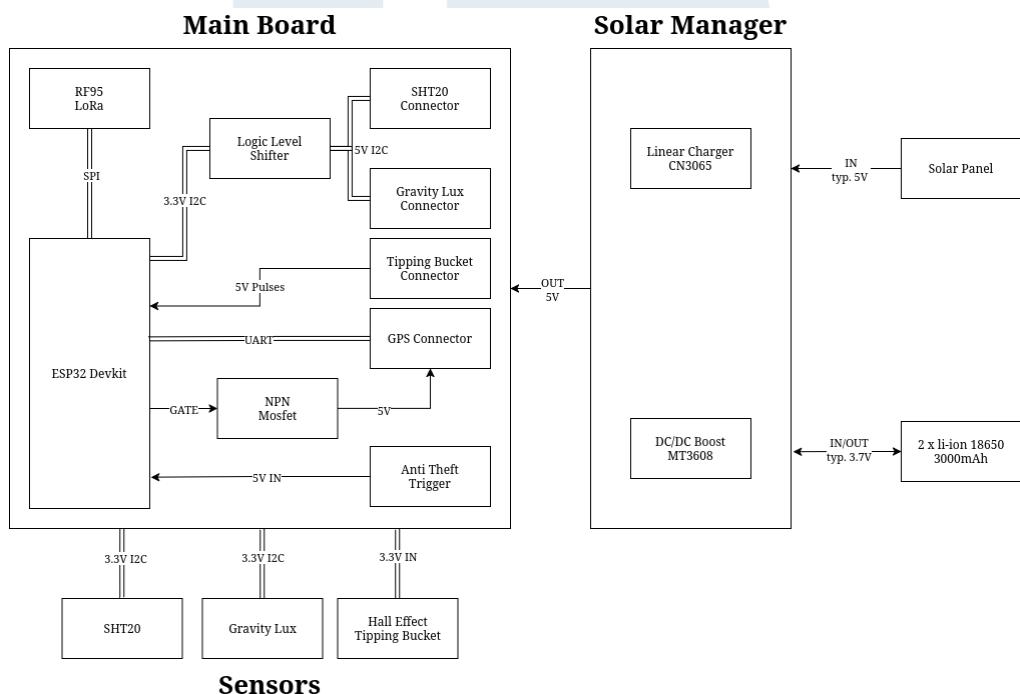


## BAB III

### METODE PENELITIAN

#### 3.1 Perancangan Solusi

Sebelum memaparkan perancangan solusi untuk node V2, sangat esensial untuk memahami terlebih dahulu untuk melakukan analisis mendalam terhadap arsitektur sistem node V1 yang sudah ada. Arsitektur V1 dapat dilihat pada gambar 3.1.



Gambar 3.1 Arsitektur MySalak V1

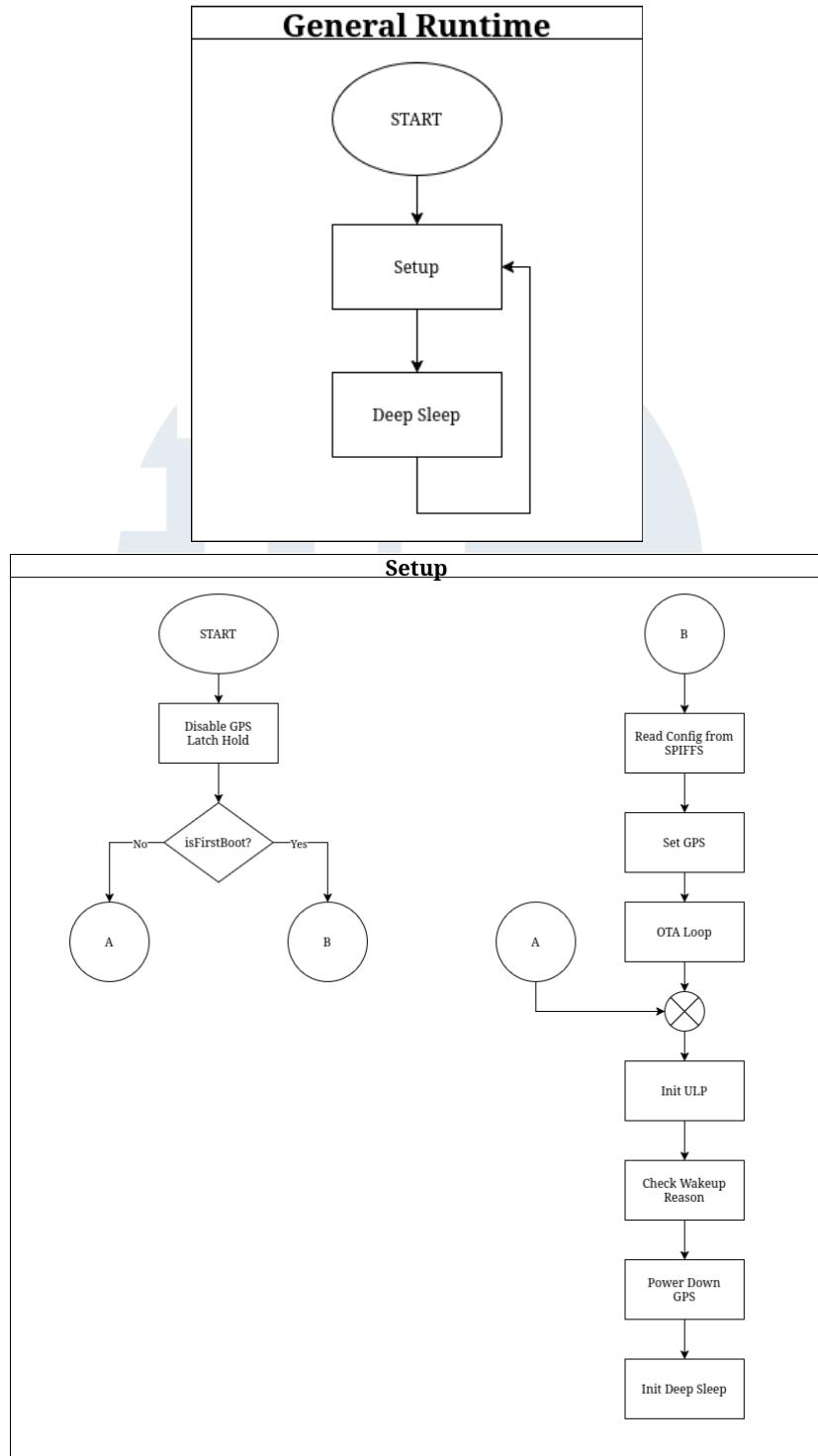
Inti dari *node* V1 adalah ESP32-Devkit V1. Untuk mengintegrasikan berbagai modul, sebuah PCB khusus yang dirancang sebagai *main board*. PCB ini menggunakan teknologi THT (*Through-Hole Technology*) dan berfungsi untuk menghubungkan berbagai komponen siap pakai (*off-the-shelf*). Komunikasi *wireless* jarak jauh diimplementasikan menggunakan *breakout board* RFM95

