

PERANCANGAN DAN IMPLEMENTASI *PROGRAMMABLE LED DISPLAY* BERBASIS MIKROKONTROLER

DESIGN AND IMPLEMENTATION OF PROGRAMMABLE LED DISPLAY BASED ON MICROCONTROLLER

Danu yahya¹, Husneni Mukhtar, S.Si., M.T., Ph.D.², Dr.Eng. Ahmad Sugiana, S.Si., M.T.³

1,2,3 Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom
danuyahya@student.telkomuniversity.ac.id,
husneni.mukhtar@telkomuniversity.ac.id, ahmad.sugiana@gmail.com

Abstrak

Informasi merupakan kebutuhan penting bagi masyarakat di zaman yang semakin maju ini. Media informasi memiliki bentuk yang beragam mulai dari yang konvensional hingga yang non konvensional. LED *display* merupakan salah satu media informasi yang banyak digunakan karena lebih menarik untuk dilihat. Pada dasarnya untuk dapat bekerja LED *display* membutuhkan sebuah mikroprosesor yang berfungsi untuk mengoperasikan data. Penggunaan mikroprosesor dirasa tidak efisien karena harganya yang tidak murah. Selain itu kontrol untuk mengoperasikan LED *display* biasa dilakukan dengan cara mengakses secara langsung dan menggunakan perangkat tambahan seperti *keyboard* ataupun *flashdisk* yang menyusahakan bagi operator jika LED *display* berada pada tempat yang tinggi. LED *display* banyak digunakan untuk *running text* yang hanya menampilkan huruf dan angka yang membuatnya monoton.

Hasil dari Tugas Akhir ini adalah sistem LED *display* yang dapat dikontrol secara *wireless* melalui *website*. Jarak yang dapat dilakukan untuk mengontrol LED *display* mencapai 45 m dengan ataupun tanpa penghalang. LED *display* dapat menganimasikan 2 buah gambar dengan penggunaan energi listrik untuk kedua jenis animasi yang disimulasikan adalah sekitar (2,9 – 3,0)Wh untuk tingkat kecerahan maksimum. Kontrol dari sistem LED *display* menggunakan mikrokontroler yang berfungsi untuk mengoperasikan data. Pengaturan yang dapat dilakukan adalah pilihan animasi, kecerahan animasi, kecepatan animasi, serta pilihan teks dan jam.

Kata Kunci : Mikrokontroler, LED *display*, *Running text*

Abstract

Information is an important need for society in this increasingly advanced era. Information media has various forms, from conventional to non-conventional. LED display is one of the most widely used information media because it is more attractive to look at. Basically, to work, LED displays require a microprocessor that functions to operate data. The use of microprocessors is considered inefficient because of the low price. In addition, control for operating the LED display is usually done by directly accessing and using additional devices such as a keyboard or flash drive which is difficult for the operator if the LED display is at a high place. LED displays are widely used for running text which only displays letters and numbers which make it monotonous.

The result of this final project is an LED display system that can be controlled wirelessly via a website. The distance that can be taken to control the LED display is up to 45 m with or without obstructions. The LED display can animate 2 images with the use of electrical energy for the two types of animation simulated is around (2.9 - 3.0) Wh for the maximum brightness level. Control of the LED display system uses a microcontroller which functions to operate the data. Settings that can be made are animation options, animation brightness, animation speed, and text and clock options.

Keywords: Microcontroller, LED display, running text

1. Pendahuluan

Informasi secara umum dapat diartikan sebagai sekumpulan data atau fakta yang telah diproses dan dikelola sedemikian rupa sehingga menjadi sesuatu yang dapat dengan mudah dimengerti dan bermanfaat bagi penerimanya. Tidak setiap data atau fakta dikatakan sebagai sebuah informasi walaupun sudah diproses, jika data atau fakta tersebut tidak bermanfaat bagi penerimanya maka tidak dapat dikatakan sebagai sebuah informasi. Informasi sendiri datang dari bahasa latin, yaitu "*Informatinem*" yang memiliki arti ide, kode, atau garis besar. Dengan kemajuan zaman yang semakin pesat, kebutuhan akan informasi di masyarakat sangat tinggi. Masyarakat sangat membutuhkan informasi untuk menunjang kehidupan di zaman ini untuk mengetahui kejadian yang nyata terjadi dan digunakan untuk mengambil keputusan. Media informasi memiliki berbagai ragam seperti: televisi, radio, lampu neon yang dibentuk huruf spanduk, baliho, papan matriks LED *display*, dan lainnya.

