**TUGAS AKHIR**

**KENDALI LAMPU SECARA JARAK JAUH**

**MENGGUNAKAN WI-FI BERBASIS ANDROID**



**Diajukan sebagai salah satu syarat akademik untuk memperoleh gelar Ahli Madya progam Diploma III pada program studi Teknik Elektro**

**Disusun oleh**

**YULIANA PAMUNGKAS**

**13.41556**

**POLITEKNIK PRATAMA MULIA**

**SURAKARTA**

**2016**

**HALAMAN PERSETUJUAN**

**Judul**

**KENDALI LAMPU SECARA JARAK JAUH**

**MENGGUNAKAN WI-FI BERBASIS ANDROID**

**Disusun Oleh**

**Yuliana Pamungkas**

**13.41556**

**Mengesahkan**

**Direktur,**

**Drs. Sunaryo, S.T., M.T.**

**NIK:195504241983091002**

**Disetujui**

**Dosen Pembimbing,**

**Yusuf Eko Rohmadi, S.Kom., M.Eng NIK: 1502.214**

**ii**

**HALAMAN PENGESAHAN**

**KENDALI LAMPU SECARA JARAK JAUH**

**MENGGUNAKAN WI-FI BERBASIS ANDROID**

**Dipertahankan di depan Penguji Tugas Akhir**

**Politeknik Pratama Mulia Surakarta**

**Program Studi Teknik Elektro**

**Dan diterima untuk memenuhi sebagai syarat**

**Guna memperoleh Gelar Ahli Madya Diploma III**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Pada Hari** | **: Selasa** |  |  |
|  | **Tanggal** | **: 21 Juni 2016** |  |  |
| **Penguji:** | |  |  |  |
| **1.** | **Drs. Muhammad Alhan, S.T., M.Eng** | | **(** | **)** |
| **2.** | **Yusuf Eko Rohmadi, S.Kom., M.Eng** | | **(** | **)** |

**Ketua,** **Sekretaris,**

**Yaya Finayani, S.T., M.Eng**

**NIK:1593.078**

**Salechan, S.T., M.Kom**

**NIK: 1592.052**

**iii**

**MOTTO**

**“Buanglah jauh-jauh rasa malu untuk sebuah ilmu yang baik”**

**“Ilmu itu ibarat buruan, sedangkan tulisan adalah talinya, ikatlah**

**buruanmu dengan tali yang kokoh”**

**(Imam Syafi’i)**

**iv**

**HALAMAN PERSEMBAHAN**

**Tugas Akhir ini kupersembahkan kepada:**

**1. Allah subhanahu wa ta’ala yang telah memberikan kelancaran atas terselesainya Tugas Ahir ini.**

**2. Ibuku tercinta “Triyani” yang dengan keiklasan dan kesusah payahannya telah memberikan segalanya untukku.**

**3. Kedua kakakku “Mbak Ita dan Mbak Dwi” yang selalu memberikan motivasinya.**

**4. Dosen pembimbing sekaligus kakak iparku “Mas Yusuf” yang dengan sabar membimbing dan memberi masukan dalam pembuatan Tugas Ahir.**

**5. Dosen-dosen Teknik Elektro, terimakasih atas bimbingannya selama masa kuliah.**

**6. Semua teman-teman Teknik Elektro angkatan 2013 yang telah memberi dorongan.**

**7.** **Sahabatku “Hilya, Rina, Vivi, Husna, Farah” untuk do’a dan semangatnya.**

**v**

**PRAKATA**

**Puji syukur kehadirat Alloh Subhanahu wata’ala yang telah memberikan rahmat dan barokah-Nya sehingga Tugas Akhir dengan judul “Kendali Lampu Secara Jarak Jauh Menggunakan Wi-Fi Berbasis Android” dapat diselesaikan dengan baik. Laporan Tugas Akhir ini disusun guna memenuhi syarat untuk memperoleh gelar Ahli Madya pada program studi Teknik Elektro Politeknik Pratama Mulia Surakarta.**

**Dalam menyelesaikan Tugas Akhir ini penulis mendapatkan banyak dukungan dan bantuan dari banyak pihak. Untuk itu penulis mengucapkan banyak terimakasih yang tak terhingga kepada:**

1. **Bapak Yusuf Eko Rohmadi, S.Kom., M.Eng selaku dosen pembimbing tugas akhir yang telah memberikan bimbingan terbaiknya.**
2. **Ibu Yaya Finayani, S.T., M.Eng selaku Ketua Jurusan Teknik Elektronika dan Bapak Salechan, S.T., M.Kom selaku Sekretaris Jurusan Teknik Elektronika yang telah memberikan kemudahan dalam penyelesaian tugas akhir ini.**
3. **Bapak Sudarno, S.T., M.Eng selaku dosen Teknik Elektronika yang telah memberikan bimbingannya dalam hal pemrograman.**
4. **Bapak Drs. Sunaryo, S.T., M.Eng selaku Direktur Politeknik Pratama Mulia Surakarta yang telah memberikan kesempatan untuk belajar.**
5. **Segenap Dosen di Jurusan Teknik Elektronika yang telah memberikan ilmunya kepada penulis.**

**vi**

1. **Segenap staf Karyawan dan Karyawati Politeknik Pratama Mulia yang telah memberikan banyak bantuannya dalam proses belajar.**
2. **Ibuku Triyani yang telah memberikan segala dukungannya sehingga penulis dapat menyelesaikan studi D-3.**
3. **Rekan-rekan Teknik Elektronik angkatan 2013 yang selalu memberikan dukungan dan motivasinya, serta semua pihak yang tidak bisa penulis sebutkan satu persatu.**

**Penulis menyadari bahwa laporan tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan. Akhir kata semoga laporan tugas akhir ini dapat memberikan manfaat serta memberikan wawasan tambahan kepada para pembaca dan khususnya bagi penulis sendiri.**

**Surakarta,** **Juni 2016**

**Penulis,**

**Yuliana Pamungkas**

**vii**

**INTISARI**

**Perangkat Android menjadi salah satu tren teknologi mobile smartphone yang saat ini banyak digunakan oleh masyarakat. Fitur-fitur yang ditanamkan di dalamnya membuat perangkat tersebut dapat dimanfaatkan di luar fungsinya sebagai perangkat seluler, misalnya sebagai pengendali perangkat lain. Wi-Fi adalah salah satu media transfer data secara wireless dengan jangkauan yang luas dibanding media wireless lainnnya. Tujuan dari tugas akhir ini adalah mengembangkan sistem kendali lampu menggunakan media Wi -Fi berbasis Android.**

**Kendali lampu melalui Android memanfaatkan mikrokontroler pada kit Arduino Uno sebagai komunikator dengan lampu. Media wireless antara Arduino dengan Android menggunakan modul Wi -Fi ESP8266-12-12 melalui alamat IP addres. Sedangkan antara ESP dan Arduino Uno menggunakan komunikasi secara serial, yaitu melalui Tx dan Rx. Sistem kerjanya adalah Android memanggil alamat IP addres yang berisi pesan On/Off pada ESP dan diteruskan kepada Arduino Uno untuk mengendalikan nyala lampu.**

**Sistem kendali lampu dengan nama SKLW dapat berfungsi dengan baik. Modul ESP8266-12 memberikan jangkauan kendali yang baik jika ditempatkan pada lingkungan terbuka dengan jangkauan hingga 30 meter.**

**Kata kunci: Android, Arduino Uno, ESP8266-12, Sistem kendali lampu**

**viii**

**ABSTRACT**

**Today, device based Android called “smartphone” is most of the mobile technology that to become trend in the people lifestyle. There are many features embedded to its system, it make an Android device can be utilized beyond its function as a cellular device, such as controlling other devices. Wi-Fi is one of the wireless data transfer wich have large range than the others. The object in this paper is to develop a light control system using a Wi-Fi based on Android.**

**Int this control system, the microcontroller in a Arduino Uno board used to communicate with light. Then, communication between Arduino Uno and Android using wi-fi modul ESP8266-12-12. While, communication between ESP dan Arduino using serial configuration through Tx and Rx pin. The Android app will againts the IP address of ESP8266-12-12 that contain On/Off data messages then forwarded to the Arduino Uno to control lights**

**The light control system namely SKLW worked properly. The wi-fi modul ESP8266-12-12 provides a good range up to 30 meters in outdoor environment.**

**Keywords: Android, Arduino Uno, ESP8266-12, light control system**

**ix**

**DAFTAR ISI**

**HALAMAN JUDUL ------------------------------------------------------------------------ i**

**HALAMAN PERSETUJUAN -------------------------------------------------------------ii**

**HALAMAN PENGESAHAN ------------------------------------------------------------ iii**

**MOTTO -------------------------------------------------------------------------------------- iv**

**HALAMAN PERSEMBAHAN ----------------------------------------------------------- v**

**PRAKATA ----------------------------------------------------------------------------------- vi**

**INTISARI ---------------------------------------------------------------------------------- viii**

**ABSTRACT --------------------------------------------------------------------------------- ix**

**DAFTAR ISI --------------------------------------------------------------------------------- x**

**DAFTAR GAMBAR ---------------------------------------------------------------------- xii**

**DAFTAR TABEL ------------------------------------------------------------------------ xiii**

**DAFTAR LAMPIRAN ------------------------------------------------------------------ xiv**

**BAB I PENDAHULUAN ------------------------------------------------------------------ 1**

**1.1** **Latar Belakang -------------------------------------------------------------------- 2**

**1.2** **Perumusan Masalah -------------------------------------------------------------- 6**

**1.3** **Tujuan Tugas Akhir -------------------------------------------------------------- 6**

**1.4** **Manfaat Tugas Akhir ------------------------------------------------------------- 7**

**BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI ------------------------ 8**

**2.1** **Kendali Wireless ------------------------------------------------------------------ 8**

**2.2** **Teknologi Wireless --------------------------------------------------------------- 9**

**2.3** **Android --------------------------------------------------------------------------- 13**

**2.4** **Arduino --------------------------------------------------------------------------- 15**

**2.5** **Modul ESP8266-12 ------------------------------------------------------------- 19**

**2.6** **Program Basic for Android (B4A) ------------------------------------------- 20**

**BAB III METODE PERANCANGAN SISTEM ------------------------------------- 29**

**3.1** **Alat dan Bahan ------------------------------------------------------------------ 29**

**3.2** **Langkah Kerja ------------------------------------------------------------------- 31**

**3.3** **Skenario Pengujian ------------------------------------------------------------- 33**

**x**

**BAB IV HASIL DAN PEMBAHASAN ----------------------------------------------- 35**

**4.1** **Rancangan Sistem Kendali Lampu ------------------------------------------- 35**

**4.2** **ESP8266-12 sebagai Webserver ---------------------------------------------- 37**

**4.3** **Aplikasi Kendali Lampu pada Android ------------------------------------- 42**

**4.4** **Pemrograman pada Arduino Uno -------------------------------------------- 45**

**4.5** **Driver Lampu dengan MOC3041 -------------------------------------------- 47**

**4.6** **Sumber Tegangan DC ---------------------------------------------------------- 48**

**4.7** **Cara Kerja Alat ------------------------------------------------------------------ 49**

**4.8** **Ujicoba Sistem ------------------------------------------------------------------ 49**

**BAB V KESIMPULAN DAN SARAN ------------------------------------------------ 51**

**5.1** **Kesimpulan ---------------------------------------------------------------------- 51**

**5.2** **Saran ------------------------------------------------------------------------------ 51**

**DAFTAR PUSTAKA --------------------------------------------------------------------- 52**

**LAMPIRAN -------------------------------------------------------------------------------- 53**

**xi**

**DAFTAR GAMBAR**

**Gambar 2.1 Model umum komunikasi data secara wireless** **9**

**Gambar 2.2 Simbol Bluetooth** **10**

**Gambar 2.3 Tag dan Reader RFID** **12**

**Gambar 2.4 Arsitektur Sistem Operasi Android** **14**

**Gambar 2.5 Klasifikasi produk Arduino di negara USA** **15**

**Gambar 2.6 Bagian-bagian Arduino Uno** **16**

**Gambar 2.7 Tampilan Arduino IDE** **17**

**Gambar 2.8 Modul ESP8266 tipe 12** **20**

**Gambar 2.9 USB to TTL** **20**

**Gambar 2.10 Konfigurasi Path** **23**

**Gambar 2.11 Tampilan awal lembar kerja B4A** **23**

**Gambar 2.12 Tampilan Designer** **24**

**Gambar 2.13 Tampilan Generate Members** **25**

**Gambar 2.14 Tampilan tempat source code dituliskan** **25**

**Gambar 2.15 Tampilan AVD Manager** **26**

**Gambar 2.16 Tampilan pembuatan Emulator** **27**

**Gambar 3.1 Topologi Kendali Lampu secara Wireless** **29**

**Gambar 4.1 Flowchart Koneksi Android dengan modul ESP8266-12** **35**

**Gambar 4.2 Flowchart kontrol ON/OFF lampu** **36**

**Gambar 4.3 Instalasi Board ESP8266-12 Arduino Ide** **38**

**Gambar 4.4 Skema USB to Serial dan ESP8266-12** **39**

**Gambar 4.5 Properti dari ESPap** **40**

**Gambar 4.6 Tampilan Web Browser pada ESP8266-12** **42**

**Gambar 4.7 Desain Aplikasi Kendali Lampu** **42**

**Gambar 4.8 Tampilan Aplikasi di Android** **45**

**Gambar 4.9 Skema Driver Lampu AC** **47**

**Gambar 4.10 Skema Catu Daya DC 5 Volt** **48**

**xii**

**DAFTAR TABEL**

**Tabel 2.1 Perkembangan Versi Android** **14**

**Tabel 2.2 Bagian dan komponen Arduino IDE** **18**

**Tabel 4.1 Karakter huruf sebagai penanda ON/OFF lampu** **37**

**Tabel 4.2 Daftar Komponen User Interface Aplikasi Kendali Lampu** **43**

**xiii**

**DAFTAR LAMPIRAN**

**Lampiran 1. Kode Pemrograman Web Server ESP8266-12** **53**

**Lampiran 2. Kode Pemrograman Aplikasi di Android** **55**

**Lampiran 3. Kode Pemrograman Arduino Uno** **58**

**xiv**

**BAB I**

**PENDAHULUAN**

**Perkembangan perangkat seluler yang begitu cepat saat ini semakin mempermudah manusia dalam memenuhi kebutuhannya khususnya dalam bidang informasi. Kemudahan tersebut didukung oleh kemudahan akses internet yang dapat dilakukan dimanapun dan kapanpun dengan bermodal akses data internet yang disediakan oleh provider GSM maupun CDMA. Kemudahan ini selanjutnya berdampak pada aktifitas-aktifitas manusia yang kemudian dapat dilakukan dengan menggunakan perangkat seluler yang dimilikinya. Contoh yang sering dijumpai adalah aktifitas berbasis online seperti transaksi perbankan melalui internet, booking tiket pesawat online, mencari lokasi suatu tempat dengan GPS dan masih banyak contoh aktifitas-aktifitas manusia yang bisa dilakukan melalui perangkat seluler.**

**Smartphone adalah istilah yang paling tepat untuk menamakan satu perangkat bergerak dengan banyak fungsi tersebut. Saat ini penggunaan perangkat seluler tidak terbatas seperti pada contoh-contoh yang telah disebutkan di atas. Dengan hardware yang dimilikinya sebuah smartphone bisa digunakan untuk keperluan yang lain, misalnya digunakan sebagai alat kontrol perangkat lain. Sebagai contoh, fitur Bluetooth yang dimiliki oleh smartphone dapat digunakan sebagai alat komunikasi untuk mengendalikan sebuah komputer atau laptop secara jauh. Contoh lain adalah perpaduan antara teknologi Wi-Fi dan Bluetooth yang digunakan untuk keperluan pelacakan suatu objek (object tracking). Dengan kecanggihan teknologi yang dimilikinya menjadikan smartphone mempunyai**

**1**

**2**

**peranan penting dalam membangun sebuah “sistem-pintar” yang bisa diterapkan dalam kehidupan sehari-hari. Salah satu contoh pengembangan “sistem-pintar” tersebut adalah smarthome.**

**Smartphone yang tersebar luas di kalangan masyarakat saat ini mempunyai sistem operasi yang berbeda-beda, diantaranya: Blackberry, iOS, Android dan Windows Phone. Salah satu keunggulan smartphone adalah dia mampu menjalankan lebih dari satu aplikasi dalam waktu yang bersamaan atau disebut dengan multitasking. Selain itu beberapa fitur canggih telah ditanamkan dalamperangkat tersebut, sehingga sebuah smartphone bisa digunakan lebih dari sekedar untuk keperluan komunikas berbasis suara (voice call) atau berbasis teks (message), misalnya untuk panggilan dengan gambar (video call). Keunggulan smartphone tersebut kemudian dikembangkan untuk keperluan yang lain, seperti kontrol atau kendali perangkat lain menggunakan smartphone yaitu dengan membuat suatu aplikasi yang diinstal di dalamnya. Android adalah salah satu sistem operasi pada smartphone yang berbasis open source dari Google yang bisa dikembangkan secara bebas oleh para pemakainya. Sehingga dengan mudah pengguna dapat membuat suatu aplikasi sesuai dengan kebutuhannya yang dapat diinstal dan dijalankan pada perangkat seluler yang dimilikinya.**

**1.1 Latar Belakang**

**Android adalah salah satu tren teknologi seluler yang saat ini banyak diminati oleh pengguna perangkat mobile. Hal ini dapat diamati dengan banyaknya produk smartphone berbasis Android dengan harga yang murah, sehingga tidak menutup kemungkinan masyarakat menengah bawah untuk bisa**

**3**

**memilikinya dengan mudah. Seperti yang telah dijelaskan pada sub-bab sebelumnya bahwa sistem Android merupakan sistem operasi pada perangkat seluler yang berbasis open source yang bisa dikembangkan secara bebas. Tetapi hal tersebut tidak lantas menunjukkan bahwa semua pengguna perangkat ini mampu mengembangkannya, misalnya dengan membuat suatu aplikasi tertentu.**

**Dalam tulisan ini penulis membedakan dengan dua istilah kepada para pengguna smartphone, yaitu: “pengguna pasif” dan “pengguna aktif”. Pengguna pasif berarti pengguna perangkat mobile yang tidak mempunyai kemampuan untuk mengembangkan dan membuat aplikasi dalam sistem smartphone dalam hal ini sistem Android, atau dengan kata lain pengguna yang hanya bisa menjalankan fasilitas-fasilitas bawaan dari smartphone itu sendiri. Sedangkan pengguna aktif adalah pengguna yang bisa mengembangkan dan membuat aplikasi dalam sistem smartphone, atau sering disebut dengan developer. Terdapat kumpulan bermacam-macam aplikasi yang bisa diunduh oleh para pengguna smartphone, misalnya pengguna Android melalui menu Playstore. Aplikasi yang terdapat pada playstore dapat diunduh secara gratis dengan syarat harus mempunyai akun dari Google.**

**Di dalam sebuah smartphone Android telah ditanamkan beberapa fitur pendukung, seperti fitur yang digunakan untuk keperluan transfer data yaitu Wi-Fi, Bluetooth dan NFC (Near Field Communication). Namun tidak semua smartphone memiliki fitur yang sama, misalnya NFC yang belum tentu dimiliki oleh semua smartphone. Wi-Fi merupakan fitur yang dimiliki oleh setiap smartphone yang sering digunakan oleh para penggunanya. Ketika kebutuhan data**

**4**

**seluler bisa digantikan oleh tersedianya jaringan internet secara wireless (hotspot) maka fitur Wi-Fi bisa diaktifkan. Sehingga hal ini akan mengurangi biaya pemakaian data seluler.**

**Menurut pengamatan penulis, tidak semua fitur yang ditanamkan dalam smartphone digunakan oleh para penggunanya. Misalnya fitur Wi-Fi yang hanya digunakan ketika tersedia jaringan internet secara wireless atau hotspot, fitur Bluetooth yang digunakan ketika transfer data saja, bahkan saat ini pengguna lebih menyukai menggunakan aplikasi tertentu untuk keperluan transfer data misalnya menggunakan aplikasi sosial media seperti Whatsapp, BBM, Line dan sebagainya dan tentu saja memerlukan biaya karena berbasis data internet. Hal tersebut menunjukkan bahwa banyaknya kemudahan-kemudahan yang bisa diperoleh melalui penggunaan smartphone. Salah satu kemudahan yang bisa diperoleh yaitu komunikasi berbasis tanpa kabel (wireless). Komunikasi tanpa kabel (wireless) selain beberapa fitur seperti Wi-Fi, Bluetooth, Infra merah (IR) dan NFC, juga dapat dilakukan menggunakan media seluler. Diantara media komunikasi data secara wireless tersebut media seluler menempati urutan tertinggi dalam hal biaya.**

**Mobilitas manusia dalam kegiatannya menuntut setiap aktifitas dapat dilakukan dimanapun, salah satunya adalah kendali perangkat yang ada di rumah secara jarak jauh menggunakan perangkat seluler. Akan tetapi tidak serta merta hanya menggunakan perangkat seluler saja, karena dibutuhkan perangkat lain sebagai antarmuka antara perangkat seluler dengan perangkat yang akan dikendalikan. Saat ini telah beredar sebuah produk berbasis mikrokontroler yang**

**5**

**dikemas dalam sebuah kit, dikenal dengan nama Arduino. Secara umum arduino digunakan sebagai antarmuka, karena di dalamnya terdapat beberapa pin yang bisa difungsikan sebagai input dan output. Dengan memasukkan program ke dalam IC mikrokontroler di dalamnya maka arduino dapat dimanfaatkan pada banyak keperluan. Selain itu terdapat modul-modul yang bisa ditambahkan atau digabungkan dengan kit arduino tersebut. Misalnya modul bluetooth, Wi-Fi dan beberapa shield seperti Ethernet shield dan sebagainya. Jika media wireless khususnya Wi-Fi membutuhkan infrastruktur dan biaya mahal, maka dengan adanya produk arduino ini memberikan solusi dalam hal biaya.**

**Dengan alasan tersebut maka pada Tugas Akhir ini akan dibuat sebuah rancang bangun berupa alat kendali nyala lampu secara jarak jauh dengan memanfaatkan teknologi Wi-Fi yang terdapat pada smartphone berbasis pada sistem operasi Android. Kendali nyala lampu yang direncanakan merupakan gabungan antara perangkat android sebagai pusat pengendali oleh pengguna dengan mikrokontroler pada kit arduino sebagai antarmuka antara perangkat android dengan lampu sebagai objek yang dikendalikan.**

**Namun rancang bangun berupa alat kendali tersebut nantinya terbatas pada cakupan wilayah tertentu atau bersifat lokal, karena koneksi secara wireless terbatas pada jangkauan modul Wi-Fi yang digunakan dan tidak menggunakan koneksi internet. Walaupun jangkauan sinyal hanya bersifat lokal, Wi-Fi merupakan teknologi komunikasi secara wireless yang mempunyai jangkauan paling luas dibandingkan dengan teknologi lain seperti Bluetooth, IR, RFID dan NFC.**

**6**

**Peralatan kontrol atau kendali melalui Android tidak bisa dilakukan begitu saja hanya dengan membuat suatu aplikasi yang diinstal pada smartphone. Terdapat tiga bagian penting dalam pembuatan alat kendali ini yaitu (1) bagian kendali; (2) media wireless; dan (3) antarmuka sebagai driver terhadap output. Pada bagian pertama berupa sebuah aplikasi kendali yang kemudian diinstal pada Android, bagian kedua merupakan media komunikasi yang digunakan untuk mengirimkan perintah dari Android. Media yang digunakan disini adalah media Wi-Fi. Sedangkan pada bagian ketiga adalah antarmuka berupa mikrokontroler yang dikemas dalam satu kit bernama Arduino. Perintah yang dikirimkan dari Android secara wireless kemudian diterjemahkan oleh mikrokontroler sehingga menghasilkan keluaran berupa nyala dan mati beberapa lampu.**

**1.2 Perumusan Masalah**

**Berdasarkan latar belakang masalah yang telah diuraikan sebelumnya, maka dapat diambil beberapa rumusan masalah dalam pembuatan Tugas Akhir ini, diantaranya:**

1. **bagaimana cara memanfaatkan fitur Wi-Fi pada smartphone Android sebagai media kontrol terhadap perangkat lain;**
2. **bagaimana cara membangun komunikasi data antara smartphone Android dengan mikrokontroler sehingga menghasilkan kendali nyala lampu PLN;**

**1.3 Tujuan Tugas Akhir**

**Tujuan yang hendak dicapai pada pembuatan tugas akhir ini adalah:**

**7**

1. **Memanfaatkan Wi-Fi sebagai media kendali perangkat keras lain berupa nyala lampu.**
2. **Mengembangkan sistem kendali perangkat keras yang menggabungkan antara mikrokontroler (kit Arduino) dengan Android.**
3. **Menjelaskan bagaimana komunikasi secara wireless dapat digunakan untuk mengontrol perangkat keras.**