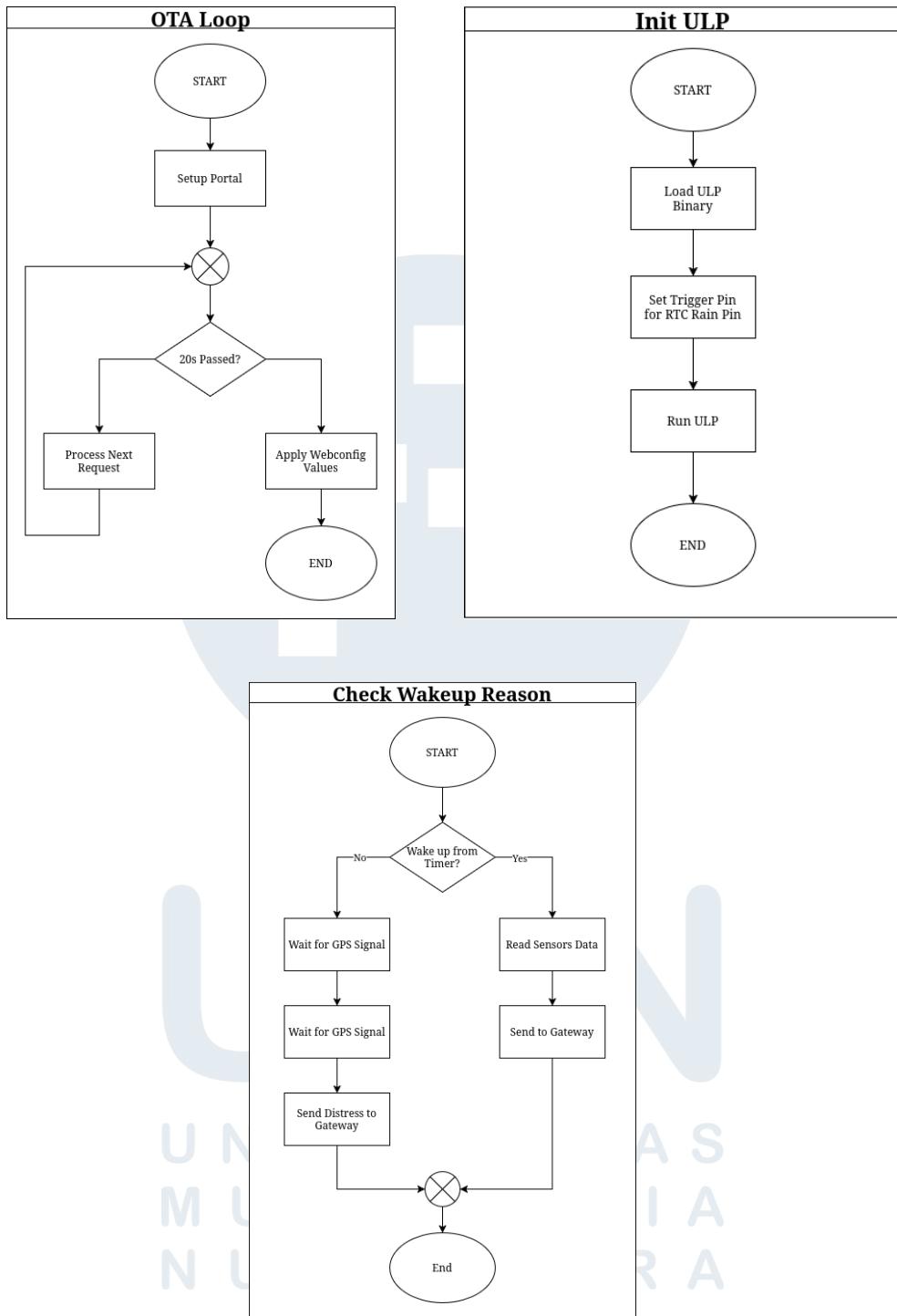
yang berbasis *chip* LoRa SX1276. Modul ini terhubung ke mikrokontroler melalui *interface* SPI.

*Main board* ini menyediakan beberapa konektor untuk *interfacing* dengan peripheral eksternal:

- Dua konektor I2C yang ditenagai oleh 5V. Karena adanya perbedaan tegangan dengan logika ESP32 (3.3V), sebuah *logic level shifter* diintegrasikan pada PCB.
- Satu konektor untuk sensor curah hujan (*tipping bucket*) yang beroperasi pada tegangan 3.3V.
- Satu konektor untuk sistem “anti-pencurian” yang memonitor pin wakeup ESP32. Pin ini akan mendeteksi perubahan level logika menjadi LOW jika konduktivitas pada pipa besi penyangga terputus.
- Satu konektor untuk modul GPS apabila sistem “anti-pencurian” ini menyala.

*Power supply* untuk V1 dirancang menggunakan modul eksternal terpisah, yaitu DFRobot Solar Power Manager. Modul ini berbasis IC CN3065 dan dilengkapi dengan DC-DC Boost Converter MT3608 yang menghasilkan keluaran stabil 5V. Keluaran 5V ini kemudian dihubungkan langsung ke pin VIN pada papan ESP32-Devkit. Akibatnya, untuk mendapatkan tegangan 3.3V yang dibutuhkan oleh chip ESP32, sistem mengandalkan regulator LDO AMS1117 bawaan dari papan DevKit tersebut, sebuah komponen yang dikenal memiliki efisiensi rendah.





### Gambar 3.2 Alur *software* MySalak V1

Alur kerja paling atas dari firmware V1 ditunjukkan pada Gambar 3.2, khususnya pada diagram "General Runtime". Diagram ini mengilustrasikan bahwa node beroperasi menggunakan pola siklus *one-shot*. Setiap siklus dimulai dengan node bangun dari deep sleep, kemudian langsung menjalankan fungsi *setup()* satu kali. Setelah seluruh logika program di dalam *setup()* selesai dieksekusi, node akan segera kembali ke mode *Deep Sleep*. Fungsi *loop()* sengaja dibiarkan kosong dan tidak pernah dieksekusi, sejalan dengan arsitektur *one-shot* yang dirancang.

