# PENERAPAN ALGORITMA *PRUNED EXACT LINEAR TIME* (PELT) DALAM DEDUPLIKASI DATA SENSOR PADA SISTEM PEMANTAUAN TANAMAN ANGGUR

**VIKTORIUS VALENTINO**

**H1051201050**

**SKRIPSI**

**PROGRAM STUDI REKAYASA SISTEM KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS TANJUNGPURA**

**PONTIANAK**

**2025**

# PENERAPAN ALGORITMA *PRUNED EXACT LINEAR TIME* (PELT) DALAM DEDUPLIKASI DATA SENSOR PADA SISTEM PEMANTAUAN TANAMAN ANGGUR

**VIKTORIUS VALENTINO**

**H1051201029**

Skripsi
Sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
pada Program Studi Rekayasa Sistem Komputer

**PROGRAM STUDI REKAYASA SISTEM KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS TANJUNGPURA**

**PONTIANAK**

**2025**

## LEMBAR PENGESAHAN

| | | |
|---|---|---|
| Judul Tugas Akhir | : | Penerapan Algoritma *Pruned Exact Linear Time* (PELT) Dalam Deduplikasi Data Sensor Pada Sistem Pemantauan Tanaman Anggur |
| Nama Mahasiswa | : | Viktorius Valentino |
| NIM | : | H1051201050 |
| Jurusan/Program Studi | : | Rekayasa Sistem Komputer |
| Tanggal Lulus | : | 3 Maret 2025 |
| SK Pembimbing | : | No.904/UN22.8/TD.06/2024 /Tanggal 21 Maret 2024 |
| SK Penguji | : | No.683/UN22.8/TD.06/2025 /Tanggal 27 Februari 2025 |

Dosen Pembimbing

| Pembimbing 1 | Pembimbing 2 |
|---|---|
| Uray Ristian, S.Kom., M.Kom. | Dwi Marisa Midyanti, S.T., M.Cs. |
| 199012012019031017 | 198003192015042001 |

Dosen Penguji

| Ketua Penguji | Anggota Penguji |
|---|---|
| Syamsul Bahri, S.Kom., M.Cs. | Hafiz Muhardi, S.T., M.Kom. |
| 198802272015041001 | 199007232023211017 |

| Pimpinan Sidang (merangkap anggota penguji) | Sekretaris Sidang (merangkap anggota penguji) |
|---|---|
| Uray Ristian, S.Kom., M.Kom. | Dwi Marisa Midyanti, S.T., M.Cs. |
| 199012012019031017 | 198003192015042001 |

Mengesahkan
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Tanjungpura

Prof. Dr. Gusrizal, S.Si., M.Si.
NIP. 197108022000031001

## PERNYATAAN INTEGRITAS AKADEMIK

Saya yang bertanda tangan di bawah ini:

| | |
|---|---|
| Nama | : Viktorius Valentino |
| NIM | : H1051201050 |
| Program Studi/ Jurusan | : Rekayasa Sistem Komputer |
| Fakultas | : Matematika dan Ilmu Pengetahuan Alam |

dengan ini menyatakan bahwa dokumen ilmiah Tugas Akhir yang disajikan ini tidak mengandung unsur pelanggaran integritas akademik sesuai Peraturan Menteri Pendidikan, Kebudayaan, Riset, Dan Teknologi Republik Indonesia Nomor 39 Tahun 2021. Apabila di kemudian hari dokumen ilmiah Tugas Akhir ini mengandung unsur pelanggaran integritas akademik sesuai ketentuan perundangan tersebut, maka saya bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Demikian pernyatan ini untuk digunakan sebagaimana mestinya.

