

PERANCANGAN 3D PRINTER DAN IMPLEMENTASI IOT MENGGUNAKAN SMARTPHONE ANDROID SEBAGAI HOST SERVER

Bravelyan Edgar Randista Ellyandrew*¹⁾, Nina Setiyawati²⁾

1. Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Indonesia

2. Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Indonesia

Article Info

Kata Kunci: *3D printer*; internet of things; *Octoprint*; *smartphone*

Keywords: *3D printer*; internet of things; *Octoprint*; *smartphone*

Article history:

Received 8 Agustus 2024

Revised 21 September 2024

Accepted 1 Oktober 2024

Available online 1 September 2025

DOI :

<https://doi.org/10.29100/jipi.v10i3.6354>

* Corresponding author.

Corresponding Author

E-mail address:

672020081@student.uksw.edu

ABSTRAK

Dalam era digital ini internet of things (IoT) telah mengubah cara berinteraksi dengan perangkat dan sistem sehari-hari. Salah satu aplikasi menarik dari internet of things adalah pada *3D printer*. *3D printing* adalah teknologi manufaktur aditif yang memungkinkan pembuatan objek tiga dimensi dengan lapisan demi lapisan, memungkinkan kreativitas tanpa batas dan inovasi cepat. Meskipun memiliki banyak keunggulan seperti efisiensi produksi dan minimnya limbah, *3D printing* menghadapi tantangan biaya tinggi dalam pembuatan mesin. Pada penelitian ini, penerapan internet of things pada *3D printer* dilakukan dengan menggunakan *smartphone* Android menggunakan aplikasi open source *octo4a* untuk menjalankan *Octoprint*. Ini menawarkan solusi portabel dan hemat biaya dibandingkan penggunaan komputer. Penelitian ini mengintegrasikan penggunaan *smartphone* untuk mengurangi biaya dan mempermudah aksesibilitas. Hasil penelitian menunjukkan bahwa penggunaan *smartphone* Android sebagai *host server* berhasil mempermudah pengoperasian *3D printer* secara Internet of things. Dengan aplikasi *octo4a smartphone* dapat menjalankan *Klipper* dan *Octoprint*, serta berfungsi sebagai webcam untuk memantau proses cetak. Pendekatan ini menawarkan alternatif yang lebih ekonomis dan mudah diakses, menjadikan *smartphone* sebagai pilihan yang efektif untuk mengendalikan *3D printer*.

ABSTRACT

In this digital era, the Internet of Things (IoT) has revolutionized the way we interact with devices and everyday systems. One interesting application of IoT is in *3D printing*. *3D printing* is an additive manufacturing technology that enables the creation of three-dimensional objects layer by layer, allowing for limitless creativity and rapid innovation. Despite many advantages such as production efficiency and minimal waste, *3D printing* faces high costs in machine construction. This study explores the implementation of IoT in *3D printers* using an Android smartphone with the open-source *Octo4a* application to run *Octoprint*. This offers a portable and cost-effective solution compared to using a computer. The research integrates the use of smartphones to reduce costs and enhance accessibility. The results show that using an Android smartphone as a host server successfully facilitates the operation of a *3D printer* via IoT. With the *Octo4a* application, the smartphone can run *Klipper* and *Octoprint*, and also function as a webcam to monitor the printing process. This approach provides a more economical and accessible alternative, making smartphones an effective choice for controlling *3D printers*.

I. PENDAHULUAN

INTERNET of Things (IoT) telah mengubah cara berinteraksi dengan perangkat dan sistem sehari-hari. Salah satu aplikasi menarik dari IoT adalah pada *3D printer*, yang memungkinkan pengguna untuk mengontrol dan memantau proses cetak melalui *smartphone* dan komputer.

Pengertian *3D Printing* adalah mesin manufaktur aditif dalam paradigma membuat desain produk, prototipe pada industri manufaktur yang ada pada saat ini. Teknologi ini memungkinkan pembuatan objek tiga dimensi dengan lapisan demi lapisan, memungkinkan kreativitas tanpa batas dan inovasi yang cepat [1][2]. Menurut Arianti *3D printing* adalah proses pencetakan desain digital menjadi benda padat tiga dimensi diproduksi oleh mesin *3D printer* [3]. *3D printer* bisa mencetak pola rumit dengan tahapan proses yang dibuat menggunakan komputer dengan software *Computer Aided Engineering (CAD)* [4]. *3D printer* memiliki keunggulan sebagai mesin yang dapat dioperasikan secara otomatis, dimana fungsi tersebut dapat memangkas biaya dan memiliki optimalisasi produksi serta efisiensi waktu dalam produksinya [5]. Namun biaya yang tinggi dalam membangun mesin *3D printer* menjadi hambatan [6].

3D printer adalah perangkat yang menciptakan objek padat dengan volume dan bentuk tiga dimensi dari desain digital. Berbeda dengan mesin dengan proses subtraktif yang mereduksi bahan melalui pemotongan, pengeboran, dan penyayatan *3D printer* tidak menghasilkan limbah produksi. Proses pencetakan objek 3D dengan *3D printer* dimulai dengan mengonversi desain ke objek format *file* yang sesuai dengan firmware yang digunakan, yaitu G-Code [7]. Dalam *3D printer*, yang menjadi elemen utama adalah *controller*. *Controller* dalam *3D printer* memainkan peran vital dalam mempengaruhi kualitas dan efisiensi hasil cetak *3D printer*. *Controller* dalam *3D printer* terbagi menjadi dua yaitu *open-loop controller* dan *closed-loop controller*. *Close-loop controller* lebih sederhana dan lebih hemat biaya, sementara *open-loop controller* memberikan presisi dan pengulangan yang lebih baik karena memungkinkan perubahan perintah berdasarkan pada data yang diterima oleh sensor [8].

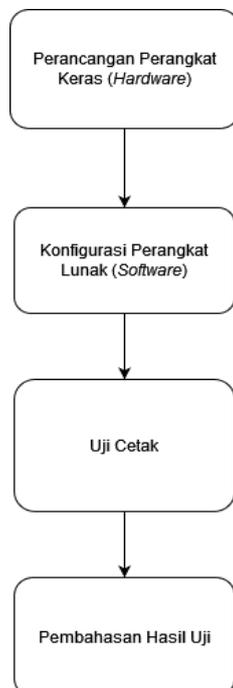
Internet of things (IoT) adalah suatu sistem, terdapat perangkat komputer dan internet serta perangkat mekanisasi secara digital yang terhubung dan terintegrasi satu sama lain [9]. Menurut Reni (2022) *Internet of things* sistem yang terintegrasi satu sama lain, dengan cepat dapat mengirimkan informasi dari mesin melalui internet dan ditampilkan oleh perangkat komputer sebagai *host control* pada mesin yang bekerja secara mekanis [10]. Penggunaan IoT dapat digunakan pada 3d print, dengan mempermudah akses koneksi dimana saja dengan kedua perangkat terkoneksi pada internet menggunakan perantara aplikasi pihak ketiga [14]. Octo4a adalah aplikasi yang dibuat oleh pengguna github bernama "feelfreelinux" yang memungkinkan pengguna untuk menjalankan *Octoprint* di perangkat Android yang terhubung menggunakan OTG. Aplikasi ini dirancang untuk memberikan solusi yang mudah dan portabel untuk mengontrol *3D printer* menggunakan *smartphone* Android tanpa memerlukan komputer [11].

