

BAB II

LANDASAN TEORI

2.1 Pagar

Pagar adalah alat yang digunakan untuk membatasi atau mengelilingi pekarangan, tanah, rumah, kebun yang bertujuan untuk mengamankan rumah dari hewan buas, maling dan lain sebagainya (Rumahmin, 2015). Pagar perlu dikendalikan secara otomatis untuk proses buka tutupnya, karena dapat menghemat waktu yang ada.

2.2 Lampu

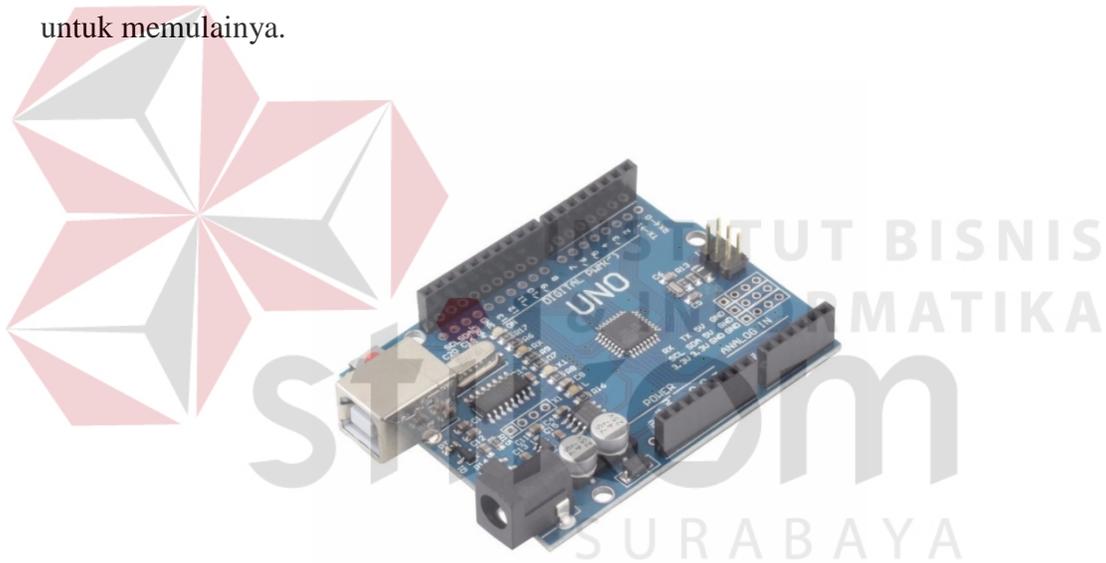
Lampu Listrik adalah suatu perangkat yang dapat menghasilkan cahaya saat dialiri arus listrik. Arus listrik yang dimaksud ini dapat berasal tenaga listrik yang dihasilkan oleh pembangkit listrik terpusat (*Centrally Generated Electric Power*) seperti PLN dan Genset ataupun tenaga listrik yang dihasilkan oleh Baterai dan Aki (Dickson, 2015). Sering lupa mematikan lampu jika tidak diperlukan, TV tetap nyala walaupun tidak ada yang menonton, baru ingat setrika pakaian setiap mau dipakai (sumber : www.kompasiana.com). Maka dari itu lampu perlu dikendalikan secara otomatis karena jika lupa untuk mematikan lampu, maka tagihan listrik dapat diminimalkan.

2.3 Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan

softwaranya memiliki bahasa pemrograman sendiri. (sumber : www.jurnalkompi.com)

Arduino UNO mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.



Gambar 2.1 Arduino UNO

sumber : www.jurnalkompi.com

2.4 Bluetooth

Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu

menyediakan layanan komunikasi data dan suara secara *real-time* antara *host-host bluetooth* dengan jarak jangkauan layanan yang terbatas. *Bluetooth* sendiri dapat berupa *card* yang bentuk dan fungsinya hampir sama dengan *card* yang digunakan untuk *wireless local area network* (WLAN) dimana menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada *bluetooth* mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah.

