

## DAFTAR PUSTAKA




- Azriel, F. A., Sutaji, & Deni. (2025). Pengembangan Fitur Logging Dan Dashboard Monitoring Admin Pada Layanan Chatbot Whatsapp Berbasis Laravel. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(3), 1313–1320. <https://journal.eng.unila.ac.id/index.php/jitet/article/view/8438/3842>
- Basir, S., Buckmaster, D., Raturi, A., & Zhang, Y. (2024). From pen and paper to digital precision : a comprehensive review of on - farm recordkeeping. In *Precision Agriculture* (Vol. 25, Issue 5). Springer US. <https://doi.org/10.1007/s11119-024-10172-7>
- Hakim, L., Wibowo, P. S., & Hendra, G. (2026). Penguatan Layanan Informasi dan Laporan Masyarakat di Desa Kalirejo melalui Implementasi Chatbot WhatsApp Terintegrasi Sistem Dashboard. *Jurnal Pengabdian Kepada Masyarakat*, 5(2), 446–460. <https://journal.appipgri.id/index.php/arc/article/view/272/167>
- Hidayatullah, M. M., Riana, F., Setiawan, F. A., & Djiwanti, S. R. (2026). Rancang bangun website dan chatbot informasi tanaman jahe. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 10(2), 2750–2757. <https://www.ejournal.itn.ac.id/jati/article/view/17782/9210>
- Izzulhaq, M. R., & Putra, R. R. J. (2025). Rancang Bangun Sistem Informasi Perpustakaan Terintegrasi Chatbot WhatsApp. *Digital Transformation Technology (Digitech)*, 5(1), 66–75. <https://itscience-indexing.com/jurnal/index.php/digitech/article/view/5281/4363>
- Jean, D., & Sihombing, C. (2024). Optimizing Agricultural Data Management : A Collaborative Approach through Extreme Programming in Software Development. *Jurnal Info Sains: Informasi Dan Sains*, 14(01), 12–22. <https://doi.org/10.54209/infosains.v14i01.3718>
- Lassuta, R. C. I., Oslanb, Y., & Wibowoc, A. (2024). Program Bantu Perkiraan Hasil Panen Tanaman Jagung Berbasis Web Studi Kasus : Kelompok Tani Mutiara Desa Maunggal Karya. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 6(4), 609–620. <https://jurnal.unidha.ac.id/index.php/jteksis/article/view/1253/893%0A%0A>
- Nur, E., Susanto, E., & Akbar, M. (2026). Implementasi Chatbot WhatsApp Berbasis NLP untuk Otomasi Pemesanan pada UMKM Perikanan dengan Hybrid BERT dan GPT. 6(November 2025).
- Perry, D. E. (2006). *A Case for White-box Testing Using Declarative Specifications Poster Abstract Danhua Shao Sarfraz Khurshid*. 19(7), 2006.
- Pratama, A., & Hasibuan, P. K. (2019). Sistem informasi data panen dinas pertanian. *Jurnal Sistem Informasi ISSN, P : 2598-5(E: 2599-0330)*, 63–71. <https://pdfs.semanticscholar.org/eb63/b0af042e9a23861df942e517c5849e29>