Terdapat beberapa penelitian yang menjadi acuan dalam penelitian ini. Penelitian pertama oleh Reza yang berjudul "Nonlinear Material Modeling for Mechanical Characterization of 3D printed PLA Polymer With Different Infill Densities" menggunakan Raspberry pi sebagai *controller* untuk menjalankan firmware *Klipper* untuk meningkatkan akurasi pada kelengkungan. Penelitian ini tidak menggunakan perangkat *controller* Android. Pada penelitian yang kedua, menurut Bambang (2023) bahwa penggunaan perangkat lunak berupa *Octoprint* dapat digunakan untuk menghubungkan dari perangkat komputer kepada mesin cetak 3 dimensi. Akses *Octoprint* dapat digunakan pada perangkat komputer berbasis sistem operasi linux, sehingga pengoperasian mesin 3 dimensi dapat diakses dengan mudah [13]. Pada penelitian ketiga yang berjudul "IoT based 3D printer", penelitian ini berisi pembuatan *3D printer* yang dapat dikendalikan secara remote dengan menggunakan sistem *Octoprint* berbasis IoT yang diinstal pada raspberry PI. Hasil dari penelitian ini membuktikan bahwa penggunaan *Octoprint* memungkinkan pengoperasian *3D printer* secara nirkabel dan transfer *file* tanpa menggunakan *SD card* [14]. Pada penelitian keempat yang dilakukan oleh putu (2021) menambahkan IoT pada mesin 3D printer menggunakan perangkat Raspberry Pi. Implementasi IoT yang dilakukan pada mesin cetak *3D printer* meningkatkan waktu dan produktivitas yang lebih efisien dan optimal [15].

Berdasarkan pada penelitian terdahulu, dimana penggunaan perangkat host *server* yang bervariasi pada implementasi IoT terhadap *3D printer* sehingga hal tersebut dapat dijadikan gap analysis pada penelitian ini. penelitian ini penerapan IoT untuk *3D printer* pada menggunakan *smartphone* Android sebagai *server host* menggunakan aplikasi open source octo4a untuk menjalankan *Octoprint*, dimana hal tersebut menjadi keterbaruan dalam penelitian ini. Pada penelitian ini hanya berfokus pada perancangan *3D printer*, serta implementasi IoT pada *3D printer*. Kekurangan pada penelitian terdahulu tidak adanya hasil dan pembahasan pada tingkat akurasi dari hasil produksi atau pembuatan pengujian *3D printer*.

II. METODE PENELITIAN

Pada penelitian ini terdapat metode penelitian, tahapan dalam melakukan implementasi dapat dijabarkan pada Gambar 1



Gambar. 1. Metode penelitian

Metode penelitian yang dilakukan terdapat empat tahapan; 1) **Tahapan pertama** perancangan perangkat keras dilakukan yaitu, untuk melakukan persiapan guna untuk pengujian yang akan dilakukan dengan menggunakan perangkat yang tepat untuk pengujian; 2) **Tahapan kedua** konfigurasi perangkat lunak, dimana konfigurasi perangkat lunak memiliki fungsi untuk mendukung kelancaran dalam proses uji cetak dan kontrol terhadap perangkat keras secara penuh; 3) **Tahapan ketiga** uji cetak, dengan dilakukan uji cetak bertujuan untuk mendapatkan hasil uji yang dilakukan; 4) **Tahapan keempat** pembahasan hasil uji, pembahasan berfokus pada pengujian sistem IoT dilakukan dengan uji gerak secara remote dan melakukan analisis pada kualitas cetak secara IoT menggunakan metode Taguchi.

A. Spesifikasi Perangkat Keras 3D printer

Perancangan perangkat keras dari 3D printer mengikuti rancang bangun 3D printer REPRAP. Pada 3D printer ditambahkan *smartphone* Android yang dihubungkan ke Arduino Mega 2560 yang terhubung dengan RAMPS 1.6 untuk mengontrol kerja 3D printer. pemilihan perangkat keras dikarenakan umumnya ditemukan di pasaran. Pada spesifikasi pereangkat keras yang digunakan disajikan pada Tabel 1 sebagai berikut ini:

TABEL I
 SPESIFIKASI MESIN 3D PRINTER

Spesifikasi	KETERANGAN
Panjang ruang kerja	39cm
Lebar ruang kerja	39cm
Tinggi ruang kerja	800cm
Daya PSU	360watt (12v 30A)
Stepper sumbu gerak	Nema 17 1.6A moons 17HDB401-01N
Stepper extruder	Nema 17 1.4A moons 17HD2405-12NS
Hotend heater	Heater 12V, 40W
Sensor suhu	100K NTC B 3950 ±1%
Driver stepper	A4988
Host server	<i>Smartphone</i> Android

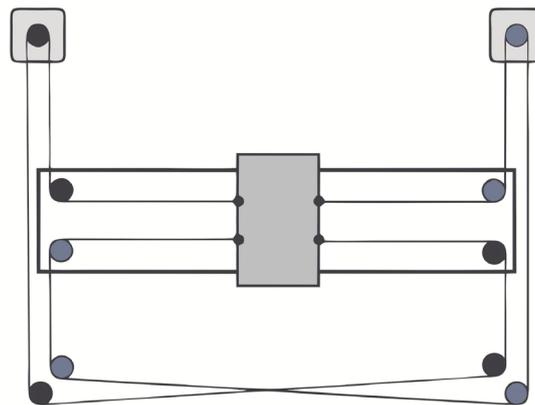
Spesifikasi perangkat keras *3D printer* menyesuaikan pada Tabel 1 digunakan dalam perancangan mesin *3D printer* dan penggunaan *host server* yang digunakan menggunakan perangkat *smartphone* Android. *3D printer* IoT memiliki integrasi perangkat keras yang saling berhubungan. Pada *smartphone* Android terinstal *octo4a* untuk menjalankan *Octoprint* sebagai *host server* dan firmware *Klipper* untuk memproses G-code yang di hubungkan pada arduino mega 2560 dan RAMPS 1.6 agar dapat mengendalikan dan menerima sinyal dari berbagai komponen seperti *motor stepper*, *sensor extruder*, *hotend*. Struktur mesin yang kokoh menyediakan fondasi yang stabil sementara motor stepper menggerakkan nozzle dan hotend sesuai instruksi yang diterjemahkan pada firmware *Klipper* dari perangkat lunak *slicing*. Extruder yang terhubung melelehkan *filament* melalui *hotend* yang suhunya di kendalikan untuk memastikan presisi dan kualitas cetakan. Sensor suhu dan *endstop* berperan dalam menjaga kondisi optimal selama pencetakan dan semua diatur melalui RAMPS 1.6.