2.4.1 Bluetooth HC-05

Bluetooth HC-05 adalah sebuah modul Bluetooth SPP (Serial Port Protocol) yang mudah digunakan untuk komunikasi serial Wireless (Nirkabel) yang mengkonversi port serial ke Bluetooth. HC-05 menggunakan modulasi Bluetooth V2.0 + EDR (Enhanced Data Rate) 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2,4 GHz.

Modul ini dapat digunakan sebagai slave maupun master. HC-05 memiliki 2 mode konfigurasi, yaitu AT mode dan Communication mode. AT mode berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan Communication mode berfungsi untuk melakukan komunikasi Bluetooth dengan piranti lain.



Gambar 2.2 Bluetooth HC-05

Sumber : <https://www.itead.cc>

Dalam penggunaannya, HC-05 dapat beroperasi tanpa menggunakan Driver khusus. Untuk berkomunikasi antar Bluetooth, minimal harus memenuhi dua kondisi berikut :

1. Komunikasi harus antara master dan slave.
2. Password harus benar (saat melakukan pairing).

Jarak sinyal dari HC-05 adalah 30 meter, dengan kondisi tanpa halangan.

Adapun spesifikasi dari HC-05 adalah :

Hardware :

- Sensitivitas -80dBm (Typical)
- Daya transmit RF sampai dengan +4dBm.
- Operasi daya rendah 1,8V – 3,6V I/O.
- Kontrol PIO.

- Antarmuka UART dengan baudrate yang dapat diprogram.
- Dengan antenna terintegrasi.

Software :

- Default baudrate 9600, Data bit : 8, Stop bit = 1, Parity : No Parity, Mendukung baudrate : 9600, 19200, 38400, 57600, 115200, 230400 dan 460800.
- Auto koneksi pada saat device dinyalakan (default).

Auto reconnect pada menit ke 30 ketika hubungan putus karena Range koneksi.

2.5 RTC (*Real Time Clock*)

RTC (Real Time Clock) merupakan suatu chip IC yang berfungsi menghitung waktu mulai detik, menit, jam, hari, tanggal, bulan, hingga tahun dengan akurat dan menjaga atau menyimpan data waktu tersebut secara real time dengan sumber catu daya sendiri dan dinilai cukup akurat sebagai pewaktu (timer) dikarenakan menggunakan osilator Kristal eksternal. Sehingga saat perangkat mikrokontroler yang difungsikan untuk mengakses RTC sebagai sumber data waktu dimatikan, data waktu tidak akan hilang selama baterai yang terhubung pada RTC tersebut tidak mati (agus, 2016).

2.5.1 RTC DS1302

RTC yang digunakan adalah RTC DS1302, merupakan produk keluaran maxim. Fitur yang ditawarkan antara lain penghitung secara *real time* (untuk detik, menit, jam, hari, tanggal, bulan, dan tahun), format waktu dapat disetting ke dalam format 12 jam (AM/PM) atau 24 jam, memiliki kemampuan penyesuaian jumlah

hari/bulan terhadap tahun kabisat, 31 x 8 *Battery-Backed General-Purpose* RAM, menggunakan antarmuka i2c (*Serial Data* dan *Serial Clock*), menggunakan tegangan input kerja 5 Vdc, harus memakai baterai *back-up* dengan *range* 2-5 Vdc, konsumsi arus pada baterai *back-up* yaitu hanya 200nA, *Range* suhu kerja optimal antara -40°C sampai +85°C (agus, 2016).