326a.pdf%0A%0A

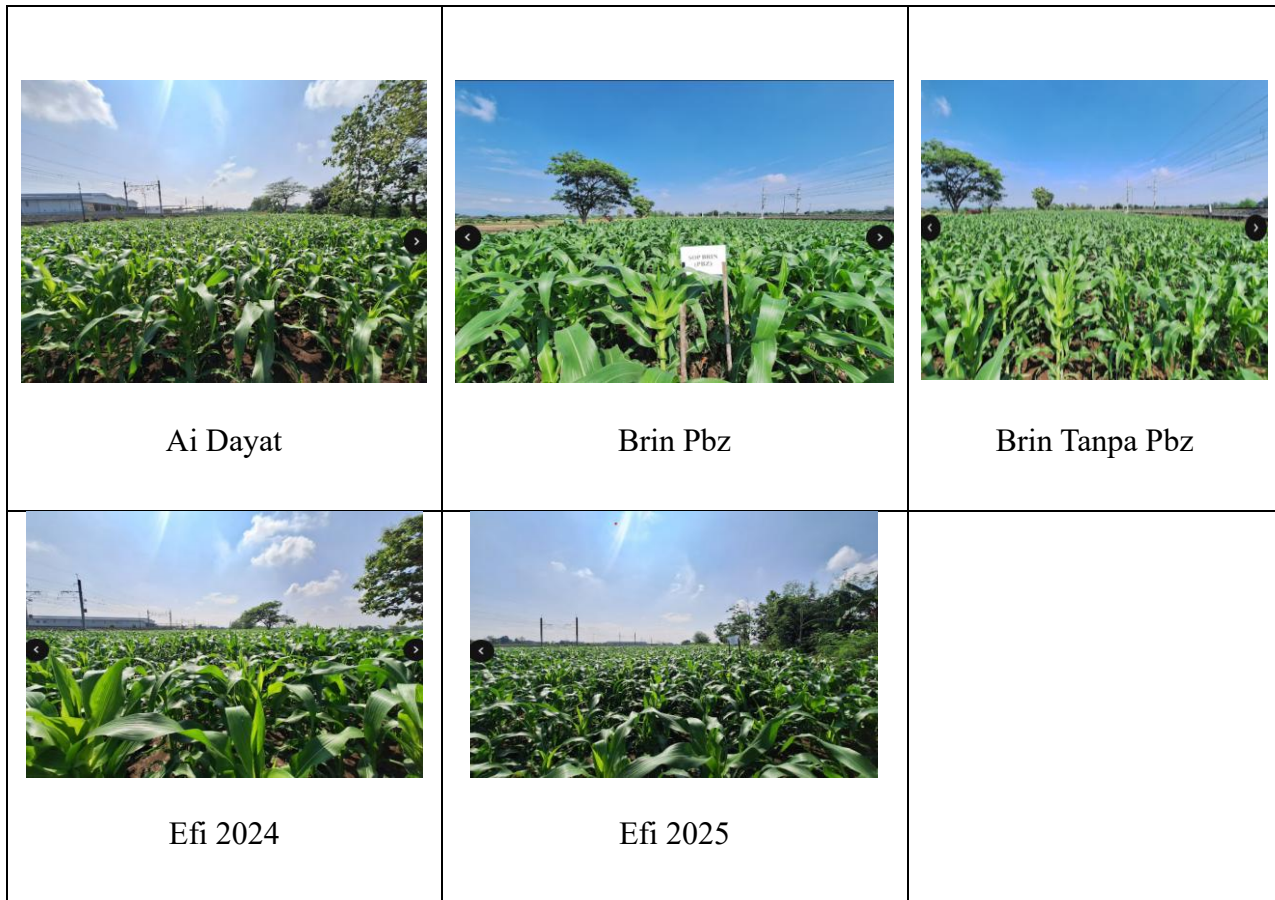
- Putra, A. I. D., & Moh Agus Suriyanto. (2021). Menakar Penerapan Standar Operasional Prosedur Budidaya Untuk Pengendalian Kualitas Hasil Panen Jagung. *Jurnal Manajemen Bisnis*, 18(4), 448–458.  
file:///C:/Users/user/Downloads/3278-Article Text-8845-2-10-20220430 (1).pdf
- Sulistiawan, S., Nugroho, I. M., & Defriani, M. (2025). Implementasi Whatsapp Gateway Pada Sistem Informasi Pengaduan Masyarakat ( Studi Kasus : Kabupaten Purwakarta ). *Jurnal Mahasiswa Teknik Informatika*, 9(5), 8493–8498. <https://ejournal.itn.ac.id/jati/article/view/15090/8197%0A%0A>
- Syah, R. F., & Utomo, A. P. (2025). Functionality Evaluation and Testing of Web-Based Sparepart Sales System Using Black-Box Method. *Jurnal Teknologi Informatika Dan Komputer MH. Thamrin*, 11(2), 492–508.  
<https://journal.thamrin.ac.id/index.php/jtik/article/view/2683/2479%0A%0A>
- Tni, L. M., Wiranto, S., & Mh, S. H. (2022). *digital twin*. 1–8.
- Virgiansyah, M. R. (2025). Sistem Chatbot Helpdesk WhatsApp Berbasis Large Language Model Gemini dengan Fine-Tuning Menggunakan Retrieval-Augmented Generation dan Qdrant Vector Database. *MEANS (Media Informasi Analisa Dan Sistem)*, 10(2), 144–153.  
file:///C:/Users/user/Downloads/5932-Article Text-15705-1-10-20260124 (2).pdf%0A%0A
- Wiryandhani, Firly, Widodo, Suprih, Andrian, & Rian. (2023). Perancangan Purwarupa Dashboard Digital Untuk Pengukuran Produktivitas Kerja Dengan Metode Design Thiniking. *2023*, 17(1), 351–363.  
file:///C:/Users/user/Downloads/Perancangan\_Purwarupa\_Dashboard\_Digital .pdf
- Yesi, J. E., & Noviyanti, P. (2025). Rancang Bangun Sistem Informasi Pertanian Berbasis Web pada Kelompok Tani di Bengkayang. *INSTINK: (Jurnal Inovasi Pendidikan, Teknologi Informasi & Komputer)*, 4(2), 55–68.  
<https://journal.unuha.ac.id/index.php/Instink/article/view/4713/1230>
- Yin, C., Zhang, Q., Mao, X., & Chen, D. (2024). Research of real-time corn yield monitoring system with DNN- based prediction model. *Original Research*, 10.3389(1453823), 1–13. <https://doi.org/10.3389/fpls.2024.1453823>