LED *display* merupakan salah satu media informasi yang banyak digunakan karena lebih menarik. LED *display* menggunakan sebuah mikroprosesor sebagai otak untuk dapat memproses data sehingga dapat menampilkan tulisan pada LED *display*. Mikroprosesor yang digunakan untuk sistem LED *display* memiliki harga yang tidak murah. LED *display* biasanya digunakan sebagai *running text* yang menampilkan huruf atau angka. Penggunaan LED *display* pada umumnya diletakkan pada tempat yang tinggi agar mudah terlihat. Untuk dapat memperbaharui tampilan LED *display* maka operator harus mengakses sistem secara langsung menggunakan *keyboard* maupun menggunakan media *flashdisk*.

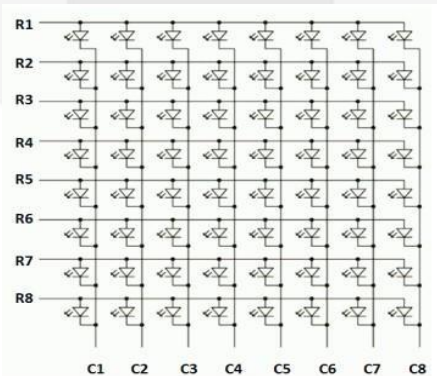
Mikroprosesor dapat digantikan tugasnya oleh sebuah mikrokontroler yang memiliki fungsi yang sama namun dengan harga yang relatif lebih murah serta memiliki kemampuan untuk berkomunikasi secara nirkabel, hal ini dapat mempermudah operator untuk mengganti tampilan pada LED *display* tanpa perlu mengakses secara langsung. Dengan menggunakan teknologi Wi-Fi maka penggantian tampilan pada LED *display* dapat dilakukan melalui *website* yang di dalamnya terdapat pengaturan untuk tampilan pada LED *display*. Dengan tambahan fitur animasi yang dapat ditampilkan maka membuat LED *display* lebih menarik.

Penelitian ini menghasilkan purwarupa LED *display* yang dapat dikontrol secara nirkabel melalui *website* untuk menampilkan animasi sederhana. Dengan penggunaan pengontrol yang murah diharapkan dapat lebih efisien diaplikasikan di berbagai kebutuhan seperti LED *display* pada pabrik, petunjuk arah jalan, digital baliho dan lainlain.

2. Tinjauan Pustaka

2.1 RGB LED *Display*

Light emitting diode (LED) merupakan semikonduktor yang dapat mengubah energi listrik menjadi cahaya, berupa perangkat keras dan padat (*solid-state component*) sehingga unggul dalam ketahanannya (*durability*)[1]. LED *display* atau LED matriks merupakan kumpulan dari beberapa LED yang disusun membentuk baris dan kolom (Gambar 1). Dengan kumpulan beberapa LED yang telah disusun sedemikian rupa, maka dibutuhkan banyak pin untuk dapat mengontrolnya. Contohnya 8x8 LED matriks (Gambar 1) membutuhkan 64 pin I/O. Dengan menyusun semua anode pada baris (R1-R8), dan katoda pada kolom (C1-C8), dapat mengurangi penggunaan pin menjadi 16. Untuk penggunaan yang lebih besar seperti RGB LED matriks 16x32 yang memiliki 512 LED yang tersusun dalam baris dan kolom menjadinya memiliki lebih banyak pin I/O, oleh sebab itu dibutuhkan *shift register* yang dapat mengurangi penggunaan pin dari LED matriks. *Shift register* sendiri merupakan sebuah perangkat yang berfungsi untuk menyimpan data sementara dan menggeser data tersebut. Dalam *shift register* terdapat pin *clock* yang berfungsi untuk menerima data ketika diberi nilai *high* atau *low*, pin data untuk menyimpan data yang diberikan, dan pin *latch* yang berfungsi untuk mengeluarkan data tersebut.



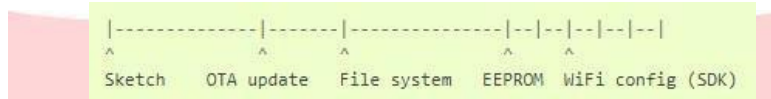
Gambar 1 8x8 LED matriks

2.2 Websocket

Websocket merupakan sebuah protokol yang dikembangkan oleh HTML5 dengan penerapan komunikasi *fulldublex* antara *server* dengan *client*. Penggunaan protokol yang satu ini merupakan standar baru pada komunikasi *real time* pada *web* dikarenakan protokol ini dapat mengurangi lalu lintas jaringan menjadi lebih ringan dibandingkan dengan metode *polling* dan *long-polling* yang sebelumnya telah menggunakan koneksi dua arah dengan menjaga dua koneksi tetap terhubung. Mekanisme kerja *websocket* sendiri adalah dengan dimulainya proses *client* dan *server* melakukan *handshake* ketika proses tersebut berhasil maka protokol dapat mengirim dan menerima data secara *dual channel*.

2.3 SPIFFS (Serial Peripheral Interface Flash File System)

Dalam sebuah mikrokontroler terdapat memori atau *penyimpanan* yang digunakan untuk menyimpan data program yang telah dibuat. Dalam penyimpanan itu sendiri terdapat beberapa jenis data penyimpanan dengan ukuran tertentu (Gambar 2).