B. Teknik Analisis

Teknik analisis menggunakan uji cetak menggunakan metode taguchi dengan mengukur akurasi dimensi, dengan menggunakan ukuran yang sudah ditentukan dengan ukuran 20x20x20 mm serta diukur melalui jangka sorong[20]. Pengujian dilakukan sebanyak sepuluh kali. Pengukuran keberhasilan dinilai dari hasil akurasi dimensional pada pengaturan kecepatan cetak yang berbeda. Indikator keberhasilan implementasi IoT pada *3D printer* dinilai dari pemantau an secara *real time* dan dapat dikontrol pada jarak jauh.

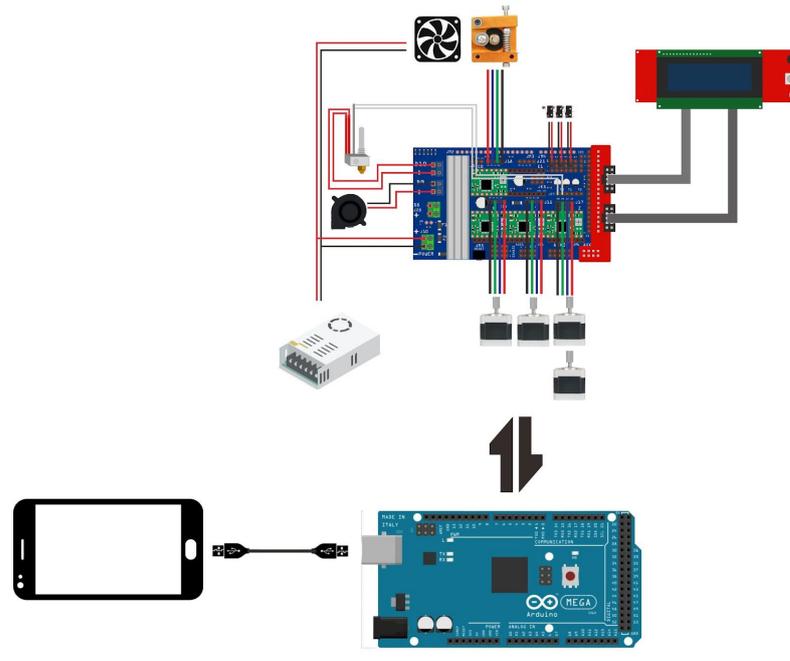
III. HASIL DAN PEMBAHASAN

A. Konfigurasi Perangkat Keras



Gambar. 2. Mekanisme CoreXY

Perancangan perangkat keras *3D printer* berbasis IoT menggunakan konfigurasi tipe *3D printer CoreXY*. Pemilihan desain *CoreXY* didasarkan pada pertimbangan bahwa konfigurasi ini menawarkan stabilitas yang lebih tinggi dibandingkan dengan desain lainnya. Menurut Yuris (2022), konfigurasi *CoreXY* mampu memberikan tingkat akurasi yang lebih baik dibandingkan dengan mesin cetak 3D yang menggunakan konfigurasi Cartesian. Dalam penelitian sebelumnya, konfigurasi *CoreXY* juga telah terbukti unggul dalam hal efisiensi ruang dan pengurangan biaya produksi selama proses pencetakan. Gambar 2 menampilkan tampilan konfigurasi dari *3D printer* berbasis *CoreXY*, yang menunjukkan bagaimana desain ini memaksimalkan penggunaan ruang sambil tetap mempertahankan akurasi dan kualitas cetakan yang optimal.

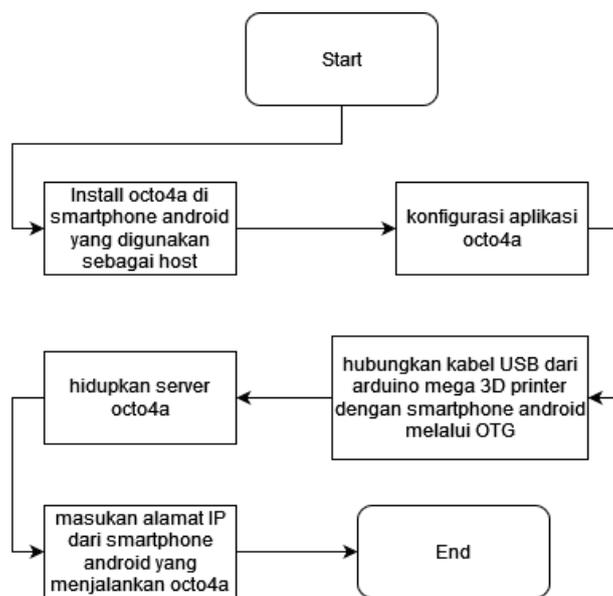


Gambar. 3. Konfigurasi Mesin Cetak 3 Dimensi

Pada Gambar 3. menunjukkan arsitektur bahwa perangkat pada mesin *3D printer* terhubung dengan RAMPS 1.6 dengan kendali Arduino Mega 2560. Perintah juga dimasukan melalui *smartphone* Android yang berperan sebagai *host server* yang terhubung ke Arduino Mega 2560 melalui USB OTG secara IoT.

Penggunaan perangkat keras REPRAP mempermudah perancangan sistem perangkat keras *3D printer*, karena didukung oleh banyaknya panduan dan dokumentasi yang tersedia. Panduan ini membantu pengguna dalam memahami proses perakitan, konfigurasi, dan pemecahan masalah, sehingga meminimalkan hambatan teknis dan mempercepat proses pengembangan. Dukungan komunitas yang luas juga memastikan bahwa informasi dan solusi untuk berbagai tantangan teknis mudah diakses, membuat implementasi dan modifikasi perangkat keras menjadi lebih efisien dan terstruktur.

