara 0,1 – 1 mikrometer merupakan emisi produk-produk pembakaran dan aerosol fotokimia. Partikel yang mempunyai diameter kurang dari 0,1 mikrometer masih belum diidentifikasi.

Tabel 2.7 Kateogri Konsentrasi PM₁₀ ISPU (Kep.Bapedal No.107, 1997)

ISPU	24 jam PM 10 ($\mu\text{g}/\text{m}^3$)	KATEGORI	RENTANG
50	50	BAIK	0-50
100	150	SEDANG	51-100
200	350	TIDAK SEHAT	101-199
300	420	SANGAT TIDAK SEHAT	200-299
400	500	BERBAHAYA	300-LEBIH
500	600		

Berdasarkan Peraturan Pemerintah Republik Indonesia Nomer 41 Tahun 1999 tentang pengendalian pencemaran udara ,pengaruh indeks standar pencemaran udara untuk parameter PM10.

1. Rentang Nilai 0 – 50 pada indeks standar pencemaran udara merupakan kategori baik, yaitu tidak ada efek yang terjadi.
2. Rentang Nilai 51 – 100 pada indeks standar pencemaran udara merupakan kategori sedang, yaitu penurunan pada jarak pandang
3. Rentang Nilai 101 – 119 pada indeks standar pencemaran udara merupakan kategori tidak sehat, yaitu jarak pandang turun dan terjadi pengotoran debu dimana mana.
4. Rentang Nilai 200 – 299 pada indeks standar pencemaran udara merupakan kategori sangat tidak sehat, yaitu meningkatnya sensitivitas pada penderita penyakit asma dan bronchitis.
5. Rentang Nilai 300 – lebih pada indeks standar pencemaran udara merupakan kategori berbahaya, yaitu berbahaya pada semua populasi yang terpapar.

2.8 Gas Karbon Monoksida (CO)

Gas karbon monoksida adalah sejenis gas yang tidak berwarna, tidak berbau, tidak berasa dan tidak mudah larut dalam air, beracun dan berbahaya. Zat gas CO ini akan mengganggu pengikatan oksigen yang membentuk karboksihemoglobin (COHb) pada darah sehingga CO lebih mudah terikat oleh darah dibandingkan dengan oksigen dan gas-gas lainnya Paparan CO diketahui

dapat mempengaruhi kerja jantung (sistem kardiovaskular), sistem saraf pusat, juga janin, dan semua organ tubuh yang peka terhadap kekurangan oksigen (Halmas, 2015).

Menurut Isnaini (2012) dalam skripsinya menjelaskan bahwa CO yang terdapat di atmosfer terbentuk dari salah satu proses berikut :Pembakaran yang tidak lengkap terhadap karbon atau komponen yang mengandung karbon. Reaksi antar karbon dioksida (CO₂) dan komponen yang mengandung karbon pada suhu tinggi. Karbon dioksida (CO₂) dapat terurai kembali menjadi CO dan oksigen (O₂) pada suhu tinggi Selain itu, berbagai proses geofisika dan biogenic diketahui dapat memproduksi gas CO seperti aktivitas vulkanik, emisi gas alami, pancaran listrik dari kilat, serta sumber lainnya. Tetapi kontribusi CO ke atmosfer yang berasal dari sumber alami relatif lebih kecil dibandingkan dari sumber aktivitas manusia seperti transportasi, pembakaran bahan bakar minyak, industri dan sumber lainnya.

Table 2.8 Kategori Konsentrasi CO ISPU(Kep. Bapedal No. 107, 1997)

ISPU	8 JAM CO (ppm)	KATEGORI	RENTANG
50	5	BAIK	0-50
100	10	SEDANG	51-100
200	17	TIDAK SEHAT	101-199
300	34	SANGAT TIDAK SEHAT	200-299
400	46	BERBAHAYA	300-LEBIH
500	57.5		

2.9 Faktor Meteorologi

Faktor Meteorologi mempunyai peran yang sangat utama dalam menentukan kualitas udara di suatu daerah, baik kualitas udara perkotaan, pedesaan maupun alami. Polutan akan mengalami proses pengenceran dikarenakan faktor-faktor meteorologis yang mempengaruhinya, diantaranya arah dan kecepatan angin, suhu udara, stabilitas atmosfer, mixing height, dan turbulensi (Fairuzi 2012).

Dalam sistem pencemaran udara, intensitas emisi pencemar sumber akan masuk ke dalam atmosfer sebagai medium penerima. Atmosfir sendiri

merupakan suatu medium yang sangat dinamik, ditandai dengan kemampuan-kemampuan sebagai berikut (Soedomo, 2001) :Kemampuan atmosfer tersebut sangat ditentukan oleh berbagai faktor meteorologi seperti temperatur, kelembaban, kecepatan dan arah angin, dan curah hujan.