**1.4 Manfaat Tugas Akhir**

**Manfaat yang bisa diperoleh dari pembuatan alat dalam Tugas Akhir ini adalah:**

1. **Memberikan pengetahuan dasar mengenai pemrograman Android.**
2. **Prototipe yang dibuat dapat diterapkan pada keadaan yang sebenarnya oleh kampus Politama.**
3. **Sebagai referensi pada tugas akhir dengan tema yang sama untuk bisa dikembangkan lebih lanjut.**

**BAB II**

**TINJAUAN PUSTAKA DAN LANDASAN TEORI**

**2.1 Kendali Wireless**

**Dengan ditemukannya teknologi berbasis nirkabel atau tanpa kabel (wireless), maka teknologi ini kemudian menggantikan teknologi komunikasi data yang berbasis pada kabel (wired communication). Contoh mudah yang sering kita jumpai di sekitar kita adalah perangkat input pada komputer yaitu mouse dan keyboard yang menggunakan teknologi bluetooth, mesin cetak (printer) wireless, headset bluetooth dan lain-lain.**

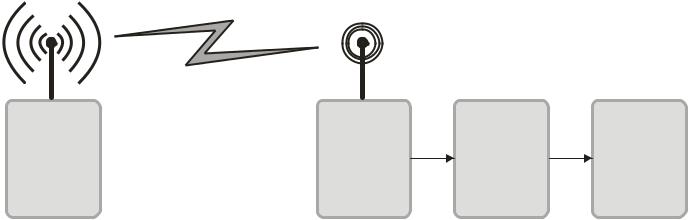
**Kendali secara wireless berarti suatu perangkat yang bisa dikendalikan secara jarak jauh menggunakan alat tertentu yang memanfaatkan media udara dalam kinerjanya. Yang dimaksud dengan alat tertentu tersebut bisa berupa remote, komputer atau perangkat seluler seperti smartphone. Penggunaan remote sebagai kendali secara jarak jauh bisa ditemukan pada perangkat lampu rumah tangga, seperti lampu LED. Remote juga sering ditemukan pada televisi. Contoh-contoh penggunaan remote tersebut menggunakan teknologi infra merah (IR, infra red) dalam komunikasinya. Sedangkan kendali menggunakan komputer akan jauh lebih kompleks karena akan melibatkan beberapa infrastruktur yang bisa memakan biaya yang tidak murah.**

**Secara umum terdapat dua bagian penting dalam sistem kendali jarak jauh secara wireless, yaitu bagian pengirim (transmitter,Tx) dan bagian penerima (receiver, Rx). Bagian pengirim (Tx) akan mengirimkan data dan selanjutnya**

**8**

**9**

**diterima oleh bagian penerima (Rx). Data yang diterima oleh Rx kemudian diolah menjadi suatu bentuk output tertentu, seperti ditunjukkan pada Gambar 2.1. Sehingga setelah terjadi komunikasi data antara Tx dan Rx akan melibatkan perangkat pendukung yang bertindak sebagai pengolah sinyal. Salah satu contoh perangkat keras pengolah sinyal adalah mikrokontroler. Diperlukannya perangkat pengolah sinyal karena sangat dimungkinkan data yang dikirimkan dari Tx kepada Rx bisa dalam bentuk teks yang perlu diubah dalam bentuk data logika 1 dan 0.**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tx** | **Rx** | **pengolah** | **Output** |  |
| **sinyal** |  |

**Gambar 2.1 Model umum komunikasi data secara wireless**

**2.2 Teknologi Wireless**

**Teknologi wireless merupakan media yang digunakan untuk melakukan pertukaran data antara dua atau lebih perangkat tanpa menggunakan media kabel. Terdapat beberapa teknologi wireless yang berkembang hingga saat ini untuk keperluan komunikasi data, diantaranya Infra merah (IR), Bluetooth (BT), Ultrasonik, RFID, Wi-Fi dan Teknologi Seluler. Diantara beberapa teknologi berbasis wireless tersebut teknologi seluler masih dikategorikan berdasarkan generasinya, yaitu AMPS (1G), GSM (2G), GPRS (2G), UMTS (3G), HSDPA (3,5G) dan LTE (4G/WiMax). Teknologi seluler merupakan teknologi komunikasi data yang bersifat luas dan membutuhkan infrastruktur yang kompleks. Sedangkan teknologi wirelesss yang lain merupakan teknologi komunikasi yang**

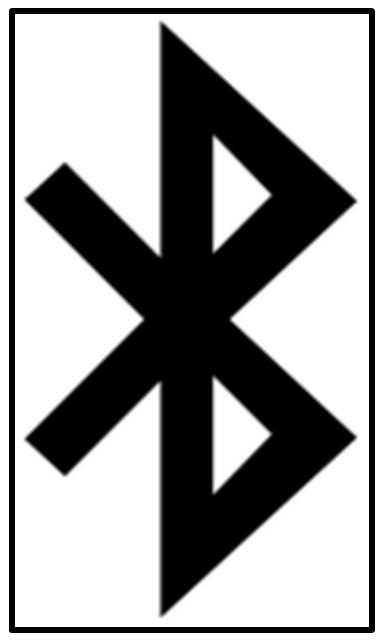
**10**

**bersifat lokal atau memiliki jangkauan wilayah (range) yang terbatas. Di bawah ini bahasan mengenai beberapa teknologi berbasis wireless.**

**2.2.1 Bluetooth**

**Bluetooth merupakan teknologi wireless yang bekerja pada frekuensi 2,4GHz dan dikategorikan dalam tiga kelas menurut jangkauan wilayahnya. Menurut S. Li, dkk (2013), tiga kelas bluetooth tersebut yaitu (1) bluetooth kelas 1 yang mempunyai jangkauan hingga 100m; (2) bluetooth kelas 2 dengan jangkauan hingga 10m; dan (3) bluetooth kelas 3 dengan jangkauan 5m. Sampai saat ini generasi bluetooth telah berkembang pada versi 4 atau dikenal dengan BLE (Bluetooth Low Energy) yang termasuk pada bluetooth kelas 3.**

**Jaringan yang dibentuk melalui teknologi bluetooth adalah PAN (personal aera network) yaitu bentuk komunikasi yang dapat dibentuk secara bebas dan dapat dibentuk kapanpun ketika dibutuhkan, yang disebut dengan adhoc. Saat dua atau lebih perangkat bluetooth akan melakukan komunikasi data maka perlu melakukan satu proses koneksitas yaitu pairing. Pada proses ini akan dibutuhkan sebuah persetujuan atau konfirmasi antara kedua atau lebih perangkat bluetooth untuk bisa melakukan koneksi. Biasanya pada proses pairing ini ditandai dengan adanya sinkronisasi melalui kode pada masing-masing perangkat.**



**Gambar 2.2 Simbol Bluetooth**

**11**

**2.2.2 Infra Merah (IR)**

**Infra merah (IR) merupakan salah satu teknologi komunikasi data berbasis wireless yang mempunyai spektrum gelombang microwave dengan radiasi cahaya yang tidak bisa dilihat oleh mata telanjang. Infra merah bekerja pada frekuensi 300GHz – 400THz dengan panjang gelombang 750nm – 1mm. Dengan panjang gelombang dalam ukuran nanometer tersebut maka IR mempunyai jangkauan yang pendek. Dalam kinerjanya antara pemancar dan penerima tidak boleh adanya penghalang, atau disebut dengan line of sight (LOS). Jangkauan yang pendek dan syarat LOS merupakan kelemahan dari teknologi IR ini. Akan tetapi IR juga mempunyai keunggulan seperti: pemeliharaan yang mudah, tidak mudah terpengaruh (interferensi) oleh adanya gelombang lain karena IR bekerja pada frekuensi tinggi. Walaupun IR mempunyai beberapa kelemahan, IR tetap dibutuhkan dan diguakan pada perangkat-perangkat tertentu misalnya: remot kontrol, optical disk, mouse komputer dan lain-lain.**

**2.2.3 Radio Frequency Identification (RFID)**

**Radio Frequency Identification (RFID) merupakan bagian dari teknologi komunikasi dan informasi yang mempunyai kemampuan untuk mendeteksi keberadaan suatu objek yang secara otomatis terintegrasi dengan sebuah sistem sehingga tidak ada interferensi secara manual dari pihak lain (George Roussos-2008). Dalam komunikasinya terdapat dua bagian yaitu tag dan reader. Tag berfungsi sebagai pengirim data dan reader sebagai penerima data yang terdapat pada sistem komputer. Bentuk komunikasi antara kedua bagian ini dilakukan secara wireless.**

**12**

**Terdapat tiga jenis dalam RFID yaitu passive, active dan semi-passive. Jenis RFID passive dalam kinerjanya tidak membutuhkan sumberdaya dari baterai, sinyal atau gelombang yang dipancarkan oleh reader akan menjadi sumberdaya oleh tag. Namun RFID jenis ini mempunyai jangkauan yang pendek tetapi memiliki bentuk fisik yang sangat kecil. Jenis yang kedua yaitu RFID active, dalam kinerjanya sebuah tag disertai dengan sumberdaya berupa baterai. Jenis RFID ini mempunyai kelebihan dalam hal jangkauan akan tetapi bentuk fisiknya lebih besar dibanding jenis pertama. Sedangkan jenis ketiga RFID semi-passisve hampir mirip dengan jenis active hanya saja selama reader belum memancarkan sinyalnya tag tidak akan mentransmisikan data kepada reader. Sehingga baterai yang berada dalam tag hanya untuk memberikan suplai pada chip saja.**



**Gambar 2.3 Tag dan Reader RFID**

**2.2.4 Wi-Fi (Wireless Fidelity)**

**Wi-Fi adalah media transfer data tanpa kabel yang bekerja pada frekuensi 2,4GHz dan 5GHz dengan kecepatan transfer hingga 54 Mbps. Wi-Fi termasuk dalam keluarga IEEE 802.11 dengan variannya yaitu 802.11a, 802.11b dan 802.11g (Jin-Shyan Lee, dkk - 2007).**

**13**

**Jangkauan Wi-Fi yang bersifat lokal juga dikenal dengan istilah WLAN (Wireless Local Area Network).**

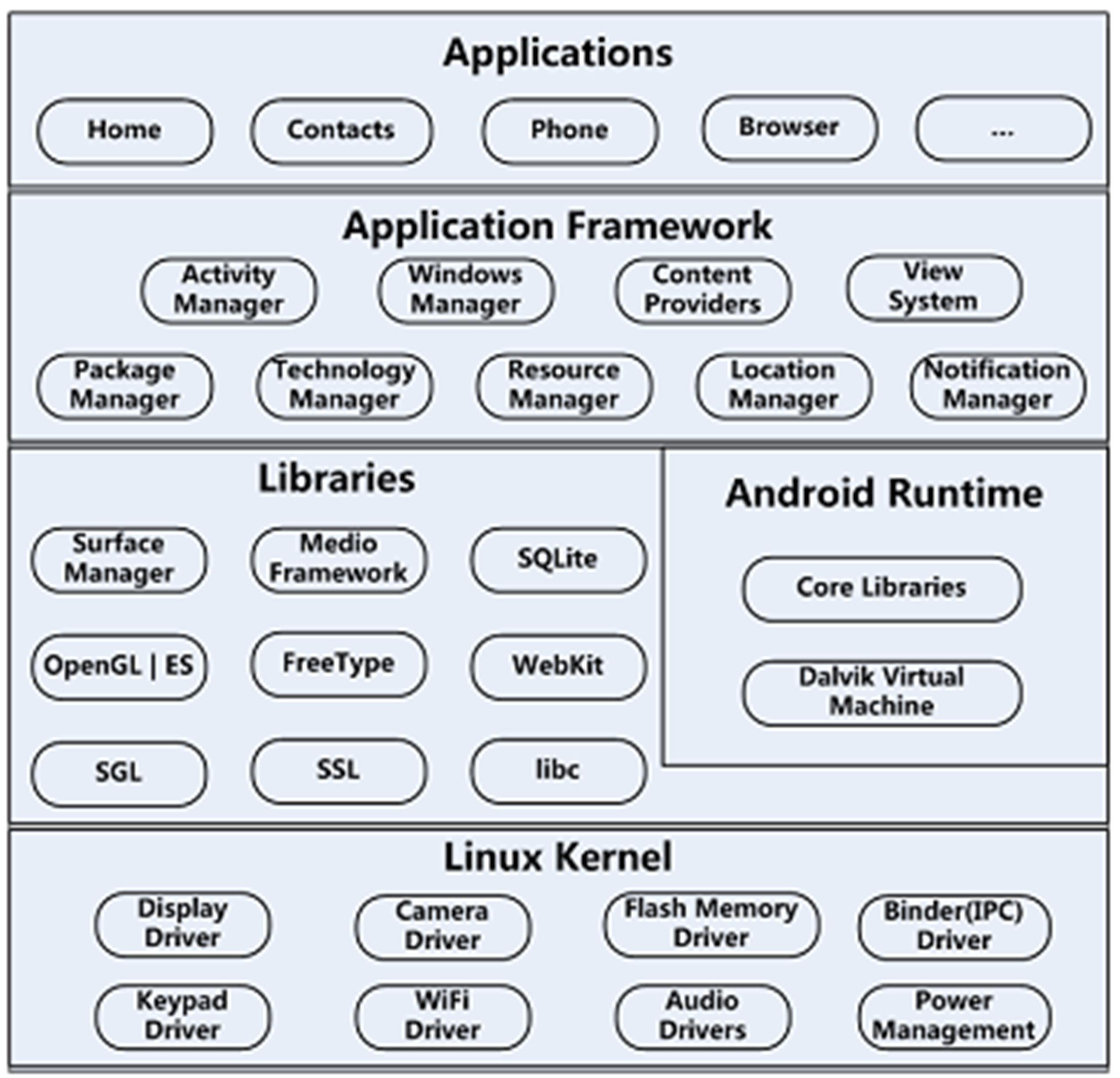
**Pada frekuensi kerja tertinggi yaitu 5GHz Wi-Fi dapat memberikan jangkauan sinyalnya hingga 100 meter pada wilayah terbuka. Sedangkan pada wilayah tertutup jangkauan Wi-Fi akan berkurang karena pengaruh propagasi atau pantulan dari benda-benda yang menjadi penghalangnya. Namun dibanding dengan teknologi wireless lainnya, Wi-Fi masih menempati urutan tertinggi dalam hal kinerjanya.**

**2.3 Android**

**Android merupakan platform perangkat lunak dan Sistem Operasi yang dikembangkan oleh Google beserta Open Handset Alliance yang diperuntukkan pada perangkat mobile. Platform perangkat lunak yang digunakan pada perangkat Android tersebut berbasis pada kernel Linux 2.6 (G. Chang, dkk - 2010), dimana kernel Linux tersebut juga merupakan sebuah lapisan (layer) dalam sebuah arsitektur sistem operasi Android yang ditunjukkan pada Gambar 2.4.**

**Sistem yang berbasis open source ini kemudian membuka peluang kepada para penggunanya untuk bisa membuat sendiri aplikasi-aplikasi yang bisa diinstal pada ponselnya. Aplikasi dengan ekstensi yang bernama apk ini dapat dibuat dari berbagai macam platform, seperti Linux, Windows dan MacOS. Selain itu pihak Google telah menyiapkan store yang berisi produk-produk aplikasi Android yang bisa diunduh secara gratis maupun yang berbayar, yaitu melalui Playstore.**

**14**

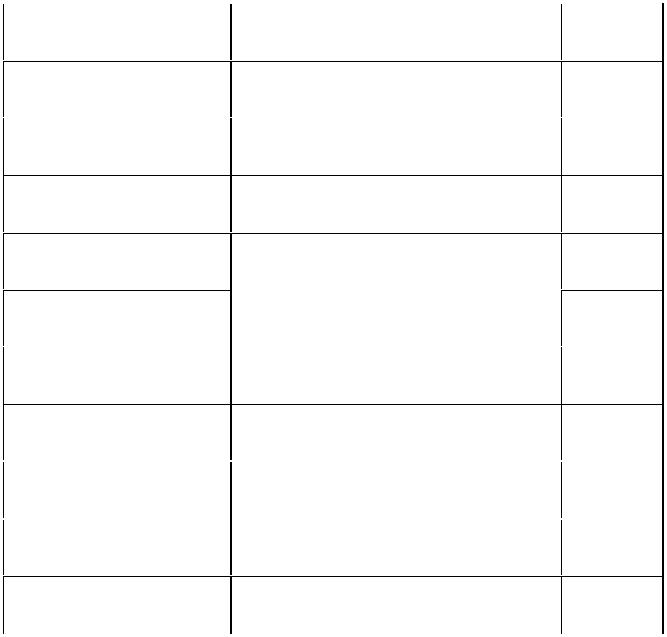


**Gambar 2.4 Arsitektur Sistem Operasi Android**

**Dalam referensi, hingga saat ini perkembangan versi Android telah mencai versi 6 pada level API 23 yang ditunjukkan pada Tabel 2.1.**

**Tabel 2.1 Perkembangan Versi Android**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Versi** | **Nama** | **API** |  |
| **2.2** | **Froyo** | **8** |  |
| **2.3.3 - 2.3.7** | **Gingerbread** | **10** |  |
| **4.0.3 - 4.0.4** | **Ice Cream Sandwich** | **15** |  |
| **4.1.x** |  | **16** |  |
| **4.2.x** | **Jelly Bean** | **17** |  |
| **4.3** |  | **18** |  |
| **4.4** | **KitKat** | **19** |  |
| **5.0** | **Lollipop** | **21** |  |
| **5.1** | **22** |  |
|  |  |
| **6.0** | **Marshmallow** | **23** |  |

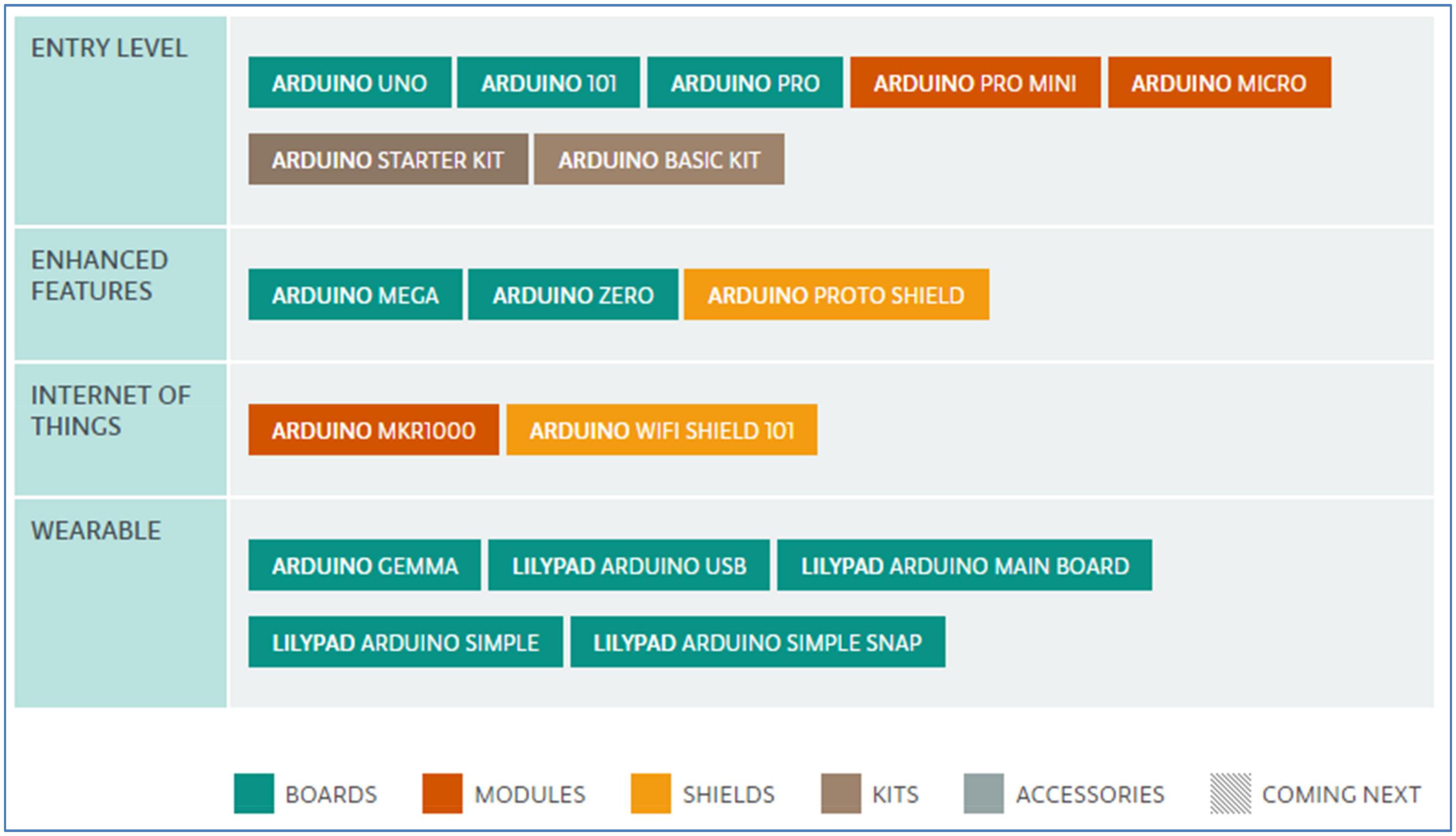


**15**

**2.4 Arduino**

**Arduino adalah sebuah prototipe yang berbasis open-source yang tujuannya untuk mempermudah penggunaan software dan mempermudah dalam mengendalikan hardware. Terdapat dua bagian penting dalamsebuah produk Arduino, yaitu board Arduino dan Arduino Software (dikenal dengan nama Arduino IDE). Dalam sebuah board Arduino, tersusun atas satu chip utama yaitu sebuah mikrokontroler dengan beberapa komponen pendukung lainnya.**

**Produk Arduino terbagi pada 4 macam tingkatan yaitu: Entry Level, Enhanced Features, Internet of Things dan Wearable. Masing-masing tingkatan produk mempunyai versinya masing-masing seperti ditunjukkan pada Gambar 2.5.**



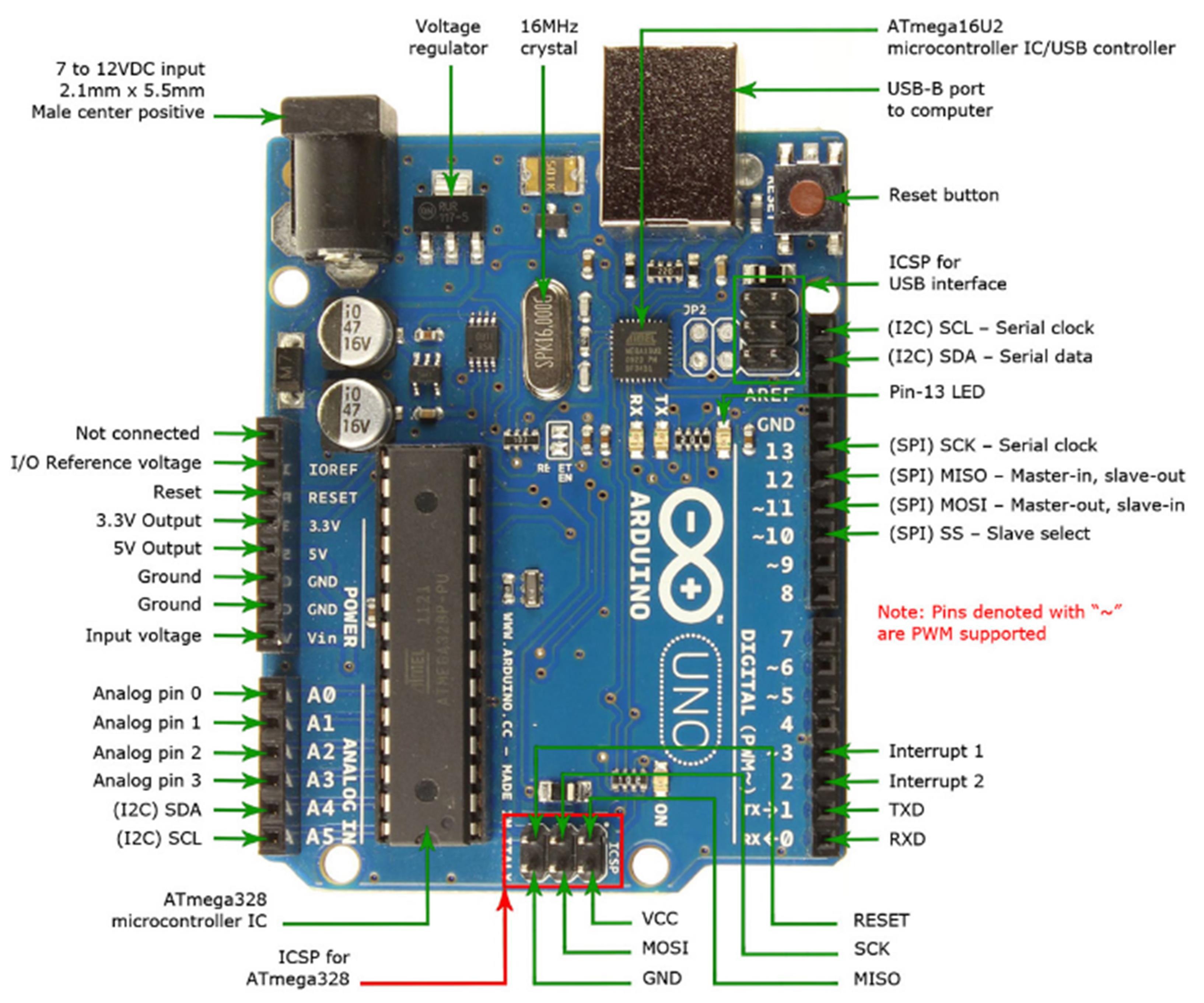
**Gambar 2.5 Klasifikasi produk Arduino di negara USA**