Gambar 2 Alokasi penyimpanan

Ketika menyimpan sebuah program yang telah dibuat, maka program tersebut akan dialokasikan ke penyimpanan *sketch*. Sedangkan kapasitas dari penyimpanan *sketch* terbatas. Namun dengan menggunakan SPIFFS maka dapat mengakses dan menyimpan *file* dengan kapasitas yang lebih besar, di mana *file* tersebut dapat diletakkan pada penyimpanan *file system*. Penyimpanan *sketch* dan *file system* dapat digunakan secara bersamaan dan tidak mengganggu satu sama lain.

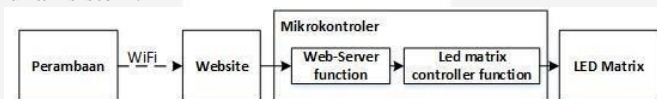
3. Perancangan Sistem

3.1 Prinsip Kerja Sistem

Prinsip kerja sistem ini secara umum adalah ketika seorang *user* dapat mengatur tampilan pada LED *display* melalui *website* menggunakan *smartphone/computer* yang terdapat fitur Wi-Fi di dalamnya. *User* harus berada dalam satu jaringan yang sama dengan LED *display*, lalu membuka aplikasi peramban pada *smartphone/computer* yang digunakan untuk menuju ke alamat *website* yang di dalamnya terdapat beberapa pilihan seperti gambar, kecerahan, teks dan *clock* yang dapat diatur dan setelah dipilih maka pilihan-pilihan tersebut akan tampil pada LED *display*.

3.2 Diagram Blok Sistem

Gambar 3 menunjukkan *diagram* blok dari sistem ini, dalam diagram blok menjelaskan komponen modul penyusun keseluruhan sistem.



Gambar 3 Diagram blok sistem

3.3 Fungsi dan Fitur

Terdapat beberapa fungsi dan fitur di dalam LED *display* sistem ini di antaranya:

1. Peramban yang berfungsi sebagai media untuk mengakses alamat *website*, akses peramban dapat melalui perangkat komputer maupun *Smartphone* yang terhubung dalam satu jaringan dengan sistem.
2. *Website* adalah sebuah antarmuka yang disediakan untuk memilih beberapa pengaturan yang tersedia, fitur input yang terdapat di dalamnya seperti menampilkan animasi, menampilkan teks dan *clock*, pengaturan *delay* dan kecerahan LED matriks.
3. Mikrokontroler di sini berperan sebagai kontroler LED *display* dan sebagai web-server.
4. LED *display* sebagai *aktuator* di mana menampilkan semua pengaturan yang telah ditentukan.

3.4 Desain Sistem

Terdapat 2 desain sistem yang digunakan, yaitu desain perangkat lunak dan desain perangkat keras. Untuk desain perangkat lunak diprogram menggunakan *Arduino IDE*, sedangkan untuk desain perangkat keras mencakup perancangan catu daya, sinyal, dan mekanik.

3.4.1 Perangkat Lunak

A. Perancangan Data Gambar

Untuk dapat menampilkan animasi berupa gambar dalam LED *display* maka gambar tersebut harus di ubah ke dalam bentuk kode warna untuk setiap *pixel* pada gambar tersebut. Dalam hal ini harus menggunakan

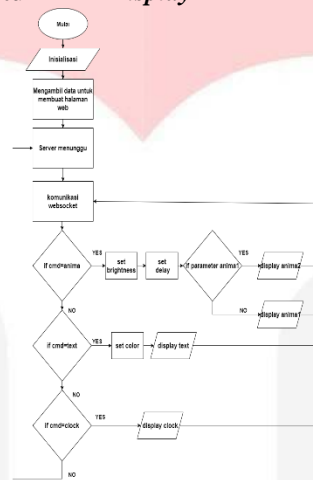
gambar berukuran 26×32 *pixel* yang nantinya akan di *convert* menjadi kode warna. Berikut adalah tahap-tahap untuk mengubah gambar untuk menghasilkan kode warna.

1. Siapkan gambar dengan ukuran 26×32 *pixel* .
2. Buka http://www.rinkydinkelectronics.com/t_imageconverter565.php untuk mengubah gambar yang kita miliki menjadi bentuk kode warna
3. Unduh hasil konversi tadi yang berformat .c
4. Buka file tersebut dan copy kode warna yang berada di dalam tanda kurung { }
5. Buka aplikasi note dan paste kode warna tersebut.
6. Lalu simpan file tersebut
7. Lakukan hal yang sama untuk gambar berikutnya.