Pada alur *setup()*, yang diilustrasikan pada frame "Setup", langkah pertama adalah mematikan GPS *Latch Hold* untuk memastikan modul GPS benar-benar non-aktif, terutama saat *first boot*. Selanjutnya, *firmware* melakukan pengecekan *isFirstBoot*. Jika ini adalah *boot* pertama, *node* akan membaca konfigurasi dari SPIFFS dan mengkonfigurasi GPS untuk fitur anti-pencurian. Setelah itu, sebuah *loop OTA* (*Over-the-Air*) diaktifkan selama 20 detik.

Alur "OTA Loop" ini, yang di detailkan pada frame-nya, akan membuka *access point* Wi-Fi (SoftAP) dan menjalankan *web server* sebagai portal setup. Jika ada perangkat yang terhubung dalam 20 detik, durasi *loop* akan diperpanjang hingga perangkat terputus atau *node* melakukan *restart* setelah *firmware update*. Setelah interaksi selesai atau 20 detik berlalu, konfigurasi web akan disimpan ke SPIFFS.

Selesai OTA, mikrokontroler mengkonfigurasi ULP *Co-Processor* untuk perhitungan *pulse*, seperti yang ditunjukkan pada frame "Init ULP". Program biner ULP dimuat, pin RTC di-assign, dan ULP mulai berjalan di *background* untuk memonitor *pulse* selama mode *deep sleep*.

Jika *node* mendeteksi ini bukan *first boot*, alur program akan langsung melompat ke *sub-routine* "Check Wakeup Reason". Seperti yang di ilustrasikan pada frame "Check Wakeup Reason", mikrokontroler akan memeriksa pemicu bangunnya. Terdapat dua kondisi:

- Jika bangun karena *timer deep sleep* selesai, *node* akan membaca data sensor lingkungan dan kemudian mengirimkan *payload* melalui LoRa ke *gateway*.
- Jika bangun akibat pemicu "anti-pencurian" (GPS *pin trigger*), *node* akan menyalakan modul GPS (dengan mengaktifkan N-channel MOSFET), menunggu sinyal GPS berhasil didapat, dan kemudian mengirimkan notifikasi peringatan melalui *gateway*.

Setelah salah satu dari alur tersebut selesai dieksekusi, *node* akan kembali memasuki mode *deep sleep* untuk menghemat daya.

Dalam alur kerja tersebut, beberapa upaya optimalisasi daya telah diimplementasikan, antara lain:

- Mode *Deep Sleep* digunakan sebagai strategi utama untuk meminimalkan konsumsi daya dengan mematikan CPU utama hampir sepanjang waktu.
- Frekuensi CPU secara eksplisit diturunkan menjadi 80MHz dari kecepatan maksimalnya untuk mengurangi konsumsi daya dinamis.
- Sebelum memasuki deep sleep, firmware secara eksplisit menidurkan modul LoRa dan mengonfigurasi pin-pin GPIO menggunakan `rtc_gpio_hold_en()` untuk mencegah kondisi floating yang dapat menyebabkan kebocoran arus.

### 3.1.1 Perancangan *Hardware*

Perancangan perangkat keras *node* V2 berpusat pada pemilihan mikrokontroler yang mampu menjawab kelemahan V1. Setelah melalui proses evaluasi, STM32U585 dari STMicroelectronics dipilih sebagai MCU. Kriteria seleksi utama yang bersifat mutlak adalah kompatibilitas penuh dengan Arduino *framework* dan *library* RadioHead. Kebutuhan ini bersifat mutlak untuk menjaga *backward compatibility* dengan infrastruktur *gateway* yang ada, yang telah dirancang untuk memproses paket data dengan format *datagram* dari *library* tersebut. Di dalam batasan kompatibilitas itu, STM32U585 terpilih karena arsitektur *ultra-low-power* yang superior, yang menawarkan mode-mode tidur dengan konsumsi arus di level nanoampere, serta performa core ARM Cortex-M33 yang lebih dari memadai untuk tugas akuisisi dan pemrosesan data sensor. Kombinasi antara kompatibilitas yang terjamin dan efisiensi daya yang terdepan di kelasnya menjadikan STM32U585 sebagai fondasi yang ideal untuk arsitektur perangkat keras *node* V2.

Untuk perancangan *node* V2, diputuskan untuk tetap menggunakan jajaran sensor yang sama seperti yang telah diimplementasikan dan divalidasi pada V1, yaitu sensor suhu dan kelembaban DFRobot SHT20, sensor intensitas cahaya DFRobot Gravity *Ambient Light*. Namun, ada perubahan pada sensor hall effect switch untuk *tipping-bucket* yang menggunakan DRV5032.

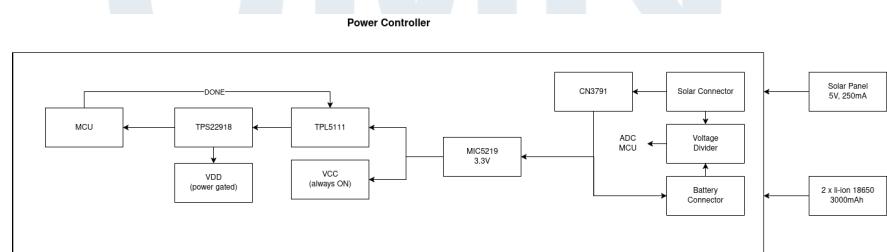
Justifikasi utama di balik keputusan ini adalah bahwa hasil analisis *post-mortem* pada V1 secara jelas mengidentifikasi sumber inefisiensi daya berada pada level arsitektur mikrokontroler, sistem *power supply*, dan *peripheral* yang memiliki *quiescent current* yang tinggi. Sensor-sensor yang digunakan pada V1 telah terbukti andal dalam melakukan akuisisi data di lapangan.

Strategi paling optimal yang diidentifikasi dari tinjauan pustaka, seperti yang didemonstrasikan oleh Cerchecci et al. (2018), adalah penggunaan *timer* eksternal berdaya sangat rendah seperti IC TPL5111 untuk mencapai kondisi *true power-off* pada keseluruhan sistem. Arsitektur ini secara teoritis menawarkan efisiensi tertinggi karena konsumsi daya MCU menjadi nol absolut saat tidak aktif.

Berbeda dengan V1 yang hanya melakukan *power gating* pada modul GPS, desain V2 mengimplementasikan power gating menggunakan *load switch* TPS22918. *Load switch* ini langsung dikontrol oleh IC *timer* TPL5111 untuk mengontrol daya secara *high-side*.

