1. **PENDAHULUAN**
   1. **Latar Belakang**

Kualitas udara dan suhu di dalam ruang kelas sangat berpengaruh terhadap kenyamanan kegiatan belajar mengajar. Buruknya kualitas udara di dalam suatu ruangan dapat disebabkan oleh banyak faktor, salah satunya adalah tingginya kadar karbon dioksida (CO2) yang berasal dari proses metabolisme tubuh manusia. Menurut Pudjiatuti (1997), semakin banyaknya individu yang menempati suatu ruangan, kecilnya ukuran ruangan, serta ventilasi yang kurang baik mengakibatkan kadar karbon dioksida yang dihasilkan semakin tinggi pula.

Karbon dioksida sendiri adalah senyawa kimia yang terdiri dari dua atom oksigen yang terikat dengan sebuah atom karbon. Senyawa ini dihasilkan oleh semua hewan, tumbuhan, fungi, dan mikroorganisme pada proses respirasi dan fotosintesis. Konsentrasi karbon dioksida di atmosfer bumi berjumlah sekitar 387 ppm, meskipun jumlah ini dapat berubah tergantung lokasi dan waktu. Konsentrasi karbon dioksida yang berlebih pada suatu tempat dapat berdampak buruk terhadap makhluk hidup disekitarnya. Eksposur karbon dioksida dapat menyebabkan turunnya performa dan produktivitas manusia (Satish, 2012). Menurut publikasi *online* yang dimuat pada *website* Wisconsin Dept. of Health Services, kadar karbon dioksida normal di dalam ruangan berpenguni berkisar antara 350 – 1.000 ppm. Individu yang terpapar karbon dioksida pada level 1.000 – 2.000 ppm akan mulai merasa mengantuk dan kelelahan., pada level 2.000 – 5.000 ppm individu akan mulai merasakan sakit kepala, sesak nafas, dan sulit untuk berkonsentrasi.

Selain karbon dioksida, faktor lainnya yang memengaruhi kenyamanan ruang kelas adalah suhu. Menurut Lippsmeier (1994), batas kenyamanan termal pada daerah khatulistiwa berkisar antara 22,5°C hingga 29°C. Sama halnya dengan karbon dioksida, tempertatur yang terlalu rendah atau yang terlalu tinggi dapat berdampak terhadap turunnya performa dan produktivitas manusia (Gunawan, 2017).

Internet of Things merupakan konsep yang telah banyak diterapkan dalam berbagai bidang, salah satunya adalah pemantauan kualitas udara. Dengan menggunakan konsep IoT, kondisi dan kualitas udara seperti suhu, konsentrasi karbon dioksida, dan lain-lain dapat diukur dan diolah menjadi informasi. Pengukuran dengan menggunakan konsep IoT tentu sangat mudah dibanding dengan pengukuran secara manual karena semua proses dilakukan secara otomatis. IoT yang pada umumnya terhubung di jaringan Internet memiliki banyak keuntungan, salah satunya adalah pemantauan data dapat dilakukan secara *realtime*.

Penelitian yang membahas tentang pengukuran kualitas udara telah ada sebelumnya, seperti yang dilakukan oleh Yulfiani Fikri (2013) tentang sistem monitoring kualitas udara, dan Victor Wijaya (2016) tentang pemantauan gas pada lingkungan, namun penelitian tersebut hanya sebatas perancangan alat pemantauan kualitas udara saja. Oleh karena itu, penulis ingin mengembangkan penelitian tersebut dengan mengaplikasikannya pada ruang kelas yang sedang digunakan untuk kegiatan belajar-mengajar. Tujuan dari penelitian ini adalah untuk mengetahui pengaruh beberapa faktor seperti banyaknya jumlah pelajar yang menempati suatu ruangan dan kondisi ruangan itu sendiri terhadap kualitas udara yang dihasilkan. Penelitian ini diharapkan dapat memberikan hasil berupa informasi kriteria kelas ideal dalam kegiatan belajar. Konsentrasi karbon dioksida dan suhu yang tinggi akan menghasilkan suasana kelas yang tidak nyaman sehingga menurunkan performa belajar siswa (Satish, 2012).

Untuk melakukan penelitian ini, penulis menggunakan mikrokontroler Arduino, sensor gas CO2, dan sensor suhu yang diletakan pada beberapa kelas dengan kondisi yang berbeda untuk mengukur konsentrasi karbon dioksida dan suhu di dalam ruangan tersebut. Selain itu dikembangkan pula sistem informasi monitoring berbasis web guna mempermudah pembacaan dan pengolahan data sensor. Pada saat proses pengambilan data, responden akan diminta untuk memasuki ruangan. Jumlah reponden akan ditingkatkan secara bertahap guna mengetahui pengaruh perubahan kadar karbon dioksida terhadap jumlah individu yang menempati ruangan. Data tersebut akan digunakan untuk mengetahui kondisi ruangan yang ideal ditinjau dari segi kualitas udara.

* 1. **Rumusan Masalah**

Berdasarkan identifikasi dari latar belakang, maka penulis menarik beberapa rumusan masalah yang membantu penulis untuk mencapai sasaran dalam penelitian

1. Bagaimana cara merancang dan mengembangkan sistem yang dapat menampilkan konsentrasi karbon dioksida pada suatu ruangan yang dapat diakses melalui web?
2. Bagaimana hasil pengukuran konsentrasi karbon dioksida dan suhu untuk menentukan ruang kelas ideal bagi proses belajar mengajar?
   1. **Tujuan Penelitian**

Adapun tujuan penelitian ini adalah:

1. Menjelaskan cara merancang dan mengembangkan sistem informasi monitoring berbasis web.
2. Merancang sistem informasi monitoring yang dapat memantau kualitas udara secara real time.
3. Menjelaskan hasil pengukuran kualitas udara berdasarkan kondisi kelas tertentu untuk menentukan ruang kelas ideal bagi proses belajar mengajar.
   1. **Manfaat Penelitian**

Dengan adanya penelitian ini, diharapkan dapat memberikan gambaran mengenai tahap pengembangan sistem informasi monitoring kualitas udara, serta memberikan manfaat bagi instansi pendidikan dalam menentukan jumlah pelajar yang sesuai untuk menempati suatu ruang kelas dengan tetap menjaga kenyamanan ruangan ditinjau dari segi kualitas udara.

* 1. **Batasan Masalah**

Penelitian ini hanya membahas pengukuran kondisi dan kualitas udara berupa konsentrasi karbon dioksida dan suhu udara.

1. **TINJAUAN PUSTAKA**
   1. **Smart City**

*Smart city* atau kota pintar merupakan konsep perencanaan kota dengan memanfaatkan perkembangan teknologi. *Smart city* diharapkan dapat membuat hidup lebih mudah dengan tingkat efisiensi dan efektifitas yang tinggi. Penerapan *smart city* didasarkan pada upaya untuk mengatasi berbagai masalah yang ada di kota, sehingga suatu kota layak dihuni dan kehidupan yang ada di dalamnya lebih baik dari sebelumnya (Hasibuan, 2019).

Dengan *smart city*, berbagai macam informasi yang ada dapat diperoleh dengan memanfaatkan sensor yang terpasang di setiap sudut kota, dianalisis menggunakan aplikasi, dan kemudian disajikan sesuai dengan kebutuhan pengguna. Berikut merupakan beberapa konsep *smart city* (Hasibuan, 2019).

1. Sebuah kota dapat berkinerja dengan baik dengan berpandangan ke dalam ekonomi, pemerintahan, mobilitas, penduduk, dan lingkungan hidup.
2. Kota dapat mengontrol dan mengintegrasikan semua infrastruktur, seperti jalan, jembatan, kereta, bandara, gedung, komuniasi, air, listrik, dan lainnya.
3. *Smart city* dapat menghubungkan infrastruktur fisik dan infrastruktur IT untuk meningkatkan kecerdasar kota.
4. Smart city membuat fasilitas yang meliputi pendidikan, kesehatan, dan transportasi lebih cerdas dan efisien.

Menurut Cohen, yang dikutip dari *smartcity.jakarta.go.id*, smart city memiliki 6 indikator, diantaranya:

1. *Smart Economy*

Pilar ini meliputi event internasional, pembangunan dan penelitian, serta pembangunan startup dan perusahaan baru.

1. *Smart Governance*.

Hal penting pada kota pintar adalah keterbukaan dan kemudahan akses data-data informasi pemerintahan oleh masyarakat. Informasi terkait kepentingan publik disosialikasikan dengan baik sehingga terjadil komunikasi yang baik antara pemerintah dan masyarakat. Oleh karena itu, penyediaan internet yang baik serta sumber daya manusia di pemerintahaan berperan dalam mewujudkan *smart governance*.