Setelah mendapatkan kode warna dari setiap gambar maka hal berikutnya adalah menyimpan *file* tersebut ke dalam *file system* bersamaan dengan data *file* halaman *website*. Untuk dapat menyimpan *file* tersebut ke dalam *file system* maka dibutuhkan *SPIFFS*. Cara penggunaan *SPIFFS* akan dijelaskan di bawah. 1. Simpan *file server* dan *file* gambar pada 1 folder dan beri nama data

2. Buka aplikasi Arduino IDE
3. Sambungkan mikrokontroler dengan PC dan sesuaikan port pada Arduino IDE
4. Pada Arduino IDE buka tab *tools*>*sketch data upload*

B. Perancangan Program Kontrol Untuk LED Display

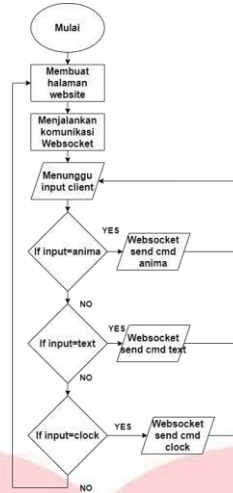


Gambar 4 *Flowchart* pengontrol

Pertama ketika sistem ini dinyalakan maka sistem akan menginisialisasi variabel, *websocket*, *Filesystem*, dan pustaka *LED display*. Lalu sistem akan mendeklarasikan fungsi-fungsi yang ada seperti menyambungkan dengan *wifi* yang telah diberikan, dan juga mengambil data untuk tampilan *website* pada *filesistem* melalui *SPIFFS* serta ketika terhubung maka *server* memberikan *IP address* yang akan terlihat pada *LED display*. Lalu proses selanjutnya adalah *server* menunggu adanya komunikasi antara *client* menggunakan komunikasi *websocket*, Ketika *server* telah terhubung dengan *client*. Ketika perintah yang diberikan oleh *webservice* adalah *anima* maka proses selanjutnya adalah parameter kecerahan lalu parameter kecepatan, jika parameter adalah *anima1* maka akan ditampilkan *anima1* namun jika bukan maka akan ditampilkan *anima2*. Jika perintah bukan *anima* melainkan teks maka prosesnya adalah parameter warna lalu akan ditampilkan teks sesuai dengan parameter. Namun jika perintah adalah *clock* maka akan ditampilkan *clock* dan jika bukan maka proses akan kembali pada *serverhandclient*.

C. Perancangan Webservice

Untuk *perancangan webservice* di perhatikan pada gambar 7 yang menjelaskan bagaimana *webservice* ini bekerja.

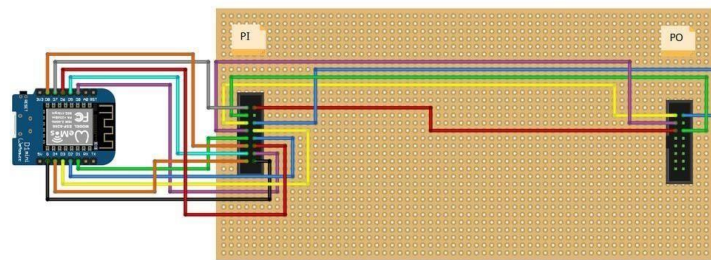


Gambar 5 Flowchart webservice

Pertama ketika sistem dimulai maka sistem akan mengambil data berupa tampilan yang akan di berikan pada *website*, lalu sistem menjalankan *websocket* dan mengambil data yang diberikan oleh *client*. Jika data yang diberikan oleh *client* berupa *anima* maka *websocket* akan memberikan perintah *anima* dan parameternya dan proses kembali untuk menunggu input dari *client*, namun jika tidak maka proses berikutnya adalah jika data yang diberikan adalah teks maka *websocket* memberikan perintah teks dan parameternya dan akan menunggu input dari *client*, namun jika datanya bukan teks maka proses selanjutnya jika data yang diberikan berupa *clock* maka *websocket* memberikan perintah *clock* namun jika datanya bukan *clock* maka proses akan kembali ke komunikasi *websocket*.

3.4.2 Perangkat Keras

Perancangan perangkat keras terdiri dari mikrokontroler dan LED *display* yang menggunakan sumber 5 volt DC. Modul LED *display* yang digunakan berukuran 32x64 *pixel* di mana memiliki 2048 titik. LED *display* harus di berikan sumber 5 volt DC untuk dapat bekerja dengan baik.



Gambar 6 Pengkabelan pin mikrokontroler dan LED display

Tabel 1 Pin mikrokontroler dan LED display

Pin mikrokontroler ke PI LED display		Pin PI dan PO pada LED display	
PI	ESP8266	PI	PO
A	D1	R1	R0
B	D2	G0	R1
C	D8	G1	G0
D	D6	B0	G1
STB/LAT	D0		
P_OE	D4		
CLK	D5		
R0	D7		

Gambar 8 menjelaskan pengkabelan pada mikrokontroler dan LED *display*, di mana terdapat PI dan PO pada LED *display*. Mikrokontroler terhubung pada PI LED *display*, sedangkan untuk PO LED *display* terhubung pada PI LED *display* yang ditunjukkan oleh tabel 1.