Pontianak, 3 Maret 2025




Viktorius Valentino
H1051201050

# PENERAPAN ALGORITMA *PRUNED EXACT LINEAR TIME* (PELT) DALAM DEDUPLIKASI DATA PADA SISTEM PEMANTAUAN TANAMAN ANGGUR

## ABSTRAK

Penerapan teknologi *Internet of Things* (IoT) dalam sistem pemantauan tanaman anggur menghasilkan volume data sensor yang besar, dengan banyak data yang memiliki nilai serupa secara berurutan. Hal ini menyebabkan peningkatan signifikan dalam kebutuhan penyimpanan di *cloud server*. Penelitian ini mengimplementasikan algoritma *Pruned Exact Linear Time* (PELT) untuk mendeteksi titik perubahan (*changepoint*) dalam data guna mengurangi duplikasi dan mengoptimalkan penggunaan penyimpanan. Hasil penelitian menunjukkan bahwa deduplikasi dengan PELT menggunakan *penalty Akaike Information Criterion* (AIC) berhasil menurunkan jumlah data dari 2.203.730 menjadi 249.730 baris, sementara dengan *Bayesian Information Criterion* (BIC), jumlahnya berkurang lebih drastis menjadi 115.939 baris. Selain itu, ukuran data yang awalnya 206,476 MB berkurang menjadi 23,382 MB dengan AIC dan 10,86 MB dengan BIC. Berdasarkan hasil pengurangan tersebut, efisiensi kompresi yang diperoleh dengan *penalty* AIC mencapai efisiensi sebesar 88,68% dan *penalty* BIC mencapai 94,7%. Hasil ini membuktikan bahwa algoritma PELT efektif dalam mendeteksi titik perubahan untuk deduplikasi data sensor, sehingga secara signifikan mengurangi kebutuhan penyimpanan di *cloud server*.

**Kata kunci:** *Internet of Things* (IoT), deduplikasi data, algoritma PELT, titik perubahan, efisiensi penyimpanan.

***Implementation of the Pruned Exact Linear Time (PELT) Algorithm for Sensor Data Deduplication in a Vine Monitoring System***


## *Abstract*

*The implementation of Internet of Things (IoT) technology in vine monitoring systems results in large volumes of sensor data, with many data having similar values in quick succession. This leads to a significant increase in storage requirements on cloud servers. This research implements the Pruned Exact Linear Time (PELT) algorithm to detect changepoints in the data to reduce duplication and optimise storage usage. The results showed that deduplication with PELT using the Akaike Information Criterion (AIC) penalty successfully reduced the amount of data from 2,203,730 to 249,730 rows, while with the Bayesian Information Criterion (BIC), the amount was reduced more drastically to 115,939 rows. In addition, the original data size of 206.476 MB was reduced to 23.382 MB with AIC and 10.86 MB with BIC. Based on the reduction results, the compression efficiency obtained with penalty AIC reached an efficiency of 88.68% and penalty BIC reached 94.7%. These results prove that the PELT algorithm is effective in detecting change points for sensor data deduplication, thus significantly reducing storage requirements on the cloud server.*



***Keywords*****:** *Internet of Things (IoT), data deduplication, PELT algorithm, changepoints, storage efficiency.*

# PRAKATA

Puji dan syukur kepada Tuhan YME, atas berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas penulisan skripsi pada Program Studi Rekayasa Sistem Komputer tahun 2025 dengan judul "PENERAPAN ALGORITMA *PRUNED EXACT LINEAR TIME* (PELT) DALAM DEDUPLIKASI DATA SENSOR PADA SISTEM PEMANTAUAN TANAMAN ANGGUR" sebagai syarat bagi penulis untuk mendapatkan gelar Sarjana Komputer (S. Kom) pada Jurusan Rekayasa Sistem Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tanjungpura Pontianak.