**Sifat dari pengembang Arduino yang open-source membuka peluang kepada pihak-pihak lain untuk menciptakan modul atau produk yang bisa diintegrasikan dengan Arduino, seperti contohnya modul Wi-Fi, Bluetooth, GSM dan lain-lain.**

**16**

**2.4.1 Board Arduino Uno**

**Kit atau board Arduino Uno merupakan salah satu jenis board Arduino yang berbasis pada mikrokontroler ATmega 328. Mikrokontroler jenis ini banyak digunakan karena mudah dalam pemrogramannya, low-powered dan murah harganya. Selain Arduino Uno, Atmega 328 juga dapat ditemukan pada board Arduino Nano. Bagian-bagian dari board Arduino Uno ditunjukkan pada Gambar 2.6.**



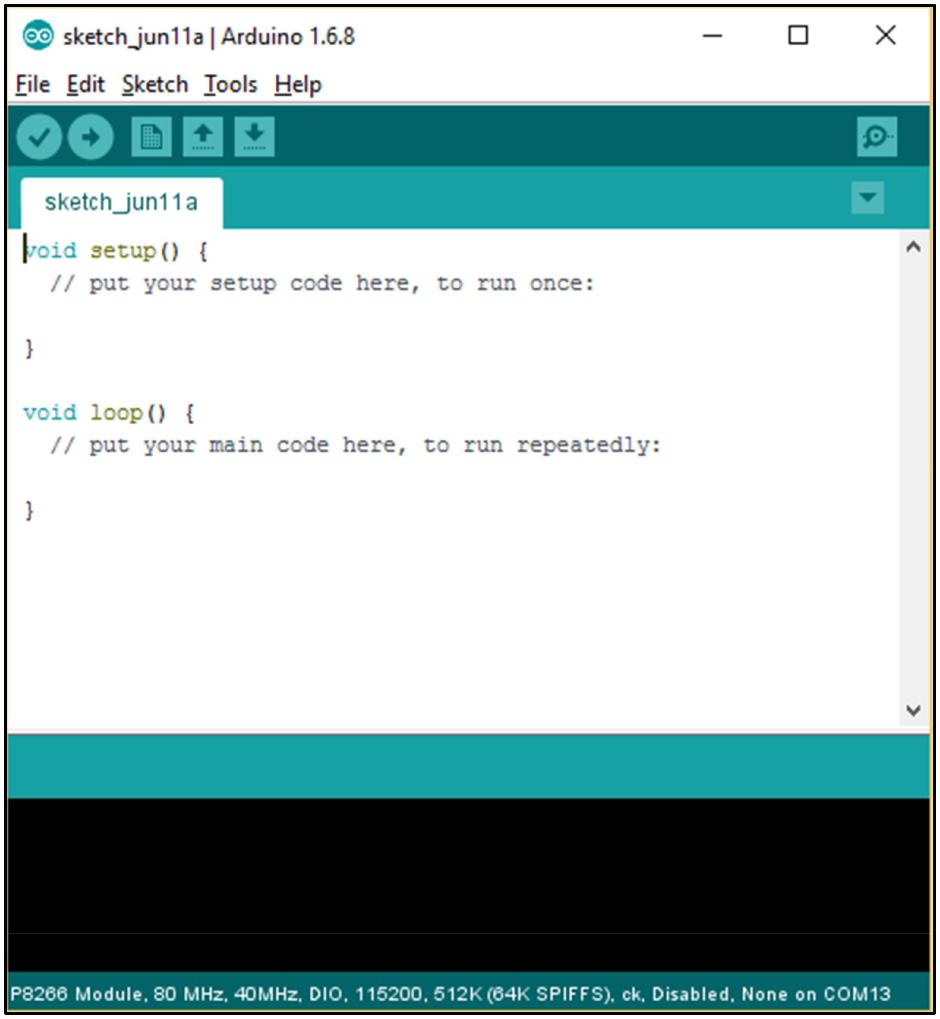
**Gambar 2.6 Bagian-bagian Arduino Uno**

**Arduino Uno bekerja pada tegangan DC 5 volt, sedangkan tegangan yang boleh digunakan untuk mensuplainya adalah 7-12 volt pada arus 40mA.**

**17**

**2.4.2 IDE Arduino**

**Arduino IDE (Integrated Development Environment) adalah software yang digunakan untuk meng-upload kode program ke dalam board. Di dalamnya berupa tampilan teks editor (Gambar 2.7) untuk menuliskan program dalam format bahasa C. Program yang telah dituliskan pada Arduino dinamakan dengan Sketches dengan nama ekstensi ino. Di dalam software ini telah disediakan beberapa contoh sketches yang bisa dimodifikasi sesuai dengan kebutuhan. Contoh tersebut dapat dilihat pada menu File – Examples.**

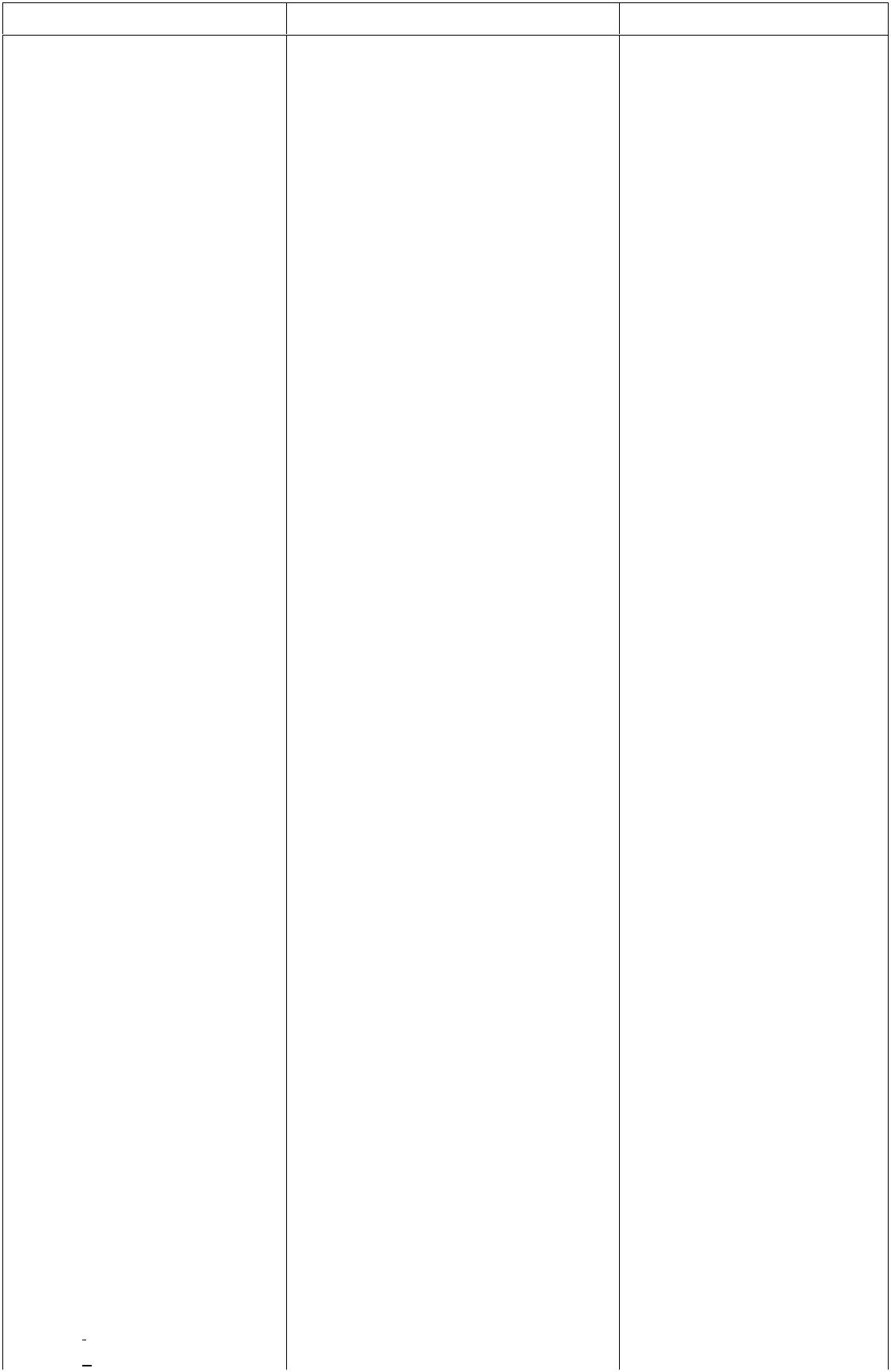


**Gambar 2.7 Tampilan Arduino IDE**

**Dalam penulisan program terbagi menjadi 3, yaitu: structure, values (variables dan constants) dan functions. Komponen dari masing-masing bagian tersebut ditunjukkan pada Tabel 2.2.**

**18**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Tabel 2.2 Bagian dan komponen Arduino IDE** | | | | | |  |  |  |
|  |  | **Structure** |  |  |  | **Values** |  |  |  | **Function** |  |
| **1.** | **Setup ()** | |  | **1.** | **Constants** | | **1.** | | **Digital I/O** | |  |
| **2.** | **Void ()** | |  |  |  | **HIGH | LOW** |  |  |  | **pinMode()** |  |
| **3.** | **Control Structures** | | |  |  | **INPUT | OUTPUT | INPU** |  |  |  | **digitalWrite()** |  |
|  |  | **if** |  |  |  | **T\_PULLUP** | **2.** | |  | **digitalRead()** |  |
|  |  | **if...else** |  |  | **LED\_BUILTIN** | **Analog I/O** | |  |
|  |  | **for** |  |  |  | **true | false** |  |  |  | **analogReference()** |  |
|  |  | **switch case** |  |  |  | **integer constants** |  |  |  | **analogRead()** |  |
|  |  | **while** |  | **2.** |  | **floating point constants** |  |  |  | **analogWrite() -** |  |
|  |  | **do... while** |  | **Data Types** | | **3.** | |  | **PWM** |  |
|  |  | **break** |  |  |  | **void** | **Due & Zero only** | |  |
|  |  | **continue** |  |  |  | **boolean** |  |  |  | **analogReadResolut** |  |
|  |  | **return** |  |  |  | **char** |  |  |  | **ion()** |  |
| **4.** |  | **goto** |  |  |  | **unsigned char** |  |  | **analogWriteResolu** |  |
| **Further Syntax** | |  |  |  | **byte** | **4.** | |  | **tion()** |  |
|  |  | **; (semicolon)** |  |  |  | **int** | **Advanced I/O** | |  |
|  |  | **{} (curly braces)** | |  |  | **unsigned int** |  |  |  | **tone()** |  |
|  |  | **// (single line** |  |  |  | **word** |  |  |  | **noTone()** |  |
|  |  | **comment)** |  |  |  | **long** |  |  |  | **shiftOut()** |  |
|  | **/\* \*/ (multi-line** | |  |  | **unsigned long** |  |  |  | **shiftIn()** |  |
|  |  | **comment)** |  |  |  | **short** | **5.** | |  | **pulseIn()** |  |
|  | **#define** |  |  |  | **float** | **Time** | |  |
| **5.** |  | **#include** |  |  |  | **double** |  |  |  | **millis()** |  |
| **Arithmetic Operators** | | |  |  | **string - char array** |  |  |  | **micros()** |  |
|  |  | **= (assignment** |  |  |  | **String - object** |  |  |  | **delay()** |  |
|  |  | **operator)** |  | **3.** |  | **array** |  |  |  | **delayMicroseconds** |  |
|  | **+ (addition)** |  | **Conversion** | | **6.** | |  | **()** |  |
|  |  | **- (subtraction)** |  |  |  | **char()** | **Math** | |  |
|  |  | **\* (multiplication)** | |  |  | **byte()** |  |  |  | **min()** |  |
|  |  | **/ (division)** |  |  |  | **int()** |  |  |  | **max()** |  |
| **6.** |  | **% (modulo)** |  |  |  | **word()** |  |  |  | **abs()** |  |
| **Comparison Operators** | | |  |  | **long()** |  |  |  | **constrain()** |  |
|  |  | **== (equal to)** |  | **4.** |  | **float()** |  |  |  | **map()** |  |
|  |  | **!= (not equal to)** | | **Variable Scope & Qualifiers** | |  |  |  | **pow()** |  |
|  |  | **< (less than)** |  |  |  | **variable scope** | **7.** | |  | **sqrt()** |  |
|  |  | **> (greater than)** | |  |  | **static** | **Trigonometry** | |  |
|  |  | **<= (less than or** | |  |  | **volatile** |  |  |  | **sin()** |  |
|  |  | **equal to)** |  | **5.** |  | **const** |  |  |  | **cos()** |  |
|  | **>= (greater than or** | | **Utilities** | | **8.** | |  | **tan()** |  |
| **7.** |  | **equal to)** |  |  |  | **sizeof()** | **Characters** | |  |
| **Boolean Operators** | | |  |  | **PROGMEM** |  |  |  | **isAlphaNumeric()** |  |
|  |  | **&& (and)** |  |  |  |  |  |  |  | **isAlpha()** |  |
|  |  | **|| (or)** |  |  |  |  |  |  |  | **isAscii()** |  |
| **8.** |  | **! (not)** | **Access** |  |  |  |  |  |  | **isWhitespace()** |  |
| **Pointer** | |  |  |  |  |  |  | **isControl()** |  |
|  | **Operators** | |  |  |  |  |  |  |  | **isDigit()** |  |
|  |  | **\* dereference** |  |  |  |  |  |  |  | **isGraph()** |  |
|  |  | **operator** |  |  |  |  |  |  |  | **isLowerCase()** |  |
| **9.** | **& reference operator** | |  |  |  |  |  |  | **isPrintable()** |  |
| **Bitwise Operators** | | |  |  |  |  |  |  | **isPunct()** |  |
|  |  | **& (bitwise and)** | |  |  |  |  |  |  | **isSpace()** |  |
|  |  | **| (bitwise or)** |  |  |  |  |  |  |  | **isUpperCase()** |  |
|  |  | **^ (bitwise xor)** | |  |  |  |  |  |  | **isHexadecimalDigi** |  |



**19**

**~ (bitwise not) ~~<~~ < (bitshift left) ~~>>~~ (bitshift right)**

1. **Compound Operators ++ (increment)**

**-- (decrement) += (compound addition)**

**-= (compound subtraction)**

**\*= (compound multiplication)**

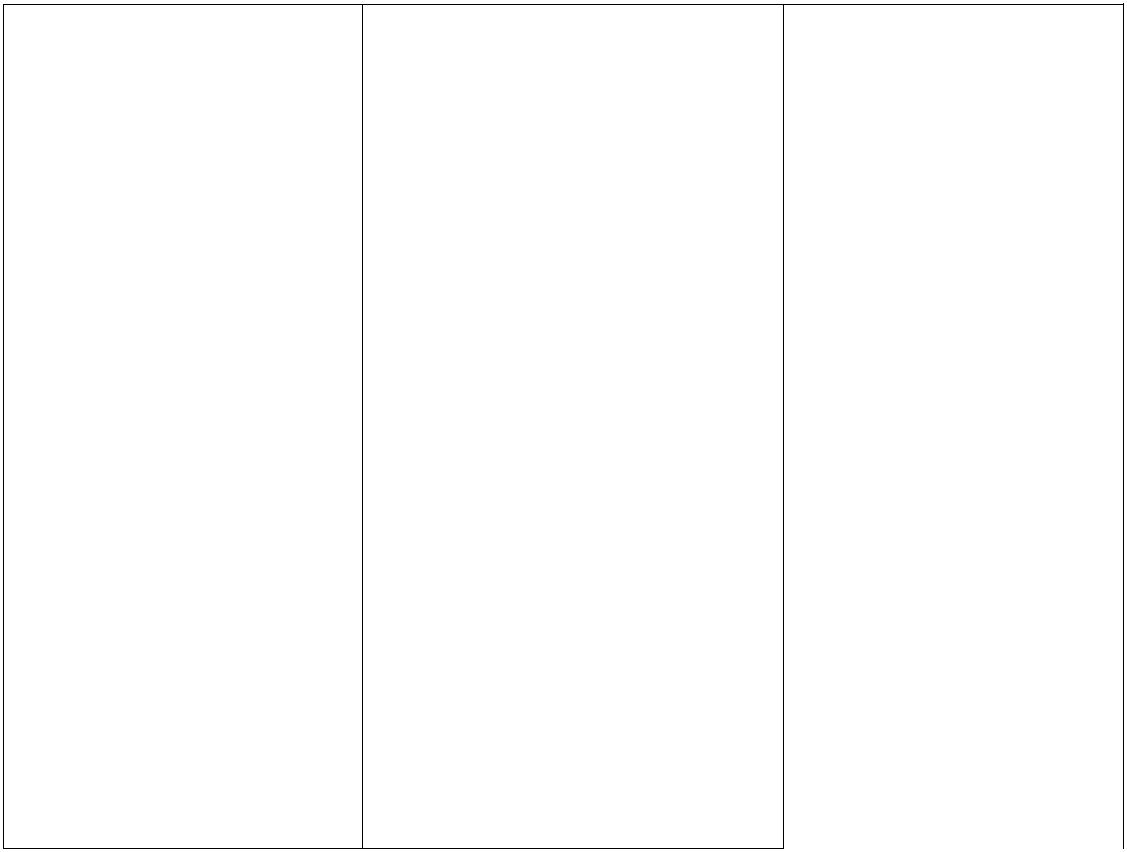
**/= (compound division)**

**%= (compound modulo)**

**&= (compound bitwise and)**

**|= (compound bitwise or)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **9.** |  | **t()** |  |  |
| **Random Numbers** | | |  |
|  |  | **randomSeed()** | |  |
| **10.** |  | **random()** | |  |
| **Bits and Bytes** | | |  |
|  |  | **lowByte()** | |  |
|  |  | **highByte()** | |  |
|  |  | **bitRead()** | |  |
|  |  | **bitWrite()** | |  |
|  |  | **bitSet()** | |  |
|  |  | **bitClear()** | |  |
| **11.** |  | **bit()** |  |  |
| **External Interrupts** | | |  |
|  |  | **attachInterrupt()** | |  |
| **12.** |  | **detachInterrupt()** | |  |
| **Interrupts** | | |  |
|  |  | **interrupts()** | |  |
| **13.** |  | **noInterrupts()** | |  |
| **Communication** | | |  |
|  |  | **Serial** | |  |
| **14.** |  | **Stream** | |  |
| **USB** | | **(32u4 based** |  |
|  | **boards** | | **and Due/Zero** |  |
|  | **only)** | |  |  |
|  |  | **Keyboard** | |  |
|  |  | **Mouse** | |  |



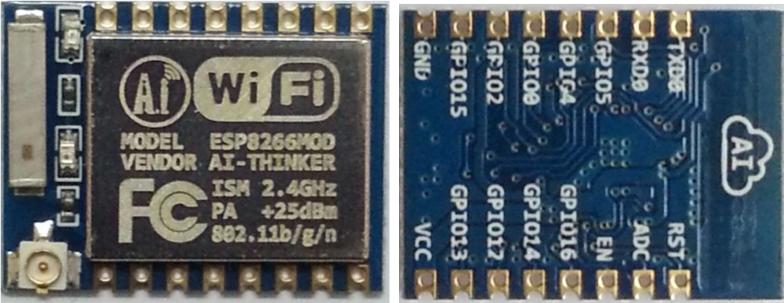
**2.5 Modul ESP8266-12**

**ESP8266-12 adalah modul Wi-Fi di luar Arduino yang bisa diintegrasikan ke dalam pemrograman Arduino. Ketika modul disambungkan dengan board Arduino maka sinyal Wi-Fi akan dapat dihasilkan dengan alamat IP “192.168.4.1”. Dalam kinerjanya modul ESP membutuhkan tegangan maksimal 3,3 volt yang bisa disuplai melalui board Arduino Uno.**

**Banyak tipe dari modul ESP ini, biasanya tipe atau versi dari modul ini ditandai dengan angka atau huruf di belakang nama ESP822, misalnya ESP8266-**

1. **Namun penulisan ini tidak akan ditemui pada chip yang terpasang pada modul. Tipe atau versi ESP dilihat dari bentuk fisik dan jumlah kaki/pin yang dipunyai serta melihat datasheetnya.**

**20**



**Gambar 2.8 Modul ESP8266 tipe 12**

**Untuk meng-upload kode program (sketches) ke dalam modul ESP dibutuhkan USB-TTL konverter (Gambar 2.9). Sedangkan software yang digunakan adalah dengan memanfaatkan Arduino Ide. Melalui At command modul ini dapat diaktifkan dalam 3 mode, yaitu: Station, Access Poin (AP) dan gabungan dari keduanya.**



**Gambar 2.9 USB to TTL**

**2.6 Program Basic for Android (B4A)**

**Terdapat banyak program yang dapat digunakan untuk membuat aplikasi (apk) pada sistem android. Beberapa program tersebut diantaranya: B4A, App Inventor, Visual Basic, Eclipse dan lain-lain. Dasar pemrograman dari program-program tersebut menggunakan bahasa pemrograman Java. Dalam tulisan ini akan dijelaskan mengenai aplikasi Basic for Android (B4A).**

**B4A adalah aplikasi yang digunakan untuk membangun suatu aplikasi pada smartphone android (ber-ekstensi apk) yang bisa dijalankan pada PC berbasis Windows. Sedangkan aplikasi hasil olah dari B4A yang sudah berekstensi apk**

**21**

**tidak dapat dijalankan pada Windows. Pada B4A disediakan tool yang digunakan untuk membuat emulator (virtual device) android yang fungsinya untuk menjalankan file apk sebelum dijalankan pada perangkat android yang sesungguhnya.**

**2.6.1 Sistem Requirement B4A**

**Aplikasi B4A dapat diperoleh melalui website: www.basic4android.com yang bisa diunduh secara gratis. Namun versi yang diberikan adalah versi trial atau ujicoba selama beberapa waktu, jika ingin memperoleh versi yang penuh harus melakukan pembayaran lisensi. Akan tetapi aplikasi dengan versi penuh (full version) bisa diperoleh dengan mencari di mesin pencari seperti Google, yang sengaja disediakan oleh pihak atau personal tertentu. Aplikasi yang digunakan dalam Tugas Akhir ini menggunakan versi 5.20.**

**Dalam instalasinya diperlukan beberapa persyaratan atau beberapa komponen yang perlu diperhatikan, diantaranya:**

1. **Sistem Komputer**

**Sangat disarankan komputer (PC, Laptop) yang digunakan mempunyai RAM minimal 2GB, kemampuan grafis di atas standar dan space harddisk yang banyak. Juga perlu diperhatikan versi sistem yang digunakan, misalnya Windows 32 bit (x86) atau 64 bit (x64).**

1. **Java JDK**

**JDK atau Java Development Kit adalah sebuah lingkungan pengembangan suatu perangkat lunak yang digunakan untuk membangun aplikasi java dan applet (program inti). Di dalam JDK ini terdapat dua komponen yaitu JVM**

**22**

**(Java Virtual Machine) dan JRE (Java Runtime Environment). Master instalasi JDK dapat diperoleh melalui alamat www.oracle.com, dan hal yang perlu dicermati adalah dimanakah JDK akan diinstal. Yaitu disesuaikan dengan versi windows nya, apakah 32 bit atau 64 bit.**

1. **SDK Android**

**SDK atau Software Development Kit adalah perangkat yang digunakan untuk mengembangkan aplikasi pada platform android. Di dalam SDK terdapat beberapa tool pengembangan seperti emulator, cource code, dokumentasi, librari, debugger dan tutorial. SDK android dapat diperoleh melalui alamat http://developer.android.com/tools/help/sdk-manager.html.**