1. *Smart People*

Warga yang pintar juga penting untuk mewujudkan suatu kota pintar. Hal utama dan yang paling penting adalah pendidikan.

1. *Smart Mobility*

Pada pewujudan smart city, penggunaan transportasi publik harus digalakkan. Kenyamanan warga dalam menggunakan angkutan umum harus terjamin, misalnya dengan menyediakan media transportasi yang aman, cepat, dan tepat waktu*.*

1. *Smart Environment*

Kota pintar harus memiliki lingkungan hidup yang dikelola secara pintar pula. Ketersediaan ruang hijau yang memadai, pengeloaan sampah, dan penanganan global warming adalah beberapa poin yang perlu dilakukan untuk mewujudkan *smart environment*.

1. *Smart Living*

Faktor lingkungan hidup yang sehat, keamanan, serta kebahagiaan menjadi hal yang penting untuk mewujudkan *smart living*.

***Smart buildings***

*Smart buildings* atau bangunan pintar merupakan salah satu bagian dari *smart environment* yang merupakan indikator dari *smart city.* Dalam pengembangan *smart city, smart building* dianggap penting karena dapat memberikan kenyamanan, keselamatan, dan penghematan energi, yang berlangsung secara otomatis, pada gedung atau rumah (Usmanto, 2017). *Smart building system* dapat digunakan untuk mengendalikan hampir semua peralatan di gedung atau rumah, mulai dari pengaturan lampu hingga ke berbagai peralatan rumah tangga. Sistem ini dapat dikendalikan dengan menggunakan perintah suara, ataupun kendali jarak jauh (*remote*).

**Penerapan *Internet of Things* pada smart building**

*Internet of Things* merupakan konsep yang dalam penerapannya berupaya untuk mengintegrasikan semua perangkat elektronik menggunakan jaringan internet. Salah satu penerapan IoT adalah pada pengembangan *smart house* dan *smart building* (Junaidi, 2015). Pada *smart building,* pengembangan IoT dapat berupa keamanan otamatis, penanganan kebakaran, pengendalian lampu/cahaya, monitoring, ventilasi, dan manajemen energi.

* 1. ***Internet of Things***

Menurut McKinsey Global Institute*, Internet of Things* adalah konsep yang memungkinkan terhubungnya mesin dan benda lainnya dengan sensor untuk memperoleh dan mengolah data sendiri. Data pada benda didapat dengan menggunakan sensor yang diletakkan pada benda yang bersangkutan. Data tersebut kemudian dikirim melalui jaringan internet untuk selanjutnya diolah sendiri atau berkomunikasi dengan benda-benda lain yang terhubung di jaringan yang sama. Kelebihan IoT dibanding pengukuran secara konvensional adalah semua proses dilakukan secara otomatis. Hal ini sangat membantu jika data yang dibutuhkan harus dapat diakses secara cepat.

**Pemanfaatan IoT di berbagai bidang**

Beberapa contoh penerapan IoT adalah sebagai berikut:

* 1. Bidang Pertanian

Saat ini di bidang pertanian, khususnya di Indonesia sudah banyak yang menggunakan teknologi IoT dalam membantu peningkatan hasil pangan. Contohnya dalah penyiraman tanaman otomatis, pendeteksi hama, *smart agriculture*, dan lain-lain.

1. Bidang Kesehatan

IoT telah banyak membantu pekerjaan di bidang kesehatan, diantaranya pemantauan kondisi pasien jarak jauh, sensor untuk mengurangi rasa nyeri pada kaki, sensor untuk monitoring infus, dan lain sebagainya.

1. Bidang Keamanan

Beberapa contoh penerapan IoT pada bidan keamanan antara lain *remote control* garasi melalui *smartphone*, pemantauan kamera pengawas, pendeteksi maling, sensor pendeteksi asap/kebakaran, dan lain sebagainya.

1. Bidang Kelautan

Saat ini, teknologi IoT juga sudah diterapkan dalam bidang kelautan misalnya monitoring perahu nelayan saat sedang tidak melaut, monitoring kondisi cuaca, pendeteksi tsunami, dan lain sebagainya.

1. Bidang Pendidikan

Penerapan teknologi IoT sangat berperan terhadap kualitas pendidikan, diantaranya monitoring kondisi ruang kelas, sistem monitoring anak, dan lain sebagainya.

**IoT pada pemantauan kualitas udara**

Internet of Things merupakan konsep yang telah banyak diterapkan dalam berbagai bidang, salah satunya adalah pemantauan kualitas udara. Dengan menggunakan konsep IoT, kondisi dan kualitas udara seperti suhu dan konsentrasi karbon dioksida dapat diukur menggunakan sensor dan kemudian dikirim melalui jaringan internet untuk kemudian diolah menjadi informasi. IoT sangat membantu dalam pengukuran kondisi udara yang harus dilakukan secara cepat terus menerus.

Beberapa sensor yang dapat digunakan dalam pengukuran kondisi dan kualitas udara diantaranya adalah DHT11 (sensor suhu), MG811 (sensor karbon dioksida), MQ7 (sensor karbon monoksida), PM10 (sensor debu), dan lain-lain. Untuk mengirimkan data yang diperoleh oleh sensor-sensor ini, diperlukan sebuah adapter jaringan ethernet atau Wi-Fi, salah satunya yang paling bayak digunakan adalah ESP8266. Data sensor akan dikirim melalui jaringan internet ke web server dan kemudian diolah menjadi informasi.

**Penelitian IoT di bidang pemantauan kualitas udara**

Telah banyak penelitian IoT yang membahas mengenai kualitas udara, seperti yang dilakukan oleh Yulfiani Fikri (2013) tentang sistem monitoring kualitas udara, dan Victor Wijaya (2016) tentang pemantauan gas pada lingkungan.

Penelitian yang dilakukan oleh Yulfiani Fikri (2013) berfokus pada pengukuran kondisi gas-gas berbahaya seperti CO, NO, dan NO2 yang dapat memberikan dampak negatif bagi kesehatan. Penelitian dilakukan menggunakan mikrokontroller AVR tipe ATmega 8535 sebagai unit pusat kontrol. Data yang didapat oleh sensor dikirim ke web server untuk dapat dipantau.

Penelitian selanjutnya adalah yang dilakukan oleh Victor Wijaya (2016). Penelitian ini berfokus pada pengukuran gas seperti CO dan CO2 serta suhu. Sama halnya dengan penelitian Fikri (2013), penelitian ini mendeteksi kualitas udara dan kemudian menampilkannya pada aplikasi web.

Pada penelitian tersebut, pengembangan sistem hanya terbatas pada perancangan alat dan pembacaan data sensor saja. Maka penelitian ini bertujuan untuk mengembangkan penelitian sebelumnya, dimana data sensor yang telah dikumpulkan akan diolah dan dijadikan dasar dalam menentukan ruang kelas ideal bagi proses belajar.

* 1. **Sistem**

Sistem merupakan kumpulan elemen yang saling terkait dan bekerja sama untuk mengolah masukan (input) sampai menghasilkan keluaran (output) yang diinginkan. Menurut Al-Fatta (2007), suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu:

* 1. Komponen

Suatu sistem terdiri dari beberapa komponen yang saling berinteraksi dan membentuk suatu kesatuan. Komponen sistem dapat berupa suatu submistem atau bagian-bagian dari sistem.

* 1. Batas Sistem

Batas sistem merupakan daerah yang membatasi suatu sistem dengan sistem lainnya. Batas suatu sistem menunjukan ruang lingkup sistem tersebut.

* 1. Lingkungan Luar Sistem

Lingkungan luar sistem merupakan hal-hal diluar batas sistem yang memengaruhi operasional sistem. Lingkungan luar sistem dapat bersifat menguntungkan ataupun merugikan sistem. Lingkungan luar sistem yang menguntungkan harus tetap dijaga dan dipelihara, sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, karena dapat menganggu kelangsungan hidup sistem.

* 1. Penghubung Sistem

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Dengan adanya penghubung sistem ini, suatu subsistem dapat berintegrasi dengan subsistem lainnya untuk membentuk satu kesatuan.

* 1. Masukan Sistem

Masukan adalah sesuatu yang dimasukkan ke dalam sistem untuk kemudian diproses.

1. Keluaran Sistem

Keluaran adalah sesuatu yang dihasilkan oleh sistem setelah tahapan proses sebelumnya.

1. Pengolah Sistem

Pengolah sistem merupakan bagian yang mengolah masukan menjadi keluaran.

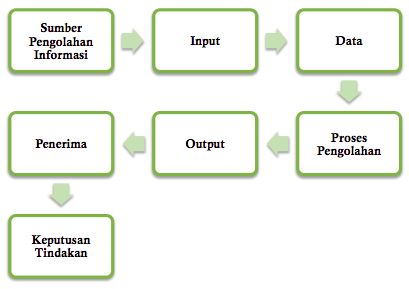
1. Sasaran Sistem

Suatu sistem yang dibuat harus memiliki tujuan dan sasaran. Suatu sistem dapat dikatakan berhasil jika sasaran atau tujuannya terpenuhi.