2.10 IoT (Internet of Things)

Dalam bahasa sederhana, konsep IoT dapat digambarkan sebagai terhubungnya suatu objek fisik ke jaringan internet. Objek fisik ini dapat berupa peralatan elektronik yang melakukan sensing atau actuator. Ada banyak pilihan untuk mengimplementasikan peralatan berbasis IoT. Utamanya adalah peralatan berbasis embedded (peralatan tertanam) yang dilengkapi dengan modul jaringan, baik wired (kabel) atau wireless (nirkabel) sehingga peralatan ini dapat berkomunikasi dengan lingkungan luar (Agus Kurniawan, 2016:1-3).



Gambar 2.10 IoT (Internet of Things) (www.enterpriseioforum.com, akses 20-10-2018)

2.11 Perangkat Hardware

Perangkat Hardware yang digunakan dalam penelitian ini yaitu MQ-7 sebagai pendeteksi gas CO, Shinyei PPD42NS sebagai pendeteksi partikel PM10 dan PM2.5, DHT11 sebagai sensor suhu dan kelembaban, Wemos D1 (Arduino) sebagai mikrokontroler, RTC DS 1302 sebagai realtime, LCD dan PC/laptop sebagai data output.

2.11.1 Wemos D1 (Arduino)

Wemos D1 merupakan salah satu modul board yang dapat berfungsi dengan arduino khususnya untuk project yang mengusung konsep IOT. Wemos dapat running standalone tanpa perlu dihubungkan dengan mikrokontroler, berbeda dengan modul wifi lain yang masih membutuhkan mikrokontroler sebagai pengontrol atau otak dari rangkaian tersebut, wemos dapat running stand-alone karena didalamnya sudah terdapat CPU yang dapat memprogram melalui serial port atau via OTA serta transfer program secara wireless. Chipset Wemos memiliki 2 buah chipset yang digunakan sebagai otak kerja antara lain. a. Chipset ESP8266 ESP8266 merupakan sebuah chip yang memiliki fitur Wifi dan mendukung stack TCP/IP. Modul kecil ini memungkinkan sebuah mikrokontroler terhubung kedalam jaringan Wifi dan membuat koneksi TCP/IP hanya dengan menggunakan command yang sederhana. Dengan clock 80 MHz chip ini dibekali dengan 4MB eksternal RAM serta mendukung format IEEE 802.11 b/g/n sehingga tidak menyebabkan gangguan bagi yang lain.

Chipset CH340 CH340 adalah chipset yang mengubah USB serial menjadi serial interface, contohnya adalah aplikasi converter to IrDA atau aplikasi USB converter to Printer. Dalam mode serial interface, CH340 mengirimkan sinyal penghubung yang umum digunakan pada modem. CH340 digunakan untuk mengubah perangkat serial interface umum untuk berhubungan dengan bus USB secara langsung.



Gambar 21.11.1 Wemos D1

2.11.2 MQ-7 Sensor gas CO

Sensor MQ-7 merupakan sensor gas karbon monoksida (CO) buatan Hanwei yang memiliki sensitivitas tinggi dan waktu respon yang cepat. Keluaran yang dihasilkan oleh sensor ini berupa sinyal analog. Sensor ini memerlukan supply tegangan sebesar 5V direct current (DC). Sensor MQ-7 dapat mendeteksi konsentrasi gas CO mulai dari 20 ppm hingga 2000 ppm. Modul sensor ini sudah dilengkapi dengan potensiometer untuk mengatur sensitivitas. Sensor ini terdiri dari tabung keramik Al_2O_3 , Tin Dioksida SnO_2 lapisan sensitif, elektroda pengukuran serta heater yang digabungkan dalam suatu lapisan kerak yang terbuat dari plastik dan stainless steel (Hanwei Electronics, 2013).



Gambar 2.11.2 Sensor MQ-7 (Hanwei Electronics, 2013)

Sensor gas MQ-7 dengan dt-Sense modul seperti pada Gambar 2.4 sudah dilengkapi dengan analog to digital converter (ADC). Melalui ADC, nilai masukan yang berupa sinyal analog akan dikonversi menjadi sinyal desimal sehingga mudah dibaca oleh pengguna. ADC mempunyai resolusi 10 bit dalam modulnya, sehingga data yang dikirimkan ke mikrokontroler ATmega328 AVR sudah dalam bentuk digital. Pada sensor ini terdapat nilai resistensi sensor (R_s) yang dapat berubah bila mendeteksi gas CO dan juga sebuah heater yang digunakan sebagai pembersih sensor dari kontaminasi udara yang berasal dari luar. Apabila terdeteksi gas CO di udara, maka tegangan output pada sensor akan meningkat, sehingga konsentrasi gas akan menurun dan terjadi proses deoksidasi, akibatnya permukaan dari muatan negatif oksigen akan berkurang, ketinggian permukaan sambungan penghalang pun akan ikut terjadi. Karakteristik output sensor bila mendeteksi keberadaan gas CO yaitu output tegangan semakin besar sesuai dengan besarnya kadar CO dalam ppm (Hanwei Electronics, 2013).

