

**IMPLEMENTASI DAN ANALISIS PERFORMANSI LAMPU PINTAR
BERBASIS *INTERNET OF THINGS* DENGAN MEDIA KONTROL
APLIKASI ANDROID**

***IMPLEMENTATION AND ANALYSIS OF SMART LAMP BASED ON
INTERNET OF THINGS USING ANDROID APPLICATION AS MEDIA
CONTROL***

Ghifary Aldo Riyadi¹, Dr. Ir. Rendy Munadi, M.T.², Sussi, S.Si., M.T.³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹ghifaryaldo@student.telkomuniversity.ac.id, ²rendymunadi@telkomuniversity.ac.id,
³sussiss@telkomuniversity.ac.id

Abstrak

Perkembangan teknologi khususnya telekomunikasi setiap waktunya berkembang sangat pesat, *Internet of Things* (IoT) hadir untuk memudahkan manusia dalam melakukan kegiatannya. Dalam penelitian ini, dilakukan sebuah perancangan sistem lampu pintar menggunakan mikrokontroler yang telah terintegrasi dengan *chip wireless*. Sistem ini menciptakan teknologi *remote controlling* dengan jaringan internet sebagai penghubung antara *server* dan *client* menggunakan protokol HTTP. Sistem ini didesain untuk semua kalangan agar dapat menghemat listrik melalui kontrol lampu secara otomatis sesuai kebutuhan. Dari hasil pengujian sistem lampu pintar dapat berfungsi dengan baik. Serta pengguna berhasil mengontrol lampu secara otomatis maupun manual menggunakan aplikasi android. Dari analisis *delay* waktu pengukuran sensor menunjukkan sensor ultrasonik dan sensor cahaya dapat bekerja dengan baik. Dari hasil pengujian *end to end* didapatkan *delay* terendah terdapat pada mode sensor ultrasonik dengan posisi lampu menyala ke mati. Pada pengujian dengan jaringan serat optik dari mikrokontroler ke server didapatkan *delay* terendah pada mode tombol on, dan dari aplikasi android ke server *delay* terendah pada mode sensor ultrasonik saat kondisi lampu menyala. Pada pengujian dengan jaringan 4G dari mikrokontroler ke server didapatkan *delay* terendah pada mode tombol off, dan dari aplikasi android ke server *delay* terendah pada mode tombol off.

Kata Kunci: IoT, HTTP, mikrokontroler, sensor, lampu.

Abstract

The development of technology every time is growing very rapidly especially telecommunication, Internet of Things (IoT) is here to make easier human activity. In this study, smart lamp system was designed using a microcontroller that has been integrated with a wireless chip. This system creates remote controlling technology with an internet network as a connection between server and client using the HTTP protocol. This system is designed for all people to save electrical energy using automatically lamp control. From the testing this system can work well. And the user can control the lamp automatically or manually using android application. From the delay analysis ultrasonic and light sensor can work well. On network testing using fiber optic from microcontroller to server on button has the lowest delay, and from android application to server ultrasonic sensor when the lamp turn on has the lowest delay. On 4G network testing from microcontroller to server off button has the lowest delay, and from android application to server off button has the lowest delay.

Key Word: IoT, HTTP, microcontroller, sensor, lamp.

1. Pendahuluan

Teknologi semakin hari semakin berkembang dengan pesat secara terus menerus, tak terkecuali dalam bidang telekomunikasi. Internet yang sekarang menjadi kebutuhan umat manusia pun semakin dikembangkan untuk menunjang dan mempermudah kehidupan manusia, salah satunya dalam bidang *Internet of Things (IoT)*. Penggunaan energi yang boros dapat mempengaruhi bumi yang ditempati manusia. Isu tentang perubahan iklim, degradasi lingkungan, semakin banyak dibahas karena dapat mengancam kehidupan makhluk yang ada di bumi[1]. Maka dari itu penelitian ini membahas tentang lampu pintar yang dapat mengurangi pemborosan energi sehingga dapat mengurangi dampak buruk yang ditimbulkan manusia dalam menggunakan energi secara berlebihan. Menurut penelitian yang dilakukan oleh Lucio Cibattoni, Alessandro Freddi, Gianluca Ippoliti, Maurizio Marcantonio, Davide Marchei, Andrea Monteriù, dan Matteo Pirro yang berjudul *A Smart Lighting System for Industrial and Domestic Use* bahwa penggunaan lampu pintar dengan sensor tertentu dapat mengurangi penggunaan energi sebesar 50% [2].

