BAB 2
LANDASAN TEORI

2.1 *Smart Home*

Home Automation dapat dikatakan sebagai ekstensi perumahan dari "otomatis bangunan/*building automation*". Hal ini berhubungan dengan kegiatan dan pekerjaan di rumah yang otomatis terlaksana. *Home Automation* mungkin termasuk *control* yang terpusat pada pencahayaan, HVAC (pemanas, ventilasi dan AC), peralatan, dan *system* lainnya, untuk memberikan kenyamanan, keamanan, efisiensi energi dan kemudahan yang lebih baik.

Sebuah *system* otomatisasi rumah mampu mengintegrasikanperangkat listrik di rumah dengan satu sama lainnya. Teknik-teknik yang digunakan dalam *Home Automation* termasuk yang diotomatisasi bangunan dengan pengendalian kegiatan domestic seperti *system* hiburan rumah, *houseplant* dan penyiraman halaman dan lainya. Perangkat dapat dihubungkan melalui jaringan Komputer untuk memungkinkan *control* dengan komputer pribadi dan memungkinkan akses *remote* dari internet. Melalui integrasi teknologi informasi dengan lingkungan rumah, sistem dan peralatan dapat berkomunikasi secara terpadu yang menghasilkan kenyamanan, efisiensi energi dan manfaat keselamatan [12]. *Smart Home* ini yang akan menjadi tema tugas akhir kali ini.

2.2 Aplikasi Mobile

Menurut Buyens, aplikasi mobile berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain.

Maka aplikasi *mobile* dapat di artikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah – pindah dari satu tempat ke tempat yang lain serta mempunyai ukuran yang kecil. Aplikasi *mobile* ini dapat di akses melalui perangkat nirkabel, pager, PDA, telepon seluler, *smartphone*, dan perangkat sejenisnya [13].

2.2.1 Karakteristik Perangkat Mobile

Perangkat mobile memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi mereka memiliki karakteristik yang sangat berbeda dari sistem desktop. Berikut karakteristik perangkat mobile, diantaranya :

a. Ukuran yang kecil

   Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

b. Memory yang terbatas

   Perangkat mobile juga memiliki memory yang kecil, yaitu *primary* (RAM) dan *secondary* (*disk*). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari memory, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.

c. Daya proses yang terbatas

   Sistem mobile tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti harddisk dan RAM, Anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

d. Mengkonsumsi daya yang rendah

   Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

e. Kuat dan dapat diandalkan

   Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

f. Konektivitas yang terbatas

   Perangkat mobile memiliki bandwith rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi wireless.

g. Masa hidup yang pendek

   Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah handphone, mereka booting dalam hitungan detik dan kebanyakan orang tidak mematikan handphone mereka bahkan ketika malam hari. PDA akan menyala jika anda menekan tombol power mereka [13].

2.3 Android

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar (*Smartphone*) dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti begerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile* dan *Nvidia* [14].

Beberapa pengertian lain dari *Android,* yaitu :

1. Merupakan *platform* terbuka (*Open Source*) bagi para programmer untuk membuat atau mengembangkan aplikasi.
2. Merupakan sistem operasi yang dibeli *Google Inc* dari *Android Inc*.
3. Bukan bahasa pemrograman, akan tetapi hanya menyediakan lingkungan hidup atau *runtime environment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk *device* dengan sistem memori yang kecil.

2.3.1 Sejarah Android

Pada Juli 2000, *Google* bekerjasama dengan *Android Inc*., perusahaan yang berada di Palo Alto, California, Amerika Serikat. Para pendiri *Android Inc*. bekerja pada *Google,* diantaranya Andy Rubin, Rich Miner, Nick Sears dan Chris White. Saat itu banyak yang menganggap fungsi *Android Inc.* hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa *Google* hendak memasuki pasar telepon seluler. Di perusahaan *Google,* tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh *Kernel Linux*. Hal ini menunjukan indikasi bahwa *Google* sedang bersiap menghadapi persaingan dalam pasara telepon seluler. Versi *android* terbaru yaitu versi 5.0. (*KitKat*). *Android* juga sudah bergabung dengan beberapa *smart mobile* seperti *LG, Samsung, Sony Ericsson, HTC, Motorola* dan lainnya.

Sekitar September 2007 sebuah studi melaporkan bahwa *Google* mengajukan hak paten aplikasi telepon seluler (akhirnya *Google* mengenalkan *Nexus One,* salah satu jenis telepon pintar GSM yang menggunakan *Android* pada sistem operasinya. Telepon seluler ini diproduksi oleh *HTC Corporotaion* dan tersedia di pasaran pada 5 Januari 2010).

Pada Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android ARM Holdings, Atheros Communications,* diproduksi oleh *Asustek Computer Inc. Garmin Ltd, Softbank, Sony Ericsson Toshiba Corp* dan *Vodafone Group Plc.* Seiring pembentukan *Open Handset Alliance,* OHA mengumumkan produk perdana mereka, *Android,* perangkat bergerak (*Mobile*) yang merupakan modifikasi *kernel Linux* 2.6. sejak *Android* dirilis telah dilakukan berbagai pembaruan berupa perbaikan *bug* dan penmbahan fitur baru [14].

2.3.2 Versi Android

Banyak *smartphone* dan PC Tablet menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versi, fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah HTC Dream yang

dirilis pada tanggal 22 Oktober 2008. Beberapa uraian versi *android* seperti dibawah ini :

1. **Android versi 1.1**

   Pada 9 Maret 2009, *Google* merilis *Android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan Email.

2. ***Android* versi 1.5 (*Cupcake*)**

   Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan *Android* dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset bluetooth*, animasi layar dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. ***Android* versi 1.6 (*Donut*)**

   Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, *camcorder* dan galeri yang diintegrasikan pada CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-spech engine*. Kemampuan dial kontak teknologi *text to change speech* (tidak tersedia pada semua ponsel).

4. ***Android* versi 2.0/2.1 (*Eclair*)**

   Pada 3 Desember 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah mengoptimalkan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML 5, daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, *digital zoom*, dan *Bluetooth* 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan

kompetisi aplikasi *mobile* terbaik (*killer apps*-aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset Android,* semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi *Android*. Aplikasi terkenal yang diubah ke dalam sistem operasi *Android* adalah Shazam, *Backgrounds* dan *WeatherBug*. Sistem operasi *Android* dalam situs internet juga dianggap penting untuk menciptakan aplikasi *Android* asli, contohnya oleh *MySpace* dan *Facebook*.

5. ***Android* versi 2.2 (*Froyo* : *Frozen Yoghurt*)**

   Pada 20 Mei 2010, *Android* versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi dengan V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel dan kemampuan *auto update* dalam aplikasi Android Market.

6. ***Android* versi 2.3 (*Gingerbread*)**

   Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antarmuka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM. efek audio baru (*reverb, equalization, headphone virtualization,* dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC) dan dukungan jumlah kamera yang lebih dari satu.

7. ***Android* versi 3.1 (*Honeycomb*)**

   Android Honeycomb dirancang khusus untuk tablet. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga

mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet peratam yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom.

8. ***Android* versi 4.0 (*Ice Cream Sandwich*)**

   Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline* dan berbagi informasi dengan menggunakan NFC.

9. ***Android* versi 4.1 (*Jelly Bean*)**

   Pada tanggal 27 Juni 2012 Google mengumumkan *Android* versi 4.1 (*Jelly Bean*) dalam konferensi Google I/O berdasarkan kernel Linux 3.0.31. *Jelly Bean* adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuak pengguna (*User Interface*). Pembaruan ini diwujudkan dalam proyek *Butter*, perbaikan ini termasuk antisipasi sentuh, *triple buffering*, perpanjangan waktu vsync dan peningkatan *frame rate* hingga 60fps untuk menciptakan UI yang lebih halus. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

10. ***Android* versi 4.4 (KitKat)**

    Android versi 4.4 (KitKat) direncanakan akan dirilis pada bulan Oktober 2013. Setelah sebelumnya beredar rumor bahwa Android versi berikutnya setelah *Jelly Bean* diperkirakan akan diberi nomor 5.0 dan dinamai *Key Lime Pe* [14].

2.3.3 Komponen Aplikasi Android

Fitur penting Android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur *scroller* dan aplikasi lain telah mengembangkan fitur *scroller* yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup

menggunakan *scroller* yang telah ada. Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya Android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi main() pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan [14].

2.3.3.1 *Activities*

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan.

Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity* ditentukan oleh posisinya dalam tumpukan *acitivity*, LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* Android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut [14].

Ketika *activity* diambil dan disimpan dalam tumpukkan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi :

1. ***Active***

Setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukkan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.

2. ***Paused***

   Dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused*. Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak dapat menerima masukkan dari pengguna. Dalam kasus ekstrim, Android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.

3. ***Stopped***

   Ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped*. *Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive*.

4. ***Inactive***

   Kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah ditiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali. Kondisi transisi ini sepenuhnya ditangani oleh manajer memori Android. Android akan memulai menutup aplikasi yang mengandung *activity inactive*, kemudian *stopped activity*, dan dalam kass luar biasa *paused activity* juga akan ditutup [14].

2.3.3.2 *Services*

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak [14].

2.3.3.3 *Intents*

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman *web*, dan lain sebagainya. *Intents* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. *Intents* dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin menampilkan *URL*, sistem akan menentukan komponen apa yang dibutuhkan oleh *Intents* tersebut [14].

2.3.3.4 *Broadcast Receiver*

Broadcast Receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *Broadcast* berasal dari sistem misalnya, Batre sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa *default* pada perangkat. Sama halnya dengan *service*, *Broadcast Receivers* tidak menampilkan antarmuka pengguna. Namun, *Broadcast Receivers* dapat menggunakan *Notification Manager* untuk memberitahukan sesuatu kepada pengguna [14].