Semua hasil yang diperoleh ini tidak terlepas dari segala dukungan dan bantuan dari berbagai pihak yang tidak pernah berhenti mendorong penulis agar dapat menyelesaikan tugas akhir ini. Sebagai penghargaan dari penulis, izinkanlah penulis mengucapkan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Gusrizal, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tanjungpura Pontianak.
2. Bapak Tedy Rismawan, S.Kom., M.Cs., selaku Ketua Jurusan Rekayasa Sistem Komputer FMIPA Universitas Tanjungpura Pontianak.
3. Bapak Uray Ristian, S.Kom., M.Kom., selaku dosen Pembimbing Pertama yang sudah memberi masukan serta meluangkan waktu dan pikiran yang memberikan bimbingan dalam penulisan skripsi.
4. Ibu Dwi Marisa Midyanti, S.T., M.Cs., selaku dosen Pembimbing Kedua sekaligus dosen Pembimbing Akademik (PA) yang sudah meluangkan waktu dan pikiran dalam bimbingan selama kuliah dan juga dalam penulisan skripsi.
5. Bapak Syamsul Bahri, S.Kom., M.CS., selaku dosen Penguji Pertama, yang telah memberikan kontribusi dengan memberikan masukan dan pertanyaan yang memperkaya isi dari tugas akhir ini.
6. Bapak Hafiz Muhardi, S.T., M.Kom., selaku dosen Penguji Kedua, yang telah memberikan kontribusi dengan memberikan masukan dan pertanyaan yang memperkaya isi dari tugas akhir ini.
7. Kedua orang tua yang telah memberikan dukungan moral maupun dukungan material dalam proses penyelesaian skripsi ini.

8. Teman-teman Batu's Ahmad Abdul Hanafi, Gustavo Dipo Elegant, Indra Sukma Fratama, Gregorius Valentine dan Muhammad Wira Perdana yang memberi semangat dan saran dalam tugas akhir ini sehingga dapat terselesaikan.
9. Helaria Flafiana Ena yang selalu memberikan semangat dan dukungan dalam penulisan tugas akhir ini.
10. Teman-teman Scicosome yang memberi semangat dan saran dalam tugas akhir ini sehingga dapat terselesaikan.
11. Teman-teman seperjuangan terutama angkatan 2020 (SyComAs) yang telah memberikan dukungan, semangat dan membantu dalam mengerjakan tugas akhir ini.
12. Semua pihak yang terlibat langsung maupun yang tidak langsung dalam penulisan skripsi ini.

Penulis menyadari dalam penyusunan skripsi ini masih memiliki kekurangan dan belum sempurna. Oleh karena itu, penulis mengharapkan saran dan kritik yang bersifat membangun. Semoga skripsi ini dapat memberikan manfaat bagi penulis maupun pembaca.

Pontianak, 3 Maret 2025



Viktorius Valentino

# DAFTAR ISI

LEMBAR PENGESAHAN ..... ii
PERNYATAAN INTEGRITAS AKADEMIK ..... iii
ABSTRAK ..... iv
*Abstract* ..... v
PRAKATA ..... vi
DAFTAR ISI ..... viii
DAFTAR GAMBAR ..... xii
DAFTAR TABEL ..... xiv
DAFTAR KODE PROGRAM ..... xvi
BAB 1 PENDAHULUAN ..... 1
1.1 Latar Belakang ..... 1
1.2 Rumusan Masalah ..... 3
1.3 Tujuan Penelitian ..... 4
1.4 Batasan Masalah ..... 4
1.5 Manfaat Penelitian ..... 4
1.6 Sistematika Penulisan ..... 5
BAB 2 DASAR TEORI ..... 6
2.1 Perbandingan dengan penelitian perdahulu ..... 6
2.2 Mikrokontroler ESP32 ..... 7
2.3 Sensor DHT11 ..... 8
2.4 Sensor *Soil Moisture* ..... 8
2.5 *Pruned Exact Linear Time* (PELT) ..... 9
2.6 *Aikake Information Criterion* (AIC) ..... 12
2.7 *Bayesian Information Criterion* (BIC) ..... 13
2.8 *Cloud Server* ..... 14
2.9 Laravel ..... 14
2.10 MySQL ..... 15
*2.11 Job Queue* ..... 15
2.12 Efisiensi Kompresi ..... 15
BAB 3 METODE PENELITIAN ..... 17
3.1 Studi Literatur ..... 18