2.6 Relay

Menurut Andrianto(2015) *relay* adalah komponen elektronika berupa saklar elektronik yang dikendalikan oleh arus listrik. Secara prinsip kerja, *relay* merupakan tuas saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya. Ketika solenoid dialiri arus listrik, tuas akan mendapat tarikan medan magnet yang dihasilkan dari solenoid sehingga kontak saklar akan menutup. Pada saat arus tidak diterima solenoid maka gaya magnet akan hilang, dan saklar akan kembali terbuka. Susunan kontak pada *relay* adalah sebagai berikut:

1. *Normally Open*: saklar dari *relay* yang dalam keadaan normal (relay tidak diberi tegangan) tidak terhubung dengan *common*.
2. *Normally Close*: saklar dari *relay* yang dalam keadaan normal (relay tidak diberi tegangan) terhubung dengan *common*.

Relay berfungsi sebagai *output*. Dimana *relay* mendapatkan *input* dari pin *output* yang member logika 1 (*high*) apabila mendapatkan input dari pin GPIO bernilai 1 (*high*) sehingga relay dalam kondisi NC (*Normally close*). Begitu juga sebaliknya.



Gambar 2.3 Relay

2.6.1 Relay Single 2 Channels

Relay single 2 channels merupakan Relay 5V dengan 2 channel output. Dapat digunakan sebagai saklar elektronik untuk mengendalikan perangkat listrik yang memerlukan tegangan dan arus yang besar. Kompatibel dengan semua mikrokontroler (khususnya Arduino, 8051, 8535, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic) maupun Raspberry Pi .

Relay 2 Channel ini memerlukan arus sebesar sekurang-kurangnya 15-20mA untuk mengontrol masing-masing channel. Disertai dengan relay high-current sehingga dapat menghubungkan perangkat dengan AC250V 10A (Gerai, 2014). Jika menggunakan mikrokontroler dengan tegangan kerja 3.3V, Anda tetap dapat menggunakan Relay 2 channel ini dengan cara :

1. Lepas jumper JD-VCC
2. Hubungkan JD-VCC dengan external power 5V lainnya.

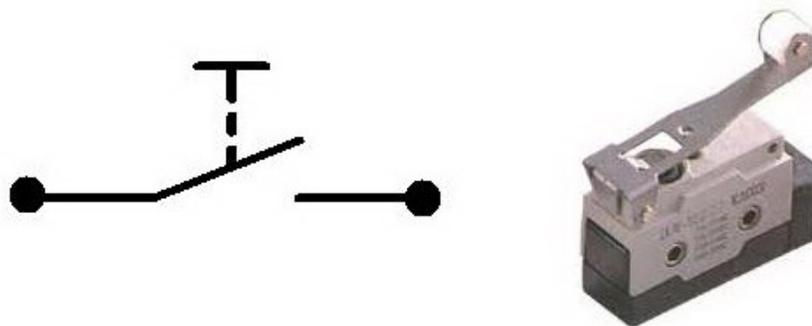


Gambar 2.4 Relay single 2 channels

Sumber : <https://www.geraicerdas.com>

2.7 Limit Switch

Limit switch merupakan jenis saklar yang dilengkapi dengan katup yang sebagai pengganti tombol. Prinsip kerjanya sama seperti saklar *Push ON* yaitu hanya akan menghubungkan ketika katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutus ketika katup tidak ditekan. Limit switch digunakan untuk memberhentikan pagar ketika proses membuka dan menutup.



