

## DAFTAR PUSTAKA

- Andesta, Deri, and Rian Ferdian. 2018. "Sistem Keamanan Sepeda Motor Berbasis Mikrokontroler dan Modul GSM." *Journal of Information Technology and Computer Engineering* 51-63.
- Ayuningtyas, Belinda, and Syaeful Ilman. 2021. "IP Camera Surveillance System Using Android Application Based on Arduino." *Jurnal Teknik dan Informatika* 1-12.
- Babusiak, Branko , Stefan Borik, and Ludmila Balogova. 2017. "Textile electrodes in capacitive signal sensing."
- Bayu Fandidarma, Ridam Dwi Laksono, and Krisna Warih Bintang Pamungkas. 2021. "Rancang Bangun Mobil Remote Control Pemantau Area berbasis IoT menggunakan ESP 32 Cam." *Jurnal ELECTRA : Electrical Engineering Articles* 31-38.
- Difa, M. Kelvin, Suroso, and Jon Endri. 2021. "Implementasi Sistem Pengenalan Wajah Sebagai Automatic Door Lock Menggunakan Modul ESP32 CAM." *Technological Journal*.
- Gura, Dmitry, Ivan Markovskii, Nafset Khusht, Irina Rak, and Saida Pshidatok. 2021. "A Complex for Monitoring Transport Infrastructure Facilities Based on Video Surveillance Cameras and Laser Scanners." *Transportation Research Procedia* 775-782.
- Izrofi, Azis, Shoffin Nahwa Utama, and Oddy Virgantara Putra. 2021. "Rancang Bangun Robot Pemotong Rumput Otomatis Menggunakan Wireless Kontroller Modul ESP32-CAM Berbasis Internet of Things (IoT)." *Jurnal TEKNOINFO* 45-55.
- Kryknavev, Bauyrzhan, Muhammad Fahad Farooqui, Farooqui Muhammad ilal,, Mohammad Vaseem, and Atif Shamim. 2017. "A wearable



tracking device inkjet-printed on textile." *Microelectronics Journal* 40-48.

Kurilkin, Alexey V, Oksana O Vyatkina, and Sergey A Mityagin. 2015. "Evaluation of urban mobility using surveillance cameras." *4th International Young Scientists Conference on Computational Science* 364-371.

Meligiatri, Tania, Heroe Wijanto, and Edwar. 2021. "Perancangan dan Realisasi Antena Mikrostrip Tekstil Dual Band untuk Penerima GPS." *e-Proceeding of Engineering* 8359.

Mola, Tahmidul Islam, Steven Goodman, Nicholas Schleif, Mary Ellen Berglund, Cade Zacharias, Crystal Compton, and Lucy Dunne. 2017. "Surface-Mount Manufacturing for E-Textile Circuits."

Munirman, Sufianti, Elyas Palantei, Merna Baharuddin, and Zulfahmi Rizal. 2020. "An IoT Wearable Communication Prototype Tested in Indoor and Outdoor Environments ." *International Conference on Communication, Networks and Satellite*.

Novian, Ervan, Levy Olivia Nur, and Bambang Setia Nugroho. 2020. "Antena Dual Band Frekuensi 2.45 GHz dan 5.85 GHz untuk Aplikasi Telemedis." *e-Proceeding of Engineering* 8970-8977.

Rafika, Agung Setiani, Asep Saefullah, and Andri Ahmad Gozali. 2016. "Wireless Surveillance Camera with AMC (Automatic Motion Captures) System ."

Ronanki, Deepak, Apoorva Kelkar, and Sheldon S. Williamson. 2019. "Extreme Fast Charging Technology—Prospects to Enhance Sustainable Electric Transportation." 1-17.

Ryzkiansyah, Novaria Elsari, Wijaya Kurniawan, and Barlian Henryranu rasetio. 2018. "Perancangan Smart Surveillance System Pada



Smarthome Menggunakan NI MyRio." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* 2854-2860 .

Salamah, Yasmin, and Basari. 2022. "Desain Arsitektur Sistem Pemantauan Tanda Vital dan Postur Bayi Berbasis Wearable dalam Pencegahan Kejadian Bayi Mati Mendadak." *Medika Teknika : Jurnal Teknik Elektromedik Indonesia*.

Sangsari, Annisa, Isnawaty, and Fid Aksara. 2016. "Analisis QOS (Quality Of Service) pada Layanan Video Streaming yang menggunakan Protokol RTMP (Rel Time Messaging Protocol)." *semanTIK* 177-188.

Shim, Jae Hong, and Young Im Cho. 2015. "A Mobile Robot Localization using External Surveillance Cameras at Indoor." *International Workshop on Communication for Humans, Agents, Robots, Machines and Sensors* 502-507.

Subektiningsih, Renaldi, and Pramudhita Ferdiansyah. 2022. "Analisis Perbandingan Parameter QoS Standar TIPHON Pada Jaringan Nirkabel Dalam Penerapan Metode PCQ."