2.3.3.5 *Content Providers*

Content Providers digunakan untuk mengelola dan berbagi *database*. Data dapat disimpan dalam file sistem, dalam database *SQLite*, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *Content Provider* memungkinkan antar

aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya [14].

2.3.4 Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada Android [14] :

1. ***Foreground Activity***

   Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

2. ***Background Service***

   Aplikasi yang memiliki interaksi terbatas dengan user, selain dari pengaturan konfigurasi, semua dari prosesnya tidak tidak tampak pada layar. Contohnya aplikasi penyaringan panggilan atau sms *auto respon*.

3. ***Intermittent Activity***

   Aplikasi yang masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di *background* dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik. Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

2.3.5 Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk didalamnya *Symbian*, *iPhone*, *Windows Mobile*, *BlackBerry*, *Java Mobile Edition*, *Linux Mobile* (LiM), dan banyak lagi. Namun ada beberapa hal yang menjadi

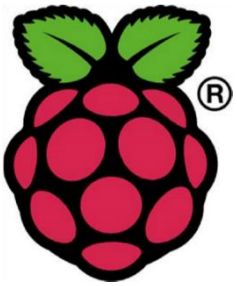
kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada *platform* lain, Android adalah yang pertama menggabungkan hal seperti berikut [14] :

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan *Linux* dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royality*. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dan tanpa terikat oleh *vendor* manapun.
2. Arsitektur komponen dasar Android terinspirasi dari teknologi *internet Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, *database* SQL, *browser* dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suara terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh Flash menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun *game* yang berbeda.

Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin *virtual Dalvik*, sehingga kode program *portable* antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi lyr semua dapat disesuaikan dengan program [14]. Pada tugas akhir ini aplikasi mobile android yang akan digunakan untuk pengontrol rumah pintar.

2.4 Raspberry Pi

Raspberry Pi merupakan sebuah komputer yang berukuran tidak lebih besar dari sebuah kartu kredit. *Processor* yang menjadi jantung dari Raspberry Pi adalah *Broadcom* BCM2835 *system-on-chip* (SoC) multimedia *processor*, yang mampu memproses grafik atau gambar, suara dan komunikasi *hardware*, dibangun ke dalam satu komponen yang terintergrasi dengan chip memori 256 MB [15]. Raspberry Pi mempunyai logo di pasaran sebagai berikut:



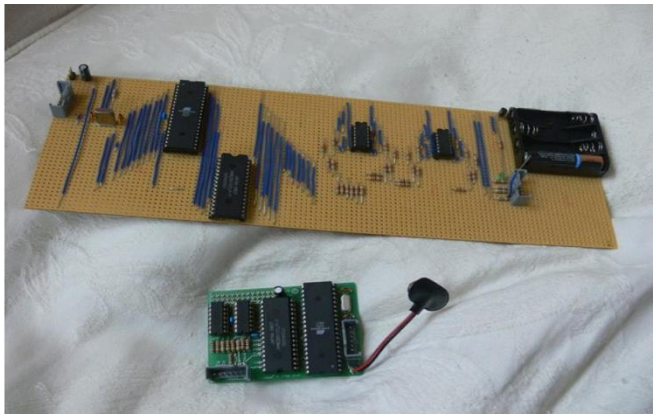
Sumber Gambar : Traynor, B. [15]

Gambar 2. 1 Logo Raspberry Pi

2.4.1 Sejarah Raspberry Pi

Ide di balik sebuah komputer kecil dan murah untuk anak-anak muncul pada 2006, ketika Eben Upton dan rekan-rekannya di *University of Laboratorium Computer Cambridge*, termasuk Rob Mullins, Jack Lang dan Alan Mycroft, khawatir tentang penurunan tahun-ke-tahun tingkat keterampilan siswa Tingkat A yang menerapkan Ilmu Komputer pada setiap tahun akademik. Pada 1990-an di mana sebagian besar anak-anak menerapkan hobi mereka sebagai programmer, tetapi pemandangan di tahun 2000-an sangat berbeda; kebanyakan dari mereka hanya mempunyai keterampilan mendesain sebuah web [15].

Dari 2006 hingga 2008, Eben merancang beberapa versi dari apa yang sekarang telah menjadi Raspberry Pi.



Sumber Gambar : Foundation, R. P. [15]

Gambar 2. 2 Raspberry Pi Edisi Tahun 2006

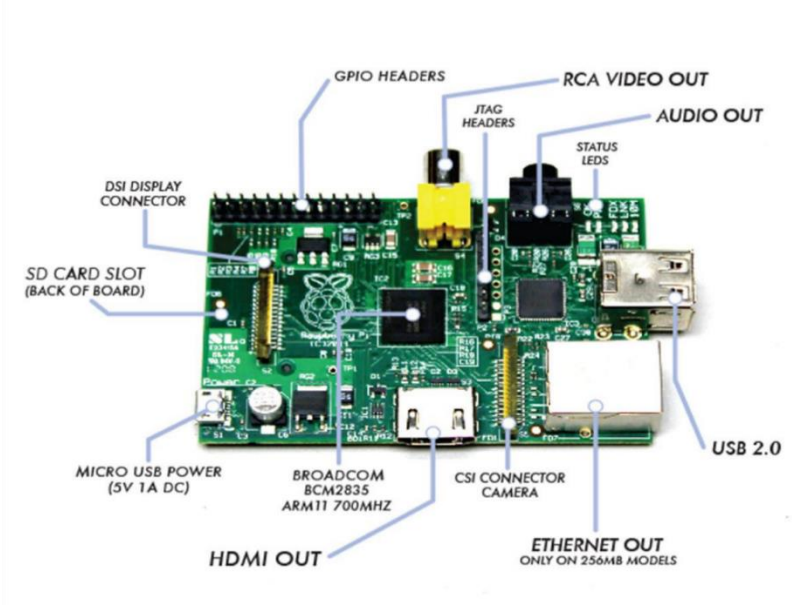
Board tersebut menggunakan ATmega644 dengan *clock* 22,1MHz dan 512K SRAM untuk data dan penyimpanan *framebuffer*. 19 dari 32 baris GPIO Atmel yang digunakan untuk menggerakkan bus alamat SRAM. Untuk menghasilkan sinyal video 320 × 240, Atmel akan menambahkan alamat, dan jalur data diberi *supply* melalui *buffer* IC seri-74HC *trio summing-point* DAC sederhana; selama *blanking* horizontal dan vertikal, ia bebas untuk melakukan operasi lain.

Pada tahun 2008, prosesor dirancang untuk perangkat *mobile* yang menjadi lebih terjangkau, dan cukup kuat untuk mendukung multimedia yang sangat baik, mereka merasa dengan fitur tersebut akan membuat anak-anak tertarik untuk belajar pemrograman pada perangkat murni pemrograman berorientasi. Proyek ini mulai terlihat sangat realistis. Eben (sekarang seorang arsitek chip pada *Broadcom*), Rob, Jack dan Alan, bekerja sama dengan Pete Lomas, MD dari desain *hardware* dan pembuatan perusahaan *Norcott Technologies*, dan David Braben, co-penulis BBC *Mikro Game Elite*, untuk membentuk Raspberry Pi *Foundation*.

Tiga tahun kemudian, mereka memulai produksi massal melalui penawaran pembuatan berlisensi dengan Elemen 14/*Premier Farnell* dan RS Elektronika [15].

2.4.2 Spesifikasi dan Hardware

Raspberry Pi memiliki 14 komponen utama. Diantaranya 2 port USB 2.0, *Ethernet*, HDMI, slot SD Card, DSI *display Connector*, Micro USB Power (5V 1A DC), *Broadcom* BCM2835 Arm11, CSI *Connector Camera*, CSI JTAG *Header*, RCA *Video Out*, *Led Status*, *Ethernet Controller* dan 26 *General Purpose Input Output* (GPIO).



Sumber Gambar : Traynor, B. [16]

Gambar 2. 3 Konfigurasi Hardware Raspberry Pi Model B

Berikut merupakan tabel penjelasan dari Gambar 2.3 :

Tabel 2. 1 Penjelasan Komponen Raspberry Pi [16]

No	Komponen	Penjelasan
1	Arm 11 76JZF 700 Mhz	Merupaan Inti atau otak dari Raspberry Pi.
2	HDMI *Out*	HDMI 1.3a digunakan untuk tampilan yang menggunakan sinyal HDMI.
3	CSI c*onnector camera*	Kamera serial interface dengan 15 pin flat flex kabel header.