Sesaat sebelum memasuki mode *Standby*, MCU akan mengirim sinyal *DONE* untuk memberi tahu ke *timer* operasional telah selesai dan harus masuk ke mode *standby*, yang secara efektif memutus total suplai daya ke semua *peripheral* di region *power-gated*, termasuk mikrokontroler, seluruh sensor dan modul LoRa. Langkah ini memastikan bahwa konsumsi daya dari komponen-komponen tersebut dapat dieliminasi sepenuhnya selama periode *standby*.

### 3.1.2 Perancangan *Power Controller*



Gambar 3.3 Perancangan *Power Controller*

Untuk mengatasi inefisiensi pemanenan daya pada V1, sistem *power supply* V2 dirancang ulang dengan menggunakan IC CN3791 sebagai kontroler pengisian daya surya. Pemilihan IC ini didasarkan pada dua

justifikasi utama yang sangat relevan untuk aplikasi *IoT node* berdaya rendah.

Pertama, IC CN3791 mengimplementasikan algoritma Maximum Power Point Tracking. Penggunaan MPPT sangat krusial untuk panel surya berukuran kecil yang dayanya sangat dipengaruhi oleh kondisi pencahayaan. IC ini secara konstan menyesuaikan impedansi inputnya untuk memastikan panel surya selalu beroperasi pada titik tegangan dan arus di mana ia menghasilkan daya paling maksimal. Berbeda dengan *linear charger* pada V1 yang membuang kelebihan tegangan sebagai panas, MPPT secara efisien mengubah seluruh daya yang tersedia menjadi arus pengisian yang optimal untuk baterai, sehingga secara signifikan meningkatkan efisiensi pemanenan energi, terutama pada kondisi berawan atau saat sudut matahari tidak ideal.

Kedua, IC CN3791 dirancang dengan mempertimbangkan efisiensi saat tidak ada proses pengisian. Berdasarkan datasheet-nya, IC ini memiliki *sleep mode* berdaya sangat rendah. Mode ini aktif secara otomatis ketika tegangan input dari panel surya turun di bawah tegangan baterai misalnya pada malam hari atau saat cuaca sangat gelap. Dalam kondisi *sleep* ini, konsumsi arus dari IC itu sendiri hanya sekitar  $9\mu\text{A}$ . Fitur ini sangat penting karena memastikan bahwa sirkuit pengisian daya tidak menjadi beban parasitik yang menguras baterai saat tidak ada energi yang bisa dipanen, menjaga energi yang tersimpan untuk operasional *node*.

Untuk menyediakan tegangan operasional yang stabil bagi komponen-komponen *node*, regulator LDO MIC5219-3.3V digunakan dalam desain V2. Regulator dikonfigurasi untuk menghasilkan keluaran 3.3V yang menjadi jalur daya utama untuk mikrokontroler dan sensor.

Berdasarkan datasheet, LDO MIC5219 memiliki arus diam tipikal hanya 100 nA (0.1  $\mu$ A). Karena *node* akan menghabiskan lebih dari 99% waktunya dalam *Standby*, konsumsi daya dari regulator itu sendiri menjadi faktor yang sangat dominan. Dengan arus diam yang sangat rendah, LDO MIC5219 memastikan bahwa regulator tidak menjadi sumber pemborosan energi yang signifikan selama periode tidak aktif yang panjang. Namun, keluarannya kini dibagi menjadi dua jalur:

1. Region VCC (Always-On): Jalur daya ini selalu aktif dan ditenagai langsung oleh LDO. Region ini secara khusus didedikasikan untuk sub-sistem *pulse counter* yang harus tetap beroperasi 24/7, bahkan saat mikrokontroler mati total. Komponen yang ditenagai oleh VCC meliputi sensor Hall-effect DRV5032 dan IC binary counter CD4040BM.
2. Region VDD (Power-Gated): Jalur daya ini adalah domain utama sistem yang mencakup semua komponen boros daya, termasuk mikrokontroler STM32U585, modul LoRa (RFM95), dan sensor-sensor I2C (SHT20, LUX, I/O expander PCF8575). Jalur ini akan dimatikan total selama periode tidur.

Implementasi *power gating* utama kini ditangani oleh arsitektur *timer* eksternal. Sistem ini menggunakan IC *timer* TPL5111 yang dikombinasikan dengan IC *load switch* TPS22918. Mekanisme kerjanya adalah sebagai berikut:

1. IC TPL5111 berfungsi sebagai "jam alarm" utama sistem. Interval tidurnya (misalnya, 1 jam) diatur oleh resistor eksternal. IC TPL5111 ditenagai oleh region VCC yang *always-on* dan memiliki konsumsi daya sangat rendah.

2. *Load switch* TPS22918 berfungsi sebagai gerbang daya (*high-side*) utama untuk seluruh region VDD. Saklar ini dikontrol oleh TPL5111.

Adapun siklus operasi nya sebagai berikut:

- Selama periode *standby*, IC TPL5111 menjaga *load switch* TPS22918 dalam kondisi mati, sehingga seluruh region VDD mati total sehingga konsumsi arus 0.
- Ketika timer berakhir, IC TPL5111 menyalakan *load switch* TPS22918.
- Region VDD kini mendapat daya, dan mikrokontroler STM32U585 akan booting dari awal.
- Mikrokontroler menjalankan semua tugasnya, membaca sensor, membaca nilai counter dari IC PCF8575, dan mengirim data LoRa.
- Setelah selesai, mikrokontroler mengirim sinyal HIGH ke pin DONE pada IC TPL5111.
- Menerima sinyal DONE, IC TPL5111 segera mematikan *load switch* TPS22918, memutus total daya ke mikrokontroler dan mengembalikan sistem ke kondisi *standby*.

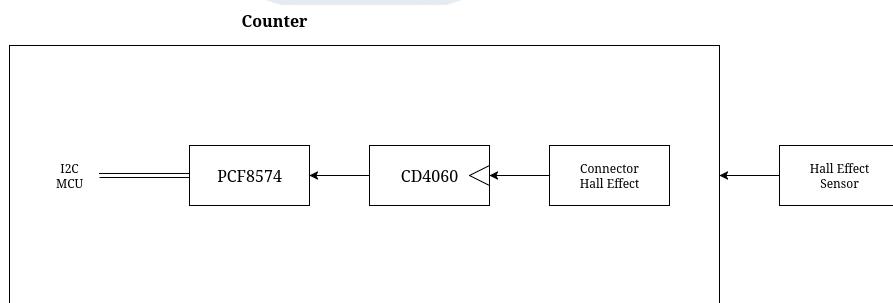
Arsitektur ini secara efektif mencapai efisiensi daya siaga maksimum, karena konsumsi daya keseluruhan sistem saat tidur tidak lagi bergantung pada *sleep* MCU, melainkan hanya pada konsumsi arus di level mikroampere dari IC TPL5111, *load switch* TPS22918 dan sirkuit *counter*.