3.2 Analisis Kebutuhan ..... 18
3.2.1 Kebutuhan Perangkat Keras ..... 18
3.2.2 Kebutuhan Perangkat Lunak ..... 18
3.3 Perancangan Sistem ..... 18
3.3.1 Perancangan Perangkat Keras ..... 19
3.3.2 Perancangan Perangkat Lunak ..... 19
3.4 Implementasi ..... 19
3.5 Pengujian ..... 20
3.6 Data Hasil Pengujian ..... 20
3.7 Kesimpulan dan Saran ..... 20
BAB 4 PERANCANGAN ..... 21
4.1 Deskripsi Penelitian ..... 21
4.2 Perancangan Arsitektur Sistem ..... 21
4.3 Perancangan Perangkat Keras ..... 23
4.3.1 Perancangan Sistem Pemantauan Kelembapan Tanah ..... 23
4.3.2 Perancangan Sistem Pemantauan Suhu Dan Kelembapan Udara ..... 25
4.3.3 Perancangan Keseluruhan Perangkat Keras ..... 26
4.4 Perancangan Perangkat Lunak ..... 27
4.5 Perancangan Basis Data ..... 29
4.5.1 Tabel *Users* ..... 32
4.5.2 Tabel *Nodes* ..... 33
4.5.3 Tabel *Plant Pots* ..... 33
4.5.4 Tabel *Node Send Logs* ..... 34
4.5.5 Tabel *Soil Moistures* ..... 35
4.5.6 Tabel *Air Temperatures* ..... 35
4.5.7 Tabel *Humidities* ..... 36
4.5.8 Tabel *Deduplication Settings* ..... 36
4.5.9 Tabel *Deduplication Histories* ..... 37
4.6 Perancangan Antar Muka ..... 38
4.6.1 Perancangan Halaman *Log In* ..... 38
4.6.2 Perancangan Halaman Dashboard ..... 39
4.6.3 Perancangan Halaman Log Pengiriman ..... 39
4.6.4 Perancangan Halaman Deduplikasi ..... 40
4.6.5 Perancangan Halaman Pengaturan *Node* ..... 41

4.6.6 Perancangan Halaman Pengaturan Pot Tanaman ..... 42
4.6.7 Perancangan Halaman Pengaturan *User* ..... 42
4.6.8 Perancangan Halaman *Profile* ..... 43
*4.7* Perancangan *Black Box* ..... 43
BAB 5 IMPLEMENTASI, PENGUJIAN DAN PEMBAHASAN ..... 45
5.1 Implementasi Perangkat Keras ..... 45
5.1.1 Implementasi Sistem Pemantauan Pada NodeMCU ESP32 Dan DHT11 45
5.1.2 Implementasi Sistem Pemantauan Pada NodeMCU ESP32 Dan Sensor *Soil Moisture* ..... 46
5.1.3 Implementasi Keseluruhan Perangkat Keras ..... 46
5.2 Implementasi Perangkat Lunak ..... 47
5.2.1 Implementasi Sistem Komunikasi Antara Mikrokontroller Dan *Cloud Server*..... 47
5.2.2 Implementasi Antarmuka *Website* ..... 48
5.2.3 Implementasi Basis Data..... 54
5.2.4 Implementasi Algoritma PELT Pada Sistem Deduplikasi..... 56
5.3 Implementasi Kode Program..... 58
5.3.1 Kode Program Sistem Pemantauan Pada Mikrokontroller dan Sensor DHT11 58
5.3.2 Kode Program Sistem Pemantauan pada Mikrokontroller dan Sensor *Soil Moisture* ..... 59
5.3.3 Kode Program Penerimaan dan Penyimpanan Data Sensor DHT11 Pada *Cloud Server*..... 61
5.3.4 Kode Program Penerimaan dan Penyimpanan Data Sensor *Soil Moisture* Pada *Cloud Server* ..... 62
5.3.5 Kode Program Algoritma *Pruned Exact Linear Time*..... 63
5.3.6 Kode Program *Job Queue* ..... 65
5.3.7 Kode Program Pencatatan Riwayat Deduplikasi ..... 70
5.3.8 Kode Program *Artisan* Untuk Menjalankan Sistem Deduplikasi ... 70
5.4 Pengujian..... 71
5.4.1 Pengujian Jumlah Baris Data Sensor Pada Setiap *Node*..... 72
5.4.2 Pengujian Ukuran Data Sensor Pada Setiap *Node*..... 81
5.4.3 Pengujian *Black Box* ..... 90
5.5 Pembahasan..... 91