Pada penelitian paper yang berjudul *An IOT by Information Retrieval approach: Smart Lights controlled using WiFi* yang dilakukan oleh Navjot Kaur Walia, Parul Karla, Deepti Mehrotra menggunakan koneksi WiFi dan aplikasi android/pc namun terdapat kekurangan dimana mereka tidak memasukkan sensor kedalam perangkatnya, sehingga lampu pintar hanya dapat dikendalikan secara manual[3].

2. Dasar Teori dan Perancangan

2.1 Internet of Things (IoT)

Internet of Things menjadi salah satu teknologi yang hangat diperbincangkan dan terus dikembangkan. IoT adalah sebuah perangkat cerdas yang saling terhubung dengan suatu sistem yang membuat mesin pintar saling berinteraksi dan berkomunikasi dengan mesin lain, lingkungan, infrastruktur, sensor, maupun *radio frequency identification (RFID)*[4].

2.2 ESP32

ESP32 adalah sebuah mikrokontroler yang dilengkapi dengan modul Wifi dan Bluetooth. CPU yang digunakan adalah Xtensa Dual-Core 32-bit LX6 dengan ROM sebesar 128KB dan SRAM 416KB. Daya yang dibutuhkan ESP32 sebesar 2,5V sampai dengan 3,6V. Untuk keamanan IEEE 802.11 dapat menggunakan WFA, WPA/WPA2 dan WAPI, dilengkapi dengan 2 pin input sensor 3.3 V dan 5 V, 2 pin digital, dan 15 pin analog[5].

2.3 Sensor Cahaya

Sensor cahaya terdiri dari lapisan semi konduktor aktif yang diendapkan pada substrat isolasi. Cara kerja sensor cahaya ketika ada cahaya jatuh ke sensor cahaya maka foton yang merupakan partikel terkecil cahaya akan menembus lapisan semikonduktor tipe N (katoda) dan memasuki lapisan konduktor tipe P (anoda). Foton tersebut kemudian bertabrakan dengan elektron elektron yang terikat sehingga elektron terpisah dari intinya dan terbentuklah *hole*. Pemisahan antara muatan positif dan negatif menyebabkan perbedaan potensial pada persimpangan PN, ketika dihubungkan suatu beban ke katoda dan anoda maka elektron akan mengalir melalui beban atau kabel dari katoda ke anoda atau yang sering disebut aliran listrik.

2.4 Sensor Ultrasonik

Sensor ultrasonik digunakan dalam tugas akhir ini sebagai bagian dari salah satu fitur yang akan ada pada lampu pintar. Sensor ini memiliki pemancar dengan simbol T dan penerima dengan simbol R. Sensor Ultrasonik memerlukan 4 pin yaitu VCC sebagai sumber arus listrik, ground, trig, dan echo. Cara kerja dari sensor ini adalah pemancar memancarkan gelombang suara kearah depan, jika ada objek yang terkena gelombang suara tersebut maka gelombang suara akan terpantul dan diterima kembali oleh bagian penerima dari sensor[6].

2.5 Relay

Relay adalah komponen elektronika berupa saklar elektrik yang digunakan untuk menyambung atau memutuskan arus listrik yang mengalir melwatinya. Relay terdiri dari 4 bagian penting yaitu koil, armature, saklar, *iron core*. Relay memiliki 2 kontak poin yaitu: *normally close* dan *normally open*.

2.6 ThingSpeak

ThingSpeak adalah sebuah layanan *platform* IoT bersifat *open source* yang dapat digunakan untuk mengumpulkan, menampilkan, dan menganalisa data secara langsung dalam suatu *cloud*. Beberapa kemampuan dari ThingSpeak adalah:

1. Dapat menampilkan data sensor secara *real-time*.
2. Menjalankan analisa IoT secara otomatis berdasarkan jadwal yang telah disusun.
3. Dapat membuat purwarupa dan sistem IoT tanpa harus mengatur server atau membuat *web software*[7].