No	Komponen	Penjelasan
4	*Ethernet Out*	Mendukung fungsi Wake on-LAN dan TCP/IP.
5	USB 2.0	2 Port USB, fungsi USB disediakan oleh SMSC LAN9512 yang baik untuk menghemat ruang PCB.
6	Led Indikator	4 Led indicator, ACT warna hijau untuk status SD *Card*. PWR warna merah artinya Raspberry Pi menerima daya listrik. FDX warna jingga ketika koneksi jaringan adalah *full duplex*. LNK warna jingga artinya ketika *Ethernet* terhubung dengan jaringan. LED100 ketika koneksi yang berjalan adalah 100Mbps.
7	Audio *Output*	Sebagai *stereo audio output.*
8	JTAG *Header*	Digunakan untuk memprogram SoC dan chip SMSC didalam board.
9	RCA *Video Output*	Sebagai *video output* pada Raspberry Pi.
10	GPIO Header	Terdiri dari 26 pin yang digunakan untuk pengontrolan suatu perangkat. Untuk pengontrolan itu sendiri dapat dilakukan menggunakan suatu aplikasi. Beberapa fitur dari pin tersebut adalah RX, TX, PWM, DCin 5V dan 3.3V.
11	SD *Card Slot*	Sebagai slot untuk SD *Card* yang terlah terisi sistem operasi agar dapat melakukan booting.
12	DSI *Display connector*	Digunakan untuk Display LCD seperti LCD pada ponsel.
13	Micro USB power	Power input dengan 5V1A untuk memenuhi kebutuhan tegangan dan arus pada Raspberry

Raspberry Pi dibuat dalam dua varian, Model A dan Model B. Berikut merupakan spesifikasi dari Raspberry Pi:



Gambar 2. 4 Raspberry Pi Type A



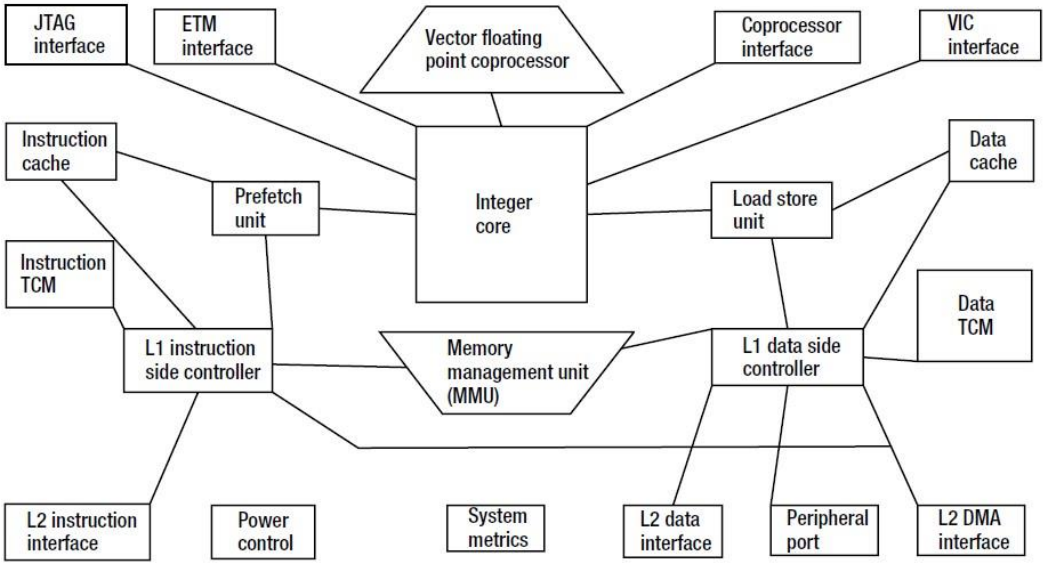
Gambar 2. 5 Raspberry Pi Type B

Tabel 2. 2 Spesifikasi *Raspberry Pi* Model A dan Model B [15]

Spesifikasi	Model A	Model B
System-on-Chip (SoC)	Broadcom BCM2835 (CP U + GPU. SDRAM is a separate chip stacked on to p)	
CPU	700 MHz ARM11 ARM11 76JZF-S core	
GPU	Broadcom VideoCore IV, OpenGL ES 2.0,OpenVG 1080p30 H.264 high-profile encode/decode	
Memori (SDRAM)	256 MiB (planned with 128 MiB, upgraded to 256 MiB on 29 Feb 2012)	256 MiB (until 15 Oct 2012); 512 MiB (since 15 Oct 2012)
USB 2.0	1 (provided by the BCM2835)	2 (via integrated USB hub)
Video *output*s	Composite video | Comp osite RCA, HDMI (not at the same time)	
Audio *output*s	TRS connector | 3.5 mm ja ck, HDMI	
Audio *input*s	none, but a USB mic or so und-card could be added	
Onboard Storage	Secure Digital | SD / MM C / SDIO card slot	
Onboard Networks	None	10/100 wired Ethernet RJ45
Low-level peripherals	*General Purpose Input O utput* (GPIO) pins, *Serial Peripheral Interface Bus* (SPI), I²C, I²S, *Universal asynchronous receiver/tra nsmitter* (UART)	
Real-time clock	None	
Power ratings	300 mA, (1.5 W)	
Power source	5 V (DC) via Micro USB t ype B or GPIO header	
Size	85.0 x 56.0 mm x 15mm	85.0 x 56.0 mm x 17mm

2.4.3 Arsitektur Raspberry Pi

Raspberry Pi menggunakan sistem operasi berbasis *kernel Linux.Raspbian* merupakan Sistem operasi berbasis *Debian* yang dapat bebas dioptimalkan untuk perangkat keras *Raspberry Pi*, yang dirilis pada bulan Juli 2012 [18].



Sumber Gambar : Horan, B. [18]

Gambar 2. 6 Diagram Blok Arsitektur Raspberry Pi

GPU *hardware* diakses melalui gambar firmware yang di-load ke GPU saat boot dari SD-card. Gambar *firmware* dikenal sebagai kumpulan biner, sementara *driver Linux* yang terkait adalah sumber tertutup (*closed source*). Aplikasi perangkat lunak menggunakan panggilan ke sumber tertutup *run-time library* yang pada gilirannya menjadi panggilan *open source driver* dalam *Linux kernel*. API *driver kernel* spesifik untuk perpustakaan tersebut bersifat tertutup. Aplikasi video menggunakan *OpenMAX*, aplikasi 3D menggunakan *OpenGL ES* dan 2D aplikasi menggunakan *OpenVG* yang pada nantinya menggunakan EGL. *OpenMAX* dan EGL menggunakan *open source kernel driver*.

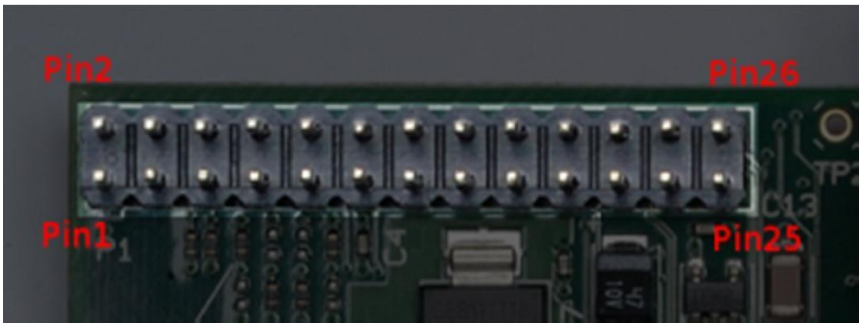
Pada 19 Februari 2012, Yayasan *Raspberry Pi* merilis bukti konsep kartu SD *image* yang dapat dimuat ke *SD Card* untuk menghasilkan sebuah sistem operasi yang pertama. *Image* didasarkan pada *Debian 6.0* dengan LXDE *desktop* dan *Midori browser*, ditambah berbagai alat pemrograman. *image* tersebut berjalan pada QEMU yang memungkinkan *Raspberry Pi* akan ditiru pada berbagai platform lainnya [18]. Raspberry pi ini digunakan untuk sebagai otak sistem rumah pintar yang untuk mengontrol semua perangkat yang ada pada rumah pintar.

2.4.4 *General Purpose Input Output* (GPIO)

Selain USB, Ethernet dan port HDMI, Raspberry Pi menawarkan antarmuka tingkat rendah dimaksudkan untuk menghubungkan lebih langsung dengan chip dan modul subsistem. GPIO (*General Purpose Input Output*) ini mempunyai 2x13 pin header termasuk SPI, I2C, UART serial, 3V3 dan daya 5

Antarmuka ini tidak "*plug and play*" dan memerlukan driver atau library tersendiri untuk menghindari miss-wiring. Pin menggunakan tingkat logika 3V3 dan tidak toleran terhadap tingkat 5V.

General Purpose Input Output (GPIO) adalah pin pada sebuah chip yang dapat dikontrol oleh *software*.



Gambar 2. 7 Bentuk pin *header* GPIO Raspberry Pi

General Purpose Input Output (GPIO) menyediakan 8 pin GPIO dan ditambah akses ke I²C, SPI, UART, serta 3,3 V, +5 V dan GND. Pin satu adalah pin di kolom pertama dan di baris bawah.



Gambar 2. 8 Layout pin GPIO Raspberry Pi

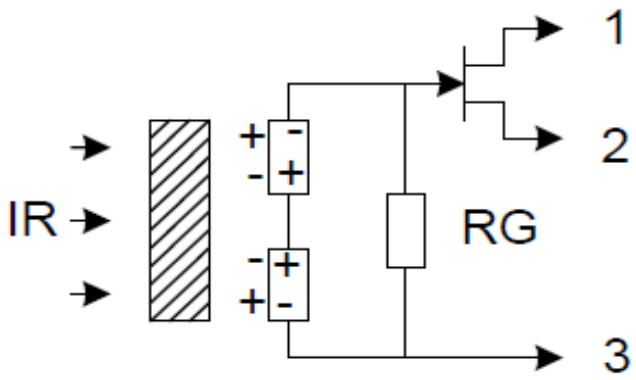
Semua pin GPIO dapat dikonfigurasi ulang untuk menyediakan fungsi alternatif, SPI, PWM, I²C dan sebagainya. Tetapi pada saat konfigurasi ulang hanya pin GPIO 14 & 15 ditugaskan untuk fungsi UART alternatif, kedua dapat beralih kembali ke GPIO untuk memberikan total 17 GPIO pin.

2.5 Sensor PIR (*Passive Infra Red*)

Sensor adalah komponen yang mengubah besaran fisis menjadi besaran listrik Franky Chandra dan Deni Arifianto [19]. Sensor yang digunakan pada sistem ini adalah Sensor PIR.