BAB 6 PENUTUP ..... 94
6.1 Kesimpulan ..... 94
6.2 Saran ..... 94
DAFTAR PUSTAKA ..... 95
LAMPIRAN ..... 98

# DAFTAR GAMBAR

Gambar 2.1 Mikrokontroller ESP32 ..... 8
Gambar 2.2 Sensor DHT11 ..... 8
Gambar 2.3 Sensor *Soil Moisture* ..... 9
Gambar 2.4 Diagram Alir Cara Kerja Algoritma PELT ..... 11
Gambar 2.5 Contoh Penerapan Algoritma PELT ..... 12
Gambar 3.1 Diagram Alir Tahapan Metode Penelitian ..... 17
Gambar 4.1 Arsitektur Sistem ..... 22
Gambar 4.2 Perancangan Sistem Pemantauan Kelembapan Tanah ..... 24
Gambar 4.3 Perancangan Sistem Pemantauan Suhu Dan Kelembapan Tanah ..... 26
Gambar 4.4 Rancangan Keseluruhan Perangkat Keras ..... 27
Gambar 4.5 Diagram Alir Sistem Deduplikasi Pada Data Kelembapan Tanah ..... 28
Gambar 4.6 Diagram Alir Sistem Deduplikasi Pada Data Suhu Udara & Kelembapan Udara ..... 29
Gambar 4.7 Rancangan Basis Data Pertama, Kedua dan Ketiga ..... 31
Gambar 4.8 Rancangan Basis Data Keempat ..... 32
Gambar 4.9 Halaman *Log In* ..... 39
Gambar 4.10 Halaman Dashboard ..... 39
Gambar 4.11 Halaman Log Pengiriman ..... 40
Gambar 4.12 Halaman Beranda Deduplikasi ..... 40
Gambar 4.13 Halaman Jumlah Data Per *Node* ..... 41
Gambar 4.14 Halaman Tabel Data Per Sensor ..... 41
Gambar 4.15 Halaman Pengaturan Node ..... 42
Gambar 4.16 Halaman Pengaturan Pot Tanaman ..... 42
Gambar 4.17 Halaman Pengaturan *User* ..... 43
Gambar 4.18 Halaman *Profile* ..... 43
Gambar 5.1 Implementasi Sistem Monitoring Pada NodeMCU ESP32 Dan Sensor DHT11 ..... 45
Gambar 5.2 Implementasi Sistem Pemantauan Pada NodeMCU ESP32 Dan Sensor *Soil Moisture* ..... 46
Gambar 5.3 Implementasi Keseluruhan Perangkat Keras ..... 47