B. Konfigurasi Perangkat lunak



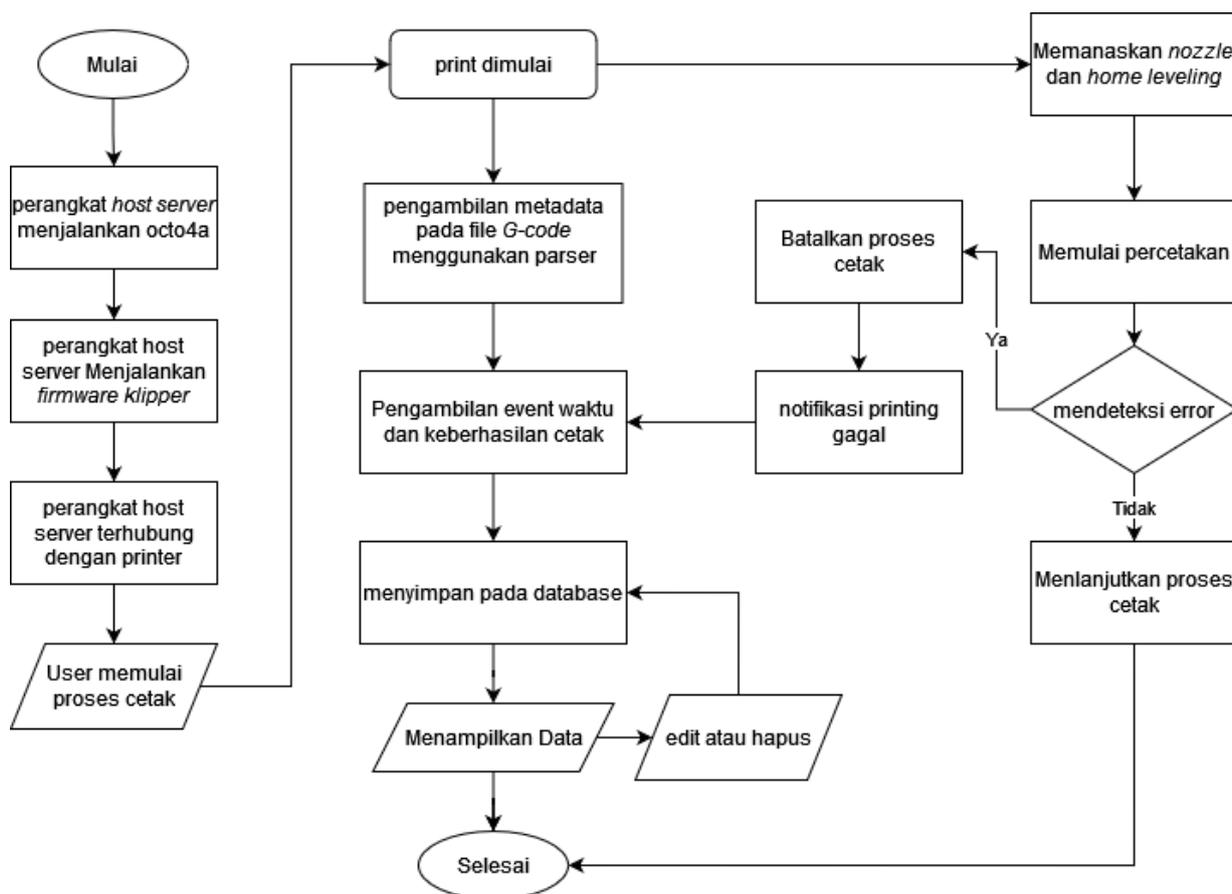
Gambar. 4. Konfigurasi Perangkat Lunak

Konfigurasi sistem IoT untuk *3D printer* menggunakan Aplikasi yang bernama Octo4a. Octo4a adalah aplikasi yang memungkinkan penggunaan *Octoprint* pada perangkat *Smartphone* Android. *Octoprint* adalah aplikasi

kontrol printer 3D *open source* yang dibuat oleh Gina Häußge untuk menyediakan berbagai fitur untuk mengendalikan, memonitor, dan mengelola proses pencetakan 3D melalui web antarmuka[18],[19]. Untuk mengakses *Octoprint* melalui internet perlu melakukan port forwarding melalui router sesuai dengan port yang tertera pada Octo4a. Dengan Octo4a, *smartphone* Android dapat digunakan sebagai *server Octoprint*, yang biasanya dijalankan di Raspberry Pi atau komputer lainnya. Gambar 4 menunjukkan diagram dari konfigurasi 3D Octo4a.

Perangkat *smartphone* Android ditambahkan pada mesin 3D *printer* yang dihubungkan ke kontroler melalui OTG. Pada firmware *Klipper 3D printer* melakukan proses komputasi pada perangkat eksternal sebagai host. *Smartphone* Android berperan sebagai host *server 3D printer* dapat dikendalikan secara IoT dengan antarmuka website dan perangkat pemrosesan untuk firmware *Klipper*. *Klipper* adalah firmware open source dan gratis yang dapat digunakan untuk mengkonfigurasi dan mengendalikan beberapa mikrokontroler. Selain itu, *Klipper* juga kompatibel dengan *Octoprint*. Sebagai hasilnya, pengendalian unit kendali FDM melalui *Octoprint* secara IoT menjadi mungkin. Firmware *Klipper* memiliki kemampuan yang canggih dan kinerja yang efisien dalam membantu mempercepat proses pencetakan tanpa mengurangi kualitas dan integritas produk[17].

instalasi dan penerapan firmware *Klipper* pada aplikasi octo4a melalui terminal ssh, pada terminal yang octo4a perlu melakukan penginstalan python menggunakan perintah ‘cd Octo* && pip3 install .’ dan pembaruan setup ‘tools’, ‘wheel’, dan ‘tornado’ menggunakan perintah ‘pip3 install --upgrade setuptools wheel tornado --no-cache-dir’, dan unduh setup octo4a menggunakan command ‘curl -s <https://raw.githubusercontent.com/feelinux/octo4a/master/scripts/setup-octo4a.sh> | bash -s’, setelah terunduh ekstrak menggunakan perintah ‘unzip *Octoprint.zip*’. untuk perintah instalasi *Klipper* pada octo4a menggunakan command ‘bash -c “\$(curl -fsSL <https://raw.githubusercontent.com/feelinux/octo4a/master/scripts/setup-Klipper.sh>)”’. Setelah *Klipper* sudah terinstal pada aplikasi octo4a pada antar muka web *Octoprint* perlu melakukan instalasi plugin bernama ‘*octoKlipper*’ pada pengaturan. Alur kerja dari 3D *printer* dapat dilihat pada Gambar 5.



