**LAPORAN AKHIR**

**PENERAPAN ALGORITMA DECISION TREE DALAM KLASIFIKASI TRANSAKSI PPN DAN PPH PADA SISTEM INFORMASI PAJAK DIGITAL: STUDI KASUS PADA DINAS PERDAGANGAN DAN PERINDUSTRIAN KABUPATEN BANDUNG**

**A logo of a school

AI-generated content may be incorrect.**

**Diajukan Oleh**

**MUHAMMAD RENDY KRISNA**

**2305102004**

**PROGRAM STUDI MANAJEMEN INFORMATIKA**

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA POLITEKNIK NEGERI MEDAN**

**MEDAN**

**2025**

# LEMBAR PERSEMBAHAN

Dengan segala kerendahan hati, penulis mengucapkan syukur ke hadirat Allah SWT. atas segala kekuatan dan karunia-Nya yang telah diberikan kepada penulis selama proses penyusunan laporan akhir ini. setiap doa dan harapan senantiasa penulis panjatkan, agar tugas akhir ini dapat diselesaikan dengan baik dan Penuh semangat. Dalam setiap proses yang dijalani, penulis selalu berusaha melibatkan Allah SWT. agar segala usaha yang dilakukan tidak menjadi beban atau penyesalan, melainkan menjadi wujud syukur atas segala nikmat yang telah diberikan-Nya.

Dengan rasa syukur dan bangga, penulis mempersembahkan laporan akhir ini kepada orang tua tercinta. Rasa Terima kasih yang begitu dalam selalu penulis rasakan atas kehadiran dan peran orang tua dalam setiap langkah kehidupan penulis.

Penulis mempersembahkan laporan akhir ini kepada mamak yang sangat penulis cintai dan sayangi. Tidak ada kata yang mampu menggambarkan seberapa besar perjuangan dan pengorbanannya dalam hidup penulis. Terima kasih yang terhingga atas kasih sayang, cinta, dan perhatian yang tulus, yang selalu mamak berikan tanpa pamrih. Setiap nasihat, doa, dan semangat yang mamak berikan menjadi kekuatan besar dalam menjalani setiap proses kehidupan. Penulis selalu bersyukur dan berterima kasih, karena Allah SWT. telah menghadirkan sosok mamak yang luar biasa dalam hidup penulis.

Laporan akhir ini juga penulis persembahkan kepada teman-teman, yaitu Jefry, Hafizh, Bima, Mifdhal yang telah mendampingi, mendukung dan membantu penulis selama ini. Meskipun masa perkulihan terasa singkat, namun kebersamaan dan perjuangan yang dilalui Bersama adalah kenangan yang tak ternilai.

Penulis sangat bersyukur dan berterima kasih atas setiap momen yang telah dilalui bersama. Terakhir, laporan ini penulis persembahkan untuk diri sendiri, yang telah bersabar, berjuang, bersemangat, dan bekerja keras hingga titik ini. Semoga ini menjadi awal yang baik dari setiap perjuangan dan langkah baru dalam perjalanan hidup penulis.

# ABSTRAK

Penelitian ini bertujuan untuk menganalisis penerapan algoritma Decision Tree dalam mengklasifikasikan transaksi Pajak Pertambahan Nilai (PPN) dan Pajak Penghasilan (PPh) pada sistem informasi pajak digital di Dinas Perdagangan dan Perindustrian Kabupaten Bandung. Latar belakang penelitian ini adalah masih rendahnya akurasi sistem pajak digital dalam membedakan jenis pajak secara otomatis, yang menyebabkan kesalahan klasifikasi dan keterlambatan pelaporan. Penelitian menggunakan pendekatan kuantitatif dengan metode eksperimen, di mana algoritma Decision Tree (CART) diterapkan pada dataset transaksi pajak digital yang mencakup variabel nilai transaksi, jenis barang atau jasa, status PKP, dan metode pembayaran. Hasil pengujian menunjukkan bahwa algoritma Decision Tree memiliki tingkat akurasi tinggi dalam mengklasifikasikan transaksi pajak, dengan variabel jenis transaksi, nilai nominal, dan status PKP sebagai faktor paling berpengaruh. Model ini terbukti meningkatkan efisiensi serta akurasi sistem pajak digital dibandingkan metode manual sebelumnya dan mampu mengurangi kesalahan klasifikasi yang berpotensi menimbulkan kebocoran penerimaan pajak daerah. Secara keseluruhan, penerapan algoritma Decision Tree memberikan pengaruh positif terhadap peningkatan kinerja sistem informasi pajak digital dan menjadi rekomendasi pengembangan sistem klasifikasi pajak yang lebih adaptif, efisien, dan transparan.

**Kata Kunci** : Decision Tree, Pajak Pertambahan Nilai (PPN), Pajak Penghasilan (PPh), Sistem Informasi Pajak Digital.

# ABSTRACT

*This research aims to analyze the implementation of the Decision Tree algorithm in classifying Value Added Tax (VAT) and Income Tax (PPh) transactions within the digital tax information system of the Department of Trade and Industry, Bandung Regency. The study was conducted based on the low accuracy of existing digital tax systems in automatically distinguishing tax types, which often leads to classification errors and delays in tax reporting. The research employed a quantitative approach using an experimental method, in which the Decision Tree (CART) algorithm was applied to a digital tax transaction dataset consisting of variables such as transaction value, type of goods or services, taxpayer status (PKP), and payment method. The results indicate that the Decision Tree algorithm achieved a high level of accuracy in classifying tax transactions, with transaction type, nominal value, and taxpayer status identified as the most influential variables. The model significantly improved the efficiency and accuracy of the digital tax system compared to manual methods and reduced misclassification errors that could lead to local tax revenue losses. Overall, the application of the Decision Tree algorithm positively impacts the performance of the digital tax information system and serves as a recommendation for developing adaptive, efficient, and transparent tax classification systems within local government institutions.*

***Keywords:*** *Decision Tree, Value Added Tax (VAT), Income Tax (PPh), Digital Tax Information System.*

# KATA PENGANTAR

Dengan penuh rasa syukur, penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, karena berkat rahmat dan anugerah-Nya, laporan akhir ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada Program Studi Manajemen Informatika, Jurusan Teknik Komputer dan Informatika. Adapun judul dari laporan ini adalah “Analisis Penerapan Algoritma Decision Tree Dalam Klasifikasi Transakdi PPN Dan PPH Pada Sistem Informasi Pajak Digital: Studi Kasus Pada Dinas Perdagangan Dan Perindustrian Kabupaten Bandung”. Dalam proses pembuatan laporan akhir ini penulis banyak sekali mendapatkan bantuan dan bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Tuhan Yang Maha Esa yang senantiasa memberikan kesehatan dan kekuatan, sehingga penulis dapat menyelesaikan laporan magang ini.
2. Bapak Dr.Ir. Idham Kamil, S.T., M.Kom. selaku Direktur Politeknik Negeri Medan.
3. Bapak Kadri Yusur, S.T., M.Kom. selaku Ketua Jurusan Teknik Komputer dan Informatika.
4. Ibu Yulia Agustina Dalimunthe, S.Si., M.Kom. selaku Koordinator Program Studi Manajemen Informatika.
5. Bapak Ismael, S.Kom., M.Kom. selaku dosen pembimbing yang senantiasa memberikan arahan dan bimbingan selama penyusunan laporan magang ini.
6. Seluruh Staff Pengajar dan Administrasi Program Studi Manajemen Informatika, Jurusan Teknik Komputer dan Informatika.
7. Pihak staff/petugas bagian Perlengkapan Politeknik Negeri Medan yang telah memberikan izin dan membantu penulis dalam proses penelitian.
8. Kepada Orang Tua dan Keluarga saya yang sudah mau memberikan dorongan dan semangat dalam proses pengerjaan laporan akhir ini.
9. Temen-temen Penulis yang sudah menemani selama 3 tahun masa perkuliahan dan yang telah membantu selama masa sulit saat perkuliahan.

Penulis berharap laporan ini dapat memberikan gambaran yang jelas mengenai proses dan hasil dari tugas akhir yang telah dilaksanakan, serta dapat menjadi bahan referensi maupun pembelajaran bagi pihak lain. Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh karena itu penulis sangat mengharapkan kritik dan saran yang membangun untuk perbaikan di masa yang akan datang.

Medan, 02 November 2025

Muhammad Rendy Krisna

# DAFTAR ISI

[LEMBAR PERSEMBAHAN i](#_heading=h.7jquv0vhwf5c)

[ABSTRAK ii](#_heading=h.wn3kb8ue8gpc)

[ABSTRACT iii](#_heading=h.xjrhcv7sc8cy)

[KATA PENGANTAR iv](#_heading=h.qwk6qalxknfb)

[DAFTAR ISI vi](#_heading=h.w9dvzoc889bh)

[DAFTAR TABEL ix](#_heading=h.cyax6d4cmpk6)

[DAFTAR GAMBAR x](#_heading=h.ia6ovbpa2vzo)

[BAB I 1](#_heading=h.3bg4x37d1kcc)

[1.1 Latar Belakang 1](#_heading=h.1n7oggxh6ukt)

[1.2 Rumusan Masalah 2](#_heading=h.vib4p34jhj6k)

[1.3 Batasan Masalah 3](#_heading=h.5mqqjo3rhit4)

[1.4 Tujuan Laporan Akhir 4](#_heading=h.qkcsp3mbq79f)

[1.5 Manfaat Laporan Akhir Tugas akhir 5](#_heading=h.eghzh6djd38w)

[1.6 Sistematika Laporan 6](#_heading=h.jrqlcf7k952p)

[BAB II 8](#_heading=h.ifzzlp4walvq)

[LANDASAN TEORI 8](#_heading=h.iiqb64365de)

[2.1 Penelitian Terdahulu 8](#_heading=h.4luwfes452tq)

[2.2 Landasan Teori 11](#_heading=h.oau9qdkbdzif)

[2.2.1 Algoritma Decision Tree 11](#_heading=h.6tvme3hyxjv0)

[2.2.2 Konsep Dasar *Decision Tree* 12](#_heading=h.nnhlrn3eku6l)

[2.2.3 Tahapan Algoritma *Decision Tree* 14](#_heading=h.12omjg3lpv7w)

[2.2.4 Langkah-Langkah Algoritma *Decision Tree* CART 15](#_heading=h.s5aepfrndbqd)

[2.2.5 Klasifikasi Transaksi PPN dan PPh 19](#_heading=h.9aq90zy0772i)

[2.2.6 Pajak Digital 20](#_heading=h.dd8d52nnp083)

[2.2.7 Informasi Pajak Digital 21](#_heading=h.uxy4lg7melks)

[2.2.8 Peran *Decision Tree* dalam Klasifikasi Transaksi Pajak 21](#_heading=h.qvgk6nkkmn7d)

[2.2.9 Metode SDLC (Software Development Life Cycle) 22](#_heading=h.9i5mnkyt7j50)

[2.2.10 Native PHP 25](#_heading=h.3y1cai4kleyf)

[2.2.11 HTML 26](#_heading=h.5jaau7f2g0kb)

[2.2.12 Tailwind Css 26](#_heading=h.k6cyinuheq0a)

[2.2.13 Database 26](#_heading=h.d5wmf6qze2jl)

[2.2.14 Mysql 27](#_heading=h.lt3ozli8fr26)

[2.2.15 PHPMyAdmin 27](#_heading=h.v1k6xywvn0h3)

[2.2.16 Laragon 27](#_heading=h.x5ir134vivrv)

[2.2.17 Visual Studio Code 28](#_heading=h.nwen6z7m3ypy)

[2.2.18 Blackbox 28](#_heading=h.bllfhr5ifcqf)

[2.2.19 Python 28](#_heading=h.o204nt752l2a)

[2.2.20 Unified Modeling Language (UML) 29](#_heading=h.ksmdv5ap07fb)

[BAB 3 34](#_heading=h.nrhtmt294a5s)

[METODE PENELITIAN 34](#_heading=h.ujp0tpjqvm1e)

[3.1 Alat dan Bahan Pembuatan Rancangan 34](#_heading=h.of15bbxwcc48)

[3.1.1 Alat 34](#_heading=h.x3ypj99mtv7m)

[3.2 Metode Pengumpulan 34](#_heading=h.dmlllcc9oazt)

[3.2.1 Wawancara 35](#_heading=h.f7e326812wud)

[3.3 Metode Pengembangan 35](#_heading=h.rjalg29sks0l)

[3.4 Analisis Kebutuhan Sistem 36](#_heading=h.aer46aiktawf)

[3.4.1. Analisis Sistem Berjalan 36](#_heading=h.2qz6bqjpfzpc)

[3.5 Langkah Perancangan 37](#_heading=h.sdpgw9k3ni1u)

[3.5.1 Perancangan Use Case Diagram 38](#_heading=h.4613i6fdjsl2)

[3.5.2 Perancangan *Activity Diagram* 43](#_heading=h.woovksbfqp5w)

[3.5.1 Perancangan Class Diagram 57](#_heading=h.ycpe8djdlff5)

[3.7.1 Perancangan Database 67](#_heading=h.qu0jx5emceh7)

[BAB 4 71](#_heading=h.14bkf2cjvkg1)

[HASIL DAN PEMBAHASAN 71](#_heading=h.wz06ls3zvt8v)

[4.1 Hasil 71](#_heading=h.uj0nj024gt7r)

[4.1.1 Halaman 71](#_heading=h.7o4nc5dk9l1j)

[4.2 Pembahasan 81](#_heading=h.nyv9hdu5bwe6)

[4.2.1 Kelebihan Sistem Informasi Klasifikasi dan Manajemen Pajak (SIKAPA) 82](#_heading=h.mimsip1ls82a)

[4.2.2 Kekurangan Sistem Informasi Klasifikasi dan Manajemen Pajak (SIKAPA) 84](#_heading=h.z6x8gh65pjxc)

[BAB 5 84](#_heading=h.b4yx2ox2vpt1)

[KESIMPULAN DAN SARAN 84](#_heading=h.yan8tk22lauj)

[5.1 Simpulan 84](#_heading=h.32ss3t3hiuea)

[5.2 Saran 85](#_heading=h.wowh8uskdcw7)

[DAFTAR PUSTAKA 87](#_heading=h.m7t7popfjwo)

# **DAFTAR TABEL**

[Tabel 2. 1 Penelitian Terdahulu 8](#_heading=h.niy78xy27p2x)

[Tabel 2. 2 Tabel Ringkasan Kegiatan 18](#_heading=h.dcsvwm8a4lul)

[Tabel 2. 3 Simbol-simbol Use Case Diagram 29](#_heading=h.37ec6m7ugm95)

[Tabel 2. 6 Simbol-simbol Class Diagram 32](#_heading=h.mm64azqs0d3n)

[Tabel 3. 1Deskripsi Actor Sistem Pakar Rekomendasi Jurusan Vokasi 39](#_heading=h.rpx5mx99jlpl)

[Tabel 3. 2 Deskripsi usecase Sistem Klasifikasi Pajak 40](#_heading=h.c7zsfczh1709)

[Tabel 3. 3 Diagram Login Admin 44](#_heading=h.rxncvg6bfiot)

[Tabel 3. 4 Activity Diagram Login pegawai 45](#_heading=h.uywo81nzic9i)

[Tabel 3. 5 Activity Diagram Input Transaksi Manual Pegawai 47](#_heading=h.c1us7z93u9ih)

[Tabel 3. 6 Activity Diagram Input Transaksi Manual Admin 49](#_heading=h.hrr4rpp51x50)

[Tabel 3. 7 Activity Diagram Input Transaksi Excel(otomatis Ai) 51](#_heading=h.7ouaou5mszfy)

[Tabel 3. 8 Activity Diagram Validasi Transaksi 51](#_heading=h.6e46yldm1je7)

[Tabel 3. 9 Activity Diagram Cetak Laporan 53](#_heading=h.z7mkkg8ec9lg)

[Tabel 3. 10 Activity Diagram Kelola Rekaan 54](#_heading=h.kxlquf90y9z8)

[Tabel 3. 11 Activity Diagram Kelola Manajemen Pengguna 55](#_heading=h.imbsub5ycfsl)