1. **Perangkat Android (optional)**

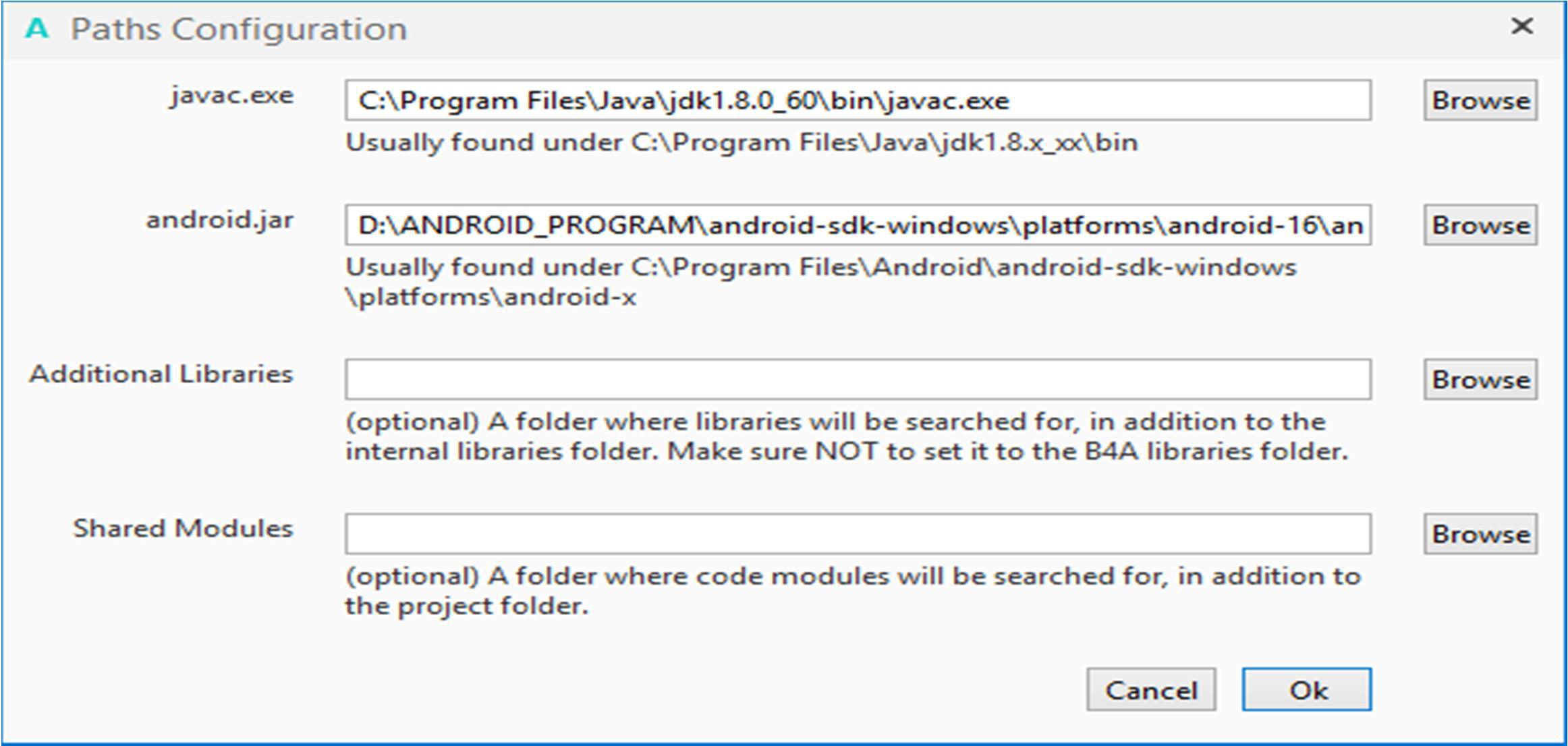
**Dalam eksekusinya B4A membutuhkan tempat untuk menjalankan file apk. Walaupun telah disediakan tool untuk membuat emulator melalui AVD, terkadang pengembang juga menggunakan pilihan untuk dijalankan langsung pada perangkat android. Hal ini dikarenakan PC atau komputer yang digunakan tidak cukup cepat untuk menjalankan emulator karena keterbatasan spesifikasi dari PC atau komputer.**

**2.6.2 Instalasi B4A**

**Sebelum instalasi dimulai, terlebih dahulu untuk memperbarui perangkat lunak Net Framwork atau dot.net walaupun sudah terintegrasi pada sistem windows, terkadang B4A akan meminta versi dari dot.netnya. Instalasi dimulai dari JDK dilanjutkan SDK (dilanjutkan dengan download komponen API, application programming interface sesuai dengan versi android yang dibutuhkan).**

**23**

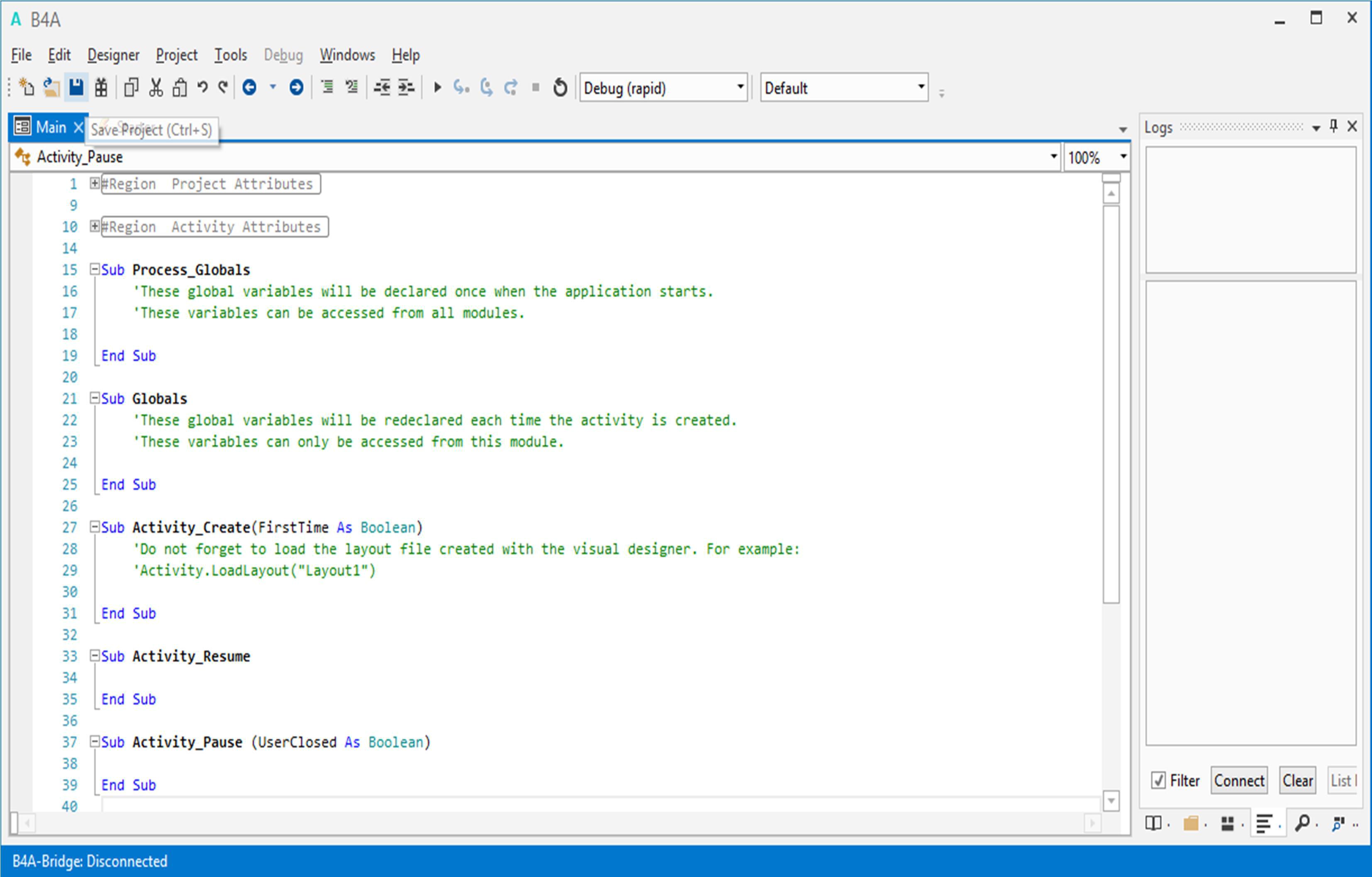
**Setelah itu dilanjutkan dengan instalasi B4A dan konfigurasi path Javac.exe dan android.Jar dari instalasi JDK melalui menu Tool-configure path.**



**Gambar 2.10 Konfigurasi Path**

**2.6.3 Pemrograman dengan B4A**

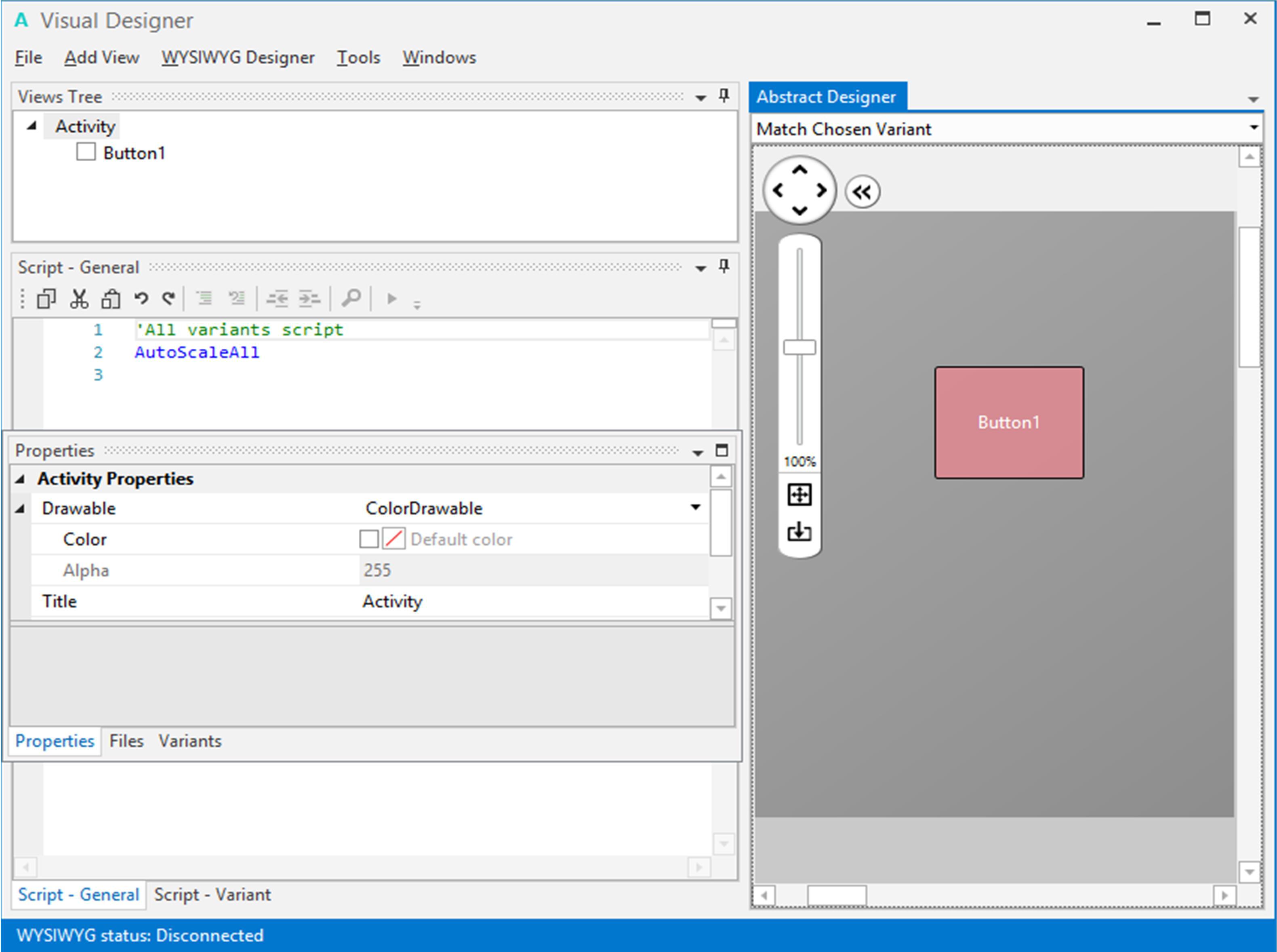
**Untuk memulai pemrograman terlebih dahulu membuat folder tempat menyimpan projek yang akan dikerjakan. Setelah membuka layar kerja B4A melalui menu File, projek disimpan dengan nama tertentu.**



**Gambar 2.11 Tampilan awal lembar kerja B4A**

**24**

**Untuk mendesain tampilan atau rancangan user interface dilakukan melalui menu Designer – Open designer seperti yang ditunjukkan pada Gambar 2.12. seperti pada langkah sebelumnya harus terlebih dahulu disimpan dengan nama tertentu (misal: utama). Pada tampilan tersebut kemudian bisa ditempatkan objek-objek yang menjadi kebutuhan aplikasi yang akan dibuat, seperti Button, Label dan lain sebagainya. Untuk menempatkan objek-objek yang akan menjadi fungsi dipilih menu Add View dengan cara pilih (klik mouse: contoh pada Gambar 2.12 adalah tombol Button), sehingga pada layar sebelah kanan akan muncul objek yang telah dipilih.**

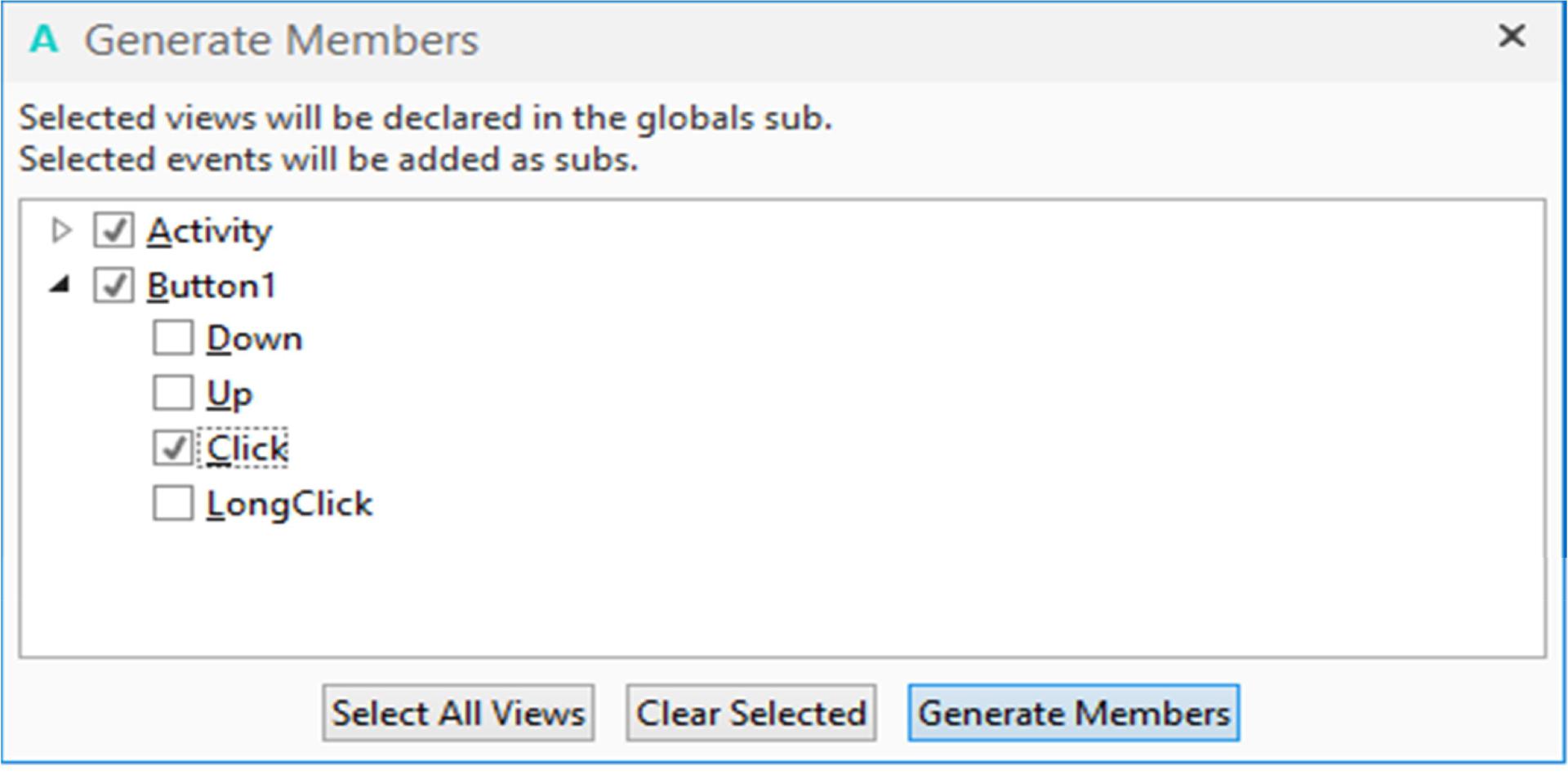


**Gambar 2.12 Tampilan Designer**

**Untuk mengatur properti dari objek (misal tombol Button) disediakan Properties seperti size untuk mengatur ukuran tombol, color, name, text dan lain-lain. Sebelum memulai menuliskan source code dari objek yang telah dipilih,**

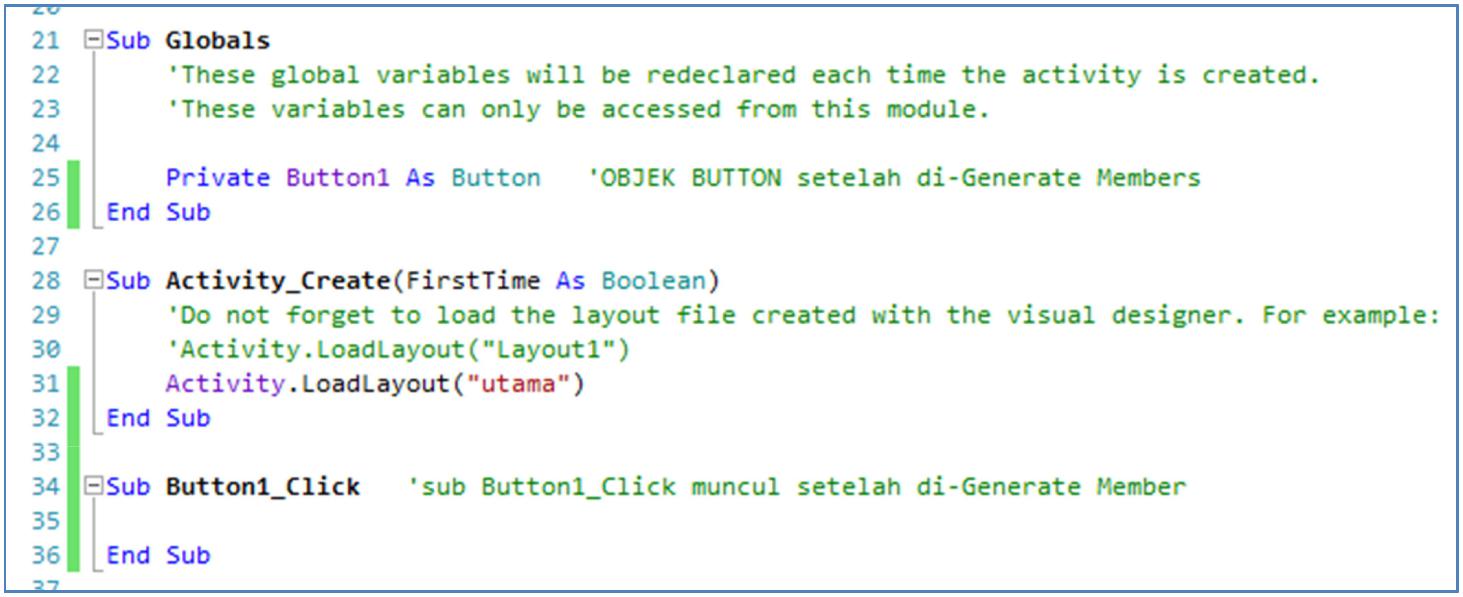
**25**

**maka terlebih dahulu objek-objek pada designer di-generate melalui menu Tool-Genereate Members, ditunjukkan pada Gambar 2.13.**



**Gambar 2.13 Tampilan Generate Members**

**Setelah proses ini selesai barulah bisa menuliskan source code pada tampilan awal Gambar 2.11. Sangat perlu diperhatikan dan diingat untuk memanggil nama file pada designer pada Sub Activities\_Create. Pada contoh ini nama yang diberikan adalah “utama” maka kode yang dituliskan ditunjukkan pada Gambar 2.14 berikut ini:**

