BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi yang sangat pesat membuat pemerintah tidak cukup hanya menjadi regulator namun juga perlu menjadi fasilitator ataupun akselerator di era industri 4.0 ini. Salah satunya adalah pengembangan ekosistem *Internet of Things* (IoT).

Internet of Things (IoT) adalah sebuah gagasan dimana semua benda di dunia nyata dapat berkomunikasi satu dengan yang lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan *internet* sebagai penghubung. Dengan adanya konsep *Internet of Things* (IoT) ini maka dapat membantu meringankan aktivitas sehari-hari manusia. Karena *Internet of Things* (IoT) ini menghubungkan suatu alat, instansi atau yang lainya menjadi saling terhubung satu sama lain melalui koneksi jaringan *internet*. Selain itu dengan adanya konsep *Internet of Things* (IoT) ini maka dapat mewujudkan suatu konsep *smart city* yang ideal, karena *Internet of Things* (IoT) sendiri mengintegrasikan berbagai macam fasilitas pemerintah di suatu kota/kabupaten tersebut menjadi satu sehingga memudahkan Aparatur Sipil Negara dalam menjalankan roda pemerintahan.

Pengelolaan sumber daya air sangat berkaitan dengan lingkungan dan dampaknya terhadap tata kehidupan masyarakat di Indonesia, dimana telah berlangsung dalam kecepatan yang melampaui kemampuan untuk mencegah dan mengendalikan degradasi lingkungan. Kondisi ini perlu mendapat perhatian serius

dalam pengelolaan dan penataan kekayaan alam secara berimbang sehingga memberikan manfaat bagi manusia

Sistem irigasi yang terdapat di indonesia kebanyakan masih memakai sistem manual, yaitu dimana membuka dan menutup saluran irigasi ke sawah masih tradisional atau bantuan manusia. Dimana petani merasa kesulitan untuk mengatur irigasi pada sawah yang jauh dari rumah. Dan kurangnya petugas sekarang yang mau menjadi pengontrol pintu irigasi sawah (Ulu-ulu). Jadi sering kali petani pulang pergi dari rumah ke sawah hanya untuk mengecek saluran irigasi sehingga mengurangi efektifitas petani. Kelebihan penelitian ini dari penelitian sebelumnya yaitu pengontrolan sistem irigasi sawah dapat dilakukan dimanapun dan kapanpun ketika kita terkoneksi dengan jaringan *internet* dengan menggunakan aplikasi *android*.

Untuk mengatasi masalah tersebut, diperlukan sebuah sebuah sistem baru dan otomatis untuk membantu meringankan pekerjaan petani. Dengan memanfaatkan *handphone* khususnya pada fasilitas *internet*, maka penerapan *internet* sebagai sarana untuk kendali jarak jauh. Sistem ini akan mempunyai akses untuk membuka dan menutup portal saluran irigasi yang telah dibuat dan cara mengendalikan sistem tersebut dari jarak jauh melalui jaringan *internet* yang sekarang ini dinamakan *Internet of Things (IoT),* sehingga dapat memudahkan para petani dalam mengontrol sistem irigasi sawah.

Dengan memperhatikan potensi-potensi sumber daya air, lahan sawah serta sumber daya petani, maka Kabupaten Sidenreng Rappang sebagai penghasil utama pangan beras di Indonesia bagian timur. Hal ini terutama didukung oleh

jaringan irigasi teknis yang mampu mengairi sawah sepanjang tahun. Dan menunjang program Ketahanan Kedaulatan Pangan Nasional. Kabupaten Sidenreng Rappang mempunyai iklim yang dipengaruhi 2 (dua) musim yaitu musim hujan dan musim kemarau, yang ada pada pantai barat dan pantai timur Provinsi Sulawesi Selatan.

Selanjutnya dengan memanfaatkan peluang-peluang dan kekuatan-kekuatan yang ada serta kelemahan-kelemahan dan menghindari ancaman yang timbul maka pengelolaan sumber daya air harus dilakukan dengan efektif dan efisien sehingga dapat memberikan produktivitas yang maksimal.

Berdasarkan uraian dan permasalahan diatas maka penulis merancang sistem kontrol untuk memberikan kemudahan bagi masyarakat, selama pemrograman ini berjalan baik serta sistem kontrol yang di buat ini semoga dapat lebih interaktif, dengan ini penulis membuat tugas akhir yang berjudul "**Kontrol Jarak Jauh Sistem Irigasi Sawah Berbasis *Internet of Things* (IoT)".**

B. Rumusan Masalah

Adapun yang menjadi rumusan masalah dalam penelitian ini adalah :

Bagaimana mengendalikan Pintu Air Irigasi sawah berbasis *Internet of Things* (IoT).

C. Batasan Masalah

Adapun yang menjadi batasan masalah dalam penelitian ini adalah :

Sistem yang akan di buat berbentuk *prototype* dan Aplikasi monitoring untuk mengontrol jarak jauh pintu irigasi sawah.

D. Tujuan Penelitian

Adapun tujuan masalah penerapan konsep *Internet of Things* (IoT) ini adalah terwujudnya sebuah *prototype*, yang nantinya dapat mengontrol Pintu Air Bangunan Irigasi.

1. Membuat sistem kontrol irigasi berbasis *Internet of Things* (IoT)
2. Mengetahui dan memahami *Internet of Things* (IoT) pada *prototype* kontrol Pintu Air Bangunan Irigasi

E. Manfaat Penelitian

1. Manfaat bagi penulis.

   Agar penulis dapat mengaplikasikan ilmu dan teori yang didapat di bangku perkuliahan, serta dapat menambah pengetahuan tentang *Internet of Things* (IoT) secara mendalam.
2. Manfaat bagi petugas

   Diharapkan dapat menjadi alternatif bagi petugas mengontrol pintu irigasi pada sawah yang jauh dari rumah.

F. Sistematika Penulisan

Adapun sistematika penulisan pada Tugas Akhir ini yaitu:

BAB I PENDAHULUAN

Berisi tentang uraian latar belakang pemilihan judul Kontrol Jarak Jauh Sistem Irigasi Sawah Berbasis *Internet Of Things* (IoT), rumusan masalah, batasan masalah, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Dalam bab ini berisikan beberapa teori-teori yang mendukung dalam pembahasan penyelesaian tugas akhir ini serta bahasa pemrograman yang digunakan sehingga memudahkan penulis dalam menyelesaikan masalah.

BAB III METODE PENELITIAN

Pada bab ini menjelaskan mengenai tahapan-tahapan yang dilalui dalam penyelesaian tugas akhir ini, yaitu lokasi penelitian dan waktu penelitian, jenis penelitian, metode pengumpulan data, alat dan bahan penelitian, metode pengujian serta gambaran desain sistem yang akan dirancang atau dibuat.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan mengenai analisis sistem yang berjalan, analisis sistem yang diusulkan, desain sistem, rancangan perangkat keras, Implementasi perangkat keras, rancangan perangkat lunak, rancangan aplikasi *web*, dan mengenai metode pengujian dan teknik pengujian.

BAB V PENUTUP

Pada bab ini menjelaskan mengenai kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Sebelumnya

Pengertian penelitian adalah suatu proses penyelidikan yang dilakukan secara aktif, tekun, dan sistematis, dimana tujuannya untuk menemukan, menginterpretasikan, dan merevisi fakta-fakta.

Penelitian sebelumnya :

1. Judul "Rancang bangun sistem irigasi tanaman otomatis menggunakan *wireless sensor network*" (Institut Teknologi Sepuluh September, 12 September 2015), sistem yang digunakan adalah *wireless sensor network* dalam memantau kondisi suatu lahan yang diimplementasikan kedalam sistem irigasi otomatis, sistem ini juga diberikan *sensor Soil Moisture* untuk mengukur kelembaban tanah, *Transceiver* dan *receiver* untuk komunikasi data, *android* untuk memproses data serta pintu air dan *Fuzzy Logic* sebagai *controller*. (Muhammad rifai dan Suwito)
2. Judul "Rancang bangun sistem otomatisasi irigasi pipa lahan sawah berbasis tenaga surya" (program studi teknik sipil dan Lingkungan SPs Institut Pertanian Bogor, 14 April 2015), sistem yang digunakan adalah sistem kontrol otomatis pada permukaan air lahan sawah pada level sesuai kebutuhan tanaman. Sistem kontrol otomatis ini dibangun dengan memanfaatkan teknologi digital, *mikrokontroler* dan jaringan *sensor*. *Mikrokontroler Arduino Uno ATMega328p* digunakan sebagai sistem kendali

otomatis untuk menggerakkan sistem aktuasi kran air *elektris Valworx 561086* berdasarkan kelembaban tanah dan tinggi muka air di lahan sawah yang dideteksi oleh sensor. (Sudirman Siraid)

3. Judul "Monitoring dan kontrol sistem irigasi berbasis IoT menggunakan banana Pi" (Departemen Teknik Elektro, Institut Teknologi Sepuluh November, 2018), sistem ini menggunakan sensor kelembaban tanah atau *hygrometer (SBC) Banana Pi. SBC* yang terhubung dengan *internet* akan mengirimkan data yang didapat dari sensor basis data *firebase*, kemudian petani yang menggunakan aplikasi pada ponsel *android* dapat mengakses data sensor pada basis data tersebut, digunakan aplikasi pada ponsel *android* untuk memudahkan petani melihat kondisi tanah. Selain itu petani dapat melakukan kontrol jarak jauh untuk mengatur debit air yang disiram ke tanah. (Andrie Wijaya dan Muhammad Rivai)

B. Konsep Dasar Sistem

a. Pengertian Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Jogiyanto, 2015). Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata, seperti tempat, benda dan orang-orang yang betul-betul ada dan terjadi.

Sistem dapat didefinisikan sekumpulan elemen yang saling terkait dan terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat

dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.

b. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat - sifat yang tertentu yaitu :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi atau saling bekerjasama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem yang bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Keluaran *(output)* dari satu subsistem akan menjadi masukan *(input)* untuk subsistem yang lainnya dengan melalui penghubung.

5. Masukan Sistem

   Masukan *(input)* adalah energi yang dimasukkan kedalam sistem, dapat berupa masukan perawatan *(Maintenance Input)* dan masukan sinyal *(Signal Input).*

6. Keluaran Sistem

   Keluaran *(output)* adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

   Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

   Suatu sistem mempunyai tujuan atau sasaran, sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. (Kadir Abdul.2015).

C. Sistem Monitoring

Sistem adalah sekumpulan elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan dan merupakan sekumpulan komponen yang saling bekerjasama untuk mencapai tujuan guna memperbaiki organisasi kearah yang lebih baik.

Banyak definisi dalam menjelaskan pengertian sistem. Namun demikian, pada umumnya definisi itu menggambarkan bahwa pengertian sistem mengandung dua konotasi, yaitu benda atau entitas, dan proses atau metode.

Schrode dan Voich (1974) dalam bukunya yang berjudul *organization and management: basic systems concept* misalnya, menyatakan bahwa sistem adalah "*whole compounded of several parts"* (suatu kesatuan yang tersusun dari sejumlah elemen.

Monitoring adalah proses pengumpulan informasi mengenai apa yang sebenarnya terjadi selama proses implementasi atau penerapan program. Monitoring adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objek program/memantau perubahan yang fokus pada proses dan keluaran.

Menurut Edi Suharto monitoring pada dasarnya adalah pemantauan suatu kegiatan proyek atau program sosial yang dilaksanakan pada saat kegiatan tersebut sedang berlangsung.

Menurut Nalahudin Muhlisin monitoring merupakan suatu proses pengumpulan dan menganalisis informasi dari penerapan suatu program termasuk mengecek secara regular untuk melihat apakah kegiatan atau program itu berjalan sesuai dengan rencana sehingga masalah yang dilihat atau ditemui dapat diatasi. Monitoring menyediakan data dasar untuk menjawab permasalahan.

Dari beberapa pendapat diatas dapat dipahami pengertian monitoring adalah pemantauan suatu program yang dilaksanakan saat kegiatan tersebut sedang berlangsung, monitoring menyediakan data dasar untuk menjawab permasalahan.

Sedangkan yang dimaksud dengan system monitoring adalah layanan yang melakukan proses pengumpulan data dan melakukan analisis terhadap data-data tersebut dengan tujuan untuk memaksimalkan seluruh sumber daya yang dimiliki.

D. Pengertian Perancangan

Menurut Ahmad Shukri Mohd (2015:5) perancangan sistem adalah sebuah teknik pemecahan masalah yang saling melengkapi (dengan analisis sistem) yang merangkai kembali bagian-bagian komponen yang menjadi sistem yang lengkap. Hal ini melibatkan penambahan, penghapusan dan perubahan-perubahan bagian relative pada sistem awal (aslinya) yang tidak didukung oleh ketiga pilar ini tidak dapat dikatakan sebagai informasi yang berguna, tetapi merupakan sampah (*garbage*).

Sistem informasi adalah cara-cara yang terorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelolah, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan. (Krismaji, 2015).

E. *Internet of Things* (IoT)

Perkembangan teknologi semakin pesat dari waktu ke waktu. Mulai dari mobil pintar *(smart car)* yang bisa berjalan sendiri ke berbagai tujuan tanpa pengemudi manusia, hingga perangkat rumah pintar *(smart home)* semacam Alexa yang bisa otomatis bersuara mengingatkan untuk melakukan aktivitas sesuai jadwal. Seluruh teknologi terbaru ini adalah bagian dari *Internet of Things. Internet of Things (IoT)* adalah sebuah konsep di mana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan adanya interaksi dari manusia ke manusia atau dari manusia ke komputer. *Internet of*

Things (IoT) adalah struktur di mana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer. (Burange & Misalkar, 2015).

IoT sudah berkembang pesat mulai dari penggabungan teknologi nirkabel, *Micro Electromechanical Systems (MEMS)* dan juga Internet. *Internet of Things (IoT)* menggunakan beberapa teknologi yang secara garis besar digabungkan menjadi satu kesatuan diantaranya sensor sebagai pembaca data, koneksi *internet* dengan beberapa macam topologi jaringan, Radio *Frequency Identification (RFID), wireless sensor network* dan teknologi yang terus akan bertambah sesuai dengan kebutuhan. *Internet of Things (IoT)* juga bisa mencakup teknologi-teknologi sensor lainnya, seperti teknologi nirkabel maupun kode QR yang sering kita temukan di sekitar kita, contoh penerapannya dalam benda yang ada di dunia nyata adalah untuk pengolahan bahan pangan, elektronik, dan berbagai mesin atau teknologi lainnya yang semuanya tersambung ke jaringan lokal maupun global lewat sensor yang tertanam dan selalu menyala aktif. *Internet of Things (IoT)* ini mengacu pada mesin atau alat yang bisa diidentifikasikan sebagai representasi virtual dalam strukturnya yang berbasis *Internet*.

Tantangan terbesar yang bisa menjadi hambatan dalam mengkonfigurasi *Internet of Things (IoT)* adalah menjembatani kesenjangan antara dunia fisik dan dunia informasi dan bagaimana menyusun jaringan komunikasinya, dikarenakan jaringan yang dibutuhkan oleh IoT sangatlah kompleks. Selain itu, *Internet of Things (IoT)* juga memerlukan suatu sistem keamanan yang cukup ketat. Di

samping masalah tersebut, biaya pengembangan *Internet of Things (IoT)* yang mahal juga sering menjadi faktor penyebab kegagalan, sehingga pembuatan dan pengembangannya bisa berakhir gagal produksi berakhir gagal produksi.



Gambar. 2.I. Cara kerja Internet of Things (IoT)
(Sumber: www.dewaweb.com)

Cara kerja *Internet of Things (IoT)*, dengan memanfaatkan suatu argumentasi *pemrograman*, di mana tiap - tiap perintah argumen tersebut dapat menghasilkan suatu interaksi antar mesin yang telah terhubung secara otomatis tanpa campur tangan manusia dan tanpa dibatasi oleh jarak yang jauh. *Internet* menjadi penghubung antara kedua interaksi mesin tersebut. Manusia dalam *Internet of Things (IoT)* tugasnya hanyalah menjadi pengatur dan pengawas dari mesin-mesin yang bekerja secara langsung tersebut.

Unsur-unsur pembentuk IoT yang mendasar adalah:

1. Kecerdasan Buatan *(Artificial Intelligence*/AI), *Internet of Things (IoT)* membuat hampir semua mesin yang ada menjadi *"Smart"* (pintar). Ini berarti *Internet of Things (IoT)* bisa meningkatkan segala aspek kehidupan

kita dengan pengembangan teknologi yang didasarkan pada AI. Pengembangan teknologi yang ada dilakukan dengan pengumpulan data, algoritma kecerdasan buatan, dan jaringan yang tersedia. Contohnya sederhana seperti meningkatkan atau mengembangkan perangkat lemari es/kulkas sehingga dapat mendeteksi jika stok susu dan sereal sudah hampir habis, bahkan bisa juga membuat pesanan ke supermarket secara otomatis jika stok akan habis.

2. Konektivitas dalam *Internet of Things (IoT)*, ada kemungkinan untuk membuat atau membuka jaringan baru, dan jaringan khusus IoT. Jaringan ini tidak lagi terikat hanya dengan penyedia utamanya saja. Jaringannya tidak harus berskala besar dan mahal, bisa tersedia pada skala yang jauh lebih kecil dan lebih murah. *Internet of Things (IoT)* bisa menciptakan jaringan kecil di antara perangkat sistem.
3. Sensor merupakan pembeda yang membuat *Internet of Things (IoT)* unik dibanding mesin canggih lainnya. Sensor ini mampu mendefinisikan instrumen, yang mengubah *Internet of Things (IoT)* dari jaringan standar dan cenderung pasif dalam perangkat, sehingga menjadi suatu sistem aktif yang dapat diintegrasikan ke dunia nyata dalam kehidupan sehari-hari.
4. Keterlibatan Aktif *(Active Engagement), Internet of Things (IoT)* mengenalkan paradigma yang baru bagi konten aktif, produk, maupun keterlibatan layanan.

5. Perangkat Berukuran Kecil. *Internet of Things (IoT)* memanfaatkan perangkat-perangkat kecil yang dibuat khusus agar menghasilkan ketepatan, skalabilitas, dan fleksibilitas yang baik.



Gambar. 2.2. Penerapan Internet of Things (IoT)
(Sumber: www.dewaweb.com)

Berbagai macam penerapan *Internet of Things (IoT)* dalam beberapa sektor antara lain:

1. Pertanian

   Beberapa diantaranya seperti mengumpulkan data soal suhu, curah hujan, kelembaban, kecepatan angin, serangan hama, dan muatan tanah. Data-data tersebut bisa dipakai untuk mengotomatisasi teknik pertanian dan dapat dipakai untuk mengambil keputusan *(decision making)* berdasarkan informasi yang diperoleh sehingga dapat meningkatkan kualitas dan kuantitas, meminimalkan risiko dan limbah, serta mengurangi upaya yang diperlukan dalam mengelola tanaman. Sebagai contoh, petani sekarang sudah bisa memantau suhu dan kelembaban tanah dari jauh, dan bahkan menerapkan

data yang diperoleh *Internet of Things (IoT)* untuk program pemupukan yang lebih merata.