[Tabel 3. 12 Activity Diagram Kelola Pengaturan system 56](#_heading=h.ablb7ntydy6v)

[Tabel 3. 13 Perancangan Table Users 67](#_heading=h.6cxdatkzpje4)

[Tabel 3. 14 Perancangan Tabel Rekanan 67](#_heading=h.7v7eaws6eug7)

[Tabel 3. 15 Rancangan table kategori 68](#_heading=h.ysvog0pntzdn)

[Tabel 3. 16 Rancangan table Transaksi 68](#_heading=h.miffols4975i)

[Tabel 3. 17 Rancangan Table Pengaturan 69](#_heading=h.hd48klmpmogf)

[Tabel 3. 18 Rancangan Table log\_aktivitas 69](#_heading=h.ewqffm6hhtso)

[Tabel 3. 19 Rancangan table dokumen\_pendukung 70](#_heading=h.kipzbovrvsfa)

[Tabel 4. 1 Hasil Pengujian Sistem dengan Metode Black Box Testing 80](#_heading=h.xfcac5jm70gq)

# **DAFTAR GAMBAR**

[Gambar 2. 1 Flowchart Tahapan Algoritma CART 17](#_heading=h.6k0dixxoxrl6)

[Gambar 2. 2 Metode Waterfall 23](#_heading=h.y3n0u6huzwid)

[Gambar 2. 3 Metode Prototype 24](#_heading=h.rxdpaubd5gdm)

[Gambar 3. 1 Use Case Diagram Sistem Klasifikasi Pajak 39](#_heading=h.mabj2eh1kxym)

[Gambar 3. 2 Activity Diagram Login Admin 43](#_heading=h.c151d5nn2wrr)

[Gambar 3. 3 Activity Diagram Login Pegawai 45](#_heading=h.vvq0qju5loei)

[Gambar 3. 4 Activity Diagram Input Transaksi Manual Pegawai 46](#_heading=h.jrlkb3v72a2s)

[Gambar 3. 5 Activity Diagram Input Transaksi Manual Admin 48](#_heading=h.5ayez5hzni97)

[Gambar 3.7 Gambar 3. 6 Activity Diagram Input Transaksi Excel(otomatis Ai) 50](#_heading=h.cmnmtkcwutt)

[Gambar 3. 7 Activity Diagram Validasi Transaksi 51](#_heading=h.4thekpeggns3)

[Gambar 3. 8 Activity Diagram Cetak Laporan 53](#_heading=h.fm0dg2mbfjms)

[Gambar 3. 9 Activity Diagram Kelola Rekaan 54](#_heading=h.rpbhrnm0l4st)

[Gambar 3. 10 Activity Diagram Kelola Manajemen Pengguna 55](#_heading=h.vfbtz5kqer63)

[Gambar 3. 11 Activity Diagram Kelola Pengaturan system 56](#_heading=h.z07flr212jtd)

[Gambar 3. 12 Class Diagram Sistem Klasifikasi Pajak 58](#_heading=h.fbyml6lmd7fv)

[Gambar 3. 13 Halaman Login Admin 60](#_heading=h.3m4rvxix3f5f)

[Gambar 3. 14 Perancangan Halaman Dashboard 61](#_heading=h.ene7h5d3g7x3)

[Gambar 3. 15 Perancangan Halaman Input Transaksi 61](#_heading=h.yzpejjsefe6d)

[Gambar 3. 16 Perancangan Halaman Transaksi 62](#_heading=h.beqservhqpu8)

[Gambar 3. 17 Halaman input transaksi excel 63](#_heading=h.4ev2mt1lcvtk)

[Gambar 3. 18 Halaman Cetak Laporan 64](#_heading=h.assuep64l5sn)

[Gambar 3. 19 Halaman Input Rekanan 65](#_heading=h.pwo874mkt1ty)

[Gambar 3. 20 Halaman Input Kategori 66](#_heading=h.9cbgk29cwk7p)

[Gambar 3. 21 Halaman Logout 66](#_heading=h.nmzobnra9gpm)

[Gambar 4. 1 Halaman Login Admin 71](#_heading=h.9sf5s0rg6f83)

[Gambar 4. 2 Halaman Dashboard Admin 72](#_heading=h.o6snu3gjnann)

[Gambar 4. 3 Halaman Input Transaksi 72](#_heading=h.gnas6ykugveb)

[Gambar 4. 4 Halaman Daftar Transaksi 73](#_heading=h.akjpfypfo5xi)

[Gambar 4. 5 Halaman Monitoring 74](#_heading=h.e0wndsurg0sc)

[Gambar 4. 6 Halaman Import Excel(AI) 74](#_heading=h.no13yakgexch)

[Gambar 4. 7 Halaman Cetak Laporan 75](#_heading=h.lau6u5fspa8s)

[Gambar 4. 8 Halaman Priview Cetak Laporan 75](#_heading=h.ydam2cmqvojm)

[Gambar 4. 9 Halaman Master Data Rekanan 76](#_heading=h.vp6v40ucqin6)

[Gambar 4. 10 Halaman Manajemen Pengguna 76](#_heading=h.s5n1rg62k44f)

[Gambar 4. 11 Halaman Mengelola Pengaturan 77](#_heading=h.4pf4zggjk3ny)

[Gambar 4. 12 Halaman Login Pegawai 77](#_heading=h.buepw3h6okkl)

[Gambar 4. 13 Halaman Input Transaksi Staff 78](#_heading=h.k6zil1mgsqa5)

[Gambar 4. 14 Data Transaksi Staff 79](#_heading=h.yk6itnjs4llm)

[Gambar 4. 15 Halaman Monitoring Pajak 79](#_heading=h.tym25nhovaf5)

**BAB I  
PENDAHULUAN**

### Latar Belakang

Perkembangan teknologi informasi telah membawa perubahan besar dalam tata kelola administrasi publik, termasuk dalam sistem perpajakan. Transformasi digital memungkinkan proses pengumpulan, pengelolaan, dan pelaporan pajak dilakukan secara terintegrasi dan efisien melalui sistem informasi pajak digital. Salah satu tantangan utama dalam sistem ini adalah kemampuan untuk mengklasifikasikan transaksi pajak dengantepat**,** terutama dalam membedakan antara Pajak Pertambahan Nilai (PPN)danPajakPenghasilan (PPh)**.** Klasifikasi yang akurat sangat penting karena berpengaruh langsung terhadap keakuratan perhitungan, pelaporan, serta penerimaan pajak pemerintah.

Dalam konteks modernisasi sistem pajak, algoritma kecerdasan buatan seperti *Decision Tree* memiliki potensi besar untuk meningkatkan kemampuan analisis dan pengambilan keputusan otomatis berbasis data transaksi. Algoritma ini mampu memetakan hubungan antara variabel seperti nilai transaksi, jenis barang atau jasa, status PKP, hingga metode pembayaran untuk menentukan klasifikasi pajak secara objektif. Berbagai studi menunjukkan efektivitas *Decision Tree* dalam bidang perpajakan. Hanafia Lubis & Saragih (2024) memperoleh akurasi 93,73% dalam klasifikasi kesadaran wajib pajak PBB, sementara Intan Diah Hardyatman & Hasan (2025) membuktikan bahwa model ini lebih unggul dibandingkan *Naive Bayes* dalam analisis sentimen publik terkait kebijakan kenaikan pajak. Hasil penelitian tersebut memperlihatkan kemampuan *Decision Tree* dalam menghasilkan klasifikasi yang transparan, mudah diinterpretasikan, dan dapat diimplementasikan pada sistem pajak digital daerah.

Dinas Perdagangan dan Perindustrian Kabupaten Bandung telah memiliki sistem informasi pajak digital sebagai langkah menuju efisiensi administrasi pajak. Meski demikian, proses klasifikasi transaksi di sistem tersebut masih dilakukan secara semi-manual dan bergantung pada validasi operator, sehingga rawan terjadi kesalahan identifikasi jenis pajak. Penerapan *Decision Tree* diharapkan mampu mengoptimalkan proses tersebut dengan membangun model klasifikasi berbasis data transaksi riil yang tersimpan dalam sistem. Model ini akan dikembangkan melalui tahapan pengumpulan data, pelatihan algoritma, dan integrasi ke sistem pajak digital agar dapat secara otomatis mengenali pola transaksi serta menentukan kategori pajak yang sesuai dengan akurasi tinggi dan waktu proses yang lebih singkat.

Penerapan *Decision Tree* dalam sistem informasi pajak digital daerah bukan hanya menjadi bentuk inovasi teknologi, tetapi juga langkah strategis menuju tata kelola pajak yang efisien, transparan, dan berbasis data. Dengan kemampuan mengklasifikasikan transaksi secara otomatis, sistem ini dapat membantu aparatur pajak mempercepat validasi, mengurangi kesalahan input manual, serta meningkatkan akurasi laporan penerimaan. Lebih jauh lagi, model ini dapat menjadi contoh implementasi nyata reformasi perpajakan digital yang berorientasi pada efektivitas dan akuntabilitas kinerja pemerintah daerah.

### 1.2 Rumusan Masalah

Transformasi digital di sektor perpajakan telah menjadi bagian penting dalam upaya meningkatkan efisiensi, transparansi, dan akurasi pengelolaan pajak, baik di tingkat nasional maupun daerah. Digitalisasi memungkinkan proses pelaporan dan klasifikasi pajak dilakukan secara lebih cepat dan terintegrasi. Namun demikian, kenyataan di lapangan menunjukkan bahwa sistem informasi pajak digital yang diterapkan, khususnya di lingkungan pemerintah daerah, belum sepenuhnya optimal dalam mengklasifikasikan jenis pajak secara otomatis, terutama dalam membedakan antara Pajak Pertambahan Nilai (PPN) dan Pajak Penghasilan (PPh). Permasalahan ini tidak hanya berdampak pada efisiensi sistem, tetapi juga berpotensi menimbulkan kesalahan dalam pencatatan dan pelaporan pajak.

Seiring meningkatnya kompleksitas dan volume transaksi, kebutuhan akan sistem yang mampu melakukan klasifikasi transaksi secara cerdas dan akurat semakin mendesak. Salah satu pendekatan yang potensial untuk menjawab tantangan ini adalah penerapan algoritma kecerdasan buatan, khususnya *Decision Tree*, yang telah terbukti efektif dalam proses klasifikasi di berbagai bidang. Namun, penerapan algoritma ini dalam konteks sistem perpajakan digital lokal, seperti pada Dinas Perdagangan dan Perindustrian Kabupaten Bandung, masih belum banyak dikaji secara mendalam. Berdasarkan latar belakang tersebut, dapat diidentifikasi beberapa permasalahan yang melatarbelakangi pentingnya penelitian ini, yaitu:

1. Belum Optimalnya Sistem Informasi Pajak Digital dalam Mengklasifikasi Jenis Pajak?
2. Rendahnya Akurasi Klasifikasi Transaksi PPN dan PPh pada Sistem Eksisting?
3. Kurangnya Implementasi Algoritma Kecerdasan Buatan untuk Pengambilan Keputusan Otomatis dalam Pajak Digital?
4. Belum Tersedianya Model Klasifikasi Transaksi Pajak Berbasis *Decision Tree* yang Diuji Secara Lokal?
5. Ketiadaan Rekomendasi Teknis dan Akademik dalam Pengembangan Sistem Pajak Digital Daerah Berbasis AI?

### 1.3 Batasan Masalah

Dalam sebuah kajian terapan, penetapan batasan masalah menjadi langkah penting untuk menjaga pembahasan tetap terarah dan fokus terhadap tujuan penelitian. Batasan masalah berfungsi untuk memperjelas ruang lingkup kajian, sekaligus menegaskan aspek-aspek yang termasuk maupun yang berada di luar cakupan analisis. Dengan adanya batasan yang jelas, penelitian dapat berfokus pada variabel yang relevan, terukur, dan mendukung pengujian hipotesis secara objektif.

Kajian ini difokuskan pada penerapan algoritma *Decision Tree* dalam proses klasifikasi transaksi pajak digital yang membedakan antara PPN dan PPh pada sistem informasi pajak digital di Dinas Perdagangan dan Perindustrian Kabupaten Bandung. Data yang digunakan mencakup variabel nilai transaksi, jenis barang atau jasa, status PKP, serta metode penarikan pajak. Model *Decision Tree* dibangun dan diuji menggunakan metrik akurasi, presisi, recall, dan F1-score. Pembahasan hanya dibatasi pada proses klasifikasi dan evaluasi kinerja model di tingkat pemerintah daerah, tanpa mencakup integrasi dengan sistem e-faktur nasional. Dengan batasan ini, penelitian diharapkan mampu menunjukkan sejauh mana algoritma *Decision Tree* dapat meningkatkan efisiensi dan ketepatan sistem klasifikasi pajak digital daerah.

1. Pembahasan difokuskan pada klasifikasi transaksi pajak digital yang berkaitan dengan (PPN) dan (PPh) dalam sistem informasi pajak digital milik Dinas Perdagangan dan PerindustrianKabupaten Bandung. Data yang digunakan dalam penelitian ini terbatas pada transaksi digital yang tercatat dalam sistem informasi pajak digital milik Dinas Perdagangan dan Perindustrian Kabupaten Bandung.
2. Evaluasi kinerja model hanya difokuskan pada pengukuran metrik objektif seperti akurasi, presisi, recall, dan F1-score untuk menilai tingkat ketepatan klasifikasi transaksi PPN dan PPh.
3. Cakupan pembahasan tidak mencakup aspek hukum, kebijakan perpajakan, maupun integrasi dengan sistem pajak nasional seperti e-Faktur dan e-Bupot, melainkan berfokus pada penerapan algoritma Decision Tree dalam konteks sistem pajak digital di tingkat pemerintah daerah.

### 1.4 Tujuan Laporan Akhir

Laporan akhir ini disusun dengan tujuan untuk mengembangkan solusi berbasis kecerdasan buatan dalam mendukung peningkatan efisiensi dan akurasi sistem informasi pajak digital di tingkat daerah. Melalui penerapan algoritma *Decision Tree*, penelitian ini diharapkan mampu membantu aparatur pajak dalam mengidentifikasi dan mengklasifikasikan jenis pajak secara otomatis berdasarkan data transaksi digital yang tersedia. Secara lebih spesifik, tujuan dari laporan akhir ini adalah sebagai berikut:

1. Mengembangkan dan mengimplementasikan model klasifikasi transaksi pajak digital berbasis algoritma *Decision Tree* guna membedakan jenis pajak antara PPN dan PPh secara akurat dan efisien.
2. Mengidentifikasi variabel atau fitur transaksi yang paling berpengaruh terhadap hasil klasifikasi jenis pajak dalam sistem informasi pajak digital, sehingga dapat digunakan sebagai dasar pengambilan keputusan dan pengembangan kebijakan berbasis data.

### 1.5 Manfaat Laporan Akhir Tugas akhir

Adapun manfaat dari pembangunan sistem dan penyusunan laporan proyek ini adalah sebagai berikut:

1. Meningkatkan Efisiensi dan Keamanan Pengelolaan Data Operasional Sistem ini mempermudah proses pencatatan stok masuk dan keluar serta riwayat harga beli bahan baku yang sebelumnya dilakukan secara manual menjadi terdigitalisasi dan terstruktur. Hal ini memastikan seluruh data tersimpan dengan rapi dalam *database*, sehingga meminimalisir risiko kehilangan data, memudahkan penelusuran riwayat transaksi, dan membuat pengelolaan administrasi usaha menjadi lebih efisien.
2. Mendukung Pengambilan Keputusan Strategis Berbasis Data Sistem memberikan kemampuan analisis tren tahunan menggunakan metode *Year Over Year* (YoY) yang disajikan dalam bentuk visualisasi grafik dan tabel. Informasi ini memberikan gambaran objektif mengenai fluktuasi harga dan pola kebutuhan stok dari tahun ke tahun, sehingga pemilik usaha dapat menentukan waktu pembelian bahan baku yang paling tepat dan hemat biaya, serta menghindari kekurangan atau penumpukan stok

### 1.6 Sistematika Laporan

Penyusunan laporan tugas akhir ini menggunakan sistematika penulisan sebagai Berikut:

**BAB 1 PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, tujuan laporan akhir, manfaat laporan akhir, serta sistematika laporan.