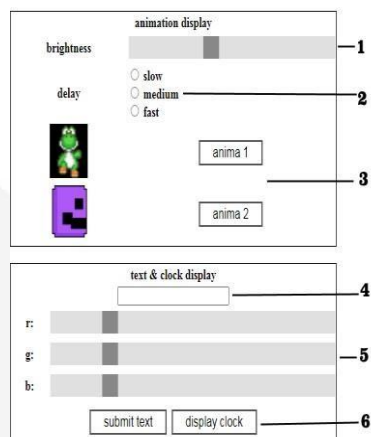
4. Hasil Dan Analisis

4.1 Sistem Programmable LED display

Pada sub-bab ini akan ditampilkan hasil dari sistem *programmable* LED *display* yang terdiri dari perangkat keras LED matriks *display* Gambar 7 dan tampilan *website* Gambar 8. Pada awal sistem dihidupkan maka sistem akan mengakses jaringan *WiFi* yang nama dan kata sandinya telah dideklarasikan pada program. Kemudian LED *display* akan menampilkan IP *address* yang digunakan untuk mengakses *website* yang di dalamnya terdapat beberapa pilihan sebagai *input* yang digunakan mikrokontroler.



Gambar 7 Hardware LED matriks *display* ukuran 32x64 *pixel* dengan total LED 2048 buah

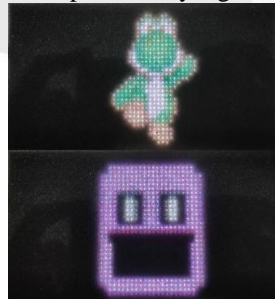


Gambar 8 Tampilan kontroler pada *website* (1)Pengaturan kecerahan tampilan animasi, (2)Pengaturan kecepatan tampilan animasi, (3)Dua pilihan animasi yang dapat digunakan, (4)Kotak input teks, (5)Pengaturan warna RGB, (6)Pengaturan tampilan teks atau *clock*

Untuk menampilkan animasi maka harus mengatur kecerahan dan kecepatan yang di inginkan lalu memilih animasi yang ingin di tampilkan. Untuk menampilkan teks dapat memilih warna yang dapat di atur nilai R, G dan B dan untuk *clock* akan menampilkan jam yang diambil dari internet.

4.2 Spesifikasi Karakter Animasi

Pada Tugas Akhir ini, telah diprogram dua bentuk animasi yang digunakan. Kedua animasi seperti yang ditampilkan pada Gambar 9 dan memiliki spesifikasi yang tertera pada Tabel 3



Gambar 9 Dua bentuk pilihan animasi yang digunakan pada tampilan *display*.

Tabel 3 Spesifikasi animasi 1 dan animasi 2

No.	Hal	Anima 1	Anima 2
1	Jumlah LED yang digunakan	332 - 365	448 – 585
2	Warna LED yang digunakan	Putih, hijau, coklat	Magenta, putih
3	Komposisi warna RGB	Putih(255,255,255) Hijau(56,184,56) Coklat(80,56,40)	Magenta(176,87,248) Putih(255,255,255)

4.3 Pengujian Waktu Pengiriman Dan Penerimaan Data

Tujuan pengujian ini adalah untuk mengetahui kualitas jaringan pada saat waktu pengiriman dan penerimaan data pada sistem ini. Pengujian dilakukan dengan cara melihat waktu ketika data diterima oleh mikrokontroler menggunakan *Arduino IDE* dan aplikasi *chrome* untuk melihat waktu saat *client* mengirimkan data. Hasil dari pengujian ini dapat dilihat pada Gambar 12, di mana waktu *delay* rata-rata data dikirimkan dari *client* melalui *website* dan diterima oleh *server* adalah 93,5ms, dengan waktu *delay* tercepat dan terlambat adalah 45 ms dan 165 ms.



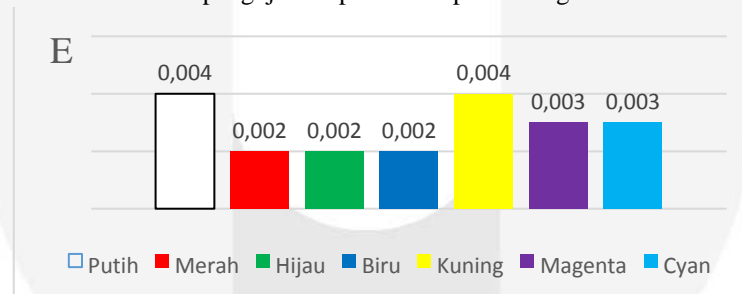
Gambar 10 Grafik waktu pengiriman data dari *client* ke *server*(waktu *delay*)