* 1. **Informasi**

Informasi adalah data yang telah diolah menjadi bentuk yang berarti bagi penerimanya dan dapat digunakan bagi pengambilan keputusan saat ini atau saat mendatang. Informasi diolah menjadi bentuk yang lebih berguna dan berarti bagi penerimanya (Kadir, 2014). Sumber informasi adalah data yang merupakan gambaran kegiatan-kegiatan yang nyata.

Informasi merupakan hal yang sangat penting dalam pengambilan keputusan organisasi. Jika informasi yang dihasilkan akurat, maka informasi tersebut dapat digunakan sebagai pedoman untuk segala tindakan penting yang tepat pula. Siklus informasi adalah gambaran proses data menjadi informasi yang bermanfaat bagi pengguna dalam pengambilan keputusan.



**Gambar 1.** Siklus Informasi

* 1. **Sistem Informasi**

Menurut Kadir (2014), sistem informasi adalah sejumlah komponen yang terdiri atas manusia, komputer, teknologi informasi, dan prosedur kerja, serta memiliki sesuatu yang diproses, dan dimaksudkan untuk mencapai suatu tujuan tertentu.

Sistem informasi merupakan suatu sistem didalam organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, bersifat manajerial, mendukung operasi, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan. Sistem informasi terdiri dari beberapa komponen yang disebut dengan istilah blok bangunan, yaitu:

1. Blok masukan

Blok masukan meliputi data yang masuk ke dalam sistem informasi. Data yang dimaksud termasuk metode-metode dan media yang digunakan untuk menangkap data yang dimasukkan.

1. Blok model

Blok model terdiri dari kombinasi prosedur, logika, dan metode matematik yang digunakan untuk memanipulasi data input untuk menghasilkan keluaran yang diinginkan.

1. Blok keluaran

Blok ini meliputi hasil keluaran yang berupa informasi berkualitas dan dokumentasi yang berguna bagi pemakai sistem. Kualitas informasi didasarkan atas data masukan dan prosedur pengolahan.

1. Blok teknologi

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan serta mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian secara keseluruhan.

1. Blok basis data

Blok basis data merupakan kumpulan data yang saling berhubungan satu dengan yang lain. Basis data disimpan di perangkat keras komputer dan perangkat keras digunakan untuk memanipulasinya.

1. Blok kendali

Blok kendali meliputi perancangan hal-hal untuk mencegah terjadinya sesuatu yang dikira dapat menggangu dan merusak sistem. Beberapa faktor yang dapat merusak sistem informasi, misalnya bencana alam, temperatur tinggi, air, debu, ataupun kecurangan-kecurangan dalam sistem itu sendiri. Pengendalian perlu dilakukan agar jika terjadi kesalahan, dapat langsung diatasi (Hutahaean, 2015).

* 1. **Monitoring**

Menurut Peraturan Pemerintah Nomor 39 Tahun 2006 (dalam IPDN, 2011), monitoring merupakan kegiatan mengamati suatu keadaan atau kondisi secara seksama. Tujuan monitoring adalah agar semua data yang diperoleh dari hasil pengamatan dapat menjadi acuan dan landasaan dalam mengambil keputusan yang diperlukan. Monitoring merupakan proses pengukuran dan pengumpulan data yang berfokus pada proses keluaran.

* 1. **Sistem Informasi Monitoring**

Sistem informasi monitoring adalah sistem informasi yang dikembangkan untuk mengetahui perubahan yang terjadi akibat pergerakan veriabel secara kontinyu dalam kurun waktu tertentu. Sistem informasi monitoring digunakan untuk melihat faktor-faktor yang menyebabkan perubahan untuk selanjutnya dianalisis dan dijadikan acuan untuk pengambilan keputusan oleh manajemen.

* 1. **Karbon Dioksida**

Karbon dioksida (CO2) adalah gas yang terdiri dari dua atom oksigen yang terikat dengan atom karbon secara kovalen. Karbon dioksida dinilai penting karena digunakan oleh tumbuhan untuk melakukan proses fotosintesis. Di atmosfer bumi, konsentrasi karbon dioksida mencapai 387 ppm berdasarkan volume, meskipun nilai ini dapat berubah tergantung waktu dan lokasi. Pada jumlah yang sedikit, karbon dioksida dinilai tidak berbahaya, namun dalam jumlah yang banyak, karbon dioksida dapat berbahaya bagi kesehatan makhluk hidup, atau bahkan dapat menyebabkan koma (Badan POM RI, 2010).

Karbon dioksida dihasilkan oleh semua makhluk hidup pada proses respirasi dan digunakan oleh tumbuhan pada proses fotosintesis. Selain berasal dari makhluk hidup, kejadian alam juga menghasilkan gas karbon dioksida seperti pembakaran bahan bakar fosil, aktivitas volkanik, dan proses geotermal lain seperti pada mata air panas.

Berdasarkan penelitian yang dilakukan oleh Satish (State University of New York, 2012), konsentrasi karbon dioksida dapat memengaruhi kinerja otak manusia dalam mengambil keputusan. Penelitian ini menunjukan bahwa performa manusia dalam melakukan pengambilan keputusan akan mulai berkurang saat karbon dioksida berada pada level diatas 1.000 ppm. Performa akan terus berkurang secara drastis saat konsentrasi mencapai level 2.500 ppm.

Winconsin Dept. of Health Service membahas dalam publikasi *online*-nya pengaruh konsentrasi kabon dioksida terhadap kesehatan manusia.

**Tabel 1**. Pengaruh konsentrasi karbon dioksida terhadap kesehatan manusia menurut Winconsin Department of Health Service

|  |  |
| --- | --- |
| **Konsentrasi (ppm)** | **Efek** |
| 250 - 350 | Keadaan di luar ruangan |
| 230 – 1.000 | Keadaan normal dalam ruangan berpenghuni dimana penghuni dapat bernafas dengan baik |
| 1.000 – 2.000 | Penghuni mulai merasa mengantuk dikarenakan kualitas udara yang buruk |
| 2.000 – 5.000 | Penghuni mulai mengalami kelelahan, sakit kepala, sesak nafas, serta peningkatan detak jantung. |
| 5.000 | Kualitas udara tidak wajar, penghuni kekurangan oksigen |
| 40.000 | Sangat berbahaya bagi kesehatan |

Meningkatnya konsentrasi karbon dioksida pada suatu ruangan selain disebabkan oleh banyaknya orang yang menghuni ruangan tersebut, juga disebabkan oleh faktor lain seperti luas ruangan dan alur udara masuk dan keluar melalui ventilasi (Pudjiastuti, 1997).

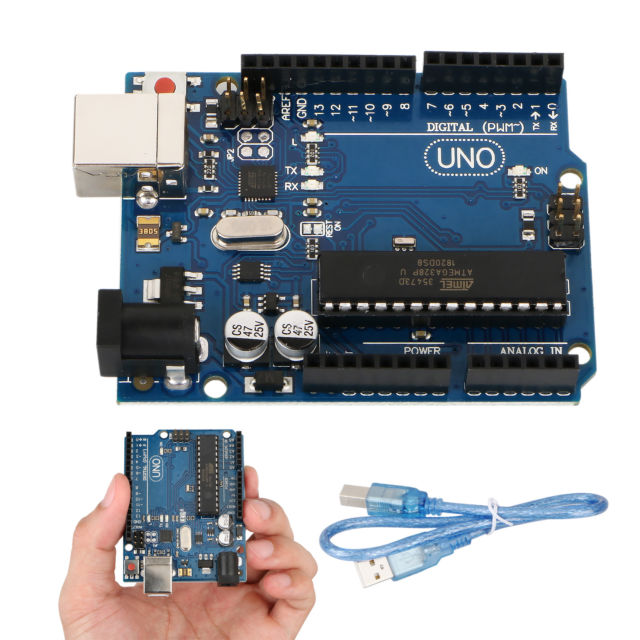
* 1. **Temperatur Udara**

Temperatur udara merupakan salah satu faktor yang paling berpengaruh terhadap kondisi nyaman manusia. Menurut Lippsmeir (1994), batas kenyamanan termal pada daerah khatulistiwa berada pada kisaran 22,5°C – 29°C. Faktor lainnya yang memengaruhi kenyamanan diantaranya adalah radiasi panas, kelembaban udara, dan kecepatan udara (Setyohadi, 2011).

* 1. **Arduino**

Arduino adalah papan elektronis yang di dalamnya terdapat mikrokontroler buatan perusahaan Atmel dan berbagai perangkat pendukung yang memungkinkan siapa saja dengan mudah dapat membuat berbagai proyek elektronika. Terdapat berbagai jenis Arduino, salah satu yang terkenal dinamakan Genuino atau Arduino Uno (Kadir, 2018).

