

Received : February 08, 2021
Accepted : February 12, 2021
Published : March 03, 2021

Conference on Community Engagement Project
<https://journal.uib.ac.id/index.php/concept>

Perancangan Dan Implementasi Sistem Smart Office Pada Pt. Dunia Berjaya Abadi Menggunakan Internet Of Things

Benny Kosasih¹, Tony Wibowo²

Universitas Internasional Batam,^{1,2}

Email korespondensi: bennykosasih.1631088@uib.ac.id, tony.wibowo@uib.ac.id

Abstrak

Internet of Things (IoT) mulai terkenal dimasyarakat ketika industri 4.0 mulai diterapkan, hal yang menjadi daya tarik IoT salah satu adalah efisiensi energi. Perusahaan PT. Dunia Berjaya Abadi adalah perusahaan yang menjual jasa transportasi. dengan IoT adalah saling terhubungnya cloud dengan Microcontroller Unit(MCU) sehingga dapat diakses langsung secara online. Praktik ini adalah untuk memberikan sentuhan teknologi dari masa ini supaya mempermudah pekerjaan dan efisiensi penggunaan energi melalui internet layaknya Smart Office. IoT didukung oleh teknologi komunikasi MQTT dan TLS untuk berkomunikasi dengan cloud. Praktik ini dimulai dari perencanaan dan diakhiri oleh penerapan *Smart Office* untuk PT. Dunia Berjaya Abadi.

Kata Kunci : Internet of Things, Smart Office, Cloud.

Abstract

Internet of Things (IoT) began to be famous in the community when industry 4.0 began to be implemented, one of the things that attracts IoT is energy efficiency. PT. Dunia Berjaya Abadi is a company that sells transportation services. with IoT is the interconnection of the cloud with the Microcontroller Unit (MCU) so that it can be accessed directly online. This practice is to provide a touch of technology from this era so as to facilitate work and efficient use of energy through the internet like a Smart Office. IoT is supported by MQTT and TLS communication technology to communicate with the cloud. This practice starts with planning and ends with the implementation of Smart Office for PT. Dunia Berjaya Abadi

Keywords: *Internet of Things, Smart Office, Cloud.*

Pendahuluan

Smart Office menurut Giacobbe et al., (2018) adalah peningkatan efisiensi penggunaan energi, ramah lingkungan dan peningkatan kenyamanan perkantoran. Ramah lingkungan adalah hal yang penting sebagai tolak ukur pintar, khususnya dalam pengurangan polusi. Kelestarian lingkungan terdiri

dari, sosial, ekonomi dan lingkungan. Dalam aspek sosial kantor pintar harus didesain untuk membantu para pekerja dalam pengaturan komunikasi yang membantu para pekerja untuk lebih produktif, misalkan pekerjaan yang membutuhkan konsentrasi tinggi, para pekerja lebih nyaman jika lebih minim interaksi sosial dalam kantor, akan tetapi pekerjaan lain yang membutuhkan kerja sama

seperti projek membutuhkan bantuan input dari teman lainnya. Kantor pintar seharusnya didesain untuk membantu para pekerja dalam sosial interaksi dikantor sesuai kebutuhan para pekerja karena minimnya interaksi ataupun terlalu banyak interaksi, akan mengganggu kemampuan pekerjaan pengguna kantor (Tuzcuoglu et al., 2020).

Pemakaian jarak jauh atau *remote control* membantu kita dalam hal mengatur lebih cepat dan lebih nyaman. *Remote control* juga bisa membuat pengguna menghemat energi, dan memberikan pengguna kenyamanan lebih baik daripada sistem tradisional. Pendeteksi api dapat dipasang pada IoT untuk dapat mendeteksi resiko terjadinya kebakaran sebelum Api itu membesar (Delsing, 2017; Ko et al., 2020).