4.4 Pengujian Energi Listrik

4.4.1 Pengujian Energi Listrik LED Berdasarkan Warna

Pengujian ini dilakukan untuk mengetahui berapa energi listrik yang dibutuhkan oleh 1 LED. Energi listrik diukur menggunakan *watt*-meter, dan pengujian dilakukan pada setiap warna. Semua LED yang ada pada sistem dinyalakan dalam warna yang sama, sehingga dapat diketahui konsumsi energi listrik rata-rata

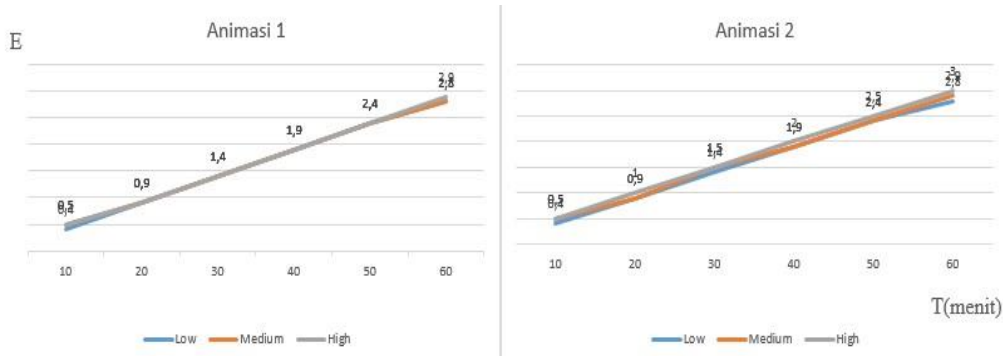
w untuk 1 LED dalam 1 jam, atau energi listrik, yaitu $W_{1LED} = \dots$. Hasil dari pengujian dapat dilihat pada 2048 gambar 11.



Gambar 11 Energi listrik rata-rata yang dibutuhkan oleh 1 LED dalam 1 jam untuk masing-masing 7 warna

4.4.2 Pengujian Energi Listrik Untuk Animasi 1 Dan Animasi 2

Pengujian ini menggunakan alat *watt* meter yang diposisikan di antara *power supply* dan sistem LED *display*. Pengukuran dilakukan setiap 10 menit selama 1 jam untuk animasi 1 dan animasi 2 sebanyak 3 kali di mana animasi diatur pada kecerahan rendah, sedang, dan tinggi. Hasil dari pengujian kedua animasi tersebut dapat dilihat pada Gambar 12.



Gambar 12 Grafik konsumsi energi listrik animasi 1 dan animasi 2 berdasarkan tingkat kecerahan (*low, medium, dan high*)

Penggunaan energi listrik untuk kedua jenis animasi terlihat cenderung sama dalam waktu yang sama juga. Perbedaan tingkat kecerahan tidak mempengaruhi konsumsi energi listrik secara signifikan, hanya terjadi perbedaan yang sangat sedikit, yaitu sekitar (0,0 - 0,1) watt untuk animasi 1 dan (0,1 – 0,2) watt untuk animasi 2 setiap 10 menit. Konsumsi energi listrik kedua animasi selama 1 jam adalah hanya sekitar 3 watt. Dengan demikian, penggunaan energi listrik untuk kedua animasi ini tergolong kecil dan irit sehingga penggunaan energi listrik pada sistem ini dapat dikatakan efisien.

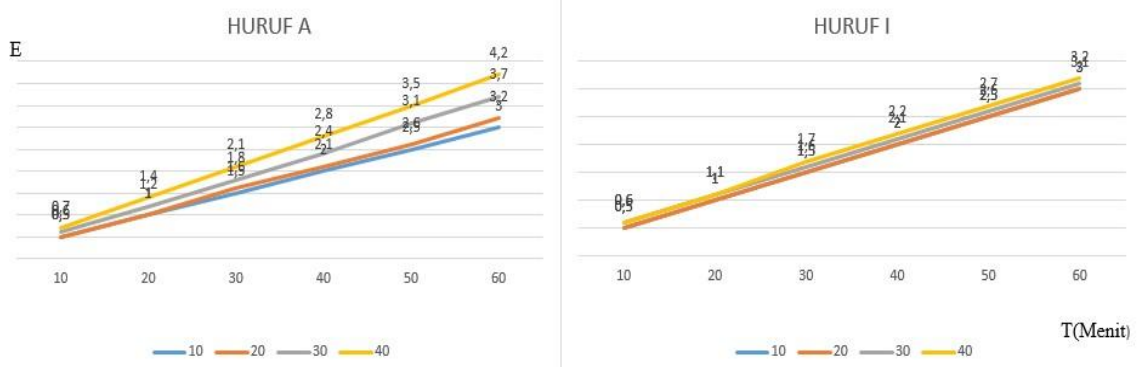
4.4.3 Pengujian Energi Listrik Untuk Tampilan Tulisan Berdasarkan Jumlah LED Yang Hidup

Tujuan pengujian ini adalah untuk mengetahui konsumsi energi listrik LED *display* jika hanya menampilkan teks. Pengujian dilakukan dengan cara melihat penggunaan energi listrik pada *wattmeter* setiap ± 10 menit selama 1 jam dengan menampilkan huruf A dan I di mana untuk spesifikasinya terdapat pada tabel 4. Huruf yang ditampilkan menggunakan warna putih dan tidak ada animasi.