Gambar 5.4 Implementasi Sistem Komunikasi Antara Mikrokontroller Dan *Cloud Server* ..... 48
Gambar 5.5 Implementasi Halaman *Log In* ..... 49
Gambar 5.6 Implementasi Halaman *Dashboard* ..... 49
Gambar 5.7 Implementasi Halaman Log Pengiriman ..... 50
Gambar 5.8 Implementasi Halaman Beranda Deduplikasi ..... 50
Gambar 5.9 Implementasi Halaman Jumlah Baris dan Ukuran Data Per *Node*.... 51
Gambar 5.10 Implementasi Halaman Tabel Data Per Sensor ..... 51
Gambar 5.11 Implementasi Halaman Pengaturan *Node* ..... 52
Gambar 5.12 Implementasi Halaman Pengaturan Pot Tanaman ..... 52
Gambar 5.13 Implementasi Halaman Pengaturan *User* ..... 53
Gambar 5.14 Implementasi Halaman *Profile* ..... 53
Gambar 5.15 Implementasi Basis Data Pertama Dan Kedua ..... 54
Gambar 5.16 Implementasi Basis Data Ketiga ..... 56
Gambar 5.17 Cron *Job* Untuk Menjalankan Sistem Deduplikasi Pada *Cloud Server* ..... 58

# DAFTAR TABEL

Tabel 2.1 Perbandingan dengan penelitian terdahulu ..... 6
Tabel 4.1 Penggunaan Pin ESP32 Dengan Sensor *Soil Moisture* ..... 24
Tabel 4.2 Penggunaan Pin ESP32 Dengan DHT11 ..... 26
Tabel 4.3 Tabel *Users* ..... 32
Tabel 4.4 Tabel *Nodes* ..... 33
Tabel 4.5 Tabel *Plant Pots* ..... 34
Tabel 4.6 Tabel *Node Send Logs* ..... 34
Tabel 4.7 Tabel *Soil Moistures* ..... 35
Tabel 4.8 Tabel *Air Temperatures* ..... 35
Tabel 4.9 Tabel *Humidities* ..... 36
Tabel 4.10 Tabel *Deduplication Settings* ..... 36
Tabel 4.11 Tabel *Deduplication Histories* ..... 37
Tabel 4.12 Tabel *Black Box* ..... 44
Tabel 5.1 Perbandingan Jumlah Baris Data pada *Node* Pertama dengan *Penalty* AIC ..... 72
Tabel 5.2 Perbandingan Jumlah Baris Data pada *Node* Pertama dengan *Penalty* BIC ..... 72
Tabel 5.3 Perbandingan Jumlah Baris Data pada *Node* Kedua dengan *Penalty* AIC ..... 73
Tabel 5.4 Perbandingan Jumlah Baris Data pada *Node* Kedua dengan *Penalty BIC* ..... 73
Tabel 5.5 Perbandingan Jumlah Baris Data pada *Node* Ketiga dengan *Penalty* AIC ..... 74
Tabel 5.6 Perbandingan Jumlah Baris Data pada *Node* Ketiga dengan *Penalty* BIC ..... 75
Tabel 5.7 Perbandingan Jumlah Baris Data pada *Node* Keempat dengan *Penalty* AIC ..... 76
Tabel 5.8 Perbandingan Jumlah Baris Data pada *Node* Keempat dengan *Penalty* BIC ..... 76