Sensor MQ-7 ini memerlukan perlakuan khusus pada tingkat suhu dan kelembaban tertentu. Kondisi lingkungan yang disarankan untuk penggunaan sensor ini yaitu pada suhu antara -25°C hingga 50°C , kelembaban tidak melebihi dari 95%, dan kadar oksigen (O_2) adalah 21% karena konsentrasi oksigen dapat mempengaruhi sensitivitas. Gambar 2.5 menunjukkan bahwa standar pengukuran MQ-7 mempunyai 2 (dua) komponen sensitif yaitu rangkaian pemanasan yang mempunyai kontrol terhadap waktu pemanasan (tegangan yang tinggi dan tegangan yang rendah secara sirkular) dan sensor yang merespon terhadap perubahan pada bagian input (Hanwei Electronics, 2013).

Sensor yang digunakan dalam penelitian untuk mengukur kadar gas CO di udara ambient yaitu sensor gas MQ-7 dengan dt-Sense modul yang sudah dilengkapi dengan fitur analog to digital converter (ADC). Melalui ADC, nilai input yang berupa sinyal analog akan dikonversi menjadi sinyal desimal pada mikrokontroler ATmega328 AVR sehingga mudah oleh pengguna. Jika modul sensor MQ-7 mendeteksi adanya gas CO di

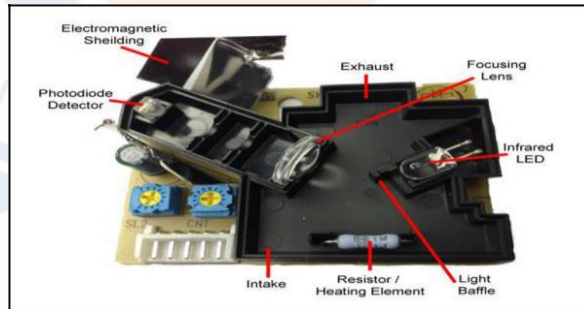
udara ambien, maka modul sensor MQ-7 membutuhkan catu daya 5 volt yang dihubungkan pada kaki VCC dan GND serta pin AO yang dihubungkan ke analog A1 dari mikrokontroler ATmega328 AVR untuk mengirimkan data jumlah konsentrasi gas pencemar CO ke mikrokontroler. Kemudian mikrokontroler mengolah data tersebut untuk ditampilkan pada LCD sebagai output dan tersimpan di dalam micro SD. Nilai konsentrasi CO akan ditampilkan dalam satuan ppm.

dengan Arduino Membuat program dengan *Software* arduino adalah inti dari pembuatan alat ini. Alat ini tidak akan berfungsi dengan jika code source atau coding tidak dituliskan didalam *Software* arduino. Fungsi *Software* ini adalah sebagai pengendali dalam mengendalikan seluruh proses yang ada didalam sistem.

2.11.3 Shinyei PPD42NS

PPD42NS merupakan dust sensor buatan shinyei yang mampu mendeteksi partikel dengan ukuran minimum 1 μm . Dalam pengoperasiannya sensor ini membutuhkan catu daya 5V yang berasal dari mikrokontroler yang dihubungkan pada pin Vcc dan memanfaatkan prinsip photodiode negatif yang memberikan dua digital outputs yang berkisar antara HIGH (tegangan lebih tinggi dari 4.0V) yang dapat menghitung jumlah partikel yang lebih besar dari 1 μm dan LOW (tegangan lebih rendah dari 0,7V) yang dapat menghitung jumlah partikel yang lebih besar dari 2.5 μm yang menunjukkan jumlah partikel di udara (Shinyei, 2010).

Pada sensor ini terdapat dua pin threshold P1 dan P2. Pin P1 digunakan untuk mengukur partikulat berukuran $>1 \mu\text{m}$ sedangkan Pin P2 digunakan untuk mengukur partikulat berukuran $>2.5 \mu\text{m}$. dapat dilihat pada Gambar 2.11.3 merupakan bentuk fisik dari dust sensor Shinyei PPD42NS .



Gambar 2.11.3 Bentuk Fisik Dust Sensor Shinyei PPD42NS
(takingspace.org, akss 2018)

Sensor PPD42NS mengukur tingkat partikulat di udara dengan menghitung waktu Lo Pulse Occupancy time (waktu LPO) pada satuan waktu yang diberikan. Waktu LPO sebanding dengan konsentrasi partikulat (PM). Sensor PPD42NS adalah sensor partikel hamburan cahaya (Weekly et al., 2013; Holstius et al., 2014).