Untuk memungkinkan *node* memantau status dayanya sendiri, desain V2 mengintegrasikan sirkuit untuk mengukur dua tegangan krusial, yaitu tegangan baterai dan tegangan panel surya. Pengukuran ini penting untuk mengetahui sisa kapasitas baterai (*State of Charge*) dan untuk mengevaluasi performa di lapangan.

Karena pin *Analog-to-Digital Converter* (ADC) pada mikrokontroler STM32U585 memiliki batas tegangan masukan (umumnya 3.3V, tergantung dari *reference voltage* VDDA), maka tegangan dari baterai dan panel surya yang lebih tinggi harus diturunkan terlebih dahulu. Untuk tujuan ini, dua rangkaian *voltage divider* yang identik diimplementasikan. Masing-masing rangkaian terhubung ke pin ADC yang berbeda dan berfungsi untuk menurunkan tegangan ke rentang yang aman untuk diukur.

Untuk meminimalkan konsumsi daya, rangkaian pembagi tegangan ini dirancang menggunakan resistor bernilai sangat tinggi. Pemilihan resistor bernilai tinggi ini sangat krusial karena memastikan bahwa arus yang terus-menerus mengalir melalui rangkaian pembagi ini sangat kecil (di level nanoampere), sehingga tidak menjadi sumber kebocoran arus yang signifikan dan menjaga efisiensi daya sistem secara keseluruhan.

### 3.1.3 Perancangan Counter



Gambar 3.4 Perancangan Counter untuk menghitung *pulse*

Untuk mengatasi inefisiensi dari penggunaan ULP *co-processor* pada V1, arsitektur V2 mengadopsi pendekatan yang lebih radikal dan efisien yaitu dengan mematikan total mikrokontroler STM32U585 dan mengalihkan tugas penghitungan *pulse* ke sirkuit *counter* eksternal yang dirancang khusus untuk tugas ini yang dirancang seperti gambar 3.4. Meskipun STM32U585 memiliki mode-mode

berdaya sangat rendah seperti LPBAM, tujuan utamanya adalah mencapai kondisi daya-mati total untuk mengeliminasi bahkan konsumsi arus di level nanoampere sekalipun, sehingga memaksimalkan efisiensi daya secara absolut.

Sirkuit *counter* eksternal ini terdiri dari tiga komponen:

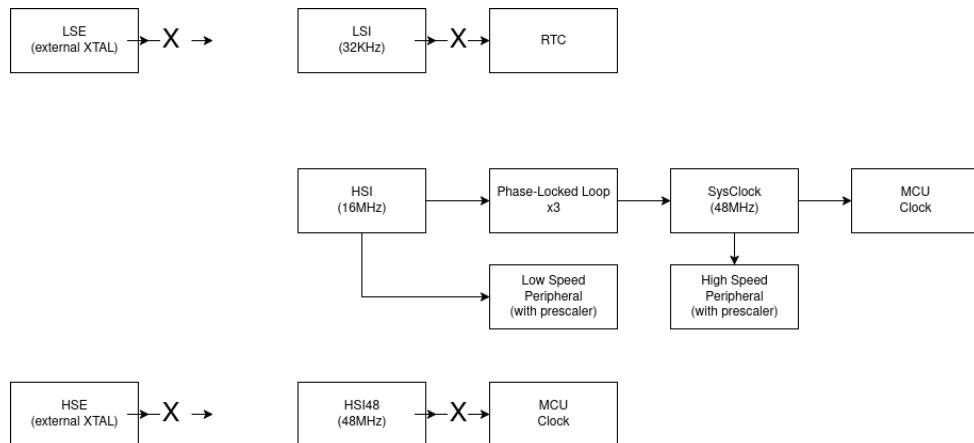
1. *Hall-effect switch* DRV5032. Pemilihan komponen ini merupakan peningkatan krusial dari arsitektur V1, yang sebelumnya menggunakan modul berbasis sensor A3144 dan komparator LM393 yang relatif boros daya (mengonsumsi di orde miliampere).
2. *Binary Counter* IC CD4040 adalah sebuah *12-stage binary ripple counter* berdaya sangat rendah yang berfungsi sebagai otak dari sub-sistem ini. *Pulse* dari sensor *Hall-effect tipping bucket* langsung ke pin *clock* dari IC ini. CD4040 akan secara mandiri dan terus-menerus menghitung setiap *pulse* yang masuk, dengan konsumsi daya yang sangat minimal (dalam orde microampere), bahkan saat seluruh bagian lain dari *node* dalam keadaan mati.
3. Karena keluaran dari *counter* CD4040BM bersifat paralel (beberapa pin merepresentasikan nilai biner), maka diperlukan cara yang efisien untuk membacanya. Di sinilah PCF8575, sebuah *I/O expander*, berperan. IC ini terhubung ke pin-pin keluaran *counter* dan berfungsi untuk "menerjemahkan" data paralel tersebut agar dapat diakses oleh mikrokontroler melalui antarmuka I2C yang hanya membutuhkan dua kabel.

Justifikasi efisiensi dari arsitektur ini adalah sebagai berikut saat *node* bangun (dihidupkan oleh *load switch*), STM32U585 hanya perlu dinyalakan sesaat, membaca nilai hitungan yang sudah terkumpul melalui I2C, lalu segera mematikan kembali seluruh sistem. Dengan demikian, tugas penghitungan yang berlangsung terus-menerus selama periode tidur ditangani oleh sirkuit berdaya sangat rendah, sementara mikrokontroler yang jauh lebih kompleks hanya aktif selama beberapa detik untuk mengambil hasil akhirnya. Pendekatan ini secara drastis lebih efisien daripada membuat bagian manapun dari mikrokontroler (bahkan ULP) tetap aktif selama periode siaga.