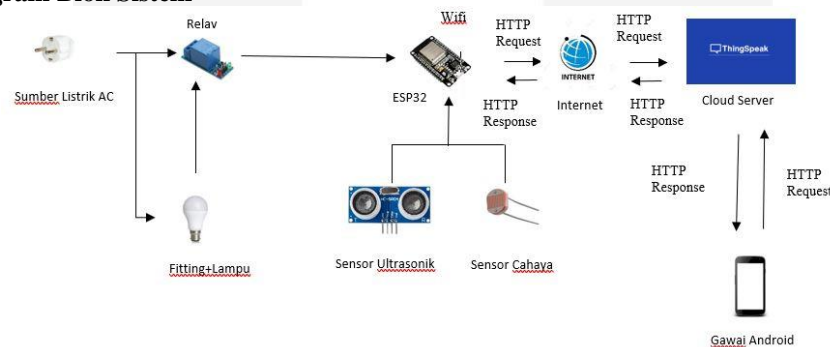
2.7 Android

Android adalah sistem operasi buatan Google yang bersifat *open source* berbasis Linux untuk perangkat layar sentuh seperti *smartphone* atau komputer tablet[8]. Kode yang bersifat *open source* dan lisensi perizinan pada Android memungkinkan untuk memodifikasi perangkat lunak secara bebas dan didistribusikan dengan mudah. Fitur yang disediakan android diantaranya antarmuka, aplikasi, dan pengelolaan memori[9].

2.8 Hyper Text Transfer Protocol

Hypertext Transfer Protocol (HTTP) adalah sebuah protokol jaringan pada lapis aplikasi yang digunakan untuk sistem sistem informasi yang terdistribusi, kolaboratif dan hypermedia. Pada tahun 1990 HTTP digunakan pada World-Wide Web secara global. HTTP dapat berfungsi sebagai protokol untuk menghubungkan klien, server dan perangkat IoT. HTTP bekerja dengan melakukan pertukaran *request* dan *response* antara server dan klien[10].

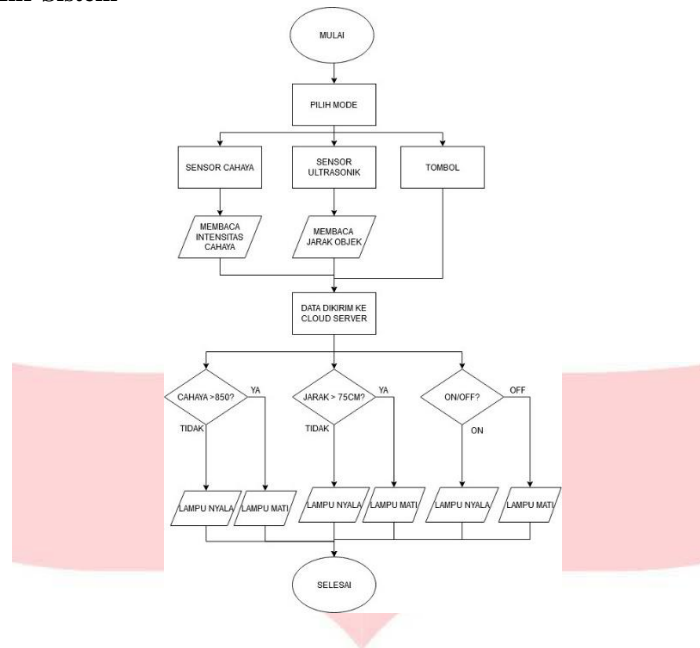
2.9 Diagram Blok Sistem



Gambar 2.1 Diagram Blok Sistem.