PIR merupakan sebuah sensor berbasis infrared. Akan tetapi, tidak seperti sensor infrared kebanyakan yang terdiri dari IR LED dan fototransistor. PIR tidak memancarkan apapun seperti IR LED. Sesuai namanya "Passive", sensor ini hanya merespon energi dari pancaran sinar inframerah pasif yang dimiliki oleh setiap benda yang terdeteksi olehnya. Benda yang dapat dideteksi oleh sensor ini biasanya adalah tubuh manusia[19].

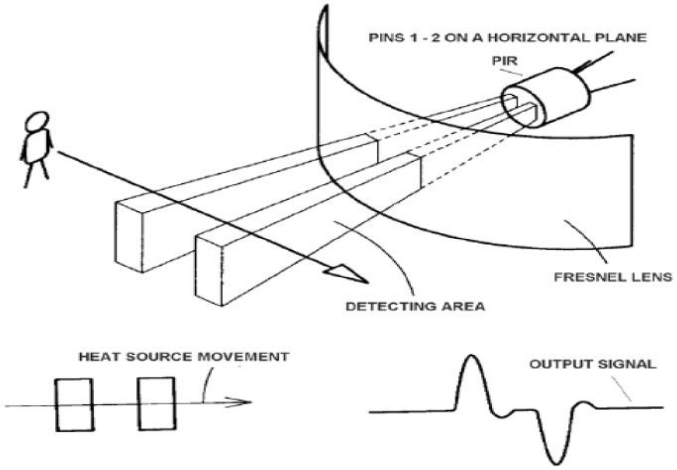
Menurut Mohd. Syaryadhi etal [19], PIR sensor mempunyai dua elemen sensing yang terhubungkan dengan masukan dengan susunan seperti yang terdapat dalam Gambar berikut:



Sumber Gambar : Mohd. Syaryadhi[19]

Gambar 2. 9 Diagram Internal Rangkaian sensor PIR

Jika ada sumber panas yang lewat di depan sensor tersebut, maka sensor akan mengaktifkan sel pertama dan sel kedua sehingga akan menghasilkan bentuk gelombang seperti ditunjukkan dalam Gambar 2.2 Sinyal yang dihasilkan sensor PIR mempunyai frekuensi yang rendah yaitu 0,2 – 5 Hz.



Sumber Gambar : Mohd. Syaryadhi[19]

Gambar 2. 10 Arah Jangkauan Gelombang Sensor PIR (*Passisve Infra Red*)

Benda yang dapat memancarkan panas berarti memancarkan radiasi infra merah. Benda – benda ini termasuk makhluk hidup seperti binatang dan tubuh manusia. Tubuh manusia dan binatang dapat memancarkan radiasi infra merah terkuat yaitu pada panjang gelombang 9,4 μm. Radiasi infra merah yang dipancarkan inilah yang menjadi sumber pendeteksian bagi detektor panas yang memanfaatkan radiasi infra merah. (Mohd. Syaryadhi etal., 2007) [19].

Deni Arifianto, (2011) menyebutkan modul sensor PIR memiliki karakterisasi sebagai berikut :

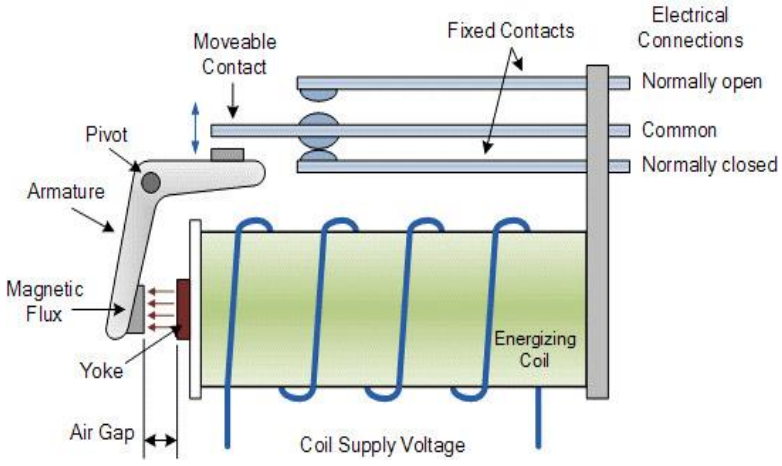
- Tegangan Catu Daya : 4.7 – 12 VDC
- Jangkauan Deteksi Sensor : 5 meter pada sudut 0 derajat
- Output sensor tegangan High : 5 VDC
- Output lebar pulsa : 0.5 s

Sensor PIR digunakan untuk mendeteksi gerak (manusia) yang berfungsi sebagai keamanan rumah.

2.6 Relay

Prinsip Kerja Relay - Relay adalah Saklar *(Switch)* yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/*Switch*). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (low power) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan Relay yang menggunakan Elektromagnet 5V dan 50 mA mampu menggerakan Armature Relay (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.

2.6.1 Bagian-bagian Relay Elektromekanik



Gambar 2. 11 Struktur Sederhana Relay

Keterangan :

1. Bingkai : biasanya bingkai dipakai pada relay, dan bingkai ini biasanya terbuat dari bahan yang cukup kuat untuk melindungi bagian relay.
2. Lilitan : lilitan ini terdiri dari kawat yang digulung dan biasanya terdapat pada bagian relay, biasanya lilitan ini terdiri dari dua atau lebih kawat atau logam

yang bisa mengantarkan listrik dengan cukup baik, ketika lilitan ini di aliri arus listrik maka akan menghasilkan medan elektromagnetik.

3. Kontak : bagian pada relay yang berfungsi sebagai komponen yang bisa membuat kondisi tersambung atau terbka pada relay, atau istilah lainnya adalah bagain relay yang bisa membuat kondisi relay terbuka atau tertutup.
4. Armature/angker : adalah bagian relay yang bisa bergerak, yang biasanya bagian ini bisa membuka dan menutup kontak, dan biasanya armature tertarik oleh kawat yang menyangganya.

2.6.2 Jenis- jenis Relay

Relay mempunyai beberapa jenis yang umum ditemui disekitar kita, seperti berikut ini pengertian relay berdasarkan jenisnya:

1. ***Subminiature*** **Relay**

Subminiature relay adalah jenis relay mekanik yang dirancang untuk saklar pada rangkaian elektronika yang mempunyai arus besar, komponen ini terdiri dari lilitan untuk *alternate current* (AC) atau *direct current* (DC), relay DC biasanya digerakan dengan tegangan yang sudah ditetapkan deperti 6, 12 atau 24 volt DC dengan hambaan lilitan masing-masing sekitar 40, 160 dan 650 ohm, sedangkan untuk relay AC dengan hambatan lilitan masing-masing 3400 dan 13600 ohm, kecepatan switchingnya berada pada kecepatan 10 sampai dengan 100 ms (*mocro second*) dengan rating arus kisaran 2-15 Ampere.

2. ***Miniatur*** **Relays**

Pengertian relay ini hampir mirip dengan *subminiature* relay tetapi relay jenis ini dirancang untuk mempunyai sensitivitas lebih besar dari *subminiature* relay serta untuk arus yang lebih kecil, biasanya jenis relay ini khusus dirancang untuk tegangan direct current (DC) serta digerakan oleh tegangan DC tetapi bisa juga didesain untuk saklar pada arus *alternate current* (AC), biasanya komponen ini bisa bekerja pada tegangan 5,6,12 dan 25 volt DC dengan hambatan lilitan dari 50 sampai 3000 ohm.

3. Relay *Reed*

Pengertian relay ini serta bentuk relay ini terdiri dari dua strip logam tipis dimana berfungsi sebagai kontak yang bisa bergerak, relay *reed* ini biasanya ditempatkan pada wadah kaca atau transparan dan dikelilingi oleh kumparan magnet.

4. *Solid-State* Relays

Relay ini dari bahan semikonduktor. Relay *solid-state*, termasuk transistor (FET, BJTs) dan *thyristor* (SCR, triac, diacs, dll). Relay *solid-state* memiliki kecepata *switching* yang fenmenal atau cukup tinggi. Namun, perangkat ini biasanya memiliki hambatan yang tinggi dan memerlukan pengaturan yan lebih rumit, dan relay jenis ini kurang kuat bila terjadi *overload* dibandingkan dengan relay elektromekanik.

Relay digunakan untuk menggerakan kontak saklar pada lampu yang ada di rumah pintar.

2.7 ACS712

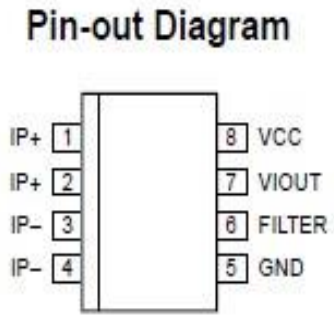
Pengukuran arus biasanya membutuhkan resistor *shunt* yaitu resistor yang dihubungkan secara seri pada beban dan megubah arus menjadi tegangan. Tegangan trsebut biasanya di umpankan ke *current transformer* terlebih dahulu sebelum masuk kerangkaian pengkondisi signal. Teknologi *hall effect* yang diterapkan oleh perusahaan *Allegro* menggantikan *resistor shunt* dan *current transformer* menjadi sebuah sensor yang ukuran yang relatif jauh lebih kecil [20].



Sumber Gambar : Honey Well [20]

Gambar 2. 12 Sensor Arus ACS712-30A

Istilah *Hall Effect* dikenal setelah Edwin H. Hall (1855-1938) menemukan bahwa jika arus listrik mengalir melalui penghantar yang ditempatkan pada *tranverse* medan magnet yang kuat, akan menghasilkan beda potensial yang melewati penghantar pada kedua sudut penghantar itu. *Hall Effect Sensor* adalah suatu transduser yang dapat mengubah besaran medan magnet menjadi besaran listrik yaitu berupa tegangan. Sensor *Hall effect* digunakan untuk mendeteksi kedekatan, keberadaan atau ketiadaan medan magnet dari suatu objek dengan kritis. Sensor *Hall effect* digunakan untuk sensor perpindahan, sensor letak atau jarak, sensor kecepatan dan sensor arus [20].