### 3.1.4 Perancangan *System Clock*

Sebagai bagian krusial dari optimalisasi *firmware*, perancangan *system clock* untuk STM32U585 akan difokuskan pada pencapaian efisiensi daya maksimum selama fase aktif. Berbeda dengan arsitektur V1 yang menggunakan ESP32 terbatas pada frekuensi operasional minimum di 80MHz, pendekatan V2 adalah menghindari penggunaan frekuensi maksimal MCU. Sistem akan dikonfigurasi untuk berjalan pada frekuensi 48MHz yang dinilai lebih dari cukup untuk kebutuhan pemrosesan data sensor dan komunikasi LoRa, namun menawarkan penghematan daya yang signifikan.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.5 Ilustrasi konfigurasi *System Clock* untuk mode hemat daya

Strategi untuk mencapai konfigurasi ini, seperti yang diilustrasikan pada gambar 3.5 dan diuraikan dalam implementasi fungsi *SystemClock\_Config*, adalah sebagai berikut:

- Hanya menggunakan osilator internal sehingga tidak membutuhkan osilator eksternal sehingga menghindari penggunaan daya berlebih.
- Menggunakan osilator internal HSI (16 MHz) yang hemat daya sebagai sumber utama, menghindari kebutuhan osilator eksternal (HSE) yang lebih boros.
- Memanfaatkan *Phase-Locked Loop* internal untuk melipatgandakan frekuensi HSI secara presisi menjadi 48 MHz untuk *system clock* (SYSCLK) dan *clock peripheral* (CLK48).
- Menerapkan *voltage scaling* ke level terendah yang diizinkan untuk frekuensi 48 MHz, yaitu *PWR\_REGULATOR\_VOLTAGE\_SCALE3*. Langkah ini

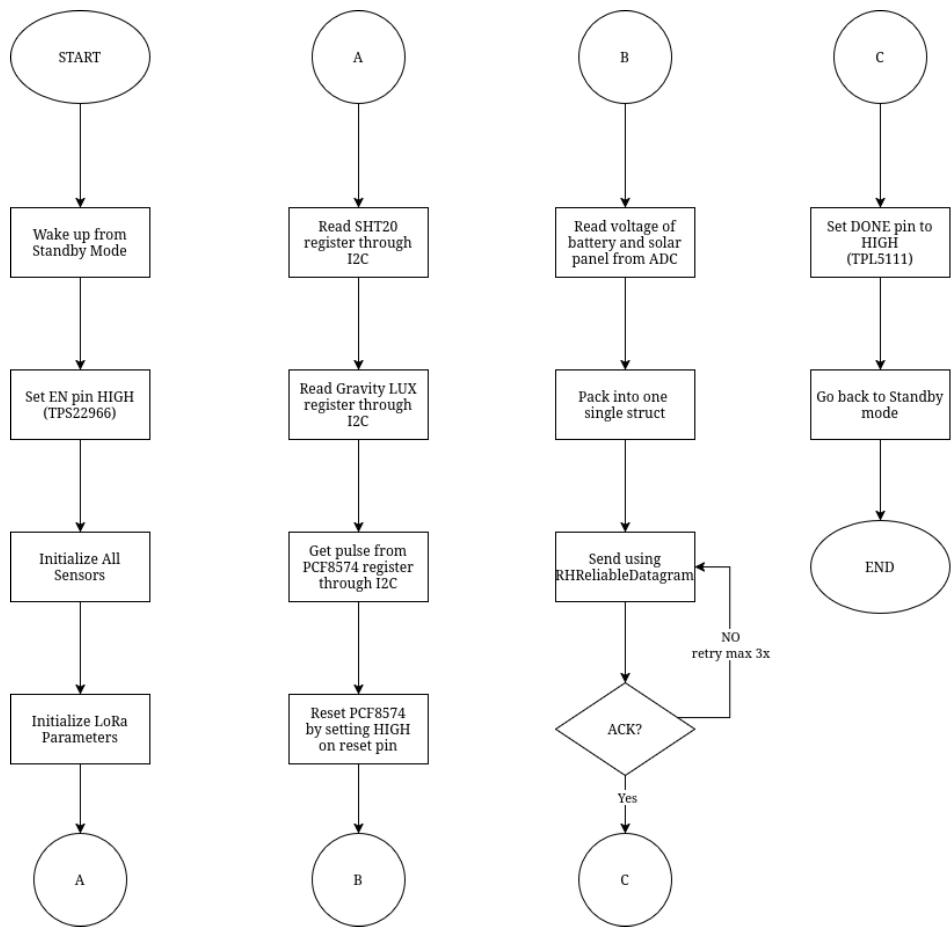
secara fundamental mengurangi konsumsi daya dinamis, karena konsumsi daya proporsional dengan tegangan.

- Mengkonfigurasi *clock peripheral* (seperti LPUART1) untuk juga menggunakan sumber HSI, serta menonaktifkan secara eksplisit semua osilator yang tidak terpakai (seperti HSI48 untuk MCU dan LSI untuk RTC) untuk mengeliminasi konsumsi daya parasitik.

### 3.1.5 Perancangan Firmware

*Firmware* untuk *node* V2 dirancang ulang sepenuhnya dari V1 dengan filosofi desain untuk meminimalkan waktu aktif dan memaksimalkan efisiensi eksekusi. Berbeda dengan V1 yang memiliki beberapa alur kondisional, *firmware* V2 berjalan dalam satu alur eksekusi linear setiap kali bangun. Alur kerja ini diilustrasikan secara visual pada diagram alur di gambar 3.6.





Gambar 3.6 Diagram alur *firmware*

Siklus operasional dimulai ketika *node* dibangunkan dan setelah semua sensor dan modul LoRa mendapatkan daya, *firmware* akan menginisialisasi antarmuka I2C dan SPI, serta melakukan konfigurasi parameter untuk modul LoRa.

Proses akuisisi data kemudian dilakukan secara berurutan:

1. Membaca data suhu/kelembapan (SHT20) melalui protokol I2C.
2. Membaca intensitas cahaya (LUX) melalui protokol I2C.
3. Membaca akumulasi hitungan *pulse* curah hujan dari *counter* eksternal melalui I/O *expander* PCF8575, dan mengirim sinyal untuk me-reset *counter* tersebut kembali ke nol.

4. Data telemetri sistem, yaitu tegangan baterai dan panel surya, juga dibaca melalui ADC untuk pemantauan kesehatan *node*.

Semua data ini kemudian dikemas ke dalam sebuah *struct payload* untuk dikirimkan melalui LoRa menggunakan *library* RadioHead dengan protokol RHReliableDatagram. *Firmware* akan menunggu *acknowledgement* dari *gateway*. Apabila tidak ada respon, proses pengiriman akan diulang hingga maksimal tiga kali. Segera setelah transmisi berhasil (atau setelah semua upaya *retry* gagal), langkah terakhir adalah mematikan kembali seluruh *peripheral* dengan pin DONE menjadi HIGH.

### 3.2 Metode Pengujian

Tujuan utama dari tahap pengujian adalah untuk memvalidasi secara kuantitatif apakah rancang bangun *node* V2 berhasil mencapai target utama yaitu peningkatan signifikan pada jumlah hari operasional jika dibandingkan dengan arsitektur V1. Untuk mencapai tujuan ini, pengujian akan difokuskan pada pengukuran profil konsumsi daya kedua sistem secara presisi. Data konsumsi daya inilah yang akan menjadi dasar untuk melakukan analisis perbandingan kuantitatif dan mengestimasi masa pakai operasional kedua node.