## LAMPIRAN A

Gambar tanaman Jagung umur 30 HST

 <p data-bbox="448 857 571 896">Ai Dayat</p>	 <p data-bbox="906 857 1029 896">Brin Pbz</p>	 <p data-bbox="1225 857 1428 896">Brin Tanpa Pbz</p>
 <p data-bbox="448 1373 571 1411">Efi 2024</p>	 <p data-bbox="906 1373 1029 1411">Efi 2025</p>	

Gambar tanaman Jagung umur 40 HST



Gambar tanaman Jagung umur 50 HST



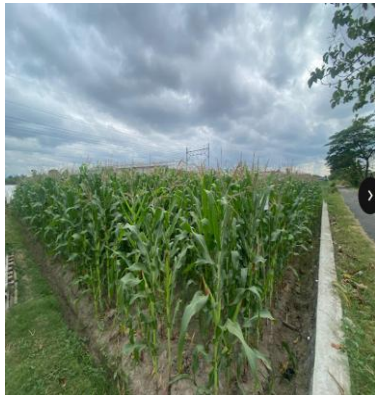


Efi 2024



Efi 2025

Gambar tanaman Jagung umur 60 HST



Ai Dayat



Brin Pbz



Brin Tanpa Pbz



Efi 2024



Efi 2025

Gambar tanaman Jagung umur 70 HST

 <p data-bbox="459 913 576 952">Ai Dayat</p>	 <p data-bbox="884 927 1000 965">Brin Pbz</p>	 <p data-bbox="1230 927 1433 965">Brin Tanpa Pbz</p>
 <p data-bbox="459 1561 576 1599">Efi 2024</p>	 <p data-bbox="884 1554 1000 1592">Efi 2025</p>	





Adapun implementasi tool pada backend ditunjukkan pada potongan kode SIJAGUNG berikut.

```
import {McpServer} from
 '@modelcontextprotocol/sdk/server/mcp.js'
import {StdioServerTransport} from
 '@modelcontextprotocol/sdk/server/stdio.js'
import { z } from 'zod'
import { PeriodicService } from
 '../services/PeriodicService'
import { ProductivityModel } from
 '../models/productivityModel'
import {ProductivityService} from
 '../services/ProductivityService'
import { DestructiveModel } from
 '../models/destructiveModel'
import { DestructiveService } from
 '../services/DestructiveService'

const server = new McpServer({
  name: 'AI-Assistent MCP Server',
  version: '1.0.0'
})
```

```

// tinggi tanaman

server.registerTool(
  'createDataTinggiTanaman',
  {
    title: 'Create Data Tinggi Tanaman (cm)',
    description:
      'create data tinggi tanaman with properties SOP, HST
30, HST 40, HST 50, HST 60 and HST 70',
    inputSchema: {
      sop: z.string(),
      HST30: z.number().optional(),
      HST40: z.number().optional(),
      HST50: z.number().optional(),
      HST60: z.number().optional(),
      HST70: z.number().optional()
    },
    outputSchema: {
      data: z.object({
        periodicId: z.number(),
        periodicData: z.any(),
        periodicName: z.string(),
        periodicCategory: z.string()
      })
    }
  }
)

```

```

    }
  },
  async (data) => {
    const record = await PeriodicService.create({
      periodicName: 'tinggi tanaman',
      periodicCategory: 'TINGGI_TANAMAN',
      periodicData: {
        sop: data.sop,
        HST30: data.HST30,
        HST40: data.HST40,
        HST50: data.HST50,
        HST60: data.HST60,
        HST70: data.HST70
      }
    })
    const output = record.toJSON()
    return [
      {
        content: [{ type: 'text', text: JSON.stringify(output)
}],
        structuredContent: {
          data: {
            periodicId: output.periodicId,
            periodicName: output.periodicName,