**BAB 2 TINJAUAN PUSTAKA**

Bab ini membahas mengenai penelitian terdahulu dan landasan teori yang menjadi kajian pustaka dalam penulisan laporan akhir ini. Serta, memaparkan masalah yang berhubungan dengan sistem yang akan dibuat.

**BAB 3. METODE PENELITIAN**

Bab ini berisi tentang metode perancangan sistem mulai dari alat dan bahan, pengumpulan data, hingga metode pengujian.

**BAB 4. HASIL DAN PEMBAHASAN**

Bab ini menjelaskan tentang hasil akhir dari tahapan sebelumnya mengenai perancangan sistem informasi, serta pembahahan sistem yang berupa penjabaran dari sistem yang dibuat.

**BAB 5 KESIMPULAN DAN SARAN** Bab ini berisikan penjelasan mengenai kesimpulan dan saran dari hasil akhir yang telah dibangun, untuk menjawab semua permasalahan memberikan saran yang berisi masukin untuk mengembangkan dan melengkapi sistem informasi yang sudah dirancang dan dibangun dimasa yang akan datang.

**DAFTAR PUSTAKA**

Daftar pustaka berikan informasi mengenai semua sumber hasil penelitian terdahulu yang digunakan dalam pembuatan laporan akhir ini, baik berupa jurnal, buku, artikel, maupun sumber-sumber internet.

# BAB II

## LANDASAN TEORI

### 2.1 Penelitian Terdahulu

Dalam penulisan tugas akhir ini, penulis memperoleh banyak sumber informasi dari berbagai penelitian terdahulu yang berkaitan dengan topik yang diangkat. Berikut adalah sejumlah penelitian yang berkaitan dengan tugas akhir penulis,

Tabel 2. 1 Penelitian Terdahulu

| **Nama Penulis** | **Tahun** | **Judul** | **Metode & hasil** | **Penerbit** | **Persamaan**  **& Perbedaan** |
| --- | --- | --- | --- | --- | --- |
| Tjandrida & Dermawan | 2025 | Implementasi Algoritma *Decision Tree* pada Sistem Informasi Manajemen Inventarisasi Fakultas Vokasi UNESA | Penggunaan *Decision Tree* dalam sistem inventarisasi perangkat dan aset kampus Sistem mampu mengklasifikasikan status barang aset dengan akurasi (tidak disebutkan nilai pasti publik) | Universitas Negeri Surabaya (UNESA) | Persamaan : menggunakan algoritma *Decision Tree* untuk klasifikasi sistem  Perbedaan : Fokus pada sistem inventarisasi aset, bukan klasifikasi transaksi pajak seperti PPN/PPh atau awareness pajak |
| Rokhimakhu  mullah , Ningsih, Firmansyah, Hanafiyah | 2024 | Analisis potensi e‑commerce melalui implementasi data mining dalam perpajakan: Sebuah studi komparasi | Deskriptif-komparatif; pendekatan data mining umum (tidak spesifik ke algoritma)  Menunjukkan bahwa e‑commerce merupakan potensi besar sumber pajak; data mining penting untuk tax gap dan identifikasi pajak | Juremi: Jurnal Riset Ekonomi | Persamaan :  membahas penggunaan data mining dalam perpajakan e‑commerce  Namun, Tidak menyertakan algoritma klasifikasi spesifik (Naïve Bayes / *Decision Tree*); tidak menggunakan data primer transaksi PPN/PPh secara teknis |
| Haris, Muhammad | 2022 | Evaluasi Digitalisasi Perpajakan Studi Kasus di KPP Pratama Magelang | Kualitatif deskriptif-analitisl.  Digitalisasi pajak meningkatkan efisiensi administrasi dan kepatuhan wajib pajak. | DSpace UII (Universitas Islam Indonesia), laporan penelitian internal | Persamaan :  Sama-sama mengkaji digitalisasi layanan perpajakan dan dampaknya terhadap efektivitas sistem perpajakan dan kepatuhan wajib pajak |
| Maskur & Wibowo | 2024 | *Taxpayer Awareness Classification Using Decision Tree and Naïve Bayes Methods* | *Decision Tree* & Naïve Bayes; klasifikasi 666.580 data PBB  *Decision Tree*: 93,73% akurasi; Naïve Bayes: 85,61% akurasi | Jurnal JAIC (Polibatam) | Persamaan ; Sama-sama menggunakan *Decision Tree* untuk klasifikasi pajak |
| Septaraja, Joannes, Radhi, Parhusip. | 2024 | Implementasi Algoritma *Decision Tree* untuk Prediksi Efisiensi Biaya Bensin Kendaraan Bermotor Parenggean Menuju Palangkaraya | Observasi langsung (jenis kendaraan, jarak tempuh, kondisi jalan); model *Decision Tree* (C4.5)  Akurasi mencapai 90%; pohon keputusan memperlihatkan pola konsumsi bahan bakar yang efektif | Informatech: Jurnal Ilmiah Informatika dan Komputer | Persamaan : menggunakan algoritma *Decision Tree* untuk klasifikasi dan efisiensi proses berbasis data |
| Ramadhon | 2024 | Implementasi Algoritma *Decision Tree* untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank | Penggunaan *Decision Tree* pada dataset pelanggan bank (~74.258 baris, fitur saldo, lokasi, status) | Universitas Djuanda | Persamaan : menerapkan *Decision Tree* untuk klasifikasi berbasis atribut sistem  Perbedaan :  Dataset domain berbeda (nasabah bank), bukan transaksi perpajakan PPN/PPh |

### 2.2 Landasan Teori

Dalam penelitian ini, penulis menyusun landasan teori yang didapat dari analisis terhadap berbagai sumber literatur yang berkaitan dan relevan terhadap topik utama yang akan diteliti. Berikut adalah landasan teori yang digunakan :

#### 2.2.1 Algoritma Decision Tree

* 1. Pengertian Algoritma *Decision Tree*

*Decision Tree* merupakan salah satu algoritma yang sangat dikenal dalam pendekatan *supervised learning*, yang sering dimanfaatkan untuk menyelesaikan permasalahan klasifikasi dan regresi. Cara kerja algoritma ini adalah dengan membagi kumpulan data ke dalam sub-kelompok berdasarkan nilai-nilai atribut tertentu secara berulang, hingga terbentuk struktur menyerupai pohon keputusan. Struktur tersebut terdiri dari simpul akar (*root node*), simpul internal (*internal node*), dan simpul daun (*leaf node*) yang menggambarkan hasil akhir dari proses pengambilan keputusan (Gunawan & Astuti, 2022).

Menurut Kirmani dkk. (2024:65), *Decision Tree* menggunakan metode pembelajaran rekursif, di mana data dibagi berdasarkan fitur yang paling membedakan. Pendekatan ini sangat efektif untuk mengolah data bertipe kategorikal maupun numerik. Setiap percabangan dalam struktur pohon mencerminkan keputusan atas atribut tertentu, sementara ujung-ujung daun menunjukkan hasil dari klasifikasi atau nilai hasil regresi (Jakse, 2024:117).

Proses pemisahan data dalam algoritma ini biasanya mengacu pada tiga ukuran utama, yaitu *Entropy, Information Gain,* dan *Gini Index*. *Entropy* berfungsi untuk mengukur tingkat ketidakpastian dalam distribusi suatu data. Jika nilai *entropy* rendah, maka data tersebut lebih homogen. Sementara itu, Information Gain mengindikasikan besarnya penurunan entropy setelah data dipisah berdasarkan suatu atribut. Sedangkan Gini Index digunakan untuk mengukur ketidakhomogenan atau impurity dalam data dan sering kali diaplikasikan pada algoritma CART (*Classification and Regression Tree*) (Marlina et al., 2023)

* 1. Metode Pemisahan Data (*Splitting Criteria)*

Adapun rumus-rumus dasar yang digunakan adalah sebagai berikut:

Entropy:

Entropy(S)=-∑i=1npilog⁡2(pi)

Dimana pi adalah proporsi data kelas ke-i dalam S.

Information Gain:

Gain(S,A)=Entropy(S)-∑v∈Values(A)∣Sv∣∣S∣⋅Entropy(Sv)

Gini Index:

Gini(D)=1-∑j=1m(pj)2

Algoritma Decision Tree dikenal luas karena kemudahan interpretasinya, efisiensi dalam hal komputasi, serta kemampuannya menghasilkan model yang transparan. Renganathan (2021:6) menyebutkan dalam bukunya Machine Learning for Data Scientists, bahwa struktur pohon sangat mudah dipahami karena menyerupai cara berpikir manusia dalam mengambil keputusan.

Namun demikian, seperti yang dikemukakan oleh Ambareesh dan Augustine (2023:53), salah satu kelemahan utama algoritma ini adalah kecenderungannya mengalami overfitting terhadap data pelatihan. Untuk mengatasi hal tersebut, metode pruning atau pemangkasan cabang digunakan, yakni dengan menghapus bagian pohon yang tidak berkontribusi besar terhadap akurasi model.

#### 2.2.2 Konsep Dasar *Decision Tree*

*Decision Tree* merupakan salah satu metode pembelajaran terawasi (*supervised learning*) yang digunakan baik untuk masalah klasifikasi maupun regresi. Konsep dasar dari algoritma ini adalah memecah suatu himpunan data ke dalam kelompok-kelompok yang lebih kecil berdasarkan nilai atribut tertentu, sehingga pada akhirnya terbentuk struktur pohon keputusan. Struktur ini terdiri atas simpul akar (*root node*), simpul internal (*internal node*), dan simpul daun (*leaf node*). Simpul akar melambangkan keseluruhan himpunan data, simpul internal merepresentasikan kondisi atau uji atribut tertentu, sedangkan simpul daun berisi hasil akhir klasifikasi atau prediksi (Gunawan & Astuti, 2022).

Proses pembentukan pohon keputusan dilakukan melalui pendekatan rekursif, yaitu pemisahan data yang terus berulang sampai mencapai kondisi tertentu. Pada setiap langkah, algoritma memilih atribut yang paling baik untuk memisahkan data berdasarkan suatu kriteria. Kriteria pemisahan yang sering digunakan antara lain Entropy, Information Gain, dan Gini Index. Entropy digunakan untuk mengukur tingkat ketidakpastian distribusi data, semakin kecil nilai entropy, semakin homogen data dalam suatu node. Information Gain menunjukkan besarnya penurunan entropy setelah data dipisahkan berdasarkan suatu atribut. Sementara itu, Gini Index mengukur ketidakhomogenan atau impurity suatu node dan banyak digunakan dalam algoritma *Classification and Regression Tree* (CART) (Marlina et al., 2023).

Menurut Kirmani dkk. (2024), prinsip utama *Decision Tree* adalah membagi data ke dalam dua atau lebih cabang berdasarkan nilai atribut yang paling diskriminatif, lalu mengulangi proses tersebut pada setiap cabang. Proses ini berhenti apabila node yang terbentuk telah homogen, kedalaman maksimum tercapai, atau jumlah data pada node terlalu kecil untuk dilanjutkan. Dengan demikian, pohon keputusan dibangun dengan memaksimalkan homogenitas setiap node, sehingga aturan klasifikasi dapat ditarik dari jalur akar hingga daun dalam bentuk aturan logis *if–then* yang mudah dipahami (Jakse, 2024).

Keunggulan utama dari konsep *Decision Tree* adalah sifatnya yang transparan dan interpretatif. Struktur pohon menyerupai pola pikir manusia dalam mengambil keputusan, sehingga hasil klasifikasi dapat ditelusuri dengan jelas. Renganathan (2021) menegaskan bahwa *Decision Tree* termasuk model *white-box*, artinya setiap keputusan yang dihasilkan dapat dipahami oleh pengguna, berbeda dengan model *black-box* seperti jaringan saraf tiruan. Karakteristik ini menjadikan *Decision Tree* banyak digunakan pada bidang yang menuntut akuntabilitas, misalnya kesehatan, perbankan, dan perpajakan.

#### 2.2.3 Tahapan Algoritma *Decision Tree*

Algoritma *Classification and Regression Tree* (CART) merupakan salah satu varian dari *decision tree* yang menggunakan pemisahan biner pada setiap node dengan pemilihan atribut berdasarkan nilai Gini Index (Sharma & Iqbal, 2023). Tahapan penerapan algoritma ini diawali dengan perumusan tujuan klasifikasi dan penentuan variabel prediktor yang relevan dengan permasalahan penelitian. Selanjutnya dilakukan persiapan data yang mencakup proses pembersihan, penanganan nilai hilang, serta pembagian data menjadi set pelatihan dan set pengujian menggunakan teknik stratifikasi agar proporsi kelas tetap seimbang (Chen et al., 2024). Setelah data siap, pohon dibentuk mulai dari akar dengan melibatkan seluruh data pelatihan. Pada setiap node dievaluasi seluruh kemungkinan pemisahan atribut, baik numerik maupun kategorikal, lalu dihitung nilai Gini untuk masing-masing pemisahan. Rumus Gini Index dirumuskan sebagai:

dengan adalah proporsi data kelas ke dalam node . Pemisahan terbaik dipilih berdasarkan penurunan nilai Gini terbesar sehingga node hasil pembagian semakin homogen (Orounla, 2023). Tahap berikutnya adalah proses pertumbuhan pohon yang berlangsung secara rekursif hingga kondisi berhenti tercapai, misalnya ketika seluruh data dalam node telah homogen, kedalaman maksimum tercapai, atau ukuran sampel terlalu kecil untuk dilanjutkan. Pohon yang terbentuk sering kali bersifat kompleks dan rentan mengalami *overfitting*. Oleh sebab itu, CART menerapkan metode pemangkasan dengan pendekatan Minimal *Cost-Complexity Pruning* (MCCP) untuk menyederhanakan struktur pohon tanpa mengurangi kualitas prediksi secara signifikan.

dengan adalah tingkat kesalahan pohon , jumlah daun, dan merupakan parameter kompleksitas (Katsevich, 2022). Pemilihan nilai umumnya dilakukan melalui uji silang (*cross-validation*) sehingga diperoleh pohon yang sederhana, mudah diinterpretasikan, dan memiliki kinerja prediksi yang stabil.

Tahap akhir adalah evaluasi model menggunakan metrik klasifikasi biner, seperti akurasi, presisi, recall, F1-score, serta ROC-AUC untuk mengukur kinerja pemisahan antar kelas (Mohammadi-Pirouz et al., 2024). Hasil akhir pohon kemudian divisualisasikan dalam bentuk aturan “jika maka” sehingga mudah dipahami oleh pihak non-teknis, misalnya staf dinas pajak. Dengan demikian, tahapan algoritma CART menyediakan kerangka yang sistematis dalam mengolah data transaksi pajak digital menjadi keputusan klasifikasi yang jelas dan dapat dipertanggungjawabkan.

#### 2.2.4 Langkah-Langkah Algoritma *Decision Tree* CART

Dalam algoritma *Classification and Regression Tree* (CART), proses pembentukan pohon keputusan dimulai dengan penentuan tujuan klasifikasi dan variabel target yang ingin diprediksi. Setelah itu dilakukan tahap persiapan data, yaitu pembersihan data, penanganan nilai yang hilang, serta pembagian dataset menjadi data latih dan data uji secara proporsional. Pada tahap berikutnya dihitung nilai Gini Index untuk setiap atribut. Gini Index digunakan untuk mengukur tingkat ketidakmurnian suatu node, dan atribut dengan penurunan nilai Gini terbesar akan dipilih sebagai dasar pemisahan.

Pemisahan pada algoritma CART selalu bersifat biner, sehingga setiap node hanya menghasilkan dua cabang. Proses ini dijalankan secara rekursif hingga kondisi berhenti tercapai, misalnya ketika seluruh data dalam node telah homogen, kedalaman pohon sudah mencapai batas maksimum, atau jumlah data dalam node terlalu sedikit untuk dilanjutkan. Pohon yang dihasilkan sering kali terlalu kompleks sehingga berisiko mengalami *overfitting*. Untuk mengatasinya, CART menerapkan metode Minimal *Cost-Complexity Pruning* (MCCP), yaitu memangkas cabang yang tidak memberikan kontribusi besar terhadap akurasi model.