Gambar. 5. Alur kerja

Untuk menyimpan riwayat pencetakan 3D *printer* dilakukan pembuatan *plugin history job*. Pada proses pengambilan data diambil dari *file G-code* yang diproses menggunakan parser dan penanganan event cetak yang dipicu *Octoprint* seperti PRINT_DONE, PRINT_FAILED, dan METADATA_STATISTICS_UPDATED. ‘universalparser’ berfungsi untuk memilih parser yang sesuai berdasarkan *file G-code* dan menyimpan semua data informasi ke dalam database, dan memperbarui status cetak jika diperlukan.

Kode Program 1. Parser

```
class UniversalParser:
    def __init__(self, file_path, logger=None):
        self.parser_factory = None
        self.name = None
        self.version = None
        self.file = open(file_path, "r")
        self._logger = logger if logger else logging.getLogger(__name__)
        for parser in [CuraParser(), Slic3rParser(), Simplify3DParser()]:
            if parser.detect(self.file):
                self.parser_factory = parser
                self.name = parser.name
                self.version = parser.version

    def parse(self):
        if self.parser_factory:
            parameters = self.parser_factory.parse(self.file)
            parameters.update({"slicer_name": self.name})
            parameters.update({"slicer_version": self.version})
        else:
            parameters = {}
            self._logger.info("Can't parse additional parameters from %s" %
self.file.name)
        return parameters
```

Kode Program 1 digunakan untuk mempermudah proses analisis file *slicing* 3D dengan secara otomatis mendeteksi dan mengekstrak informasi penting dari *G-code* tanpa memerlukan intervensi manual untuk menentukan jenis slicer yang digunakan.

Kode Program 2. Pengambilan data

```
def eventHandler(self, event, payload):
    from Octoprint.events import Events
    import json
    import time
    from operator import itemgetter
    import sqlite3

    supported_event = None
    if event in [Events.PRINT_DONE, Events.PRINT_FAILED, Events.METADATA_STATISTICS_UPDATED]:
        supported_event = event

    if supported_event is None:
        return

    if supported_event is not Events.METADATA_STATISTICS_UPDATED:
        self._logger.info("Name: " + payload["name"])
        self._logger.info("Origin: " + payload["origin"])
        self._logger.info("Path: " + payload["path"])
        self._logger.info("Path_On_disk: " + self._file_manager.path_on_disk(payload["origin"],
payload["path"]))

    try:
        fileData = self._file_manager.get_metadata(payload["origin"], payload["name"])
        self._logger.info("Got metadata from name")
    except:
        self._logger.info("Error getting metadata from name, trying with path")
        try:
            fileData = self._file_manager.get_metadata(payload["origin"], payload["path"])
        except:
            self._logger.info("Error getting metadata from name and path, terminating")
            fileData = None

    fileName = payload["name"]
    if fileData is None:
        self._logger.info("FileData came out empty, trying to get it from path")
        fileData = self._file_manager.get_metadata(payload["origin"], payload["path"])

    if fileData is not None:
        self._logger.info("found fileData")
        timestamp = 0
        success = None
        estimatedPrintTime = 0
        gcode_parser = UniversalParser(self._file_manager.path_on_disk(payload["origin"],
payload["path"]), logger=self._logger)
        parameters = gcode_parser.parse()
        currentFile = {
            "fileName": fileName,
```

```
        "note": "",
        "parameters": json.dumps(parameters)
    }
    self._logger.info(json.dumps(parameters))
    # Analysis - looking for filament usage
    if "analysis" in fileData:
        if "filament" in fileData["analysis"]:
            if "tool0" in fileData["analysis"]["filament"]:
                filamentVolume = fileData["analysis"]["filament"]["tool0"]["volume"]
                filamentLength = fileData["analysis"]["filament"]["tool0"]['length']

                currentFile["filamentVolume"] = filamentVolume if filamentVolume is not
None else 0
                currentFile["filamentLength"] = filamentLength if filamentLength is not
None else 0

            if "tool1" in fileData["analysis"]["filament"]:
                filamentVolume = fileData["analysis"]["filament"]["tool1"]["volume"]
                filamentLength = fileData["analysis"]["filament"]["tool1"]['length']

                currentFile["filamentVolume2"] = filamentVolume if filamentVolume is not
None else 0
                currentFile["filamentLength2"] = filamentLength if filamentLength is not
None else 0

            estimatedPrintTime = fileData["analysis"]["estimatedPrintTime"] if "estimat-
edPrintTime" in fileData["analysis"] else 0

            if not currentFile.get("filamentVolume"):
                currentFile["filamentVolume"] = 0

            if not currentFile.get("filamentLength"):
                currentFile["filamentLength"] = 0

            if "time" in payload:
                currentFile["printTime"] = payload["time"]
            else:
                printTime = self._comm.getPrintTime() if self._comm is not None else ""
                currentFile["printTime"] = printTime

            if "history" in fileData:
                history = fileData["history"]
                newlist = sorted(history, key=itemgetter('timestamp'), reverse=True)
                if newlist:
                    last = newlist[0]
                    success = last["success"]

            if not success:
                success = False if event == Events.PRINT_FAILED else True

            timestamp = int(time.time())
            currentFile["success"] = success
            currentFile["timestamp"] = timestamp

            self._history_dict = None
            conn = sqlite3.connect(self._history_db_path)
            cur = conn.cursor()
            cur.execute("INSERT INTO print_history (fileName, note, filamentVolume, fila-
mentLength, printTime, success, timestamp, parameters) VALUES (:fileName, :note, :filamentVolume,
:filamentLength, :printTime, :success, :timestamp, :parameters)", currentFile)
            conn.commit()
            conn.close()

        else:
            self._logger.info("fileData not found")
            try:
                fileData = self._file_manager.get_metadata(payload["storage"], payload["path"])
            except:
                fileData = None

            if "history" in fileData:
                history = fileData["history"]
                newlist = sorted(history, key=itemgetter('timestamp'), reverse=True)
                if newlist:
                    last = newlist[0]
                    success = last["success"]
                    timestamp = int(last["timestamp"])
                    conn = sqlite3.connect(self._history_db_path)
                    cur = conn.cursor()
```