Untuk memastikan perbandingan yang adil, fitur anti-pencurian, GPS dan OTA pada *firmware node* V1 akan dinonaktifkan secara total. Dengan demikian, kedua *node* akan menjalankan beban kerja fungsional yang setara, yaitu hanya melakukan akuisisi data sensor lingkungan dan transmisi LoRa.

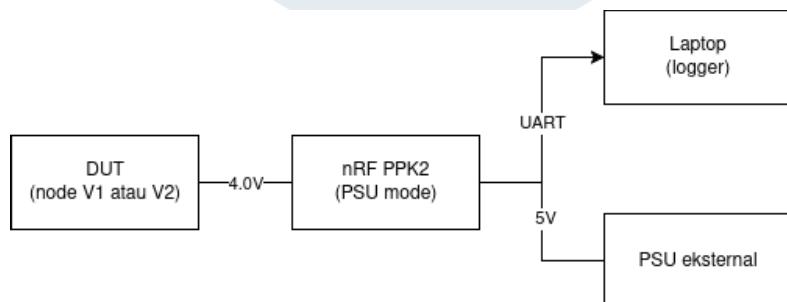
Untuk mencapai tujuan tersebut, pengujian akan dilaksanakan melalui dua skenario yang saling melengkapi yaitu pengujian laboratorium untuk pengukuran presisi dan pengujian lapangan untuk validasi sistem secara keseluruhan.

### 3.2.1 Pengujian Laboratorium (Garis Dasar Konsumsi Daya)

Pengujian laboratorium dirancang untuk mengukur dan membandingkan profil konsumsi daya secara kuantitatif antara *node* V1 dan V2. Tujuannya adalah untuk mendapatkan data presisi mengenai konsumsi arus selama satu siklus operasional penuh. Data ini akan digunakan untuk menghitung konsumsi daya rata-rata, yang kemudian menjadi dasar untuk mengestimasi masa pakai baterai secara teoretis dan memvalidasi efektivitas dari perancangan ulang yang dilakukan.

Konfigurasi pengujian, seperti yang diilustrasikan pada gambar 3.7, terdiri dari dua komponen utama:

1. Device Under Test (DUT) yaitu *Node IoT* V1 atau V2.
2. Alat ukur lab *Nordic Power Profiler Kit 2* yang mensimulasikan baterai di 4.0V yang menjadi tegangan masuk utama.



Gambar 3.7 Skematik Konfigurasi Pengujian Laboratorium

Data logger akan diprogram untuk mengambil sampel data dengan *rate* 100Hz. *Sampling rate* ini dipilih sebagai kompromi untuk dapat menangkap transisi daya yang cepat. Misalnya, saat LoRa melakukan transmisi, sambil tetap menghasilkan data yang dapat diolah.

### 3.2.2 Skenario Pengujian Laboratorium

Pengujian akan dilakukan dalam dua skenario operasional yang berbeda untuk mensimulasikan kondisi lingkungan yang beragam:

#### 1. Skenario Kondisi Normal (Tanpa Hujan)

*Node* akan dibiarkan menjalankan satu siklus operasional standarnya yaitu bangun dari mode *standby*, membaca sensor, mengirim data melalui LoRa, dan kembali ke mode *standby*. Skenario ini akan mengukur profil *baseline* dari *node*.

#### 2. Skenario Simulasi Hujan

*Node* akan menjalankan siklus operasional yang sama dengan Skenario A, namun dengan tambahan simulasi peristiwa curah hujan. Untuk menciptakan kondisi pengujian yang lebih realistik yang meniru aktuasi sensor fisik, sebuah alat uji (*test jig*) khusus telah dirancang. Alat ini menggunakan motor *servo* untuk menggerakkan mekanisme jungkat-jungkit yang dilengkapi magnet. Magnet ini akan memicu sensor *Hall-effect* pada *node* secara periodik setiap 1 menit sekali, sebuah frekuensi yang dipilih untuk merepresentasikan kondisi "hujan sedang" secara konsisten. Skenario ini bertujuan untuk mengukur dan membandingkan secara langsung dampak konsumsi daya dari arsitektur penghitungan *pulse* yang berbeda, yaitu antara ULP *co-processor* (pada V1) dan sirkuit *counter* eksternal (pada V2).

### 3.2.3 Metrik dan Analisis Data Pengujian Laboratorium

Data mentah yang berupa *timeseries* konsumsi arus akan dianalisis untuk mendapatkan metrik-metrik berikut:

- Arus rata-rata ( $I_{avg}$ ) yaitu konsumsi arus rata-rata selama satu siklus penuh. Ini adalah metrik utama untuk menghitung masa pakai baterai.
- Arus puncak ( $I_{peak}$ ) konsumsi arus tertinggi yang tercatat saat transmisi LoRa.

Masa pakai baterai teori akan diestimasi menggunakan formula berikut, dengan asumsi total kapasitas baterai adalah 7000mAh,

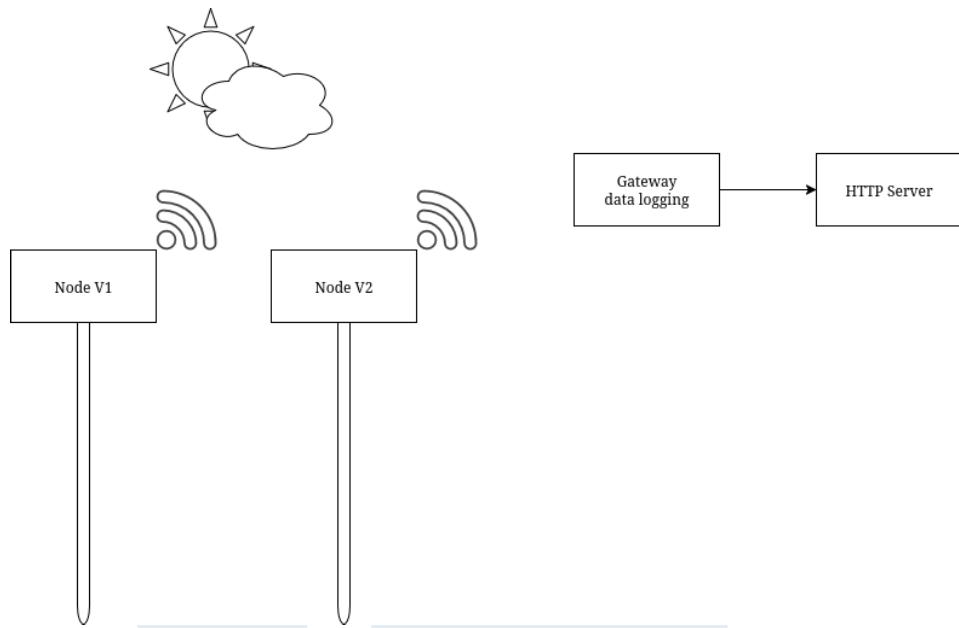
$$Battery\ Life\ (hours) = \frac{C_{battery}\ (mAh)}{I_{avg}\ (mA)}$$

Hasil dari kedua *node* (V1 dan V2) pada kedua skenario akan disajikan dalam bentuk tabel perbandingan dan plot grafik untuk memvisualisasikan perbedaan profil konsumsi daya secara jelas.