2. Energi

   Sejumlah perangkat yang memakan energi besar saat ini sudah bisa terintegrasi dengan konektivitas *internet* yang memungkinkan mesin-mesin maupun jaringan untuk berkomunikasi dalam menyeimbangkan pembangkitan listrik serta penggunaan energi yang lebih hemat dan efektif. Perangkat ini juga bisa memungkinkan akses *remote control* dari pengguna. Selain itu, bisa juga mengaktifkan fungsi semacam penjadwalan (misalnya untuk menyalakan/mematikan mesin pemanas, mengendalikan oven, mengubah kondisi pencahayaan dari terang menjadi redup hingga ke gelap, dan lain sebagainya).

3. Lingkungan

   Aplikasi pemantauan lingkungan dari *Internet of Things (IoT)* biasanya menggunakan sensor dalam membantu terwujudnya perlindungan lingkungan. Contoh penerapannya seperti pemantauan kualitas udara atau air, kondisi atmosfer atau tanah, dan dapat mencakup pemantauan terhadap satwa liar dan habitatnya. *Internet of Things (IoT)* juga dimanfaatkan dalam penanggulangan bencana seperti sistem peringatan dini Tsunami atau gempa bumi. Perangkat *Internet of Things (IoT)* dalam hal ini memiliki jangkauan geografis yang sangat luas serta mampu bergerak.

4. Otomatisasi Rumah

Perangkat *Internet of Things (IoT)* dimanfaatkan untuk memantau dan mengontrol sistem mekanis dan elektronik yang digunakan pada berbagai jenis bangunan, seperti industri atau perumahan. Selain itu juga dapat mengontrol penggunaan energi secara *realtime* dalam mengurangi konsumsi energi. Perangkat *Internet of Things (IoT)* dapat terintegrasi menjadi sistem rumah pintar. Contoh penerapannya seperti: lampu menyala secara otomatis ketika malam hari, kemudian lampu akan mati secara otomatis pada saat jadwal tidur. Pagi hari, taman Anda akan disiram air oleh mesin penyiram otomatis. Begitu juga dengan kulkas Anda yang bisa memesan stok makanan sendiri ketika habis.

5. Medik dan Kesehatan

Dalam dunia medik dan kesehatan, Perangkat *Internet of Things (IoT)* akan merekam data-data kesehatan pasien dan ditransfer langsung ke tenaga medis maupun Rumah Sakit. Data-data yang bisa dideteksi dan dikirimkan seperti detak jantung, tingkat gula dalam darah, dan lain sebagainya. *Smartphone* atau ponsel akan jadi alat pemantau kesehatan yang canggih dan tentunya bisa sangat membantu. Salah satu contoh penerapannya seperti, tempat tidur pintar yang bisa otomatis memberitahukan dokter atau perawat ketika pasien hendak bangun dari tempat tidur.

6. Transportasi

Penerapan *Internet of Things (IoT)* dapat membantu manusia dalam berkomunikasi, mengontrol, dan pemrosesan informasi pada berbagai aspek

sistem transportasi. Interaksi dinamis yang terjadi antara komponen-komponen itu berasal dari sebuah sistem transportasi. Sistem tersebut memungkinkan komunikasi antara kendaraan dengan kontrol lalu lintas yang lebih efektif, parkir yang lebih cerdas, manajemen logistik dan armada, kontrol kendaraan, dan juga terkait faktor keselamatan maupun bantuan di jalan raya.

F. Irigasi Sawah

Lahan sawah adalah suatu tipe penggunaan lahan, yang untuk pengelolaannya memerlukan genangan air. Oleh karena itu sawah selalu mempunyai permukaan datar atau yang didatarkan (dibuat teras), dan dibatasi oleh pematang untuk menahan air genangan. Berdasarkan sumber air yang digunakan dan keadaan genangannya, sawah dapat dibedakan menjadi sawah irigasi, sawah tadah hujan, sawah lebak, dan sawah pasang surut.

Sawah irigasi adalah sawah yang sumber airnya berasal dari tempat lain melalui saluran-saluran yang sengaja dibuat untuk itu. Sawah irigasi dibedakan atas sawah irigasi teknis, sawah irigasi setengah (semi) teknis, dan sawah irigasi sederhana. Sawah irigasi teknis air pengairannya berasal dari waduk, dam atau danau dan dialirkan melalui saluran induk (*primer*) yang selanjutnya dibagi-bagi ke dalam saluran-saluran sekunder dan tersier melalui bangunan pintu-pintu pembagi. Sawah irigasi sebagian besar dapat ditanami padi dua kali atau lebih setahun, tetapi sebagian ada yang hanya dapat ditanami padi sekali setahun bila ketersediaan air tidak mencukupi terutama yang terletak di ujung-ujung saluran

primer dan jauh dari sumber airnya. Sawah irigasi teknis dan setengah teknis dibedakan berdasarkan sistem pengelolaan jaringan irigasinya. Irigasi teknis seluruh jaringan irigasi dikuasai dan dipelihara oleh pemerintah, sedangkan irigasi setengah teknis pemerintah hanya menguasai bangunan penyadap untuk dapat mengatur dan mengukur pemasukan air. Irigasi sederhana adalah pengairan yang sumber airnya dari tempat lain (umumnya berupa mata air) dan salurannya dibuat secara sederhana oleh masyarakat petani setempat, tanpa bangunan-bangunan permanen.

Sawah tadah hujan adalah sawah yang sumber airnya tergantung atau berasal dari curah hujan tanpa adanya bangunan-bangunan irigasi permanen. Sawah tadah hujan umumnya terdapat pada wilayah yang posisinya lebih tinggi dari sawah irigasi atau sawah lainnya sehingga tidak memungkinkan terjangkau oleh pengairan. Waktu tanam padi akan sangat tergantung pada datangnya musim hujan.

Sawah pasang surut adalah sawah yang irigasinya tergantung pada gerakan pasang dan surut serta letaknya di wilayah datar tidak jauh dari laut. Sumber air sawah pasang surut adalah air tawar sungai yang karena adanya pengaruh pasang dan surut air laut dimanfaatkan untuk mengairi melalui saluran irigasi dan drainase. Sawah pasang surut umumnya terdapat di sekitar jalur aliran sungai besar yang terkena pengaruh pasang surut air laut. Pada lahan pasang surut dibedakan empat tipologi lahan berdasarkan jangkauan luapan air pasang, yaitu tipe luapan A, B, C dan. Tipe luapan A dan B mempunyai potensi untuk persawahan karena dapat terjangkau air pasang dan biasanya terdapat lebih dekat

ke pantai, namun mempunyai kendala potensi kemasaman tanah atau salinitas tinggi. Sedangkan tipe luapan C dan D karena posisinya lebih tinggi dan jangkauan air pasang lebih terbatas, sehingga potensinya lebih sesuai untuk tegalan atau tanaman tahunan.

Sawah lebak adalah sawah yang diusahakan di daerah rawa dengan memanfaatkan naik turunnya permukaan air rawa secara alami, sehingga di dalam sistem sawah lebak tidak dijumpai sistem saluran air. Sawah ini umumnya terdapat di daerah yang relatif dekat dengan jalur aliran sungai besar (permanen) yaitu di backswamp atau rawa belakang dengan bentuk wilayah datar agak cekung, kondisi drainase terhambat sampai sangat terhambat, permukaan air tanah dangkal bahkan hingga tergenang dimusim penghujan, selalu terkena luapan banjir atau kebanjiran dari sungai di dekatnya selama jangka waktu tertentu dalam satu tahun. Oleh karena itu sawah ini baru dapat ditanami padi setelah air genangan menjadi dangkal (surut), dan terjadi umumnya pada musim kemarau. Lahan lebak demikian digolongkan sebagai lebak dalam jika terletak di sebelah dalam, topografi cekung, tergenang relatif dalam dan terus-menerus, sedangkan lebak tengahan pada transisi antara lebak dalam dan lebak pematang. Di daerah lebak dangkal atau tergolong sebagai lebak pematang umumnya dapat ditanami padi dua kali, sedangkan di lebak dalam hanya ditanami padi sekali setahun.

G. Mikrokontroler

Mikrokontroler adalah mikroprosesor yang dikhususkan untuk instrumentasi dan kendali. Mikrokontroler merupakan suatu alat elektronika

digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus (Zarkasyi, 2013).

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu serpih (*chip*) yang biasanya digunakan untuk sebuah *embedded system* (sistem yang dibentuk guna menjalankan satu atau lebih dari suatu fungsi tertentu secara *real time*). Mikrokontroler biasanya berukuran kecil karena didesain hanya untuk satu fungsi tertentu pada suatu sistem. Pemanfaatan mikrokontroler umumnya digunakan di bidang kendali dan instrumentasi elektronik. Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory),* RAM (*Read-Write Memory*), beberapa Port masukan maupun keluaran, dan beberapa peripheral seperti pencacah/pewaktu, ADC *(Analog to Digital converter*), DAC (*Digital to Analog converter*) dan *serial* komunikasi.

Setiap mikrokontroler memiliki arsitektur yang berbeda - beda tergantung perancangannya. Meskipun demikian, setiap arsitektur mikrokontroler pada dasarnya memiliki keseragaman pada pokok-pokok dan cara kerjanya. Berdasarkan arsitektur, mikrokontroler terbagi dua yaitu:

1. CISC (*Complex Instruction Set Computing)*

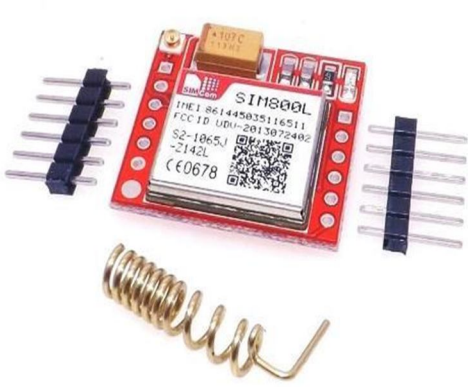
   Complex Instruction Set Computing (CISC) atau kumpulan instruksi komputasi kompleks. Adalah suatu arsitektur komputer di mana setiap instruksi akan menjalankan beberapa operasi tingkat rendah, seperti pengambilan dari memori (*load*), operasi aritmatika, dan penyimpanan ke dalam memori (*store*) yang saling bekerja sama.

2. RISC *(Reduced Instruction Set Computer)*

Reduced Instruction Set Computer (RISC). Merupakan bagian dari arsitektur mikroprosessor, berbentuk kecil dan berfungsi untuk negeset instruksi dalam komunikasi diantara arsitektur yang lainnya (Setiawan, 2006).

H. Modul Gsm Sim 800L

Modul SIM800L merupakan salah satu jenis modul GSM/GPRS serial yang paling popular digunakan oleh para penghobi, maupun professional elektronika untuk berbagai keperluan pengendalian jarak jauh. Untuk saat ini, terdapat beberapa tipe dari *Breakout Board*, tetapi yang paling popular yaitu versi mini dengan kartu GSM jenis Micro SIM Modul SIM800L. Modul ini dapat mengirim sms, menerima sms, melakukan panggilan, serta komunikasi paket data *internet*.



Gambar 2. 3 Modul GSM SIM 800l
Sumber (www.elektronika-id.com)

Keterangan PinOut :

1. ANT : Antena
2. VCC : tegangan masukan 3.7 – 4.2Vdc

3. RST : *Reset*
4. RX : Rx Data *Serial*
5. TX : Tx Data *Serial*
6. GND : *Ground*
7. RING : ketika ada telepon masuk
8. DTR
9. MIC + : ke *microphone* kutub +
10. MIC – : ke *microphone* kutub –
11. *Speaker* + : ke *speaker* atau *amplifier* kutub +
12. *Speaker* – : ke *speaker* atau *amplifier* kutub –
13. *Micro Sim* (Kartu GSM)

Spesifikasi modul SIM800L :

1. Operasi tegangan: 3.7 ~ 4.2V.
2. Bekerja pada *frequency* jaringan GSM yaitu *Quad Band* (850/900/1800/1900Mhz)
3. GPRS *multi-slot class* 1~12 (*option*) tetapi *default* pada *class* 12
4. Suhu pengoperasian normal : 40°C ~ 85°C
5. Menggunakan *port* TTL serial *port*, sehingga dapat langsung diakses menggunakan *microcontroller* tanpa perlu memerlukan MAX232
6. *Transmitting power*
7. *Power module automatically boot, homing network*

8. Terdapat *Led* pada modul yang berfungsi sebagai indikator. Apabila pada *module* terhubung dengan jaringan GSM maka LED akan berkedip perlahan, akan tetapi apabila tidak ada sinyal maka LED akan berkedip cepat.
9. Ukuran *module* : 2.5cm x 2.3cm

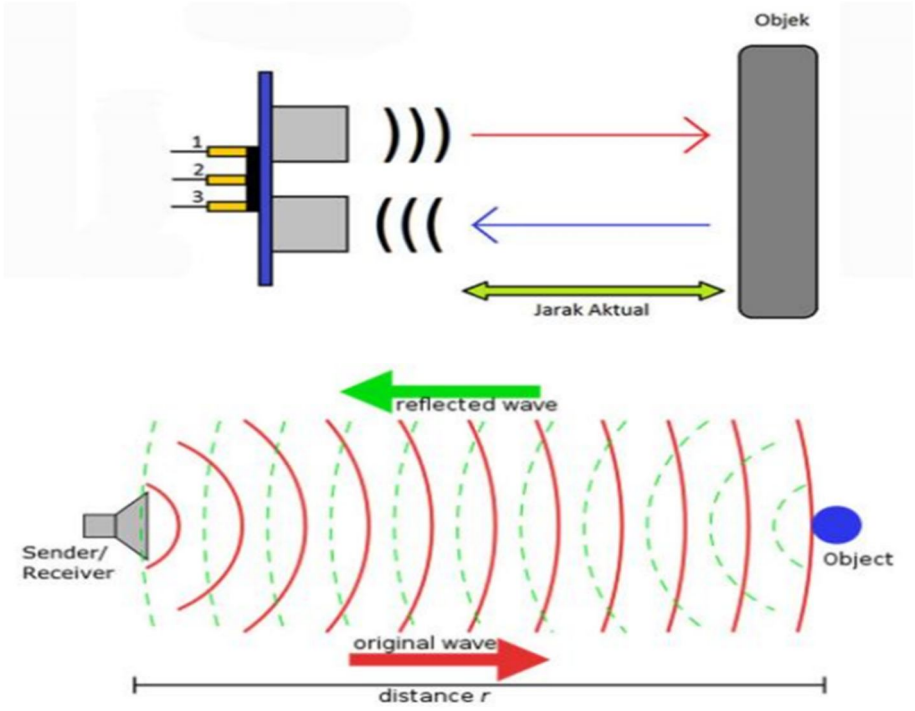
I. Sensor Ultrasonik HC-SR04

Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan *frekuensi* tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik).

Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat didengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba lumba. Bunyi ultrasonik nisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa.

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan *piezoelektrik* dengan frekuensi tertentu. *Piezoelektrik* ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah *osilator* diterapkan pada benda tersebut. Secara umum, alat ini akan

menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima. Dapat dilihat di gambar cara kerja sensor ultrasonik 2.2 :



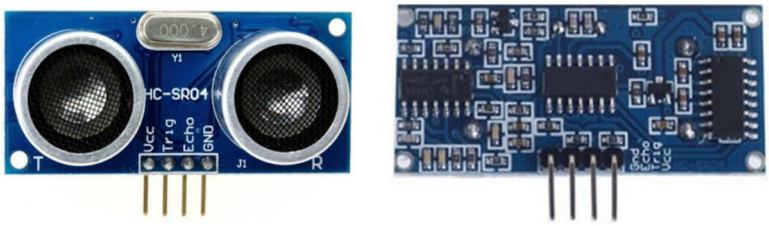
Gambar. 2.4 Cara kerja sensor ultrasonic
(Sumber: https://eprints.akakom.ac.id)

Secara detail, cara kerja sensor ultrasonik adalah sebagai berikut :

1. Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz.

2. Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.
3. Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut.

Sensor ini merupakan sensor ultrasonik siap pakai, satu alat yang berfungsi sebagai pengirim, penerima, dan pengontrol gelombang ultrasonik.



Gambar. 2.5 Sensor Ultrasonik HC-SR04
(Sumber: https://eprints.akakom.ac.id)

Fungsi Pin-pin HC-SR04

1. VCC = 5V *Power Supply*. *Pin* sumber tegangan positif sensor.
2. *Trig* = *Trigger*/Penyulut. *Pin* ini yang digunakan untuk membangkitkan sinyal ultrasonik.
3. *Echo* = *Receive*/Indikator. Pin ini yang digunakan untuk mendeteksi sinyal pantulan ultrasonik.
4. GND = *Ground*/0V *Power Supply*. *Pin* sumber tegangan negatif sensor.

Spesifikasi Sensor HC-SR04

1. Tegangan : 5V DC
2. Arus statis : < 2mA

3. Level *output* : 5v – 0V
4. Sudut sensor : < 15 derajat
5. Jarak yg bisa dideteksi : 2cm – 450cm (4.5m)
6. Tingkat keakuratan : up to 0.3cm (3mm)

J. Motor servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik *loop* tertutup (servo), sehingga dapat di *set-up* atau di atur untuk menentukan dan memastikan posisi sudut dari poros *output* motor. motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian *gear*, rangkaian kontrol dan *potensiometer*. Serangkaian *gear* yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan *potensiometer* dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo dapat dilihat pada gambar dibawah.



Gambar. 2.6 motor servo
(Sumber: https://Teknik Elektronika.com)

Spesifikasinya Motor Servo :

1. Tegangan Kerja : 4,8 – 6 Vdc
2. Torsi : 1,6 kg/cm
3. Arus : < 500 mA
4. Dimensi : 22 x 12,5 x 29,5 cm
5. Berat : 9 gr
6. Kecepatan Putaran : 0,12 detik/60 derajat

K. Pengertian *Website* dan HTTP

1. Definisi *Website*

Sebuah situs *web* (sering pula disingkat menjadi situs saja*; website*, *site*) adalah sebutan bagi sekelompok halaman *web* (*webpage*), yang umumnya merupakan bagian dari suatu nama *domain* (*domain name*) atau *sub domain* di *world wide web* (*WWW*) di *internet*. WWW terdiri dari seluruh situs *web* yang tersedia kepada publik. Halaman-halaman pada sebuah situs *web* diakses dari sebuah *URL* yang menjadi "akar" (*root*), yang disebut *homepage* (halaman induk; sering diterjemahkan menjadi "beranda", "halaman muka"), dan biasanya disimpan dalam *server* yang sama. Tidak semua situs *web* dapat diakses dengan gratis. Beberapa situs *web* memerlukan pembayaran agar dapat menjadi pelanggan, misalnya situs-situs yang menampilkan situs-situs berita, layanan surat elektronik (*email*), dan lain-lain.