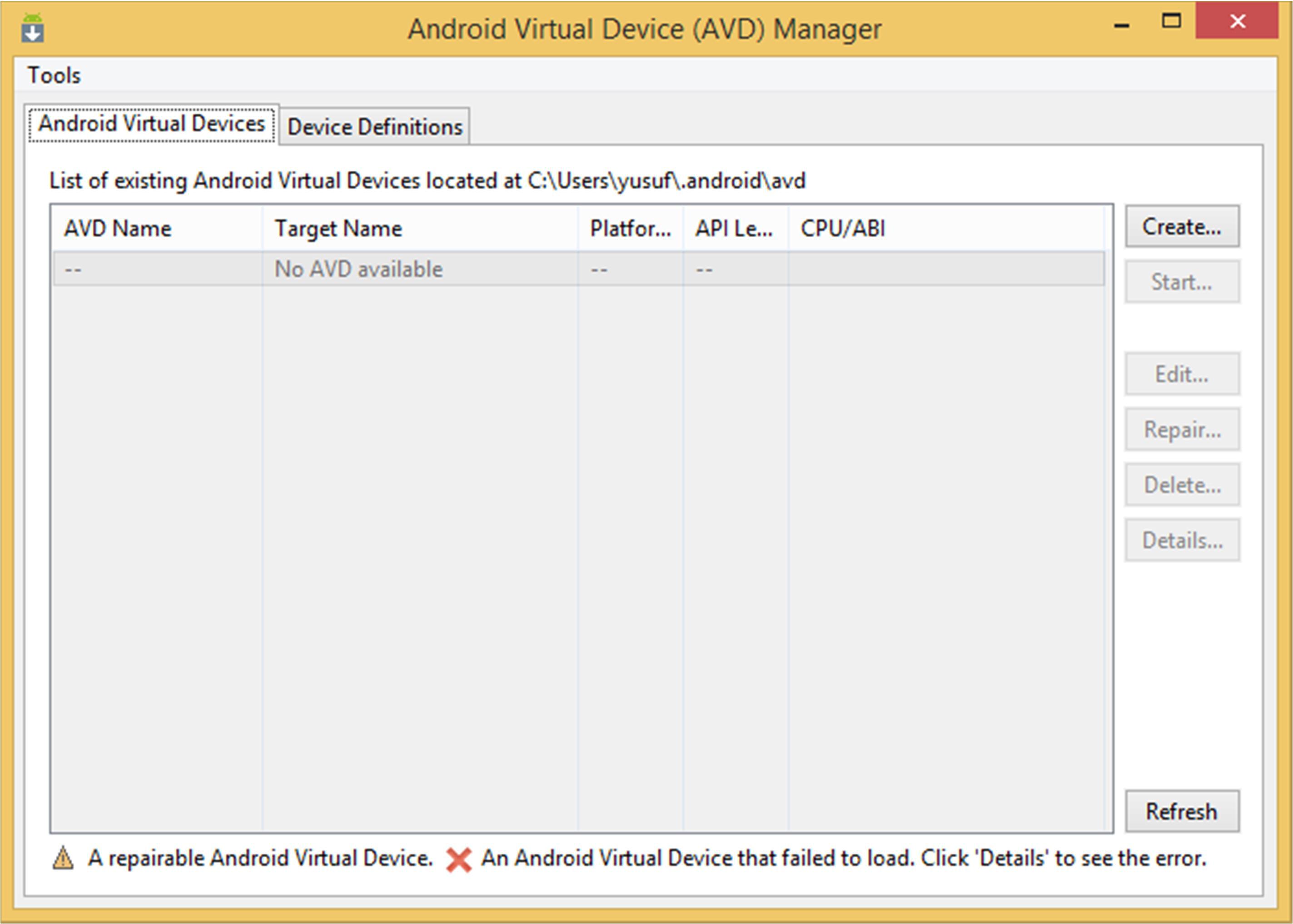


**Gambar 2.14 Tampilan tempat source code dituliskan**

**26**

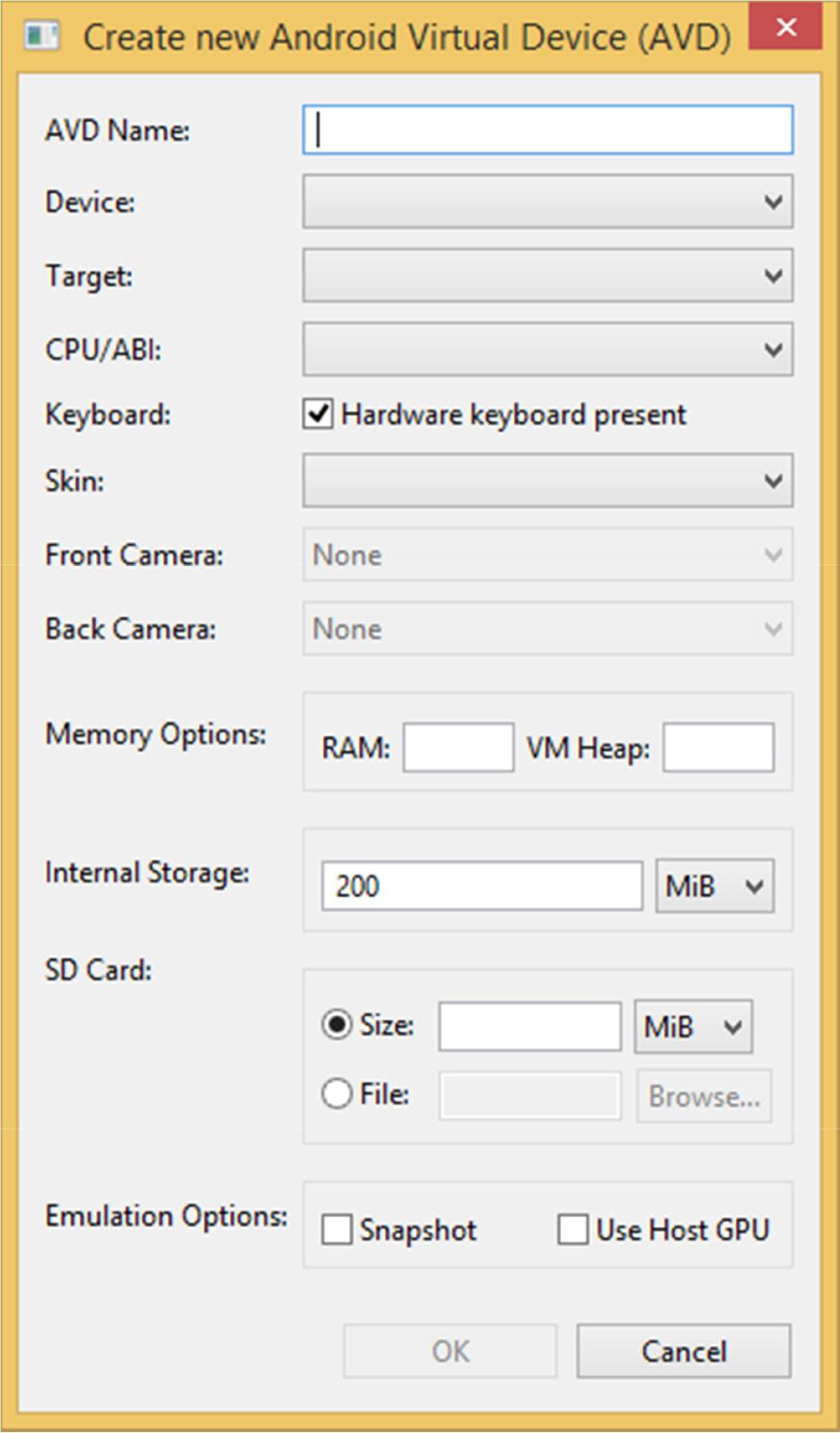
**2.6.4 Perangkat Virtual Android (emulator)**

**Perangkat virtual android atau emulator android adalah tempat dimana aplikasi melalui B4A dapat dijalankan. Perangkat virtual ini berupa perangkat Android yang dijalankan di atas sistem operasi komputer. Emulator ini dapat dibuat melalui menu Tool – Run AVD Manager, seperti ditunjukkan pada Gambar 2.15. Untuk menambahkan atau membuat emulator dipilih tab create, seperti ditunjukkan pada Gambar 2.16. Pembuatan emulator akan mengurangi ruang RAM pada komputer yang digunakan, sehingga komputer dengan spesifikasi sedang akan mengalami load yang lama.**



**Gambar 2.15 Tampilan AVD Manager**

**27**



**Gambar 2.16 Tampilan pembuatan Emulator**

**2.6.5 B4A Bridge**

**B4A Bridge digunakan untuk membuat koneksi dengan perangkat emulator atau dengan perangkat android. Koneksi dengan emulator dilakukan pada tampilan View Designer, melalui menu WYSIWYG Designer. Sedangkan untuk koneksi dengan perangkat Android terdapat dua cara, yaitu:**

1. **Dengan Bluetooth atau WLAN**

**Koneksi dengan fasilitas ini dilakukan dengan cara menginstal aplikasi B4A-Bridge yang dapat diunduh melalui Playstore dan diinstal pada perangkat android. Syarat untuk bisa melakukan koneksi dengan cara ini adalah komputer yang digunakan memiliki Bluetooth atau WLAN.**

**28**

1. **Dengan kabel USB**

**Aplikasi dapat dijalankan langsung menggunakan perangkat android yaitu dengan menggunakan kabel USB yang disambungkan pada komputer. Syarat yang harus dipenuhi adalah pada perangkat android pilihan Pengembang pada USB Debug harus di aktifkan. Artinya mengijinkan aplikasi yang berasal dari luar tanpa lisensi dari pihak Google dapat diinstal ke dalam perangkat android.**

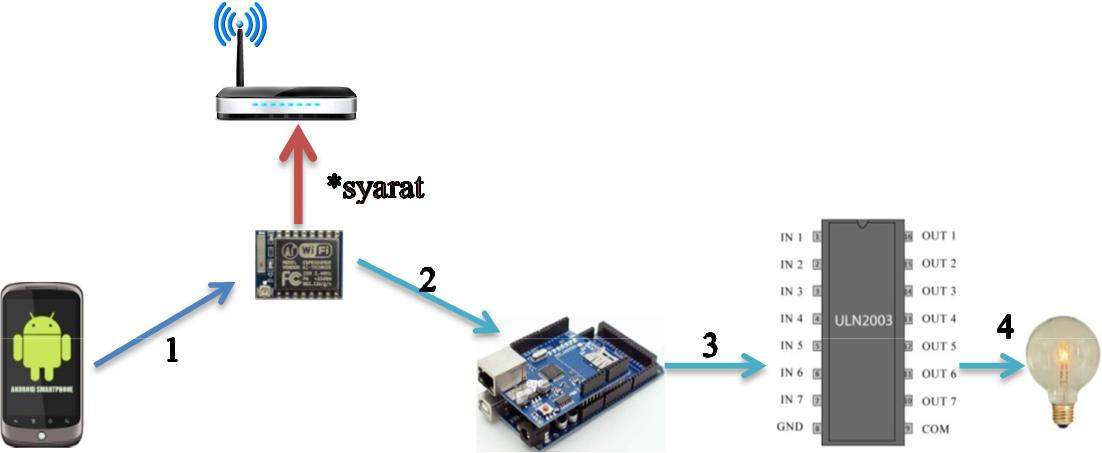
**2.6.6 Menjalankan Program**

**Setelah desain user interface dan program telah selesai dibuat, maka program siap dijalankan. Untuk menjalankan program maka dipilih menu Project – Compile&Run atau dengan tombol F5. Aplikasi dapat dijalankan di atas emulator atau langsung dijalankan melalui perangkat android sesungguhnya melalui B4A-Bridge atau melalui kabel USB.**

**BAB III**

**METODE PERANCANGAN SISTEM**

**Alat yang akan dibuat merupakan kendali berbasis wireless, topologi dari alat ini ditunjukkan pada Gambar 3.1 berikut ini:**



**Gambar 3.1 Topologi Kendali Lampu secara Wireless**

**Pera ngkat android akan bisa mengontrol lampu jika modul Esp terkoneksi dengan**

**Titik Akses (AP). Jika pada “\*Syarat” tidak terpenuhi maka kendali lampu dari**

**Android tidak akan berjalan dengan semestinya atau bahkan tidak berfungsi.**

**3.1 Alat dan Bahan**

**3.1.1 Alat**

1. **Smartphone Android**

**Handphone berbasis android digunakan untuk menjalankan aplikasi kendali lampu secara jarak jauh, dengan kata lain handphone ini berlaku sebagai pengirim data atau transmitter (Tx).**

**30**

1. **Modul Wi-Fi ESP8266-12**

**Modul penghasil sinyal Wi-Fi yang digunakan sebagai media komunikasi nirkabel dengan perangkat di luar Arduino serta sebagai penyimpan data dalam sebuah webserver.**

1. **Titik Akses (AP)**

**Ketika ESP8266-12 diaktifkan maka dia akan mencari sinyal Wi-Fi yang menjadi titik akses. Titik Akses yang digunakan adalah perangkat Hanphone yang bisa difungsikan sebagai jaringan hotspot (thatering).**

1. **Board Arduino Uno**

**Board arduino yang digunakan adalah arduino uno sebagai tempat pengelolaan program masukan dan keluaran (input – output). Pesan yang masuk dari modul ESP kemudian diterjemahkan sebagai output nyala-mati lampu.**

1. **Driver lampu**

**Driver lampu berupa saklar elektronis yang menerima masukan tegangan DC dan digunakan untuk mengontrol perangkat berbasis tegangan AC. Komponen pada driver lampu yaitu:**

* 1. **IC ULN2003**
  2. **MOC3041**
  3. **Resistor**

1. **Komputer**

**Komputer digunakan untuk membuat program pada Arduino dan membuat aplikasi android. Aplikasi yang digunakan yaitu:**

**31**

1. **Arduino IDE**

**Digunakan untuk membuat program berbasis mikro yang nantinya di uplaod ke dalam chip mikrokontroler pada kit arduino.**

1. **Basic for Android (B4A)**

**Yaitu software yang digunakan untuk membuat aplikasi berbasis android.**

**3.1.2 Bahan**

**Bahan yang digunakan pada perancangan sistem kendali lampu ini yaitu: berupa source code untuk pemrograman mikrokontroler pada kit arduino dan source code untuk pemrograman pada aplikasi B4A.**

**3.2 Langkah Kerja**

**3.2.1 Setup Modul ESP8266-12**

**Setup yang dikerjakan adalah memfungsikan modul Wi-Fi ESP8266-12 sebagai web server. Dengan fungsinya sebagai web server maka modul ESP akan mempunyai sebuah alamat IP (Internet Protocol) yang berperan sebagai gateaway, dimana alamat IP tersebut akan dapat diakses melalui browser yaitu dengan menuliskan IP pada kolom address bar pada browser. Untuk bisa mengirimkan pesan kepada ESP8266-12 maka perlu dibuat sebuah pemrograman sederhana. Penulisan program (source code) bisa dibuat melalui Arduino Ide. Data berupa pesan yang dikirimkan kepada ESP8266-12 nantinya dimanfaatkan untuk mengontrol nyala lampu.**

**32**

**3.2.2 Merancang Aplikasi Android**

**Aplikasi yang dibuat nantinya berupa file berekstensi apk yang diinstal pada perangkat Android. Rancangan apk ini dibuat menggunakan aplikasi Basic for Android (B4A) yang terinstal pada komputer atau laptop, dengan terlebih dahulu merancang user interface. User interface adalah tampilan aplikasi yang akan digunakan oleh pengguna pada perangkat Android. Aplikasi yang dijalankan pada Android berfungsi untuk memanggil alamat IP pada ESP8266-12 yang telah dirancang sebelumnya.**

**3.2.3 Merancang Aplikasi Arduino Uno**

**Board Arduino Uno digunakan sebagai antarmuka antara Android dengan lampu. Arduino Uno diprogram untuk membaca pesan yang terdapat pada ESP8266-12 dan selanjutnya pesan tersebut diterjemahkan dalam bentuk pengaktifan kaki-kaki (pin) Arduino sebagai output. Cara menterjemahkan pesan yang diterima dari ESP8266-12 dituliskan dalam bentuk pemrograman yang ditulis pada Arduino Ide.**

**3.2.4 Merancang Driver Lampu**

**Driver lampu berupa penguat sinyal output dari arduino oleh IC ULN2003. Untuk bisa mengontrol nyala lampu AC 220 volt maka dibutuhkan saklar elektronis yang memanfaatkan optocoupler 3041. Sebelum diterapkan pada keadaan sesungguhnya, ujicoba skema rangkaian driver lampu tersebut dilakukan pada papan rangkaian percobaan (project board).**

**33**

**3.2.5 Merancang Sumberdaya DC**

**Sumberdaya yang dimaksud adalah power supply yang digunakan untuk menghidupkan kit arduino. Jika pada langkah kerja pertama ujicoba dilakukan dengan sumberdaya yang berasal dari USB Port komputer, maka ketika alat sudah menjadi alat yang utuh maka dibutuhkan sebuah power supply tersendiri sebagai pemasok tegangan pada kit arduino. Selain sebagai pemasok tegangan untuk kit arduino, power supply juga digunakan sebagai pemasok tegangan pada rangkaian driver lampu.**

**3.2.6 Ujicoba Alat dan Packing**

**Setelah bagian-bagian sistem telah dikerjakan maka dilakukan ujicoba program, tujuannya tentu saja untuk mengetahui apakah sistem secara keseluruhan dapat berjalan sesuai dengan rancangan yang diinginkan. Setelah dirasa sistem telah bekerja dengan baik maka langkah terakhir adalah packing, tujuannya tidak lain adalah alat yang dibuat bisa dirapikan menjadi satu kesatuan yang utuh.**

**3.3 Skenario Pengujian**

**Skenario pengujian alat dimaksudkan untuk mengetahui jarak maksimal yang bisa dijangkau oleh modul wifi tipe ESP8266-12 dengan perangkat android. Pengujian minimal dilakukan pada tiga keadaan yaitu: dekat, menengah dan jauh. Pada kondisi jauh, yaitu antara pengirim (perangkat android) dengan penerima (modul wifi ESP 8266-12) bisa diketahui jarak maksimal yang bisa dijangkau, dengan catatan tidak ada halangan/rintangan antara pengirim dan penerima.**

**34**

**Selain itu ujicoba juga dilakukan pada keadaan yang penuh dengan halangan/rintangan, yaitu antara bagian penerima dengan pengirim terpisah oleh bangunan yang tertutup misalnya diletakkan pada ruang yang berbeda. Hal ini dimaksudkan untuk mengetahui bagaimana jangkauan sinyal wifi dari modul wifi ESP8266-12 dan perangkat android pada ruang tertutup, tujuannya adalah untuk menentukan bagaimana meletakkan modul wifi ESP8266-12 secara tepat sesuai dengan karakteristiknya ketika diimplementasikan pada keadaan yang sesungguhnya.**

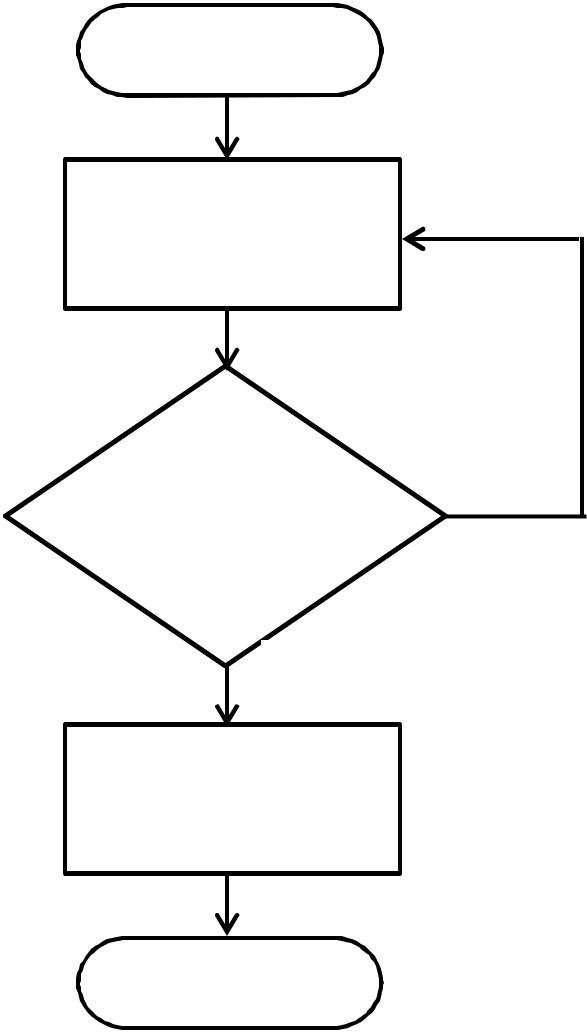
**BAB IV**

**HASIL DAN PEMBAHASAN**

**4.1 Rancangan Sistem Kendali Lampu**

**Sebelum perangkat android bisa digunakan untuk mengontrol nyala lampu, terlebih dahulu perangkat android dikoneksikan dengan modul ESP8266-12 pada pilihan jaringan Wi-Fi yang tersedia, nama yang diberikan pada modul Wi-Fi ESP8266-12 adalah ESPap. Proses koneksi antara perangkat android dan modul ESP8266-12 ditunjukkan pada Gambar 4.1 berikut ini:**

**Mulai**



**Mancari Wifi**

**(IP ESP8266-12)**

**Lakukan**

**koneksi?**

**Aplikasi kontrol**

**ON/OFF lampu**

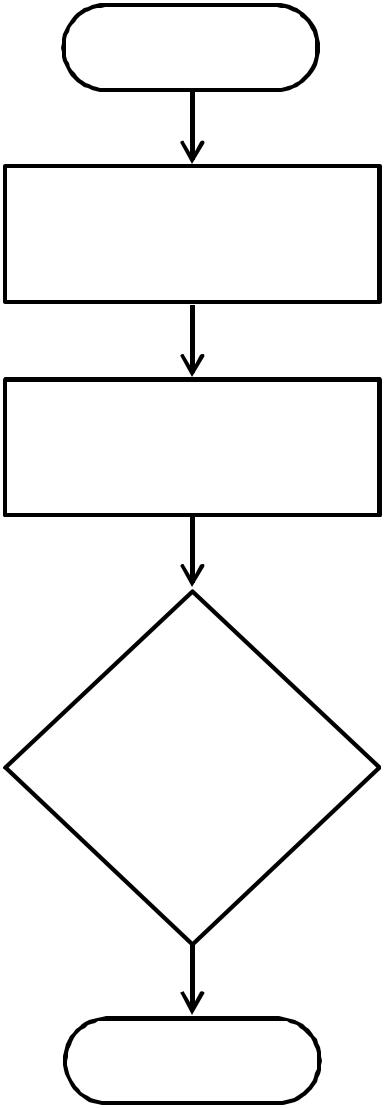
**Selesai**

**Gambar 4.1 Flowchart Koneksi Android dengan modul ESP8266-12**

**35**

**36**

**Selanjutnya setelah dilakukan koneksi, maka perangkat android dapat digunakan untuk mengontrol nyala lampu melalui aplikasi yang telah diinstal di dalamnya. Proses kontrol nyala lampu ditunjukkan pada Gambar 4.2 berikut ini:**



**Memanggil IP ESP8266-12-**

**12**

**Mengirimkan karakter’char’**

**1/a, 2/b, 3/c, 4/d, 5/e, 6/f,**

**7/g, 8/h, 0, 9**

**Jika karakter**

**= ‘char’,**

**maka Lampu**

**On/Off**

**Gambar 4.2 Flowchart kontrol ON/OFF lampu**

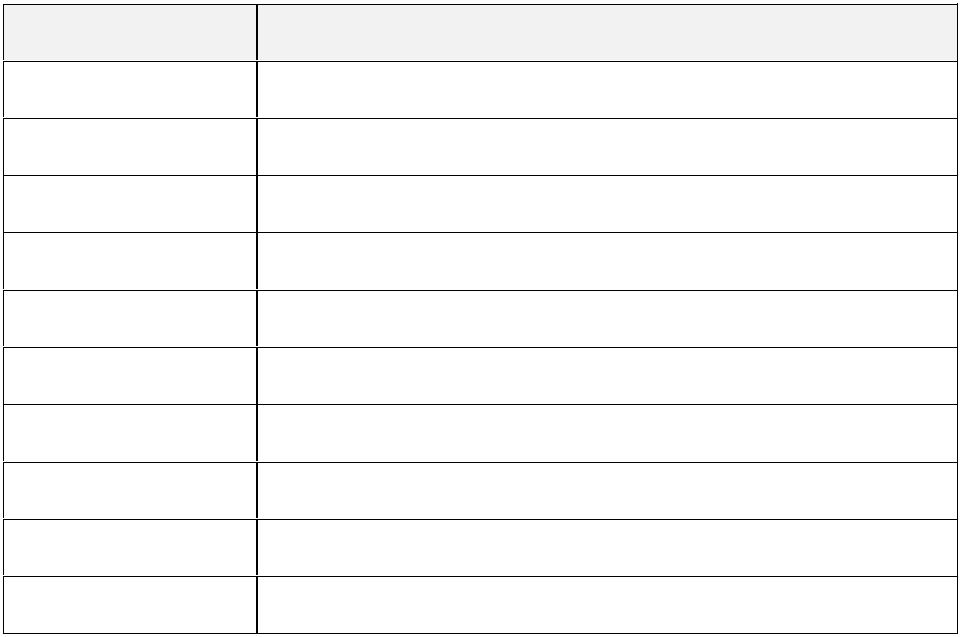
**Berdasarkan Gambar 4.2 maka kontrol nyala lampu dilakukan dengan cara mengirimkan karakter berupa huruf untuk mewakili ON dan OFF lampu 1 sampai dengan lampu 8. Aplikasi pada android berfungsi untuk memanggil alamat IP yang terdapat pada ESP8266-12 yang berisi karakter angka dan huruf. IP yang**

**37**

**diberikan oleh ESP8266-12 yang merupakan gateway-nya adalah “192.168.4.1”. Karakter yang dikirimkan kepada board Arduino Uno melalui modul ESP8266-12 ditunjukkan pada Tabel 4.1 berikut ini:**

**Tabel 4.1 Karakter huruf sebagai penanda ON/OFF lampu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Karakter** |  | **Output** |  |
| **1 dan a** | **Jika 1 maka L1 ON, jika a maka L1 OFF** | |  |
| **2 dan b** |  |  |  |
| **3 dan c** | **Jika 4 maka L4** | **ON, jika d maka L1 OFF** |  |
| **4 dan d** |  |
| **5 dan e** | **Jika 5 maka L5** | **ON, jika e maka L1 OFF** |  |
| **6 dan f** | **Jika 6 maka L6** | **ON, jika f maka L1 OFF** |  |
| **7 dan g** | **Jika 7 maka L7** | **ON, jika g maka L1 OFF** |  |
| **8 dan h** | **Jika 0 maka semua L ON** | |  |
| **0** |  |
| **9** | **Jika 9 maka semua L OFF** | |  |