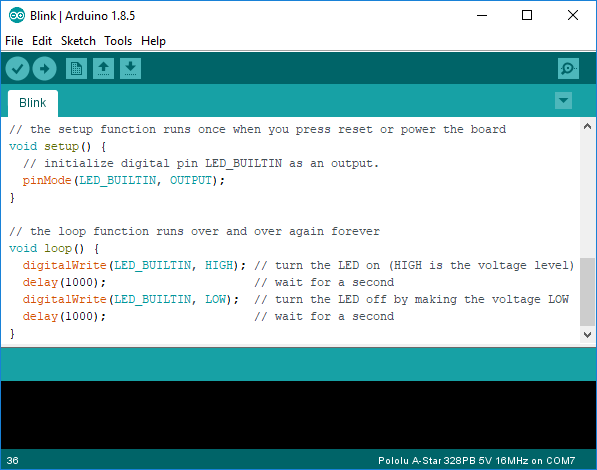
Arduino adalah alat prototipe elektronika yang mudah digunakan baik dari sisi perangkat keras/hardware maupun perangkat lunak/software. Kelebihan utama Arduino adalah jumlah komunitas yang besar sehingga tersedia *code library* dan *hardware support module* yang banyak. Arduino merupakan pengendali mikro single-board yang bersifat sumber terbuka *(open source)* (Kadir, 2018)*.*



**Gambar 2.** Papan Arduino Uno

Untuk memprogram Arduino, diperlukan sebuah software yang bernama Arduino IDE. Arduino IDE adalah suatu software IDE (Integrated Development Environment) yang banyak digunakan untuk membuka, menyunting, mevalidasi, dan mengunggah kode Arduino ke papan Arduino. Aplikasi ini dapat diunduh secara gratis melalui website resmi Arduino. Program yang digunakan pada Arduino disebut dengan *sketch*. *Sketch* sendiri adalah source code Arduino yang ditulis menggunakan bahasa pemrograman C/C++. Pemrograman Arduino menggunakan umumnya menggunakan *library* khusus yang dibuat untuk mengontrol hardware yang terpasang di papan Arduino.

Arduino merupakan pilihan populer dalam pengembangan sistem berbasis IoT. Selain mendukung banyak modul dan sensor, komunitas pengguna Arduino juga banyak sehingga jika terdapat kendala dalam perancangan alat, banyak orang yang dapat membantu untuk mengatasinya. Faktor harga yang relatif murah juga menjadi pertimbangan dalam memilih Arduino.



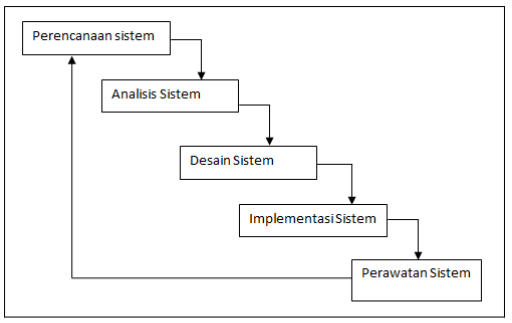
**Gambar 3.** Antarmuka Arduino IDE

Sebagai suatu alat prototipe, pemrogram dapat dengan bebas memasang sensor-sensor maupun perangkat-perangkat lain yang diinginkan ke papan arduino. Adapun beberapa perangkat yang dapat bekerja dengan arduino sebagai berikut:

1. DT-Sense Gas Sensor MG-811, berfungsi sebagai sensor pengukur konsentrasi karbon dioksida.
2. HDT11, berfungsi sebagai sensor pengukur temperatur udara.
3. ESP8266, berfungsi sebagai perangkat penghubung arduino ke perangkat lain melalui jaringan Wi-Fi.
   1. **Metode Pengembangan Perangkat Lunak**

Metodologi pengembangan sistem informasi adalah suatu metode yang dilakukan untuk mengembangkan sistem informasi berbasis komputer. Menurut Ladjamudin, metode System Development Life Cycle atau disingkat dengan SDLC merupakan metode pengembangan yang berfungsi sebagai mekanisme untuk mengindentifikasi perangkat lunak. Proses pengembangan perangkat lunak melewati beberapa tahapan, mulai dari perencanaan perangkat lunak tersebut, pengoperasian, hingga pemeliharaan (Supriyanto, 2010).

Tahapan SDLC dikerjakan secara berurut dimulai dari perencanaan, analisis, desain, implementasi, serta perawatan.



**Gambar 4.** Siklus Hidup Pengembangan Sistem

* 1. Perencanaan Sistem

Tahapan perencanaan menekankan pada aspek studi kelayakan sistem. Aktivitas yang terjadi pada tahapan ini meliputi pembentukan tim pengembang, mengidentifikasi tujuan dan ruang lingkup pengembangan, mengidentifikasi masalah, serta menentukan strategi yang akan dilakukan dalam pengembangan sistem.

* 1. Analisis Sistem

Tahapan ini meliputi studi literatur untuk menemukan kasus yang dapat ditangani oleh sistem, penentuan kasus dan permodelan sistem, mengklasifikasi masalah dan solusi yang mungkin diterapkan, serta mendefinisikan kebutuhan sistem.

* 1. Perancangan Sistem

Pada tahap ini, semua kasus yang telah dibahas pada tahap perancangan, direalisasikan menurut desain yang telah dirancang sebelumnya. Beberapa aktivitas yang ada pada tahapan ini diantaranya perancangan *user interface* dan pembuatan skema database.

* 1. Implementasi Sistem

Tahapan implementasi sistem meliputi pengujian dan perbaikan serta penggunaan aplikasi.

* 1. Pemeliharaan Sistem

Sistem yang telah beroperasi membutuhkan administrator yang bertugas untuk menjaga sistem agar tetap dapat beroperasi dengan baik, dan mampu menangani berbagai permasalahan yang mungkin terjadi.

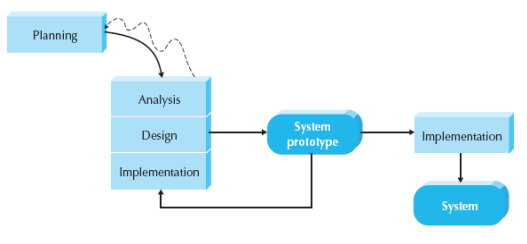
**Tabel 2**. Tabel Perbanding Model Software Development Life Cycle

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model /Features** | **Waterfall** | **V-Shape** | **CMM** | **RUP** | **Prototype** | **Incremental** | **Spiral** | **RAD** | **JAD** | **Agile** |
| Requiment Specifications | Beginning | Beginning | At second level | Beginning | Frequently changed | Begining | Begining | Time-box release | Prototype | Frequently changed |
| Understanding Requiments | Well Understood | Easly understood | Easily understood | Difficul to understand | Not well understood | Well understood | Well understood | Easily understood | Easily understood | Well Understood |
| Cost | Low | Expensive | High | Expensive | High | Low | expensive | Low | expensice | Very High |
| Guarantee of Success | Low | High | High | Not guaranteed | Good | High | High | Good | Low but for long period | Very High |
| Resource Control | Yes | Yes | Yes | Yes | No | Yes | Yes | Yes | Yes | No |
| Cost Control | Yes | Yes | Varies | Yes | No | No | Yes | Yes | Yes | Yes |
| Simplicity | Simple | Intermediate | Intermediate | Simple and Clear | Simple | Intermediate | Intermediate | Very Simple | Simple | Complex |
| Risk Involvement | High | Low | Varies acc to level | Critical risks in the early stages | Low | Easlity manage | Low | Very Low | Varies | Reduced |
| Expertise Required | High | Medium | Varies acc to level | Yes | Medium | High | High | Medium | High | Very High |
| Changes Incorporated | Difficult | Difficul | Medium | Easy | Easy | Easy | Easy | Easy | Medium | Difficult |
| Risk Analysis | Only at Beginning | Yes | Yes | Yes | No Risk Analysis | No Risk Analysis | Yes | Low | Yes | Yes |
| User Involvement | Only at Begining | At the beginning | Only at beginning | At beginning and at the last phase | High | Intermediate | High | Only at the begining | In the design and development | High |
| Overlapping Phases | No such phase | No | No | Yes | Yes | No | Yes | No | No | Yes |
| Flexibility | Rigid | Little Flexible | Highly Flexible | Considerable | Highly Flexible | Less Flexible | Flexible | High | Flexible | Highly Flexible |
| Maintanance | Least Glamorous | Least | Typical | Promote Maintainability | Routine Maintance | Promotes maintabinibility | Typical | Easily maintained | Rigorously at all times | Promotes maintainabillity |
| Integrity & Security | Vital | Limited | Limited | Very important | Weak | Robust | High | Vital | Limited | Demostable |
| Reusability | Limited | To some Extent | Yes | Supports reusability of exisiting classes | Weak | Yes | Yes | Some Extent | Limited | Use Case Reuse |
| Interface | Minimal | Minimal | Crucial | User Interface | Crucial | Crucial | Crucial | Minimal | Crucial | Model driven |
| Documentation & Training required | Vital | Yes | Yes | Yes | Weak | Yes | Yes | Limited | Limited | Yes |
| Time Frame | Long | Acc to project size | Quite long | Short time freame | Short | Very Long | Long | Short | Medium | Least Possible |