Tabel 5.9 Perbandingan Jumlah Baris Data pada *Node* Kelima dengan *Penalty* AIC ..... 77
Tabel 5.10 Perbandingan Jumlah Baris Data pada *Node* Kelima dengan *Penalty* BIC ..... 77
Tabel 5.11 Efisiensi Kompresi Jumlah Baris Data Setiap *Node* dengan *Penalty* AIC ..... 78
Tabel 5.12 Efisiensi Kompresi Jumlah Baris Data Setiap *Node* dengan *Penalty* BIC ..... 79
Tabel 5.13 Perbandingan Ukuran Data pada Node Pertama dengan *Penalty* AIC 81
Tabel 5.14 Perbandingan Ukuran Data pada Node Pertama dengan *Penalty* BIC 81
Tabel 5.15 Perbandingan Ukuran Data pada *Node* Kedua dengan *Penalty* AIC .. 82
Tabel 5.16 Perbandingan Ukuran Data pada *Node* Kedua dengan *Penalty* BIC... 82
Tabel 5.17 Perbandingan Ukuran Data pada *Node* Ketiga dengan *Penalty* AIC .. 84
Tabel 5.18 Perbandingan Ukuran Data pada *Node* Ketiga dengan *Penalty* BIC .. 84
Tabel 5.19 Perbandingan Ukuran Data pada *Node* Keempat dengan *Penalty* AIC ..... 85
Tabel 5.20 Perbandingan Ukuran Data pada *Node* Keempat dengan *Penalty* BIC85
Tabel 5.21 Perbandingan Ukuran Data pada *Node* Kelima dengan *Penalty* AIC . 86
Tabel 5.22 Perbandingan Ukuran Data pada *Node* Kelima dengan *Penalty* BIC . 86
Tabel 5.23 Efisiensi Kompresi Total Ukuran Data Setiap *Node* dengan *Penalty* AIC ..... 87
Tabel 5.24 Efisiensi Kompresi Total Ukuran Data Setiap *Node* dengan *Penalty* BIC ..... 88
Tabel 5.25 Hasil Pengujian *Black Box* ..... 90

## DAFTAR KODE PROGRAM

Kode Program 5.1 Kode Program Sistem Pemantauan Pada Mikrokontroller dan Sensor DHT11. ..... 59
Kode Program 5.2 Kode Program Sistem Pemantauan pada Mikrokontroller dan Sensor *Soil Moisture* ..... 60
Kode Program 5.3 Kode Program Penerimaan dan Penyimpanan Data Sensor DHT11 pada *Cloud Server* ..... 61
Kode Program 5.4 Kode Program Penerimaan dan Penyimpanan Data Sensor *Soil Moisture* pada *Cloud Server* ..... 62
Kode Program 5.5 Kode Program Algoritma *Pruned Exact Linear Time* ..... 63
Kode Program 5.6 Kode Program *Job Queue* Untuk Data Suhu Udara ..... 65
Kode Program 5.7 Kode Program *Job Queue* Untuk Data Kelembapan Udara ... 67
Kode Program 5.8 Kode Program *Job Queue* Untuk Data Kelembapan Tanah ... 68
Kode Program 5.9 Kode Program Pencatatan Riwayat Deduplikasi ..... 70
Kode Program 5.10 Kode Program *Artisan* Untuk Menjalankan Sistem Deduplikasi ..... 70

# BAB 1
# PENDAHULUAN

## 1.1 Latar Belakang

Berkembangnya teknologi yang pesat mendorong perkembangan *Internet of Things* (IoT). Teknologi IoT merupakan teknologi yang memungkinkan alat-alat elektronik di sekitar kita terhubung dan berkomunikasi dengan jaringan internet (Dwiyatno et al., 2022). IoT memungkinkan pengguna untuk mengelola dan mengoptimalkan elektronik dan peralatan listrik yang menggunakan internet (Nandika & Amrina, 2021). Seiring dengan berkembangnya teknologi pada era sekarang, semakin banyak pula IoT yang diciptakan. Biasanya, perangkat IoT ini memproses data yang besar dan membutuhkan penyimpanan yang besar (Pusparini et al., 2020). Pada sistem pemantauan anggur ini dibutuhkan *database* sebagai penyimpanan data hasil sensor yang dikirim dari mikrokontroler. Data hasil sensor yang dikirim ke *cloud server* sering memiliki nilai yang serupa dalam rentang waktu berurutan dan menyebabkan penggunaan penyimpanan data pada *server* yang lumayan besar. Penggunaan penyimpanan yang menjadi besar akan meningkatkan biaya pemeliharaan *cloud server*. Dikarenakan untuk pembacaan data secara *real-time*, maka setiap data yang dikirimkan oleh mikrokontroler akan selalu disimpan pada *cloud server* dan seiring waktu akan menyebabkan beban penyimpanan pada *cloud server* meningkat. Oleh karena itu, untuk mengatasi masalah ini dibutuhkan sebuah solusi dengan deduplikasi data menerapkan algoritma *Pruned Exact Linear Time* (PELT).