Tahap terakhir adalah evaluasi kinerja pohon yang telah terbentuk. Evaluasi dilakukan dengan menggunakan metrik klasifikasi, antara lain akurasi, presisi, recall, F1-score, dan ROC-AUC. Melalui rangkaian tahapan tersebut, algoritma CART mampu menghasilkan pohon keputusan yang tidak hanya akurat dalam memprediksi data, tetapi juga tetap sederhana dan mudah diinterpretasikan (Orounla, 2023; Sharma & Iqbal, 2023).

Untuk memperjelas tahapan algoritma CART, alur prosesnya digambarkan dalam bentuk flowchart seperti terlihat pada Gambar 2.1.

A diagram of milk production

AI-generated content may be incorrect.

*Gambar 2. 1 Flowchart* Tahapan Algoritma *CART*

(Adaptasi Sharma & Iqbal, 2023)

Gambar 2.1 memberikan ringkasan tahapan algoritma CART secara sistematis. Dari tabel tersebut terlihat bahwa setiap tahap memiliki fungsi penting, mulai dari penentuan tujuan hingga evaluasi model. Ringkasan ini tidak hanya memperjelas alur pada flowchart sebelumnya, tetapi juga membantu pembaca memahami hubungan antar tahapan secara lebih ringkas. Dengan adanya tabel ini, setiap langkah dapat dipahami secara lebih terstruktur, sehingga mempermudah perbandingan antara teori yang dijelaskan dalam bentuk naratif dan implementasinya dalam bentuk visual. Penyajian tabel bersama flowchart diharapkan dapat memberikan gambaran yang lebih utuh mengenai proses pembentukan pohon keputusan dengan algoritma CART.

Untuk memperjelas tahapan yang telah dijelaskan pada uraian sebelumnya, ringkasan proses algoritma CART disajikan pada Tabel 2.2

Tabel Ringkasan

*Tabel 2. 2 Tabel* Ringkasan Kegiatan

| **Tahap** | **Kegiatan** | **Keterangan** |
| --- | --- | --- |
| 1. | Menentukan tujuan klasifikasi | Menetapkan variabel target, misalnya transaksi PPN dan PPh. |
| 2. | Persiapan data | Pembersihan data, penanganan nilai hilang, pembagian data latih dan uji. |
| 3. | |  | | --- |   Menghitung Gini Index | Mengukur impuritas node dengan rumus |
| 4. | Menentukan atribut terbaik | Memilih atribut dengan penurunan Gini terbesar; pemisahan selalu biner. |
| 5. | Proses rekursif | Melakukan pemisahan berulang hingga kondisi berhenti. |
| 6. | Pruning (MCCP) | Pemangkasan cabang pohon untuk mengurangi risiko *overfitting*. |
| 7. | Evaluasi model | Menggunakan akurasi, presisi, recall, F1-score, dan ROC-AUC. |

Tabel 2.2 memberikan ringkasan tahapan algoritma CART secara sistematis. Dari tabel tersebut terlihat bahwa setiap tahap memiliki fungsi penting, mulai dari penentuan tujuan hingga evaluasi model. Ringkasan ini sekaligus memperjelas alur pada flowchart sebelumnya dan mempermudah pembaca dalam memahami proses pembentukan pohon keputusan secara keseluruhan. Selain itu, penyajian tabel membantu menunjukkan keterkaitan logis antar tahapan, sehingga tampak jelas bagaimana proses klasifikasi dimulai dari tahap persiapan data, berlanjut pada perhitungan Gini Index, pemilihan atribut terbaik, hingga pemangkasan pohon dan evaluasi model. Dengan demikian, tabel ini tidak hanya merangkum uraian yang telah dipaparkan dalam bentuk naratif, tetapi juga menjadi jembatan antara teori yang dijelaskan pada bab ini dengan implementasi praktis yang akan dibahas pada bab selanjutnya. Oleh karena itu, keberadaan tabel sangat penting untuk memberikan gambaran yang lebih utuh mengenai algoritma CART.

#### 2.2.5 Klasifikasi Transaksi PPN dan PPh

1. Pengertian Transaksi Perpajakan

Dalam konteks perpajakan Indonesia, transaksi perpajakan adalah aktivitas ekonomi yang melibatkan pertukaran barang, jasa, maupun penghasilan yang memiliki konsekuensi terhadap kewajiban perpajakan, baik berupa Pajak Pertambahan Nilai (PPN) maupun Pajak Penghasilan (PPh). Menurut Direktorat Jenderal Pajak (2022), setiap transaksi yang dilakukan oleh Wajib Pajak berpotensi menjadi objek pajak apabila memenuhi syarat-syarat yang ditetapkan dalam peraturan perpajakan.

1. Pengertian dan Karakteristik PPN dan PPh

Pajak merupakan kontribusi wajib yang dikenakan oleh negara kepada setiap individu atau badan usaha atas pendapatan atau konsumsi yang dilakukan, dengan tujuan membiayai pengeluaran negara untuk kepentingan umum. Dalam sistem perpajakan Indonesia, dua jenis pajak utama yang sering terkait dengan transaksi ekonomi adalah Pajak Pertambahan Nilai (PPN) dan Pajak Penghasilan (PPh). Keduanya memiliki karakteristik, mekanisme pemungutan, dan objek yang berbeda, sehingga penting untuk dipahami dalam konteks klasifikasi transaksi digital.

Pajak Pertambahan Nilai (PPN) merupakan pajak tidak langsung yang dikenakan atas konsumsi Barang Kena Pajak (BKP) dan/atau Jasa Kena Pajak (JKP) di dalam wilayah Indonesia. Pajak ini dibayarkan oleh konsumen akhir, namun dipungut dan disetor oleh Pengusaha Kena Pajak (PKP) yang melakukan penyerahan barang atau jasa. Karakteristik utama PPN adalah bersifat multistage, artinya dikenakan di setiap rantai distribusi, namun bersifat netral karena sistem pengkreditan pajak masukan terhadap pajak keluaran. Besaran tarif PPN yang berlaku di Indonesia saat ini adalah 11% sebagaimana diatur dalam Undang-Undang No. 7 Tahun 2021 tentang Harmonisasi Peraturan Perpajakan.

Sementara itu, Pajak Penghasilan (PPh) merupakan pajak yang dikenakan atas setiap tambahan kemampuan ekonomis yang diterima atau diperoleh Wajib Pajak, baik berasal dari dalam negeri maupun luar negeri, yang dapat digunakan untuk konsumsi atau menambah kekayaan. PPh bersifat langsung, yaitu dibayar sendiri oleh pihak yang menerima atau memperoleh penghasilan. Dalam praktiknya, terdapat berbagai jenis PPh yang diklasifikasikan berdasarkan sumber penghasilan dan subjek pajaknya. Di antaranya adalah PPh Pasal 21 yang dikenakan atas penghasilan pegawai, PPh Pasal 22 atas kegiatan impor, PPh Pasal 23 atas penghasilan dari jasa tertentu, dan PPh Pasal 4 ayat (2) yang bersifat final untuk objek pajak tertentu seperti sewa atau hadiah undian. Masing-masing jenis PPh memiliki tarif, objek, dan perlakuan administratif yang berbeda.

#### 2.2.6 Pajak Digital

Pajak digital merupakan bentuk adaptasi sistem perpajakan terhadap perkembangan ekonomi berbasis teknologi informasi, khususnya aktivitas ekonomi digital seperti *e-commerce,* layanan digital lintas negara, serta transaksi daring lainnya. Seiring meningkatnya volume dan kompleksitas aktivitas ekonomi digital, pemerintah di berbagai negara, termasuk Indonesia, mulai mengembangkan kebijakan dan sistem pemungutan pajak digital yang lebih efisien dan akuntabel. Menurut Kementerian Keuangan Republik Indonesia (2022), pajak digital bertujuan untuk menciptakan keadilan fiskal antara pelaku usaha konvensional dan digital, serta memastikan kontribusi adil dari perusahaan teknologi global terhadap pendapatan negara. Pajak digital mencakup berbagai jenis pajak seperti Pajak Pertambahan Nilai (PPN) atas barang dan jasa digital, Pajak Penghasilan (PPh) atas penghasilan dari platform digital, serta Pajak Perdagangan Melalui Sistem Elektronik (PMSE).

#### 2.2.7 Informasi Pajak Digital

Informasi pajak digital merupakan bentuk transformasi sistem perpajakan konvensional ke dalam sistem berbasis teknologi informasi yang terintegrasi. Perubahan ini didorong oleh meningkatnya volume transaksi ekonomi digital dan kebutuhan akan efisiensi, transparansi, serta akurasi dalam pengelolaan data perpajakan. Sistem informasi pajak digital mencakup proses pengumpulan, pengolahan, penyimpanan, dan pelaporan data perpajakan secara elektronik melalui platform yang dikembangkan oleh otoritas pajak maupun lembaga yang berwenang. Sistem ini mendukung otomatisasi berbagai aktivitas perpajakan seperti pelaporan Surat Pemberitahuan (SPT), penerbitan faktur pajak elektronik (e-faktur), pembayaran pajak melalui e-billing, serta pengawasan dan klasifikasi transaksi wajib pajak.

Ciri utama dari informasi pajak digital adalah kemampuannya dalam menangani data dalam jumlah besar secara cepat dan akurat, serta tersedianya fitur klasifikasi transaksi berdasarkan jenis pajak secara otomatis.

#### 2.2.8 Peran *Decision Tree* dalam Klasifikasi Transaksi Pajak

Algoritma *Decision Tree* merupakan salah satu metode dalam teknik *supervised learning* yang sangat populer dalam proses klasifikasi dan prediksi, termasuk dalam konteks sistem informasi perpajakan digital. Dalam klasifikasi transaksi pajak, seperti klasifikasi transaksi yang dikenai Pajak Pertambahan Nilai (PPN), Pajak Penghasilan (PPh), atau keduanya, *Decision Tree* berfungsi sebagai alat bantu analisis yang mampu memproses sejumlah atribut transaksi untuk menghasilkan keputusan klasifikasi secara otomatis dan sistematis. Atribut-atribut yang biasanya dianalisis meliputi jenis barang/jasa, status rekanan (PKP atau non-PKP), jenis transaksi (pembelian atau penjualan), serta nilai transaksi dan kode akun. Berdasarkan atribut-atribut tersebut, pohon keputusan membentuk struktur bercabang yang merepresentasikan proses logika “jika-maka”, sehingga memudahkan sistem dalam menetapkan jenis pajak yang relevan dengan cepat dan akurat.

penyetoran, dan pelaporan pajak sepenuhnya berada di tangan wajib pajak (Direktorat Jenderal Pajak, 2021). Oleh karena itu, laporan pajak menjadi titik krusial dalam menjamin kepatuhan dan kejujuran fiskal.

#### 2.2.9 Metode SDLC (Software Development Life Cycle)

Metode SDLC (Software Development Life Cycle) adalah langkah-langkah yang diambil untuk merancang dan memodifikasi sistem, serta pola dan cara yang diterapkan dalam menciptakan sistem perangkat lunak. Metode SDLC diciptakanuntukmendukung dalah proses pembuatan produk (Setiawan, 2021). Berikut adalah beberapa metode SDLC dalam pengembangan perangkat lunak yaitu:

1. Metode Waterfall Metode Waterfall

merupakan pendekatan dalam pengembangan perangkat lunak yang mengedepankan proses kerja sistem secara berurutan atau linear. Ini berarti bahwa setiap langkah dalam proses pengembangan harus dilaksanakan dalam urutan tertentu, sehingga fase berikutnya baru dapat dimulai setelah fase 13 sebelumnya telah rampung. Sebagai contoh, fase ketiga hanya dapat dilaksanankan setelah fase pertama dan kedua selesai (Fitri Khoiry Tamami Salam & Septanto, 2024). Metode Waterfall ini menawarkan pendekatan urutan tahapan kehidupan perangkat lunak yang idmulai dari Analisis kebutuhan, desain sistem, implementasi, pengujian, pemeliharaan. (Supiyandi et al., 2022).

A diagram of a system

AI-generated content may be incorrect.

*Gambar 2. 2* Metode *Waterfall*

Sumber : Supiyandi et al., 2022

Dari penjelasan dia atas, berikut merupakan urutan langkah-langkah dalam metode pengembangan sistem informasi :

* 1. Analisis Kebutuhan*,* ditahap ini pengembang perlu memahami semua informasi terkait kebutuhan perangkat lunak, seperti fungsi yang diharapkan pengguna dan batasan yang ada. Informasi ini umumnya dikumpulkan melalui wawancara, survei, atau diskusi. Setelahnya, informasi tersebut dianalisis untuk memperoleh data lengkap mengenai kebutuhan pengguna terhadap perangkat lunak yang dikembangkan.
  2. Desain Sistem*,* dilakukan sebelum coding dimulai untuk memberikan gambarang menyeluruh tentang apa yang harus dikerjakan dan bagaimana bentuk sistem yang diinginkan. Ini juga membantu menentukan kebutuhan perangkat keras dan sistem, serta mendefinisikan arsitektur keseluruhan dari sistem yang dibuat.
  3. Implementasi, pada tahap ini proses penulisan kode dilakukan. Pembuatan perangkat lunak akan dibagi menjadi modul-modul kecil yang akan digabungkan pada tahap selanjutnya. Selain itu, tahap ini juga akan dilakukan evaluasi mendalam terhadap modul yang telah dibuat untuk memastikan apakah sudah memenuhi fungsi yang di harapkan.

1. Metode *Prototype*

Metode *Prototype* merupakan suatu pendekatan dalam analisis dan perancangan yang memungkinkan pengguna berkontribusi dalam menetapkan kebutuhan serta merancang sistem yang akan dikerjakan untuk memenuhi kebutuhan tersebut. Langkah-langkah dalam metode prototype adalah sebagai berikut :

A diagram of a process

AI-generated content may be incorrect.

*Gambar 2. 3* Metode *Prototype*

Sumber : (Sari et al., 2023)

* 1. Pengumpulan kebutuhan, pada tahap ini dilakukan untuk menjelaskan kebutuhan sistem yang akan dikembangkan berdasarkan keinginan dan kebutuhan pengguna
  2. Membangun *prototyping*, setelah mengumpulkan informasi yang diperlukan, langkah berikutnya adalah membuat ilustrasi sistem yang akan dibangun melalui *flowchart, data flow diagram (DFD), dan entity relationship diagram (ERD).*
  3. Evaluasi *prototyping*, setelah *prototype* selesai dibangun, langkah berikutnya

adalah melakukan evaluasi terhadap *prototype* dengan pengguna, dimana *prototype* akan disesuaikan dengan harapan pengguna dan akan memberikan umpan balik dalam bentuk masukin agar sistem lebih mudah dioperasikan.

* 1. Mengkodekan sistem, setelah tahap evaluasi perancangan *prototype* selesai dan telah disetujui, maka proses tersebut akan diterjemahkan ke dalam bahasa pemrograman yang sesuai.
  2. Menguji sistem, setelah proses pengkodean sistem selesai, langkah selanjutnya adalah melakukan pengujian terhadap fitur dan konten sistem yang telah dibuat.
  3. Evaluasi sistem, setelah pengujian sistem, tahap berikutnya adalah pengguna yang akan melakukan evaluasi apakah sistem tersebuth telah memenuhi harapan.
  4. Penggunaan sistem, tahap terakhir adalah sistem yang sudah siap digunakan diserahkan kepada pengguna (Sari et al., 2023).

#### 2.2.10 Native PHP

Native PHP merupakan pendekatan pemrograman *web* yang mengandalkan PHP murni tanpa menggunakan *framework*. Meskipun *framework* seperti *Laravel* atau *CodeIgniter* mempercepat proses pengembangan dengan menyediakan struktur bawaan, penggunaan native PHP memberikan fleksibilitas penuh bagi pengembang untuk mengatur arsitektur sistem secara manual. Dalam buku *PHP Programming: Developing Dynamic Web Applications* karya Mitchell (2021:45), dijelaskan bahwa pengembangan aplikasi dengan native PHP mendorong penguasaan penuh terhadap sintaksis dan alur logika, sehingga cocok digunakan untuk edukasi dasar dan proyek yang memerlukan struktur khusus. Hal ini diperkuat oleh Roberts (2020:67) dalam bukunya *Pure PHP Development Handbook*, yang menekankan bahwa native PHP cocok untuk pengembangan sistem informasi yang ringan, terutama di lingkungan dengan sumber daya terbatas.