Yang Wang (2015) mengatakan temperatur operasi sensor Shinyei PPD42NS dapat di kontrol dengan meletakkan ice pack atau pemanas disekitar chamber agar temperatur operasi dapat terjaga dan tambahkan exhaust fan pada chamber untuk membuang hawa panas yang ada di dalam chamber. Sensor sinar hamburan cahaya biasanya terdiri dari dioda pemancar inframerah (IRED) yang dapat diserap gas – gas rumah kaca, fototransistor (PT), dan lensa fokus. Saat melewati sensor, PM10 menyebabkan gelombang inframerah yang dipancarkan oleh LED terhalang sehingga terjadi hamburan cahaya yang ditangkap oleh sensor dan menghasilkan sinyal digital. Sinyal yang dihasilkan dari pendeteksian cahaya yang terhambur kemudian diteruskan ke rangkaian filter dan rangkaian amplifikasi (Holstius et al. 2014; Prabakar et al. 2015).

The 2013 US EPA Air Sensor Workshop merekomendasikan 7 (tujuh) parameter untuk diuji setelah menerima perangkat sensor kualitas udara dari pengembang atau pabrik:

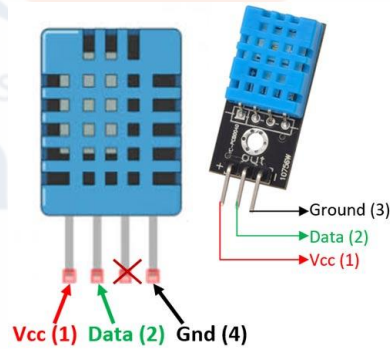
- 1) linearitas respons,
- 2) ketepatan pengukuran,
- 3) batas deteksi,

- 4) resolusi konsentrasi,
- 5) waktu respon,
- 6) gangguan padanan,
- 7) kelembaban relatif (RH) dan pengaruh suhu.

Di antara parameter ini, resolusi konsentrasi direfleksikan dalam presisi pengukuran, seperti yang diperkenalkan dalam ringkasan US EPA Air Sensor Workshop (Long et al., 2014). Untuk sensor partikel yang menggunakan metode hamburan cahaya, waktu transfer cahaya di sensor dapat diabaikan, dan waktu respon sensor terutama ditentukan oleh waktu pengangkutan elektron di perangkat, juga dapat diabaikan. Persamaan gangguan tidak perlu dipertimbangkan, karena, tidak seperti sensor gas, yang sejenisnya dapat menyebabkan respons serupa, sensor partikel dipengaruhi oleh konsentrasi dan sifat partikel saja. Komposisi partikel secara kritis mempengaruhi kinerja sensor hamburan cahaya. Penyebaran cahaya bergantung pada indeks bahan bias, sementara penyerapan cahaya material juga mempengaruhi intensitas cahaya yang diterima oleh fototransistor. .

2.11.4 Sensor DHT11 Suhu dan Kelembapan

Sensor yang digunakan untuk mengetahui Suhu dan kelembaban dalam penelitian ini yaitu sensor DHT11. Sensor DHT11 membutuhkan daya 5 volt yang dihubungkan pada kaki VCC dan GND dan pin data yang dihubungkan ke digital 3 sebagai hasil pembacaan dari sensor ke mikrokontroler) adalah coding (Bahasa pemrograman) untuk menampilkan hasil suhu dan kelembaban pada alat yang di rancang.



Gambar 2.11.4 Sensor DHT11 (Wiring ,2012)

Sensor DHT11 membutuhkan daya 5 volt yang dihubungkan pada kaki VCC dan GND dan pin data yang dihubungkan ke digital 3 sebagai hasil pembacaan dari sensor ke mikrokontroler) adalah coding (bahasa pemrograman) untuk menampilkan hasil suhu dan kelembaban pada alat yang di rancang.

2.11.5 Liquid Crystal Display (LCD) 16x2

LCD ini digunakan sebagai tampilan atau display data kadar PM10 PM2.5, gas CO, suhu dan kelembaban serta penanda waktu dari RTC. Data yang ditampilkan pada LCD merupakan data yang sudah diolah pada mikrokontroler yang sudah berupa angka informasi dan data ini akan ditampilkan secara real time.



Gambar 8.17 LCD 20x4 (Wiring ,2012)\

LCD yang digunakan menggunakan komunikasi I2C, yang dirangkai dengan pin SDA yang dihubungkan ke Analog 4 dan pin SCL yang dihubungkan ke Analog 5 di board Arduino). Sedangkan untuk catu daya membutuhkan tegangan sebesar 5V.