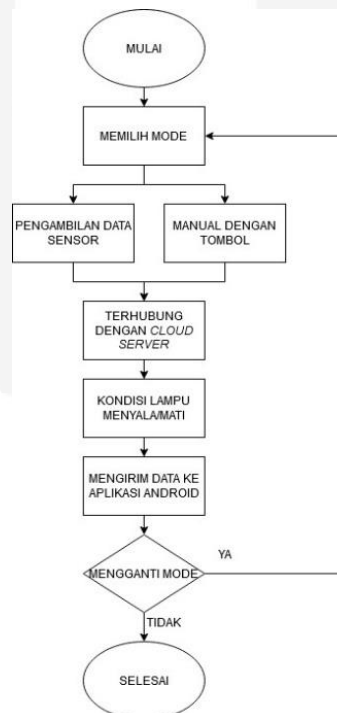
Pada gambar 2.1 menunjukkan diagram blok lampu pintar menggunakan aplikasi android. Sumber listrik AC tersambung dengan relay dan lampu, kemudian relay terhubung dengan ESP32 melalui pin digital, 5V, dan GND. Sensor cahaya dan sensor ultrasonik sebagai input untuk mengetahui kondisi cahaya, masukan dan jarak objek dengan alat, sensor-sensor tersebut terhubung dengan ESP32 sesuai dengan kebutuhan pinnya masing masing. Kemudian data yang diperoleh sensor-sensor tersebut dikirim ke ThingSpeak yang berfungsi sebagai *cloud server* dengan menggunakan protokol HTTP, dari ThingSpeak data diteruskan ke gawai android. Sensor melakukan HTTP request untuk mengirim data ke ThingSpeak, kemudian ThingSpeak mengirim data ke gawai menggunakan HTTP response. Sedangkan ketika gawai mengirim permintaan ke ThingSpeak maka dilakukan HTTP Request dari gawai ke ThingSpeak, untuk mengirimkan perintah dari ThingSpeak ke lampu maka gawai juga dapat melakukan HTTP response.

2.10 Diagram Alir Sistem



Gambar 2.2 Diagram alir sistem.

Berdasarkan gambar 2.2 dimana sensor-sensor pada mikrokontroler yang telah terkoneksi dengan internet mulai membaca kondisi lingkungan. Sensor-sensor mulai membaca data dan akan diteruskan ke *cloud server* jika intensitas cahaya kurang dari 850 maka lampu akan menyala otomatis, jika intensitas cahaya lebih dari 850 maka lampu akan mati secara otomatis. Jika jarak objek kurang dari 125cm maka lampu akan menyala, dan jika jarak objek lebih dari 75 cm maka lampu akan mati secara otomatis. Untuk pengoprasian manual menggunakan tombol, jika tombol on aktif maka lampu akan menyala dan jika tombol off aktif maka lampu akan mati.



Gambar 2.3 Diagram alir pengiriman data ke aplikasi android.

Berdasarkan gambar 2.3 proses kerja diawali dengan memilih mode yang akan digunakan, kemudian sensor atau tombol akan bekerja sesuai dengan mode yang dipilih dan mengirimkan data ke *cloud server*. Setelah data sampai ke *cloud server* kondisi lampu akan mati/menyala sesuai dengan mode yang dipilih dan nilai pembacaan oleh sensor. Data yang sudah ada pada *cloud server* dapat diambil dan ditampilkan pada aplikasi android, jika ingin mengganti mode maka dapat diakses melalui aplikasi android dan proses akan berulang kembali setelah pemilihan mode.

3. Pembahasan

3.1 Pengujian Fungsional

3.1.1 Pengujian Sensor

Dari hasil pengujian sensor ultrasonik dapat membaca jarak dengan konstan pada jarak 25 cm, 75 cm, dan 125 cm. Dengan kondisi lampu sebagai berikut:

Tabel 3.1 Hasil Uji Sensor Ultrasonik.

Jarak	Kondisi Lampu
25 cm	Menyala
75 cm	Mati
125 cm	Mati

Dari hasil pengujian sensor cahaya tidak membaca intensitas cahaya secara konstan pada kondisi cahaya yang sama, namun hal tersebut tidak berpengaruh terhadap sistem karena kesalahan membaca intensitas cahaya tidak melampaui *threshold* yang sudah ditetapkan yaitu 850. Dengan kondisi lampu sebagai berikut:

Tabel 3.2 Hasil Uji Sensor Cahaya.

Kondisi	Kondisi Lampu
Diatas <i>threshold</i>	Mati
Dibawah <i>threshold</i>	Menyala

3.1.2 Pengujian Aplikasi Android

Dari hasil pengujian menu yang ada pada aplikasi android, semua fitur dapat berjalan dengan baik, hasil analisis kinerja aplikasi android dirangkum dalam tabel berikut:

Tabel 3.3 Hasil Uji Aplikasi Android.