(Sumber: Taya & Gupta, 2011)

* 1. **Model *Prototype***

Pada model *protype*, dilakukan tahapan analisis, desain, dan implementasi secara bersamaan. Ketiga tahapan tersebut dilakukan berulang sampai sistem selesai dibangun. Pada model ini, analisis dan desain dasar sistem dibangun apa adanya serta fitur-fitur yang belum lengkap. Sistem uji coba kemudian ditunjukkan kepada pengguna untuk mendapat masukan yang nantinya digunakan untuk menganalisis ulang, mendesain ualng, dan mengimplementasikan ualng pada uji coba sistem yang kedua. Proses tersebut terus berlanjut hingga sistem dapat dikatakan telah memenuhi kebutuhan pengguna. Setelah uji coba dapat dikatakan sempurna, maka akan diimplementasikan ke dalam sistem yang sesungguhnya (Nugroho, 2016).



**Gambar 5.** Model Prototype

Kelebihan metode ini yaitu sistem yang dibuat langsung bisa digunakan oleh pengguna meskipun sistem tersebut belum bisa diterapkan pada perusahaan yang bersangkutan. Metode ini lebih baik karena pengguna dan pengembang dapat langsung saling berinteraksi dan mengerti apa yang bisa dilaklukan dan mana yang tidak bisa dilakukan (Nugroho, 2016).

Menurut Yasin (2012), model protype cocok dipakai pada kondisi berikut:

* 1. Definisi pengguna bersifat umum, pengguna belum memiliki gambaran pasti mengenai sistem yang akan dibuat.
  2. Pengembang merasa belum yakin mengenai algoritma yang digunakan, lingkungan sistem, dan bentuk serta karakteristik antarmuka.
  3. Terdapat ketidakpastian oleh pengguna mengenai apa yang dinginkan.
  4. Terdapat ketidakpastian oleh pengembang mengenai apa yang dilakukan.

Pressman (2015) menjabarkan model prototype menjadi beberapa tahapan diantaranya:

***Communication***. Tahapan komunikasi merupakan tahapan yang dilakukan dengan seluruh pihak yang terlibat dalam sistem. Tujuannya untuk mengidentifikasi syarat-syarat serta kebutuhan sistem.

***Quick Plan.*** Tahapan perencanaan pembuatan sistem secara cepat setelah proses komunikasi terjadi.

***Modelling Quick Design.*** Tahapan ini proses yang dilakukan ialah membuat model sistem serta desain sistem secara cepat dengan berfokus pada tampilan yang dapat dilihat oleh pengguna misalnya anarmuka sistem (GUI) atau format tampilan keluaran sistem.

***Construction of Prototype.*** Tahapan pembuatan prototype desain berdasarkan desain-desain yang telah dibuat.

***Deployment, Delivery and Feedback..*** Tahapan dimana prototype yang telah dibangun ditampilkan serta dievaluasi oleh stakeholders yang kemudian memberikan umpan balik yang digunakan untuk mengindetifikasi requirement lebih lanjut, sehingga bagi pengembang menjadi lebih memahami apa yang diperlukan.

* 1. ***Unified Modeling Language* (UML)**

Pada pengembangan perangkat lunak, diperlukan suatu bahasa yang dapat digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar semua orang dapat mengerti pemodal perangkat lunak. Pada awalnya sempat muncul dan berkembang *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural, kemudian ada juga *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem realt time (Iskandar, 2019).

Pada perkembangan teknologi berorientasi objek, muncul sebuah standarisasi pemodelan perangkat lunak yang dibangun dengan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* atau disingkat dengan UML. UML muncul karena adanya kebutuhan pemodeal visual untuk menggambarkan, membangun, dan dokumentasi dari perangkat lunak. UML adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks pendukung (Iskandar, 2019).

***Use Case Diagram***

Use case diagram merupakan pemodelan untuk menggambarkan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara sederhana, use case digunakan untuk memahami fungsi apa saja yang ada pada suatu sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Rosa, 2013). Use case memiliki berbagai notasi, diantaranya:

**Tabel 3.** Simbol Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Simbol | Fungsi | Notasi |
| 1. | Use Case | Merupakan fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |  |
| 2. | Aktor | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi. |  |
| 3. | Association | Sebuah jalur komunikasi antar seorang aktor dengan satu use case dimana actor tersebut dapat berpartisipasi |  |
| 4. | Extend | Penyisipan sebuah fungsionalitas tambahan ke dalam sebuah use case dasar yang bersifat opsional | <<extend>> |
| 5. | Include | Penyisipan sebuah fungsionalitas tambahan ke dalam sebuah use case dasar dimana fungsional tersebut bersifat mandatory (wajib) | <<include>> |
| 6. | Use Case Generalization | Sebuah hubungan antara sebuah use case umum dengan use case yang lebih spesifik. |  |

(Sumber : Rosa, 2013)

***Activity Diagram***

Diagram aktivitas menggambarkan aliran kerja dari sebuah sistem yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa, 2013). Berikut adalah simbol-simbol pada diagram aktivitas:

**Tabel 4.** Simbol Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Simbol | Fungsi | Notasi |
| 1. | Status Awal | Status awal aktivitas sistem. |  |
| 2. | Aktivitas | Aktivitas yang dilakukan sistem, diawali dengan kata kerja | Aktivitas |
| 3. | Percabangan | Asosiasi percabangan jika terdapat pilihan lebih dari satu. |  |
| 4. | Penggabungan | Asosiasi penggabungan dimana beberapa aktivitas digabungkan menjadi satu. |  |
| 5. | Status Akhir | Status akhir sistem |  |
| 6. | Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi | |  | | --- | |  | |  | |

(Sumber : Rosa, 2013)

***Sequence Diagram***

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek yang dikirmkan dan diterima antar objek. Untuk menggambarkan diagram sekuen perlu diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. (Rosa, 2013). Berikut adalah simbol yang digunakan pada diagram sekuen:

**Tabel 5.** Simbol Sequence Diagaram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Deskripsi** |  |
| Aktor | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang belum tentu merupakan orang. |  |
| Gaya Hidup / *Life Style* | Menyatakan kehidupan suatu objek |  |
| Objek | Menyatakan objek yang berinteraksi pesan | Nama objek : nama kelas |
| Waktu Aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dalam waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya. |  |
| Pesan Tipe Create | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat | <<create>> |
| Pesan Tipe Call | Menyatakan suatu objek memanggil operasi/metode yang ada pada dirinya sendiri | 1 : nama\_metode() |
| Pesan Tipe Send | Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi | 1 : masukan |
| Pesan Tipe Return | Menyatakan suatu objek yang telah menjalankan suatu operasi atau meyode mengasilkan suatu kembalian ke objek tertentu. | 1 : keluaran |
| Pesan Tipe Destroy | Menyatakan suatu objek mengakhir hidup objek lain, arah panah mengarah pada objek yang diakhiri. | <destroy>> |

(Sumber : Rosa, 2013)

* 1. **Rancangan Acak Lengkap (RAL)**

Rancangan percobaan merupakan bagian dari penelitian ilmiah. Jenis-jenis rancangan percobaan yang sering dilakukan adalah Rancangan Acak Lengkap, Rancangan Kelompok Acak Lengkap, Rancangan Persegi Latin, dan Rancangan Persegi Latin Graeco (Lentner, 1986).

Rancangan Acak lengkap adalah rancangan lapangan pada suatu lokasi yang homogen. Rancangan ini dikatakan acak karena setiap percobaan mempunyai peluang yang sama untuk mendapatkan perlakukan sedangkan dikatakan lengkap karena seluruh perlakuan yang dirancang dalam percobaan tersebut digunakan (Lentner, 1986).

Rancangan acak lengkap merupakan rancangan percobaan yang paling sederhana jika dibandingkan dengan rancangan-rancangan lainnya. Dalam rancangan ini sumber keragaman yang diamati hanya perlakuan dan galat. Oleh karena itu, RAL cocok digunakan untuk kondisi lingkungan, alat, dan media yang homogen (Hanafiah, 2000).