Swastika, Vanessa Mayrahma. 2023. *Kompasiana.com*. Januari Minggu. Accessed Januari 15, 2023. 1. <https://www.kompasiana.com/vanessams/55547634b67e615e14ba545b/perkembangan-teknologi-di-indonesia>.

Tang, Junqiang, Li Wan, Jennifer Schooling, Pengjun Zhao, Jun Chen, and Shufen Wei. 2022. "Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases."

Yonathan, Bryan, Yoanes Bandung, and Armein Z.R Langi. 2011. "Analisis Kualitas Layanan (QOS ) audio-video layanan kelas virtual di



jaringan digital learning pedesaan." *Konferensi Teknologi Informasi dan Komunikasi untuk Indonesia*.

Yoshiura, Noriaki, and Yoshinori Neya. 2010. "USB memory for surveillance camera software." *Procedia Social and Behavioral Sciences* 36-41.

Yullita, Winda, and Aidil Afriansyah. 2022. "Alat Pemantau Keamanan Rumah Berbasis ESP32-CAM." *Jurnal Teknologi dan Sistem Tertanam* 23-31.

Zufri, Khozainuz, and Ahmad Ikhwan. 2020. "Perancangan Sistem Keamanan Ganda Brangkas Berbasis Telegram Menggunakan Mikrokontroler ESP32-CAM." *Jurnal Teknologi dan Informatika*.



## LAMPIRAN

### Lampiran 1 Gambar Proses Pembuatan Perangkat Komunikasi Wearable



## Lampiran 2 Program Kendali untuk Perangkat Komunikasi Wearable

```

#include <TinyGPSPlus.h>
#include <HardwareSerial.h>
#include <Arduino.h>
#include <WiFi.h>
#include <WebServer.h>
#include <WiFiClient.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"

const char* ssid = "ESP32-CAM Access Point";
const char* password = "12345678";

WebServer server(80);

String uploadURL =
"http://smartwearable.000webhostapp.com/upload.php";

const int timerInterval = 10000;
unsigned long previousMillis = 0;

float latitude , longitude;
String latitude_string , longitiude_string;

TinyGPSPlus gps;

int SIMMaxRetry = 5;

HardwareSerial SerialSIM(1);
HardwareSerial SerialGPS(2);

#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM      5
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23

```



```

#define PCLK_GPIO_NUM      22
#define LED_FLASH          4

void setup() {
    pinMode(LED_FLASH, OUTPUT);

    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
    SerialSIM.begin(9600, SERIAL_8N1, 14, 15);
    SerialSIM.begin(9600, SERIAL_8N1, 14, 15);
    ledSignal(5);
    while(true) {
        if (sendATcommand("AT", "OK", 1000) == 1) {
            break;
        }
        SIMMaxRetry--;
        if (SIMMaxRetry == 0) {
            break;
        }
    }
    while(true) {
        if (setupGPRS()) {
            break;
        }
        if (SIMMaxRetry == 0) {
            break;
        }
    }
    if (SIMMaxRetry > 0) {
        ledSignal(5);
    }
    WiFi.softAP(ssid, password);
    server.on("/", HTTP_GET, rootHandler);
    server.on("/live", HTTP_GET, liveStreamHandler);
    server.onNotFound(notFoundHandler);
    server.begin();

    ledSignal(5);
    camera_config_t config;
    config.grab_mode = CAMERA_GRAB_LATEST;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;

```



```

config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
if (psramFound()) {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_CIF;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    delay(1000);
    ESP.restart();
}
}

void loop() {
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= timerInterval)
    {
        while (SerialGPS.available() > 0) {
            if (gps.encode(SerialGPS.read())) {
                if (gps.location.isValid()) {
                    latitude = gps.location.lat();
                    latitude_string = String(latitude , 6);
                    longitude = gps.location.lng();
                    longitiude_string = String(longitude , 6);
                }
            }
        }
    }

    WiFiClient client = server.client();
    if (!client.connected() && SIMMaxRetry > 0) {
        endPhoto();
    }
}

```





```

        previousMillis = currentMillis;
    }
    server.handleClient();
}

void rootHandler(void) {
    server.sendHeader("Content-Type", "text/html");

    String htmlContent = "<html><head><title>Smart
Wearable</title></head><body style=\"text-
align:center\"><h1>Smart Wearable</h1>";
    htmlContent += "<iframe src=\"/live\" width=\"800\"
height=\"600\"></iframe>";
    if (latitude_string.length() > 0 ||
longitude_string.length() > 0) {
        htmlContent += "<h5>GPS: " + latitude_string + ", "
+ longitude_string;"</h5></body></html>";
    } else {
        htmlContent += "<h5>GPS: No
Location</h5></body></html>";
    }

    server.send(200, "text/html", htmlContent);
}

void liveStreamHandler(void) {
    camera_fb_t * fb = NULL;

    const char HEADER[] = "HTTP/1.1 200 OK\r\nAccess-
Control-Allow-Origin: *\r\nContent-Type: multipart/x-
mixed-replace; boundary=FFOOLLBIT\r\n";
    const char BOUNDARY[] = "\r\n--FFOOLLBIT\r\n";
    const char CTNTTYPE[] = "Content-Type:
image/jpeg\r\nContent-Length: ";
    const int hdrLen = strlen(HEADER);
    const int bdrLen = strlen(BOUNDARY);
    const int cntLen = strlen(CTNTTYPE);

    char buf[32];
    int s;

    WiFiClient client = server.client();

    client.write(HEADER, hdrLen);
    client.write(BOUNDARY, bdrLen);
    while (true) {
        (!client.connected()) break;
        (fb) {