### 3.2.4 Pengujian Lapangan

Pengujian lapangan bertujuan untuk memvalidasi dan membandingkan kinerja *power budget* secara keseluruhan antara *node* V1 dan V2 dalam kondisi operasional dunia nyata. Seperti yang diilustrasikan pada gambar 3.8, pengujian ini dilakukan di lingkungan kampus Universitas Multimedia Nusantara, Tangerang, Banten.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.8 Konfigurasi pengujian lapangan komparatif

Penulis menyadari sepenuhnya bahwa lokasi pengujian ini memiliki karakteristik iklim tropis yang tidak identik dengan lokasi implementasi target di Sleman, Yogyakarta. Namun, validitas pengujian untuk tujuan penelitian ini tetap terjaga karena sifatnya yang komparatif. Kedua unit, *node* V1 dan V2, diuji secara simultan dan berdampingan di bawah kondisi lingkungan yang sama persis. Dengan demikian, semua variabel eksternal seperti intensitas cahaya matahari, suhu, dan curah hujan memberikan pengaruh yang setara pada kedua perangkat.

Fokus dari pengujian ini bukanlah untuk mendapatkan data absolut masa pakai baterai di Sleman, Yogyakarta. Melainkan untuk membuktikan secara adil dan kuantitatif seberapa besar peningkatan ketahanan operasional V2 relatif terhadap V1. Periode pengujian ditetapkan selama maksimal satu bulan, namun, pengujian dapat dianggap selesai lebih awal apabila salah satu dari dua kondisi berikut terpenuhi:

1. Telah muncul tren degradasi tegangan yang jelas, konsisten, dan signifikan antara *node* V1 dan V2.
2. Terjadi anomali kritis, seperti kegagalan operasional total pada salah satu *node* (misalnya, baterai habis sepenuhnya).

Pendekatan ini memastikan bahwa kesimpulan ditarik berdasarkan data yang sudah cukup representatif, bukan berdasarkan durasi waktu yang arbitrer.

### 3.2.5 Konfigurasi Pengujian Lapangan

Untuk menjamin validitas perbandingan (*apple-to-apple comparison*), kedua unit uji (*Node V1* dan *Node V2*) dikonfigurasi dengan spesifikasi sumber daya yang identik. Masing-masing *node* dilengkapi dengan panel surya yang sama dan paket baterai Li-Ion yang terdiri dari dua baterai 18650 yang disusun secara paralel, dengan total kapasitas 7000 mAh. Sebelum pengujian dimulai, baterai pada kedua *node* dipastikan dalam kondisi terisi penuh.

Mengingat *Node V1* tidak memiliki fitur pemantauan tegangan bawaan, dilakukan modifikasi perangkat keras dengan menambahkan rangkaian pembagi tegangan (*voltage divider*) eksternal yang terhubung ke pin ADC ESP32. Modifikasi ini bertujuan untuk menyetarakan kapabilitas telemetri V1 dengan V2, yang sudah memiliki fitur monitoring tegangan baterai dan panel surya secara *native* dalam desainnya.

Selama periode pengujian, kedua *node* menjalankan *firmware* operasional normal dengan seluruh sensor aktif. Pengujian dikonfigurasi sebagai berikut:

- *Node* beroperasi secara otonom dengan interval setiap satu jam sekali operasi normal.

- Pada setiap siklus aktif, *node* membaca data dari seluruh sensor lingkungan. Selain data standar (suhu, kelembapan, curah hujan, intensitas cahaya), struktur paket data (*payload*) LoRa dimodifikasi untuk menyertakan data telemetri kritis, yaitu tegangan baterai aktual dan tegangan panel surya saat itu.
- Data yang diterima oleh *gateway* di laboratorium akan disimpan ke *database* lengkap dengan *timestamp* penerimaan untuk keperluan analisis *timeseries*.

Untuk menguji ketahanan sistem di bawah kondisi pencahayaan yang beragam, pengujian lapangan dilaksanakan di dua lokasi berbeda di area Universitas Multimedia Nusantara:

1. Lokasi A (Kondisi Penyinaran Optimal), ditempatkan di area terbuka Lantai 3 Gedung C. Lokasi ini merepresentasikan kondisi ideal di mana panel surya mendapatkan paparan sinar matahari langsung sepanjang hari tanpa adanya halangan atau naungan dari vegetasi sama sekali.





Gambar 3.9 Dokumentasi penempatan *node* di lokasi atap terbuka  
(Gedung C)

2. Lokasi B (Kondisi Penyinaran Terhalang/Naungan): Ditempatkan di area taman dekat parkir mobil Gedung A. Lokasi ini dipilih karena memiliki densitas vegetasi yang tinggi, di mana tajuk pohon menutupi sebagian besar cahaya matahari yang masuk. Kondisi ini mensimulasikan skenario "buruk" di perkebunan salak yang rimbun, yang bertujuan untuk menguji efektivitas fitur MPPT pada *node* V2 dalam memanen daya dari cahaya difusi atau redup dibandingkan dengan sistem pengisian linear pada *node* V1.



Gambar 3.10 Dokumentasi penempatan *node* di lokasi taman bervegetasi lebat (Gedung A)

### 3.2.6 Metrik dan Analisis Data Pengujian Lapangan

Data tegangan baterai yang terkumpul dari kedua *node* di dua tempat tersebut akan diplot ke dalam sebuah grafik deret waktu serta analisis dan uji komparatif berdasarkan normalitas data. Analisis akan difokuskan pada:

- Menganalisis tren untuk membandingkan laju degradasi tegangan pada dan efektivitas pengisian ulang antara *node* V1 dan *node* V2.
- Menganalisis data tegangan secara statistik untuk membuktikan perbedaan kapabilitas kedua node dalam mempertahankan

kapasitas baterai (diukur sebagai level tegangan), dan memvalidasi apakah V2 secara signifikan lebih unggul.