Fitur Aplikasi	Yang dilakukan aplikasi	Keterangan
<i>Button</i> TOMBOL	Menjadikan mikrokontroler mode Tombol.	Dapat bekerja dengan baik.
<i>Button</i> ON	Memberikan perintah ke mikrokontroler untuk menutup sirkuit elektrik melalui relay.	Dapat bekerja dengan baik.
<i>Button</i> OFF	Memberikan perintah ke mikrokontroler untuk membuka sirkuit elektrik melalui relay.	Dapat bekerja dengan baik.
<i>Button</i> SENSOR CAHAYA	Menjadikan mikrokontroler menjadi mode Sensor Cahaya sehingga relay akan menutup/membuka sirkuit elektrik sesuai dengan intensitas cahaya.	Dapat bekerja dengan baik.
<i>Button</i> REFRESH	Mengambil data intensitas cahaya dari <i>cloud server</i> ThingSpeak.	Dapat bekerja dengan baik.

<i>Button</i> SENSOR ULTRASONIK	Menjadikan mikrokontroler mode Sensor Ultrasonik sehingga relay akan menutup/membuka sirkuit elektrik sesuai dengan jarak objek.	Dapat bekerja dengan baik.
<i>Button</i> REFRESH	Mengambil data jarak objek dari <i>cloud server</i> ThingSpeak.	Dapat bekerja dengan baik.

3.1.3 Pengujian Cloud Server

Cloud server dapat berjalan dengan semestinya karena setelah *cloud server* mampu menerima data sensor yang dikirimkan dari mikrokontroler ESP32 melalui jaringan internet. Dan *cloud server* juga mampu menyediakan data yang akan diminta oleh aplikasi android untuk ditampilkan. Data yang dapat ditampilkan pada aplikasi android adalah Jarak Objek yang didapat dari sensor ultrasonik dan Intensitas Cahaya yang didapat dari sensor cahaya.

3.2 Pengujian Sistem End to End

Tabel 3.4 Hasil Uji Delay Sistem End to End.

Mode	Rata-rata Delay
Tombol On	2,16 s
Tombol Off	2,28 s
Sensor Cahaya (on ke off)	3,87 s
Sensor Cahaya (off ke on)	2,71 s
Sensor Ultrasonik (on ke off)	2,2 s
Sensor Ultrasonik (off ke on)	2,05 s

Berdasarkan tabel 3.4 dapat diketahui bahwa mode Sensor Ultrasonik ketika kondisi lampu menyala ke mati memiliki *delay* terendah, dan mode Sensor Cahaya ketika kondisi lampu menyala ke mati memiliki *delay* paling tinggi.

3.3 Pengujian Dengan Jaringan Wifi Serat Optik

Tabel 3.5 Hasil Jaringan Wifi Serat Optik.

Mode	ESP32 ke/dari ThingSpeak		Aplikasi Android ke/dari ThingSpeak	
	Delay	Throughput	Delay	Throughput
Tombol On	0,35 s	7,485 kbps	1,31 s	1,668 kbps
Tombol Off	0,36 s	7,401 kbps	0,69 s	2,691 kbps
Sensor Cahaya On	0,47 s	4,631 kbps	0,61 s	4,462 kbps
Sensor Cahaya Off	0,47 s	4,682 kbps	0,67 s	3,462 kbps
Sensor Ultrasonik On	0,49 s	4,463 kbps	0,34 s	7,127 kbps
Sensor Ultrasonik Off	0,49 s	4,488 kbps	0,54 s	5,280 kbps

Berdasarkan tabel 3.5 dapat diketahui pada pengujian aliran data menggunakan jaringan wifi serat optik antara ESP32 dengan ThingSpeak mode Tombol On memiliki *delay* terendah dan *throughput* tertinggi, sedangkan pada pengujian delay aliran data antara aplikasi android dan ThingSpeak mode Sensor Ultrasonik dengan kondisi lampu menyala memiliki *delay* terendah dengan *throughput* yang paling tinggi.

3.4 Pengujian Dengan Jaringan Wifi 4G

Tabel 3.6 Hasil Uji Jaringan Wifi 4G.