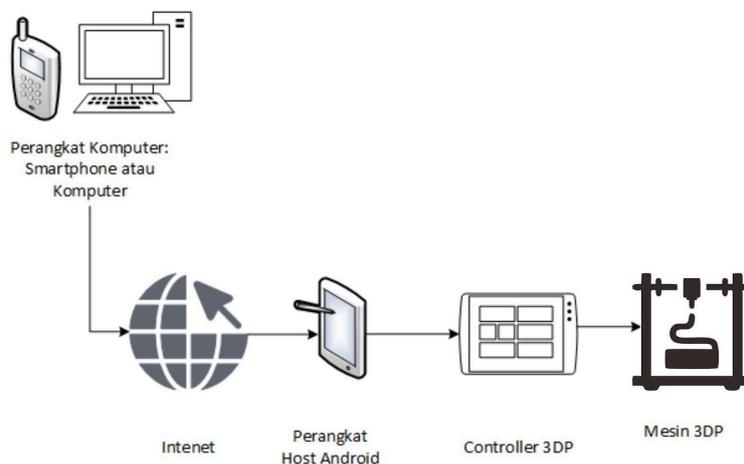
```
timestamp))
    cur.execute("UPDATE print_history SET success = ? WHERE timestamp = ?", (success,
conn.commit()
conn.close()
```

Kode Program 2 digunakan untuk melakukan pengambilan data yang didapat setelah proses *parser* yang meliputi nama, waktu cetak, dan penggunaan filament dan melalui event jam dan tanggal percetakan serta status hasil percetakan berhasil atau gagal. Data yang didapat dari metadata dan event akan disimpan dalam basis data SQLite. Plugin akan membuat koneksi ke database, menyisipkan data baru atau memperbarui entry berdasarkan event cetak. Riwayat cetak dari *server* ditampilkan menggunakan fungsi ‘gethistorydata’ dalam format Tab Separated Values (TSV).

Untuk memastikan bahwa konfigurasi perangkat lunak pada *3D printer* berfungsi dengan baik, proses pengujian dimulai dengan memeriksa fungsionalitas dasar, seperti konektivitas antara perangkat lunak dan printer serta respons terhadap perintah dasar seperti *Home* dan *Move Axis*. Setelah itu, kalibrasi printer, baik otomatis maupun manual, dilakukan untuk memastikan akurasi sumbu dan leveling bed. Pengujian parameter cetak dilakukan dengan menjalankan cetakan uji sederhana menggunakan berbagai pengaturan seperti kecepatan dan suhu untuk memastikan bahwa perangkat lunak mengelola parameter cetak dengan tepat. Selanjutnya, *file G-code* diuji untuk memastikan perangkat lunak dapat memproses dan mengeksekusi instruksi dengan benar.

Pengujian performa dilakukan dengan mencetak model kompleks dalam waktu lama untuk mengevaluasi stabilitas perangkat lunak. Selain itu, integrasi dengan sistem lain, seperti *Octoprint* dan *Klipper*, diuji untuk memastikan alur kerja dan koordinasi yang baik antar komponen. Pengujian keamanan dilakukan untuk memastikan bahwa fitur kontrol akses berfungsi dengan baik. *User experience* diuji dengan melibatkan pengguna untuk memastikan antarmuka perangkat lunak intuitif dan efisien. Terakhir, penanganan kesalahan diuji dengan sengaja memicu kondisi error untuk memverifikasi bahwa perangkat lunak memiliki mekanisme yang memadai dalam mendeteksi dan menangani masalah. Melalui serangkaian pengujian ini, diharapkan perangkat lunak dapat berfungsi dengan baik dan memberikan hasil cetakan yang optimal.

C. Arsitektur Mesin Cetak 3 Dimensi

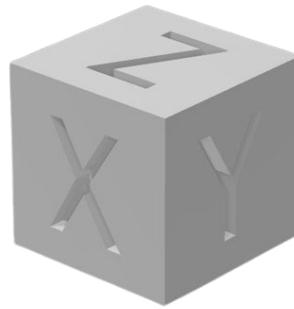


Gambar. 6. Arsitektur Mesin Cetak 3 Dimensi

Pada Gambar 6 menunjukkan pengujian ini memiliki arsitektur dengan alur proses perangkat *user* akan berfungsi sebagai control yang dapat memberi perintah dan menerima informasi yang terhubung pada perangkat *server* melalui jaringan internet untuk mengendalikan dan memantau aktifitas *3D printer*.

D. Uji Cetak

Pengujian yang dilakukan adalah dengan mencetak 3D objek dengan metode IoT (melalui antar muka web) dengan mencetak XYZ cube. Gambar 7 merupakan bentuk dari XYZ cube yang menjadi objek pengujian cetak penelitian ini.



Gambar. 7. XYZ Cube

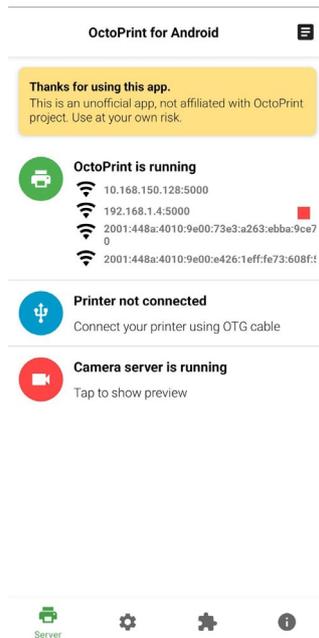
E. Hasil Perancangan

Realisasi dari perancangan *3D printer* IoT terdiri dari perangkat keras dan perangkat lunak. Gambar 8 Menunjukkan hasil dari realisasi *3D printer* IoT dengan konfigurasi corexy. Adapun tampilan dari antarmuka perangkat lunak Octo4a pada smarphone Android yang menjadi *host server* ditunjukkan pada Gambar 9.

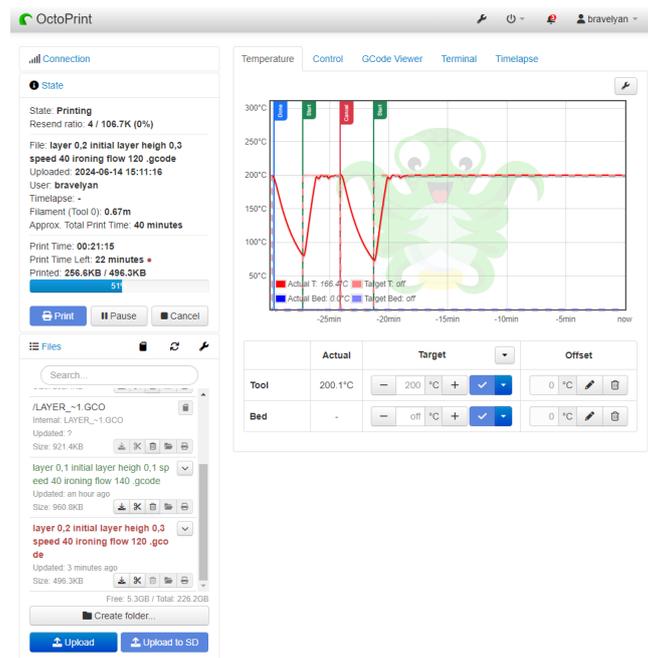
Antarmuka *Octoprint* yang berjalan perangkat *user* beserta tampilan plugin riwayat cetak ditunjukkan pada Gambar 10 dan Gambar 11. Tampilan *plugin history* yang disajikan pada Gambar 11 memperlihatkan nama file diikuti oleh indikator status pencetakan (berhasil atau gagal) pada kolom pertama. Kolom kedua dan ketiga menampilkan tanggal dan waktu saat pencetakan dimulai, sedangkan kolom keempat menunjukkan jumlah filament yang digunakan.