Gambar 2.5 Limit Switch

Sumber : <https://www.elektronika-dasar.web.id>

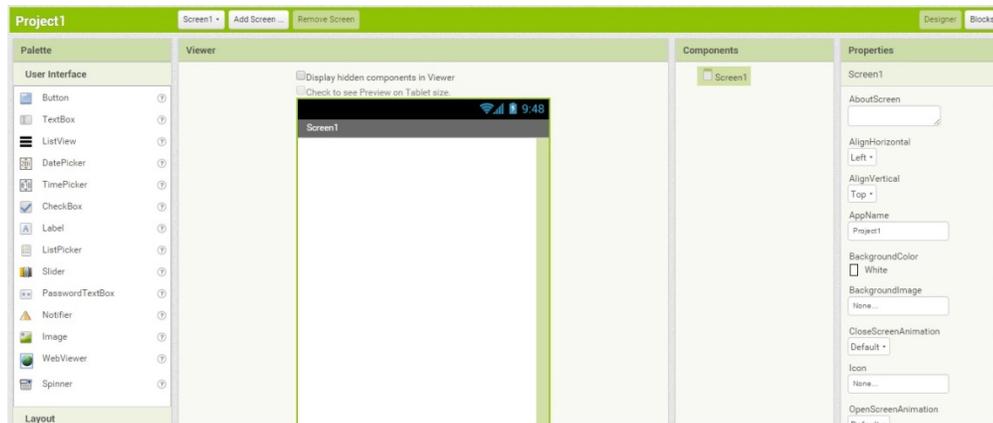
2.8 Sistem Operasi Android

Android adalah *operating system* berbasis linux yang diperuntukkan khusus untuk *mobile device* seperti *smartphone* atau *PC table*, persis seperti Microsoft windows yang sangat dikenal baik oleh pengguna komputer dan laptop, jika kita analogikan, android adalah windowsnya sedangkan *smartphone* atau *PC table* adalah unit komputernya (andik dan edi, 2015).

2.9 App Inventor

MIT App Inventor adalah aplikasi *web source code* yang awalnya dikembangkan oleh google, kemudian dikembangkan oleh Massachusetts Institute of Technology (MIT). App inventor memudahkan seorang programmer pemula untuk memprogram komputer dalam menciptakan aplikasi perangkat lunak bagi sistem operasi android. App inventor menggunakan antar muka grafis yang serupa dengan antar muka pengguna pada Scratch dan StarLogo TNG, sehingga memudahkan pengguna untuk *men-drag and drop* objek visual untuk menciptakan aplikasi yang bisa dijalankan oleh aplikasi android. Dalam menciptakan app inventor, Google melakukan beberapa riset yang berhubungan dengan komputasi edukasional dan menyelesaikan lingkungan pengembang *online* google.

App inventor dikembangkan oleh Google dan MIT Media Lab, menggunakan bahasa pemrograman Java dan Kawa *Scheme* karena kedua bahasa pemrograman tersebut digunakan untuk memudahkan pengguna app inventor untuk melakukan menjalankan proses perhitungan dan beberapa kebutuhan lainnya yang dibutuhkan dalam sistem operasi android.



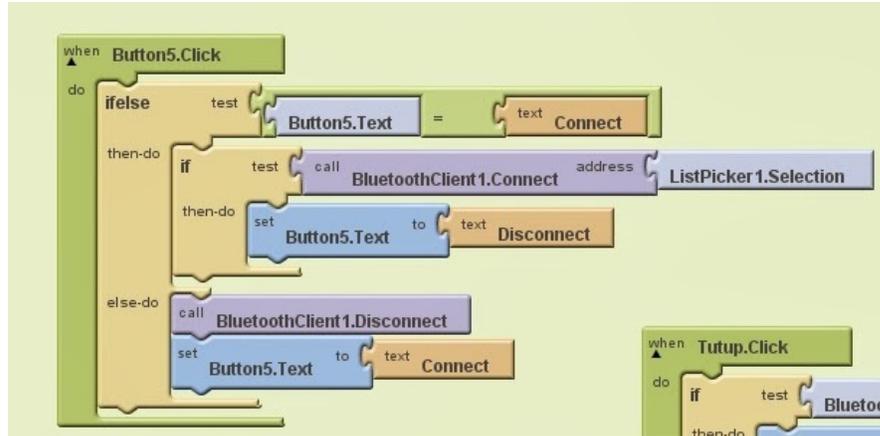
Gambar 2.6 Preview app inventor

Sumber : <http://appinventor.mit.edu>

Aplikasi app inventor selesai dibuat pada juli 2010 dan dirilis untuk *public* pada 15 desember 2010. Namun, google menghentikan *project* app inventor pada 31 desember 2011 dan sekarang app inventor dipegang MIT *Centre for Mobile Learning* dengan nama MIT App Inventor.