Cloud Computing menurut Barolli et al., (2019) pemrograman dibatasi oleh kebutuhan ekonomi bukan oleh keterbatasan teknis, sebuah program paradigma yang memberikan efisiensi dalam penggunaan sumber daya cloud menyesuaikan dengan kebutuhan bisnis. Walaupun begitu implementasikan banyak domain dan memberikan utilitas yang besar, sesuai dengan perkembangan ini semakin banyak klien yang menyimpan datanya di *cloud*. Cloud adalah infrastruktur penting dalam membangun sistem IoT (Delsing, 2017; Wazid et al., 2020).

Ditengah permasalahan COVID-19, *smart city* bisa membantu *social distancing* yang harus dilakukan untuk menurunkan jumlah kontak orang ke orang yang terjadi pada masyarakat. Bekerja dari rumah merupakan keharusan pada masa COVID-19, dan dipercaya bahwa setelah masa COVID-19 masih tetap

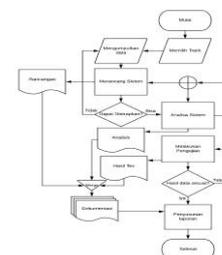
bekerja dari rumah IoT dipercaya akan membantu dalam hal mempermudah para pekerja bekerja dari rumah (Kaushik & Guleria, 2020). Bekerja dari rumah memberikan beberapa keuntungan antara lain, peningkatan efisiensi dalam penggunaan energi, emisi CO₂ yang lebih rendah pada April 2020 jika dibandingkan tahun lalu dengan total penurunan 17% CO₂, peningkatan dalam interaksi para pekerja dengan orang rumah (Kylili et al., 2020; Onoda, 2020).

PT. Dunia Berjaya Abadi model bisnis jasa sewa mobil, perusahaan memiliki sebuah gedung perkantoran, dan para pekerjanya sering keluar dari kantor secara terburu-buru, sehingga sistem kontrol jarak jauh akan mempermudah pengguna kantor untuk menggunakan energi secara lebih efisien.

Untuk meningkatkan kinerja para pekerja kantor, penulis percaya bahwa IoT akan berdampak besar pada peningkatan efisiensi, dan kualitas kerja. Interaksi antara pekerja, dan kantor pintar, antara lain: dapat memberikan perasaan nyaman, meningkatkan mood, akan menghemat waktu, menjadi alat yang hebat dalam membantu orang-orang dalam produktivitas pekerjaan, meningkatkan efisiensi, dan efektivitas (Pathan & Deval, 2020).

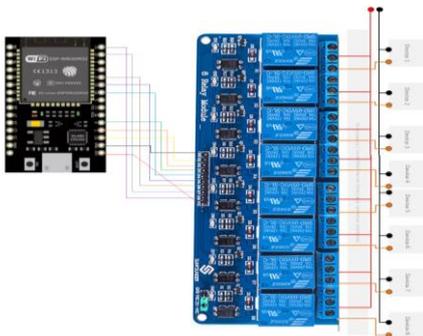
Metode

Berikut rancangan penelitian akan dipresentasikan dalam bentuk flowchart :



Gambar 1 Rancangan Penelitian

Untuk membuat sistem ini diperlukan banyak perangkat keras maupun perangkat lunak antara lain: Komputer yang akan digunakan sebagai perangkat yang mengkonfigurasi alat. Spesifikasi Processor Intel i7-4750HQ CPU frekuensi 2.00GHz dan memori RAM 12288MB RAM; ESP32DEVKITV1 adalah Internet of Things, dengan spesifikasi Processor Tensilica Xtensa Dual-Core 160 atau 240 MHz, ROM 448 KB, SRAM: 520 KB, bisa WiFi 802.11n, dan Bluetooth V 4.2; Sistem operasi Ubuntu 18.04 untuk komputer yang menjalankan aplikasi Arduino IDE 1.8.9 untuk menjalankan esp32. Ketika merancang sistem *Smart Office* telah ditentukan bahwa yang paling dibutuhkan untuk perusahaan adalah akses jarak jauh untuk kontrol peralatan elektrik pada kantor, Gambar 2 menunjukkan perancangan skematik esp32