Tabel 4 Spesifikasi jumlah LED untuk huruf A dan I

Jumlah huruf A dan I	Jumlah LED hidup huruf A	Jumlah LED hidup huruf I
10	160	110
20	320	220
30	480	330
40	640	440

Hasil pengujian terlihat pada Gambar 4.8. Terlihat bahwa pada saat menampilkan huruf A sebanyak 40 karakter yang mengaktifkan 640 LED menggunakan energi listrik hingga 4,2Wh sedangkan untuk huruf I yang mengaktifkan 440 LED menggunakan 3,2Wh

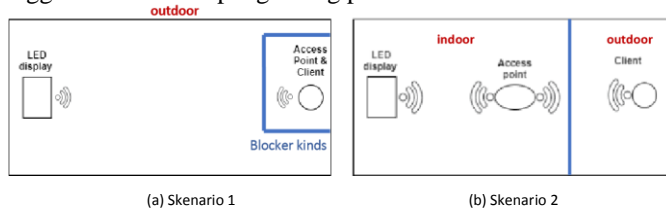


Gambar 13 Konsumsi energi listrik ketika menampilkan karakter huruf

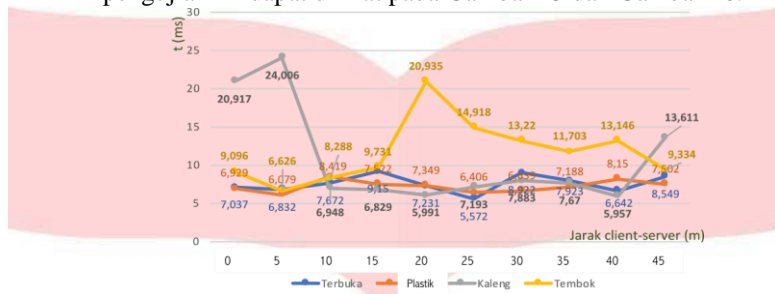
4.5 Pengujian Waktu Respon Server Pada LED Display Dengan Kontroler Berdasarkan Jarak

Pengujian ini bertujuan untuk mengetahui waktu respon *server* saat melakukan transmisi data berdasarkan jarak *client* dan *server*. Pengujian dilakukan dengan 2 skenario, pertama adalah AP (*access point*) dan *client* adalah dari satu perangkat, yaitu *handphone* (Gambar 14a). Skenario kedua adalah menjadikan *router* sebagai

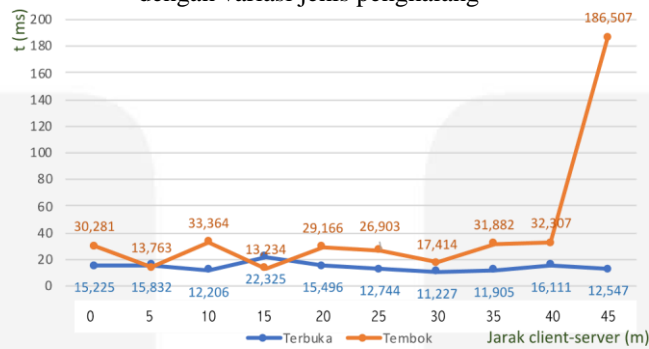
AP dan *handphone* sebagai *client* (Gambar 14b). Pengujian dilakukan pada selasar Gedung N FTE Universitas Telkom dengan variasi penggunaan *blocker/* penghalang pada *client*.



Gambar 14 Konfigurasi pengujian waktu pengiriman data dengan dua skenario. Terdapat *blocker* pada pengiriman data dari *client* ke *server* pada masing-masing skenario. (a) Pengujian pada outdoor, *client* terhalang oleh plastik, kaleng, dan tembok. (b) Pengujian dengan *client* berada di luar ruangan. Hasil dari pengujian ini dapat dilihat pada Gambar 15 dan Gambar 16.