2.9.1 Bahasa Blok Untuk Aplikasi Mobile

App inventor adalah sebuah pemrograman visual yang digunakan untuk mengembangkan aplikasi berbasis android dengan dukungan fitur berupa *drag-drop tool*. Anda dapat mendesain *user interface* dari sebuah aplikasi dengan menggunakan *web GUI (Graphical User Interface) builder*, kemudian anda dapat menspesifikasikan *behavior* aplikasi dengan memasang blok yang sesuai seperti saat anda bermain *puzzle*.



Gambar 2.7 Contoh blok app inventor

sumber : <http://appinventor.mit.edu>

Contoh blok dari versi app inventor yang dijelaskan pada gambar 2.6 menampilkan sebuah aplikasi mesin penjawab sms, aplikasi tersebut dapat digunakan untuk proses tanya jawab sms dari nomor yang telah ditentukan sebelumnya. Pada gambar di atas juga menjelaskan bahwa blok program lebih mudah dipahami daripada program tradisional yang berupa kode. Sehingga dapat disimpulkan app inventor dapat membangun sebuah aplikasi yang bisa berbicara sendiri dan aplikasi yang dibuat dapat diubah lagi pada saat pengiriman respon.

2.9.2 Kemampuan App Inventor

Beberapa hal menyenangkan yang akan didapatkan jika mempelajari app inventor dapat dijelaskan sebagai berikut :

- a. Just for fun, maksudnya aplikasi app inventor menawarkan proses eksplorasi dan *discovery*. App inventor dibuka pada *web browser*, dengan menghubungkan *smartphone* ke internet kemudian tinggal menyusun beberapa blok app inventor seperti yang ditunjukkan pada gambar 2.6 Setelah aplikasi selesai dibangun, maka aktifkan dan dicoba untuk berinteraksi dengan *smartphone*.

- b. *Prototype*, jika mempunyai ide untuk membuat aplikasi maka tuangkanlah dengan menggunakan app inventor. *Prototype* adalah sebuah model kasar dan belum lengkap namun mewakili ide-ide dasar yang ada.
- c. Membangun aplikasi sesuai dengan kepribadian. Jika bosan dengan aplikasi yang sudah tersedia, maka app inventor dapat membangun aplikasi sesuai dengan keinginan.
- d. Mengembangkan aplikasi secara lengkap. App inventor tidak hanya digunakan untuk membuat *prototype system* atau mendesain tampilan luar. Namun dapat membangun aplikasi tersebut menjadi sebuah aplikasi yang bersifat *general purpose*. Pada bagian desain dalam app inventor, menyediakan semua pemrograman fundamental seperti perulangan dengan kondisi dalam bentuk *form block*.
- e. *Tech and Learn*. Meskipun dengan bekal pendidikan dari sekolah menengah, sekolah tinggi atau universitas, app inventor adalah sebuah guru dan peralatan belajar yang paling efisien. Hal ini baik bagi yang ingin belajar tentang sains komputer, karena dapat belajar saat membuat aplikasi.

Kebanyakan orang berpikir, bahwa app inventor sangat mudah untuk digunakan. Karena bersifat visual dan mempunyai fitur *drag and drop interface*. Adapun kelebihan dari app inventor adalah sebagai berikut.