**Kelebihan dan Kelemahan RAL**

Menurut Lentner (1986), kelebihan RAL adalah sebagai berikut:

* 1. *Fleksibel*. Disesuaikan dengan sumber keragaman yang ada dan tidak ada batasan antara jumlah perlakukan atau ulangan.
  2. *Analisis dan perhitungan mudah*. Dari semua perancangan percobaan, RAL adalah yang paling mudah dalam analisisnya, walaupun dalam keadaan jumlah ulangan dan perlakuan tidak sama.
  3. *Derajat bebas estimasi maksimum terdapat pada error*. Ini hanya berlaku untuk percobaan-percobaan kecil atau pengamatan dimana variasi luar besar.

Kelemahan dari RAL adalah relatif tidak efisien bila terdapat faktor dari luar yang tidak dapat dikendalikan.

**Perhitungan RAL**

Berikut adalah *layout* data pada RAL:

**Tabel 5.** Tabel data pengamatan RAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Perlakuan** | **Ulangan** | | | **Total** | **Rerata** | **Standar Deviasi** |
| **I** | **II** | **III** |
| A | Y11 | Y12 | Y13 | Y1. | Y1./n | S1. |
| B | Y21 | Y22 | Y23 | Y2. | Y2./n | S2. |
| C | Y31 | Y32 | Y33 | Y3. | Y3./n | S3. |
| Total | Y.1 | Y.2 | Y.3 | Y.. |  |  |

Perhitungan rancangan acak lengkap digambarkan dengan persamaan (1), (2), (3), dan (4) berikut:



Faktor koreksi (FK) = . (1)



Jumlah Kuadrat Total (JKT) = ……………..……………..…… (2)



Jumlah Kuadrat Perlakuan (JKP) = ….……………………..…… (3)

Jumlah Kuadrat Galat (JKG) = JKT – JKP…………………………………………… (4)

**Tabel 6.** Daftar analisis ragam menurut RAL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SK** | **Db** | **JK** | **KT** | **Fhitung** | **Ftabel** | |
| **5%** | **1%** |
| Perlakuan (p) | t-1 | JKP | JKP/(t-1) | KTP/KTG | F(0,05, dbp, dbg) | F(0,01, dbp, dbg) |
| Galat (g) | t(r-1) | JKG | JKG/t(r-1) | - | - | - |
| Total | tr-1 | JKT | - | - | - | - |

Kaidah keputusan:

1. Jika Fhitung ≤ Ftabel 5%, berarti perlakuan tidak berpengaruh nyata, dan ditandai dengan simbol **tn** (tidak nyata).
2. Jika Fhitung ≥ Ftabel 5%, dan Fhitung ≤ Ftabel 1%, berarti perlakukan berpengaruh nyata, dan ditandai dengan simbol asterik (**\***).
3. Jika Fhitung ≥ Ftabel 1%, berarti perlakuan berpengaruh sangat nyata, dan ditandai dengan simbol asterik ganda (**\*\***).

(Sumber: Kristilya)

1. **METODOLOGI PENELITIAN**
   1. **Waktu dan Tempat Penelitian**

Penelitian dilakukan di Fakultas Sains dan Teknologi, Uiversitas Jambi yang beralamat di Jl. Lintas Jambi – Muara Bulian Km. 15, Mendalo Indah, Jambi. Penelitian dilakukan dalam waktu **????** bulan, dimulai dari bulan Juli 2019 hingga **??????**.

* 1. **Bahan dan Alat Penelitian**

Alat yang dibutuhkan untuk penelitian ini adalah sebagai berikut:

1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah komputer dengan spesifikasi prosesor Intel i5, RAM 8 GB, dan SSD 240 GB, Arduino board, sensor karbon dioksida MG-811, sensor suhu DHT11, serta adapter Wi-Fi ESP8266.

1. Perangkat Lunak

Perangkat lunak yang digunakan adalah sistem operasi elementary OS 5, Google Chrome, Sublime Text 3, Ardunio IDE, dan XAMPP.

* 1. **Kerangka Kerja Penelitian**

Penelitian ini memiliki suatu perencanaan dan alur langkah kerja yang sistematis sebagai berikut:

Identifikasi masalah

Studi literatur

Pengumpulan data

Perancangan sistem

Implementasi sistem

Pengujian sistem

Pembuatan Laporan

**Gambar 6.** Kerangka Kerja Penelitian

* 1. Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah penelitian dan menentukan batasan masalah yang akan dibahas dalam penelitian.

* 1. Studi Literatur

Studi literatur adalah tahap pengumpulan informasi sebagai teori dasar dalam penelitian. Literatur yang dimaksud dapat bersumber dari jurnal, buku, artikel online, dan sumber lainnya yang berhubungan dengan penelitian ini.

* 1. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang berkaitan dengan peneltian. Untuk melakukan penelitian ini, diperlukan informasi mengenai dampak paparan karbon dioksida terhadap kesehatan manusia. Selain itu, diperlukan pula informasi mengenai aspek kenyamanan termal seseorang di dalam suatu ruangan. Informasi-informasi tersebut akan digunakan sebagai acuan untuk menentukan ruang kelas ideal bagi proses belajar mengajar.

* 1. Perancangan Sistem

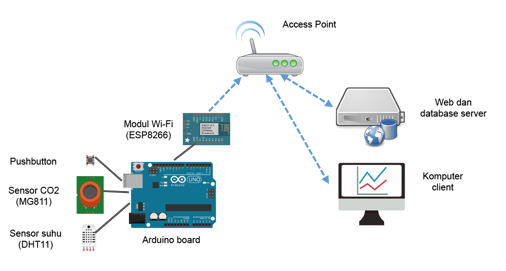
Pada tahap ini dilakukan analisis kebutuhan sistem yang akan dirancang. Kebutuhan yang dimaksud meliputi data-data yang digunakan, perancangan desain sistem menggunakan *use case, sequence diagram, dan class diagram*, serta perancangan desain antarmuka.

* 1. Implementasi Sistem

Pada tahap implementasi sistem, dilakukan pengembangan fitur-fitur sistem sesuai dengan rancangan yang telah dibuat sebelumnya. Rancangan yang dibuat akan diimplementasikan kedalam bentuk kode program. Pada penelitian ini, pengembangan sistem dilakukan dengan menggunakan *framework* Laravel, bahasa pemgrograman PHP, dan basis data MySQL.

* 1. Pengujian Sistem

Pengujian dilakukan untuk memastikan sistem yang dibuat sesuai dengan kebutuhan yang telah didefinisikan sebelumnya. Pada penelitian ini, pengujian dilakukan dengan mengaplikasikan sistem ke beberapa ruang kelas untuk mengukur konsentrasi karbon dioksida dan suhu pada ruang yang bersangkutan. Papan Arduino yang telah terpasang sensor konsentrasi karbon dioksida dan sensor suhu akan mengumpulkan data kualitas udara di suatu ruangan. Data sensor akan dikirim ke *web server* dengan bantuan modul Wi-Fi. Data yang telah masuk ke *web server* akan disimpan ke dalam *database*. Pengguna sistem yang ingin melihat data sensor dapat mengakses *web server* yang terhubung di jaringan. Pengguna akan disuguhkan data-data seperti konsentrasi karbon dioksida, suhu udara, dan banyaknya orang yang menempati suatu ruangan. Data yang masuk akan ditampilkan dalam bentuk grafik untuk memudahkan pembacaan data.



**Gambar 7.** Arsitektur alat dan program di jaringan

* 1. Pembuatan Laporan

Tahapan terakhir adalah pendokumentasian hasil temuan ke dalam bentuk laporan.

* 1. **Objek dan Variabel Penelitian**

Penelitian dilakukan di suatu ruang kelas yang berlokasi di Fakultas Sains dan Teknologi, Universitas Jambi, Jl. Lintas Jambi – Muara Bulian Km. 15, Mendalo Indah, Jambi. Responden yang dilibatkan dalam penelitian adalah 40 mahasiswa Universitas Jambi yang diminta untuk memasuki ruangan kelas saat dilakukan pengukuran konsentrasi karbon dioksida.

Hal yang diteliti dalam penelitian ini adalah pengaruh jumlah individu yang menempati ruang kelas terhadap konsentrasi karbon dioksida dan suhu dalam ruang tersebut. Faktor-faktor luar yang perlu diperhatikan dan dikendalikan saat pengambilan data diantaranya adalah ukuran ruangan, keberadaan AC dan kipas angin, serta ventilasi. Adapun spesifikasi ruangan yang digunakan adalah:

1. Ruangan memiliki ukuran lebar 7 meter dan panjang 10 meter**.**
2. Ruangan tidak memiliki AC dan kipas angin.
3. Pada saat pengambilan data, jendela dalam keadaan tertutup.

Pada penelitian ini, jumlah mahasiswa berperan sebagai variabel bebas dan data nilai konsentrasi karbon dioksida dan suhu berperan sebagai variabel tetap.