Sumber Gambar : HoneyWell[20]

Gambar 2. 13 Pin Sensor Arus ACS712-30A.

Tabel 2. 3 Tabel Deskripsi Pin Sensor Arus ACS712-30A

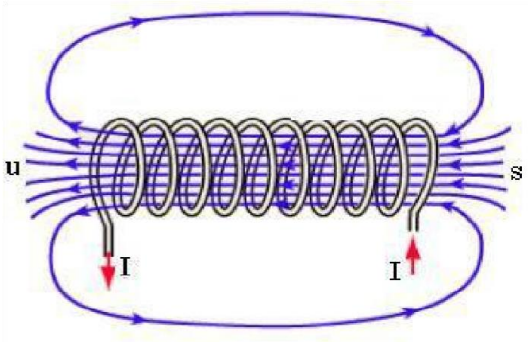
Number	Name	Description
1 and 2	IP+	*Terminals for current being sensed; fused internally*
3 and 4	IP-	*Terminals for current being sensed; fused internally*
5	GND	*Signal ground terminal*
6	FILTER	*Terminal for external capacitor that sets bandwidth*
7	VIOUT	*Analog output signal*
8	VCC	*Device power supply terminal*

Sensor arus listrik ACS712 pada rumah pintar ini digunakan untuk pengukur arus lisrik yang digunakan pada lampu di rumah pintar.

2.8 Solenoid

Solenoid adalah peralatan yang dipakai untuk mengkonversi sinyal elektrik atau arus listrik menjadi gerak mekanik. Solenoid dibuat dari kumparan dan inti besi yang dapat digerakkan yang berfungsi sebagai aktuator untuk membuka kunci otomatis pada brankas.

Secara skematik bentuk dari solenoid dapat dilihat pada Gambar 2.14, dimana solenoid terdiri dari *n* buah lilitan kawat berarus listrik I, medan magnet yang dihasilkan memiliki arah seperti pada gambar dibawah, dimana kutub utara magnet mengikuti aturan tangan kanan.



Gambar 2. 14 Solenoid Dengan Banyak Lilitan *n*

Kuat medan magnet pada solenoid dengan jumlah lilitan persatuan panjang *n* adalah dapat dihitung dengan menggunakan persamaan dibawah ini.

$B = \mu 0 \, . \, n \, . \, I$ …… (2.1)

Dimana : $n = \frac{N}{l}$

Keterangan : B = Medan Magnet

μ0 = Konstanta Permeabilitas Udara

n = Jumlah Lilitan

Persatuan Panjang = N

I = Arus

Prinsip kerja dari Solenoid berdasarkan pada penghantar yang membawa arus kedalam kumparan sehingga kumparan akan menimbulkan medan magnet. Medan magnet ini dibuat sedemikian rupa sehingga keadaannya selalu tolak menolak antara medan magnet. Solenoid digunakan untuk kunci otomatis yang dipasang di rumah pintar.



Gambar 2. 15 Bentuk Fisik Solenoid

2.9 Buzzer

Buzzer adalah suatu alat yang dapat mengubah sinyal listrik menjadi sinyal suara. Pada umumnya buzzer digunakan untuk alarm, karena penggunaannya yang cukup mudah yaitu dengan memberikan tegangan *input* maka buzzer akan mengeluarkan bunyi. Frekuensi suara yang dikeluarkan oleh buzzer yaitu antara 1 sampai 5 kHz. Pada tugas akhir ini menggunakan buzzer sebagai keamanan pada rumah pintar yang akan mengeluarkan peringatan seperti getaran suara atau alarm.



Gambar 2. 16 Contoh Buzzer

2.10 Ethernet

Ethernet adalah teknologi jaringan komputer bersadarkan pada kerangka jaringan area lokal (LAN). Sistem komuikasi melalui Ethernet membagi aliran data ke dalam paket individual yang disebut frame. Setiap frame, berisi alamat sumber dan tujuan serta pengecekan error data sehingga data yang rusak dapat dideteksi dan dikirim kembali. Ethernet adalah protokol LAN yang memungkinkan setiap PC "berlomba" untuk mengakses network. Sekarang Ethernet menjadi protokol LaN yang paling populer karena relatif murah dan mudah di-install serta ditangani.

Dalam sejarhnya, Ethernet dibuat oleh Xerox di tahun 1978. Ethernet telah distujui sebagai standar industri protokol LAN tahun 1983. Sebuah *network* yang menggunakan Ethernet sebagai protokol sering disebut Ethernet *network.*

2.10.1 Fungsi Ethernet

Ethernet berfungsi membantu pertukaran file dan data melalui jaringan komputer. Komputer-komputer ini berkomunikasi melalui jaringan komputer dengan bantuan dari akses fisik ke media jaringan dan sistem pengalamana tingkat rendah melalui penggunaan alamat MAC (nomor seri unik 48-bit yang disimpan dalam ROM yang dilakukan pada Ethernet). Dalam sebuah jaringan, setiap komputer memiliki kartu dengan alamat MAC yang unik.

2.10.2 Jenis-jenis Ethernet

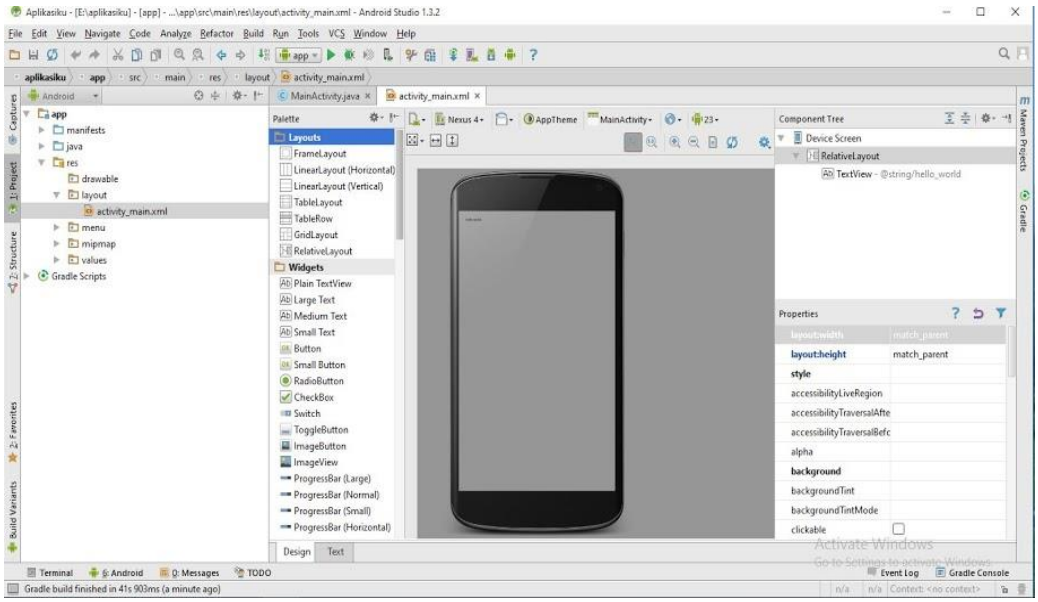
Dilihat dari kecepatannya, Ethernet terbagi menjadi empat jenis, antara lain:

1. 10 Mbit/detik, yang sering disebut sebagai Ethernet saja (standar yang digunakna: 10 Base2, 10Base5, 10BaseT, 10BaseF).
2. 100Mbit/detik, yang sering disebut sebagai Fast Ethernet (standar yang digunakna: 100 BaseFX, 100BaseT, 100BaseT4, 100BaseTX).
3. 1000 Mbit/detik atau 1 Gbit/detik, yang sering disebut sebagai Gigabit Ethernet (standar yang digunakna: 1000BaseCX, 1000BaseLX, 1000BaseSX, 1000BaseT).
4. 10000 Mbit/detik atau 10Gbit/detik. Standar ini belum banyak diimplementasikan.

2.11 Android Studio

Android studio adalah lingkungan pengembangan terpadu (*IDE)* resmi untuk pengembangan platform android, hal itu disampaikan pada tanggal 16 mei 2013 Google I/O,android studio sudah tersedia secara bebas dibawah lisensi Apache 2.0.

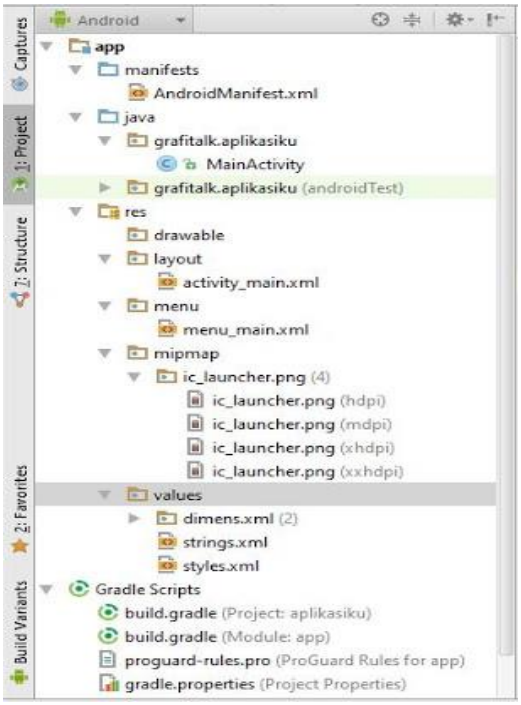
Android studio pada awalnya tahap *preview* versi 0.1 yang dipakai pada tanggal 1 mei 2013 dan memasuki tahap beta pada bulan juni 2014 dan mulai stabil dirilis pada Desember 2014 dengan versi 1.0 ,berdasarkan *jetBrains* ‘IDEA *Intellij Software*, android studio dirancang khusus untuk pengembangan android yang tersedia untuk Windows,Mac OS X dan Linux sebagai pengganti *Eclipse* [21].