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu komponen-komponen Sistem (*Sistem Components*), batasan-batasan sistem (*Sistem Boundaries*), penghubung sistem (*Sistem Interface*), lingkungan luar sistem (*Sistem Environment*), masukan sistem (*Sistem Input*), pengolah sistem (*Sistem Process*), keluaran sistem (*Sistem Output*), dan sasaran sistem (*Sistem Target*).

Menurut Hidayat dalam Daud F. Tatang (2013), "*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman."

Menurut Kustiyahningsih dalam Daud F. Tatang (2013), "*Web* adalah layanan yang didapat oleh pemakai komputer yang terhubung ke *internet*"

Browser adalah perangkat lunak untuk mengakses halaman-halaman *web* seperti *Internet Explorer, Mozilla Firefox, Opera, Safari, Google Chrome, Rock Melt*, dan lain-lain. Terminologi *website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah *domain* atau *sub domain*, yang tempatnya berada di dalam *World Wide Web (WWW)* di *internet*. Sebuah *web page* adalah dokumen yang ditulis dalam format *HTML (Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui *HTTP*, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para

pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

Halaman-halaman dari *website* akan bisa diakses melalui sebuah URL yang biasa disebut *homepage*. *URL* ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun *hyperlink-hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan. Beberapa *website* membutuhkan sub skripsi (data masukan/*subscript*) agar para *user* bisa mengakses sebagian atau keseluruhan isi *website* tersebut. Contohnya ada beberapa *situs-situs* bisnis, *situs-situs email* gratisan, yang membutuhkan *sub* skripsi agar kita bisa mengakses *situs-situs* tersebut.

Penemu *website* adalah Sir Timothy John "Tim" Berners Lee, sedangkan *website* yang tersambung dengan jaringan, pertama kalinya muncul pada tahun 1991. Maksud dari Tim ketika membuat *website* adalah untuk mempermudah tukar menukar dan memperbarui informasi kepada sesama peneliti di tempat dia bekerja. Pada tanggal 30 April 1993, CERN (tempat dimana Tim bekerja) menginformasikan bahwa WWW dapat digunakan secara gratis oleh semua orang.

Sebuah *website* bisa berupa hasil kerja dari perorangan atau individu, atau menunjukkan kepemilikan dari sebuah organisasi, perusahaan, dan biasanya *website* itu menunjukkan beberapa topik khusus, atau kepentingan tertentu. Sebuah *website* bisa berisi *hyperlink* yang menghubungkan ke *website* lain. Jadi, kadangkala perbedaan antara *website* yang dibuat oleh individu perseorangan dengan *website* yang dibuat oleh organisasi bisnis bisa saja tidak kentara.

Website ditulis, atau secara dinamik di konversi menjadi *HTML* dan akses melalui sebuah program *software* yang biasa disebut dengan *web browser*, yang dikenal juga dengan HTTP *Client*. Halaman *web* dapat dilihat atau diakses melalui jaringan komputer dan *internet*, perangkat bisa saja berupa Personal *Computer*, Laptop *Computer*, PDA atau *Cellphone*.

Sebuah *website* dibuat di dalam sebuah sistem komputer yang dikenal dengan *Web server*, juga disebut HTTP *server*, dan pengertian ini juga bisa menunjuk pada *software* yang dipakai untuk menjalankan sistem ini, yang kemudian menerima lalu mengirimkan halaman-halaman yang diperlukan untuk merespon permintaan dari *user*. *Apache* adalah bahasa *program/software* yang biasa digunakan didalam sebuah *web server*, kemudian setelah itu adalah *Microsoft Internet Information Server* (IIS).

Sebuah *website* statik, adalah salah satu bentuk *website* yang isi didalam *website* tersebut tidak dimaksudkan untuk di *update* secara berkala, dan biasanya diperbaiki secara manual oleh beberapa orang yang menggunakan *software editor*. Ada 3 tipe kategori *software editor* yang biasa dipakai untuk tujuan memperbaiki ini, mereka adalah:

1) Elemen 1, *Text* Editor;. Contohnya adalah *Notepad* atau Edit, dimana HTML diubah didalam program *editor* tersebut.
2) Elemen 2, WYSIWYG (*What You See Is What You Get*) Editor. Contohnya *Microsoft Frontpage* dan *Macromedia Dreamweaver*, dimana *situs* di *edit* menggunakan GUI (*Graphic User Interface*) dan format HTML secara otomatis di *generate* oleh editor ini.

3) Elemen 3 yaitu editor yang sudah memiliki *template*, contohnya *Rapidweaver* dan *iWeb*, dimana, editor ini memperbolehkan *user* untuk membuat dan mengupdate *website*-nya langsung ke *web server* secara cepat, tanpa harus mengetahui apapun tentang HTML. Mereka dapat memilih template yang sesuai dengan keinginan mereka, menambah gambar atau objek, mengisinya dengan tulisan dan dengan sekejap mereka sudah dapat membuat *website* tanpa harus melihat sama sekali kode-kode HTML.

Sebuah *website dynamic* adalah *website* yang secara berkala, informasi didalamnya berubah, atau *website* ini bisa berhubungan dengan user dengan berbagai macam cara atau metode, (HTTP *cookies* atau variabel *database*, sejarah kunjungan, variabel sesi, dan lain-lain) bisa juga dengan cara interaksi langsung menggunakan *form* dan pergerakan *mouse*. Ketika *web server* menerima permintaan dari *user* untuk memberikan halaman tertentu, maka halaman tersebut akan secara otomatis di ambil dari media penyimpanan sebagai respon dari permintaan yang diminta oleh *user*. Sebuah situs dapat menampilkan dialog yang sedang berlangsung diantara dua *user*, memantau perubahan situasi, atau menyediakan informasi yang berkaitan dengan sang *user*.

Ada banyak sistem *software* yang dapat dipakai untuk *menggenerate dynamic web system dan situs dynamic*, beberapa diantaranya adalah *Coldfusion* (CFM), *Active Server Pages* (ASP), *Java Server Pages* (JSP) dan PHP, bahasa program yang mampu untuk meng-generate *dynamic web system* dan situs dinamis. Situs juga bisa termasuk didalamnya berisi informasi yang diambil dari

satu atau lebih *database* atau bisa juga menggunakan teknologi berbasis XML, contohnya adalah RRS. Isi situs yang statis juga secara periodik di *generate*, atau, apabila ada keadaan dimana dia butuh untuk dikembalikan kepada keadaan semula, maka dia akan di *generate*, hal ini untuk menghindari kinerja supaya tetap terjaga.

Plugin atau dapat disebut pula *extension* tersedia untuk menambah banyaknya *feature* dan melengkapi kemampuan dari *web browser*, dimana plugin ini dipakai untuk membuka *content* yang biasanya berupa cuplikan dari gambar bergerak *(active content)* contohnya adalah *flash*, *shockwave* atau *applets* yang ditulis dalam bahasa JAVA. *Dynamic* HTML juga menyediakan untuk *user* supaya dia bisa secara *interaktif* dan *realtime*, *mengupdate* di *web page* tersebut (catatan; halaman yang dirubah, tak perlu di *load* atau di *reload* agar perubahannya dapat dilihat), biasanya perubahan yang dilakukan mereka memakai DOM dan *javascript* yang sudah tersedia pada semua *web browser* sekarang ini.

Seperti yang ditulis diatas, di luar sana ada beberapa perbedaan dalam penulisan dari terminology *website*. Walaupun *website* sudah secara umum dipakai, namun untuk *associated press stylebook*, *reuters*, *microsoft*, *academia*, dan kamus-kamus yang ada, penulisan yang mereka pakai adalah dengan menggunakan 2 kata, yaitu *website*. Hal ini karena *web* bukanlah terminologi umum, namun ia adalah kependekan dari *world wide web*.

2. Definisi HTTP

HTTP (*HyperText Transfer Protocol*) adalah protokol yang digunakan untuk mentransfer dokumen dalam *World Wide Web* (*WWW*). Protokol ini adalah

protokol ringan, tidak berstatus dan generik yang dapat dipergunakan berbagai macam tipe dokumen.

Pengembangan *HTTP* dikoordinasi oleh *World Wide Web* Consortium (W3C) dan grup bekerja *Internet Engineering Task Force* (IETF), bekerja dalam publikasi satu seri *Request For Comment (RFC),* yang paling terkenal RFC 2616, yang menjelaskan HTTP/1.1, versi HTTP yang digunakan umum sekarang ini.

HTTP adalah sebuah protokol meminta/menjawab antara *Client* dan *server* sebuah *client* HTTP seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke *port* tertentu di tuan rumah yang jauh (biasanya *port* 80). Sebuah *server* HTTP yang mendengarkan di *port* tersebut menunggu client mengirim kode permintaan (*request*), seperti "*Get* / HTTP/1.1" (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikuti dengan badan dari data tersebut. Beberapa kepala (*header*) juga bebas ditulis atau tidak, sementara lainnya (seperti tuan rumah) diperlukan oleh HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), *server* mengirim kembali kode jawaban, seperti "200 ok", dan sebuah pesan yang diminta, atau sebuah pesan *error* atau pesan lainnya.

HTTP pertama kali dipergunakan dalam WWW pada tahun 1990. Pada saat tersebut yang dipakai adalah HTTP *versi* 0.9. *Versi* 0.9 ini adalah protokol transfer dokumen secara mentah, maksudnya adalah data dokumen dikirim sesuai dengan isi dari dokumen tersebut tanpa memandang tipe dari dokumen.

Kemudian pada tahun 1996 HTTP diperbaiki menjadi HTTP versi 1.0. Perubahan ini untuk mengakomodasi tipe-tipe yang hendak dikirim beserta *encoding* yang digunakan dalam pengiriman data dokumen.

Sesuai dengan perkembangan infrastruktur *internet* maka pada tahun 1999 dikeluarkan HTTP versi 1.1 untuk mengakomodasi *proxy, cache, koneksi persistent.*

L. PHP



Gambar 2. 7 Hypertext Preprocessor (PHP)

(www.id.wikipedia.org)

PHP adalah singkatan dari *Hypertext Preprocessor*. PHP adalah bahasa *pemrograman* yang dipakai untuk membuat *web* dinamis, meskipun juga bisa dipakai untuk membuat program lain. PHP juga dikenal sebagai *pemrograman script server side*, sebab PHP di proses pada komputer *server*.

Variabel adalah suatu simbol untuk tempat penyimpanan sementara yang nilainya dapat berubah-ubah. Data yang tersimpan pada variabel akan hilang setelah program selesai dijalankan dan nilai pada variabel akan di *reset* menjadi *default*. Adapun cara penulisan variabel yaitu:

1. Untuk memberi nama variabel hanya boleh menggunakan huruf, angka dan garis bawah (_)
2. Nama variabel harus diawali dengan tanda **$**.
3. Tidak boleh menggunakan spasi pada nama variabel.
4. Variabel bersifat *case-sensitive* atau sensitif terhadap huruf besar dan kecil.

```
1 <?php
2 $nama;
3 ?>
```

Gambar. 2. 8 Penulisan Variabel Php

1. Tipe data PHP :

a. *Integer*

Tipe data *integer* adalah tipe data yang berbentuk angka yang berbentuk bilangan asli atau bilangan bulat. Seperti yang saya jelaskan sebelumnya pada tipe data *string*, untuk penulisan tipe data *integer* tidak perlu menambahkan tanda petik, karena akan dibaca sebagai *string* jika Anda menambahkan tanda petik pada tipe data *integer*.

```
1 <?php
2     $bilangan_pertama = 12;
3     $bilangan_kedua = 78;
4     //Kedua variabel di atas merupakan variabel yang bertipe data integer.
5 ?>
```

Gambar. 2. 9 Tipe Data Integer

Terdapat dua buah variabel yang bernama $bilangan_pertama dan $bilangan_kedua. Kedua variabel tersebut berisikan angka yang berbentuk bilangan bulat atau bilangan asli. Oleh sebab itu kedua variabel di atas di sebut sebagai variabel yang bertipe data *integer*.

b. *Float*

Tipe data *float* atau disebut juga tipe data *double* adalah tipe data yang berisi bilangan desimal. Cara penulisannya hampir sama dengan tipe data *integer* karena tidak memerlukan tanda petik di awal dan di akhir isi variabel *float*.

```php
<?php
    $angka = 12.177;
    //variabel di atas adalah variabel yang bertipe data float karena berisi bilangan desimal.
?>
```

Gambar. 2. 10 Tipe Data Float

c. *String*

Tipe data *string* adalah tipe data yang berbentuk *text* dan untuk cara penulisan tipe data *string* di letakkan di tengah-tengah tanda petik. Di awali dengan tanda petik dan di akhiri dengan tanda petik juga.

```php
<?php
  $tes = "Usia Saya Sekarang 23 Tahun";
  // variabel tes di atas merupakan tipe data string karena berisi text atau kalimat.
?>
```

Gambar. 2. 11 Tipe Data String

d. *Boolean*

Tipe data *Boolean* merupakan tipe data yang bernilai *True* atau *False.* Tipe data *boolean* sering digunakan untuk memeriksa ketersediaan atau memeriksa nilai kebenaran pada sebuah data.

```php
<?php
        $x = true;
        $y = false;

        echo $x; //Hasil: 1
        echo "<br>";
        echo $y; //Tidak ada hasil yang ditampilkan
?>
```

Gambar. 2. 12 Tipe Data Boolean

e. *Array*

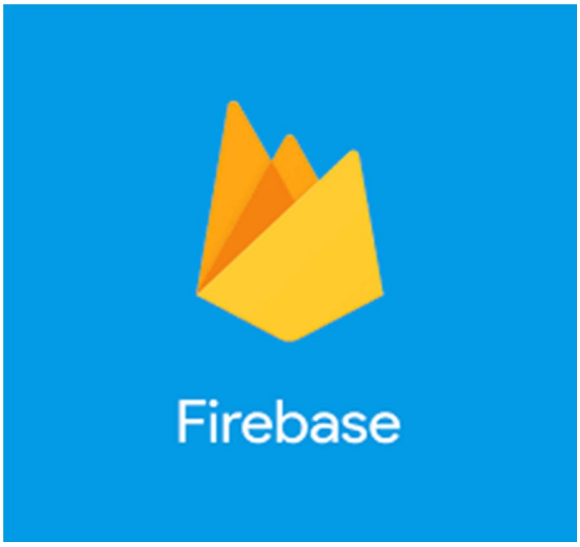
Array merupakan kumpulan dari tipe data, didalamnya memiliki banyak nilai. Jadi dapat dikatakan *array* merupakan tipe data yang terdiri dari kumpulan tipe data.

```php
<?php
  // membuat array
  $tas = array('Pulpen','Buku Tulis','Penggaris');

  // menampilkan isi array
  echo $tas[0]."<br>";
  echo $tas[1]."<br>";
  echo $tas[2]."<br>";
?>
```

Gambar. 2. 13 Tipe Data Array

M. *Firebase*



Gambar. 2. 14 firebase
(www.firebase.google.com)

Firebase adalah suatu layanan *Google* yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya *Firebase*, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Dua *fitur* yang menarik dari *Firebase* yaitu *Firebase Remote Config dan Firebase RealTime Database.*

Realtime Database adalah *database NoSQL*, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan *database* terkait. *API Realtime Database* dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan Anda untuk membangun pengalaman *realtime* yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan

tersebut. Berikut ini bagaimana proses penginstalan dan persiapan *firebase* ke aplikasi *Android*:

1. Menghubungkan aplikasi ke *Firebase*

   Tambahkan *Firebase* ke *project Android* yang mana dalam file *build.gradle level project*, pastikan untuk memasukkan *repository Maven Google* dibagian *buildscript* dan *allprojects*.

2. Menambahkan Realtime *Database* ke aplikasi

   Menambahkan dependensi untuk *Realtime Database* ke file *build.gradle* level aplikasi dengan *script* aplikasi:

   ```
   implementation 'com.google.firebase:firebase-database:19.1.0'
   ```

3. Mengkonfigurasi Aturan *Realtime Database*

   Realtime Database menyediakan bahasa aturan deklaratif sehingga programmer dapat menentukan cara data disusun, *indeks*, serta kapan data tersebut dapat dibaca dan ditulisi. Secara *default*, akses baca dan tulis ke *database* dibatasi, sehingga hanya pengguna yang terautentikasi yang dapat membaca atau menulis data. Untuk memulai tanpa menyiapkan *Autentikasi*, serta dapat mengonfigurasi aturan agar dapat diakses oleh publik. Tindakan ini membuat *database* terbuka bagi siapa saja, termasuk orang yang tidak menggunakan aplikasi. Oleh karena itu, pastikan kembali untuk membatasi *database* setelah menyiapkan *autentikasi*.

4. Menulis ke *database* Anda

   Ambil *instance database* menggunakan *getInstance ()* dan rujukan lokasi tempat dimana ingin menulis.

```
// Write a message to the database
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");
```

Sehingga bisa menyimpan berbagai jenis data ke *database* dengan cara ini, termasuk objek *Java*. Ketika menyimpan sebuah objek, *respons* dari setiap *getter* akan disimpan sebagai turunan dari lokasi ini.

5. Membaca dari *database* Anda

Agar data aplikasi diupdate secara *realtime*, maka perlu untuk menambahkan *Value Event Listener* ke referensi yang baru saja dibuat. Metode *on Data Change* () di class ini dipicu satu kali saat pemroses ditambahkan dan dipicu lagi setiap kali terjadi perubahan data, termasuk turunannya.

```
// Read from the database
myRef.addValueEventListener(newValueEventListener(){
    @Override
    publicvoid onDataChange(DataSnapshot dataSnapshot){
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        String value = dataSnapshot.getValue(String.class);
        Log.d(TAG,"Value is: "+ value);
    }
```

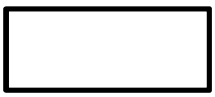
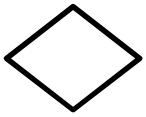
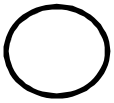
N. *Flowchart*

Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (*instruksi*) dengan proses lainnya dalam suatu program.

Dalam perancangan *flowchart* sebenarnya tidak ada rumus atau patokan yang bersifat mutlak (pasti). Hal ini didasari oleh *flowchart* (bagan alir) adalah sebuah gambaran dari hasil pemikiran dalam menganalisa suatu permasalahan dalam komputer. Karena setiap analisa akan menghasilkan hasil yang bervariasi antara satu dan lainnya. Kendati begitu secara garis besar setiap perancangan *flowchart* selalu terdiri dari tiga bagian, yaitu input, proses dan *output*.

Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *flowchart* :

Tabel 2. 1 Simbol-simbol Flowchart

Simbol	Nama	Fungsi
	Terminator	Permulaan / akhir program
	Garis alir	Arah aliran program
	Preparation	Proses inisialisasi/pemberian harga awal
	Proses	Proses perhitungan/proses pengolahan data
	Decision	Perbandingan pernyataan penyelesaian data yang memberikan pilihan untuk laghkah selanjutnya.
	Input/Output data	Proses input/output data.
	On page	Penghubung bagian-bagian flowchart yang berada pada satu halaman

O. UML (Unified Modelling Language)

UML (*Unified Modelling Language*) menurut menurut Adi nugroho (2010:6), "*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek". Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu:

a. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language* (UML):

1) *Structural things,* bagian yang relatif statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
2) *Behavioral things,* bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
3) *Grouping things,* bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem-subsistem.
4) *Notational things,* merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen.

b. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

1) Ketergantungan (d*ependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen independent akan mempengaruhi elemen yang bergantung padanya.
2) Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
3) Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada diatasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi..
4) Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.

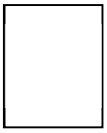
c. Diagram, *Unified Modelling Language* (UML) menyediakan Sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

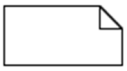
1. diagram *use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang di bangu. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai. (*Sholiq, 2006*)

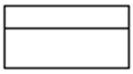
Adapun simbol-simbol *Use Case* Diagram antara lain :

Tabel 2. 2 Simbol Use Case Diagram

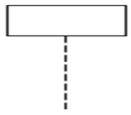
NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

NO	GAMBAR	NAMA	KETERANGAN
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 2. 3 Simbol Class Diagram

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

Tabel 2. 4 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

Tabel 2. 5 Simbol StateChart Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

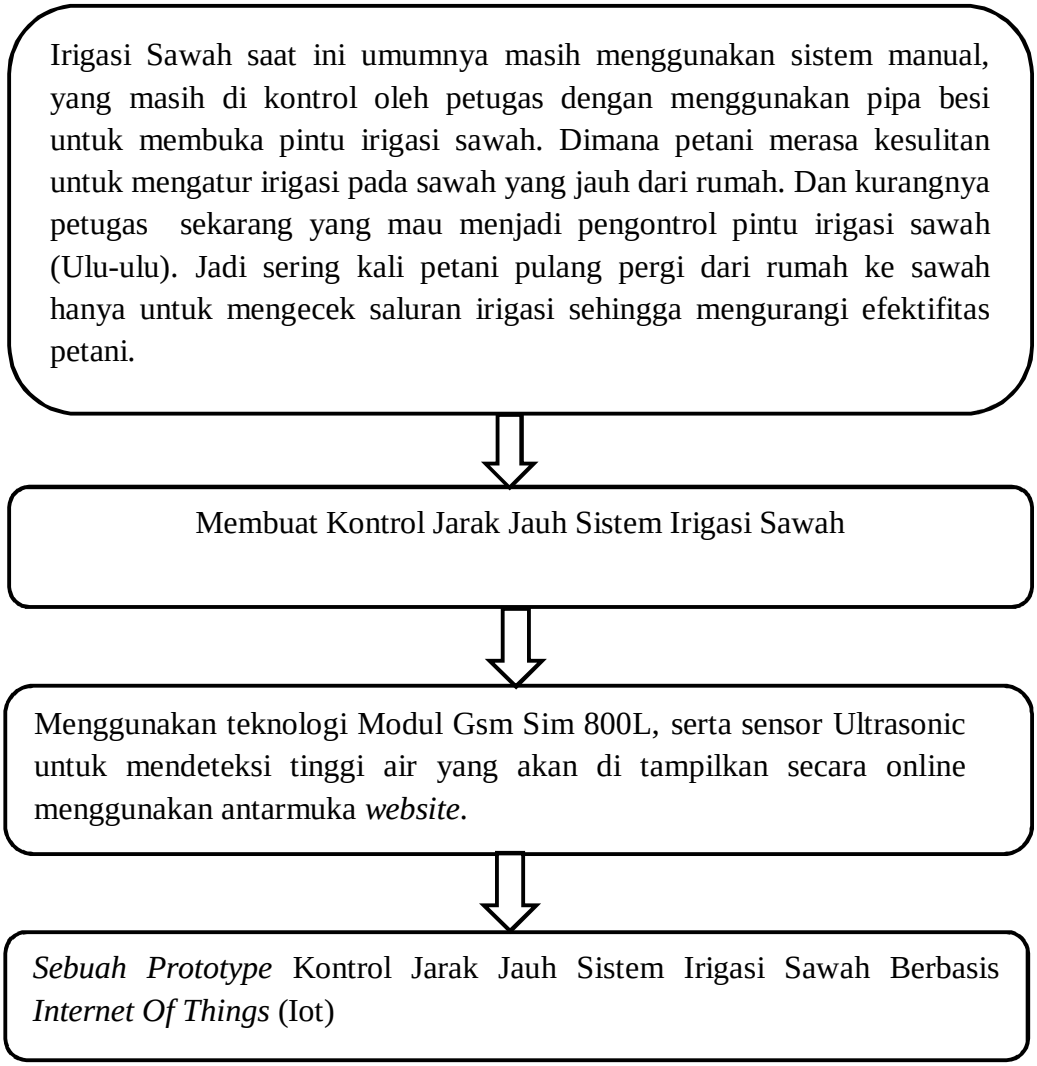
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 6 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 dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

P. Kerangka Pikir

Untuk memperjelas alur penelitian ini dibuatkan kerangka pikir sebagai :



BAB III

METODE PENELITIAN

A. Lokasi dan Waktu Penelitian

1. Penelitian yang dilakukan ini berlokasi di Pintu Air Bangunan Irigasi kelompok tani Pammasewae di Desa Panrenge'e Kelurahan Araten Kabupaten Sidenreng Rappang.
2. Waktu yang dipergunakan untuk pelaksanaan penelitian adalah ± 2 bulan .

B. Jenis Penelitian

Untuk membantu kelancaran pengumpulan data, maka penulis menggunakan beberapa metode antara lain :

1. Penelitian Lapangan (*Field Research*)

   Yaitu penulis langsung melakukan penelitian pada objek. Data dan keterangan yang diperoleh dikumpulkan melalui pengamatan maupun tanya jawab langsung sesuai dengan data dan keterangan yang dibutuhkan. Dalam hal ini menggunakan :

   a. Pengamatan (*Observation*), yaitu melakukan pengamatan secara langsung pada Pintu Air Bangunan Irigasi kelompok tani Pammasewae di Desa Panrenge'e Kelurahan Araten Kabupaten Sidenreng Rappang untuk mengetahui bagaimana mekanisme Kontrol Pintu Air yang sedang berjalan. Dari hasil observasi ini dapat mengetahui masalah-masalah yang sering terjadi serta kelemahan-kelemahan yang ada pada mekanisme yang sedang berjalan.

b. Wawancara (*Interview*), yaitu melakukan tanya jawab secara langsung dengan berbagai pihak yang berkompeten dan mengetahui permasalahan yang dibahas. Dalam hal ini penulis melakukan wawancara secara langsung kepada ketua kelompok tani Pammasewae di Desa panrenge'e Kelurahan Araten Kabupaten Sidenreng Rappang.

2. Kajian Kepustakaan (*Literature Study*)

Merupakan pengumpulan data dengan cara mencari hasil penelitian sejenis yang menerapkan konsep *Internet of Things* (IoT) pada mekanisme kontrol.

C. Metode Pengumpulan Data

Untuk memperoleh data-data yang dibutuhkan dalam rangka melakukan penelitian, maka penulis mengumpulkan data melalui dua cara yaitu :

1. Secara Langsung

Di Dalam memperoleh data-data yang dibutuhkan, maka peneliti melakukan survey lokasi pada Pintu Air Bangunan Irigasi kelompok tani Pammasewae di Desa Panrenge'e Kelurahan Araten Kabupaten Sidenreng Rappang dan wawancara langsung pada pengelolaan Pintu Air Bangunan irigasi.

2. Secara Tidak Langsung

Mengumpulkan data hasil penelitian sejenis yang menerapkan konsep *Internet of Things* (IoT) pada mekanisme kontrol.

D. Alat dan Bahan Penelitian

Untuk melakukan proses penelitian dalam pembuatan aplikasi, maka diperlukan perangkat keras dan perangkat lunak guna mendukung kegiatan penelitian tersebut.

1. Perangkat keras

   Perangkat keras yang digunakan untuk membuat aplikasi Kontrol Jarak Jauh Sistem Irigasi Sawah dapat dilihat pada tabel berikut :

   Tabel 3. 1 Spesifikasi Perangkat Keras

Spesifikasi	
Merk laptop	Acer
Processor Laptop	*Intel*(R) *Celeron*(R) CPU 1007U
RAM Laptop	RAM 4GB
Module GSM	*Modul* GSM GPRS TTGO T-CALL V1.3 Sim800L ESP32
Jenis Sensor	Sensor Ultrasonik HC-SR04
Perangkat Tambahan	Motor Servo, Kabel Jumper, Papan *Breadboard*

2. Perangkat lunak

   Perangkat lunak yang digunakan untuk membuat aplikasi Kontrol Jarak Jauh sistem Irigasi Sawah dapat dilihat sebagai berikut :

   Tabel 3. 2 Spesifikasi Perangkat Lunak

Spesifikasi	
Sistem operasi	*Windows* 10
Tool Pemrograman	*Arduino IDE*
Bahasa Pemrograman	PHP & JAVASCRIPT
Database	*Firebase*

3. Perangkat *Android Mobile*

Perangkat *android mobile* yang digunakan untuk menginstall dan menjalankan aplikasi Kontrol Jarak Jauh sistem Irigasi Sawah dapat dilihat pada tabel berikut :

Tabel 3. 3 Spesifikasi Perangkat Android Mobile

Spesifikasi	
Merk	Redmi 5 Plus
Versi Android	7.1.2 N2G47H
Processor	*Snapdragon* 712
RAM	3 GB
Resolution	2436x1125
CPU	2.0 GHz

Untuk melakukan penelitian ini digunakan beberapa alat, yang terdiri dari :

a. Perangkat Keras

1) Laptop dengan spesifikasi *Intel(R) Celeron(R*
2) TTGO T-Call V1.3 ESP32 *Wireless Module* GPRS *Antenna* SIM *Card* SIM800L
3) Sensor Ultrasonik HC-SR04
4) Motor Servo
5) *Smartphone*

b. Perangkat Lunak

1) *Windows* 10
2) PHP
3) Arduino IDE

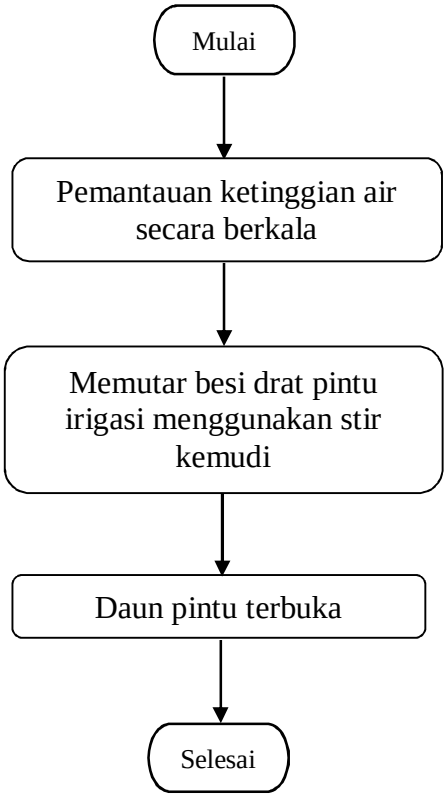
E. Metode Pengujian

Metode pengujian yang dilakukan pada penelitian ini adalah pengujian yang dipakai untuk mengetahui kinerja *prototype* pada saat melakukan proses kontrol. Metode pengujian yang dilakukan terdiri dari :

Metode pengujian *black box* untuk mengetahui berfungsinya semua fungsi pada *prototype* yang telah dibuat;

F. Desain Sistem

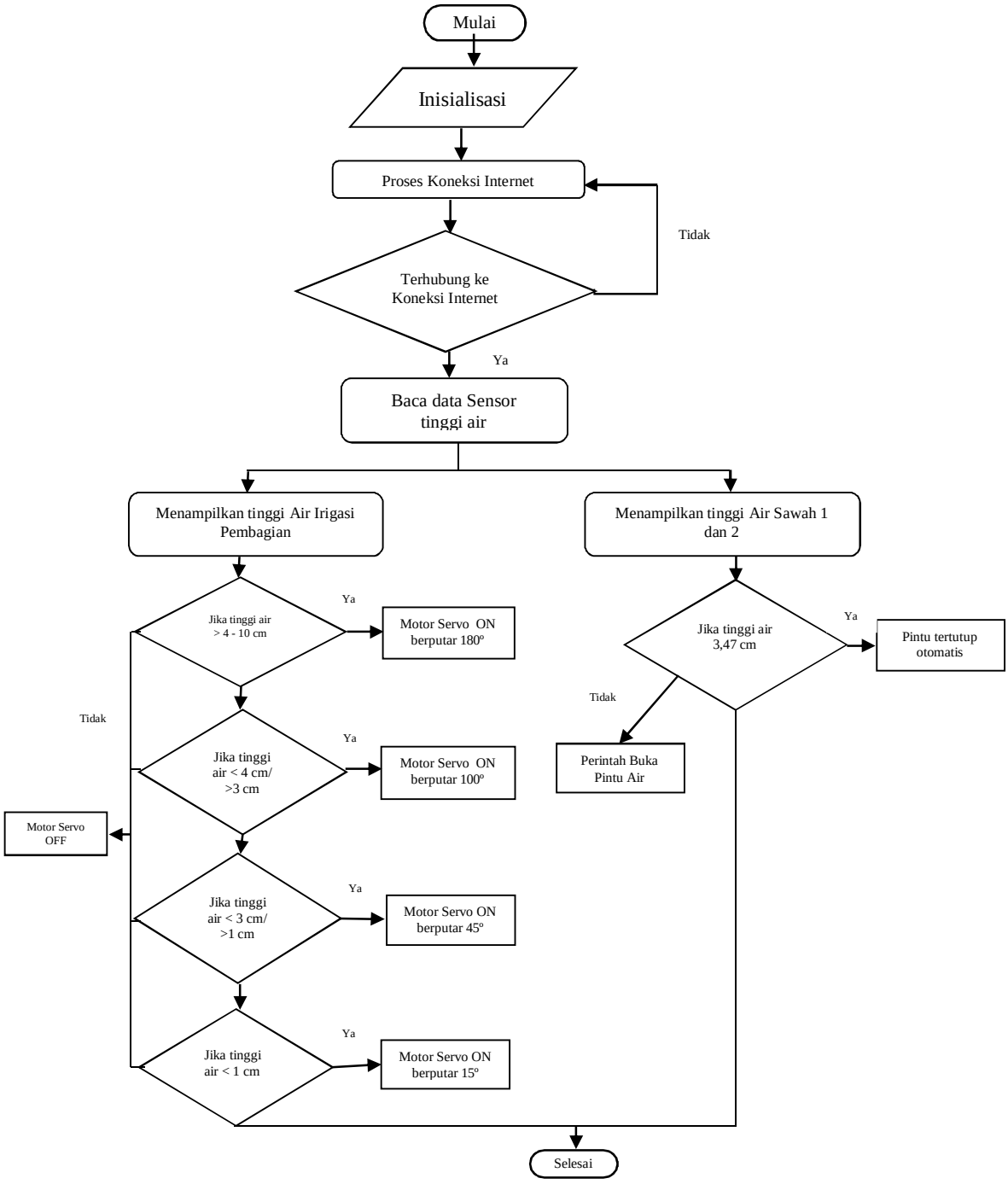
1. Sistem berjalan



Gambar 3. 1 Sistem Yang Berjalan

Pada gambar di atas menjelaskan bagaimana pengontrolan air secara berkala dan pengontrolan pintu masih manual atau menggunakan stir kemudi membuka pintu irigasi sawah.

2. Sistem yang diusulkan

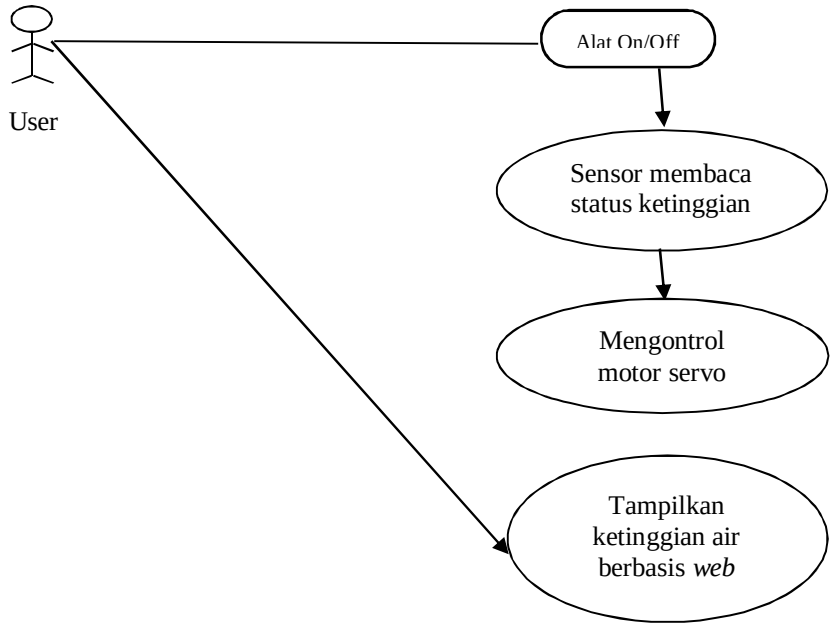


Gambar 3.2 flowchart sistem monitoring yang diusulkan

Penjelasan dari *flowchart* tersebut adalah pertama melakukan inisialisasi jaringan, apabila jaringan telah terhubung maka alat akan membaca data sensor ketinggian air serta mengirim data ke *database* lalu menampilkannya ke dalam *website*. Pada data ketinggian air irigasi pembagian mengikuti tinggi air pembukaan pintu, jika tinggi air > 4-10 cm maka motor servo akan berputar 180º, tinggi air <4 cm atau > 3 cm maka motor servo akan berputar 100º, tinggi air <3 cm atau > 1 cm maka motor servo akan berputar 45º, tinggi air < 1 cm maka motor servo akan berputar 15º, dan pada data ketinggian air sawah, jika tinggi mencapai 3,47 cm maka pintu akan tertutup otomatis dan pintu buka secara manual karena ada saatnya sawah terisi air.