Gambar 2 Rancangan Skematik ESP32

Pada tahap Instalasi Ubuntu 18.04 LTS, penulis melakukan instalasi *operating system* Ubuntu versi 18.04 *Long Term Support* pada komputer. Pada tahapan ini penulis akan menggunakan USB *flash drive*, dan Rufus versi 3.5. pertama download ubuntu 18.04 LTS, kemudian menggunakan Rufus membuat *bootable* USB Ubuntu. Setelah itu masuk ke BIOS untuk booting pada *installer* lalu lanjutkan

pada instalasi Ubuntu. Selanjutnya Melakukan instalasi *toolchain* pada Ubuntu yaitu ESP-IDF yang diinstall dengan *command line* dan Arduino IDE.

Untuk menginstall arduino IDE, pertama dilakukan dengan mendownload Arduino IDE, lalu menginstal xz-utils untuk dapat menginstal Arduino IDE dengan *command line* di terminal: \$ sudo ./install.sh. setelah Port sudah terdeteksi, maka dilakukan penyambungan ESP32 ke Komputer dan pastikan terdeteksi baik secara logis maupun secara fisik (dilihat dari lampu LED yang menyala pada ESP32 pada *breadboard*. Setelah terhubung dan sudah dilakukan pengujiannya, berikutnya adalah menghubungkan ESP32 ke internet. Pertama perlu registrasi akun ubidots, mencari dan mengubah *source code* yang cocok dengan implementasi, memasang esp32, relay, dan menghubungkan kelistrikan perangkat.

Pembahasan

Pada saat implementasi ada beberapa tahapan untuk penerapan yaitu:

1. Instalasi Ubuntu 18.04 LTS

Pada tahap ini penulis melakukan instalasi operating system Ubuntu versi 18.04 *Long Term Support* pada komputer. Pertama unduh ubuntu, dengan memilih versi Ubuntu yang diinginkan (tidak harus sama dengan penulis, penulis memilih 18.04 LTS karena waktu itu adalah versi *Long Term Support* tertinggi pada waktu penelitian dilakukan), pilih unduh dalam bentuk iso. Kemudian menggunakan Rufus 3.5 membuat *bootable* USB Ubuntu yaitu dengan cara pertama sediakan usb minimal 4 GB, dilanjutkan dengan membuka rufus dan mengatur

drive properties dengan konfigurasi *device* nama usb, *boot selection* iso ubuntu, *partition scheme* GPT, tipe *system* uefi yang (non-csm) dan setelah itu atur *format option* FAT32 dan *cluster size* 4096 bytes, tekan tombol start. Kemudian akan keluar *pop-up warning* dari rufus ISOhybrid, pilih '*Write in ISO Image Mode*', klik ok. Setelah itu akan keluar konfirmasi untuk format usb, klik ok.

Pasang *bootable* USB ke komputer lalu masuk ke BIOS untuk booting pada *installer*, pertama-tama untuk memilih menggunakan keyboard tekan enter, pilih bahasa instalasi komputer, kemudian pilih install ubuntu. Kemudian akan muncul *Welcome* dari ubuntu, pilih inggris klik tombol *continue*, pilih *Keyboard Layout*, (klik *detect keyboard*) mayoritas keyboard adalah English(US) lalu tes keyboard, jika sudah cocok klik *continue*, pada bagian *update and other software* pilih boleh yang *normal installation* dan/ataupun *install third-party software for graphics and Wi-Fi hardware and additional media formats* klik *continue*, kemudian pilih jenis instalasi, kalau penulis memilih *erase disk and install ubuntu*, terus klik *install now*, pada tahap *who are you?* Isi data lengkap nama, nama komputer, *username*, *password*, *password confirmation*, pilih tipe login, lalu klik *continue*, kemudian pastikan terhubung ke internet tunggu instalasi selesai sampai muncul *installation complete* klik *restart now*. Setelah itu akan ada perintah untuk mencabut USB, dan tekan Enter pada keyboard.