Gambar 15 Pengujian waktu respon *server* terhadap kontroler (*client*) untuk skenario 1 berdasarkan jarak dengan variasi jenis penghalang



Gambar 16 Pengujian waktu respon *server* terhadap kontrol (*client*) untuk skenario 2 berdasarkan jarak dengan variasi jenis penghalang

Pengujian nilai waktu respon *server* terhadap kontrol untuk skenario kedua, yaitu *client* berada di luar ruangan dan AP di dalam ruangan, ternyata tidak berpengaruh signifikan sampai dengan jarak 40 m. Waktu respon *server* mulai melambat pada saat pengujian di jarak 45 m, namun pengiriman data tetap berhasil dilakukan. Untuk lebih lengkapnya, Tabel 5 dan Tabel 6 menampilkan waktu respon *server* terhadap *client*.

Tabel 5 Waktu respon server-client untuk skenario pertama, jarak 0-45 m.

Jenis penghalang	Waktu respon <i>server</i> (ms)		
	t_{min}	t_{max}	$(\bar{t} \pm \Delta\bar{t})$
Tidak ada (udara terbuka)	5,572	9,150	$(7,553 \pm 1,116)$
Plastik	6,079	8,150	$(7,2183 \pm 0,736)$
Kaleng	5,957	24,006	$(10,700 \pm 6,609)$
Tembok/ Bangunan	6,626	20,935	$(11,699 \pm 4,132)$

Tabel 6 Waktu respon server-client untuk skenario kedua.

	Waktu respon <i>server</i> (ms)

Jenis penghalang dan jarak <i>clientserver</i>	t_{min}	t_{max}	$(\bar{t} \pm \Delta\bar{t})$
Tidak ada, jarak (0-45) m	11,227	22,325	$14,561 \pm 3,278$
Tembok, jarak (0-40) m	13,234	33,364	$41,482 \pm 8,220$
Tembok, jarak 45 m	$t = 186, 507 \text{ ms}$		

Waktu respon *server* lebih stabil untuk skenario kedua (*router* sebagai AP dan *handphone* sebagai *client* atau *controller*), yaitu pada kondisi tanpa penghalang sejauh 45 m dan dengan penghalang sejauh 40 m.

- 5. Kesimpulan**
1. Sistem dapat menampilkan 2 animasi sederhana yang telah ditentukan serta teks dengan konsumsi daya yang tergolong kecil di mana untuk animasi (2,9 – 3,0) Wh dan untuk teks (3,0 – 3,5) Wh
 2. Sistem dapat dikontrol secara *wireless* menggunakan Wi-Fi melalui *website* yang terdapat pengaturan untuk animasi seperti pilihan animasi, kecerahan animasi, kecepatan animasi, teks dan jam.

Daftar Pustaka:

- [1]S. Diding, “Prototipe *controller* lampu penerangan LED (Light Emitting Diode) *independent* bertenaga surya”, GAMMA, vol 10, no 1, pp. 166-122, 2014
- [2]Sujarwo, “Proses sortir karet menggunakan konveyor berbasis mikrokontroler”, Teknik elektro, Universitas Jember, Jember, 2016
- [3]PM. Nishad, C. R.Manicka, “Various colour spaces and colour spaces conversion algorithms” Jurnal of Global Research in Computer Science, Vol 4, No.1, pp. 44-48, 2013
- [4]Khamdani, “Rancang bangun running-text LED display berbasis mikrokontroler sebagai media informasi di Universitas Muhammadiyah Purwokerto”, Teknik Elektro, Universitas Muhammadiyah Purwokerto, Purwokerto, 2014
- [5]S. Marco. 2016. Internet of Things with ESP8266. Birmingham:Packt Publishing
- [6]W, Vanessa. Frank Salim. Peter Moskovits. 2013. The Definitive Guide to HTML5 WebSocket. New York City:Apress Media
- [7]Lengstorf, Jason. Phil Leggetter. Alex Newman. 2013. Realtime Web Apps. New York City:Apress
- [8]Poole, Nick. 2018. “Everything You Didn’t Want to Know About RGB Matrix Panels”. <https://www.sparkfun.com/news/2650>. 15 November 2019 pukul 23:01.
- [9]P, Pieter. 2017. ”A Beginner’s Guide to the ESP8266”. <https://tttapa.github.io/ESP8266/Chap11%20%20SPIFFS.html>. 20 November 2019 pukul 2:15.
- [10]Grokhotkov, Ivan. 2017. ”Welcome to ESP8266 Arduino Core’s documentation!”. [online]. <https://arduino-esp8266.readthedocs.io/en/latest/index.html>. 20 November 2019 pukul 3:30.
- [11]R-B. ”LAB 12:BASICS OF LED DOT MATRIX DISPLAY”. [online]. <http://embedded-lab.com/blog/lab12-basics-of-LED-dot-matrix-display/>. 20 November 2019 pukul 5:00
- [12]Burgess, Phillip. ”How the Matrix Works”. [online] <https://learn.adafruit.com/32x16-32x32-rgb-LEDmatrix/how-the-matrix-works>. 29 November 2019 pukul 21:00