G. Use Case Diagram

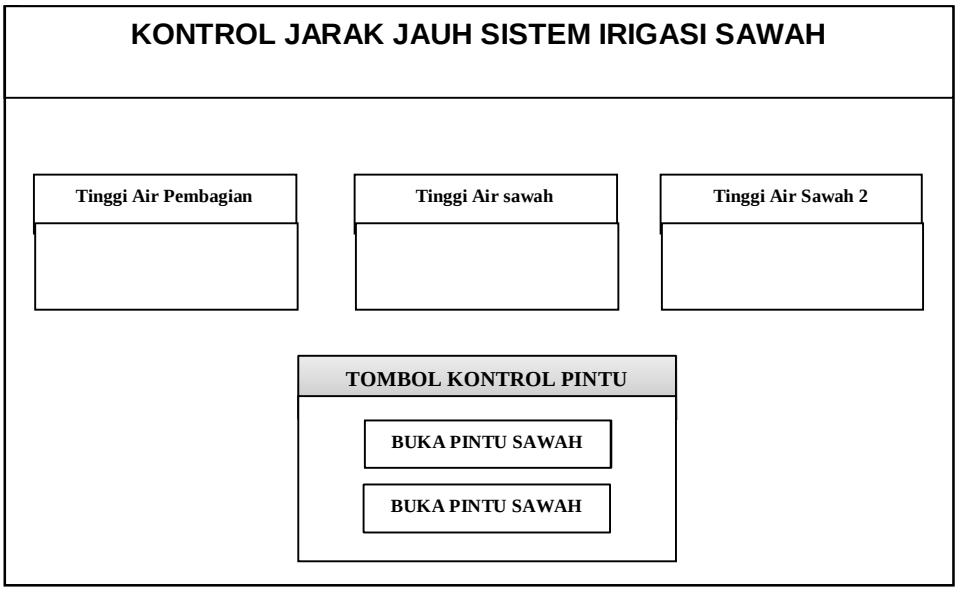
1. Diagram Use Case



Gambar 3.3 Use Case Diagram

Pada gambar diatas merupakan perancangan sistem yang dibuat menggunakan diagram *use case*. Dalam diagram *use case* ini terdapat tiga *case*, *case* pertama adalah sensor membaca status ketinggian air, *case* yang kedua mengotrol Motor Servo, *case* yang ketiga adalah tampilan status ketinggian air yang berbasis *web*.

H. User Interface



Gambar 3. 2 Rancangan menu utama kontrol jarak jauh irigasi sawah

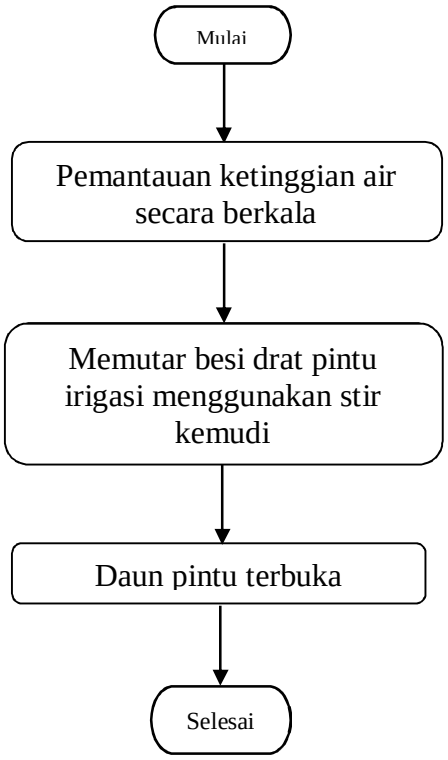
Aplikasi *web* akan menampilkan informasi tinggi air irigasi pembagian, tinggi air sawah 1, tinggi air sawah 2, serta membuka/mengontrol pintu sawah apabila memerlukan air dan akan menutup otomatis jika air sawah mencapai tinggi yang ditentukan.

BAB IV

HASIL DAN PEMBAHASAN

Pada penjelasan perancangan sistem dan analisis ini perlu diperhatikan dengan baik karena pada saat pembuatan alat dan aplikasi pertama–tama yang harus dilakukan adalah membuat rancangan *prototype* dengan jalur jalannya program. Dimulai dengan bagaimana data di-*input* kemudian diproses dengan sistem yang kita buat.

A. Rancangan sistem yang berjalan

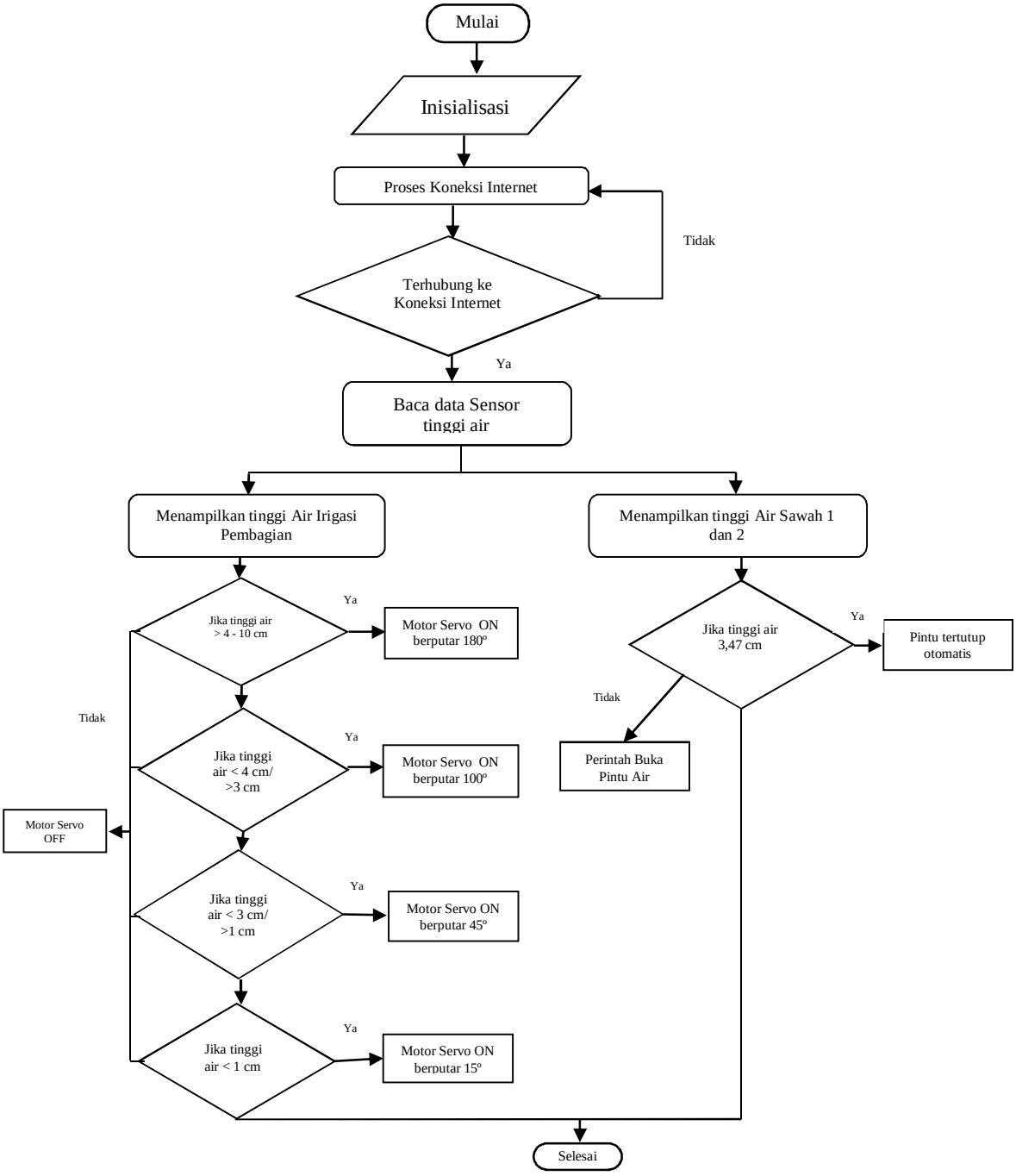


Gambar 4. 1 Sistem Yang Berjalan

Pada gambar di atas menjelaskan bagaimana pengontrolan air secara berkala dan pengontrolan pintu masih manual atau menggunakan stir kemudi membuka pintu irigasi sawah.

B. Rancangan Sistem yang Diusulkan

Sistem dimulai dari mengoperasikan komponen alat yang saling terhubung pada prototipe sistem *monitoring* tinggi irigasi sawah pada *internet of things* yaitu menghubungkan *modul gsm sim800l*, sensor ultrasonik, dan *motor servo*.



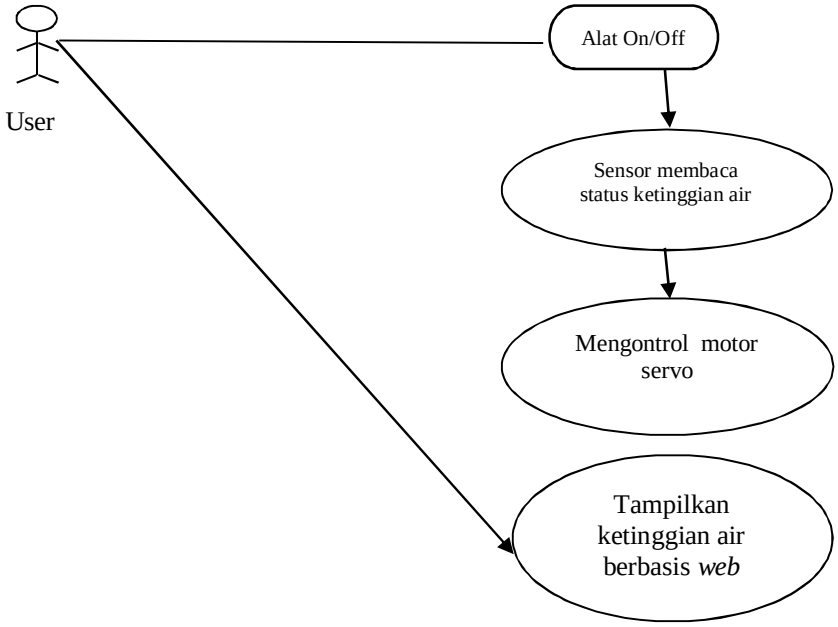
Gambar 4.2 flowchart sistem monitoring yang diusulkan

Penjelasan dari *flowchart* tersebut adalah pertama melakukan inisialisasi jaringan, apabila jaringan telah terhubung maka alat akan membaca data sensor ketinggian air serta mengirim data ke *database* lalu menampilkannya ke dalam *website*. Pada data ketinggian air irigasi pembagian mengikuti tinggi air pembukaan pintu, jika tinggi air > 4-10 cm maka motor servo akan berputar 180º, tinggi air < 4 cm atau > 3 cm maka motor servo akan berputar 100º, tinggi air < 3 cm atau > 1 cm maka motor servo akan berputar 45º, tinggi air < 1 cm maka motor servo akan berputar 15º, dan pada data ketinggian air sawah, jika tinggi mencapai 2,47 cm maka pintu akan tertutup otomatis dan pintu buka secara manual karena ada saatnya sawah terisi air.

Pemodelan dalam pengembangan sistem informasi menggunakan UML. UML suatu metode pemodelan secara visual untuk sarana perancangan sistem berorientasi objek. Tipe UML yang digunakan dalam pengembangan sistem informasi adalah *Use Case.* Diagram dibentuk sebagai berikut:

1. *Use Case*

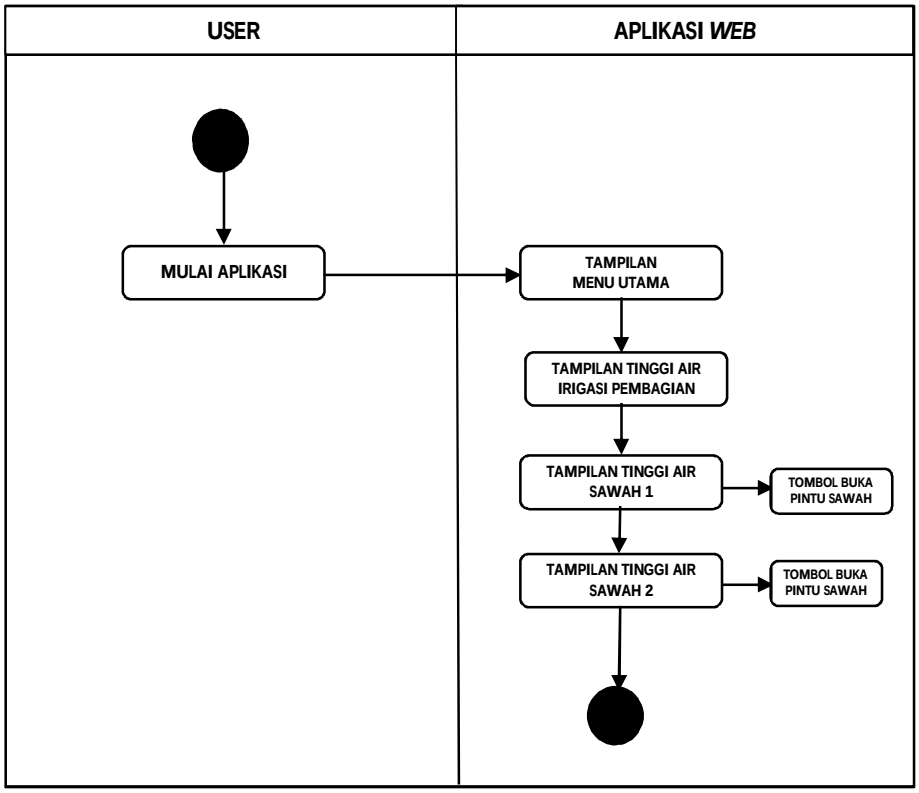
   use case dibentuk sebagai langkah awal perancangan sistem yang menggambarkan interaksi antara sistem dan aktor. *Use Case* juga dapat mendeskripsikan tipe interaksi si pemakai sistem dengan sistemnya. Berikut ini gambaran *use case* :



Gambar 4.3 Use Case Diagram

Pada gambar diatas merupakan perancangan sistem yang dibuat menggunakan diagram *use case*. Dalam diagram *use case* ini terdapat tiga *case*, *case* pertama adalah sensor membaca status ketinggian air, *case* yang kedua mengotrol Motor Servo, *case* yang ketiga adalah tampilan status ketinggian air yang berbasis *web*.

2. Activity diagram



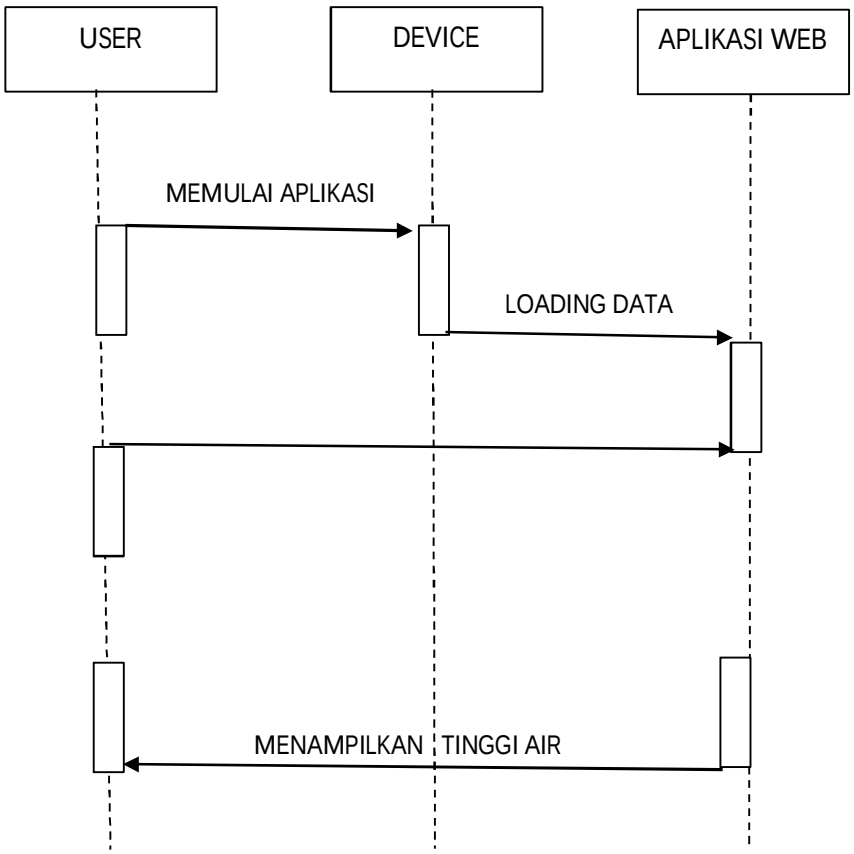
Gambar 4. 4 Activity diagram

Activity Diagram di atas menjelaskan bahwa ketika pengguna menjalankan aplikasi ini, pengguna akan langsung diarahkan ke menu utama aplikasi.

Aplikasi ini telah menampilkan status tinggi air irigasi pembagian, tinggi air sawah 1 dan 2, serta tombol buka pintu sawah secara manual.

3. *Sequence* diagram

Sequence Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah diagram *sequence*.



Gambar 4. 5 Sequence diagram

C. Desain Sistem

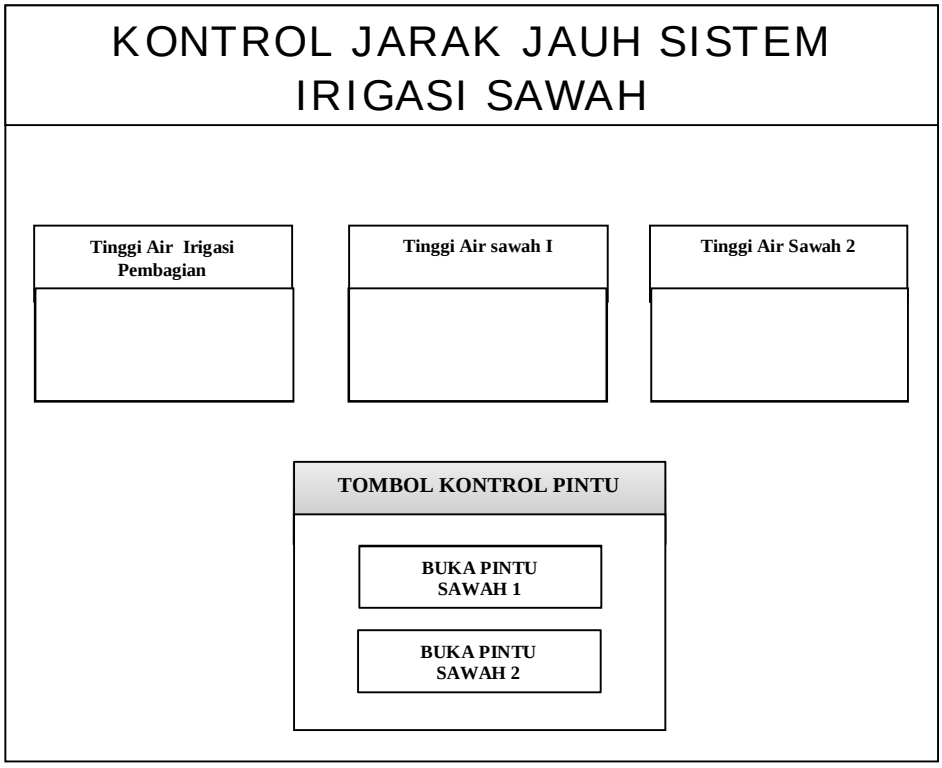
Dalam proses perancangan desain aplikasi, dalam hal ini desain tampilan perlu adanya sketsa desain yang digunakan untuk menggambarkan pembuatan media. Sketsa tersebut dibentuk dalam sebuah gambar. Tahap perancangan dimulai dengan membuat sketsa aplikasi agar pembuatan program aplikasi dapat dilakukan secara terurut.