Sumber Gambar : grafitalk.com [21]

Gambar 2. 17 *Layout* Android Studio

2.11.1 Struktur Android Studio



Gambar 2. 18 Struktur Android Studio

Berikut rincian dari struktur Android Studio :

a. App : berfungsi untuk menyimpan file file pendukung aplikasi yang akan diolah.
   1. *Manifest*
      tempat dimana konfigurasi konfigurasi pendukung project serta izin (permision) untuk mengakses hardware.
   2. Java
      tempat berdiamnya *code/scripts* java untuk menjalankan aplikasi yang akan dibuat
   3. *Res* :
      - *Drawable* : tempat untuk menyimpan gambar atau *icon* pendukung aplikasi.
      - *Layout* : tempat berkumpulnya design tampilan yang akan tampak pada tampilan aplikasi anda.
      - *Menu* : berisi *code* berupa *file* xml yang didalamnya berisi *array* tentang menu yang akan tampil dalam *project* anda, misalkan *file, new,open,close,*dsb.
      - *Mipmap* : tempat berkumpulnya *icon launcher* (icon utama aplikasi) yang ukurannya menyesuaikan *device* yang digunakan.
      - *Values* : untuk menyimpan file .xml yang menggambarkan sumber dari berbagai jenis. Sebagai contoh folder berisi *file strings*.xml dengan definisi sumber *string*. Seperti yang akan anda lihat dari menambahkan bagian *color , folder* tata letak juga dapat berisi, misalnya, deskripsi warna. Di dalam file dimens.xml, terdapat *string-string* untuk konfigurasi dimensi aplikasi. Tidak jauh berbeda dengan dimens.xml maupun *strings*.xml, pada file styles.xml terdapat deklarasi variable untuk theme.

b. *Gradle scripts* : berfungsi sebagai *configurasi* aplikasi anda dengan *device*, serta menghubungkan *library* pendukung aplikasi yang anda buat. Penggunaan android studio ini untuk membuat aplikasi rumah pintar di *smartphone* android yang sebagai perangkat bantuan untuk mengontrol rumah pintar.

2.12 MySQL

MySQL adalah sebuah program *database* server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Laguage*). MySQL merupakan *database* server *multi user* dan *multi threaded* yang tangguh. Dengan memiliki banyak *feature* MySQL bisa bersaing dengan *database* komsersial sekalipun. Tidak mengejutkan MySQL menjadi *database* pilihan untuk banyak pengguna PHP. MySQL termasuk RDBMS (*Relational Database Management System*) itulah sebabnya istilah seperti tabel, baris dan kolom digunakan pada MySQL. Pada MySQL sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data. SQL adalah sebuah standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri.

Dengan menggunakan MySQL, proses akses *database* menjadi lebih *userfriendly* dibandingkan dengan menggunakan *dBASE* atau *Clipper* yang masih menggunakan perintah-perintah pemrograman. MySQL merupakan *software database* yang masih yang masih popular di lingkungan *Linux*, kepopuleran ini sangat ditunjang oleh performa *query* dari databasenya yang saat ini bisa dikatakan paling cepat dan jarang bermaslah. MySQL ini juga sudah berjalan pada lingkungan windows [22].

MySQL juga memiliki beberapa kelebihan, antara lain :

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *windows*, *Linux*, *FreeBSD, Solaris* dan lain-lain.

2. *Open Source*

   MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan cuma-Cuma.

3. *Multi User*

   MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance Tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Coloumn Types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *integer, double, char, text, date* dan lain-lain.

6. *Command and Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam *query*.

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta password terenkripsi.

8. *Scability and Limits*

   MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan *clients* menggunakan protokol TCP/IP, *Unix socket* (UNIX) atau *Named Pipes* (NT).

10. *Localisation*

    MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

    MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemograman dengan menggunakan fungsi API (*Aplication Programming Interface*).

12. *Clients and Tools*

    MySQL dilengkapi dengan berbagai tools yang dapat digunakan untuk administrasi *database* dan pada setiap tool yang ada disertakan petunjuk online.

13. Struktur Tabel

    MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan database lainnya semacam *PosgreeSQL* ataupun *Oracle.*

Kelemahan MySQL dari dulu sampai saat ini adalah *feature-creep* artinya MySQL berusaha kompatibel dengan beberapa standar serta berusaha memenuhinya namun jika itu diungkapkan kenyataannya bahwa fitur-fitur tersebut belum lengkap dan belum berperilaku sesuai standar. Contoh fitur *SUB-SELECT* (nesting *SELECT* dalam *SELECT*) yang tidak optimal dan sering salah parsing *query* SQL dan jalan keluarnya dengan memecah menjadi beberapa query [22].

2.13 JavaScript

Menurut Sunyoto, JavaScript adalah bahasa *scripting* yang popular di internet dan dapat bekerja di sebagian besar *browser* popular seperti *Internet Explorer* (IE), *Mozilla FireFox*, *Netscape*, dan *Opera*. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT.

Keunggulan javascript yang juga dikenal dengan nama ECMAScript yaitu dapat berjalan di semua *platform* dengan *browser* yang mendukung JavaScript, dan hampir semua *platform* dan *browser* saat ini mendukung JavaScript. Contoh dari aplikasi yang dibangun dengan JavaScript adalah *Google Maps* yang dapat berjalan di atas *Linux, Windows,* dan *Mac OS*. JavaScript juga semakin dan terus berkembang, seperti pertumbuhan pustaka (*library*) yang memudahkan untuk menavigasi dokumen, memlilih elemen DOM, membuat animasi, menangani *event* dan mengembangkan aplikasi *Ajax*. JavaScript adalah bahasa pemograman *client-side* yang *cross-platform* (berjalan di banyak *platform*) dan bersifat bebas (untuk dimodifikasi dan gratis) juga diadopsi secara universal [23].

2.14 JSON

Berdasarkan website resminya www.json.org, JSON merupakan bagian dari bahasa pemrograman JavaScript (Standard ECMA-262 3rd *Edition – December* 1999). JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa *data-interchange* yang ideal.

JSON dibangun dalam dua struktur:

1 Beberapa pasangan dari nama/nilai. Dalam beberapa bahasa pemrograman biasa disebut dengan istilah *object, record, struct, tabel hash, key list* atau *associative array*.
2 Nilai-nilai yang terusun secara *ordered list*. Biasa disebut dengan *array, vector, list* atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang *universal*, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format

data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON.

2.15 Jaringan Internet

Jaringan Internet adalah gabungan jaringan-jaringan komputer di seluruh dunia dan merupakan rangkaian komputer terbesar di dunia yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut.

Internet awalnya merupakan suatu rencana dari Departemen Pertahanan Amerika Serikat (*US Department of Defense*) pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan ARPANET atau *Advanced Research Project Agency Network*. Beberapa universitas di Amerika Serikat diantaranya *UCLA, Stanford, UC Santa Barbara* dan *University of Utah*, diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan empat computer di lokasi universitas berbeda tersebut.

Tahun 1985, Internet Modern memperoleh dukungan ketika *National Science foundation* membentuk NSFNET, menghubungkan lima pusat superkomputer di *Princeton University, Pittsburgh, University of California* di *San Diego, University of Illinois di Urbana-Champaign,* dan *Universitas Cornell*. Beberapa jaringan regional segera dikembangkan; Akhirnya, pemerintah menindahkan bagian-bagian ARPANET untuk NSFNET.

2.16 Definisi IP *Address*

IP *Address* merupakan deretan bilangan biner diantara 32 bit hingga 128 bit yang dipakai sebagai media untuk mengidentifikasi untuk setiap perangkat komputer yang terhubung pada jaringan komputer (intranet / internet). Bilangan biner 32 bit dipakai untuk setiap IP *Address* versi IPv4, sedangkan bilangan biner 128 bit digunakan untuk setiap versi IP *Address* IPv6.

IP *Address* akan berguna sebagai data identifikasi setiap *device* (komputer dan perangkat lainnya) yang terhubung ke jaringan komputer yang memanfaatkan internet *protocol* sebagai media penghubungnya. Adapun fungsi dari IP *address* yaitu :

1. Alat Identifikasi *Host* Atau Antar Muka Pada Jaringan Komputer.
   Fungsi IP *Address* yang pertama adalah sebagai alat identifikasi *host* ataupun antar muka jaringan komputer. Jika diilustrasikan seperti kehidupan nyata, maka IP *Address* berfungsi sebagai nama ataupun identitas seseorang. Dalam hal ini, seperti halnya nama, setiap komputer memiliki IP *Address* yang unik dan berbeda antara satu dengan yang lainnya (yang terkoneksi pada satu jaringan komputer).

2. Alamat Lokasi Jaringan.
   Fungsi IP *Address* yang kedua adalah sebagai penunjuk alamat lokasi jaringan. Jika kita ilustrasikan kembali dalam kehidupan nyata, maka IP *address* dapat diilustrasikan sebagai penunjukkan alamat rumah tempat tinggal seseorang. IP *Address* akan menunjukkan lokasi keberadaan sebuah komputer, berasal dari daerah mana, ataupun negara mana. Dalam hal ini, seperti halnya dalam kehidupan nyata, ada rute / jalan yang harus ditempuh agar data yang diinginkan bisa sampai ke komputer yang ingin dituju.

