

LAMPIRAN

Lampiran 1. Berita Acara Ujian Skripsi



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN TUGAS AKHIR
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Marie Pangestu
Nim : 2118101
Jurusan : Teknik Informatika S-1
Judul : Eksplorasi Pola Penjualan Produk Basic Menggunakan Algoritma Fp-Growth Pada Paket Bundling (Studi Kasus : Toko Hijab Zoya Kawi, Malang)

Dipertahankan Dihadapan Majelis Penguji Tugas Akhir Jenjang Strata Satu(S-1)
Pada


Hari : Rabu
Tanggal : 21 Januari 2025
Nilai : 86 (A)

Panitia Ujian Tugas Akhir :
Ketua Majelis Penguji



Yosep Agus Prabhoto ST., MT
NIP.P 1031000432

Anggota Penguji :

Dosen Penguji I


Suryo Adi Wibowo, S.T. M.T.
NIP. P. 103100438

Dosen Penguji II


All Mahmud, B.Eng. Ph.D.
NIP. P. 1031000429

Lampiran 2. Formulir Perbaikan Skripsi



PT BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigure-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika ,
maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Marie Pangestu
NIM : 2118101
JURUSAN : Teknik Informatika S-1
JUDUL : EKSPLORASI POLA PENJUALAN PRODUK BASIC
MENGUNAKAN ALGORITMA FP-GROWTH PADA PAKET
BUNDLING (STUDI KASUS : TOKO HIJAB ZOYA KAWI,
MALANG)

No.	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	21 Januari 2025	<ol style="list-style-type: none"> 1. Mengapa algoritma fp-growth. 2. Pengujian terkait rumusan masalah. 3. Berapa hak akses serta kebutuhan fungsional sesuai hak akses. 4. Perancangan database. 5. Pengujian terkait akurasi. 6. Jawaban dari rumusan masalah yang tersaji dalam pengujian. 7. Bagaimana agar produk bisa multiyear? 8. Tutorial program. 	
2.	Penguji II	21 Januari 2025	<ol style="list-style-type: none"> 1. Bab 3.4 dan 4.4 banyak halaman kosong. 2. Pengujian user minimal 10 user. 3. Produk -> mestinya ada gambar(web). 4. Bab 2 tambahkan tentang toko hijab, hijabnya apa saja? 5. About us. Nama + Nim + Gambar. 6. Perancangan database + relasi antar tabel. 	

Anggota Penguji :

Dosen Penguji I

Suryo Adi Wibowo, S.T. M.T.
NIP. P. 1031100438

Dosen Pembimbing I

Ahmad Falsol, S.T. M.T.
NIP. P. 1031000431

Dosen Penguji II

Ali Mahmudi, B.Eng. Ph.D.
NIP. P. 1031000429

Mengetahui :

Dosen Pembimbing II

Deddy R. Anisriar, S.Kom. M.Cs.
NIP. P. 1032000578

Lampiran 3. SK Dosen Pembimbing 1



PT. BNI (PERSERO) MALANG
BANK NAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417630 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III.INF/TA/2024
Lampiran : ---
Perihal : Pembimbing Utama Skripsi

Kepada : **Yth. Bpk/Ibu Ahmad Faisol ST., MT.**
Dosen Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Marie Pangestu
Nim : 2118101
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

21 Agustus 2024 s/d 21 Pebruari 2025

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengstahui
Program Studi Teknik Informatika S-1
Ketua,

Yosep Agus Pranoto, ST., MT.
NIP. P. 1031000432

Form S-4a

Lampiran 4. SK Dosen Pembimbing 2



PT. BNI (PERSERO) MALANG
BANK NAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417836 Fax. (0341) 417834 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III.INF/TA/2024
Lampiran : ---
Perihal : Pembimbing Pendamping Skripsi

Kepada : **Yth. Bpk/Ibu Deddy Rudhistiar S.Kom., M.Cs.**
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Marie Pangestu
Nim : 2118101
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

21 Agustus 2024 s/d 21 Pebruari 2025

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1

Yosep Agus Pranoto, ST., MT.
NIP.P. 1031000432

Form S-4a

Lampiran 5. Formulir Bimbingan Dosen Pembimbing 1

FORMULIR BIMBINGAN SKRIPSI

Nama : Marie Pangestu

Nim : 2118101

Masa Bimbingan : 21 AGUSTUS 2024 s/d 21 FEBRUARI 2025

Judul Skripsi : EKSPLORASI POLA PENJUALAN PRODUK BAKU MENGGUNAKAN ALGORITMA FP-GROWTH PADA PAKET BUNDIRING (STUDI KASUS: TOKO HUJAB Zoya KAWI, MALANG)...