2.12 Software Arduino

Software arduino yang digunakan adalah driver dan IDE, walaupun masih ada beberapa *Software* lain yang sangat berguna selama pengembangan arduino. IDE atau Integrated Development Environment suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan Arduino. IDE arduino merupakan *Software* yang sangat canggih ditulis dengan menggunakan java. IDE arduino terdiri dari:

1. Editor Program Sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa processing
2. Compiler Sebuah modul yang mengubah kode program menjadi kode biner bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa processing.
3. Uploader Sebuah modul yang memuat kode biner dari komputer ke dalam memory di dalam papan Arduin
4. New Sketch Membuka window dan membuat sketch baru.
5. Open Sketch Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino
6. Save Sketch menyimpan sketch, tapi tidak disertai dengan mengkompile.
7. Serial Monitor Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.
8. Keterangan Aplikasi pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal “Compiling” dan “Done Uploading” ketika kita mengcompile dan mengupload sketch ke board Arduino

9. Konsol log Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika ada kesalahan pada sketch yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.



Gambar 3.10. Arduino IDE
(<https://www.arduino.cc/en/main/software>)

2.13 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah salah satu alat bantu yang sangat handal dalam dunia pengembangan sistem berorientasi obyek karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka ke bentuk baku, mudah dipahami lengkap dengan mekanisme efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain (Prabowo Pudjo Widodo Herlawati, 2011).

Tujuan atau fungsi dari penggunaan UML Inilah beberapa tujuan atau fungsi dari penggunaan UML, yang diantaranya Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah. Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya








digunakan untuk memodelkan perangkat lunak (*Software*) saja. Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

2.13.1 Class Diagram

Class Diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok : Nama (Stereotype), Attribute, Metoda (Operation). Attribute dan operation dapat memiliki salah satu sifat berikut Private, tidak dapat dipanggil dari luar class yang bersangkutan. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya. Public, dapat dipanggil oleh siapa saja.

Hubungan Antar Class Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas.”). Pewarisan, yaitu hubungan hirarki antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang dipassing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.

Tabel 2.13.1 Daftar Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Rosa A.S dan M. Shalahuddin, 2013)

2.13.2 Use Case Diagram

Diagram *use case* menunjukkan tiga aspek dari system yaitu actor, use case dan system/sub system boundary. Actor mewakili peran orang, system yang lain atau alat ketika berkomunikasi dengan use case.

Use case diagram terdiri dari :

1) Actor

Actor adalah abstraction dari orang dan system yang lain yang mengaktifkan fungsi dari target system. Orang atau system bisa muncul dalam beberapa pesan.

2) Use Case

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Setiap use case harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan actor. Nama use case boleh terdiri dari beberapa kata dan tidak boleh ada dua use case yang memiliki nama yang sama. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.











3) Relasi/Asosiasi Use Case

Asosiasi menggambarkan aliran data atau informasi. Asosiasi atau relasi juga digunakan untuk menggambarkan bagaimana actor terlibat dalam use case. Relasi (relationship) digambarkan sebagai bentuk garis antara dua simbol dalam use case diagram.

Selain terdapat relasi-relasi antara actor dan use case, juga terdapat relasi-relasi antara *use case-use case*. Ada beberapa jenis relasi antara use case, yaitu:

- a. *Include*, digunakan untuk menggambarkan bahwa suatu *use case* merupakan fungsionalitas dari *use case* lainnya. Biasanya digunakan untuk menghindari pengkopian suatu use case karena sering dipakai.
- b. *Extend*, digunakan untuk menunjukkan bahwa satu use case merupakan tambahan fungsional dari use case yang lain jika kondisi atau syarat tertentu yang dipenuhi.

Tabel 2.13.2. Simbol *Use Case Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

(Rosa A.S dan M. Shalahudin 2013)

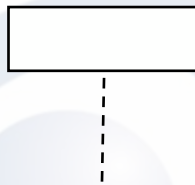
2.13.3 Sequence Diagram

Sequence diagram biasa digunakan untuk menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang mentrigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk actor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Diagram ini menunjukkan sejumlah contoh obyek dan message

(pesan) yang diletakkan diantara obyek-obyek ini di dalam use case.. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi.

A. *Obyek/Participant*

Obyek diletakkan di dekat bagian atas diagram dengan urutan dari kiari ke kanan. Setiap participant terhubung dengan garis titik-titik yang disebut lifeline. Sepanjang lifeline ada kotak yang disebut activation. Activation mewakili sebuah eksekusi operasi dari participant. Panjang kotak ini berbanding lurus dengan durasi activation.



Gambar 8. Paricipant pada sebuah Sequence Diagra

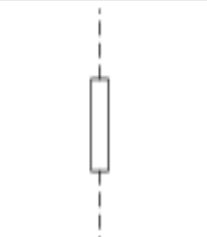

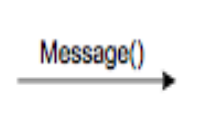

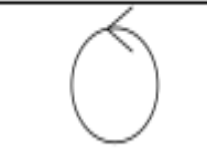
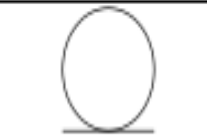
B. *Message*

Sebuah *message* bergerak dari satu participant ke participant yang lain dan dari satu lifeline ke lifeline lain. Sebuah participant bisa mengirim sebuah message kepada dirinya sendiri.