2.17 Apache

Apache merupakan server web yang dapat di jalankan di berbagai sistem operasi yang berbesa seperti *Unix, BSD, Linux, Windows* dan *Novell Netware* yang bertugas untuk melayani fasilitas web dengan menggunakan protocol http.

Apache sendiri bersifat *open source*, yang artinya dapat digunakan secara bebas oleh semua pengguna. Apache di kenal memiliki fitur-fitur canggih seperti penanganan pesan kesalahan yang dapat di konfigurasi serta otentikasi berbasis

basis data, dan memiliki tampilan GUI yang memungkinkan penanganan server menjadi mudah.

Web server Apache akan bekerja saat pengguna mengetikkan *request* melalui *protocol* http:// untuk membuka suatu halaman. Apache akan menjawab *request* yang di ketikkan *user* dan kemudian menampilkan halaman yang di minta.

Secara sederhana dalam pembuatan web akan menggunakan PHP sebagai bahasa pemrograman untuk memberikan intruksi-intruksi tertentu kepada komputer.

Kemudian SQL akan berfungsi sebagai *database* yang akan menyimpan setiap data yang di inginkan dalam bentuk tabel-tabel. Adapun fungsi tag HTML dan CSS dalam pembuatan web adalah brfungsi untuk menampilkan dan mengatur tampilan di setiap halaman.

Hingga ssat ini webserver Apache masih menjadi perangkat lunak yang populer dan paling banyak di gunakan di seluruh dunia. Apache sudah memasuki versi 2.4.16 yang released pada 15 Juli 2015.

2.17.1 Kelebihan Apache Server

Adapun kelebihan pada Apache server ini, yaitu:

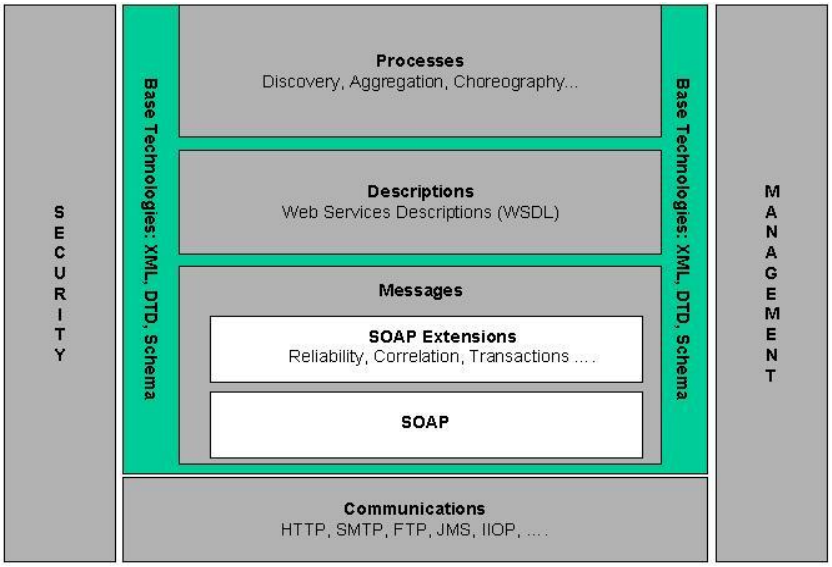
1. Mampu beroperasi pada berbagai *platform* sistem operasi.
2. Mudah dalam menambahkan periferal lainnnya ke dalam *platform* web servernya, misalnya : untuk menambahkan modul, cukup hanya menset file konfigurasinya agar mengikutsertakan modul itu ke dalam kumpulan modul lain yang sudah dioperasikan.
3. Mampu di kompilasi sesuai dengan spesifikasi HTTP yang sekarang.
4. Apache mempunyai komponen dasar terbanyak si antara webserver-webserver lain, yang berarti bahwa webserver Apache termasuk salah satu dari webserver yang lengkap.

Performansi dan konsumsi sumberdaya (*resource*) dari webserver Apache tidak terlalu banyak, hanya sebesar 20 MB untuk file-file dasarnya dan setiap daemon nya hanya memerlukan sebesar 950 KB memory per-child.

2.18 *Web Service*

Web service adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam *platform* dan antarmuka bahasa pemograman yang independen. Tujuan pengembangannya adalah untuk menjembatani komunikasi antar program, sehingga aplikasi yang satu dan aplikasi yang lain yang terdapat pada suatu jaringan yang sama atau pada jaringan berbeda dapat saling berkomunikasi asalkan menggunakan standart protokol yang ditetapkan oleh *web service*. Hal ini bisa terjadi, karena standart protokol itu tidaklah terikat pada suatu *platform* atau bahasa pemograman. Protokol itu sendiri dibangun oleh *Extensible Markup Language* (XML) yang memang kenyataannya telah didukung oleh banyak *paltform*, bahasa pemograman, dan oleh *developer* di seluruh dunia.

Arsitektur *web service* dibangun oleh beberapa layer dan teknologi yang saling berhubungan. Banyak cara untuk memvisualisaskan *service*, sama banyaknya dengan cara untuk membangun dan menggunakan *web service* gambar berikut hanyalah salah satu cara untuk menggambarkan arsitekturnya.



Sumber Gambar : Purba, Siregar[24]

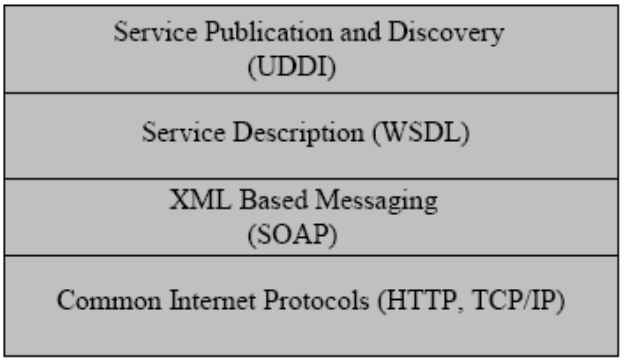
Gambar 2. 19 Arsitektur *Web Service*

Web service disusun oleh tiga komponen standart, yaitu :

1. *Simple Object Access Protocol* (SOAP), yaitu protokol yang bertanggungjawab dalam pertukuran informasi dalam lingkungan jaringan terdistribusi.
2. *Web service definition Language* (WDSL), dokumen standart yang dituliskan pada format XML, dan mendifinikasan kehadiran web *service* dalam suatu jaringan.
3. *Universal description, Discovery and Inttegration* (UDDI), yaitu suatu lokasi direktori yang beriskan *service* (layanan) dan bersifat bebas *platform* (*platform independent*), dituliskan berbasis XML dan dapat diakses oleh entittas yang berada di dalam dan luar jaringan.

Adanya standart tersebut membuat web *service* mudah di akses melalui berbagai antarmuka dan juga memberi peluang dimungkinkannya berbagai sistem yang dibangun pada platform yang berbeda dan bahasa yang berbeda untuk berkolaborasi pada dalam suatu pekerjaan [31].

2.18.1 Komponen-Komponen *Web Service*



Sumber Gambar : Purba dan Siregar[25]

Gambar 2. 20 Komponen-Komponen *Web Service*

Web service secara keseluruhan memiliki empat layer komponen seperti pada gambar di atas, yaitu:

1. Layer 1: Protokol internet standar seperti HTTP, TCP/IP
2. Layer 2: *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. Layer 3*: Web Service Definition Language* (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia [25].

2.19 Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah *programmer* didalam mendesain program dalam bentuk objek-objek dan hubungan

antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Suatu perusahaan *software* yaitu *Rational Software*, telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysist Design* (OOAD) [26].

2.19.1 UML (*Unified Modeling Language*)

UML adalah kependekan dari *Unified Modeling Language* yang merupakan suatu cara untuk menyelesaikan suatu masalah dengan mendeskripsikannya yang telah menjadi standar dalam dunia industri untuk memvisualisasikan, merancang dan mendokumentasikan sistem perangkat lunak. Dengan menggunakan UML kita dapat membangun model untuk segala bentuk dan jenis aplikasi perangkat lunak, yang mana aplikasi yang dibangun dapat berjalan pada perangkat lunak dengan sistem operasi dan jaringan apapun. Selain itu, dengan menggunakan UML, programmer dapat mengerti, memahami dengan jelas maksud, tujuan dan arah rancangan sistem, sehingga sistem perngakat lunak yang dirancang dapat dibangun dengan bahasa program apapun. Namun karena UML merupakan suatu model pengembangan sistem perangkat lunak yang berbasis *object oriented* sehingga menggnuakan bentuk *class* dan *operation* dalam konsep dasarnya, maka lebih cocok dalam pemrogramannya dengan menggunakan bahasa-bahasa pemrograman yang berbasis OOP (*Object Oriented Programming*) seperti C, C++, Java, VB dan lain sebagainya.