* 1. **Analisis Data**

Analisis data untuk menentukan apakah jumlah individu memengaruhi banyaknya jumlah konsentrasi karbon dioksida dan suhu pada ruangan dilakukan dengan menggunakan Rancangan Acak Lengkap (RAL). Data yang masuk dapat juga dijadikan gambaran untuk menentukan ruang kelas ideal ditinjau dari banyaknya konsentrasi karbon dioksida dengan membandingkan data yang diambil saat penelitian dengan data pada publikasi *online* oleh Winconsin Dept. of Health Services tentang pengaruh karbon dioksida terhadap kesehatan. Data suhu dapat dijadikan acuan untuk menggambarkan kondisi kelas yang nyaman ditinjau dari aspek kenyamanan termal dengan mengacu pada tulisan ilmiah yang dipublikasikan oleh Lippsmeier pada tahun 1994.

* 1. ***Timeline* Penelitian**

Timeline penelitian digunakan agar penelitian dapat berjalan dengan lebih terarah dan sesuai dengan tujuan penelitian. Penelitian berlangsung selama 18 minggu, dimiulai dari tahapan identifikasi masalah, studi literatur, pengumpulan data, perancangan, implementasi, pengujian, dan pembuatan laporan.

**Tabel 3**. *Timeline* Penelitian

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kegiatan | Durasi | Mulai (2019) | Selesai (2019) | Bulan | | | | |
| 7 | 8 | 9 | 10 | 11 |
| Identifikasi masalah | 1  Minggu | 15 Juli | 21 Juli |  |  |  |  |  |
| Studi literatur | 2  Minggu | 22 Juli | 4 Agustus |  |  |  |  |  |
| Pengumpulan data | 1  Minggu | 5 Agustus | 11 Agustus |  |  |  |  |  |
| Perancangan sistem | 6  Minggu | 12 Agustus | 22  September |  |  |  |  |  |
| Implementasi sistem | 4  Minggu | 23  September | 20 Oktober |  |  |  |  |  |
| Pengujian sistem | 2  Minggu | 21 Oktober | 3 November |  |  |  |  |  |
| Pembuatan laporan | 2  Minggu | 4 November | 15 November |  |  |  |  |  |

1. **HASIL DAN PEMBAHASAN**

Pada penelitian ini dilakukan perancangan sistem informasi monitoring kualitas udara menggunakan model *prototype* dengan 5 tahapan, diantaranya *communication*, *quick plan*, *modelling quick design*, *contruction of prototype*, dan *deployment, delivery, and feedback*.

Sistem yang dirancang digunakan untuk mengukur konsentrasi karbon dioksida dan temperatur udara pada ruang kelas. Data yang didapat akan diolah menggunakan rancangan percobaan Rancangan Acak Lengkap (RAL) untuk menentukan pengaruh konsentrasi karbon diokasida dan temperatur udara terhadap banyaknya jumlah individu yang menempati suatu ruangan. Hasil pengukuran tersebut nantinya dapat dijadikan acuan dalam menentukan ruang kelas ideal bagi proses belajar.

* 1. ***Communication***

Tahapan komunikasi merupakan tahapan awal pada model *prototype*. Proses komunikasi dilakukan dengan pihak *client* untuk mengetahui kebutuhan-kebutuhan serta alur sistem sesuai dengan permintaan *client*.

**Analisis Kebutuhan**

Analisis kebutuhan bertujuan untuk merumuskan, mengidentifikasi, serta menganalisis syarat-syarat yang dibutuhkan sistem untuk mendukung aktivitas pemantauan kualitas udara.

* + 1. **Kebutuhan Fungsional**

Adapun kebutuhan fungsional sistem ini adalah sebagai berikut:

* + 1. Sistem dapat menerima dan menyimpan data sensor yang dikirim melalui jaringan dengan protokol TCP/IP.
    2. Sensor diidentifikasi menggunakan kode unik yang secara otomatis di-*generate* saat sensor dibuat.
    3. Masing-masing sensor memiliki beberapa atribut diantaranya nama, satuan nilai, dan status.
    4. Sistem memberikan notifikasi ke pengguna saat status sensor berubah.
    5. Beberapa sensor dapat digabungkan ke dalam suatu grup yang disebut dengan *node*.
    6. Masing-masing *node* memiliki beberapa atribut diantaranya akses (publik atau privat) dan API *key*.
    7. Data sensor pada *node* dengan akses publik dapat dilihat oleh semua orang.
    8. Data sensor pada *node* dengan akses privat hanya dapat dilihat oleh pembuat *node*, dan hanya dapat diakses saat pengguna tersebut dalam keadaan *logged-in.*
    9. Saat mengirim data sensor, pengguna harus menyisipkan API *key* pada URL, dan API *key* tersebut harus sesuai dengan *node* tempat sensor itu berada.
    10. Data sensor ditampilkan dalam bentuk grafik garis (*line chart*).
    11. Pengguna dapat menghapus dan meng-*edit* *node* dan sensor.
    12. **Kebutuhan Nonfungsional**

Adapun kebutuhan nonfungsional sistem ini adalah sebagai berikut:

1. Sistem dapat memudahkan pengukuran kualitas udara.
2. Tampilan sistem user friendly sehingga mudah digunakan dan dipelajari.
3. Sistem dapat menyajikan informasi yang lengkap dan mudah dipahami.
   1. ***Quick Plan***

Tahapan selanjutnya adalah melakukan perencanaan untuk pengembangan sistem berdasarkan kebutuhan yang ditentukan sebelumnya. Langkah-langkan yang dilakukan meliputi penentuan teknologi, dan penentuan gambaran serta pengguna sistem.

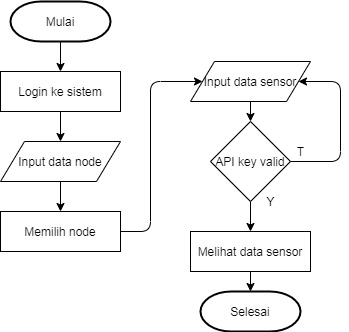
**Menentukan Teknologi**

Sistem yang dirancang menggunakan Diagram UML (*Unified Modelling Language*), untuk menggambarkan rancangan tersebut digunakan aplikasi draw.io. Pada pemograman, bahasa yang digunakan ialah PHP, dengan menggunakan framework Laravel. *Database* yang digunakan ialah MySQL. PhpMyAdmin digunakan untuk mempermudah pengelolaan *database*.

Untuk tampilan interface sistem digunakan teknologi HTML5 dengan bantuan framework Bootstrap agar disistem lebih interaktif dan responsif.

**Menentukan Gambaran Sistem**

Adapun usulan sistem yang akan dikembangankan digambarkan dengan *flowcart* berikut:



**Gambar 8.** Alur kerja sistem

**Menentukan Pengguna**

Pada sistem ini hanya terdapat satu jenis pengguna. Pengguna tersebut memiliki akses penuh atas semua *node* dan *sensor* yang dia miliki. Walaupun begitu, seorang pengguna tidak dapat mengakses atau mengubah data *node* atau sensor milik pengguna lain. Akses seorang pengguna terhadap data yang dimiliki oleh pengguna lain hanya sebatas pada melihat data sensor yang memiliki akses publik.

**Menentukan Komponen Sistem**

Menentukan komponen sistem bertujuan untuk mengetahui komponen-komponen yang dibutuhan sistem. Adapun komponen sistem antara lain:

* + - 1. Menentukan hak akses pengguna untuk mencegah akses dari pihak yang tidak memiliki hak untuk mengaksesnya.
      2. Proses terstruktur, yang meliputi:

1. Proses *login* dan *logout*.
2. Proses membuat dan mengelola *node*.
3. Proses membuat dan mengelola sensor.
4. Proses mengirimkan data sensor ke sistem.
5. Proses menampilkan data sensor.
   1. ***Modeling Quick Design***

Pada tahapan modeling quick design dilakukan beberapa kegiatan diantaranya pemodelan sistem menggunakan *use case diagram, activity diagram, sequence diagram*, perancangan basis data beserta *entity relationsip diagram*, perancangan struktur menu, serta perancangan *user interface.*

**Perancangan Use Case Diagram**

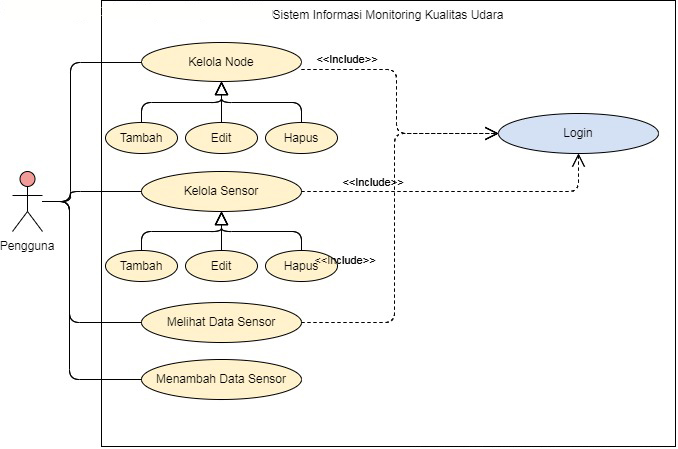
*Use case diagram* bertujuan untuk mengetahui interaksi pengguna (aktor) dengan sistem. Adapun kegiatan yang dilakukan dalam perancangan *use case diagram*, diantaranya:

1. **Identifikasi Aktor**

Pada sistem ini hanya terdapat satu jenis pengguna. Pengguna tersebut memiliki akses penuh atas semua *node* dan *sensor* yang dia miliki. Akses yang dimaksud seperti membuat dan mengelola *node* dan sensor, mengirim data sensor, serta melihat data sensor yang masuk.