Gambar. . Simbol-Simbol *Message*

Tabel 2.13.3. Simbol *Sequence Diagram*

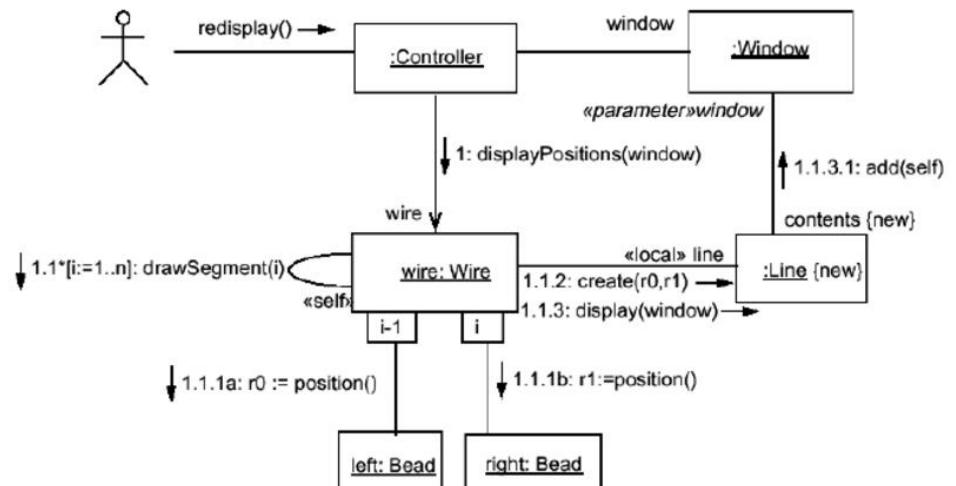
NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
		<i>Actor</i>	Digunakan untuk menggambarkan user / pengguna.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form.
4		<i>Control Class</i>	Digunakan untuk menghubungkan <i>boundary</i> dengan tabel.
5		<i>Entity Clas</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

(Rosa A.S dan M. Shalahuddin, 2013)

2.13.4 Collaboration Diagram

Menurut (Dharwiyanti, Pengantar *Unified Modelling Language*, 2011) *Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama. Dengan *collaboratioan diagram* memungkinkan untuk memodelkan pengiriman

sebuah message ke banyak obyek pada class yang sama. Demikian juga halnya untuk menunjukkan adanya obyek aktif yang mengendalikan aliran dari message. Diagram ini merupakan diagram interaksi. Diagram ini menekankan pada organisasi struktur dari objek-objek yang mengirim dan menerima pesan.



Gambar 2.13.4 Collaboration diagram

(Dharwiyanti, Pengantar *Unified Modelling Language*, 2011)

2.13.5 Activity Diagram

Activity diagram adalah tehnik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai pesan seperti halnya flowchart, letak perbedaannya adalah *Activity diagram* bisa mendukung perilaku paralel sedangkan flowchart tidak bisa.

Activity diagram dibangun dari sejumlah bentuk, dihubungkan dengan panah. Jenis Bentuk yang paling penting:

- 1) persegi panjang bulat merupakan tindakan;
- 2) berlian merupakan keputusan;
- 3) bar mewakili awal (split) atau akhir (bergabung) kegiatan bersamaan;
- 4) lingkaran hitam merupakan awal (*initial state*) dari alur kerja;

5) lingkaran hitam dikelilingi mewakili akhir (keadaan akhir).

Sebuah activity diagram mempunyai :

1) *Start point (initial node).*



Start Point

2) *End Point (activity final node).*



End Point

3) *Activities*, menggambarkan proses bisnis dan dikenal sebagai *activity state*.

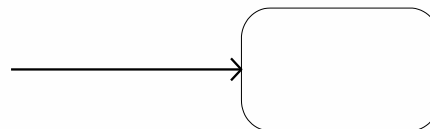


. Activities

4) *Jenis Activities*

a. *Black hole activities*

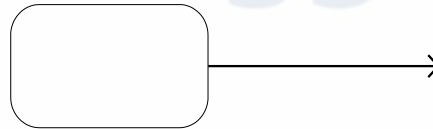
Ada masukan dan tidak ada keluaran, biasanya digunakan jika di kehendaki ada 1 atau lebih transisi.



Black Hole Activities

b. *Miracle activities*

Tidak ada masukan dan ada keluaran, biasanya dipakai pada waktu start point dan dikehendaki ada 1 atau lebih transisi.