No.	Tanggal	Uraian	Paraf Pembimbing
1.	16/10/2024	Ubah judul menjadi 'Eksplorasi pola'	<i>[Signature]</i>
2.	23/10/2024	Ubah batasan masalah pd Laporan	<i>[Signature]</i>
3.	30/10/2024	Perubahan tabel dan relasi - Tampilan form create dan cara input menu barang	<i>[Signature]</i>
4.	15/11/2024	- Perbaiki UI sistem	<i>[Signature]</i>
5.	18/11/2024	- Ubah form create transaction data	<i>[Signature]</i>
6.	26/11/2024	- Perbaiki jurnal Cabistrak, pengisian metode	<i>[Signature]</i>
7.	3/12/2024	- Perbaiki sistem (super admin, pengot password, database hasil)	<i>[Signature]</i>
8.	5/12/2024	- Periksa sistem (online) - Perbaiki laporan	<i>[Signature]</i>
9.	17/12/2024	- Perbaiki Laporan (flowchart sistem, metode, tinjauan pustaka)	<i>[Signature]</i>
10.	14/01/2024	- Perbaiki Laporan (telaah penelitian, flowchart sistem, metode, tinjauan pustaka)	<i>[Signature]</i>

Malang, 14 Jan 2024

Dosen Pembimbing

[Signature]
Ahmad Faisal S.T. MT
NIP.

Lampiran 5. Formulir Bimbingan Dosen Pembimbing 2

FORMULIR BIMBINGAN SKRIPSI

Nama : Marie Pangestu
 Nim : 2118101
 Masa Bimbingan : 21 Agustus 2024 s/d 21 Februari 2025
 Judul Skripsi : EKSPLORASI POLA PENJUALAN PRODUK BASIC MENGGUNAKAN ALGORITMA FP-GROWTH PADA PAKET BUNDLING (STUDI KASUS : TOKO HUAB ZOYA KAWI, MALANG)

No.	Tanggal	Uraian	Paraf Pembimbing
1.	17 Okt 2024	Pemberanan data.	
2.	24 Okt 2024	- Batasan masalah 4 min. sup. - Import data ke sistem.	
3.	31 Okt 2024	- Perhitungan manual algoritma, - Evaluasi metode	
4.	9 NOV 2024	- Evaluasi revisi semproy - penetapan batasan masalah	
5.	14 NOV 2024	- Perhitungan manual untuk sistem	
6.	15 NOV 2024	- Perhitungan manual untuk sistem - Evaluasi sistem dan UI (metode)	
7.	23 NOV 2024	- Revisi jurnal	
8.	6 Des Jan 2024	- perbaruan flowchart metode dan sistem	
9.	9 Jan 2025	- perbaikan tambahan class diagram Laporan	
10.	15 Jan 2025	- Demo project dan Laporan	

Malang, 15 Jan. 2024

Dosen Pembimbing

Deddy Kethica Silam 1981
 NIP.

Lampiran 6. Hasil Persentase Cek Plagiasi Turnitin



INSTITUT TEKNOLOGI NASIONAL MALANG
PERPUSTAKAAN PUSAT
Jln. Berendang Sgura gura Ilo 2 Malang 65145
Telp. (0341) 551431 Pns. 163-146-147 Fax. (0341) 553015 Website: library@itn.ac.id

FORM UJI PLAGIASI UNTUK MAHASISWA

Yang bertandatangan di bawah ini, Mahasiswa Institut Teknologi Nasional Malang:

Nama : MARIE PANGESTU
NIM : 2118101
Fakultas / Jurusan : FAKULTAS TEKNOLOGI INDUSTRI / TEKNIK INFORMATIKA S-1
Email : pangestumarie2@gmail.com
No. Tlp : 082338152545
Judul/ Jml artikel : EKSPLORASI POLA PENJUALAN PRODUK BASIC MENGGUNAKAN ALGORITMA FP-GROWTH PADA PAKET BUNDLING (STUDI KASUS: TOKO HIJAB ZOYA KAWI MALANG)

Karya ilmiah yang bersangkutan di atas melalui proses cek plagiasi menggunakan aplikasi turnitin dengan hasil kemiripan (Similarity) sebesar..... 6..... %
Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana mestinya.

Mahasiswa

MARIE PANGESTU

Malang, 02-02-2022

Pelaksana

Rahmat Wicaksono

EKSPLORASI POLA PENJUALAN PRODUK BASIC
MENGUNAKAN ALGORITMA FP-GROWTH PADA PAKET
BUNDLING (STUDI KASUS : TOKO HIJAB ZOYA KAWI, MALANG)

ORIGINALITY REPORT

6%	6%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ejurnal.teknokrat.ac.id Internet Source	2%
2	ejournal.itn.ac.id Internet Source	2%
3	eprints.itn.ac.id Internet Source	2%

Exclude quotes Off
Exclude bibliography On

Exclude matches < 2%

Lampiran 7. Dokumentasi Pengujian Blackbox



Lampiran 8. Source Code Metode FP-Growth

```
const { responseOk, responseNotFound, responseInternalServerError } =
require("../util/responseHelper");
const handleError = require("../util/handleError")
const { penjualanTgl, getTransactionsByDateRange,
findPenjualanByTgl } = require("../models/FpGrowthModel")
const { findManyProdukWarna } = require("../models/ProdukWarnaModel");
const { error } = require("winston");

const penjualanYearsMonth = handleError(async (req, res, next) =>
{
  try {
    const dates = await penjualanTgl();
    /*
      Contoh data yang dihasilkan dari variable dates:
      [
        { tgl: '2023-07-31'}
        { tgl: '2023-07-30'}
        { tgl: '2023-07-29'}
      ]
    */

    // Gunakan Set untuk memastikan kombinasi year dan month
    unik
    const uniqueYearMonth = Array.from(
      new Set( // menyimpan struktur javascript yang hanya
      menyimpan nilai unik
        dates.map((date) => {
          const tgl = new Date(date.tgl); // Konversi
          string tgl ke objek Date
          /*
            Contoh data yang dihasilkan dari variable
            tgl:
            [
              "2023-07-31T00:00:00.000Z"
              "2023-07-30T00:00:00.000Z"
              "2023-07-29T00:00:00.000Z"
            ]
          */
          return `${tgl.getFullYear()}-${tgl.getMonth()
+ 1}`; // Gabungkan year dan month sebagai string unik
          /*
            Contoh data yang dihasilkan:
            [
              "2023-7"
              "2023-7"
              "2023-7"
            ]
          */
        })
      )
    /*
      Contoh data yang dihasilkan:
      [
        "2023-7"
      ]
    */
  }
}
```

```

        */
        ).map((ym) => {
            const [year, month] = ym.split("-");
            /*
                Contoh data yang dihasilkan :
                [
                    {
                        year: "2023",
                        month: "7"
                    }
                ]
            */
            return { year: parseInt(year, 10), month:
parseInt(month, 10) };
            /*
                Contoh data yang dihasilkan :
                [
                    {
                        year: 2023,
                        month: 7
                    }
                ]
            */
        });

        responseOk(res, uniqueYearMonth);
    } catch (error) {
        next(error);
    }
});

const penjualanTransactions = handleError(async (req, res, next)
=> {
    const { startYear, startMonth, endYear, endMonth } = req.query;
    //supaya key parameter terlihat di endpoint parameter /:id

    try {
        // Pastikan parameter tahun dan bulan valid
        if (!startYear || !startMonth || !endYear || !endMonth) {
            return res.status(400).json({
                status: false,
                message:
                    "Missing required query parameters (startYear,
startMonth, endYear, endMonth)",
            });
        }

        // Panggil fungsi untuk mendapatkan transaksi berdasarkan
rentang tahun dan bulan
        const transactions = await getTransactionsByDateRange(
            startYear,
            startMonth,
            endYear,
            endMonth
        );

        // Kirimkan response
        responseOk(res, transactions);
    } catch (error) {