#### 2.2.11 HTML

HTML (*HyperText Markup Language*) merupakan bahasa standar yang digunakan untuk membuat dan menyusun halaman web. HTML bekerja dengan cara memberikan struktur terhadap elemen-elemen dalam sebuah dokumen web seperti teks, gambar, tautan, tabel, dan elemen multimedia lainnya. HTML bersifat *markup*, artinya menggunakan tag-tag tertentu untuk menandai bagian-bagian konten agar dapat dikenali dan ditampilkan oleh browser.

#### 2.2.12 Tailwind Css

Tailwind CSS adalah kerangka kerja *utility-first* yang menyediakan kelas-kelas kecil siap pakai (seperti flex, pt-4, text-center, bg-blue-500) untuk merangkai antarmuka langsung di markup tanpa harus menulis CSS kustom sejak awal. Pendekatan ini memindahkan banyak keputusan desain ke tingkat komposisi kelas pada elemen, sehingga pengembang dapat bergerak cepat dengan konsistensi visual yang terjaga dan prinsip *design tokens* yang eksplisit. Dokumentasi resmi menekankan bahwa utilitas Tailwind terhubung ke variabel tema (palet warna, skala tipografi, bayangan, dan seterusnya) dan dapat diperluas ketika dibutuhkan nilai satuan melalui sintaks *arbitrary values* berbasis tanda kurung. Dengan cara ini, Tailwind bertindak sebagai lapisan bahasa desain yang dekat dengan CSS namun lebih terstruktur untuk tim, sekaligus menjaga keluaran CSS tetap efisien karena hanya kelas yang terpakai yang di-*emit* dalam proses build.

#### 2.2.13 Database

Database merupakan sekumpulan data yang tersimpan dengan teratur di dalam komputer, sehingga dapat diatur oleh program komputer untuk memperoleh informasi dari database tersebut (Aswiputri, 2022). Database adalah suatu sistem yang dirancang untuk mengelola, menyimpan dan mengambil informasi dengan mudah. Database terdiri dari sekumpulan data yang terorganisir untuk satu atau lebih tujuan, dalam bentuk digital. Database digital ini dikelola dengan menggunakan sistem manajemen basis data (DBMS), yang menyimpan isi dari database, memungkinkan pembuatan dan pemeliharaan data, serta pencarian dan akses lainnya. Beberapa contoh database yang tersedia saat ini meliputi : MySQL, SQL Server, Ms Access, Oracle, dan PostgreSQL (Ramadhan & Mukhaiyar, 2020).

Berdasarkan uraian diatas, dapat disimpulkan bahwa Database adalah sistem terorganisir yang menyimpan data secara digital di dalam komputer, sehingga dapat dikelola, diakses, dan dimanfaatkan dengan mudah menggunakan sistem manajemen basis data (DBMS).

#### 2.2.14 Mysql

MySQL merupakan salah satu jenis *server* database yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses databasenya. Lisensi MySQL adalah FOSS License Exception dan juga tersedia versi komersialnya. MySQL dapat dijalankan di berbagai sistem operasi, termasuk Windows dan Linux. Untuk mempermudah pengelolaan MySQL, Anda dapat menggunakan perangkat lunak tertentu, salah satunya adalah phpMyAdmin (Ramadhan & Mukhaiyar, 2020).

#### 2.2.15 PHPMyAdmin

PhpMyAdmin adalah aplikasi atau alat berbasis *open source* yang bisa kita gunakan secara gratis untuk melakukan pemrograman atau administrasi pada database MySQL. PhpMyAdmin menggunakan bahasa PHP dalam pemrogramannya dan mendukung berbagai operasi MySQL, termasuk pengelolaan basis data, tabel, kolom, relasi, pengguna, izin, dan sebagainya (Ramadhan & Mukhaiyar, 2020).

#### 2.2.16 Laragon

Laragon adalah sebuah perangkat lunak gratis yang menyertakan berbagai sistem operasi sebagai *localhost* atau *server* mandiri. Laragon menawarkan beragam layanan, alat, dan fitur, termasuk Apache, PHP Server, PhpMyAdmin, MySQL, Memcached, Redis, Composer, Xdebug, Cmdre, dan Laravel (Budiman et al., 2023).

#### 2.2.17 Visual Studio Code

*Visual Studio Code* adalah software editor kode yang ringan tetapi kuat, berfungsi di desktop dan dapat digunakan pada windows, macOS, dan linux. Alat ini dilengkapi dengan dukungan bawaan untuk JavaScript, TypeScript, dan Node.Js, serta memiliki ekosistem ekstensi yang luas untuk Bahasa pemrogramann lain (seperti C++, C#, Java Phyton, PHP, dan Go) serta *runtime* (seperti .NET dan Unity) (Kurniawan & Romzi, 2022).

#### 2.2.18 Blackbox

*Blackbox* merupakan metode pengujian yang menitikberatkan pada kebutuhan fungsi dari sebuah Aplikasi. Seorang penguji memiliki kemampuan untuk merancang *Test Case* dan menilai kebutuhan fungsional Aplikasi. Sasaran dari pengujian dengan metode *blackbox* adalah untuk memperlihatkan bagaimana suatu fungsi dalam Aplikasi beroperasi, serta mengecek apakah hasil yang diperoleh sesuai dengan harapan ketika diberi masukkan data (Sasongko et al., 2021).

#### 2.2.19 Python

Python adalah bahasa pemrograman serbaguna yang menonjol karena keterbacaan sintaks, ekosistem pustaka yang luas, serta filosofi “batteries included” yang memudahkan pemula dan tetap memadai bagi proyek industri besar. Dalam lima tahun terakhir, evolusinya sangat terasa pada tiga sumbu: kinerja runtime, pengetikan statis (typing), dan ekosistem kemasan/alat. Dari sisi adopsi, survei resmi Python Software Foundation (PSF) bersama JetBrains pada 2023–2024 menegaskan daya tarik Python di berbagai domain (web, data, AI/ML), sekaligus menunjukkan basis pengguna yang terus tumbuh lintas level pengalaman. Hasil survei 2023 dirilis Agustus 2024 dan edisi 2024 dirilis 18 Agustus 2025; keduanya menggambarkan Python sebagai salah satu bahasa yang paling sering dipakai dan diminati oleh lebih dari 25–30 ribu responden global.

#### 2.2.20 Unified Modeling Language (UML)

*Unified Modeling Language (UML)* adalah suatu representasi visual yang sering dipakai untuk mendeskripsikan sistem yang berorientasi objek, dan banyak pengembang menggunakan bahasa ini (dalam format diagram) untuk merancang sistem mereka (Siking et al., 2023). Dibawah ini adalah beberapa jenis diagram *Unified Modeling Language (UML)* sebagai berikut:

1. *Use Case Diagram*

Menurut Rasiban et al. (2024), *Use Case Diagram* merupakan salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan dalam pengembangan perangkat lunak untuk secara visual menunjukkan hubungan antara berbagai aktor (baik pengguna maupun sistem dari luar) dan sebuah sistem. Diagram ini memperlihatkan cara pengguna berinteraksi dengan sistem untuk mencapai tujuan tertentu. Berikut adalah simbol-simbol dari *use case diagram* yaitu sebagai berikut :

*Tabel 2. 3* Simbol-simbol *Use Case Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
| --- | --- | --- | --- |
| 1 | A black stick figure with a white background  AI-generated content may be incorrect. | Actor | Menspesifikasikan kumpulan peran yang dijalankan oleh pengguna saat berinteraksi dengan use case. |
| 2 | A black line with a white background  AI-generated content may be incorrect. | Dependency | Hubungan di mana perubahan yang terjadi pada element yang berdiri sendiri (independent) akan mempengaruhi element yang bergantung padanya. |
| 3 | A black line on a white background  AI-generated content may be incorrect. | Generalization | Hubungan di mana objek (descendent) berbagai struktur data dan perilaku dari objek yang lebih tinggi diatasnya, yaitu objek induk (ancestor). |
| 4 | A black text with black lines  AI-generated content may be incorrect. | Include | Menspesifikasikan bahwa use case asal ditujukkan dengan jelas. |
| 5 | A black text on a white background  AI-generated content may be incorrect. | Extend | Menspesifikasikan bahwa use case tujuan memperluas perilaku dari use case asal pada suatu moment tertentu. |
| 6 |  | Association | Menghubungkan satu objek dengan objek lainnya. |
| 7 | A black square with a white background  AI-generated content may be incorrect. | System | Menspesifikasikan paket yang menampilkan sistem dengan batasan tertentu. |
| 8 | A black text on a white background  AI-generated content may be incorrect. | Use Case | Keterangan mengenai urutan tindakan yang ditujukan oleh sistem yang menghasilkan hasil yang terukur bagi seorang aktor. |
| 9 | A black circle with a white background  AI-generated content may be incorrect. | Collaboration | Interaksi antara aturan-aturan dan element lainnya yang bekerja sama untuk memberikan efek yang lebih besar daripada sekadar jumlah element-elementnya. |
| 10 | A black and white rectangular frame  AI-generated content may be incorrect. | Note | Unsur fisik yang ada saat aplikasi dioperasikan dan memerlukan suatu sumber daya komputer. |

*Sumber : Rasiban, dkk. (2024)*

1. *Activity Diagram*

Menurut Rasiban et al. (2024), *Activity Digram* adalah sejenis diagram dalam *Unified Modeling Language (UML)* yang berfungsi untuk menunjukkan alur kerja atau aktivitas dalam sebuah sistem atau proses. Berikut adalah simbol-simbol dari *activity diagram* yaitu sebagai berikut :

Tabel 2. 3 Simbol-simbol *Activity Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
| --- | --- | --- | --- |
| 1 | A black and white picture frame  AI-generated content may be incorrect. | Activity | menggambarkan cara kerja suatu bisnis, diisi dengan kata-kata kerja. Setiap activity hanya dapat memiliki satu alur masuk dan satu alur keluar. |
| 2 |  | State Transmission | Menambahkan transit antara satu activity dengan activity yang lain. |
| 3 | A black circle with white text  AI-generated content may be incorrect. | Start State | Menunjukkan di mana proses kerja dimulai. |
| 4 | A black circle with white border  AI-generated content may be incorrect. | End State | Menunjukkan di mana proses kerja berakhir. |
| 5 | A black diamond shaped object  AI-generated content may be incorrect. | Decision | Digunakan untuk menggambarkan keputusan atau tindakan yang perlu diambil dalam situasi tertentu. |
| 6 | A black arrow pointing to a rectangular object  AI-generated content may be incorrect. | Fork / Percabangan | Satu alur yang menghasilkan dua atau lebih activity yang dikerjakan secara bersamaan. |
| 7 | A black line with arrows  AI-generated content may be incorrect. | Join / Penggabungan | Beberapa alur bergabung untuk melanjutkan activity |
| 8 | A black line on a white background  AI-generated content may be incorrect. | Synchronization (Vertical / Horizontal) | Menambahkan sinkronisasi pada diagram |
| 9 | A black and white rectangular frame  AI-generated content may be incorrect. | Swimlane | Sebuah metode untuk mengelompokkan activity berdasarkan actor. Nama actor dapat ditulis di dalamnya. |

Sumber: Siking, dkk., 2023

1. *Class Diagram*

Menurut Rasiban et al. (2024), *Class Diagram* adalah suatu jenis diagram dalam *Unified Modeling Language (UML)* yang berfungsi untuk menunjukkan struktur tetap dari sebuah sistem atau aplikasi yang berbasis objek. Berikut adalah simbol- simbol dari *class diagram* yaitu sebagai berikut :

*Tabel 2. 6* Simbol-simbol *Class Diagram*

| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| --- | --- | --- | --- |
| 1 | A close-up of a label  AI-generated content may be incorrect. | Class | Kelas pada struktur sistem |
| 2 | A black circle with white background  AI-generated content may be incorrect. | Interface | Sama seperti konsep antarmuka dalam pemrograman berorientasi objek |
| 3 |  | Association | Hubungan antar kelas dengan pengertian umum, asosiasi seringkali mencakup multiplicity |
| 4 | A black line on a white background  AI-generated content may be incorrect. | Generalization | Hubungan antar kelas dengan pengertian generalisasi-spesialisasi (umum-khusus) |
| 5 |  | Aggregation | Hubungan antar kelas dengan pengertian semua bagian (whole-part) |
| 6 | A black line on a white background  AI-generated content may be incorrect. | Dependency | Hubungan antar kelas dengan pengertian ketergantungan antar kelas |
| 7 | A black and white line  AI-generated content may be incorrect. | Composition | Jika sebuah kelas tidak bisa berdiri sendiri dan harus merupakan bagian dari kelas yang lain, maka kelas tersebut memiliki relasi composition terhadap kelas tempat bergantung tersebut |
| 8 | A black line on a white background  AI-generated content may be incorrect. | Directed Association | Hubungan antar kelas dengan pengertian kelas yang satu digunakan oleh kelas lain, asosiasi biasanya disertai dengan multiplicity |

# BAB 3

## METODE PENELITIAN

### 3.1 Alat dan Bahan Pembuatan Rancangan

Dalam proses pembuatan rancangan ini, penulis menggunakan beberapa alat dan bahan sebagai berikut :

#### 3.1.1 Alat

Berikut ini merupakan alat-alat yang di pergunakan dalam proses pembuatan rancangan yaitu:

1. Perangkat Keras (Hardware)

Perangkat Keras (Hardware) yang digunakan untuk membangun dan

mengumpulkan data pada sistem ini adalah Laptop Acer Nitro V15 dengan

spesifikasi processor AMD Ryzen™ 5 6600H with Radeon™ Graphics @ 3.30 GHz, kartu grafis NVIDIA GeForce RTX 3050, serta menggunakan sistem operasi Windows 11 Home Single Language.

2. Perangkat Lunak (Software)

Perangkat lunak yang digunakan dalam membangun dan merancang sistem ini

berupa sistem operasi sebagai berikut :

1) Laragon : sebagai web server lokal untuk menjalankan PHP dan MysSQL

2) Visual Studio Code : sebagai editor kode program

3) PHP MyAdmin : sebagai tempat penyimpanan database data sistem

4) TensorFlow : sebagai tempat untuk melatih model

5) Gogle Crome : untuk uji coba dan preview website

### 3.2 Metode Pengumpulan

Data Data yang diperoleh dalam penelitian ini diperoleh melalui berbagai cara pengumpulan data yang dipilih untuk mendukung dalam membangun dan merancang sistem. Tujuan dari pengumpulan data ini adalah untuk mendapatkan informasi yang tepat dan sesuai tentang kondisi aktual di lapangan, kebutuhan pengguna, serta referensi ilmiah yang menjadi dasar pengembangan sistem. Metode yang diterapkan adalah sebagai berikut :

#### 3.2.1 Wawancara

Dalam tahap analisis kebutuhan sistem, penulis melakukan diskusi dan wawancara mendalam dengan klien, yakni Kakak Amelia Salwa Fransisca selaku Karyawan di Dinas Perdagangan dan Perindustrian Kabupaten Bandung. Wawancara ini bertujuan untuk memetakan alur bisnis (*business flow*) terkait pencatatan transaksi keuangan serta mengidentifikasi *pain points* (kendala utama) yang dialami dalam proses penentuan jenis pajak.

Berdasarkan hasil diskusi tersebut, klien memaparkan bahwa mekanisme penentuan klasifikasi pajak (PPN dan PPh) saat ini masih dilakukan secara manual dengan menganalisis satu per satu bukti transaksi. Hal ini mengakibatkan inefisiensi waktu dan tingginya risiko *human error* (kesalahan penentuan tarif atau jenis pajak). Selain itu, rekapitulasi laporan pajak yang masih bertumpu pada pencatatan manual di Microsoft Excel dinilai kurang efektif dan rawan ketidaksesuaian data. Oleh karena itu, klien membutuhkan solusi berupa sistem informasi terintegrasi yang menerapkan algoritma *Decision Tree* untuk mengotomatisasi klasifikasi pajak secara akurat dan mempercepat proses pelaporan.