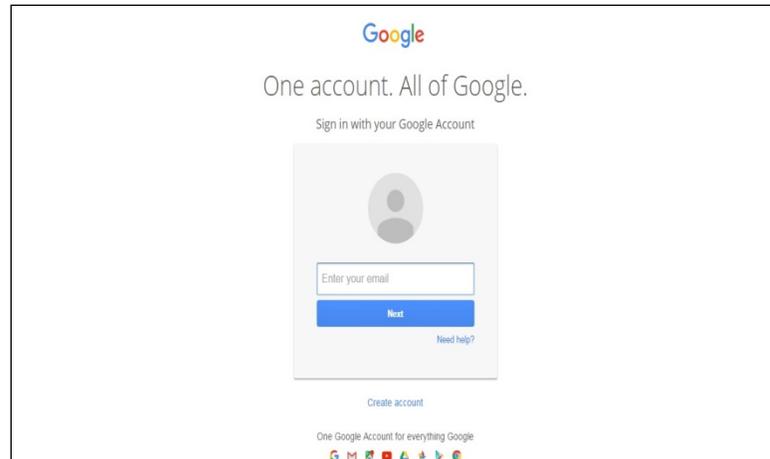
- a. Tidak perlu mengingat dan menulis instruksi, karena kesalahan dari para programmer pemula sudah muncul pesan *error* pada jendela pesan. Jadi, app inventor membuat pemula tidak frustrasi dalam mengembangkan aplikasi.

- b. Hanya seperti memilih pilihan, karena app inventor membuat komponen-komponen dan blok seperti kanvas pada lukisan atau bidang gambar. Sehingga dapat dengan mudah memprogram hanya dengan mencari blok-blok program yang sesuai dan tidak perlu mengingat instruksi atau harus membuka manual program.
- c. Seperti blok yang dipasangkan satu sama lain. Pemrograman dengan app inventor tidak menampilkan pesan *error*, akan tetapi bahasa blok pada app inventor akan mencegah anda untuk melakukan kesalahan di tempat pertama.
- d. Dapat menangani *event* secara langsung. Bahasa pemrograman tradisional telah di desain bahwa pemrograman seperti bekerja dengan resep atau beberapa instruksi. Akan tetapi, dengan *interface* grafis dan khususnya untuk aplikasi *mobile*, *event* dapat terjadi kapan saja. Sebuah event handler ibarat seperti berkata “ketika ini terjadi, aplikasi melakukan ini”. Dalam sebuah Bahasa tradisional seperti java, maka anda juga harus memahami *class*, *object* dan objek khusus yang sering disebut *listener* untuk mengekspresikan sebuah *event* sederhana.

2.9.3 Penggunaan App Inventor

Adapun langkah-langkah dalam menggunakan app inventor adalah sebagai berikut :

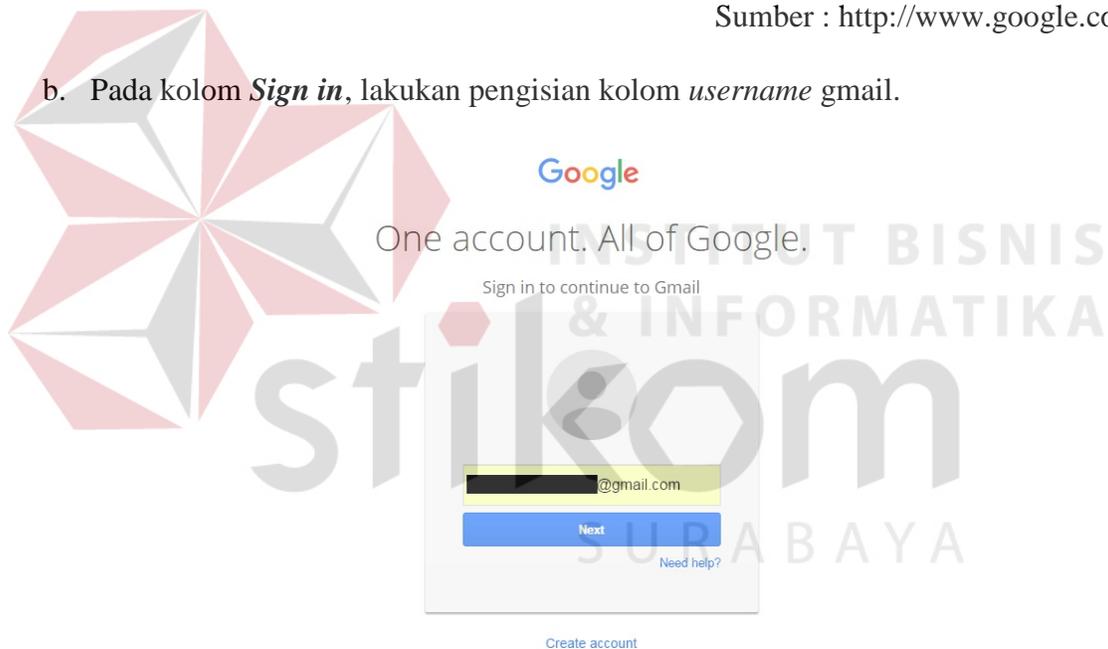
- a. *Login* terlebih dahulu pada akun gmail. Selanjutnya, *web browser* akan menampilkan jendela login gmail seperti yang ditunjukkan pada gambar 2.8.