```

```

    next(error);
  }
});

const conditionalFrequentPatternTree = handleError(async (req,
res, next) => {
    const conditionalPatternBases =
req.body.conditionalPatternBases;
    const result = {};

    for (const item in conditionalPatternBases) {
      const paths = conditionalPatternBases[item];
      const pathOccurrences = {};

      paths.forEach(pathData => {
        const path = pathData.path;
        const frequency = pathData.frequency;

        // Handle path frequencies
        const pathString = path.join(' -> ');
        const key = `${item}, Path: ${pathString}`;
        if (!result[key]) {
          result[key] = frequency;
        } else {
          result[key] += frequency;
        }

        // Handle path element frequencies (first element in
path)
        const pathElement = path[0];
        if (!pathOccurrences[pathElement]) {
          pathOccurrences[pathElement] = 0;
        }
        pathOccurrences[pathElement] += frequency;
      });

      // Add occurrences of single-element paths
      for (const pathElement in pathOccurrences) {
        const frequency = pathOccurrences[pathElement];
        const key = `${item}, Path: ${pathElement}`;

        if (frequency === 1) {
          result[key] = result[key] || 1;
        } else {
          result[key] = frequency;
        }
      }
    }

    // Format the result into the desired structure
    const formattedResult = [];
    for (const key in result) {
      const [item, pathString] = key.split(', Path: ');
      formattedResult.push({
        item,
        path: pathString,
        frequency: result[key],
      });
    }
  }
});

```

```

        responseOk(res, formattedResult)
    })

    const conditionalPatternBases = handleError(async (req, res,
next) => {
        const { frequentItemsets, fpTree } = req.body;

        // Fungsi rekursif untuk menemukan conditional patterns
        const findConditionalPatterns = (node, targetItem, path,
conditionalPatterns) => {
            if (node.item === targetItem && path.length > 0) {
                conditionalPatterns.push({
                    path: [...path],
                    frequency: node.frequency,
                });
                return;
            }

            if (node.item !== null) {
                path.push(node.item);
            }

            // Proses rekursif untuk setiap child node
            Object.entries(node.children).forEach(([childItem,
childNode]) => {
                findConditionalPatterns(childNode, targetItem,
[...path], conditionalPatterns);
            });
        };

        const conditionalPatternBases = {};

        // Proses setiap item dari frequentItemsets
        Object.entries(frequentItemsets).forEach(([item, { count }])
=> {
            const conditionalPatterns = [];
            findConditionalPatterns(fpTree, item, [],
conditionalPatterns);

            // Filter pola yang frekuensinya lebih besar dari 0
            const filteredPatterns = conditionalPatterns.filter(
                (pattern) => pattern.frequency > 0
            );

            conditionalPatternBases[item] = filteredPatterns;
        });

        // Kembalikan hasil sebagai respons
        return responseOk(res, conditionalPatternBases);
    });

    const generateFrequentPatternRules = handleError(async (req, res,
next) => {
        const {
            conditionalFrequentPatternTree,
            frequentItemsets,
            start_period,
            end_period,