2. Instalasi ESP-IDF

Melakukan instalasi toolchain pada Ubuntu yaitu ESP-IDF. Buka terminal linux masukkan *command line*:

```
$ cd ~
$ mkdir esp32
```

```
$ cd esp32 $ sudo apt-get
instal git wget make
libncurses-dev flex bison
gperf python pythonserial
$ wget
https://dl.espressif.com/dl/xtensa-esp32-elf-linux64-1.22.0-80-g6c4433a5.2.0.tar.gz
$ tar -xvf xtensa-esp32-elf-linux64-1.22.0-80-g6c4433a-5.2.0.tar.gz
$ sudo mv xtensa-esp32-elf /opt
$ export PATH=/opt/xtensa-esp32-elf/bin:$PATH
$ git clone
https://github.com/espressif/esp-idf.git esp-idf
$ cd esp-idf/
$ git checkout v3.0
$ git submodule update --init --recursive
$ export IDF_PATH=~/.esp32/esp-idf
$ cd ~/.esp32
$ mkdir apps
$ cd apps
$ git clone
https://github.com/espressif/esp-idf-template.git myapp
$ cd myapp
$ make menuconfig
$ sudo apt-get instal doxygen
$ doxygen -g
$ ~/.esp32/crosstool-NG$ ./ct-ng build
$ sudo apt instal screen
```

Untuk menjamin kelancaran proses, perlu untuk mengecek serial port yang tersedia dengan script: \$ sudo dmesg | grep -i tty

3. Instal Arduino IDE

Pertama melakukan download Arduino IDE dari www.arduino.cc, menginstal xz-utils untuk dapat menginstal Arduino IDE lalu memasukan beberapa command line di terminal:

```
$ cd ~
$ cd Downloads
$ cd arduino-1.8.9
$ sudo ./install.sh
```

Jika sudah selesai instalasi, maka akan muncul kata *done!* pada terminal. Setelah itu buka Arduino IDE, lalu buka *File > Preferences*, akan muncul kotak *setting*, masukan

https://dl.espressif.com/dl/package_esp32_index.json pada *Additional Board Manager URLs*, pilih tombol OK. Klik Tools > Board > Boards Manager kemudian cari esp32 by Espressif Systems, klik install dan Klik Tools > Board > DOITESP32DEVKITV1. Untuk Melihat Port yang terhubung, Klik Tools> Ports.

4. Menghubungkan ESP32 ke komputer

Hubungkan ESP32 ke komputer menggunakan kabel *Micro USB*, setelah itu cek serial port, masukan command line “# dmesg | grep tty” di Terminal, lalu buka putty, ‘# sudo putty’ Pilih Category> Serial, lalu ubah speed baud dari 9600 menjadi 115200. Setelah itu lakukan pengetesan pada ESP32 menggunakan library seperti WiFiScan. Lalu baca hasil *read out* pada putty atau *Serial Monitor* (Arduino IDE).

5. Menghubungkan ESP32 ke Internet menggunakan Ubidots

Pertama perlu registrasi akun ubidots, setelah itu masuk ke *dashboard* ubidots, lalu tekan *Tab Devices> Devices*. Setelah itu akan muncul halaman *Devices*, klik *Create Devices* lalu pilih *Blank*, masukan *Device Name*, masukan *Device Label*, pilih *Device* baru yang buat, klik *Add Variable > klik Raw*. Lalu buat nama masing – masing raw relay 1 sampai relay 8. Kembali ke *dashboard* ubidots, buat *function > switch. Widget behavior Static, Off Value = 0, On Value = 1, Off Message = Off, On Message = On*, dan lalu tekan centang hijau. Buat seperti ini dari 1 sampai 8. Lalu salin *API Credential*, pertamanya klik akun user (dipojok kanan atas) > *API Credential*, lalu salin Token.