Gambar. 8. *3D printer*



Gambar. 9. Antar muka octo4a pada *smartphone server*



Gambar. 10. Antar muka web Octoprint pada *smartphone user*

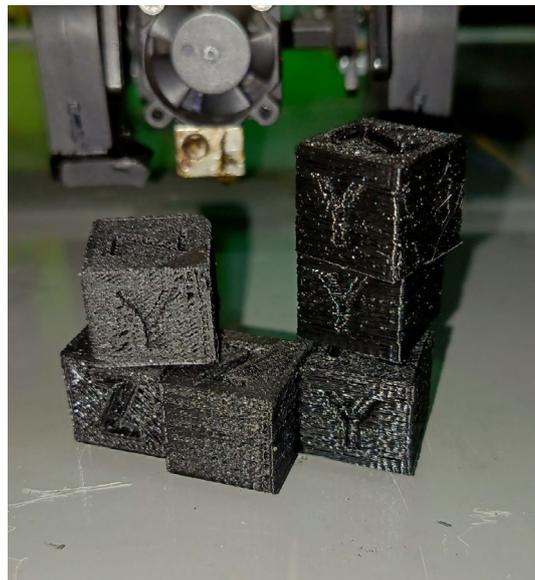
Nama	Tanggal	Waktu Cetak	Penggunaan Filamen
CFFFP_xyzCalibration_cubenmnm.g	2024-07-26 17:27	00:01:00	0.68m
CFFFP_xyzCalibration_cube.gcode	2024-07-26 18:07	00:26:22	0.77m
CFFFP_xyzCalibration_cube 04.gcode	2024-07-26 18:25	00:18:09	0.76m
CFFFP_xyzCalibration_cube.gcode	2024-07-26 18:54	00:28:19	0.77m
Total:		01:13:52	2.96m
Rata-rata:		00:18:28	0.74m

Gambar. 11. Tampilan Riwayat kerja

F. Uji Cetak dengan menerapkan IoT

Pengujian cetak menggunakan perangkat *smartphone* Android yang berjalan sebagai *server* dan memantau menggunakan camera yang terhubung langsung ke printer.

Hasil cetak *3D printer* dengan metode IOT menggunakan *smartphone* Android sebagai host *server* dilampirkan sebagai berikut.



Gambar. 12. Hasil cetak 3D

Kehandalan sistem IoT dalam mengontrol *3D printer* dari jarak jauh sangat dipengaruhi oleh beberapa faktor utama yang memengaruhi performa dan efisiensi sistem secara keseluruhan. Koneksi internet yang stabil dan cepat adalah kunci utama, karena sistem IoT membutuhkan transfer data yang lancar antara *3D printer* dan perangkat pengendali jarak jauh. Kualitas koneksi yang buruk dapat menyebabkan keterlambatan perintah, gangguan dalam komunikasi, atau bahkan kehilangan data, yang pada gilirannya dapat mempengaruhi kualitas cetakan atau menyebabkan kerusakan pada printer.

Sistem IoT (Internet of Things) yang telah dibangun menawarkan sejumlah keunggulan signifikan dibandingkan dengan sistem konvensional, terutama dalam hal pengendalian dan pemantauan perangkat *3D printer*. Salah satu keunggulan utama adalah kemampuan kontrol dan pemantauan jarak jauh. Dengan sistem IoT, pengguna dapat mengontrol dan memantau *3D printer* dari mana saja melalui internet, yang memungkinkan manajemen cetakan secara lebih fleksibel dan efisien tanpa harus berada di lokasi fisik printer. Keunggulan lainnya adalah pengumpulan

dan analisis data real-time. Sistem IoT memungkinkan pengumpulan data operasional secara otomatis dan terus-menerus, serta analisis data tersebut untuk mengidentifikasi tren, memantau performa, dan mengoptimalkan proses cetak.

Pada pengujian cetak sebanyak sepuluh kali di lampirkan pada tabel berikut.

TABEL II
 SPESIFIKASI MESIN 3D PRINTER

Kecepatan	PANJANG	LEBAR	TINGGI
20mm/s	20	20	20
25mm/s	20	20	20
30mm/s	20	20	20,50
35mm/s	21	20,10	19,10
40mm/s	20	20	20
45mm/s	21,50	20	18,70
50mm/s	20	20,7	21
55mm/s	20,5	20,1	20
60mm/s	20,40	19	20
65mm/s	19,70	20	20

Analisis data dari uji coba yang dilakukan menunjukkan bahwa hasil cetakan 3D printer masih berada dalam batas toleransi yang dapat diterima, menandakan bahwa sistem berfungsi dengan baik. Dari pengujian ini, tidak ada pola signifikan yang menunjukkan adanya penurunan kualitas cetakan, baik dalam hal akurasi dimensi, permukaan, maupun kekuatan material. Sistem IoT yang diterapkan tidak menyebabkan degradasi kualitas, dan justru memungkinkan pengendalian dan pemantauan yang lebih efisien tanpa menurunkan kualitas hasil akhir. Metrik evaluasi yang digunakan meliputi akurasi dimensi, yang memastikan bahwa ukuran hasil cetak sesuai dengan spesifikasi model; kualitas permukaan, yang dinilai berdasarkan tingkat kehalusan dan konsistensi pada permukaan cetakan; serta kekuatan struktural, yang mengukur kemampuan hasil cetak dalam menahan beban dan tekanan sesuai dengan standar material yang digunakan. Hasil evaluasi menunjukkan bahwa semua metrik berada dalam kisaran optimal, mendukung kesimpulan bahwa penerapan sistem IoT tidak hanya efektif, tetapi juga baik dalam menjaga kualitas cetakan.

IV. KESIMPULAN

Penelitian ini berhasil merancang 3D printer dengan IoT menggunakan *smartphone* Android sebagai *host server*. Penggunaan *smartphone* Android dapat mempermudah dalam penggunaan 3D printer dikarenakan adanya aplikasi Octo4a yang dapat menjalankan *Klipper* dan *Octoprint* pada *smartphone* Android. *Smartphone* Android juga memiliki kamera yang dapat menjadi webcam untuk merekam dan memantau pergerakan 3D printer dengan dibuktikan melalui uji cetak, Android dapat berfungsi sebagai *Host server* pada mesin cetak 3 dimensi agar dapat dikendalikan secara IoT. alih alih menggunakan komputer, *smartphone* Android yang sangat umum dan mudah ditemukan di era sekarang dan memiliki banyak fitur yang sangat berguna untuk menjalankan 3D printer secara IoT dengan harga sangat variatif dapat menjadi pilihan.