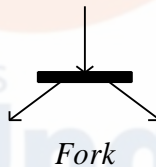
Miracle Activities

c. *Parallel activities*

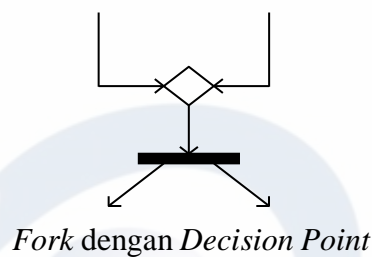
Suatu activity yang berjalan secara berbarengan. Terdiri dari :

- *Fork* (percabangan)

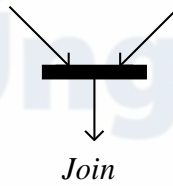
Mempunyai 1 transisi masuk dan 2 atau lebih transisi keluar.



Ketika ada lebih dari 1 transisi masuk ke fork yang sama, gabungkan dengan sebuah decision point



- *Join* (penggabungan)



d. *Decision points*

Decision point mempunyai transisi (sebuah garis dari atau ke *decision point*). Setiap transisi yang ada harus mempunyai guard (kunci).






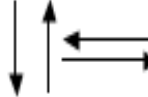
Hindarkan *decision point* yang berlebihan.

Decision point digambarkan dengan lambang wajik atau belah ketupat, seperti :



Decision Points

Tabel 2.13.5 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

(Rosa A.S dan M.Shalahudin, 2013)

2.13.6 State Machine Diagram

Diagram ini adalah state-machine diagram, berisi state, transisi, kejadian dan aktivitas. Diagram ini penting dalam memodelkan perilaku antarmuka, kelas dan kolaborasi dan menekankan kepada urutan kejadian.

2.13.7 Composite structure diagram

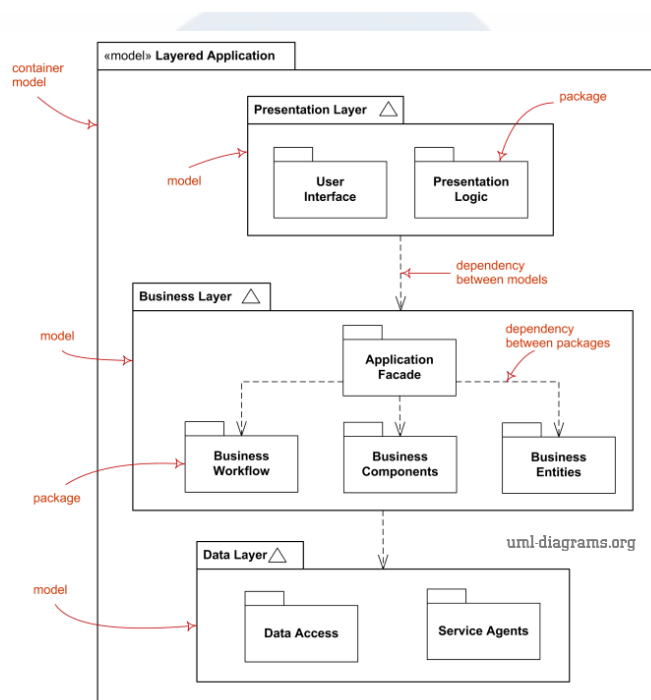
Composite structure diagram yaitu salah satu jenis diagram pada UML yang menggambarkan struktur internal dari penklasifikasian (*class, component atau use case*) dan termasuk titik-titik interaksi penklasifikasian sebagian lainnya dari suatu sistem. Ini hampir mirip seperti class diagram akan tetapi composite structure diagram menggambarkan bagian-bagian dari individu kelas saja bukan semua kelas.

2.13.8 Interaction Overview Diagram

Interaction Overview diagram yaitu salah satu jenis diagram pada UML yang berguna untuk men-visualisasikan kerjasama dan hubungan antara *activity diagram* dengan *sequence diagram*.

2.13.9 Package diagram

Package diagram yaitu salah satu jenis diagram pada UML digunakan untuk mengelompokkan kelas dan juga menunjukkan bagaimana elemen model akan disusun serta menggambarkan ketergantungan antara paket-paket.

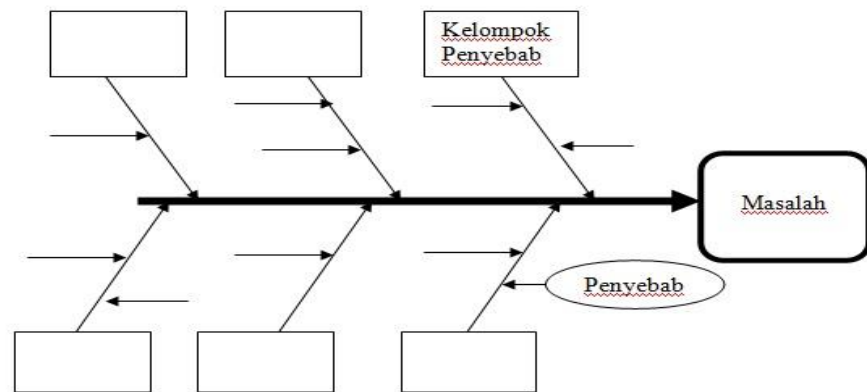


Gambar 2.13.9 *Package diagram*

2.14 Analisis Fishbone

Diagram tulang ikan atau *fishbone* diagram adalah salah satu metode/*tool* di dalam meningkatkan kualitas. Sering juga diagram ini disebut juga dengan diagram Sebab-akibat atau *cause effect* Diagram. Penemunya adalah seorang