Gambar 2.8 Halaman *login* gmail

Sumber : <http://www.google.com>

b. Pada kolom *Sign in*, lakukan pengisian kolom *username* gmail.



Gambar 2.9 Mengisi kolom *username* pada gmail

sumber : <http://www.google.com>

c. Kemudian, pada kolom *password* pada gmail lakukan pengisian *password*.



One account. All of Google.

Sign in to continue to Gmail

←

r

rachmat priambodo
[redacted]@gmail.com

.....

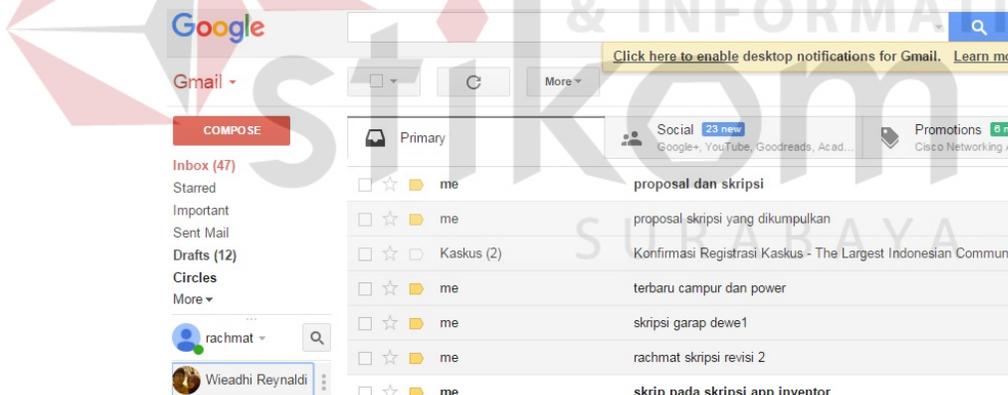
Sign in

Stay signed in [Forgot password?](#)

Gambar 2.10 Mengisi kolom *password* pada gmail

Sumber : <http://www.google.com>

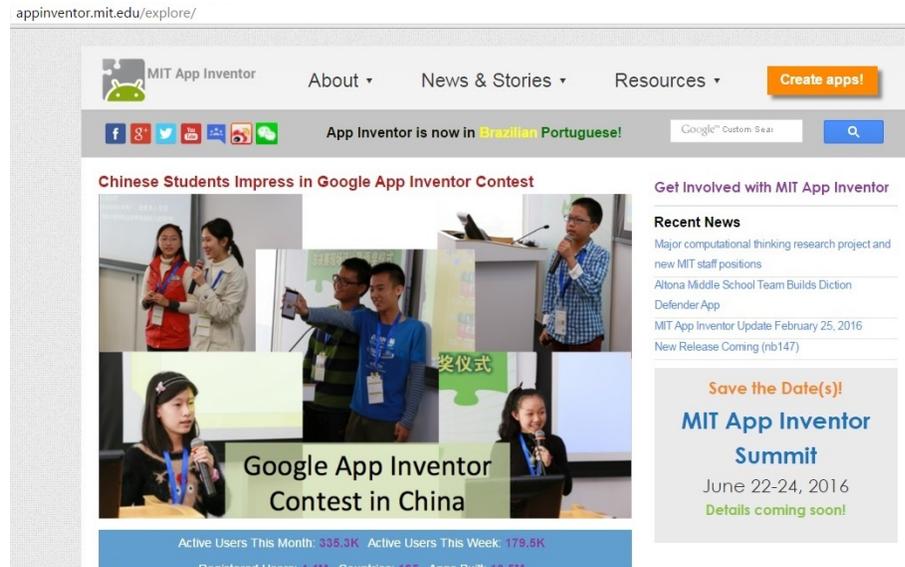
d. Masuk ke halaman utama email.