**4.2 ESP8266-12 sebagai Webserver**

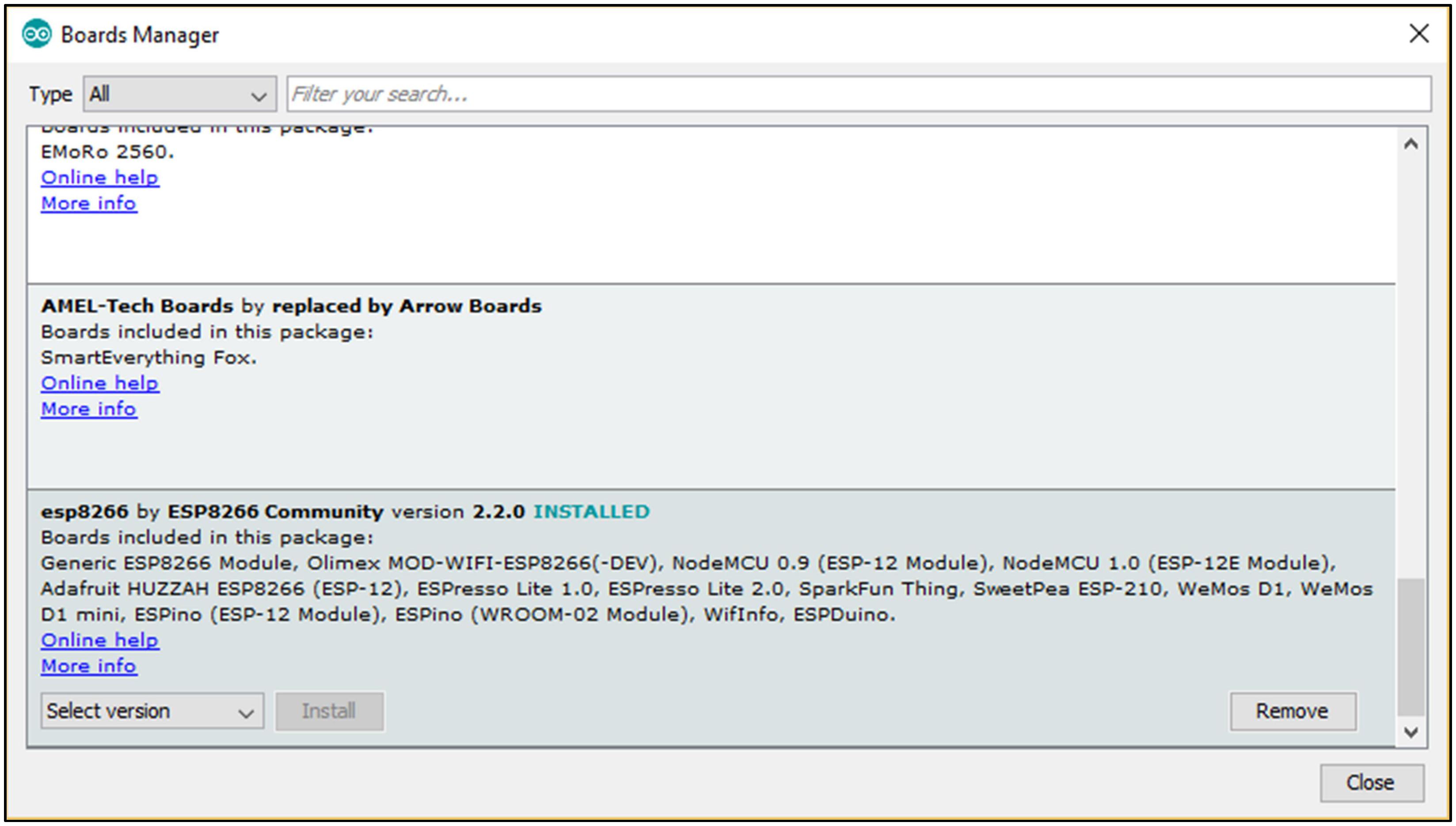
**4.2.1 Instalasi board ESP8266-12**

**Sebelum modul Wi-Fi ESP8266-12 bisa digunakan maka terlebih dahulu modul tersebut diaktifkan dengan mengisikan program tertentu ke dalam chip ESP8266-12, dalam hal ini ESP8266-12 diaktifkan sebagai web server. Untuk menuliskan (upload) pemrograman atau source code ke dalam modul ESP8266-12 dibutuhkan modul USB to Serial. Skema antara USB to Serial terhadap ESP8266-12 diperlihatkan pada Gambar 4.4. Sedangkan penulisan program dikerjakan menggunakan Arduino Ide. Ada satu syarat yang harus dikerjakan sebelum melakukan uploading program pada modul ESP yaitu mengunduh dan**

**38**

**menginstal modul ESP8266-12 pada Arduino Ide. Caranya adalah memasukkan alamat di bawah ini:**

**“http://arduino.ESP8266-12.com/stable/package\_ESP826612com\_index.json” Pada menu File – Preference, kemudian pada tab Setting masukkan alamat tersebut di atas pada kolom Additional Boards Manager URLs. Setelah itu pada menu Tools – Board arahkan pada pilihan Boards Manager, kemudian cari pilihan ESP8266-12 by ESP8266-12 Community dan kemudian lakukan instalasi.**



**Gambar 4.3 Instalasi Board ESP8266-12 Arduino Ide**

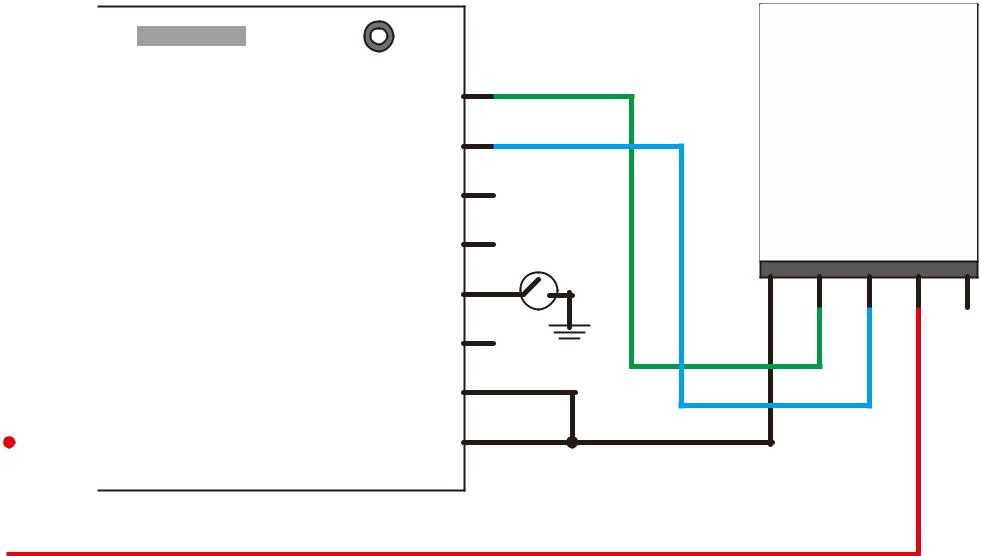
**4.2.2 Upload program web server pada ESP8266-12**

**Proses upload kode program (sketches) menggunakan Arduino Ide. Berdasarkan Gambar 4.4 untuk melakukan proses uploading, saklar S1 yaitu kaki GPIO0 harus dikoneksikan dengan ground (GND). Setelah proses selesai maka koneksi S1 harus dilepas. Saklar S2 digunakan untuk me-reset atau menyalakan ulang modul Wi-Fi.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **REST** | | **12** | |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **ADC** | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | **S2** | | | | | | |  |  | **-** | |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **EN / CH\_PD** | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **GPIO16** | | **ESP8266** | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **GPIO14** | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **GPIO12** | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **GPIO13** | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **VCC** | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**39**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TX** |  |  | **USB to TTL** |  |
|  |  |  |  |
| **RX** |  |  |  |  |
| **GPIO4** |  |  | **RXTX3,3V** |  |
| **GPIO5** | **S1** |  |  |
| **GPIO0** | **GND** | **5V** |  |
|  |  |
| **GPIO1** |  |  |  |  |
| **GPIO15** |  |  |  |  |
| **GND** |  |  |  |  |



**Gambar 4.4 Skema USB to Serial dan ESP8266-12**

**Kode pemrograman pada ESP8266-12 adalah sebagai berikut:**

**#include <ESP8266-12WiFi.h>**

**const char\* ssid = "TA Yuliana"; const char\* password = "12345678";**

**WiFiServer server(80);**

**void setup() {**

**Serial.begin(115200);**

**delay(10);**

**WiFi.begin(ssid, password);**

**Serial.println("");**

**Serial.println("WiFi connected");**

* **Start the server server.begin(); Serial.println("Server started");**

**}**

**void loop() {**

**// Check if a client has connected WiFiClient client = server.available(); if (!client) {**

**return;**

**}**

**// Wait until the client sends some data //Serial.println("new client"); while(!client.available()){ delay(1);**

**}**

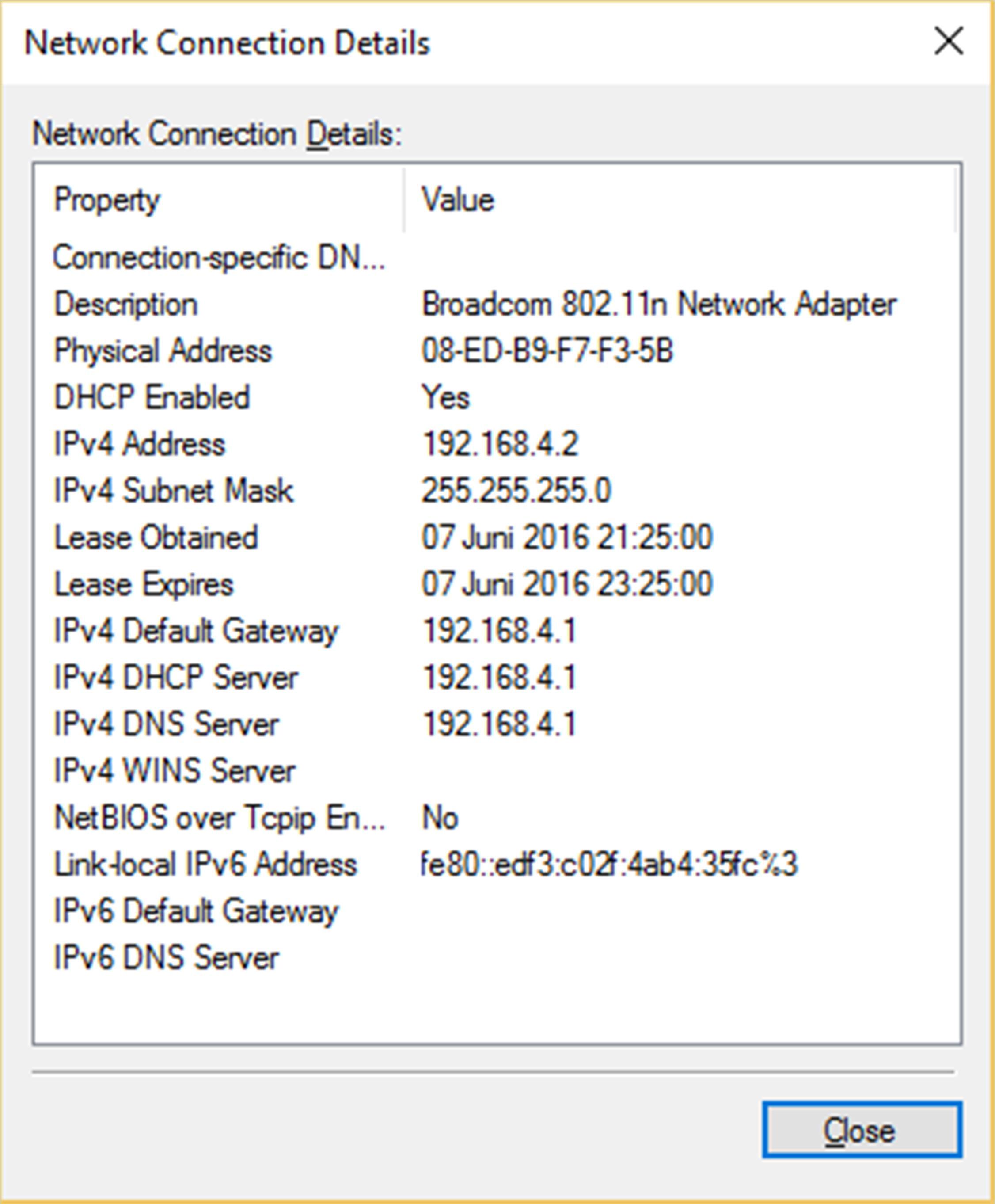
**// Read the first line of the request**

**String request = client.readStringUntil('\r'); //Serial.println(request);**

**client.flush();**

**Setelah proses uploading selesai maka Wi-Fi ESP8266-12 bisa dilihat pada daftar jaringan wireless. Nama yang muncul adalah ESPap dengan gateway 192.168.4.1 seperti diperlihatkan pada Gambar 4.5. Selanjutnya alamat IP tersebut dapat dicek melalui browser akan tetapi halaman yang ditampilkan adalah halaman kosong.**

**40**



**Gambar 4.5 Properti dari ESPap**

**Setelah uploading program web server selesai, maka langkah selanjutnya adalah membuat program yang isinya mengirimkan pesan kepada modul ESP8266-12. Pesan yang dikirimkan berupa karakter angka dan huruf seperti ditunjukkan pada Tabel 4.1. Contoh di bawah ini kode pemrograman berbasis web untuk mengirimkan pesan kepada modul ESP8266-12 untuk tiga lampu, dan untuk kode pemrograman web server secara keseluruhan dapat dilihat pada Lampiran 1.**

**char value1 = LOW;**

**char value2 = LOW;**

**char value3 = LOW;**

**if (request.indexOf("/LED1=ON") != -1) { value1 = HIGH;**

**Serial.println('1');**

**}**

**if (request.indexOf("/LED1=OFF") != -1) { value1 = LOW;**

**Serial.println('a');**

**}**

**if (request.indexOf("/LED2=ON") != -1) { value2 = HIGH;**

**Serial.println('2');**

**}**

**if (request.indexOf("/LED2=OFF") != -1) { value2 = LOW;**

**Serial.println('b');**

**}**

**if (request.indexOf("/LED3=ON") != -1) { value3 = HIGH;**

**Serial.println('3');**

**}**

**if (request.indexOf("/LED3=OFF") != -1) { value3 = LOW;**

**Serial.println('c');**

**}**

**client.print("Led pin is now: ");**

**if(value1 == HIGH) {**

**client.print("On");**

**} else {**

**client.print("Off");**

**}**

**41**

**client.println("<br><br>");**

**client.println("Click <a**

**href=\"/LED1=ON\">LED 1**

**ON</a><br>");**

**client.println("Click <a**

**href=\"/LED1=OFF\">LED 1**

**OFF</a><br>");**

**client.println("Click <a**

**href=\"/LED2=ON\">LED 2**

**ON</a><br>");**

**client.println("Click <a**

**href=\"/LED2=OFF\">LED 2**

**OFF</a><br>");**

**client.println("Click <a**

**href=\"/LED3=ON\">LED 3**

**ON</a><br>");**

**client.println("Click <a**

**href=\"/LED3=OFF\">LED 3**

**OFF</a><br>");**

**client.println("</html>");**

**delay(1);**

**}**

**Salah satu kode program yang akan ditampilkan pada halaman browser adalah:**

**client.println("Click <a href=\"/LED1=ON\">LED 1 ON</a><br>");**

**maka alamat IP yang akan dipanggil adalah 192.168.4.1/LED1=ON, sedangkan pesan data yang dikirimkan adalah pada salah satu kode program berikut ini:**

**if (request.indexOf("/LED1=ON") != -1) {**

**value1 = HIGH;**

**Serial.println('1');**

**}**

**if (request.indexOf("/LED1=OFF") != -1) {**

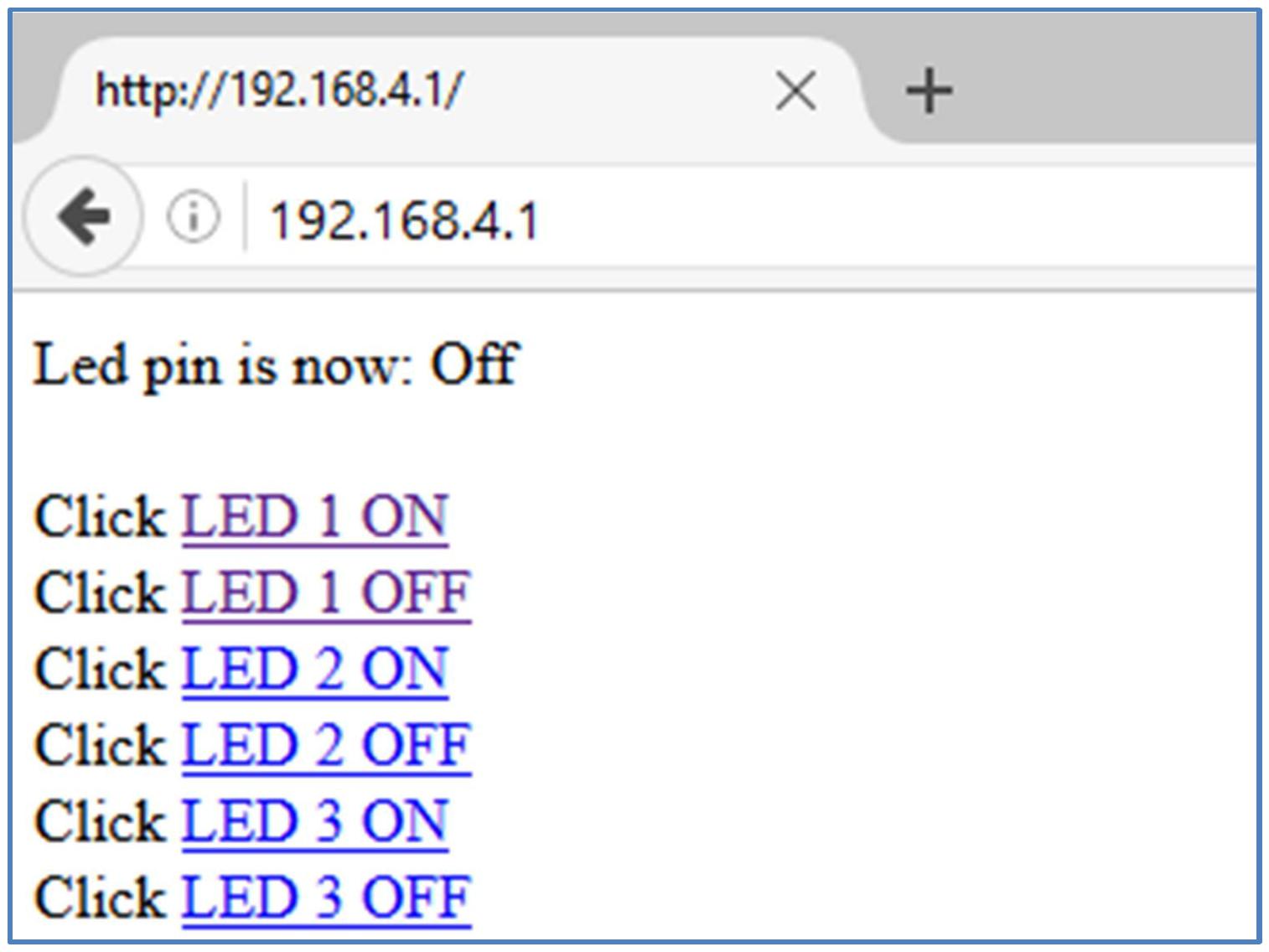
**value1 = LOW;**

**Serial.println('a');**

**}**

**Contoh kode di atas adalah kode program untuk mengirimkan pesan karakter angka “1” dan karakter huruf “a”. Tampilan secara lengkap pada browser ditunjukkan pada Gambar 4.6.**

**42**



**Gambar 4.6 Tampilan Web Browser pada ESP8266-12**

**4.3 Aplikasi Kendali Lampu pada Android**

**Aplikasi dengan ekstensi apk digunakan untuk memanggil alamat IP dengan pesan data yang telah dirancang sebelumnya. Aplikasi apk dibuat menggunakan Basic 4 Android (B4A). Aplikasi pada android berupa beberapa tombol-tombol yang akan diuraikan pada sub bab berikut ini:**

**4.3.1 Desain User Interface**

**Tampilan desain aplikasi di android ditunjukkan pada Gambar 4.7 berikut ini:**



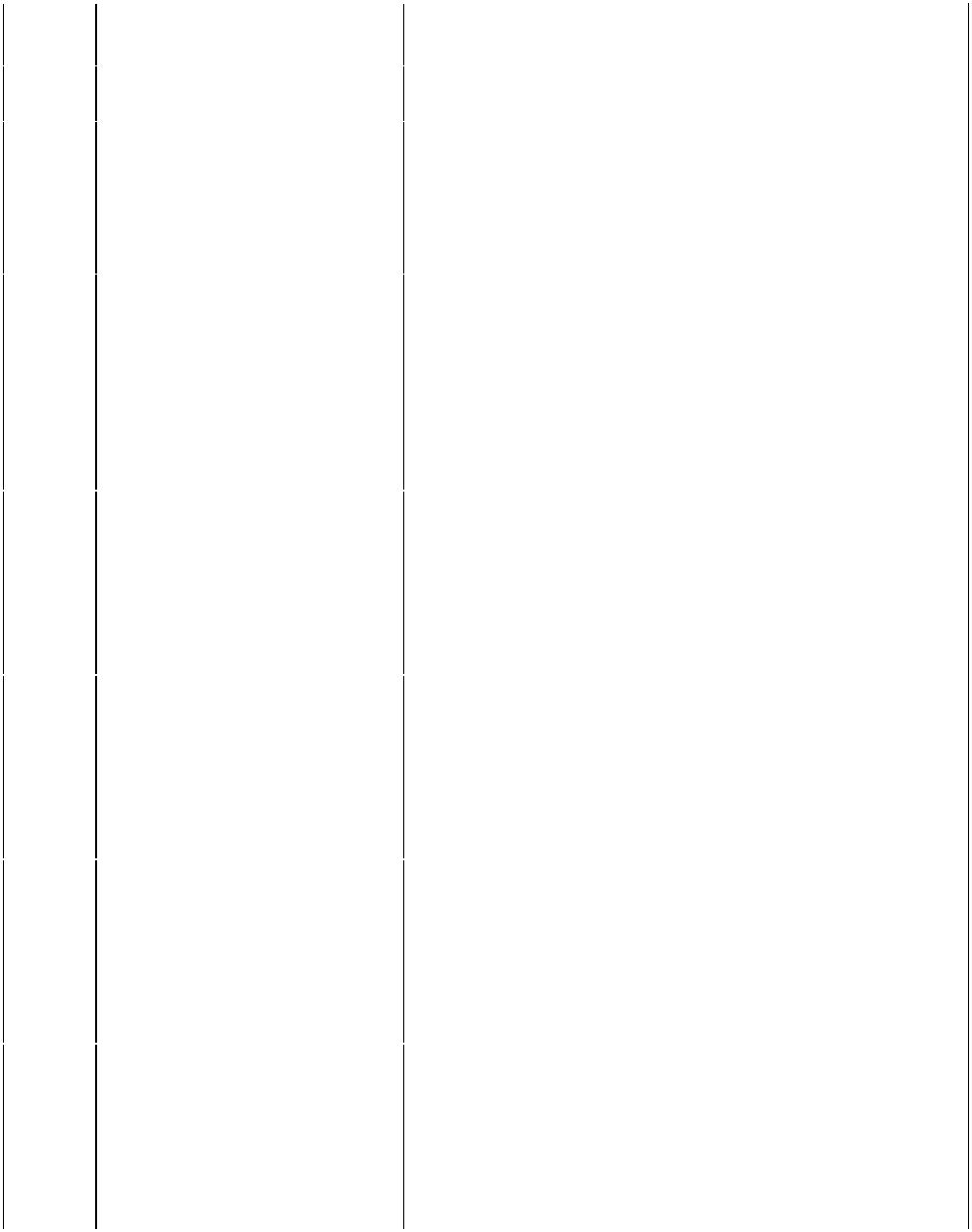
**Gambar 4.7 Desain Aplikasi Kendali Lampu**

**43**

**Daftar komponen dan properti dari tampilan aplikasi tersebut ditunjukkan pada Tabel 4.2.**

**Tabel 4.2 Daftar Komponen User Interface Aplikasi Kendali Lampu**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| **No** | **Komponen** |  |  |  | **Properti** |  |
| **1** | **Activity** |  | **Title** | **: Kendali Lampu** | |  |
| **2** | **Label1** |  | **Name** | **: Label1** | |  |
|  |  |  | **Widht** | **:** | **320** |  |
|  |  |  | **Height** | **:** | **50** |  |
|  |  |  | **Text** | **: Sistem Kendali Lampu Wirelessly** | |  |
| **6** | **Label12** |  | **Name** | **: Label12** | |  |
|  | **(Digunakan sebagai** |  | **Left/Top : 111/298** | | |  |
|  | **register)** |  | **Widht** | **:** | **100** |  |
|  |  | **Height** | **:** | **40** |  |
|  |  |  | **Text** | **: (dikosongkan)** | |  |
|  |  |  | **Visible** | **: false** | |  |
| **7** | **Button1, 3, 5, 7** |  | **Name** | **: BtnL1, L3, L5, L7** | |  |
|  |  |  | **Left/Top : 25/(63, 123, 183, 243)** | | |  |
|  |  |  | **Widht** | **:** | **130** |  |
|  |  |  | **Height** | **:** | **50** |  |
|  |  |  | **Text** | **:** | **ON** |  |
| **8** | **Button2, 4, 6, 8** |  | **Name** | **:** | **BtnL2, L4, L6, L8** |  |
|  |  |  | **Left/Top :** | | **166/(63, 123, 183, 243)** |  |
|  |  |  | **Widht** | **:** | **130** |  |
|  |  |  | **Height** | **:** | **50** |  |
|  |  |  | **Text** | **:** | **ON** |  |
| **9** | **Button9, Button10** |  | **Name** | **:** | **BtnOnAll, BtnOffAll** |  |
|  |  |  | **Left/Top :** | | **(26,166 )/341** |  |
|  |  |  | **Widht** | **:** | **130** |  |
|  |  |  | **Height** | **:** | **50** |  |
|  |  |  | **Text** | **:** | **ON ALL, OFF ALL** |  |
| **10** | **Button11** |  | **Name** | **:** | **BtnExit** |  |
|  |  |  | **Left/Top :** | | **26/402** |  |
|  |  |  | **Widht** | **:** | **270** |  |
|  |  |  | **Height** | **:** | **50** |  |
|  |  |  | **Text** | **:** | **EXIT** |  |