DAFTAR PUSTAKA

- [1] Achrén, A., & Bårdén, J. (n.d.). *Input shaping in a cantilever 3D printer, Degree project in Technology First cycle, 15 Credits.*
- [2] Bozkurt, Y., & Karayel, E. (2021). 3D printing technology; methods, biomedical applications, future opportunities and trends. In *Journal of Materials Research and Technology* (Vol. 14, pp. 1430–1450). Elsevier Editora Ltda. <https://doi.org/10.1016/j.jmrt.2021.07.050>
- [3] Arianti, A., Alpian, B., Al Fharezi, M. G., Putra, M. A. T., Priti, P., & Hermawan, R. (2023). Pemanfaatan Objek 3D Printing Sebagai Pengembangan Media Pembelajaran Bangun Ruang Berbasis Arduino Untuk Siswa Penyandang Tunanetra. *Nusantara: Jurnal Pendidikan Indonesia*, 3(3), 549–566. <https://doi.org/10.14421/njpi.2023.v3i3-11>
- [4] Mailani, Y., & Candra, O. (n.d.). RANCANG BANGUN 3D printer MENGGUNAKAN SISTEM AUTO LEVELING DENGAN MIKROKONTROLER DESIGN 3D printer USING AUTO LEVELING SYSTEM WITH MICROCONTROLLER. *Journal of Information Technology and Computer Science (INTECOMS)*, 4(2), 2021.
- [5] Salsabila, D., & Pradoto, R. G. K. (2022). Analysis of Factors Influencing the Decision to Adopt 3D Printing Technology in a Case Study of Residential Housing Development. *The 3rd ITB Graduate School Conference (IGSC) 2022*, 21, 118–131. <https://gcs.itb.ac.id/prosiding-book-igsc/>
- [6] Abutaleb, A., Alhamad, M., Al-Orf, M., Alrashed, K., & Esmaeili, S. E. (2023). the Design of a Dual Open-Source 3D printer Utilizing Fdm and Msla Printing Technologies. *International Journal of Mechatronics and Applied Mechanics*, 2023(13), 103–111. <https://doi.org/10.17683/ijomam/issue13.13>
- [7] SETIAWAN, B., PURWANTI, D. N., & DEWATAMA, D. (2023). Kontrol Resume pada Mesin 3D printer 2x2x2 Meter berbasis MCU Bigtree.

- ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, 11(2), 337.
<https://doi.org/10.26760/elkomika.v11i2.337>
- [8] Xie, Baocheng & Gao, Huaqiang & Zhang, Tingliang. (2022). A Review on the control system of 3D printer. *Recent Patents on Engineering*. 16. 10.2174/1872212116666220310164041.
- [9] Priyatna, I. P. A. C., Agung, I. G. A. P. R., & Divayana, Y. (2021). Pengaruh Penambahan Sistem IOT Raspberry PI Terhadap Kecepatan Cetak dan Tingkat Produktivitas Operator 3D printer Reprap. *Jurnal SPEKTRUM*, 08(02), 128–137.
<https://doi.org/10.24843/SPEKTRUM.2021.v08.i02.p15>
- [10] Rusnawati, R. D., & Hariyati, R. T. S. (2022). Implementasi Internet of Things Pada Layanan Kesehatan (Literature Review). *Journal Of Innovation Research And Knowledge*, 1(8), 569–574. <https://doi.org/10.53625/jirk.v1i8.1082>
- [11] Grzywok, F. (2024). Run *Octoprint* on Android. Github. <https://github.com/feelfreelinus/octo4a?tab=readme-ov-file>
- [12] Afshar, R., Jeanne, S., & Abali, B. E. (2023). Nonlinear Material Modeling for Mechanical Characterization of 3-D Printed PLA Polymer With Different Infill Densities. *Applied Composite Materials*, 30(3), 987–1001. <https://doi.org/10.1007/s10443-023-10122-y>
- [13] Wijaya, A. S., Suprianto, B., Buditjahjanto, I. G. P. A., & Baskoro, F. (2023). Uji Functional Suitability Dan Usability Pada Sistem Kendali Mesin 3D Print Berbasis IoT. *Jurnal Teknik Elektro*, 12(1), 45–53. <https://doi.org/10.26740/jte.v12n1.p45-53>
- [14] Prianto, E., Pramono, H. S., & Yuchofif. (2021). IoT-Based 3D printer Development for Student Competence Improvement. *Journal of Physics: Conference Series*, 2111(1). <https://doi.org/10.1088/1742-6596/2111/1/012002>
- [15] Putra, A. K., Fidiyanto, F., Prakoso, B., Armantya, R. Z., Sandi, M. K., & Al Haris, F. H. S. (2019). Perakitan 3D printer Fused Deposite Modeling (FDM) Berbasis Arduino Mega 2560. *Jurnal Gaung Informatika*, 12(2), 123–133. <https://doi.org/10.47942/gi.v12i2.380>
- [16] Setyoadi, Y., Hermana, R., & Mulyanto, F. D. (2022). Performance Analysis of The 3D printer Corexy FDM Type With Area X=200 Y=200 Z=200 mm. *Jurnal Teknik Mesin Mechanical Xplore (JTMMX) Mechanical Engineering*, 3(1), 26–33. <https://doi.org/10.36805/jtmmx.v3i1.2476>
- [17] Kumar, N. (2023). *Advances in Materials and Agile Manufacturing: Select Proceedings of CPIE 2023*. Springer.
- [18] Häußge, Gina. 2020. *Octoprint's* documentation. [Online] Tersedia di: <https://docs.Octoprint.org/en/master/> .[Diakses pada 30 Juni 2024].
- [19] Sampedro, G. A., Putra, M. A. P., Kim, D.-S., & Lee, J.-M. (2021). 3D printer State Prediction: A Deep Learning Model Approach. 2021 1st International Conference in Information and Computing Research (ICORE), 135–138. <https://doi.org/10.1109/iCORE54267.2021.00043>
- [20] Teknik Mesin, J., & Manufaktur Negeri Bangka Belitung, P. (2019). Optimasi Parameter Proses 3D Printing FDM Terhadap Akurasi Dimensi Menggunakan Filament Eflex. *Jurnal Teknologi Manufaktur*, 11(01).