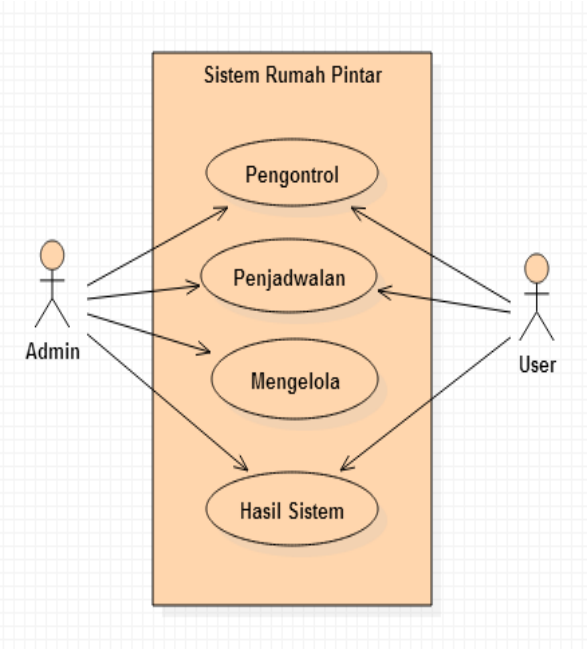
Secara lebih mendalam, UML lebih dari sekedar sebuah standard dan penemuan dari suatu notasi-notasi yang disatukan, tetapi juga berisi konsep-konsep baru yang menarik yang tidak ditemukan secara umum dalam komunitas *object oriented*. Dalam membangun suatu model perangkat lunak dengan UML, digunakan bentuk-bentuk diagram atau symbol untuk merepresentasikan elemen-elemen dalam sistem. UML ini digunakan untuk perancangan sistem yang nanti akan digunakan pada Bab 3 pada tugas akhir ini. Bentuk diagram yang digunakan untuk merepresentasikannya adalah sebagai berikut [26]:

1. *Use-case* Diagram
2. *Class* Diagram
3. *Sequence* diagram
4. *Activity* Diagram

2.19.1.1 *Use Case Diagram*

Use-case diagram merupakan suatu bentuk diagram yang menggambarkan fungsi-fungsi yang diharapkan dari sebuah sistem yang dirancang. Dalam *Usecase diagram* penekanannya adalah "*apa*" yang diperbuat oleh sistem, dan bukan "*bagaimana*". Sebuah *use-case* akan merepresentasikan sebuah interaksi antara pelaku atau *actor* dengan sistem.

Use-case diagram yang digunakan dalam mercancang suatu sistem dapat sangat membantu pada saat menyusun *requirement* sebuah sistem, mengomunikasikannya dengan klien, dan merancang pengujian untuk semua fitur yang terdapat dalam sistem. Dalam suatu sistem aplikasi *database*, *use-case diagram* sangat membantu *requierement* apa saja yang diperlukan [26].



Gambar 2. 21 Contoh *Use Case Diagram*

2.19.1.2 *Class Diagram*

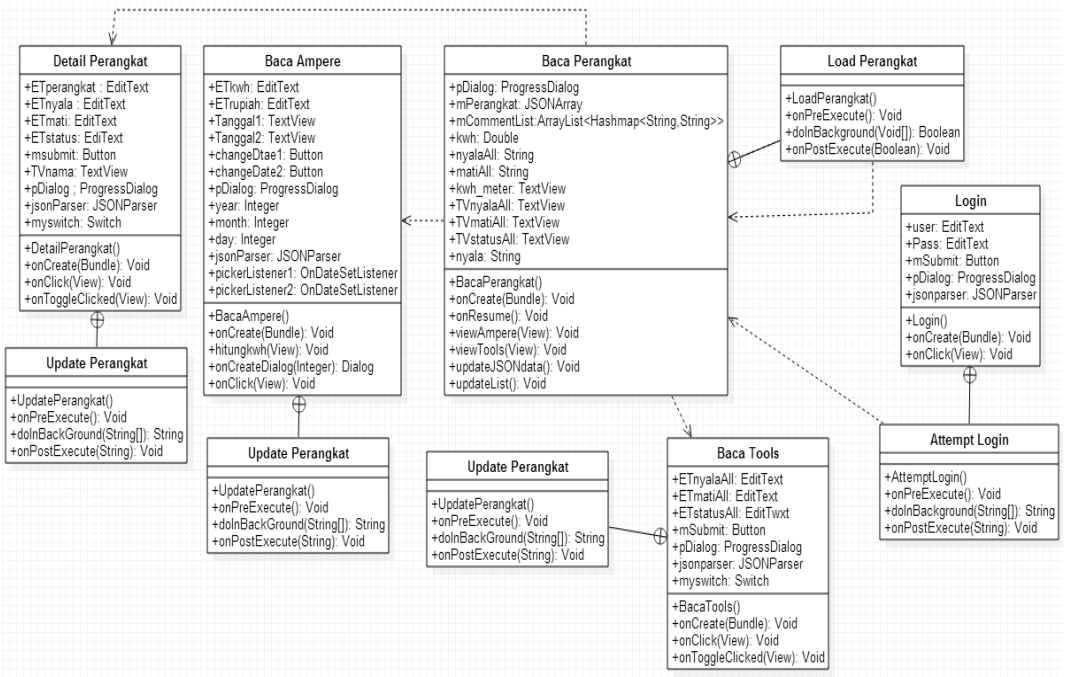
Sebuah *Class Diagram* menunjukkan struktur yang statis dari beberapa *class* dalam suatu sistem. *Class-class* merepresentasikan suatu keadaan (atribut/properti) dan yang akan dikerjakan oleh sistem (metoda/fungsi). *Class* memiliki tiga area pokok [26]:

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dalam class diagram dapat memiliki salah satu sifat seperti berikut di bawah ini:

1. *Private*, hanya dapat diakses oleh *class* itu sendiri.
2. *Protected*, hanya dapat diakses oleh *class* itu sendiri dan turunan dari *class* tersebut.
3. *Public*, dapat diakses oleh *class* selain dari *class* yang bersangkutan.

   Class dapat direpresentasikan dalam sebuah *interface* atau sebaliknya merupakan implementasi dari sebuah *interface* yang berupa *class* abstrak yang hanya tidak memiliki *attribute* dan hanya memiliki metoda. Berikut merupakan bentuk *class diagram* secara umum [26]:

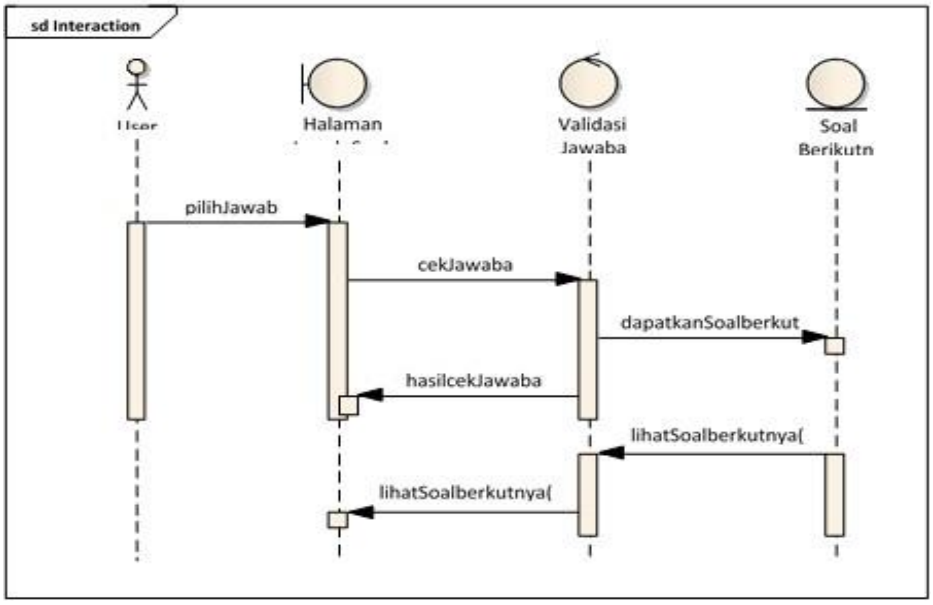


Gambar 2. 22 Contoh *Class Diagram*

2.19.1.3 *Sequence Diagram*

Sequence Diagram merupakan diagram yang mengambarkan kolaborasi yang dinamis antara obyek satu dengan yang lain. Kolaborasi ini ditunjukkan dengan adanya interaksi antar obyek di dalam dan di sekitar sistem yang berupa pesan atau instruksi yang berurutan.

Sequence diagram umumnya digunakan untuk menggambarkan suatu skenario atau urutan langkah-langkah yang dilakukan baik oleh *actor* maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil atau *output* [26].

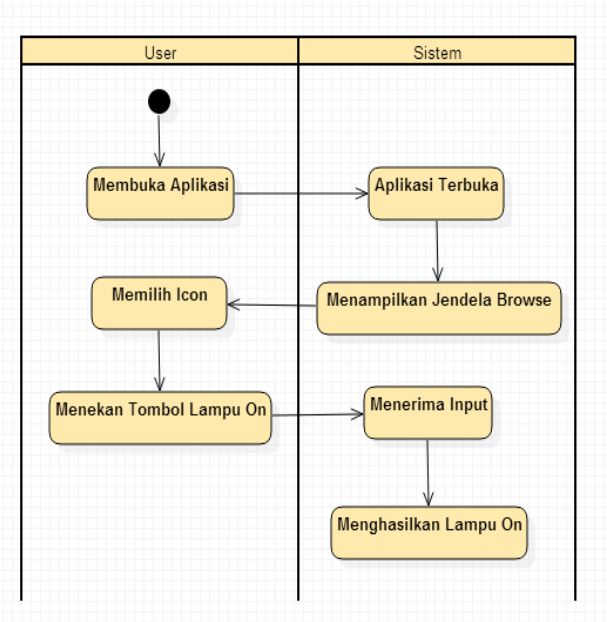


Sumber Gambar : Rosa, Shalahudin. M [26]

Gambar 2. 23 Contoh *Sequence Diagram*

2.19.1.4 *Acivity Diagram*

Sebuah *Activity Diagram* menunjukkan suatu alur kegiatan secara berurutan.*Activity Diagram* digunakan untuk mendiskripsikan kegiatan-kegiatan dalam sebuah operasi meskipun juga dapat digunakan untuk mendeskripsikan alur kegiatan yang lainnya seperti *use case* atau suatu interaksi [26].



Gambar 2. 24 Contoh *Activity Diagram*