Gambar 2.11 Halaman utama email

Sumber : <http://mail.google.com>

e. Langkah selanjutnya, buka *tab* baru pada *browser*. Pada *tab* yang baru, ketikkan alamat <http://appinventor.mit.edu> pada *address bar*.



Gambar 2.12 Halaman *login* app inventor

sumber : <http://appinventor.mit.edu>

Pilih menu pilihan



bagian kolom atas, paling kanan *login* app

inventor.

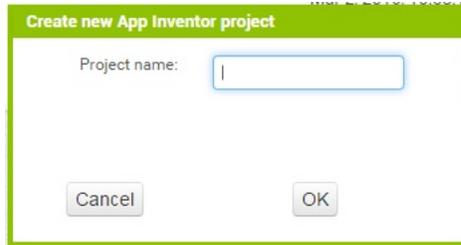
- f. Selanjutnya akan ditampilkan jendela **Projects** dari app inventor.
- g. Untuk membuat project baru, klik tombol **Start new project** pada app inventor.



Gambar 2.13 *Button Start new poject*

Sumber : <http://appinventor.mit.edu>

- h. Selanjutnya akan ditampilkan jendela **New App Inventor project**. Kemudian pada **project name**, misalnya **Project1**.



Gambar 2.14 Create New App Inventor project

sumber : <http://appinventor.mit.edu>

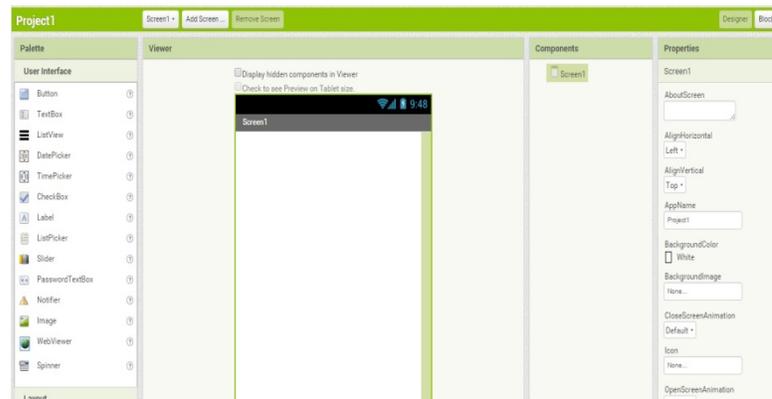
- i. Langkah selanjutnya, klik pada button **OK**. Selanjutnya, app inventor akan membuat *project* baru bernama **Project1**.



Gambar 2.15 Project baru telah berhasil dibuat

sumber : <http://appinventor.mit.edu>

- j. Selanjutnya, pada gambar 2.16 app inventor akan menampilkan secara otomatis IDE untuk *project* baru yang telah dibuat.



Gambar 2.16 Jendela *project* baru pada app inventor

sumber : <http://appinventor.mit.edu>

Project dari app inventor siap untuk digunakan dan diubah-ubah sesuai dengan keperluan yang dibutuhkan. *Project* aplikasi yang telah dibuat secara otomatis akan tersimpan ke dalam akun gmail pada *user* app inventor.