1. Perancangan Antar Muka / *Interface*

Berikut penjelasan dari beberapa rancangan antar muka dari aplikasi ini:

a) Halaman Utama

Halaman utama ini merupakan halaman yang berisi tampilan Home, tinggi air pembagian, tinggi air sawah 1, tinggi air sawah 2. Pada tombol kontrol pintu sawah untuk membuka pintu sawah..



Gambar 4. 6 Halaman utama

D. Rancangan Perangkat Keras (*Hardware*)

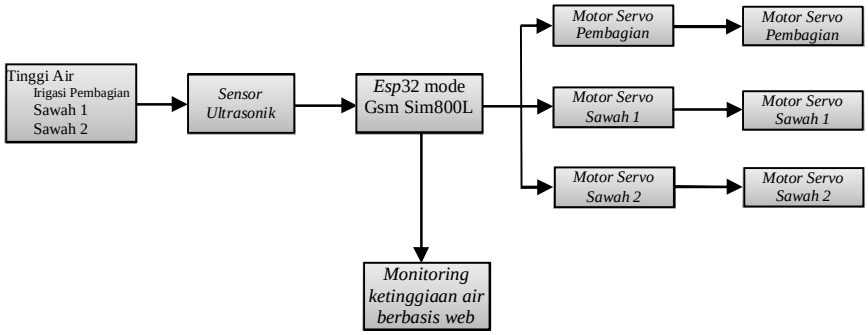
Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya sesuatu sistem beserta rangkaian elektronikanya. Rancangan ini akan menjelaskan alat yang digunakan dan memilih komponen yang tepat. Pada dasarnya rancangan prototipe sistem kontrol ini mempunyai hardware yang terdiri dari *TTGO T-CALL ESP32 Sim800l*, Sensor Ultrasonik, Motor Servo.

E. Implementasi Perangkat Keras (*Hardware*)

Dari blok diagram diatas terdapat bagian perangkat keras (*hardware*) yaitu sebagai berikut :

1. TTGO T-Call ESP32 SIM800L
2. *Sensor Ultrasonik*
3. Motor Servo

Empat perangkat keras diatas tersebut akan dilakukan perancangan dan pembuatan yang diimplementasikan sesuai sistem yang dirancang karena berupa alat/modul yang sudah jadi dan siap digunakan sesuai dengan kebutuhan.

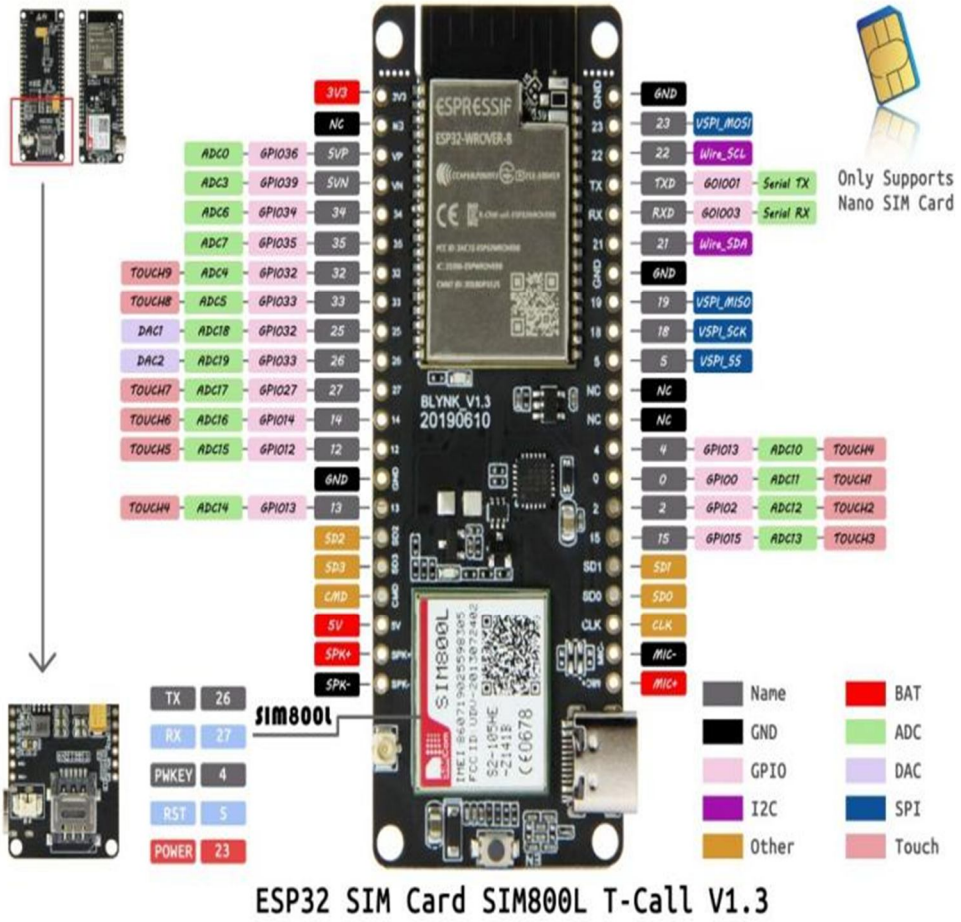


Gambar 4.7 Blok diagram sistem

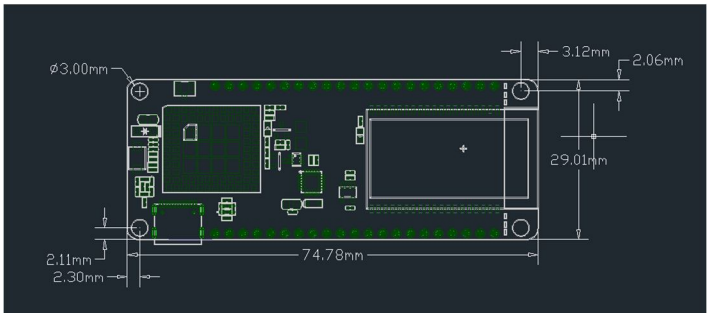
Adapun uraian masing – masing perangkat keras (*hardware*) yaitu :

1. TTGO T-Call ESP32 SIM800L

   TTGO T-Call ESP32 SIM800L dianalogikan sebagai board yang terkoneksi dengan *Esp32*. *Esp32* juga sudah terintegrasi dengan berbagai fitur selayaknya *mikrokontroler* dan kapasitas aksesnya terdapat *GSM* dan *chip* komunikasi yang berupa *USB to serial* sehingga pemrograman hanya membutuhkan kabel data *USB.*



Gambar 4.8 TTGO T-Call ESP32 SIM800L
(Sumber: www. diyusthad.com)



Gambar 4.9 grafik ukuran TTGO T-Call ESP32 SIM800L
(Sumber: www. diyusthad.com)

Tabel 4. 1 Spesifikasi TTGO T-Call ESP32 SIM800L

Nama	Keterangan
Chipset	ESPRESSIF-ESP32 240MHz *Xtensa®Mikroprosesor* tunggal/dual-core 32-bit LX6
Lampu kilat	QSPI flash 4MB / PSRAM 8MB
SRAM	SRAM 520 kB
Tombol tekan	Atur ulang
USB ke TTL	CP2104 atau CH9102F QFN24 Atau CH340K
Antarmuka Modular	UART, SPI, SDIO, I2C, PWM, I2S, ADC
Jam On-board	*Osilator kristal* 40MHz
Tegangan kerja	2.7V-3.6V
Bekerja saat ini	Sekitar 70mA
Tidur saat ini	Sekitar 1.1mA
Kartu SIM	Hanya mendukung kartu SIM Nano
Rentang suhu kerja	-40 ℃ ~ + 85 ℃
Ukuran & berat Spesifikasi catu daya	78.83mm * 28.92mm * 8.06mm(11.77g)
Catu daya	USB 5V/1A
Arus pengisian daya	500mA
Baterai mobil	Baterai *lithium* 3.7V
Konektor JST	2pin 2.0mm
USB portable	Tipe-C

Nama	Keterangan
Wi-Fi	
Standar	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC (chip esp32)
Protokol	802.11 b/g/n(802.11n kecepatan lebih tinggi to150Mbps)A-MPDU dan A-MSDU *polimerisasi*, dukungan 0.4μS Perlindungan *interval*
Rentang frekuensi	2.4GHz ~ 2.5GHz(2400M ~ 2483.5M)
Mengirimkan daya	22dBm
Jarak komunikasi	300m
Bluetooth	
Protokol Protokol	Memenuhi *bluetooth* v4.*bryant*/EDR dan BLE standard
Frekuensi Radio	Dengan-97dBm sensitivitas penerima NZIF kelas-1, kelas-2 & kelas-3 emitor AFH
Frekuensi Audio	*Frekuensi audio* CVSD & SBC
Spesifikasi perangkat lunak	
Mode Wi-Fi	Stasiun/SoftAP/SoftAP + Stasiun/P2P
Mekanisme keamanan	WPA/WPA2/WPA2-*Enterprise*/WPS
Jenis enkripsi	AES/RSA/ECC/SHA
Peningkatan *Firmware*	Unduh/OTA UART (melalui jaringan/host untuk mengunduh dan menulis *firmware*)
Pengembangan perangkat lunak	Mendukung pengembangan *server* awan/SDK untuk pengembangan *firmware* pengguna
Protokol Jaringan	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
Konfigurasi pengguna	Set AT + *instruksi*, *server* awan, *android/iOS app*
OS	*FreeRTOS*

2. Sensor ultrasonik

Sensor ultrasonik HC-SR04 merupakan pengukur jarak berbasis gelombang. Gelombang dipancarkan kemudian diterima balik oleh *receiver* ultrasonik. Jarak antara waktu pancar dan waktu terima direpresentasi dari

jarak objek yang sudah ada. Adapun pin yang digunakan untuk sensor ultrasonik yaitu:

Tabel 4.2 Penggunaan sensor ultrasonik (Sensor Irigasi Pembagian)

No.	**Pin/Soket Sensor ultrasonik**	**Keterangan**
1.	VCC	Dihubungkan ke pin 3V TTGO T-Call ESP32 SIM800L
2.	*GND*	Dihubungkan ke pin GND TTGO T-Call ESP32 SIM800L
3.	*Trigger*	Dihubungkan ke pin 15 TTGO T-Call ESP32 SIM800L
4.	*Echo*	Dihubungkan pin 0 TTGO T-Call ESP32 SIM800L

Tabel 4.3 Penggunaan sensor ultrasonik (Sensor Sawah 1)

No.	**Pin/Soket Sensor ultrasonik**	**Keterangan**
1.	VCC	Dihubungkan ke pin 3V TTGO T-Call ESP32 SIM800L
2.	*GND*	Dihubungkan ke pin GND TTGO T-Call ESP32 SIM800L
3.	*Trigger*	Dihubungkan ke pin 18 TTGO T-Call ESP32 SIM800L
4.	*Echo*	Dihubungkan pin 19TTGO T-Call ESP32 SIM800L

Tabel 4.4 Penggunaan sensor ultrasonik (Sensor Sawah 2)

No.	Pin/Soket Sensor ultrasonik	Keterangan
1.	VCC	Dihubungkan ke pin 3V TTGO T-Call ESP32 SIM800L
2.	*GND*	Dihubungkan ke pin GND TTGO T-Call ESP32 SIM800L
3.	*Trigger*	Dihubungkan ke pin 32 TTGO T-Call ESP32 SIM800L
4.	*Echo*	Dihubungkan pin 33 TTGO T-Call ESP32 SIM800L

3. Motor Servo

Motor *servo* merupakan perangkat atau aktuator putar yang dirancang dengan sistem kontrol umpan balik *loop* tertutup sehingga dapat diatur posisi sudut dari poros output motor servo. Adapun pin yang digunakan motor servo yaitu :

Tabel 4.5 Penggunaan pin motor servo (Servo Irigasi Pembagian)

No.	Pin/Soket motor *servo*	Keterangan
1.	Kabel merah VCC	Dihubungkan ke pin 3V TTGO T-Call ESP32 SIM800L
2.	Kabel merah GND	Dihubungkan ke pin GND TTGO T-Call ESP32 SIM800L
3.	Kabel kuning Data	Dihubungkan ke pin 14

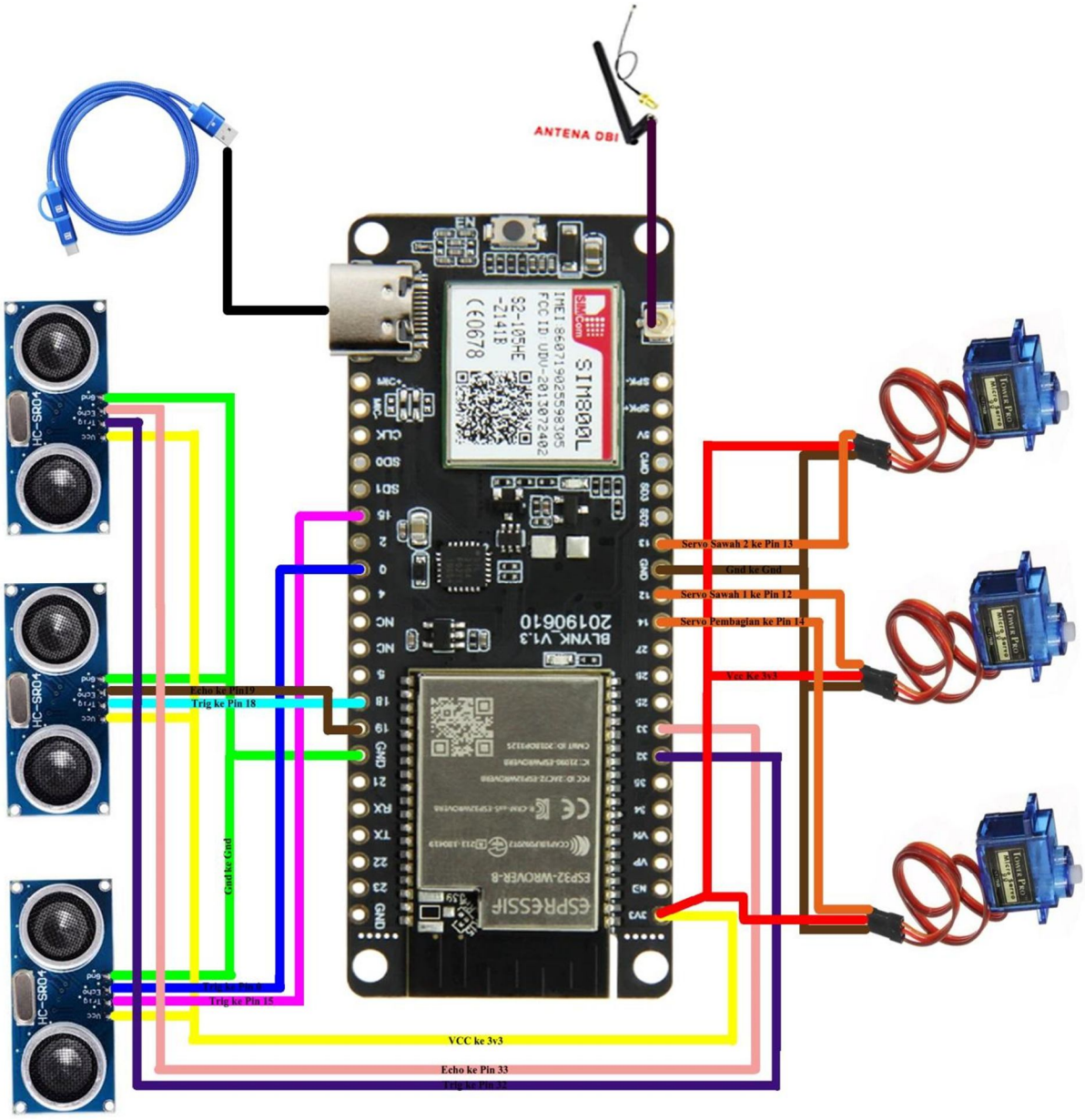
Tabel 4.6 Penggunaan pin motor servo (Servo Sawah 1)

No.	Pin/Soket motor *servo*	Keterangan
1.	Kabel merah VCC	Dihubungkan ke pin 3V TTGO T-Call ESP32 SIM800L
2.	Kabel merah GND	Dihubungkan ke pin GND TTGO T-Call ESP32 SIM800L
3.	Kabel kuning Data	Dihubungkan ke pin 12

Tabel 4.7 Penggunaan pin motor servo (Servo Sawah 2)

No.	**Pin/Soket motor *servo***	**Keterangan**
1.	Kabel merah VCC	Dihubungkan ke pin 3V TTGO T-Call ESP32 SIM800L
2.	Kabel merah GND	Dihubungkan ke pin GND TTGO T-Call ESP32 SIM800L
3.	Kabel kuning Data	Dihubungkan ke pin 13

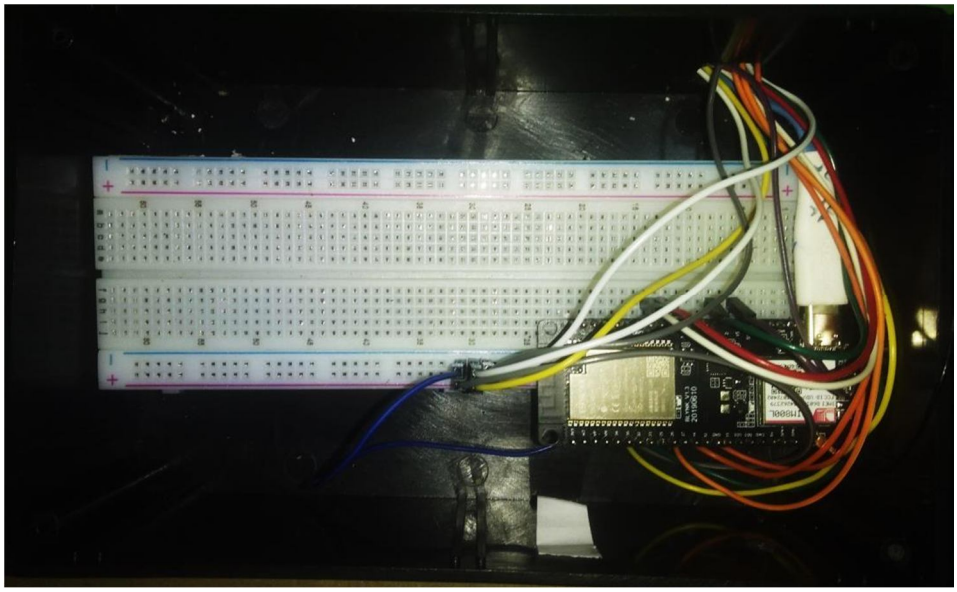
Adapun alur pengkabelan atau rangkaian perancangan sistem kontrol irigasi sawah berbasis *Internet of Things* (IoT) bisa kita lihat pada gambar dibawah.