**4.3.2 Pemrograman Android**

**Aplikasi apk dibuat menggunakan SDK 16 kemudian pada B4A versi android yang digunakan adalah 4.1 Jelly Bean. Berdasar pada Gambar 4.7, fungsi tombol**

**44**

**pada aplikasi sistem kendali lampu ini digunakan untuk memanggil alamat IP pada web server modul ESP8266-12. Salah satu kode program pada fungsi tombol, misalnya tombol “L1 ON” ditunjukkan pada kode di bawah ini, untuk kode pemrograman secara lengkap ditunjukkan pada Lampiran 2.**

**Sub BtnL1\_Click**

**If BtnL1.Text="L1 ON" Then**

**BtnL1.Text="L1 OFF"**

**BtnL1.Color=Colors.ARGB(128,200,0,0)**

**BtnOFFall.Enabled=True**

**BtnOFFall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 1 nyala","false")**

**esp.InitializeGet("http://192.168.4.1/LED1=ON")**

**droid.Execute(esp,1)**

**Else If BtnL1.Text="L1 OFF" Then**

**BtnL1.Text="L1 ON"**

**BtnL1.Color=Colors.ARGB(128,50,50,0)**

**BtnONall.Enabled=True**

**BtnONall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 1 mati","false")**

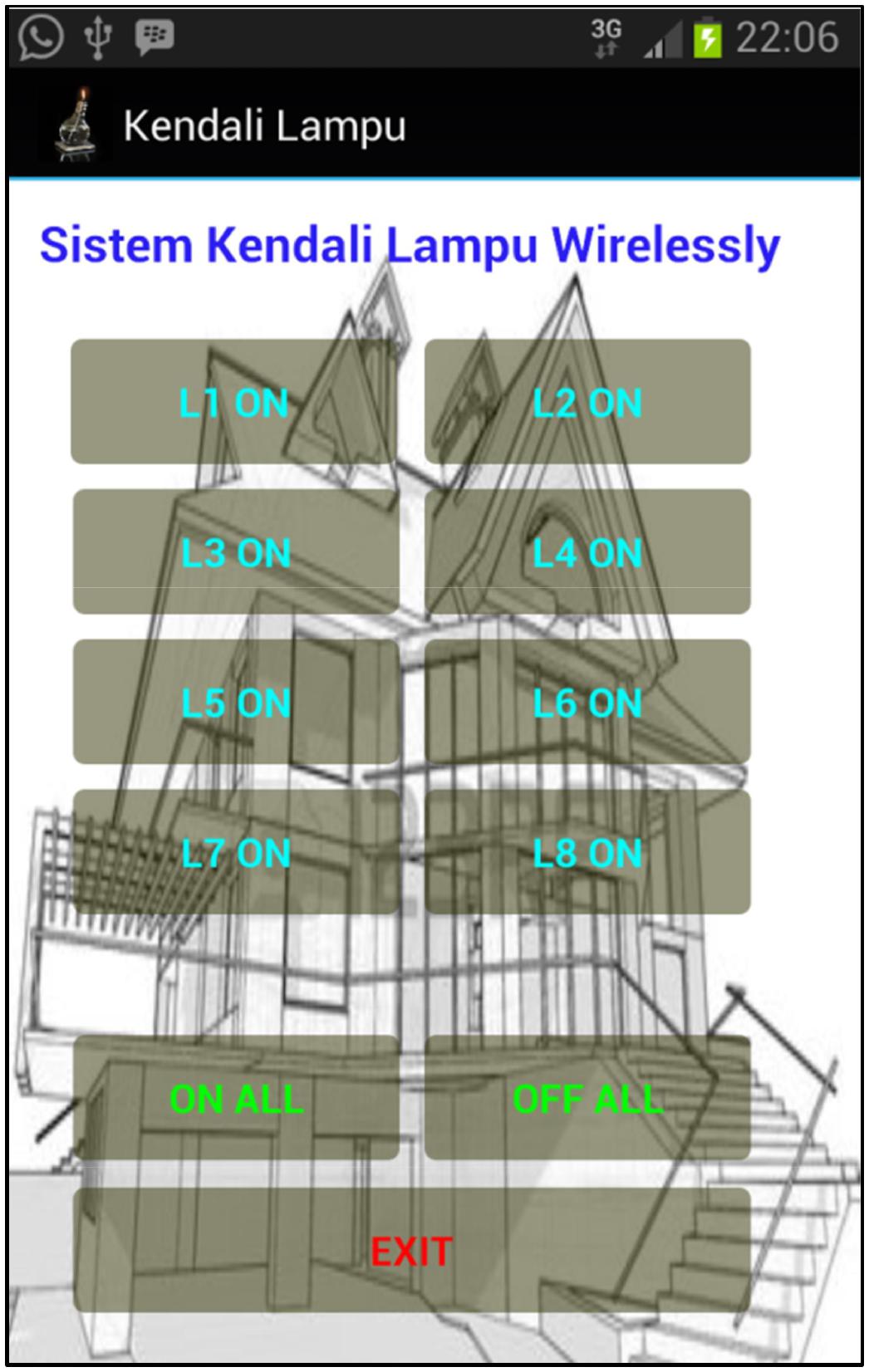
**esp.InitializeGet("http://192.168.4.1/LED1=OFF")**

**End If** **droid.Execute(esp,1)**

**End Sub**

**Begitu juga tombol-tombol yang lain mempunyai kode pemrograman yang sama, yang membedakan adalah nama urut dari LED (LED1 – LED8) yang dikendalikan. Sedangkan tombol ON All digunakan untuk menyalakan semua lampu, tombol OFF All untuk mematikan semua lampu dan tombol Exit untuk menutup aplikasi.**

**45**



**Gambar 4.8 Tampilan Aplikasi di Android**

**4.4 Pemrograman pada Arduino Uno**

**4.4.1 Seting Awal Arduino Uno**

**Arduino Uno digunakan untuk membaca data dari modul Wi-Fi ESP8266-12 melalui koneksi serial memanfaatkan pin Rx dan Tx. Walaupun pada board Arduino Uno terdapat pin Rx dan Tx namun pada pemrogramannya pin digital kaki 10, 11 digunakan sebagai kaki Rx dan Tx. Sedangkan pin yang digunakan sebagai keluaran (output) adalah pin digital nomor 2 hingga 9. Kode program pada Void Setup adalah sebagai berikut:**

**#include <SoftwareSerial.h>**

**SoftwareSerial ESP8266(10, 11); // RX, TX**

**//menentukan tipe data dari ESP8266 berbentuk CHAR dengan nama inChar char inChar;**

**void setup() {**

**//menseting komunikasi serial:**

**46**

**Serial.begin(9600);**

**ESP8266.begin(115200);**

**//menseting pin output:**

**pinMode(2, OUTPUT); digitalWrite(2, LOW); //lampu 1**

**pinMode(3, OUTPUT); digitalWrite(3, LOW);**

**pinMode(4, OUTPUT); digitalWrite(4, LOW);**

**pinMode(5, OUTPUT); digitalWrite(5, LOW);**

**pinMode(6, OUTPUT); digitalWrite(6, LOW);**

**pinMode(7, OUTPUT); digitalWrite(7, LOW);**

**pinMode(8, OUTPUT); digitalWrite(8, LOW);**

**} pinMode(9, OUTPUT); digitalWrite(9, LOW); //Lampu 8**

**4.4.2 Menterjemahkan Data sebagai Output**

**Data yang diterjemahkan sebagai output adalah pesan karakter yang diterima dari ESP8266-12 melalui android. Pemrograman ini dituliskan pada Void Loop. Kode pemrograman secara lengkap dapat dilihat pada Lampiran 3, di bawah ini adalah kode pemrograman bagaimana Arduino Uno menterjemahkannya sebagai output:**

**void loop() {**

**if (ESP8266.available()) {**

**inChar = ESP8266.read();**

**Serial.write(inChar);**

**//menterjemahkan data**

**if (inChar == '1' )**

**{ digitalWrite(2, HIGH); }**

**else if(inChar == 'a')**

**{ digitalWrite(2, LOW); }**

**if (inChar == '2' )**

**{ digitalWrite(3, HIGH); }**

**else if(inChar == 'b')**

**{ digitalWrite(3, LOW); }**

**if (inChar == '3' )**

**{ digitalWrite(4, HIGH); }**

**else if(inChar == 'c')**

**{ digitalWrite(4, LOW); }**

**if (inChar == '4' )**

**{ digitalWrite(5, HIGH); }**

**else if(inChar == 'd')**

**{ digitalWrite(5, LOW); }**

**if (inChar == '5' )**

**{ digitalWrite(6, HIGH); }**

**else if(inChar == 'e')**

**{ digitalWrite(6, LOW); }**

**if (inChar == '6' )**

**{ digitalWrite(7, HIGH); }**

**else if(inChar == 'f')**

**{ digitalWrite(7, LOW); }**

**if (inChar == '7' )**

**{ digitalWrite(8, HIGH); }**

**else if(inChar == 'g')**

**{ digitalWrite(8, LOW); }**

**if (inChar == '8' )**

**{ digitalWrite(9, HIGH); }**

**else if(inChar == 'h')**

**{ digitalWrite(9, LOW); }**

**//Lampu on semua**

**if (inChar == '0' )**

**{ digitalWrite(2, HIGH);**

**digitalWrite(3, HIGH);**

**digitalWrite(4, HIGH);**

**digitalWrite(5, HIGH);**

**digitalWrite(6, HIGH);**

**digitalWrite(7, HIGH);**

**digitalWrite(8, HIGH);**

**digitalWrite(9, HIGH);}**

**//Lampu off semua**

**else if(inChar == '9')**

**{ digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**digitalWrite(6, LOW);**

**digitalWrite(7, LOW);**

**digitalWrite(8, LOW);**

**digitalWrite(9, LOW);}}**

**47**

**4.5 Driver Lampu dengan MOC3041**

|  |
| --- |
| **Output Dari Arduino UNO** |

**IN** **R9**



**330RR10**



**330RR11**



**330RR12**



**330R**

**R13**



**330RR14**



**330RR15**



**330RR16**

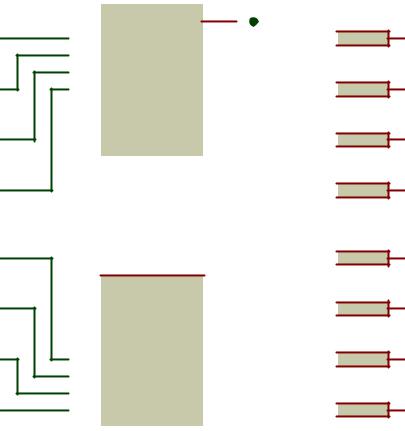


**330R**

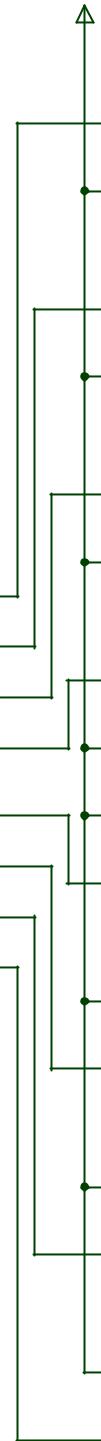
**VCC 5V**



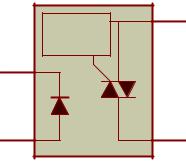
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **U9** | | |  |  | **9** |  |  |  |  |  |  | **R1** |  |
|  |  |  |  | **COM** |  |  |  |  |  |  |  |  |
|  |  |  | **ULN2003A** | |  |  |  |  |  |  |  |  |  |
| **1** |  |  | **1B** | **1C** |  | **16** |  |  |  |  |  |  |  |  |
| **2** |  |  | **2B** | **2C** |  | **15** |  |  |  |  |  |  | **330R** |  |
| **3** |  |  | **3B** | **3C** |  | **14** |  |  |  |  |  |  | **R2** |  |
| **4** |  |  | **4B** | **4C** |  | **13** |  |  |  |  |  |  |  |  |
| **5** |  |  | **5B** | **5C** |  | **12** |  |  |  |  |  |  | **330R** |  |
| **6** |  |  | **6B** | **6C** |  | **11** |  |  |  |  |  |  | **R3** |  |
| **7** |  |  |  | **10** |  |  |  |  |  |  |  |
|  |  | **7B** | **7C** |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | **330R** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | **R4** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | **330R** |  |
|  | **U10** | | |  |  |  |  |  |  |  |  |  | **R5** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | **ULN2003A** | |  | **9** |  |  |  |  |  |  | **330R** |  |
|  |  |  |  |  |  |  |  |  |  |
| **1** |  |  |  | **COM** |  | **16** |  |  |  |  |  |  | **R6** |  |
|  |  | **1B** | **1C** |  |  |  |  |  |  |  |  |  |
| **2** |  |  | **2B** | **2C** |  | **15** |  |  |  |  |  |  | **330R** |  |
| **3** |  |  | **3B** | **3C** |  | **14** |  |  |  |  |  |  | **R7** |  |
| **4** |  |  |  | **13** |  |  |  |  |  |  |  |
|  |  | **4B** | **4C** |  |  |  |  |  |  |  |  |  |
| **5** |  |  | **5B** | **5C** |  | **12** |  |  |  |  |  |  | **330R** |  |
| **6** |  |  | **6B** | **6C** |  | **11** |  |  |  |  |  |  | **R8** |  |
| **7** |  |  |  | **10** |  |  |  |  |  |  |
|  |  | **7B** | **7C** |  |  |  |  |  |  |  | **330R** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |



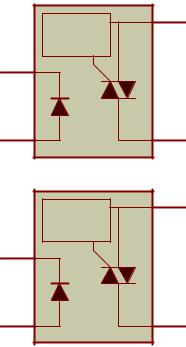
**VCC 5V**



**U1**

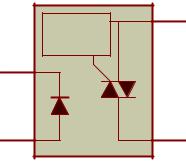


|  |  |  |
| --- | --- | --- |
|  | **Crossing** |  |
| **2** | **Zero** |  |
|  |  |
| **1** | **MOC3041M** |  |
|  |  |
|  | **U2** |  |
|  | **Crossing** |  |
| **2** | **Zero** |  |
|  |  |
| **1** | **MOC3041M** |  |
|  |  |
|  | **U3** |  |
|  | **Crossing** |  |
| **2** | **Zero** |  |
|  |  |



**1** **MOC3041M**

**U4**



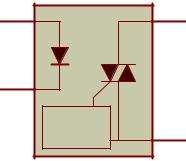
**Crossing**

**2** **Zero**

**1** **MOC3041M**

**U5**

**1** **MOC3041M**



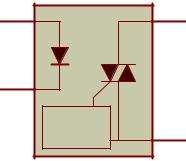
**2**

**Zero**

**Crossing**

**U6**

**1** **MOC3041M**



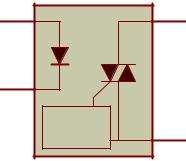
**2**

**Zero**

**Crossing**

**U7**

**1** **MOC3041M**



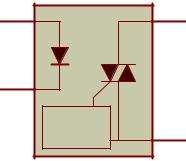
**2**

**Zero**

**Crossing**

**U8**

**1** **MOC3041M**

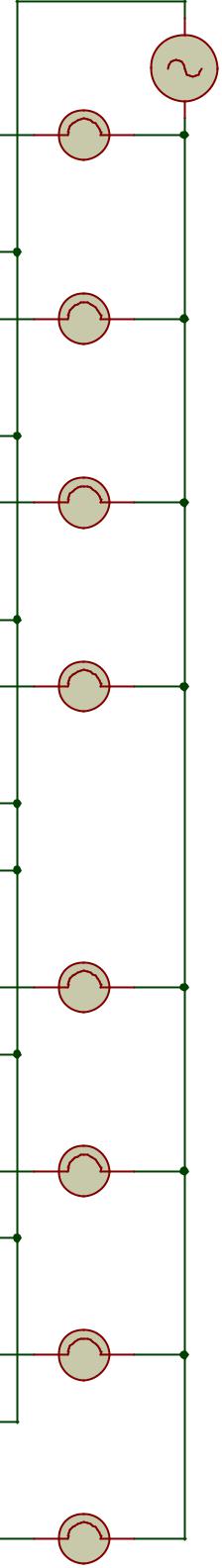


**2**

**Zero**

**Crossing**

**220 ACV**



**L1**

**4**

**6**

**L2**

**4**

**6**

**L3**

**4**

**6**

**L4**

**4**

**6**

**6**

**L5**

**4**

**6**

**L6**

**4**

**6**

**L7**

**4**

**6**

**L8**

**4**

**Gambar 4.9 Skema Driver Lampu AC**

**48**

**Keluaran dari Arduino Uno yang telah dirancang sebelumnya berupa besaran DC atau merupakan keluaran tegangan DC. Sedangkan kendali yang diinginkan adalah lampu dengan sumber arus AC. Sehingga diperlukan suatu rangkaian atau komponen yang bisa menjembatani antara tegangan DC dan AC.**

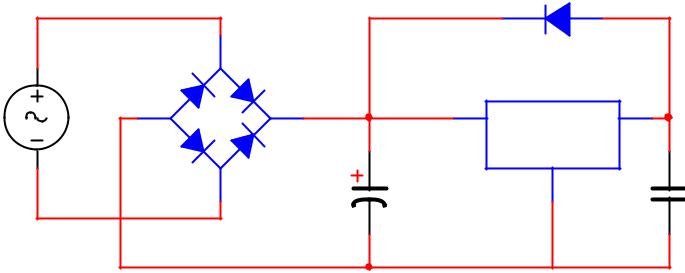
**MOC3041 adalah komponen optocoupler yang bisa digunakan sebagai saklar elektronik yang inputannya berupa tegangan DC. Untuk memperkuat arus yang dikeluarkan oleh Arduino Uno maka ditambahkan komponen yaitu ULN2003. Rangkaian driver lampu menggunakan MOC3041 dan ULN2003 ditunjukkan pada Gambar 4.9.**

**4.6 Sumber Tegangan DC**

**Rangkaian kendali lampu secara wireless terdiri dari beberapa susunan, yaitu Arduino Uno, Modul ESP8266-12 dan rangkaian driver lampu. Untuk board Arduino Uno dan rangkaian driver lampu tegangan yang dibutuhkan minimal 5 volt DC, sedangkan untuk modul ESP8266-12 tegangan yang dibutuhkan adalah 3,3 volt DC yang bisa diambilkan dari board Arduino Uno.**

**Rangkaian catu daya DC menggunakan konfigurasi penyearah 4 dengan diregulasi menggunakan IC7805. Skema rangkaian penyearah yang digunakan ditunjukkan pada Gambar 4.10.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **V1** |  |  | **1N4007** | |  | **D2** |  |  |  |
| **9 Vrms** | **D1** | **2** |  | **LM7805CT** | | | |  |  |
|  |  |  |  |
|  |  | **1** |  |  |
| **50 Hz** |  | **4** |  | **LINE** | |  | **VREG** |  |  |
| **0°** |  |  |  | **C1** | **VOLTAGE** | | |  | **C2** |  |
|  |  |  | **3** | **U1** | **COMMON** | | |  |
|  |  |  | **1mF** |  |  |  | **100nF** |  |



**Gambar 4.10 Skema Catu Daya DC 5****Volt**

**49**

**4.7 Cara Kerja Alat**

**Cara kerja Sistem Kendali Lampu secara wireless (SKLW) diuraikan sebagai berikut:**

1. **Mengaktifkan rangkaian kendali lampu yaitu Arduino Uno dan modul Wi-Fi ESP8266-12 dengan menyambungkannya kepada sumber tegangan DC.**
2. **Melakukan koneksi kepada Wi-Fi ESP8266-12 melalui perangkat android.**
3. **Menjalankan aplikasi Sistem Kendali Lampu yang telag diinstal pada android.**
4. **Memilih tombol kendali lampu yang diinginkan untuk melakukan perintah ON atau OFF.**
5. **Jika selesai, tekan tombol Exit untuk keuar dari aplikasi.**

**Pertamakali aplikasi SKLW dijalankan, keadaan awal tombol perintah pada aplikasi berada pada keadaan OFF. Sehingga, tidak ada umpan balik dari output lampu yang memberitahukan kepada aplikasi SKLW apakah dia aktif menyala atau tidak. Namun keadaan ini tidak mempengaruhi kinerja program yang ada di dalamnya. Sebagai contoh: keadaan awal Lampu1 menyala, kemudian aplikasi SKLW dijalankan. Untuk mematikan Lampu1 maka harus terlebih dahulu menekan tombol Lampu 1 pada keadaan ON dan baru kemudian menekannya lagi sehingga merubah kondisi Lampu1 mati.**

**4.8 Ujicoba Sistem**

**Ujicoba sistem bertujuan untuk mengetahui kinerja alat secara keseluruhan. Ujicoba juga bertujuan untuk mengetahui jangkauan dari modul ESP8266-12 bisa diakses oleh perangkat mobile. Untuk mengetahui kinerja sistem dilakukan pada**

**50**

**dua skenario yaitu: (1) pengujian alat pada ruang terbuka, (2) pengujian dilakukan pada ruang tertutup. Pada pengujian kedua terdapat dua kemungkinan yaitu yang pertama modul Wi-Fi berada di dalam ruang tertutup dan diakses oleh androidyang berada di luar ruangan dan yang kedua adalah sebaliknya.**

**4.8.1 Skenario 1: Ruang terbuka**

**Ujicoba skenario 1 dilakukan di kampus Politeknik Pratama Mulia Surakarta di lantai 4, di depan ruang 401 hingga 408. Modul Wi-Fi ESP8266-12 dan Akses Poin diletakkan di depan ruang 401, kemudian dilakukan uji alat setiap meter menuju ruang 408. Pada jarak di bawah 20 meter lampu masih dapat dikendalikan melalui Android, kemudian pada jarak di atas 20 meter yaitu pada jarak 21 meter sinyal Wi-Fi dari ESP8266-12 mulai melemah. Hal ini menyebabkan kendali lampu tidak dapat dilakukan.**

**4.8.2 Skenario 2: Ruang tertutup**

**Ujicoba skenario 2 dilakukan dengan cara menempatkan modul Wi-Fi ESP8266-12 dan Akses Poin di dalam Laboratorium Elektronika dalam keadaan tertutup, kemudian kendali android berada di luar ruangan. Hasil yang diberikan adalah ketika Android berada tepat di dekat pintu masuk laboratorium sinyal Wi-Fi ESP8266-12 masih dapat ditangkap oleh Android. Kemudian dengan bergeser 4 meter dari pintu laboratorium sinyal tidak dapat ditangkap.**

**Dari kedua skenario pengujian yang dilakukan menunjukkan bahwa modul Wi-Fi ESP8266-12 akan baik digunakan dalam ruang terbuka atau dengan sedikit halangan/ rintangan yang terdapat di sekitarnya.**

**BAB V**

**KESIMPULAN DAN SARAN**

**5.1 Kesimpulan**

**Dari hasil perancangan dan ujicoba sistem kendali lampu secara wireless yang telah dilakukan, dapat diambil beberapa kesimpulan:**

**a. Android dapat dimanfaatkan sebagai alat kendali lampu dengan memanfaatkan fitur Wi-Fi yang tertanam di dalamnya.**

**b. Kit Arduino Uno digunakan untuk membangun komunikasi dengan perangkat luar (lampu).**

1. **Transfer data ESP8266 dengan Arduino Uno menggunakan komunikasi serial melalui Tx dan Rx.**
2. **Jangkauan modul ESP8266 mencapai 30 meter pada lingkungan terbuka.**

**5.2 Saran**

**Beberapa saran yang bisa dikerjakan untuk pengembangan selanjutnya, diantaranya:**

1. **Ditambahkan sistem umpan balik dari output (lampu) yang dikendalikan, untuk menandakan kondisi awal dari perangkat yang dikendalikan.**
2. **Sistem kendali yang dibuat mempunyai jangkaun jaringan lokal, sehingga bisa dikembangkan sistem yang mempunyai jangkauan global seperti terkoneksi dengan internet.**
3. **Modul ESP8266-12 bisa diaktifkan sebagai stand alone sehingga tidak memerlukan Akses Poin (AP) dalam kinerjanya.**

**51**

**DAFTAR PUSTAKA**

**“Active RFID,” http://www.technovelgy.com. [Diakses: 21-May-2016].**

**“An Introduction to Infrared Technology: Applications in the Home, Classroom, Workplace, and Beyond”. http://trace.wisc.edu/docs/ir\_intro/ir\_intro.htm. [Diakses: 09-Jun-2016].**