### 3.3 Metode Pengembangan

Metode pengembangan sistem berfungsi sebagai pedoman dalam membangun sistem informasi dengan cara yang terencana dan teratur. Dalam tugas akhir ini, pendekatan yang diterapkan adalah metode Waterfall.

Metode Waterfall dipilih untuk pengembangan Sistem Informasi Klasifikasi Pajak Digital karena pendekatannya yang sistematis dan terstruktur. Dalam pengembangan sistem ini, kebutuhan sistem telah diidentifikasi secara menyeluruh sejak awal, meliputi prosedur input data transaksi keuangan, proses klasifikasi otomatis jenis pajak (PPN dan PPh) menggunakan algoritma Decision Tree, serta rekapitulasi pelaporan pajak. Metode ini sangat cocok karena setiap langkahnya dilakukan secara berurutan, dimulai dari analisis kebutuhan, desain sistem, pengembangan (coding), pengujian akurasi klasifikasi, hingga perawatan (maintenance).

Dengan menerapkan metode Waterfall, pengembangan sistem dapat dilaksanakan secara sistematis, memastikan bahwa setiap tahapan diselesaikan dengan baik sebelum melanjutkan ke tahap berikutnya. Pendekatan ini berkontribusi signifikan pada pengurangan risiko terjadinya kesalahan dalam penentuan jenis pajak dan memastikan bahwa setiap elemen sistem, terutama integrasi antara PHP dan Python, dibuat dengan teliti. Alur yang teratur mempermudah proses pengelolaan proyek, sehingga hasil akhirnya diharapkan dapat memenuhi kebutuhan pengguna di Dinas Perdagangan dan Perindustrian serta berjalan sesuai rencana.

### 3.4 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi kebutuhan pengguna,

masalah yang muncul dalam sistem manual yang ada saat ini, serta menetapkan

solusi yang sesuai melalui perancangan dan pembangunan sistem berbasis website.

Berikut adalah analisis kebutuhan sistem yang dilakukan yaitu:

#### 3.4.1. Analisis Sistem Berjalan

Dinas Perdagangan dan Perindustrian Kabupaten Bandung merupakan perangkat daerah yang berperan penting dalam mengatur, mencatat, serta menyampaikan laporan terkait aktivitas perdagangan, termasuk aspek perpajakan. Dalam lingkup perpajakan, dinas ini menangani transaksi yang berhubungan dengan Pajak Pertambahan Nilai (PPN) dan Pajak Penghasilan (PPh).

Saat ini, sistem yang digunakan masih bersifat semi-manual. Pencatatan transaksi dilakukan melalui aplikasi spreadsheet seperti Microsoft Excel serta dokumen cetak yang kemudian diarsipkan secara fisik. Proses pengelompokan transaksi pajak masih dikerjakan secara manual oleh staf Subbagian Keuangan, dengan cara membaca dokumen, menentukan jenis pajak (PPN atau PPh), lalu memasukkannya ke dalam laporan bulanan.

Meski sudah ada sistem informasi berbasis komputer, penggunaannya masih terbatas pada pencatatan transaksi tanpa dukungan algoritma pintar yang mampu melakukan klasifikasi otomatis. Kondisi ini menimbulkan risiko kesalahan input, keterlambatan dalam penyusunan laporan, serta kurang efisiennya proses analisis data pajak.

Tahapan pelaporan pajak dilakukan melalui beberapa langkah:

1. Pencatatan Transaksi – Setiap transaksi yang mengandung unsur PPN atau PPh dicatat secara manual. Petugas menentukan jenis pajak yang berlaku sesuai regulasi.
2. Rekapitulasi Data Pajak – Seluruh data transaksi dirangkum dalam laporan bulanan menggunakan filter manual. Tahap ini rawan human error, seperti kesalahan klasifikasi, perhitungan, maupun keterlambatan input.
3. Pelaporan ke Instansi Terkait – Laporan yang telah selesai kemudian diserahkan ke Subbagian Keuangan untuk diteruskan ke Kantor Pajak sesuai jadwal.

Adapun kelemahan utama dari sistem yang berjalan saat ini meliputi:

1. Tingginya potensi kesalahan dalam klasifikasi transaksi (PPN atau PPh).
2. Proses pengolahan data memakan waktu lama karena belum terotomatisasi.
3. Tidak tersedianya fitur analisis dan prediksi berbasis algoritma untuk mempercepat klasifikasi.
4. Kesulitan dalam monitoring dan audit akibat data yang tersebar antara bentuk manual dan digital tanpa adanya integrasi.

# 3.5 Langkah Metode

Pada tahap ini, dilakukan proses perhitungan algoritma Decision Tree menggunakan metode CART (Classification and Regression Tree). Algoritma ini digunakan untuk menentukan klasifikasi jenis pajak (PPN atau PPh) berdasarkan atribut transaksi. CART menggunakan Gini Index sebagai parameter untuk mengukur ketidakmurnian (*impurity*) suatu data dan menentukan atribut terbaik untuk pemisahan (*splitting*) node.

#### 3.5.1 Penentuan Variabel Data

Sebelum melakukan perhitungan, data transaksi dipetakan ke dalam variabel-variabel berikut:

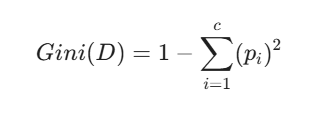
1. Atribut (Fitur):
   * Nilai Transaksi: Nominal uang dalam transaksi (Numerik).
   * Status PKP Rekanan: Apakah rekanan berstatus Pengusaha Kena Pajak atau Tidak (Kategorikal: Ya/Tidak).
   * Jenis Barang/Jasa: Kategori pengadaan (Kategorikal: Barang/Jasa).
2. Label (Target):
   * Jenis Pajak: Output klasifikasi (Kelas: PPN, PPh, atau Bebas Pajak).

#### 3.5.2 Rumus Perhitungan Gini Index

Algoritma CART memilih atribut pemecah berdasarkan nilai *Gini Impurity* terendah. Berikut adalah rumus-rumus yang digunakan dalam proses perhitungan sistem:

1. Menghitung Gini Index pada Node (Induk)

Untuk mengukur ketidakmurnian pada sebuah himpunan data $D$, digunakan rumus:

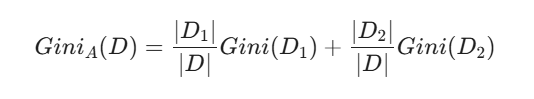


Keterangan:

* *C* : Jumlah kelas target (misalnya: PPN, PPh).
* *Pi*: Peluang probabilitas munculnya kelas $i$ dalam himpunan data *D*.

2. Menghitung Gini Index Atribut (Split)

Setelah membagi data berdasarkan atribut *A* menjadi dua himpunan bagian (*D1*dan *D2*), nilai Gini Index untuk atribut tersebut dihitung dengan rata-rata berbobot:



Keterangan:

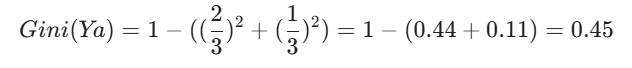
* *|D*| : Jumlah total data pada node awal.
* |*D1*| : Jumlah data pada cabang kiri (misalnya: Status PKP = Ya).
* |*D2*| : Jumlah data pada cabang kanan (misalnya: Status PKP = Tidak).

3. Menentukan Penurunan Impuritas (Delta Gini)

Sistem akan memilih atribut yang memberikan nilai Gini Index Atribut terendah, karena nilai yang lebih rendah menunjukkan klasifikasi yang lebih murni (homogen).

Langkah 2: Menghitung Gini Index Berdasarkan Atribut "Status PKP" Sistem menguji pemisahan data jika dibagi berdasarkan Status PKP (Ya vs Tidak).

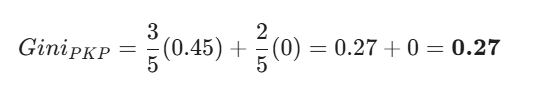
Cabang 1 (PKP = Ya): Data No. 1, 3, 4 (Total 3 data: 2 PPN, 1 PPh)



Cabang 2 (PKP = Tidak): Data No. 2, 5 (Total 2 data: 0 PPN, 2 PPh)

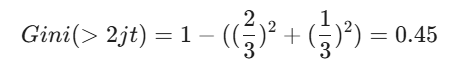


Gini Split (Status PKP):

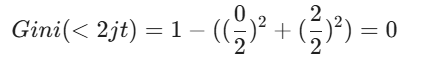


Langkah 3: Menghitung Gini Index Berdasarkan Atribut "Nilai Transaksi" Sistem menguji pemisahan jika dibagi berdasarkan Nilai > 2 Juta.

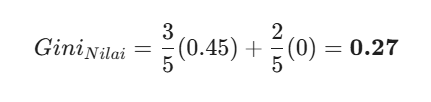
Cabang 1 (> 2 Juta): Data No. 1, 2, 4 (Total 3 data: 2 PPN, 1 PPh)



Cabang 2 (< 2 Juta): Data No. 3, 5 (Total 2 data: 0 PPN, 2 PPh)



Gini Split (Nilai Transaksi):



Langkah 4: Pengambilan Keputusan Sistem membandingkan nilai Gini Split. Jika nilai Gini sama, sistem akan menggunakan prioritas atribut (misalnya Status PKP). Node akan terus dipecah hingga nilai Gini mencapai 0 (murni) atau mencapai kedalaman maksimum pohon. Hasil perhitungan ini kemudian diterjemahkan ke dalam kode program Python/PHP untuk mengklasifikasikan data baru secara otomatis.

### 3.6 Langkah Perancangan

Perancangan sistem dilakukan setelah menyelesaikan analisis kebutuhan pengguna dan pengumpulan informasi. tujuannya adalah untuk mempermudah dalam penggambarang struktur dan alur sistem yang akan dirancang secara menyeluruh, agar proses perancangan sistem lebih terarah. Berikut adalah langkah-langkah dalam perancangan sistem ini yaitu :

#### 3.6.1 Perancangan Use Case Diagram

Untuk mendukung proses pengolahan data pajak agar lebih efektif, efisien, dan terintegrasi, diperlukan rancangan sistem baru yang mampu mengotomatisasi sebagian besar aktivitas serta meminimalisir kesalahan manual. Sistem ini diusulkan menggunakan pendekatan Use Case Diagram guna menggambarkan interaksi antara aktor dengan fungsi-fungsi utama yang tersedia pada sistem.

Dengan adanya Use Case Diagram, alur kerja setiap aktor yang terlibat—mulai dari Staff Input Data, Petugas Pajakhingga Pihak Eksternal seperti KPP atau Bapenda dapat dijelaskan secara visual. Diagram ini memperlihatkan hubungan antara aktor dengan proses, seperti klasifikasi pajak otomatis, validasi data transaksi, verifikasi laporan, pembuatan laporan pajak bulanan, hingga penyampaian laporan penetapan pajak. Rancangan ini diharapkan mampu menjadi acuan dalam pengembangan sistem baru yang lebih terstruktur, transparan, serta mendukung pengendalian internal dalam pengelolaan pajak. Untuk usulan Use Case Diagram terlihat pada gambar 3.1

A diagram of a diagram

AI-generated content may be incorrect.

*Gambar 3. 1 Use Case Diagram Sistem Klasifikasi Pajak*

1. Deskripsi *Actor*

Berikut adalah Tabel deskripsi *actor* dari Sistem Pakar Rekomendasi Jurusan yaitu :

*Tabel 3. 1Deskripsi Actor Sistem Pakar Rekomendasi Jurusan Vokasi*

| **NO** | **Nama Actor** | **Deskripsi** |
| --- | --- | --- |
| 1 | Admin | Merupakan aktor yang memegang hak akses tertinggi dalam sistem dengan otoritas penuh terhadap pengelolaan data dan konfigurasi aplikasi. Peran ini bertanggung jawab untuk melakukan validasi (persetujuan atau penolakan) terhadap setiap transaksi yang diinputkan oleh pegawai guna memastikan keakuratan perhitungan pajak. Selain itu, Admin bertugas mengelola data master (pengguna dan rekanan), melakukan impor data transaksi secara massal, serta mencetak laporan akhir rekapitulasi perpajakan sebagai dokumen pertanggungjawaban. |
| 2 | Pegawai | Merupakan aktor yang bertugas pada level operasional untuk melakukan pencatatan data transaksi harian ke dalam sistem. Peran ini memiliki hak akses terbatas yang difokuskan pada fungsi input data transaksi baru dan pengunggahan dokumen bukti pendukung (seperti scan nota atau faktur). Pegawai juga diberikan akses ke fitur monitoring untuk memantau riwayat transaksi yang telah dimasukkan, namun tidak memiliki wewenang untuk mengubah pengaturan sistem maupun memvalidasi data. |

1. Deskripsi usecase

Berikut adalah Tabel deskripsi usecase yaitu sebagai berikut:

*Tabel 3. 2 Deskripsi usecase Sistem Klasifikasi Pajak*

| **NO** | **Nama Use Case** | **Deskripsi** |
| --- | --- | --- |
| **A** | **Use CaseUmum & Admin** |  |
| 1 | Login | Proses autentikasi awal sebelum masuk ke dalam sistem. Digunakan oleh Admin untuk mengakses halaman dashboard awal. |
| 2 | Logout | Proses keluar dari sistem untuk mengakhiri sesi penggunaan dan menjaga keamanan akun. |
| 3 | Input Data Transaksi | Input transaksi fitur utama bagi admin untuk mencatat data transaksi harian satu per satu. Pengguna mengisi formulir digital yang mencakup tanggal, uraian belanja, memilih rekanan, serta memasukkan nilai nominal transaksi untuk dihitung pajaknya. |
| 4 | Lihat Data Transaksi | Admin Melihat daftar transaksi yang sudah tersimpan ke dalam database dan admin bisa melakukan (CRUD) |
| 5 | Monitoring Pajak | Admin dapat melihat Monitroring pajak yang sudah di input melalui input data transaksi manual atau import otomatis dengan menampilkan total setoran pajak, total ppn , total pph21, dan total pph23, lalu dapat memfilter halaman per tahun transaksi. |
| 6 | Import Excel ( Berbasis Ai) | Admin dapat melakukan impor data transaksi perpajakan secara massal menggunakan format file Excel, yang memungkinkan sistem membaca dan mengkalkulasi pajak secara otomatis tanpa input manual satu per satu. |
| 7 | Cetak Laporan | Admin dapat mencetak laporan akhir rekapitulasi perpajakan berdasarkan periode waktu tertentu (bulan/tahun), yang hanya menampilkan data transaksi yang telah berstatus valid. |
| 8 | Kelola Rekanan | Admin dapat mengelola (CRUD) data master Rekanan (Vendor/Perusahaan), termasuk menambah mitra baru, memperbarui informasi NPWP, serta menghapus rekanan yang sudah tidak aktif. |
| 9 | Manajemen Pengguna | Admin dapat mengelola akun pengguna sistem (Manajemen Pengguna), mengatur siapa saja pegawai yang diberikan hak akses, serta melakukan reset *password* jika diperlukan. |
| 10 | Pengaturan Sistem | Admin dapat mengatur konfigurasi global sistem pada menu Pengaturan, seperti mengubah nama instansi, logo aplikasi, dan parameter dasar lainnya. |
| **B** | **Use Case User Pegawai** |  |
| 11 | Login | Proses autentikasi awal sebelum masuk ke dalam sistem. Digunakan oleh Admin untuk mengakses halaman dashboard awal |
| 12 | Logout | Proses keluar dari sistem untuk mengakhiri sesi penggunaan dan menjaga keamanan akun. |
| 13 | Input Data transaksi | melakukan pencatatan transaksi pengeluaran harian melalui formulir input data, yang mencakup pengisian tanggal, uraian belanja, pemilihan rekanan, dan nilai nominal. |
| 14 | Data Transaksi | Pegawai dapat memantau riwayat data transaksi yang telah diinputkan sebelumnya dan melihat status terkini dari transaksi tersebut (apakah masih *Pending*, sudah *Valid*, atau *Ditolak* oleh Admin). |
| 15 | Monitoring Pajak | Pegawai dapat melihat *dashboard* monitoring pajak untuk mengetahui ringkasan angka realisasi pajak secara *real-time*, namun hanya sebatas hak akses "lihat" (*read-only*). |

#### 3.6.2 Perancangan *Activity Diagram*

*Activity Diagram* menggambarkan langkah-langkah dari setiap proses penting dalam sistem Klasifikasi Pajak, seperti proses input data Transaksi, Import Data melalui Excel, Penerapan Algoritma Decision Tree, hingga memonitoring pajak. Activity Diagram mempermudah pemahaman tentang urutan tindakan (flow) dan keputusan logika yang terjadi di dalam sistem antara pengguna dan komputer. Berikut merupakan rancangan activity diagram pada sistem yang dibangun:

* 1. Login Admin

A diagram of a system

AI-generated content may be incorrect.