Deduplikasi dilakukan dengan mengeliminasi data yang memiliki kemiripan (Hidayatullah et al., 2014). Algoritma PELT adalah algoritma yang digunakan untuk deteksi titik perubahan (*changepoint*) dalam data. Algoritma ini bekerja dengan cara meminimalkan suatu fungsi biaya yang terkait dengan jumlah dan lokasi titik perubahan yang mungkin (Dorcas Wambui, 2015). Dalam algoritma PELT, terdapat *penalty* yang merupakan biaya tambahan digunakan untuk mengontrol jumlah titik perubahan yang terdeteksi. Nilai *penalty* ini memiliki peran penting dalam menentukan sensitivitas algoritma, di mana nilai yang lebih kecil dapat menghasilkan lebih banyak titik perubahan, sedangkan nilai yang lebih besar

membuat algoritma lebih ketat dalam mendeteksi perubahan. Terdapat dua pendekatan umum dalam menentukan nilai *penalty*, yaitu *Akaike Information Criterion* (AIC) dan *Bayesian Information Criterion* (BIC) (Killick et al., 2012). Dengan demikian, algoritma PELT memungkinkan dalam mendeteksi titik-titik perubahan pada rangkaian data sensor dalam urutan waktu. Sehingga sistem ini diharapkan dapat melakukan penghematan penyimpanan data agar tidak terlalu besar pada *cloud server*.

Penelitian terkait berjudul "Sistem Pintar Untuk Anggur (Sipunggur) pada Kawasan Tropis Berbasis *Internet of Things* (IoT)" (Ruslianto, Ristian, & Hasfani, 2022). Penelitian ini membahas penerapan teknologi IoT untuk memantau dan mengontrol kondisi pertumbuhan tanaman anggur di daerah tropis. Data suhu ruangan, kelembaban tanah, dan keasaman air dipantau menggunakan sensor dan disimpan dalam *database* untuk diakses secara *online*. Hasil implementasi sistem Sipunggur memungkinkan pemantauan *real-time* kondisi lingkungan tanaman anggur melalui website. Pengguna dapat melihat data suhu, kelembaban tanah, dan keasaman air serta mengontrol pompa air melalui antarmuka aplikasi yang mudah dipahami, dapat diakses dari PC atau perangkat *mobile*.

Sementara itu, penelitian sebelumnya berjudul "*Change Point Enhanced Anomaly Detection for IoT Time Series Data"* (Apostol, Truică, Pop & Esposito, 2021). Penelitian ini mengembangkan sistem keputusan berbasis aturan yang memanfaatkan algoritma *Pruned Exact Linear Time* (PELT) untuk meningkatkan deteksi anomali dalam data deret waktu IoT. Algoritma PELT digunakan untuk menghitung solusi optimal dalam mendeteksi titik perubahan ketika jumlah titik perubahan tidak diketahui. Dalam uji coba, dataset konsumsi air dari smart meters digunakan untuk menguji algoritma deteksi anomali. Hasilnya menunjukkan bahwa pendekatan ini dapat mengurangi *false positives*, meningkatkan akurasi deteksi anomali, dan mendukung pemeliharaan prediktif dalam manajemen sumber daya air. Dengan menggabungkan algoritma PELT dengan algoritma deteksi lainnya, penelitian ini berhasil meningkatkan efisiensi deteksi anomali dalam aplikasi IoT dan manajemen sumber daya air.

Penelitian lain berjudul "*Analysis of the Response of Long-Term Vegetation Dynamics to Climate Variability Using the Pruned Exact Linear Time (PELT)*