Gambar 4. 10 Rangkaian Kabel Pengkabelan



Gambar 4. 11 Bentuk luar box



Gambar 4. 12 Bentuk dalam box

Dari gambar diatas yaitu rancangan sistem rangkaian elektronika dapat dilihat bahwa ada beberapa komponen yang digunakan yang saling terhubung satu sama lain.

F. Rancangan Perangkat Lunak (*Software*)

Setelah melakukan rancangan perangkat keras selanjutnya adalah melakukan perancangan perangkat lunak. Pada bagian perangkat lunak ini merupakan bagian pembuatan program yang akan di *upload* ke dalam *mikrokontroler* yaitu berupa perintah untuk mengelola data untuk diaplikasikan. Adapun bahasa pemrograman yang digunakan adalah berupa *sketch* program dalam bahasa C dan untuk membangun sistem kontrol jarak jauh irigasi sawah menggunakan pemrograman bahasa Html (*hypertext markup language)*.



irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
// Set serial
#define SerialMon Serial
// Set serial  AT commands (to SIM800 module)
#define SerialAT Serial1

// Configure TinyGSM library
#define TINY_GSM_MODEM_SIM800      // Modem is SIM800
#define TINY_GSM_RX_BUFFER   1024  // Set RX buffer to 1Kb


#include <Wire.h>
#include <TinyGsmClient.h>
#include <HCSR04.h>
#include <ESP32Servo.h>

//servo. Pin vcc = 3.3 volt
Servo myservo;
Servo myservo1;
Servo myservo2;
int servoSawah1 = 12;
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
int servoSawah2 = 13;
int servoPembagian = 14;

//ultrasonic. Pin vcc = 3.3 volt
UltraSonicDistanceSensor sawah1(18,19 );  // 18=triger, 19=echo
UltraSonicDistanceSensor sawah2(32, 33);  // 32=triger, 33=echo
UltraSonicDistanceSensor pembagi(15, 0);  // 15=triger, 0=echo
float sawah_1, sawah_2, pembagian;

#ifdef DUMP_AT_COMMANDS
  #include <StreamDebugger.h>
  StreamDebugger debugger(SerialAT, SerialMon);
  TinyGsm modem(debugger);
#else
  TinyGsm modem(SerialAT);
#endif

// I2C for SIM800 (to keep it running when powered from battery)
TwoWire I2CPower = TwoWire(0);
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
// TinyGSM Client for Internet connection
TinyGsmClient client(modem);

#define uS_TO_S_FACTOR 1000000     /* Conversion factor for micro seconds to seconds */
#define TIME_TO_SLEEP  3600        /* Time ESP32 will go to sleep (in seconds) 3600 seconds = 1 hour */

#define IP5306_ADDR          0x75
#define IP5306_REG_SYS_CTL0  0x00

bool setPowerBoostKeepOn(int en){
  I2CPower.beginTransmission(IP5306_ADDR);
  I2CPower.write(IP5306_REG_SYS_CTL0);
  if (en) {
    I2CPower.write(0x37); // Set bit1: 1 enable 0 disable boost keep on
  } else {
    I2CPower.write(0x35); // 0x37 is default reg value
  }
  return I2CPower.endTransmission() == 0;
}
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
String dataMasuk = "";
void setup() {
  myservo.attach(servoPembagian, 500, 2400);
  myservo1.attach(servoSawah1, 500, 2400);
  myservo2.attach(servoSawah2, 500, 2400);
  myservo.write(0);
  myservo1.write(0);
  myservo2.write(0);

  // Set serial monitor debugging window baud rate to 115200
  SerialMon.begin(115200);

  dataMasuk.reserve(200);

  // Start I2C communication
  I2CPower.begin(I2C_SDA, I2C_SCL, 400000);

  // Keep power when running from battery
  bool isOk = setPowerBoostKeepOn(1);
  SerialMon.println(String("IP5306 KeepOn ") + (isOk ? "OK" : "FAIL"));
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
// Set modem reset, enable, power pins
pinMode(MODEM_PWKEY, OUTPUT);
pinMode(MODEM_RST, OUTPUT);
pinMode(MODEM_POWER_ON, OUTPUT);
digitalWrite(MODEM_PWKEY, LOW);
digitalWrite(MODEM_RST, HIGH);
digitalWrite(MODEM_POWER_ON, HIGH);

// Set GSM module baud rate and UART pins
SerialAT.begin(115200, SERIAL_8N1, MODEM_RX, MODEM_TX);
delay(3000);

// Restart SIM800 module, it takes quite some time
// To skip it, call init() instead of restart()
SerialMon.println("Initializing modem...");
modem.restart();

// Unlock your SIM card with a PIN if needed
if (strlen(simPIN) && modem.getSimStatus() != 3 ) {
  modem.simUnlock(simPIN);
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
}

SerialMon.print("Connecting to APN: ");
SerialMon.print(apn);
if (!modem.gprsConnect(apn, gprsUser, gprsPass)) {
  SerialMon.println(" fail");
}
else {
  SerialMon.println(" OK");

  SerialMon.print("Connecting to ");
  SerialMon.print(server);
   if (!client.connect(server, port)) {
   SerialMon.println(" fail");
       }
  else {
  SerialMon.println(" OK");
  }
}}
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
void loop() {
  delay(500);
  request();
  send();
}

void request() {

    SerialMon.print("Request Data dari Web Server ");

    SerialAT.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"");
    delay(800);

    SerialAT.println("AT+SAPBR=3,1,\"APN\",\"telkomsel\"");
    delay(800);

    SerialAT.println("AT+SAPBR=3,1,\"PWD\",\"wap123\"");
    delay(800);

    SerialAT.println("AT+SAPBR=1,1");
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
    SerialAT.println("AT+SAPBR=1,1");
    SerialMon.print(" .");
    delay(1000);

    SerialAT.println("AT+HTTPINIT");
    SerialMon.print(" .");
    delay(1000);

    SerialAT.println("AT+HTTPPARA=\"URL\",\"irigasi.iot-labs.xyz/kontrol.php?Request=OK\"");
    SerialMon.println(" .");
    delay(1000);

    SerialAT.println("AT+HTTPACTION=0");
    delay(10000);

    SerialAT.println("AT+HTTPREAD");
    GetResponse(5000);

    SerialAT.println("AT+HTTPTERM");
    delay(800);
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
    SerialAT.println("AT+SAPBR=0,1");
    delay(800);

}

void GetResponse(int wait){
  String dataMasuk = "";
  SerialMon.print("response : ");
  long timeNOW2 = millis();
  while(millis()-timeNOW2 < wait){
    while(SerialAT.available()>0){
      dataMasuk += (char)SerialAT.read();
    }
    delay(1);
  }
    SerialMon.print("Full data :");
    SerialMon.println(dataMasuk);
    String data0 = parse(dataMasuk, '#', 0);
    String data1 = parse(dataMasuk, '#', 1);
    String data2 = parse(dataMasuk, '#', 2);
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
    if (data1=="0"){
      myservo1.write(180);
      }


    if (data2=="0"){
      myservo2.write(180);
      }

    SerialMon.println("------------------------------------------------------------------");

    SerialMon.print("data 0 :");
    SerialMon.println(data0);

    SerialMon.print("data 1 :");
    SerialMon.println(data1);

    SerialMon.print("data 2 :");
    SerialMon.println(data2);
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
    SerialMon.println("------------------------------------------------------------------");

    SerialMon.println();
}

String parse(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = {0, -1};
  int maxIndex = data.length()-1;

  for(int i=0; i<=maxIndex && found<=index; i++){
    if(data.charAt(i)==separator || i==maxIndex){
        found++;
        strIndex[0] = strIndex[1]+1;
        strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
  }

 return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
void send(){
   //pembacaan sensor ultrasonic
    sawah_1=  2.47-sawah1.measureDistanceCm();
    sawah_2 = 2.47-sawah2.measureDistanceCm();
    pembagian = 10.00-pembagi.measureDistanceCm();

    SerialMon.print("Tinggi Air Sawah 1 : ");
    SerialMon.print(sawah_1);
    SerialMon.println(" cm");
    SerialMon.print("Tinggi Air Sawah 2 : ");
    SerialMon.print(sawah_2);
    SerialMon.println(" cm");
    SerialMon.print("Tinggi Air Pembagian : ");
    SerialMon.print(pembagian);
    SerialMon.println(" cm");


  //kontrol servo sawah
    String status1, status2;
 if( sawah_1>=1.00 )
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

```
void send(){
   //pembacaan sensor ultrasonic
    sawah_1=  2.47-sawah1.measureDistanceCm();
    sawah_2 = 2.47-sawah2.measureDistanceCm();
    pembagian = 10.00-pembagi.measureDistanceCm();

    SerialMon.print("Tinggi Air Sawah 1 : ");
    SerialMon.print(sawah_1);
    SerialMon.println(" cm");
    SerialMon.print("Tinggi Air Sawah 2 : ");
    SerialMon.print(sawah_2);
    SerialMon.println(" cm");
    SerialMon.print("Tinggi Air Pembagian : ");
    SerialMon.print(pembagian);
    SerialMon.println(" cm");


  //kontrol servo sawah
    String status1, status2;
  if( sawah_1>=1.00 )
```

irigasi | Arduino 1.8.13

File Edit Sketch Tools Help

irigasi

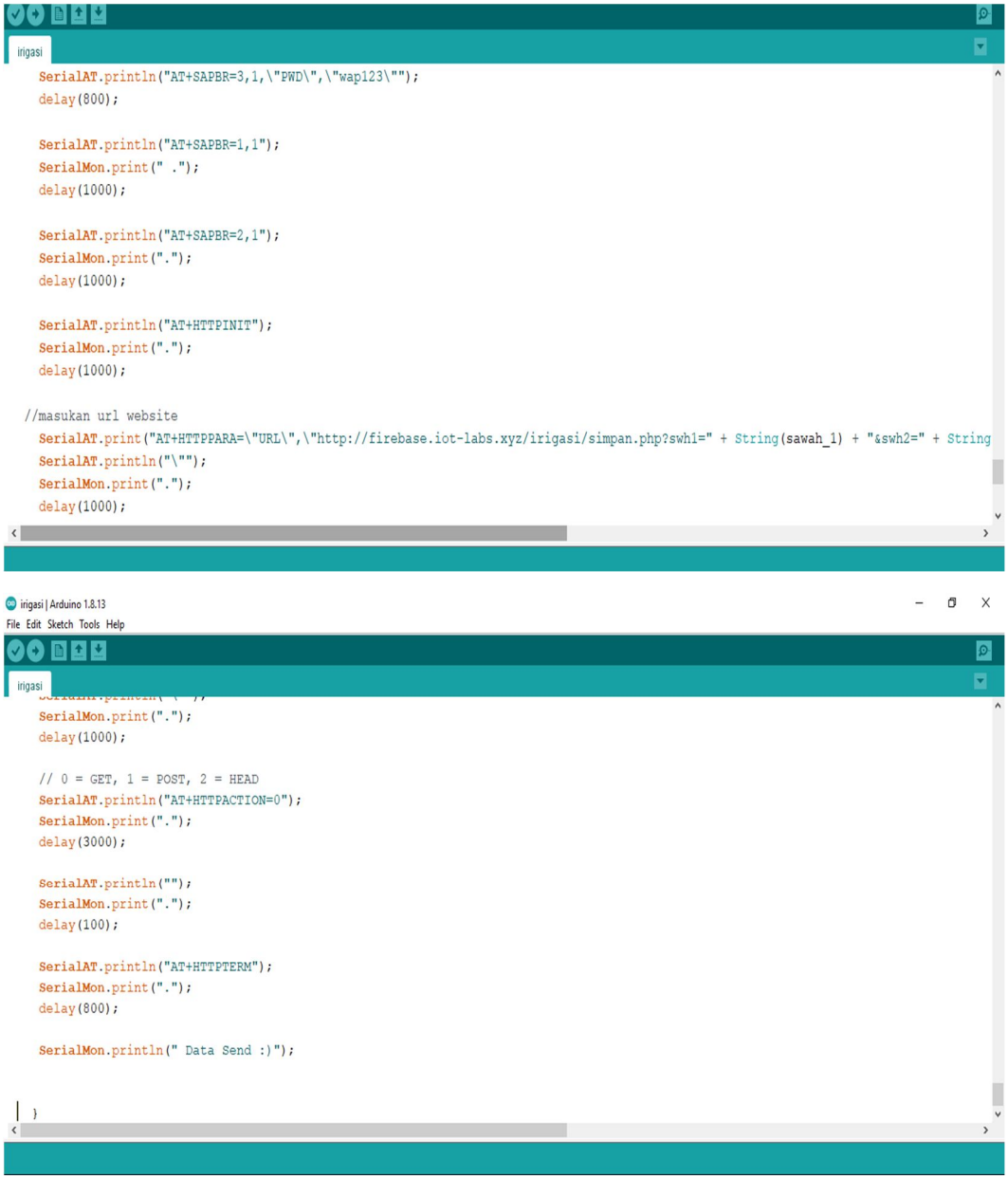
```
    {
     myservo.write(100);
    }else if(pembagian<=3.00 || pembagian>1.00)
    {
      myservo.write(45);
    }else if(pembagian<=1.00){
     myservo.write(15);
    }

    SerialMon.print("Send Data Sensor to Server");

    SerialAT.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"");
    delay(800);

    SerialAT.println("AT+SAPBR=3,1,\"APN\",\"telkomsel\"");
    delay(800);

    SerialAT.println("AT+SAPBR=3,1,\"PWD\",\"wap123\"");
    delay(800);
```



Gambar 4. 13 Rancangan Perangkat Lunak

Pada gambar *arduino ide* (*sketch*) sebuah *software* untuk memprogram *arduino* dan komponen pendukung lainnya dengan menggunakan *sintaks* pemrograman bahasa C. *Arduino ide* dibuat dari bahasa pemrograman Java yang dilengkapi dengan *library* C/C++ yang membuat operasi *input/output* lebih mudah.

G. Rancangan Aplikasi *Web*

Prototype Kontrol jarak jauh irigasi sawah merupakan rancangan monitoring otomatis yang dituang kedalam bentuk *prototype* dengan proses kerja secara otomatis.

1. Tampilan menu utama

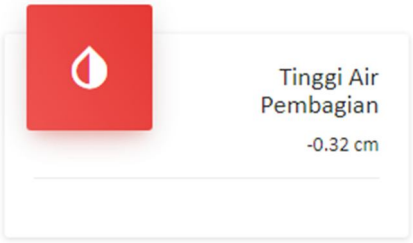
   Menu utama terdiri dari menu *Home*, *History*. Dengan adanya aplikasi ini diharapkan pengguna lebih mudah monitoring tinggi air irigasi sawah.



Gambar 4.14 Tampilan menu utama

2. Tampilan keadaan tinggi air irigasi pembagian

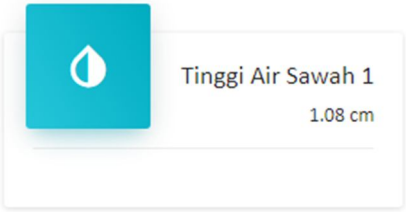
Tampilan ini merupakan tampilan berapa cm tinggi air pada irigasi pembagian.



Gambar 4.15 Keadaan tinggi air pembagian.

3. Tampilan keadaan tinggi air sawah 1

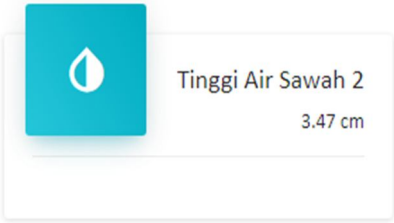
Tampilan ini merupakan tampilan berapa cm tinggi air pada sawah 1.



Gambar 4.16 Keadaan tinggi air sawah 1

4. Tampilan keadaan tinggi air sawah 2

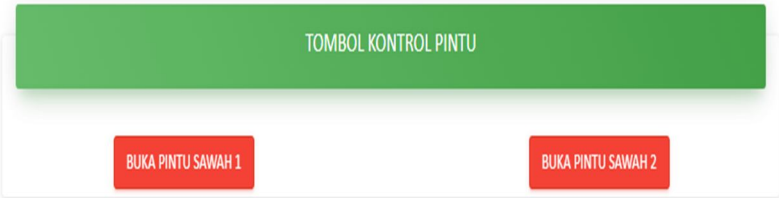
Tampilan ini merupakan tampilan berapa cm tinggi air pada sawah 2



Gambar 4.17 Keadaan tinggi air sawah 2

5. Tampilan tombol pintu buka pada sawah 1 dan 2

Tampilan ini merupakan tampilan untuk tombol kontrol untuk membuka pintu sawah 1 dan 2.



Gambar 4.18 tombol pintu buka pada sawah 1 dan 2

H. Pembahasan

1. Metode pengujian

Metode pengujian sistem merupakan proses eksekusi suatu program atau sistem keseluruhan. Pengujian menunjukan bahwa fungsi perangkat keras dan perangkat lunak bekerja sesuai dengan spesifikasi bahwa persyaratan kinerja telah terpenuhi.

2. Pengujian *Black box*

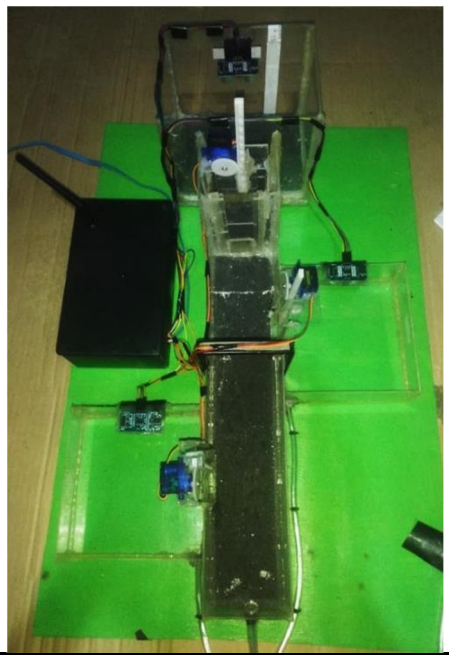
Pengujian *Black-box* merupakan suatu pengujian pada *interface* perangkat lunak. Pengujian *Black-box* kemungkinan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Perlu juga diketahui bahwa pengujian *Black-box* juga memperhatikan struktur kontrol. Meskipun didesain sedemikian rupa untuk mengungkap kesalahan. Pengujian *Black-box* digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak adalah operasional

bahwa *input* diterima dengan baik dan output dihasilkan dengan tepat serta integritas informasi *external* (misalnya file data).