```

```

        periodicCategory: output.periodicCategory,
        periodicData: output.periodicData
    }
}
}
}
)

// jumlah daun
server.registerTool(
    'createDataJumlahDaun',
    {
        title: 'Create Data jumlah daun',
        description:
            'create data jumlah daun with properties SOP, HST 30,
HST 40, HST 50, HST 60 and HST 70',
        inputSchema: {
            sop: z.string(),
            HST30: z.number().optional(),
            HST40: z.number().optional(),
            HST50: z.number().optional(),
            HST60: z.number().optional(),
            HST70: z.number().optional()
        }
    }
)

```

```

    },
    outputSchema: {
      data: z.object({
        periodicId: z.number(),
        periodicData: z.any(),
        periodicName: z.string(),
        periodicCategory: z.string()
      })
    }
  },
  async (data) => {
    const record = await PeriodicService.create({
      periodicName: 'jumlah daun',
      periodicCategory: 'JUMLAH_DAUN',
      periodicData: {
        sop: data.sop,
        HST30: data.HST30,
        HST40: data.HST40,
        HST50: data.HST50,
        HST60: data.HST60,
        HST70: data.HST70
      }
    })
  }
}

```

```

const output = record.toJSON()

return {
  content: [{ type: 'text', text: JSON.stringify(output)
}],
  structuredContent: {
    data: {
      periodicId: output.periodicId,
      periodicName: output.periodicName,
      periodicCategory: output.periodicCategory,
      periodicData: output.periodicData
    }
  }
}
)

```

```

// diameter batang
server.registerTool(
  'createDataDiameterBatang',
  {
    title: 'Create Data Diatometer Batang',
    description:

```

```

    'create data diameter batang with properties SOP, HST
30, HST 40, HST 50, HST 60 and HST 70',
  inputSchema: {
    sop: z.string(),
    HST30: z.number().optional(),
    HST40: z.number().optional(),
    HST50: z.number().optional(),
    HST60: z.number().optional(),
    HST70: z.number().optional()
  },
  outputSchema: {
    data: z.object({
      periodicId: z.number(),
      periodicData: z.any(),
      periodicName: z.string(),
      periodicCategory: z.string()
    })
  }
},
async (data) => {
  const record = await PeriodicService.create({
    periodicName: 'diameter batang',
    periodicCategory: 'DIAMETER_BATANG',

```

```

    periodicData: {
      sop: data.sop,
      HST30: data.HST30,
      HST40: data.HST40,
      HST50: data.HST50,
      HST60: data.HST60,
      HST70: data.HST70
    }
  })
  const output = record.toJSON()
  return {
    content: [{ type: 'text', text: JSON.stringify(output)
}],
    structuredContent: {
      data: {
        periodicId: output.periodicId,
        periodicName: output.periodicName,
        periodicCategory: output.periodicCategory,
        periodicData: output.periodicData
      }
    }
  }
}

```

```

)

// SPAD
server.registerTool(
  'createDataSPAD',
  {
    title: 'Create Data SPAD',
    description:
      'create data SPAD with properties SOP, HST 30, HST 40,
HST 50, HST 60 and HST 70',
    inputSchema: {
      sop: z.string(),
      HST30: z.number().optional(),
      HST40: z.number().optional(),
      HST50: z.number().optional(),
      HST60: z.number().optional(),
      HST70: z.number().optional()
    },
    outputSchema: {
      data: z.object({
        periodicId: z.number(),
        periodicData: z.any(),
        periodicName: z.string(),

```

```

        periodicCategory: z.string()
    })
}
},
async (data) => {
    const record = await PeriodicService.create({
        periodicName: 'SPAD',
        periodicCategory: 'SPAD',
        periodicData: {
            sop: data.sop,
            HST30: data.HST30,
            HST40: data.HST40,
            HST50: data.HST50,
            HST60: data.HST60,
            HST70: data.HST70
        }
    })
    const output = record.toJSON()
    return {
        content: [{ type: 'text', text: JSON.stringify(output)
    }],
        structuredContent: {
            data: {

```

```
    periodicId: output.periodicId,  
    periodicName: output.periodicName,  
    periodicCategory: output.periodicCategory,  
    periodicData: output.periodicData  
  }  
}  
}  
}  
)  
)
```