1. **Rancangan Use Case Diagram**

Berikut adalah use case diagram yang menggambarkan interaksi aktor dengan sistem:

****

**Gambar 9.** *Use case diagram*

1. **Skenario Use Case**

Skenario use case bertujuan untuk menjelaskan interaksi antara sistem dengan pengguna. Adapun daftar use case pada penelitian ini adalah sebagai berikut:

**Tabel 9**. Daftar Use Case Diagram

|  |  |  |
| --- | --- | --- |
| ID Use Case | Nama *Use Case* | Deskripsi |
| UC1 | Login | *Use sase* menggambarkan kegiatan menginputkan email dan password untuk mengakses sistem. |
| UC2 | Kelola *Node* | *Use case* menggambarkan kegiatan menambah, meng-*edit*, dan menghapus *node*. |
| UC3 | Kelola Sensor | *Use case* menggambarkan kegiatan menambah, meng-*edit*, dan menghapus sensor. |
| UC4 | Melihat Data Sensor | *Use case* menggambarkan kegiatan mengakses data sensor yang telah masuk ke sistem dalam bentur grafik garis. |
| UC5 | Menambah Data Sensor | *Use case* menggambarkan kegiatan menambahkan data sensor ke dalam sistem. |
| UC6 | Logout | *Use case* menggambarkan kegiatan keluar dari sistem. |

Adapun skenario *Use case* dalam penelitian ini dapat dilhat pada tabel-tabel dibawah ini:

1. Login

Nama *Use Case* : Login

ID *Use Case* : UC1

Deskripsi : *Use sase* menggambarkan kegiatan menginputkan email dan password untuk mengakses sistem.

**Tabel 10.** Use Case Login

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Mengunjungi alamat *website* sistem | 1. Menampilkan halaman login |
| 1. Memasukan *email* dan *password* | 1. Menampilkan halaman daftar *node* |
| 1. Mengklik tombol login |  |
| **Skenario Alternatif** |  |
| 1. Memasukan *email* dan *password* 2. Mengklik tombol *login* | 1. Menampilkan notifikasi email atau password salah |
| 1. Memasukan email dan password 2. Mengklik *tombol login* | 1. Menampilkan halaman daftar *node* |

1. Kelola Node

Nama *Use Case* : Kelola Node

ID *Use Case* : UC2

Deskripsi : *Use case* menggambarkan kegiatan menambah, meng-*edit*, dan menghapus *node*.

**Tabel 11.** Use Case Menambahkan Node

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Mengklik ikon “+” pada halaman daftar *node* | 1. Menampilkan form data *node* |
| 1. Memasukan nama *node,* deskripsi, dan akses 2. Mengklik tombol simpan | 1. Menyimpan data *node* baru |

**Tabel 12.** Use Case Mengedit Node

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Memilih *node* pada halaman daftar *node* | 1. Menampilkan halaman data sensor |
| 1. Mengklik tab “Pengaturan” | 1. Menampilkan halaman pengaruan *node* |
| 1. Mengubah nama, deskripsi dan/atau akses *node* 2. Mengklik tombol simpan | 1. Menyimpan perubahan data *node* |

**Tabel 13.** Use Case Menghapus Node

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Memilih *node* pada halaman daftar *node* | 1. Menampilkan halaman data sensor |
| 1. Mengklik tab “Pengaturan” | 1. Menampilkan halaman pengaturan *node* |
| 1. Mengklik tombol “Hapus” pada bagian “Hapus *node*” | 1. Menampilkan pesan konfirmasi penghapusan |
| 1. Mengklik “OK” untuk mengkonfirmasi | 1. Menghapus *node* |

1. Kelola Sensor

Nama *Use Case* : Kelola Sensor

ID *Use Case* : UC3

Deskripsi : *Use case* menggambarkan kegiatan menambah, meng-*edit*, dan menghapus sensor.

**Tabel 14.** Use Case Menambah Sensor

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Memilih *node* pada halaman daftar *node* | 1. Menampilkan halaman data sensor |
| 1. Mengklik tab “Sensor” | 1. Menampilkan halaman daftar sensor |
| 1. Mengklik tombol “Tambah sensor” | 1. Menampilkan form data sensor |
| 1. Mengisi nama dan deskripsi sensor, serta satuan unit yang digunakan 2. Mengklik tombol simpan | 1. Menyimpan data sensor |

**Tabel 15.** Use Case Mengedit Sensor

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Memilih *node* pada halaman daftar *node* | 1. Menampilkan halaman data sensor |
| 1. Mengklik tab “Sensor” | 1. Menampilkan halaman daftar sensor |
| 1. Mengubah data pada sensor yang ingin diubah 2. Mengklik tombol “Update setting” | 1. Menyimpan perubahan pada data sensor |

**Tabel 16.** Use Case Menghapus Sensor

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Memilih *node* pada halaman daftar *node* | 1. Menampilkan halaman data sensor |
| 1. Mengklik tab “Sensor” | 1. Menampilkan halaman daftar sensor |
| 1. Mengklik tombol “Hapus sensor” | 1. Menampilkan pesan konfirmasi penghapusan |
| 1. Mengklik tombol “OK” untuk mengkonfirmasi | 1. Menghapus sensor beserta datanya |

**Tabel 17.** Use Case Mengatur Status Sensor

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Memilih *node* pada halaman daftar *node* | 1. Menampilkan halaman data sensor |
| 1. Mengklik tab “Sensor” | 1. Menampilkan halaman daftar sensor |
| 1. Mengubah pengaturan notifikasi menjadi aktif | 1. Menampilkan *input box* untuk mengisi data status sensor |
| 1. Mengisi nama status, batas nilai, dan warna status 2. Mengklik tombol “Update setting” | 1. Menyimpan data sensor |

1. Melihat Data Sensor

Nama *Use Case* : Kelola Melihat Data Sensor

ID *Use Case* : UC4

Deskripsi : *Use case* menggambarkan kegiatan mengakses data sensor yang telah masuk ke sistem dalam bentur grafik garis.

**Tabel 18.** Use Case Melihat Data Sensor

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Memilih *node* pada halaman daftar *node* | 1. Menampilkan halaman data sensor |
| 1. Memilih rentang waktu pada bagian “filter” | 1. Menampilkan data yang masuk hanya pada rentang waktu yang ditentukan |
| 1. Memilih jumlah data yang ditampilkan pada bagian “filter” | 1. Menampilkan data terakhir yang masuk sebanyak jumlah yang ditentukan |

1. Menambah Data Sensor

Nama *Use Case* : Menambah Data Sensor

ID *Use Case* : UC5

Deskripsi : *Use case* menggambarkan kegiatan menambahkan data sensor ke dalam sistem.

**Tabel 19.** Use Case Menambah Data Sensor

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Mengakses url dengan menyisipkan kode sensor, nilai sensor, dan API key sebagai parameter | 1. Menampilkan pesan “OK” |
| **Skenario Alternatif** |  |
| 1. Mengakses url dengan menyisipkan kode sensor, nilai sensor, dan API key sebagai parameter | 1. Menampilkan pesan “Invalid key” |
| 1. Mengubah API key dengan API key yang valid | 1. Menampilkan pesan “OK” |

1. Logout

Nama *Use Case* : Logout

ID *Use Case* : UC6

Deskripsi : *Use case* menggambarkan kegiatan keluar dari sistem.

**Tabel 19.** Use Case Menambah Data Sensor

|  |  |
| --- | --- |
| **Aksi Aktor** | **Respon Sistem** |
| 1. Mengklik ikon panah kebawah pada *navigation bar* | 1. Menampilkan tombol “Logout” |
| 1. Mengklik tombol “logout” | 1. Mengeluarkan pengguna dari sistem dan mengarahkan ke halaman *login* |

**Perancangan Diagram Aktivitas**