*Gambar 3. 2 Activity Diagram Login Admin*

*Tabel 3. 3 Diagram Login Admin*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Login |
| Actor | Admin |
| Tujuan | Untuk melakukan proses login dan mengakses dashboard admin. |
| Deskripsi | Berikut adalah aktivitas login admin, saat admin membuka halaman website, Sistem akan menampilkan halaman login. Selanjutnya admin memasukkan email dan password. Sistem akan memverifikasi data yang diinput dengan data di database. Jika data sesuai, maka proses login berhasil dan akan diarahkan ke halaman dashboard. Jika tidak sesuai, maka sistem akan kembali ke halaman login dan menampilkan pesan *error*.  format PDF. |
| Kondisi Awal | Halaman Login |
| Kondisi Akhir | Halaman Dasboard (jika login berhasil) / Halaman login kembali (jika gagal) |



1. Login Pegawai

A diagram of a system

AI-generated content may be incorrect.

*Gambar 3. 3 Activity Diagram Login Pegawai*

*Tabel 3. 4 Activity Diagram Login pegawai*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Login |
| Actor | Pegawai |
| Tujuan | Untuk melakukan proses login dan mengakses dashboard pegawai. |
| Deskripsi | Berikut adalah aktivitas login pegawai, saat pegawai membuka halaman website, Sistem akan menampilkan halaman login. Selanjutnya pegawai memasukkan email dan password. Sistem akan memverifikasi data yang diinput dengan data di database. Jika data sesuai, maka proses login berhasil dan akan diarahkan ke halaman dashboard. Jika tidak sesuai, maka sistem akan kembali ke halaman login dan menampilkan pesan *error*.  format PDF. |
| Kondisi Awal | Halaman Login |
| Kondisi Akhir | Halaman Dasboard (jika login berhasil) / Halaman login kembali (jika gagal) |

1. Input Transaksi (Role: Pegawai)

A diagram of a data flow

AI-generated content may be incorrect.

*Gambar 3. 4 Activity Diagram Input Transaksi Manual Pegawai*

*Tabel 3. 5 Activity Diagram Input Transaksi Manual Pegawai*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Input Transaksi Manual |
| Actor | Pegawai |
| Tujuan | Untuk melakukan proses input data transaksi |
| Deskripsi | Berikut adalah aktivitas input transaksi manual oleh Pegawai. Saat Pegawai memilih menu "Input Transaksi", sistem akan menampilkan formulir isian data. Selanjutnya, Pegawai mengisi tanggal transaksi, uraian belanja, memilih nama rekanan, memasukkan nilai nominal, serta mengunggah bukti transaksi (nota/faktur). Setelah Pegawai menekan tombol simpan, sistem akan memvalidasi kelengkapan data. Jika data lengkap, sistem akan menyimpan data ke *database* dengan status *Pending* dan menampilkan pesan sukses. Jika ada data yang kosong, sistem akan menampilkan pesan peringatan agar Pegawai melengkapinya. |
| Kondisi Awal | Halaman Input Transaksi |
| Kondisi Akhir | Halaman Input Transaksi (jika login berhasil) / Halaman login kembali (jika gagal) |

1. Input Transaksi Manual (Role Admin)

A diagram of a flowchart

AI-generated content may be incorrect.

*Gambar 3. 5 Activity Diagram Input Transaksi Manual Admin*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Input Transaksi Manual |
| Actor | Admin |
| Tujuan | Untuk melakukan proses input data transaksi |
| Deskripsi | Berikut adalah aktivitas input transaksi manual oleh Admin. Saat Admin memilih menu "Input Transaksi", sistem akan menampilkan formulir isian data. Selanjutnya, Admin mengisi tanggal transaksi, uraian belanja, memilih nama rekanan, memasukkan nilai nominal, serta mengunggah bukti transaksi (nota/faktur). Setelah Admin menekan tombol simpan, sistem akan memvalidasi kelengkapan data. Jika data lengkap, sistem akan menyimpan data ke *database* dengan status *Pending* dan menampilkan pesan sukses. Jika ada data yang kosong, sistem akan menampilkan pesan peringatan agar Admin melengkapinya. |
| Kondisi Awal | Halaman Input Transaksi |
| Kondisi Akhir | Halaman Input Transaksi (jika login berhasil) / Halaman login kembali (jika gagal) |

*Tabel 3. 6 Activity Diagram Input Transaksi Manual Admin*

1. Import Data Excel (Aktor: Admin)

A diagram of a process

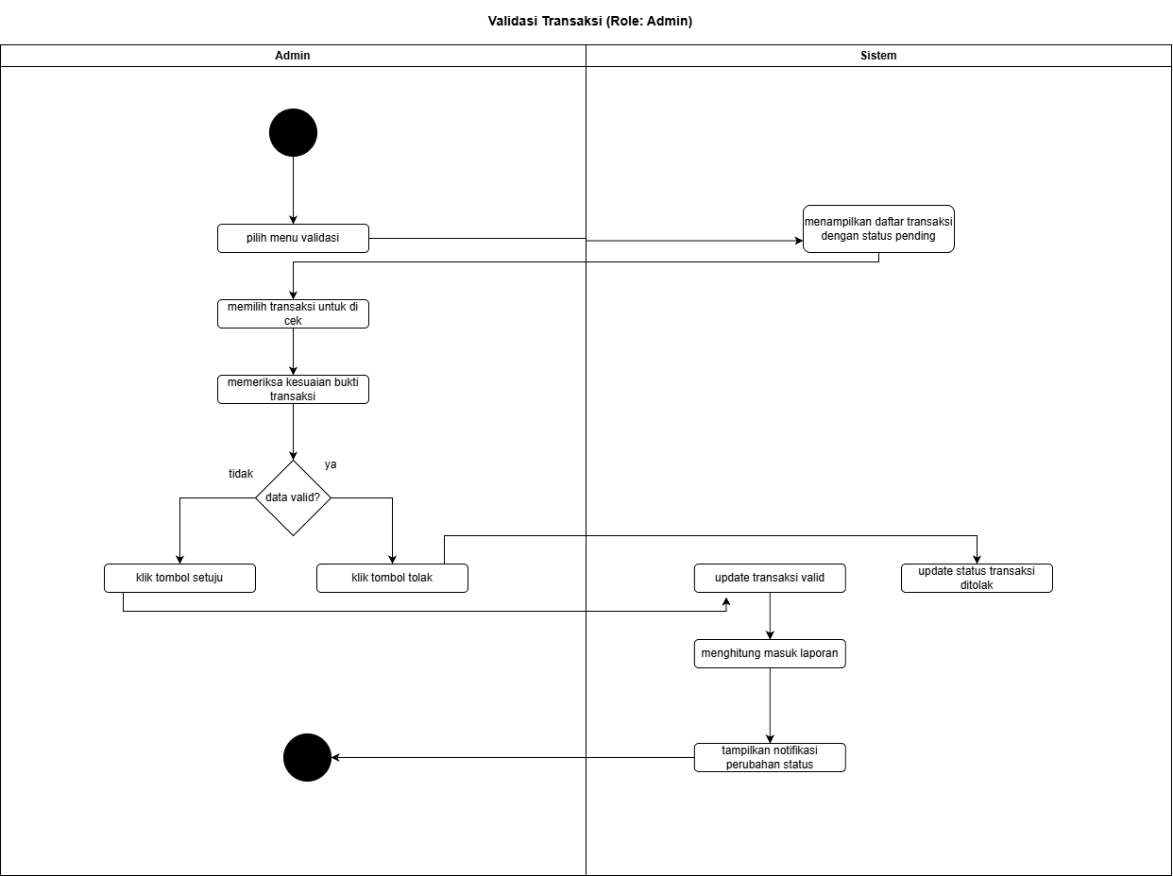
AI-generated content may be incorrect.

*Gambar 3.7 Gambar 3. 6 Activity Diagram Input Transaksi Excel(otomatis Ai)*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Input Transaksi Excel |
| Actor | Admin |
| Tujuan | Untuk melakukan proses input data transaksi secara otomatis |
| Deskripsi | Berikut adalah aktivitas impor data massal oleh Admin. Saat Admin membuka halaman "Impor Excel", sistem menyediakan opsi untuk mengunduh templat atau mengunggah file. Admin memilih file berekstensi .xlsx dari komputer lalu menekan tombol "Mulai Proses Impor". Sistem akan membaca file tersebut, memvalidasi format kolom, dan memproses setiap baris data. Sistem secara otomatis mendeteksi jenis pajak berdasarkan nilai yang ada dan menyimpannya ke *database*. Setelah proses selesai, sistem menampilkan ringkasan jumlah data yang berhasil diimpor dan jumlah data yang gagal/duplikat. |
| Kondisi Awal | Halaman Input Transaksi Excel |
| Kondisi Akhir | Halaman Input Transaksi Excel (jika login berhasil) / Halaman login kembali (jika gagal) |

*Tabel 3. 7 Activity Diagram Input Transaksi Excel(otomatis Ai)*

1. Validasi Transaksi (Role: Admin)



*Gambar 3. 7 Activity Diagram Validasi Transaksi*

*Tabel 3. 8 Activity Diagram Validasi Transaksi*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Halaman Validasi Transaksi |
| Actor | Admin |
| Tujuan | Untuk melakukan proses input data transaksi secara otomatis |
| Deskripsi | Berikut adalah aktivitas validasi transaksi oleh Admin. Saat Admin mengakses menu "Validasi Transaksi", sistem menampilkan daftar transaksi yang berstatus *Pending* (belum diperiksa). Admin memilih salah satu transaksi untuk melihat rincian dan bukti pendukung. Jika data dianggap benar dan sesuai, Admin menekan tombol "Setuju", dan sistem akan mengubah status transaksi menjadi *Valid* serta memasukkannya ke dalam perhitungan laporan. Sebaliknya, jika terdapat kesalahan, Admin menekan tombol "Tolak", dan sistem mengubah status menjadi *Ditolak*. |
| Kondisi Awal | Halaman Validasi Transaksi |
| Kondisi Akhir | Halaman Validasi Transaksi (jika login berhasil) / Halaman login kembali (jika gagal) |

1. Cetak Laporan

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Halaman Cetak Laporan |
| Actor | Admin |
| Tujuan | Untuk melakukan proses cetak laporan |
| Deskripsi | Berikut adalah aktivitas pencetakan laporan oleh Admin. Saat Admin membuka menu "Cetak Laporan", sistem menampilkan formulir filter. Admin memilih kriteria laporan seperti rentang tanggal, nama rekanan, atau jenis pajak tertentu, lalu menekan tombol "Tampilkan/Cetak". Sistem akan melakukan *query* ke *database* untuk mengambil data transaksi yang berstatus *Valid* sesuai filter. Selanjutnya, sistem men-generate tampilan laporan resmi lengkap dengan kop surat dan tabel rekapitulasi dalam format siap cetak (Print Preview/PDF). |
| Kondisi Awal | Halaman Cetak laporan |
| Kondisi Akhir | Halaman Cetak Laporan (jika login berhasil) / Halaman login kembali (jika gagal) |

A diagram of a flowchart

AI-generated content may be incorrect.

*Gambar 3. 8 Activity Diagram Cetak Laporan*

*Tabel 3. 9 Activity Diagram Cetak Laporan*

1. Kelola Rekanan / CRUD (Role: Admin)

A diagram with black circles and black dots

AI-generated content may be incorrect.

*Gambar 3. 9 Activity Diagram Kelola Rekaan*

*Tabel 3. 10 Activity Diagram Kelola Rekaan*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Halaman Kelola Rekaan |
| Actor | Admin |
| Tujuan | Untuk melakukan proses input rekaan |
| Deskripsi | Berikut adalah aktivitas pengelolaan data rekanan oleh Admin. Saat Admin masuk ke menu "Rekanan", sistem menampilkan tabel daftar mitra kerja. Jika Admin ingin menambah data, Admin menekan tombol tambah, mengisi nama perusahaan, NPWP, status PKP, dan kontak, lalu menyimpannya. Sistem akan memproses penyimpanan data baru tersebut. Admin juga dapat memilih tombol edit untuk memperbarui informasi rekanan atau tombol hapus untuk menghapus data rekanan yang sudah tidak aktif dari sistem. |
| Kondisi Awal | Halaman Kelola Rekaan |
| Kondisi Akhir | Halaman Kelola Rekaan (jika login berhasil) / Halaman login kembali (jika gagal) |

1. Manajemen Pengguna (Role: Admin)

A diagram of a company

AI-generated content may be incorrect.

*Gambar 3. 10 Activity Diagram Kelola Manajemen Pengguna*

*Tabel 3. 11 Activity Diagram Kelola Manajemen Pengguna*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Halaman Manajemen pengguna |
| Actor | Admin |
| Tujuan | Untuk melakukan proses input data pengguna |
| Deskripsi | Berikut adalah aktivitas manajemen pengguna oleh Admin. Saat Admin membuka menu "Manajemen Pengguna", sistem menampilkan daftar akun yang terdaftar. Admin menekan tombol tambah pengguna, lalu mengisi nama lengkap, *username*, *password*, dan memilih hak akses (*role*) sebagai Admin atau Pegawai. Setelah tombol simpan ditekan, sistem akan memvalidasi ketersediaan *username* dan melakukan enkripsi pada *password*. Jika valid, akun baru akan tersimpan di *database* dan dapat segera digunakan untuk *login*.. |
| Kondisi Awal | Halaman Manajemen pengguna |
| Kondisi Akhir | Halaman Manajemen Pengguna(jika login berhasil) / Halaman login kembali (jika gagal) |

1. Kelola Halaman Pengaturan Sistem

A diagram of a diagram

AI-generated content may be incorrect.