Atur kabel sesuai dengan gambar 2. Lalu gunakan program

<https://github.com/techiesms/16-Appliances-Home-Automation>. Ubah WIFISSID dan PASSWORD pada program, sesuai dengan wifi dirumah. Ubah token, dengan menggunakan token dari API Ubidots, pastikan nama pin dan tombol program sama persis di void *executed_cases()* dibagian *digital write()*. Lalu upload program ke ESP32, setelah itu lakukan pengetesan, dapat dicek menggunakan serial monitor. Jika terjadi masalah seperti tidak berfungsi, pastikan internet terhubung dan gunakan serial monitor pada Arduino IDE untuk mendeteksi apakah ESP32 dan ubidots terhubung. Jika semua tombol berfungsi (terhubung dengan ESP32) maka hasil proyek telah berhasil.

Simpulan

IoT merupakan teknologi baru yang mulai diterapkan secara nyata, dengan komunikasi menggunakan MQTT, mempermudah pengguna untuk menpengendalian ruang kerja melalui jarak jauh yang terbatas pada selama pengguna dan IoT terhubung internet. Teknologi ini memberikan kemampuan untuk dapat mengendalikan pengguna energi ruangan, tanpa harus berada di ruangan tersebut bahkan berada dibelahan dunia lain. Hal ini memberikan kemampuan untuk melakukan manajemen mikro kantor. Teori yang ada, dikumpulkan melalui wawancara, jurnal dan buku. Kesimpulan yang dapat diambil adalah, teknologi IoT merupakan teknologi yang mengalami pengembangan pesat satu dekade ini, tidak akan lama lagi sebelum IoT mulai diterapkan secara massal di masyarakat, mempelajari teknologi ini cukup penting untuk pendalaman pengetahuan tentang teknologi ini.

Daftar Pustaka

- Barolli, L., Hellinckx, P., & Natwichai, J. (2019). Advances on P2P, Parallel, Grid, Cloud and Internet Computing. In *Lecture Notes in Networks and Systems*.
- Delsing, J. (2017). *IoT Automation: Arrowhead Framework*. CRC Press.
- Giacobbe, M., Pellegrino, G., Scarpa, M., & Puliafito, A. (2018). An Approach to Implement The “Smart Office” Idea: The #SmartMe Energy System. *Journal of Ambient Intelligence and Humanized Computing*, 9(Mei), 1–19.
- Kaushik, M., & Guleria, N. (2020). The Impact of Pandemic COVID -19 in Workplace. *European Journal of Business and Management*, May, 8–18.
- Ko, E. Y., Hong, S. H., & Cha, J. sang. (2020). *A Study on Remote IoT Operating Time for Fire Detector of Smart Home*. 20(2), 235–238.
- Kylili, A., Afxentiou, N., Georgiou, L., Panteli, C., Morsink-Georgalli, P. Z., Panayidou, A., Papouis, C., & Fokaidis, P. A. (2020). The Role of Remote Working in Smart Cities: Lessons Learnt from COVID-19 Pandemic. *Energy Sources, Part A: Recovery, Utilization and Environmental Effects*, 1–16.
- Onoda, H. (2020). Smart Approaches to Waste Management for Post-COVID-19 Smart Cities in Japan. *IET Smart Cities*, 2(2), 89–94.
- Pathan, A., & Deval, N. (2020). *IOT Based Smart Office Using Wireless sensor Area Network*. 10(4), 35–44.
- Tuzcuoglu, D., Yang, D., Vries, B. de, & Sungur, A. (2020). Social Interaction in An Office Environment : A Qualitative Study After Relocation to A Smart Office. *Transdisciplinary Workplace Research Conference 2020*, 1–10.
- Wazid, M., Das, A. K., Bhat K, V., & Vasilakos, A. V. (2020). LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment. *Journal of Network and Computer Applications*, 150.