```



```

        esp_camera_fb_return(fb);
    }
    fb = esp_camera_fb_get();

    s = fb->len;
    client.write(CTNTTYPE, cntLen);
    sprintf(buf, "%d\r\n\r\n", s);
    client.write(buf, strlen(buf));
    client.write((char *)fb->buf, s);
    client.write(BOUNDARY, bdrLen);
}
}

void notFoundHandler() {
    String message = "Server is running!\n\n";
    message += "URI: ";
    message += server.uri();
    message += "\nMethod: ";
    message += (server.method() == HTTP_GET) ? "GET" :
"POST";
    message += "\nArguments: ";
    message += server.args();
    message += "\n";
    server.send(200, "text/plain", message);
}

void sendPhoto() {
    camera_fb_t * fb = NULL;
    fb = esp_camera_fb_get();
    if(!fb) {
        delay(1000);
        ESP.restart();
    }

    String fileName = latitude_string + "_" +
longitude_string;
    String head = "--FFOOLLLBIT\r\nContent-Disposition:
form-data; name=\"imageFile\"; filename=\"" + fileName
+ ".jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
    String tail = "\r\n--FFOOLLLBIT\r\n";
    uint32_t imageLen = fb->len;
    uint32_t extraLen = head.length() + tail.length();
    uint32_t totalLen = imageLen + extraLen;

    ledSignal(1);

    command[100];
    tf(command, "AT+HTTPDATA=%lu,60000", totalLen);
}

```



```

sendATcommand(command, "DOWNLOAD", 1000);
SerialSIM.print(head);

uint8_t *fbBuf = fb->buf;
size_t fbLen = fb->len;
for (size_t n=0; n<fbLen; n=n+1024) {
    if (n+1024 < fbLen) {
        SerialSIM.write(fbBuf, 1024);
        fbBuf += 1024;
    } else if (fbLen%1024 > 0) {
        size_t remainder = fbLen%1024;
        SerialSIM.write(fbBuf, remainder);
    }
}
SerialSIM.print(tail);
esp_camera_fb_return(fb);
ledSignal(2);

delay(1000);

if (sendATcommand("AT+HTTPACTION=1", "+HTTPACTION:
1,200", 60000) == 1) {
    ledSignal(3);
}

ledSignal(1);

delay(3000);
}

void ledSignal(int repeat) {
    for (int n=0; n<repeat; n++) {
        digitalWrite(LED_FLASH, HIGH);
        delay(100);
        digitalWrite(LED_FLASH, LOW);
        delay(100);
    }
    delay(1000);
}

bool setupGPRS() {
    if
(sendATcommand("AT+SAPBR=3,1,\"Contype\", \"GPRS\"",
"OK", 1000) != 1) {
        return false;
    }
}

```



```

    if
(sendATcommand("AT+SAPBR=3,1,\"APN\", \"www.xlgprs.com\"
", "OK", 1000) != 1) {
    return false;
}
if (sendATcommand("AT+SAPBR=1,1", "OK", 1000) != 1) {
    return false;
}

if (sendATcommand("AT+HTTPIPINIT", "OK", 1000) != 1) {
    return false;
}

if (sendATcommand("AT+HTTTPARA=\"CID\",1", "OK",
1000) != 1) {
    return false;
}
char command[100];
sprintf(command, "AT+HTTTPARA=\"URL\", \"%s\"",
uploadURL.c_str());
if (sendATcommand(command, "OK", 1000) != 1) {
    return false;
}
if (sendATcommand("AT+HTTTPARA=\"REDIR\",1", "OK",
1000) != 1) {
    return false;
}
if (sendATcommand
("AT+HTTTPARA=\"CONTENT\", \"multipart/form-data;
boundary=FFOOLLBIT\"", "OK", 1000) != 1) {
    return false;
}

return true;
}

int8_t sendATcommand(char* ATcommand, char*
expected_answer, unsigned int timeout) {
    uint8_t x = 0, answer = 0;
    char response[100];
    unsigned long previous;

    memset(response, 0, 100);
    delay(100);

while (SerialSIM.available() > 0)
    SerialSIM.read();

```



```
SerialSIM.println(ATcommand);

x = 0;
previous = millis();

do {
    if (SerialSIM.available() != 0) {
        response[x] = SerialSIM.read();
        x++;
        if (strstr(response, expected_answer)
!= NULL) {
            answer = 1;
        }
    }
    while ((answer == 0) && ((millis() -
previous) < timeout));

    return answer;
}
```