Pengujian *black*-box menguji beberapa aspek dasar suatu sistem dengan sedikit memperhatikan struktur *logika internal* perangkat lunak tersebut.

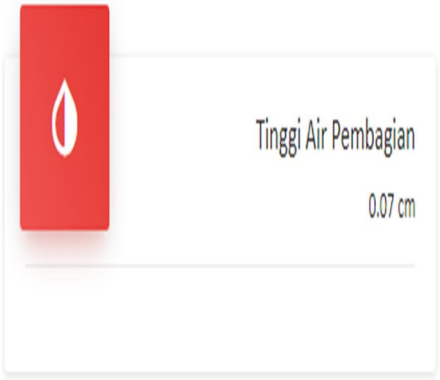
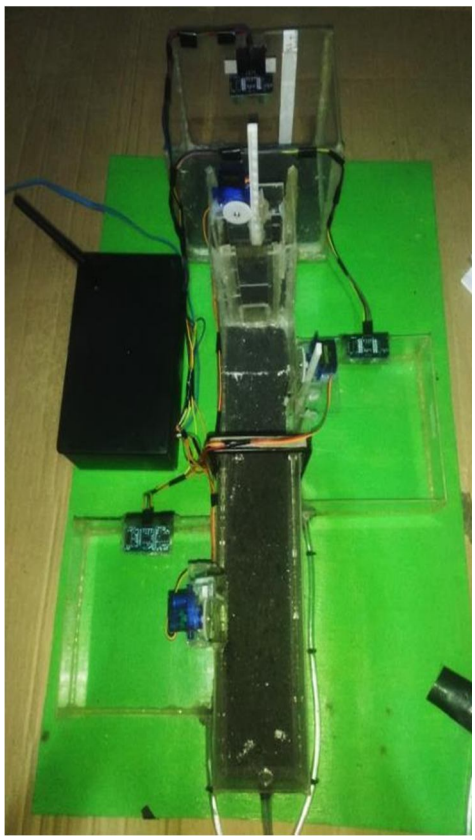
a. Pengujian tahap pertama

Tabel 4.8 Pengujian tahap pertama

Uji coba	Hasil	Keterangan
Jika sensor mendeteksi tinggi air ≥ 4 cm pada irigasi pembagian maka motor servo akan berputar 180º secara otomatis.	√	Berhasil, karena sensor mendeteksi tinggi air sehingga motor servo berputar sesuai sesuai yang ditentukan.
Foto dan *Screenshot*		
 		

b. Pengujian tahap kedua

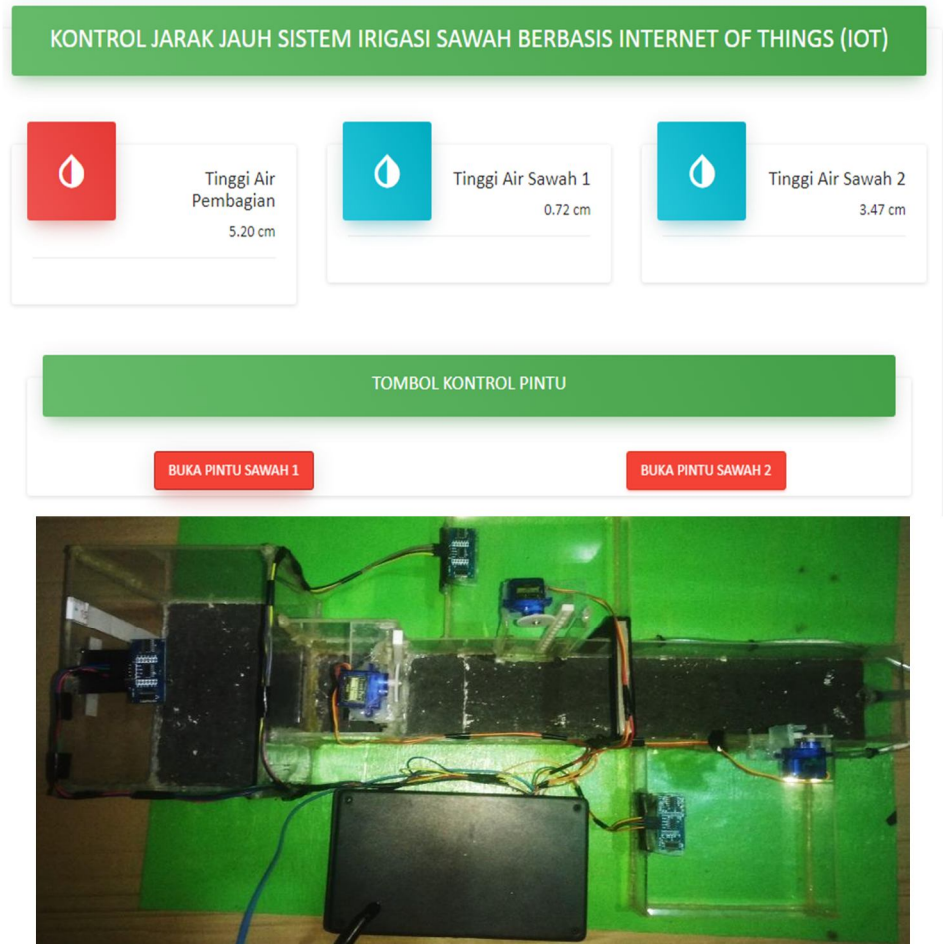
Tabel 4.9 Pengujian tahap kedua

Uji coba	Hasil	Keterangan
Jika sensor mendeteksi tinggi air ≤ 1 cm pada irigasi pembagian maka motor servo akan berputar 45º secara otomatis.	√	Berhasil, karena sensor mendeteksi tinggi air sehingga motor servo berputar sesuai sesuai yang ditentukan.
Foto dan *Screenshot*		
 		

c. Pengujian tahap ketiga

Tabel 4.10 Pengujian tahap ketiga

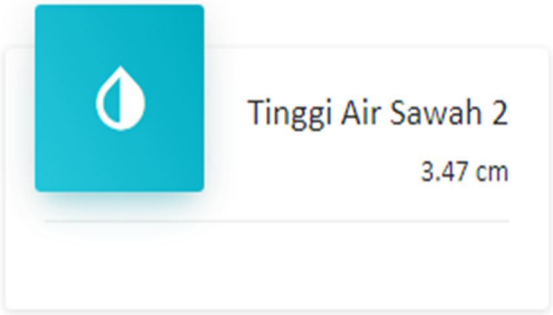
Uji coba	Hasil	Keterangan
Jika sensor mendeteksi tinggi air ≤ 1 cm pada sawah 1 dan 2 maka tombol buka aktif secara manual	√	Berhasil, karena sensor mendeteksi tinggi air yang sudah di tentukan sehingga pintu terbuka secara manual setelah menekan tombol perintah buka pintu sawah
Foto dan *Screenshot*		

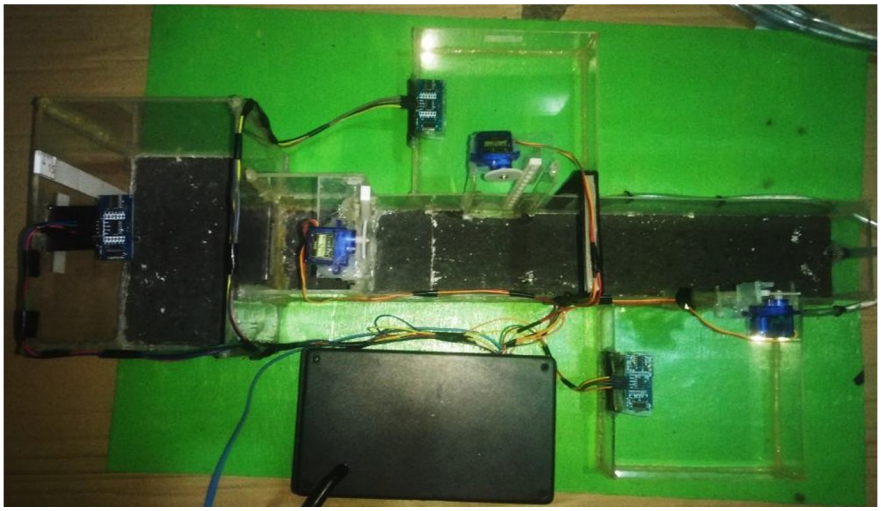


d. Pengujian tahap keempat

Tabel 4.11 Pengujian tahap keempat

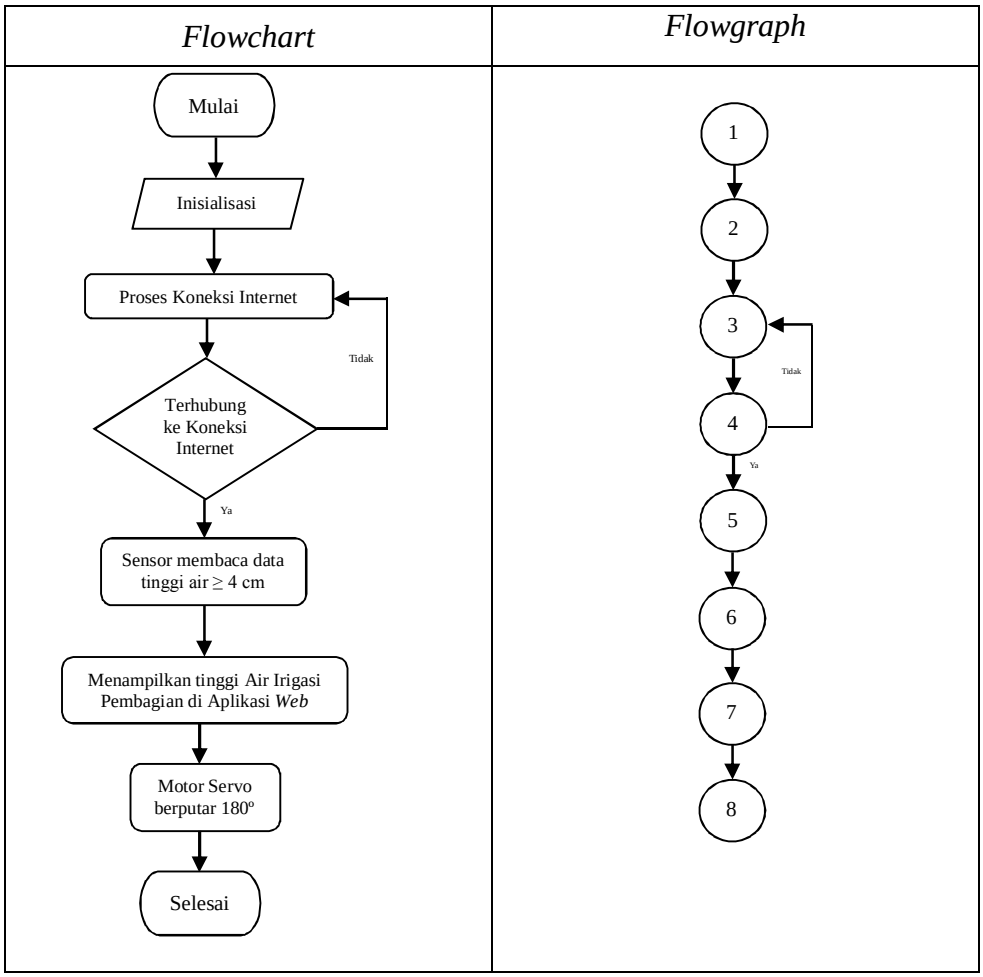
Uji coba	Hasil	Keterangan
Jika sensor mendeteksi tinggi air 3,47 cm pada sawah maka pintu akan tertutup otomatis	√	Berhasil, karena sensor mendeteksi tinggi air yang sudah di tentukan sehingga pintu tertutup secara otomatis
Foto dan *Screenshot*		





3. Pengujian *White Box*

a. *Halaman Irigasi pembagian*



Gambar 4. 19 *Flowchart* dan *Flowgrap* Irigasi

Dari gambar *Flowgraph* Irigasi Pembagian di atas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung C*yclomatic Complexcity* V(G) dari *Egde* dan *Node*:

DenganRumus : V (G) = E – N + 2

Dengan E (*edge*) = 8

N (*Node*) = 8

P (Predikat *Node*) = 1

Penyelesaian : V(G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat *Node* (P) = P + 1

= 1 + 1

= 2

2) Berdasarkan perhitungan C*yclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2

3) *Independent Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 4 – 3

Path 2 = 1 – 2 – 3 – 4 – 5– 6 – 7 – 8

4) *Graph Matrix*

5)

	1	2	3	4	5	6	7	8	Koneksi
1		1							1-1= 0
2			1						1-1= 0
3				1					1-1= 0
4			1		1				2-1= 1
5						1			1-1= 0
6							1		1-1= 0
7								1	1-1= 0
8									

Cyclomatic Complexity = 1+1 =2

b. Skala pada irigasi sawah

Tabel 4.12 Skala pada Miniatur Irigasi Pembagian Sawah

Nama	Cm	Skala
Tinggi tembok dari dasar ke permukaan	72 cm	13 cm
Tinggi saluran pembagian air	40 cm	4 cm
Tinggi pembukaan pintu pembagian irigasi	10 – 40 cm	1 – 4 cm
Tinggi pembukaan pintu sawah	15 cm	1,5 cm
Tinggi air masuk ke sawah	2-5 cm	3 cm
Tinggi Pematangan Sawah	25 cm	4 cm

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dijelaskan pada bab- bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem kontrol irigasi sawah berbasis *Internet of Things* (IoT) menghasilkan jika tinggi air pembagian < 1 cm maka motor servo akan berputar 45°, dan jika tinggi air pembagian > 4-10 cm maka motor servo akan berputar 180°, kemudian jika tinggi air pada sawah mencapai 3,47 cm maka pintu akan tertutup otomatis, jika tinggi air pada sawah < 1 cm maka tombol pintu sawah akan di kontrol secara manual di aplikasi *web*.
2. Proses pengontrolan akan terjadi *delay*, dikarenakan aplikasi mengirim data buka/atau tutup terlebih dahulu ke *web hosting* kemudian mikrokontroler mengambil data yang dikirimkan oleh aplikasi *web* untuk menggerakkan portal, dalam proses tersebut konektivitas kecepatan upload dari jaringan yang digunakan sangat berpengaruh.
3. Semakin tinggi kecepatan *upload* dari *konektifitas* yang digunakan maka *delay* pengontrolan akan semakin rendah.
4. Sim yang digunakan untuk tersambung ke *internet shield* menggunakan jasa operator telkomsel.
5. Untuk masuk ke aplikasi kontrol jarak jauh irigasi sawah *berbasis internet of things* menggunakan link *http://irigasi.iot-labs.xyz/*.

B. Saran

Adapun saran yang kami berikan sebagai berikut:

1. Diharapkan untuk pembuatan aplikasi selanjutnya dapat meningkatkan penggunaan aplikasinya dengan membuatkan aplikasi berbasis *android*.
2. Jika ingin mendapatkan hasil konektivitas yang stabil dan pengiriman data yang cepat maka diharapkan memakai alat yang mendukung jaringan 4G.

DAFTAR PUSTAKA

Andrie Wijaya, Muhammad Rivai. 2018 "*Monitoring dan kontrol sistem irigasi berbasis IoT menggunakan banana Pi*", Departemen Teknik Elektro, Institut Teknologi Sepuluh November, diakses 12 November 2019.

Ahmad Shukri Mohd.2015:5 "*Pengertian Perancangan"*, Penerbit buku UTM, Yogyakarta.

Bagus Hari Sasongko.2015 "*Mikrokontroler*".

Balittanah.Litbang.Pertanian.Go.Id/Ind/Dokumentasi/Buku/Buku%20lahan%20sawah/08peluang_Perluasan_Lahan_Sawah.Pdf

Burange, A. W., & Misalkar, H. D. (2015). "*Review of Internet of Things in Development of Smart Cities with Data Management & Privacy"*.

Ebraheim Alsaadi, Abdallah Tubaishat. (2015). *Internet of Things: Features, Challenges.*

Eljire Bagas Lewi.15 Agustus 2012 "Sistem *Monitoring* Ketinggian Air Berbasis *Internet Of Things* Menggunakan *Google Firebase*.

Hemlata Channel, Sukhesh Kothari, Dipali Kadam. (2015). *Multidisciplinary Model for Smart Agriculture using Internet-of-Things (IoT), Sensors, Cloud-Computing, Mobile-Computing & Big-Data Analysis.* Department of CE, PICT, Pune, India.

I Wayan Suastika, Basaruddin N. Tumarlan T. (2015). *Budidaya Padi Sawah Di Lahan Pasang Surut.*

Jogiyanto & Magistro : 662-663,2015 "Bagian alir *Flowchart*" menurut ANSI (*American National Standard Institute*).

Kadir Abdul.2015 "Sasaran Sistem" Penerbit buku Andi, Yogyakarta.

Kevin Aston dan Peter Waher.2015 " *Internet of things (IOT)"*

Literature Review. India : *National Institute of Industrial Engineering* (NITIE).

Muhammad rifai dan Suwito. 2016. "*Rancang bangun sistem irigasi tanaman otomatis menggunakan wireless sensor network*", Institut Teknologi Sepuluh September, diakses 12 Juli 2020.

Muchlisin Riadi. (2018) *"pengertian-tujuan-dan-jenis-jenis-irigasi"*

Maimunah.2017:38 "Pengertian Perancangan", Penerbit buku UTM, Yogyakarta.

M.U. Farooq, Sadia Mazhar, Anjum Khairi, Talha Kamal. (2015). A *Review on the Internet of Things (IoT).*

Nurul Hidayah, "*Konsep Dasar Sistem"* 22 Agustus 2016

Sudirman Said. 2015 "*Rancang bangun sistem otomatisasi irigasi pipa lahan sawah berbasis tenaga surya"*, program studi teknik sipil dan Lingkungan SPs Institut Pertanian Bogor, diakses 12 November 2019.

Sumardi Said.Universitas Sumatera Utara Medan 14 Juli 2017 "Sistem *Monitoring* Level Ketinggian Air Bendungan Menggunakan Sensor Ultrasonik Yang Di Kontrol Melalui *Android*".

Somayya Madakam, R. Ramaswamy, Siddharth Tripathi. (2015). *Internet of Things (IoT): A*

Shubham Bhatia, Abhishek Chauhan, Vaibhav K. Nigam. (2016). *The Internet of Things: A Survey on Technology and Trends.*

Yasha. (3 Agustus 2018), *"Internet of Things"*, Panduan Lengkap. Diperoleh dari: https://www.*dewaweb*.com/blog/internet-of-things/

Yosua.2015 "Data Diagram *Flow* Diagram"

Vulnerabilities. International Journal of Advanced *Computer Science* and *Information Technology* (IJACSIT). Zayed University.