```

```

    minSupport,
    minConfidence,
    totalTransactions,
  } = req.body;

  // Fungsi untuk menghitung produk set
  const hitungProdukSet = async (start_date, end_date) => {
    try {
      const [startYear, startMonth] = start_date.split('-')
        .map(Number);
      const [endYear, endMonth] = end_date.split('-')
        .map(Number);

      const monthsInRange = [];
      let currentYear = startYear;
      let currentMonth = startMonth;

      while (currentYear < endYear || (currentYear ===
        endYear && currentMonth <= endMonth)) {
        monthsInRange.push({ year: currentYear, month:
          currentMonth });
        currentMonth++;
        if (currentMonth > 12) {
          currentMonth = 1;
          currentYear++;
        }
      }

      const penjualanByMonth = [];
      for (const { year, month } of monthsInRange) {
        const startOfMonth = new Date(year, month - 1,
          1);
        const endOfMonth = new Date(year, month, 0);
        const penjualan = await
          findPenjualanByTgl(startOfMonth, endOfMonth);
        penjualanByMonth.push(...penjualan);
      }

      if (!penjualanByMonth || penjualanByMonth.length ===
        0) {
        throw new Error("Data penjualan tidak ditemukan");
      }

      const countSet = {};
      const penjualanGrouped = penjualanByMonth.reduce((acc,
        item) => {
        if (!acc[item.id_transaksi]) {
          acc[item.id_transaksi] = [];
        }
        acc[item.id_transaksi].push(item.id_produkwarna);
        return acc;
      }, {});

      const generateSubsets = (array) => {
        const results = [];
        for (let i = 1; i < (1 << array.length); i++) {
          const subset = [];
          for (let j = 0; j < array.length; j++) {
            if (i & (1 << j)) {

```

```

        subset.push(array[j]);
    }
    }
    results.push(subset);
}
return results;
};

Object.keys(penjualanGrouped).forEach(idTransaksi =>
{
    const produkIds = penjualanGrouped[idTransaksi];
    const subsets = generateSubsets(produkIds);

    subsets.forEach(subset => {
        if (subset.length >= 8) return;

        const produkSet = JSON.stringify(subset);
        countSet[produkSet] = (countSet[produkSet] ||
0) + 1;
    });
});

return Object.keys(countSet).map(produkSet => ({
    produk_set: JSON.parse(produkSet),
    count: countSet[produkSet],
}));
} catch (error) {
    throw new Error(error.message);
}
};

// Fungsi untuk mendapatkan data produk warna
const getProdukWarnaData = async () => {
    try {
        const findProdukWarna = await findManyProdukWarna();
        const produkWarnas = findProdukWarna.map((item) => ({
            id_produkwarna: item.id_produkwarna,
            id_produk: item.id_produk,
            nama_produk: item.produk?.nama_produk || null,
            id_warna: item.id_warna,
            nama_warna: item.warna?.nama_warna || null,
            harga: item.harga,
        }));

        const result = [];
        produkWarnas.forEach((produkWarna) => {
            const id = produkWarna.id_produkwarna;
            const produkId = `${produkWarna.id_produk}-
${produkWarna.id_warna}`;
            const produkNama = `${produkWarna.nama_produk}-
${produkWarna.nama_warna}`;

            if (!result[produkId]) {
                result[produkId] = {
                    id,
                    produk_id: [],
                    produk_nama: [],
                };
            }
        });
    }
}

```



```

        result[produkId].produk_id.push(produkId);
        result[produkId].produk_nama.push(produkNama);
    });

    return Object.values(result);
} catch (error) {
    console.error('Error fetching product color data:',
error);
    return { status: false, message: 'Failed to fetch
product color data' };
}
};

// Main logic
try {
    const hitungProduk = await hitungProdukSet(start_period,
end_period);
    const produkWarnaData = await getProdukWarnaData();

    const FpRules = [];
    if (Array.isArray(conditionalFrequentPatternTree)) {
        conditionalFrequentPatternTree.forEach((entry) => {
            const item = entry.item;
            const path = entry.path;
            const frequency = entry.frequency;

            if (frequentItemsets[item]) {
                const supportCount =
frequentItemsets[item].count;
                const confidence = frequency / supportCount;

                FpRules.push({
                    item,
                    path,
                    frequency,
                    confidence,
                    start_period,
                    end_period,
                    min_support: minSupport,
                    min_confidence: minConfidence,
                });
            }
        });
    } else {
        return res.status(400).json({ error: 'Invalid
conditionalFrequentPatternTree format' });
    }

    const updatedFpRules = FpRules.map((rule) => {
        const { item, path, frequency, confidence } = rule;
        const matchedPaths = [];
        let matchedId = null;

        produkWarnaData.forEach((data) => {
            if (data.produk_nama.includes(item)) {
                matchedId = data.id;
            }
        });
    });
};

```