ilmuan jepang pada tahun 60-an. Bernama Dr. Kaoru Ishikawa, metode tersebut awalnya lebih banyak digunakan untuk manajemen kualitas. Yang menggunakan data verbal (*nonnumerical*) atau data kualitatif. Dr. Ishikawa juga ditengarai sebagai orang pertama yang memperkenalkan 7 alat atau metode pengendalian kualitas (*7 tools*). Yakni *fishbone* diagram, *pareto chart*, dan *flowchart* (Yohannes Yahya Welim, 2016).



Gambar 2.14 Diagram Fishbone
(bppk.depkeu.go.id)

2.15 Metode Pengembangan Sistem

Model *Prototyping*

Sebuah prototipe adalah bagian dari produk yang mengekspresikan logika maupun fisik antarmuka eksternal yang ditampilkan. Konsumen potensial menggunakan prototipe dan menyediakan masukan untuk tim pengembang sebelum pengembangan skal besar dimulai. Melihat dan mempercayai menjadi hal yang diharapkan untuk dicapai dalam prototipe. Dengan menggunakan pendekatan ini, konsumen dan tim pengembang dapat mengklarifikasi kebutuhan dan interpretasi mereka.

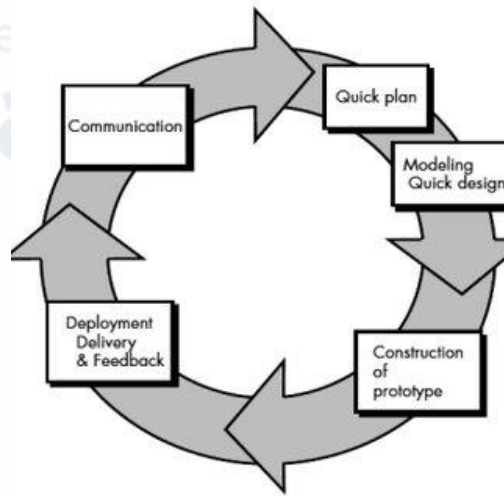
Prototyping perangkat lunak (*Software prototyping*) atau siklus hidup menggunakan prototyping (*life cycle using prototyping*) adalah salah satu

metode siklus hidup sistem yang didasarkan pada konsep model bekerja (*working model*). Tujuannya adalah mengembangkan model menjadi sistem final. Artinya sistem akan dikembangkan lebih cepat dari pada metode tradisional dan biayanya menjadi lebih rendah. Ada banyak cara untuk memprototyping, begitu pula dengan penggunaannya. Ciri khas dari metodologi ini adalah pengembang sistem (*system developer*), klien, dan pengguna dapat melihat dan melakukan eksperimen dengan bagian dari sistem komputer dari sejak awal proses pengembangan.

Dengan *prototype* yang terbuka, model sebuah sistem (atau bagiannya) dikembangkan secara cepat dan dipoles dalam diskusi yang berkali-kali dengan klien. Model tersebut menunjukkan kepada klien apa yang akan dilakukan oleh sistem, namun tidak didukung oleh rancangan desain struktur yang mendetil. Pada saat perancang dan klien melakukan percobaan dengan berbagai ide pada suatu model dan setuju dengan desain final, rancangan yang sesungguhnya dibuat tepat seperti model dengan kualitas yang lebih bagus.

Prototyping membantu dalam menemukan kebutuhan di tahap awal pengembangan, terutama jika klien tidak yakin dimana masalah berasal. Selain itu prototyping juga berguna sebagai alat untuk mendesain dan memperbaiki user interface – bagaimana sistem akan terlihat oleh orang-orang yang menggunakannya.

Salah satu hal terpenting mengenai metodologi ini, cepat atau lambat akan disingkirkan dan hanya digunakan untuk tujuan dokumentasi. Kelemahannya adalah metode ini tidak memiliki analisa dan rancangan yang mendalam yang merupakan hal penting bagi sistem yang sudah kokoh, terpercaya dan bisa dikelola. Jika seorang pengembang memutuskan untuk membangun jenis prototipe ini, penting untuk memutuskan kapan dan bagaimana ia akan disingkirkan dan selanjutnya menjamin bahwa hal tersebut telah diselesaikan tepat pada waktunya.



Gambar 2.15 *Model Prototyping*