**“Arduino - Products.” https://www.arduino.cc/en/Main/Products. [Diakses: 11-Jun-2016].**

**“Arduino** **UNO India,” Robomart. https://www.robomart.com/arduino-uno-online-india. [Diakses: 11-Jun-2016].**

**“Passive RFID,” http://www.technovelgy.com. 21-May-2016.**

**“RFID Tag Types,” RFID Labeling. Miles Technologies, Inc.**

**“What is IR wireless (infrared wireless)? -Definition from What Is.com,”**

**SearchMobileComputing.**

**http://searchmobilecomputing.techtarget.com/definition/IR-wireless.**

**[Diakses: 09-Jun-2016].**

**G. Chang, C. Tan, G. Li, and C. Zhu, “Developing Mobile Applications on the Android Platform,” in Mobile Multimedia Processing, X. Jiang, M. Y. Ma, and C. W. Chen, Eds. Springer Berlin Heidelberg, 2010, pp. 264–286.**

**George Roussos, Network RFID. London: Springer, 2008.**

**Jianye Liu and Jiankun Yu, “Research on Development of Android Applications,” in Intelligent Networks and Intelligent Systems (ICINIS), 2011 4th International Conference on, 2011, pp. 69–72.**

**Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen, “A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” presented at the Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE, 2007, pp. 46–51.**

**S. Li, B. Liu, B. Chen, and Y. Lou, “Neural network based mobile phone localization using Bluetooth connectivity,” Neural Comput. Appl., vol. 23, no. 3–4, pp. 667–675, 2013.**

**52**

**53**

**LAMPIRAN**

**Lampiran 1. Kode Pemrograman Web Server ESP8266-12**

**#include <ESP8266WiFi.h>**

**const char\* ssid = "TA Yuliana";**

**const char\* password = "12345678";**

**int ledPin = 2; // GPIO2**

**WiFiServer server(80);**

**void setup() {**

**Serial.begin(115200);**

**delay(10);**

**pinMode(ledPin, OUTPUT);**

**digitalWrite(ledPin, LOW);**

* **Connect to WiFi network Serial.println(); Serial.println(); Serial.print("Connecting to "); Serial.println(ssid);**

**WiFi.begin(ssid, password);**

**//while (WiFi.status() != WL\_CONNECTED) { //delay(500);**

**//Serial.print(".");**

**//}**

**Serial.println("");**

**Serial.println("WiFi connected");**

* **Start the server server.begin(); Serial.println("Server started");**

**}**

**void loop() {**

**// Check if a client has connected WiFiClient client = server.available(); if (!client) {**

**return;**

**}**

**// Wait until the client sends some data //Serial.println("new client"); while(!client.available()){ delay(1);**

**}**

**// Read the first line of the request**

**String request = client.readStringUntil('\r'); //Serial.println(request);**

**client.flush();**

**// Match the request**

**int value1 = LOW;**

**int value2 = LOW;**

**int value3 = LOW;**

**int value4 = LOW;**

**int value5 = LOW;**

**int value6 = LOW;**

**int value7 = LOW;**

**int value8 = LOW;**

**int valueA = LOW;**

**if (request.indexOf("/LED1=ON") != -1) { digitalWrite(ledPin, LOW);**

**value1 = HIGH;**

**Serial.println('1');**

**}**

**if (request.indexOf("/LED1=OFF") != -1) { digitalWrite(ledPin, HIGH);**

**value1 = LOW;**

**Serial.println('a');**

**}**

**if (request.indexOf("/LED2=ON") != -1) { digitalWrite(ledPin, LOW);**

**value2 = HIGH;**

**Serial.println('2');**

**}**

**if (request.indexOf("/LED2=OFF") != -1) { digitalWrite(ledPin, HIGH);**

**value2 = LOW;**

**Serial.println('b');**

**}**

**if (request.indexOf("/LED3=ON") != -1) { digitalWrite(ledPin, LOW);**

**value3 = HIGH;**

**Serial.println('3');**

**}**

**if (request.indexOf("/LED3=OFF") != -1) { digitalWrite(ledPin, HIGH);**

**value3 = LOW;**

**Serial.println('c');**

**}**

**if (request.indexOf("/LED4=ON") != -1) { digitalWrite(ledPin, LOW);**

**value4 = HIGH;**

**Serial.println('4');**

**}**

**if (request.indexOf("/LED4=OFF") != -1) { digitalWrite(ledPin, HIGH);**

**value4 = LOW;**

**Serial.println('d');**

**}**

**if (request.indexOf("/LED5=ON") != -1) { digitalWrite(ledPin, LOW);**

**value5 = HIGH;**

**Serial.println('5');**

**}**

**if (request.indexOf("/LED5=OFF") != -1) { digitalWrite(ledPin, HIGH);**

**value5 = LOW;**

**Serial.println('e');**

**}**

**if (request.indexOf("/LED6=ON") != -1) { digitalWrite(ledPin, LOW);**

**value6 = HIGH;**

**Serial.println('6');**

**}**

**if (request.indexOf("/LED6=OFF") != -1) { digitalWrite(ledPin, HIGH);**

**value6 = LOW;**

**Serial.println('f');**

**}**

**if (request.indexOf("/LED7=ON") != -1) { digitalWrite(ledPin, LOW);**

**value7 = HIGH;**

**Serial.println('7');**

**}**

**if (request.indexOf("/LED7=OFF") != -1) { digitalWrite(ledPin, HIGH);**

**value7 = LOW;**

**Serial.println('g');**

**}**

**if (request.indexOf("/LED8=ON") != -1) { digitalWrite(ledPin, LOW);**

**value8 = HIGH;**

**Serial.println('8');**

**}**

**if (request.indexOf("/LED8=OFF") != -1) { digitalWrite(ledPin, HIGH);**

**value8 = LOW;**

**Serial.println('h');**

**}**

**if (request.indexOf("/LED=ON") != -1) { digitalWrite(ledPin, LOW);**

**valueA = HIGH;**

**Serial.println('0');**

**}**

**if (request.indexOf("/LED=OFF") != -1) { digitalWrite(ledPin, HIGH); delay(500);**

**valueA = LOW;**

**Serial.println('9');**

**}**

**54**

* **Return the response client.println("HTTP/1.1 200 OK"); client.println("Content-Type: text/html"); client.println(""); // do not forget this one client.println("<!DOCTYPE HTML>"); client.println("<html>");**

**client.print("Led pin is now: ");**

**if(value1 == HIGH) {**

**client.print("On");**

**} else {**

**client.print("Off");**

**}**

**client.println("<br><br>");**

**client.println("Click <a href=\"/LED1=ON\">LED 1 ON</a><br>");**

**client.println("Click <a href=\"/LED1=OFF\">LED 1 OFF</a><br>");**

**client.println("Click <a href=\"/LED2=ON\">LED 2 ON</a><br>");**

**client.println("Click <a href=\"/LED2=OFF\">LED 2 OFF</a><br>");**

**client.println("Click <a href=\"/LED3=ON\">LED 3 ON</a><br>");**

**client.println("Click <a href=\"/LED3=OFF\">LED 3 OFF</a><br>");**

**client.println("Click <a href=\"/LED4=ON\">LED 4 ON</a><br>");**

**client.println("Click <a href=\"/LED4=OFF\">LED 4 OFF</a><br>");**

**client.println("Click <a href=\"/LED5=ON\">LED 5 ON</a><br>");**

**client.println("Click <a href=\"/LED5=OFF\">LED 5 OFF</a><br>");**

**client.println("Click <a href=\"/LED6=ON\">LED 6 ON</a><br>");**

**client.println("Click <a href=\"/LED6=OFF\">LED 6 OFF</a><br>");**

**client.println("Click <a href=\"/LED7=ON\">LED 7 ON</a><br>");**

**client.println("Click <a href=\"/LED7=OFF\">LED 7 OFF</a><br>");**

**client.println("Click <a href=\"/LED8=ON\">LED 8 ON</a><br>");**

**client.println("Click <a href=\"/LED8=OFF\">LED 8 OFF</a><br>");**

**client.println("Click <a href=\"/LED=ON\">LED ALL ON</a><br>");**

**client.println("Click <a href=\"/LED=OFF\">LED ALL OFF</a><br>");**

**client.println("</html>");**

**delay(1);**

**}**

**55**

**Lampiran 2. Kode Pemrograman Aplikasi di Android**

**#Region Project Attributes**

**#ApplicationLabel: SKLW**

**#VersionCode: 1**

**#VersionName:**

**#SupportedOrientations: unspecified**

**#CanInstallToExternalStorage: False**

**#End Region**

**#Region Activity Attributes**

**#FullScreen: False**

**#IncludeTitle: True**

**#End Region**

**Sub Process\_Globals**

**Dim esp As HttpRequest**

**Dim droid As HttpClient**

**End Sub**

**Sub Globals**

**Private Label1 As Label**

**Private BtnONall As Button**

**Private BtnOFFall As Button**

**Private BtnExit As Button**

**Private Label12 As Label**

**Private BtnL8 As Button**

**Private BtnL1 As Button**

**Private BtnL2 As Button**

**Private BtnL3 As Button**

**Private BtnL4 As Button**

**Private BtnL5 As Button**

**Private BtnL6 As Button**

**Private BtnL7 As Button**

**End Sub**

**Sub Activity\_Create(FirstTime As Boolean)**

**Msgbox("Pastikan telah terhubung dengan modul Wifi", "peringatan") Activity.LoadLayout ("main") droid.Initialize("droid") DisableStrictMode**

**BtnL1.TextColor=Colors.Cyan BtnL2.TextColor=Colors.Cyan BtnL3.TextColor=Colors.Cyan BtnL4.TextColor=Colors.Cyan BtnL5.TextColor=Colors.Cyan BtnL6.TextColor=Colors.Cyan BtnL7.TextColor=Colors.Cyan BtnL8.TextColor=Colors.Cyan BtnExit.TextColor=Colors.Red BtnOFFall.TextColor=Colors.Green BtnONall.TextColor=Colors.Green**

**Dim ButEx As ColorDrawable**

**ButEx.Initialize(Colors.ARGB(128,0,**

**0,0),5dip)**

**BtnExit.Background=ButEx**

**BtnOFFall.Background=ButEx**

**BtnONall.Background=ButEx**

**Dim Tombol As ColorDrawable**

**Tombol.Initialize(Colors.ARGB(128,**

**50,50,0),5dip)**

**BtnL1.Background=Tombol**

**BtnL2.Background=Tombol**

**BtnL3.Background=Tombol**

**BtnL4.Background=Tombol**

**BtnL5.Background=Tombol**

**BtnL6.Background=Tombol**

**BtnL7.Background=Tombol**

**BtnL6.Background=Tombol**

**BtnL8.Background=Tombol**

**BtnExit.Background=Tombol**

**BtnOFFall.Background=Tombol**

**BtnONall.Background=Tombol**

**Label12.Visible=False**

**End Sub**

**Sub droid\_ResponseSuccess (Response As HttpResponse, TaskId As Int)**

**Label12.Text =**

**Response.GetString("UTF8")**

**End Sub**

**Sub droid\_ResponseError (Response As HttpResponse, Reason As String, StatusCode As Int, TaskId As Int)**

**Log("Error connecting: " & Reason &"**

**"& StatusCode)**

**If Response <> Null Then**

**Log(Response.GetString("UTF8"))**

**Response.Release**

**End If**

**End Sub**

**menangani error : dengan library JavaObject, dipanggil di Activity\_Create (firsttime....)**

**Sub DisableStrictMode**

**Dim jo As JavaObject**

**jo.InitializeStatic("android.os.Build.VERSIO**

**N")**

**If jo.GetField( "SDK\_INT") > 9 Then Dim policy As JavaObject**

**policy =**

**policy.InitializeNewInstance("android.os.Stri ctMode.ThreadPolicy.Builder", Null)**

**56**

**policy = policy.RunMethodJO("permitAll", Null).RunMethodJO("build", Null)**

**Dim sm As JavaObject**

**sm.InitializeStatic("android.os.StrictMode").Ru nMethod("setThreadPolicy", Array(policy))**

**End If**

**End Sub**

**Sub ON\_ALL**

**BtnL1.Text="L1 OFF"**

**BtnL2.Text="L2 OFF"**

**BtnL3.Text="L3 OFF"**

**BtnL4.Text="L4 OFF"**

**BtnL5.Text="L5 OFF"**

**BtnL6.Text="L6 OFF"**

**BtnL7.Text="L7 OFF"**

**BtnL8.Text="L8 OFF"**

**BtnL8.Color=Colors.ARGB(128,200,**

**0,0)**

**BtnL7.Color=Colors.ARGB(128,200,**

**0,0)**

**BtnL6.Color=Colors.ARGB(128,200,**

**0,0)**

**BtnL5.Color=Colors.ARGB(128,200,**

**0,0)**

**BtnL4.Color=Colors.ARGB(128,200,**

**0,0)**

**BtnL3.Color=Colors.ARGB(128,200,**

**0,0)**

**BtnL2.Color=Colors.ARGB(128,200,**

**0,0)**

**BtnL1.Color=Colors.ARGB(128,200,**

**0,0)**

**BtnONall.Enabled=False BtnONall.TextColor=Colors.DarkGray BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green ToastMessageShow("semua Lampu nyala","false")**

**End Sub**

**Sub OFF\_ALL**

**BtnL1.Text="L1 ON"**

**BtnL2.Text="L2 ON"**

**BtnL3.Text="L3 ON"**

**BtnL4.Text="L4 ON"**

**BtnL5.Text="L5 ON"**

**BtnL6.Text="L6 ON"**

**BtnL7.Text="L7 ON"**

**BtnL8.Text="L8 ON"**

**BtnL1.Color=Colors.ARGB(128,50,**

**50,0)**

**BtnL2.Color=Colors.ARGB(128,50,**

**50,0)**

**End**

**Sub**

**End**

**Sub**

**End**

**Sub**

**End**

**Sub**

**BtnL3.Color=Colors.ARGB(128,50,**

**50,0)**

**BtnL4.Color=Colors.ARGB(128,50,**

**50,0)**

**BtnL5.Color=Colors.ARGB(128,50,**

**50,0)**

**BtnL6.Color=Colors.ARGB(128,50,**

**50,0)**

**BtnL7.Color=Colors.ARGB(128,50,**

**50,0)**

**BtnL8.Color=Colors.ARGB(128,50,**

**50,0)**

**BtnOFFall.Enabled=False BtnOFFall.TextColor=Colors.DarkGray BtnONall.Enabled=True BtnONall.TextColor=Colors.Green ToastMessageShow("semua Lampu mati","false")**

**Sub**

**BtnONall\_Click**

**ON\_ALL**

**esp.InitializeGet("http://192.168.4.1/L**

**ED=ON")**

**Sub droid.Execute(esp,1)**

**BtnOFFall\_Click**

**OFF\_ALL**

**esp.InitializeGet("http://192.168.4.1/L**

**ED=OFF")**

**droid.Execute(esp,1)**

**Sub**

**BtnExit\_Click**

**Activity.Finish**

**Sub**

**BtnL8\_Click**

**If BtnL8.Text="L8 ON" Then**

**BtnL8.Text="L8 OFF"**

**0,0)BtnL8.Color=Colors.ARGB(128,200,**

**BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 8**

**nyala","false")**

**esp.InitializeGet("http://192.168.4.1/L**

**ED8=ON")**

**droid.Execute(esp,1)**

**Else If BtnL8.Text="L8 OFF" Then BtnL8.Text="L8 ON" BtnL8.Color=Colors.ARGB(128, 50,50,0)**

**BtnONall.Enabled=True BtnONall.TextColor=Colors.Gre en ToastMessageShow("Lampu 8 mati","false") esp.InitializeGet("http://192.168. 4.1/LED8=OFF") droid.Execute(esp,1)**

**57**

**End If**

**End Sub**

**Sub BtnL7\_Click**

**If BtnL7.Text="L7 ON" Then**

**BtnL7.Text="L7 OFF"**

**0,0)BtnL7.Color=Colors.ARGB(128,200,**

**BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 7**

**nyala","false")**

**esp.InitializeGet("http://192.168.4.1/L**

**ED7=ON")**

**droid.Execute(esp,1)**

**Else If BtnL7.Text="L7 OFF" Then**

**BtnL7.Text="L7 ON"**

**BtnL7.Color=Colors.ARGB(128,**

**50,50,0)**

**BtnONall.Enabled=True**

**BtnONall.TextColor=Colors.Gre**

**en**

**ToastMessageShow("Lampu 7**

**mati","false")**

**esp.InitializeGet("http://192.168.**

**4.1/LED7=OFF")**

**End If** **droid.Execute(esp,1)**

**End Sub**

**Sub BtnL6\_Click**

**If BtnL6.Text="L6 ON" Then**

**BtnL6.Text="L6 OFF"**

**0,0)BtnL6.Color=Colors.ARGB(128,200,**

**BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 6**

**nyala","false")**

**esp.InitializeGet("http://192.168.4.1/L**

**ED6=ON")**

**droid.Execute(esp,1)**

**Else If BtnL6.Text="L6 OFF" Then**

**BtnL6.Text="L6 ON"**

**BtnL6.Color=Colors.ARGB(128,**

**50,50,0)**

**BtnONall.Enabled=True**

**BtnONall.TextColor=Colors.Gre**

**en**

**ToastMessageShow("Lampu 6**

**mati","false")**

**esp.InitializeGet("http://192.168.**

**4.1/LED6=OFF")**

**End If** **droid.Execute(esp,1)**

**End Sub**

**Sub BtnL5\_Click**

**If BtnL5.Text="L5 ON" Then**

**BtnL5.Text="L5 OFF"**

**0,0)BtnL5.Color=Colors.ARGB(128,200,**

**BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 5**

**nyala","false")**

**esp.InitializeGet("http://192.168.4.1/L**

**ED5=ON")**

**droid.Execute(esp,1)**

**Else If BtnL5.Text="L5 OFF" Then**

**BtnL5.Text="L5 ON"**

**BtnL5.Color=Colors.ARGB(128,**

**50,50,0)**

**BtnONall.Enabled=True**

**BtnONall.TextColor=Colors.Gre**

**en**

**ToastMessageShow("Lampu 5**

**mati","false")**

**esp.InitializeGet("http://192.168.**

**4.1/LED5=OFF")**

**End If** **droid.Execute(esp,1)**

**End Sub**

**Sub BtnL4\_Click**

**If BtnL4.Text="L4 ON" Then**

**BtnL4.Text="L4 OFF"**

**0,0)BtnL4.Color=Colors.ARGB(128,200,**

**BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 4**

**nyala","false")**

**esp.InitializeGet("http://192.168.4.1/L**

**ED4=ON")**

**droid.Execute(esp,1)**

**Else If BtnL4.Text="L4 OFF" Then**

**BtnL4.Text="L4 ON"**

**BtnL4.Color=Colors.ARGB(128,**

**50,50,0)**

**BtnONall.Enabled=True**

**End If**

**End Sub**

**BtnONall.TextColor=Colors.Gre en ToastMessageShow("Lampu 4 mati","false") esp.InitializeGet("http://192.168. 4.1/LED4=OFF") droid.Execute(esp,1)**

**58**

**End If**

**End Sub**

**Sub BtnL3\_Click**

**If BtnL3.Text="L3 ON" Then**

**BtnL3.Text="L3 OFF"**

**0,0)BtnL3.Color=Colors.ARGB(128,200,**

**BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 3**

**nyala","false")**

**esp.InitializeGet("http://192.168.4.1/L**

**ED3=ON")**

**droid.Execute(esp,1)**

**Else If BtnL3.Text="L3 OFF" Then**

**BtnL3.Text="L3 ON"**

**BtnL3.Color=Colors.ARGB(128,**

**50,50,0)**

**BtnONall.Enabled=True**

**BtnONall.TextColor=Colors.Gre**

**en**

**ToastMessageShow("Lampu 3**

**mati","false")**

**esp.InitializeGet("http://192.168.**

**4.1/LED3=OFF")**

**End If** **droid.Execute(esp,1)**

**End Sub**

**Sub BtnL2\_Click**

**If BtnL2.Text="L2 ON" Then**

**BtnL2.Text="L2 OFF"**

**0,0)BtnL2.Color=Colors.ARGB(128,200,**

**BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green ToastMessageShow("Lampu 2**

**nyala","false")**

**esp.InitializeGet("http://192.168.4.1/L ED2=ON")**

**droid.Execute(esp,1)**

**Else If BtnL2.Text="L2 OFF" Then**

**BtnL2.Text="L2 ON" BtnL2.Color=Colors.ARGB(128, 50,50,0) BtnONall.Enabled=True BtnONall.TextColor=Colors.Gre en ToastMessageShow("Lampu 2 mati","false") esp.InitializeGet("http://192.168. 4.1/LED2=OFF") droid.Execute(esp,1)**

**Sub BtnL1\_Click**

**If BtnL1.Text="L1 ON" Then**

**BtnL1.Text="L1 OFF"**

**0,0)BtnL1.Color=Colors.ARGB(128,200,**

**BtnOFFall.Enabled=True BtnOFFall.TextColor=Colors.Green**

**ToastMessageShow("Lampu 1**

**nyala","false")**

**esp.InitializeGet("http://192.168.4.1/L**

**ED1=ON")**

**droid.Execute(esp,1)**

**Else If BtnL1.Text="L1 OFF" Then**

**BtnL1.Text="L1 ON"**

**BtnL1.Color=Colors.ARGB(128,**

**50,50,0)**

**BtnONall.Enabled=True**

**BtnONall.TextColor=Colors.Gre**

**en**

**ToastMessageShow("Lampu 1**

**mati","false")**

**esp.InitializeGet("http://192.168.**

**4.1/LED1=OFF")**

**End If** **droid.Execute(esp,1)**

**End Sub**

**Lampiran 3. Kode Pemrograman Arduino Uno**

|  |  |  |
| --- | --- | --- |
| **#include <SoftwareSerial.h>** | **Serial.begin(9600);** |  |
| **SoftwareSerial ESP8266(10, 11); // RX, TX** | **ESP8266.begin(115200);** |  |
| **char inChar;** | **pinMode(2, OUTPUT); digitalWrite(2, LOW);** |  |
| **void setup() {** |  |
| **pinMode(3, OUTPUT); digitalWrite(3, LOW);** |  |
| **// menseting komunikasi serial:** | **pinMode(4, OUTPUT); digitalWrite(4, LOW);** |  |

**59**

**pinMode(5, OUTPUT); digitalWrite(5, LOW); pinMode(6, OUTPUT); digitalWrite(6, LOW); pinMode(7, OUTPUT); digitalWrite(7, LOW); pinMode(8, OUTPUT); digitalWrite(8, LOW); } pinMode(9, OUTPUT); digitalWrite(9, LOW);**

**void loop() {**

**if (ESP8266.available()) {**

**inChar = ESP8266.read();**

**Serial.write(inChar);**

**//menterjemahkan data yang diterima if (inChar == '1' )**

**{ digitalWrite(2, HIGH); }**

**else if(inChar == 'a')**

**{ digitalWrite(2, LOW); }**

**if (inChar == '2' )**

**{ digitalWrite(3, HIGH); }**

**else if(inChar == 'b')**

**{ digitalWrite(3, LOW); }**

**if (inChar == '3' )**

**{ digitalWrite(4, HIGH); }**

**else if(inChar == 'c')**

**{ digitalWrite(4, LOW); }**

**if (inChar == '4' )**

**{ digitalWrite(5, HIGH); }**

**else if(inChar == 'd')**

**{ digitalWrite(5, LOW); }**

**if (inChar == '5' )**

**{ digitalWrite(6, HIGH); }**

**else if(inChar == 'e')**

**{ digitalWrite(6, LOW); }**

**if (inChar == '6' )**

**{ digitalWrite(7, HIGH); }**

**else if(inChar == 'f')**

**{ digitalWrite(7, LOW); } if (inChar == '7' )**

**{ digitalWrite(8, HIGH); }**

**else if(inChar == 'g')**

**{ digitalWrite(8, LOW); }**

**if (inChar == '8' )**

**{ digitalWrite(9, HIGH); }**

**else if(inChar == 'h')**

**{ digitalWrite(9, LOW); } if (inChar == '0' )**

**{ digitalWrite(2, HIGH);**

**digitalWrite(3, HIGH);**

**digitalWrite(4, HIGH);**

**digitalWrite(5, HIGH);**

**digitalWrite(6, HIGH);**

**digitalWrite(7, HIGH);**

**digitalWrite(8, HIGH);**

**digitalWrite(9, HIGH);}**

**else if(inChar == '9')**

**{ digitalWrite(2, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(5, LOW);**

**digitalWrite(6, LOW);**

**digitalWrite(7, LOW);**

**digitalWrite(8, LOW);**

**} digitalWrite(9, LOW);}**