Mode	ESP32 ke/dari ThingSpeak		Aplikasi Android ke/dari ThingSpeak	
	Delay	Throughput	Delay	Throughput

Tombol On	0,38 s	8,097 kbps	0,25 s	7,005 kbps
Tombol Off	0,37 s	8,151 kbps	0,19 s	9,119 kbps
Sensor Cahaya On	0,51 s	6,474 kbps	0,60 s	4,362 kbps
Sensor Cahaya Off	0,80 s	4,038 kbps	0,78 s	3,583 kbps
Sensor Ultrasonik On	0,55 s	4,558 kbps	1,24 s	2,402 kbps
Sensor Ultrasonik Off	0,55 s	4,691 kbps	1,03 s	2,606 kbps

Berdasarkan tabel 3.6 dapat diketahui pada pengujian aliran data menggunakan jaringan wifi 4G antara ESP32 dengan ThingSpeak mode Tombol Off memiliki *delay* terendah dan *throughput* tertinggi, sedangkan pada pengujian delay aliran data antara aplikasi android dan ThingSpeak mode Tombol On memiliki *delay* terendah dengan *throughput* yang paling tinggi.

4. Kesimpulan

1. Desain pada sistem ini steker, relay dan fitting saling terhubung dengan kabel tembaga. Sensor-sensor dan relay terhubung dengan mikrokontroler ESP32 menggunakan kabel jumper. Protokol HTTP digunakan untuk pertukaran data antara mikrokontroler ESP32 dengan *cloud server* ThingSpeak dan antara aplikasi android dengan *cloud server* ThingSpeak.
2. Aplikasi android dengan fitur mode Sensor Ultrasonik dan mode Sensor Cahaya digunakan untuk mengontrol lampu menjadi mode otomatis sehingga lampu akan menyala atau mati sesuai dengan kondisi lingkungan. Mode Tombol digunakan untuk mengontrol lampu secara manual. Data yang dikirimkan dari aplikasi android dikirimkan ke *cloud server* ThingSpeak kemudian diteruskan ke mikrokontroler ESP32.
3. Sensor ultrasonik dan sensor cahaya dapat mengambil dan mengirimkan data ke mikrokontroler. Aplikasi android dapat digunakan untuk mengirimkan perintah ke mikrokontroler dan menampilkan data yang telah diambil oleh sensor-sensor. *Cloud server* dapat menampung data yang dikirimkan oleh mikrokontroler. Rata-rata *delay end to end* pada setiap mode tidak melebihi 26,5 s. Rata-rata *delay* jaringan pada tiap mode tidak melebihi 3,24 s dan *throughput* jaringan pada tiap mode tidak kurang dari 0,16 kbps.

Daftar Pustaka

- [1] Y. Tian, Z. Yu, N. Zhao, Y. Zhu, and R. Xia, "Optimized Operation of Multiple Energy Interconnection Network Based on Energy Utilization Rate and Global Energy Consumption Ratio," *2nd IEEE Conf. Energy Internet Energy Syst. Integr. EI2 2018 - Proc.*, pp. 1–6, 2018.
- [2] L. Ciabattini *et al.*, "A smart lighting system for industrial and domestic use," *2013 IEEE Int. Conf. Mechatronics, ICM 2013*, pp. 126–131, 2013.
- [3] N. K. Walia, P. Kalra, and D. Mehrotra, "An IOT by information retrieval approach: Smart lights controlled using WiFi," *Proc. 2016 6th Int. Conf. - Cloud Syst. Big Data Eng. Conflu. 2016*, pp. 708–712, 2016.
- [4] P. V. Dudhe, N. V. Kadam, R. M. Hushangabade, and M. S. Deshmukh, "Internet of Things (IOT): An overview and its applications," *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017*, pp. 2650–2653, 2018.
- [5] Espressif System IOT Team, "ESP32 Specification," p. 46, 2015.
- [6] P. Marian, "HC-SR04 Datasheet," 2015. [Online]. Available: <https://www.electroschematics.com/hc-sr04-datasheet/>. [Accessed: 09-May-2020].
- [7] "About ThingSpeak." [Online]. Available: <https://thingspeak.com/>. [Accessed: 29-Sep-2019].

- [8] Open Handset Alliance, "Android Overview." [Online]. Available: http://www.openhandsetalliance.com/android_overview.html.
- [9] Touch Devices." [Online]. Available: <https://source.android.com/devices/input/touch-devices>.
- [10] T. B.-L. R. Fielding, J. Gettys, J.C Mogul, H. Frystyk, L. Masinter, P. Leach, "Hypertext Transfer Protocol--HTTP/1.1," *TUT Text. a Usages Tech.*, vol. 3, no. First quarter, pp. 1–114, 1999.