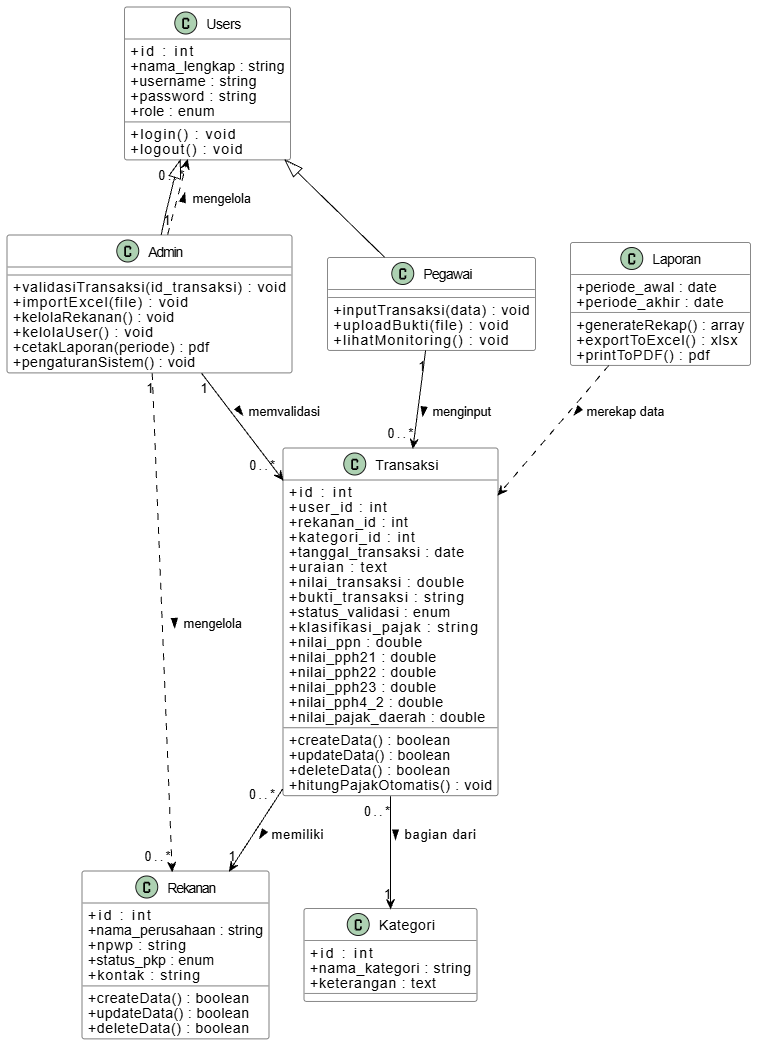
*Gambar 3. 11 Activity Diagram Kelola Pengaturan system*

*Tabel 3. 12 Activity Diagram Kelola Pengaturan system*

|  | **Identifikasi** |
| --- | --- |
| Nama Activity Diagram | Halaman Kelola pengaturan sistem |
| Actor | Admin |
| Tujuan | Untuk melakukan perubahan pada system mulai dari nama system dan logo |
| Deskripsi | Berikut adalah aktivitas konfigurasi sistem oleh Admin. Saat Admin mengakses menu "Pengaturan", sistem menampilkan formulir identitas aplikasi. Admin dapat mengubah informasi seperti nama instansi, alamat kantor, serta mengunggah logo terbaru. Setelah Admin menekan tombol simpan perubahan, sistem akan memvalidasi file gambar (logo) dan memperbarui data konfigurasi di *database*. Perubahan tersebut akan langsung diterapkan secara *real-time* pada bagian *header* aplikasi dan kop surat laporan. |
| Kondisi Awal | Halaman kelola pengaturan sistem |
| Kondisi Akhir | Halaman pengaturan sistem (jika login berhasil) / Halaman login kembali (jika gagal) |

#### 3.6.1 Perancangan Class Diagram

*Class diagram* bertujuan untuk menggambarkan struktur kelas dan hubungan antar kelas yang ada dalam sistem. *Class diagram* berfungsi untuk memodelkan entitas- entitas utama yang digunakan dalam sistem beserta atribut dan metode yang dimilikinya. *Class Diagram* juga menunjukkan bagaimana setiap entitas saling berinteraksi dan membentuk alur proses kerja yang terstruktur. Dengan adanya class diagram ini, pengembangan sistem menjadi lebih terarah karena setiap bagian telah diidentifikasi secara jelas. Berikut adalah *class diagram* dari sistem Klasisfikasi Pajak:



*Gambar 3. 12 Class Diagram Sistem Klasifikasi Pajak*

Class diagram di atas menggambarkan struktur statis perangkat lunak Sistem Manajemen Pajak yang dibangun dengan pendekatan berorientasi objek, di mana kelas *Users* bertindak sebagai *superclass* yang menampung atribut autentikasi dasar dan mewariskannya kepada dua kelas turunan (*subclass*) utama, yaitu *Admin* dan *Pegawai*. Kelas *Pegawai* dirancang dengan metode operasional untuk melakukan instansiasi pada kelas *Transaksi* melalui pencatatan data belanja dan pengunggahan bukti dukung, sementara kelas *Admin* memiliki metode manajerial yang lebih luas mencakup validasi status pada objek *Transaksi*, pengelolaan penuh (*CRUD*) terhadap data master pada kelas *Rekanan* dan *Users*, serta pengaturan konfigurasi sistem. Inti dari sistem ini terletak pada kelas *Transaksi* yang menyimpan atribut detail kalkulasi nilai perpajakan secara komprehensif dan memiliki relasi asosiasi dengan kelas *Rekanan* sebagai identitas vendor serta kelas *Kategori* sebagai pengelompokan jenis pengeluaran, yang pada akhirnya seluruh himpunan data transaksi yang telah tervalidasi akan diolah oleh kelas *Laporan* untuk menghasilkan luaran berupa dokumen rekapitulasi pajak maupun arsip digital yang akurat.

3.7.1 Perancangan User Interface

Perancangan User Interface merupakan proses merancang tampilan serta interaksi antara pengguna dan sistem, sehingga pengguna dapat menggunakan sistem dengan sederhana, efektif, dan nyaman. Berikut merupakan perancangan user interface untuk Sistem Klasifikasi Pajak :

1. Perancangan Halaman Login Admin

Gambar Mockup login merupakan pintu utama untuk mengakses sistem. Pada tahap ini pengguna diharuskan memasukkan username dan password yang telah terdaftar. Sistem kemudian melakukan validasi data ke dalam database. Apabila data yang dimasukkan sesuai, maka pengguna akan diarahkan ke halaman Dashboard utama. Namun jika data tidak valid, sistem akan menampilkan pesan kesalahan “Username atau Password salah”. Fitur ini juga berfungsi untuk menjaga keamanan data laporan keuangan, agar hanya pengguna yang berwenang (admin) yang dapat mengakses dan mengelola sistem*.*

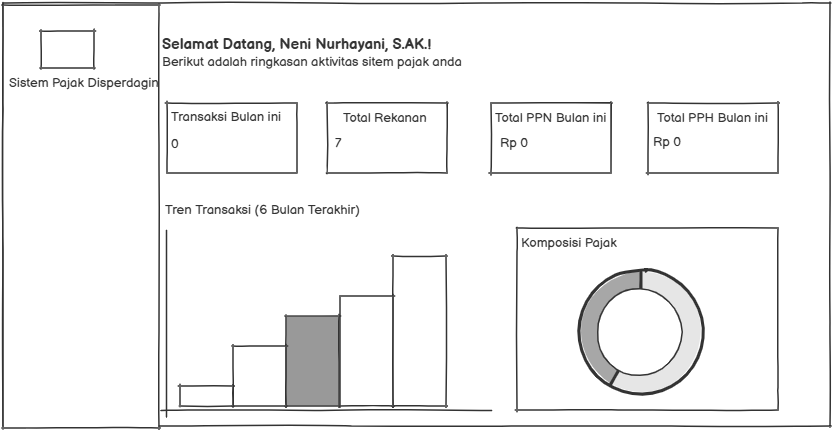
A login form with a box and a login screen

AI-generated content may be incorrect.

Gambar 3. 13 Halaman Login Admin

1. Perancangan Halaman Dashboard

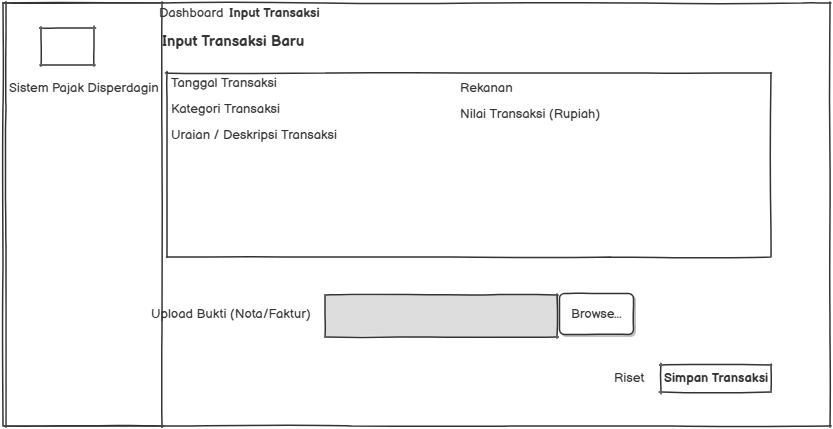
Gambar Mockup *Dashboard* berfungsi sebagai halaman utama sistem informasi laporan keuangan yang menampilkan ringkasan data penting terkait kegiatan dan penggunaan anggaran. Informasi yang disajikan mencakup total kegiatan, total anggaran, total pengeluaran, serta rata-rata serapan dana. Data tersebut divisualisasikan melalui grafik untuk mempermudah analisis cepat terhadap kondisi keuangan dinas. Selain itu, *dashboard* dilengkapi dengan menu navigasi utama untuk mengakses halaman laporan, detail perhitungan, dan Diagram. Fitur ini membantu admin dalam memantau kondisi keuangan secara menyeluruh, sekaligus menjadi titik awal interaksi pengguna setelah *login*.



Gambar 3. 14 Perancangan Halaman Dashboard

1. Perancangan Halaman Input Transaksi

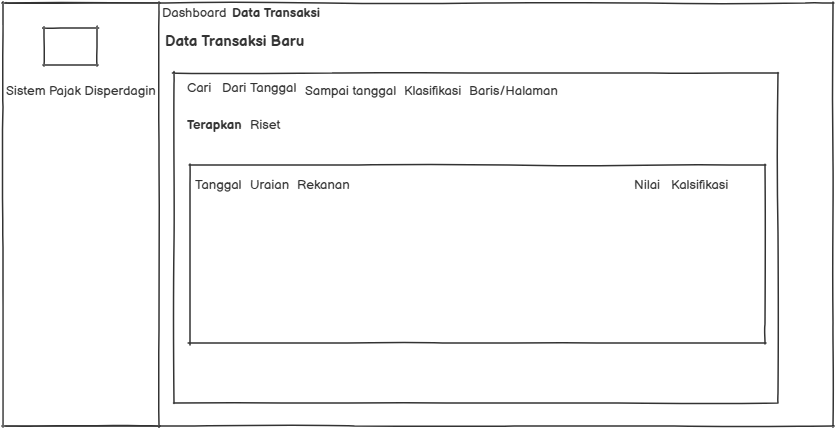
Fitur Input Transaksi dirancang untuk mempermudah admin dalam mencatat setiap transaksi pajak yang terjadi. Pada halaman ini, pengguna dapat mengisi data transaksi secara lengkap, mencakup tanggal transaksi, kategori transaksi, nama rekanan, nilai transaksi dalam rupiah, serta deskripsi atau keterangan transaksi. Selain itu, sistem juga menyediakan kolom untuk unggah bukti transaksi, seperti nota atau faktur, yang membantu mempermudah proses verifikasi di masa mendatang.



Gambar 3. 15 Perancangan Halaman Input Transaksi

1. Perancangan Halaman Data Transaksi

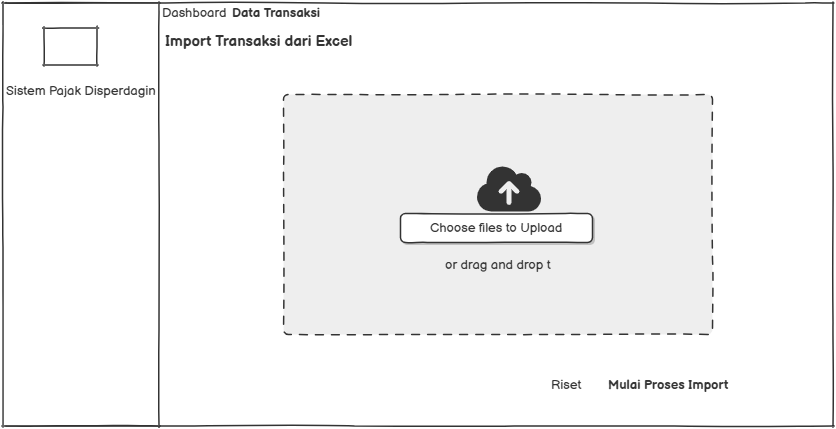
Fitur Data Transaksi berfungsi untuk menampilkan serta mengelola seluruh catatan transaksi yang telah dimasukkan ke dalam sistem. Pada halaman ini, pengguna dapat mengakses daftar transaksi secara menyeluruh, lengkap dengan informasi penting seperti tanggal, uraian, rekanan, nilai transaksi, serta hasil klasifikasi pajak yang dihasilkan sistem. Untuk memudahkan proses pencarian dan penyaringan data, sistem dilengkapi dengan fitur filter yang memungkinkan pengguna menelusuri transaksi berdasarkan kata kunci, rentang tanggal, kategori pajak, maupun jumlah data yang ingin ditampilkan pada setiap halaman.



Gambar 3. 16 Perancangan Halaman Transaksi

1. Perancangan Halaman input transaksi excel

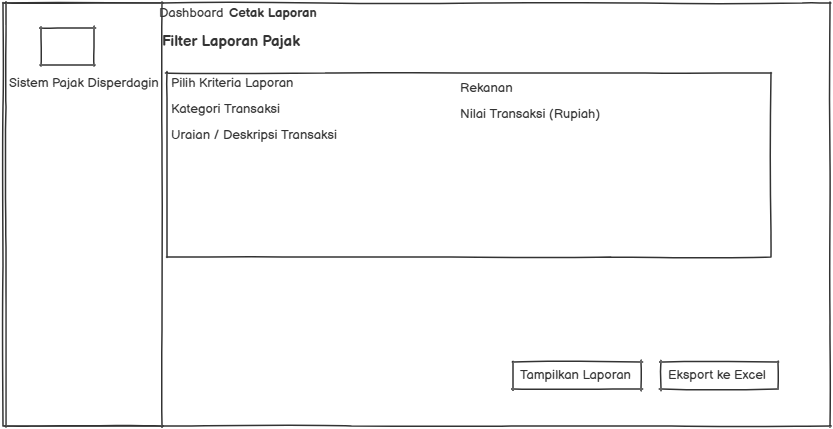
Fitur Impor Transaksi dari Excel berfungsi untuk memudahkan admin dalam melakukan input data transaksi dalam jumlah besar secara cepat dan efisien. Melalui halaman ini, pengguna dapat mengunggah file berformat .xlsx yang berisi data transaksi dengan struktur kolom tertentu, seperti tanggal, uraian transaksi, nama rekanan, serta nilai transaksi. Setelah file berhasil diunggah, sistem akan secara otomatis membaca, memvalidasi, dan memproses setiap data untuk kemudian disimpan ke dalam basis data. Dengan adanya fitur ini, proses pencatatan transaksi menjadi lebih efisien, praktis, dan akurat dibandingkan dengan metode input manual satu per satu.



Gambar 3. 17 Halaman input transaksi excel

1. Perancangan Halaman cetak laporan

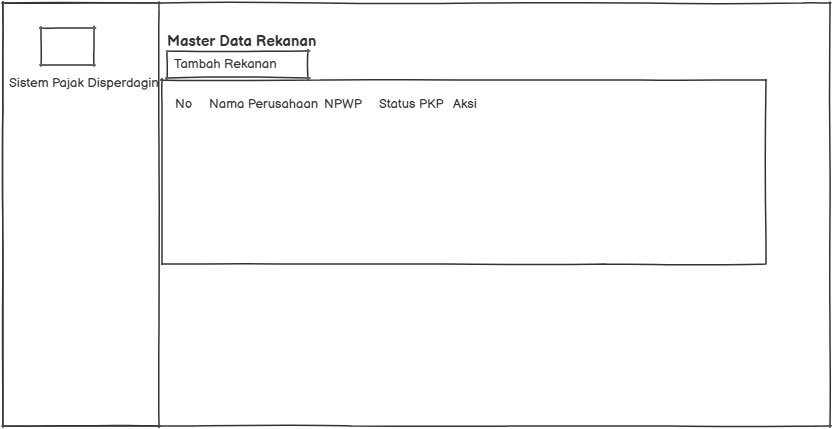
Fitur Cetak Laporan berfungsi untuk menghasilkan laporan pajak berdasarkan data transaksi yang telah tersimpan di dalam sistem. Pada halaman ini, admin dapat menentukan berbagai kriteria laporan sesuai kebutuhan, seperti rentang tanggal, nama rekanan, hingga jenis pajak tertentu. Melalui penggunaan filter tersebut, laporan yang dihasilkan menjadi lebih terarah, spesifik, dan relevan dengan tujuan pengguna. Selain dapat ditampilkan langsung pada layar, sistem juga menyediakan fitur ekspor laporan ke dalam format Excel (.xlsx). Fasilitas ini memudahkan pengguna dalam melakukan pengolahan data lanjutan maupun penyimpanan arsip laporan pajak secara lebih praktis dan terstruktur.



Gambar 3. 18 Halaman Cetak Laporan

1. Perancangan Halaman data Rekanan

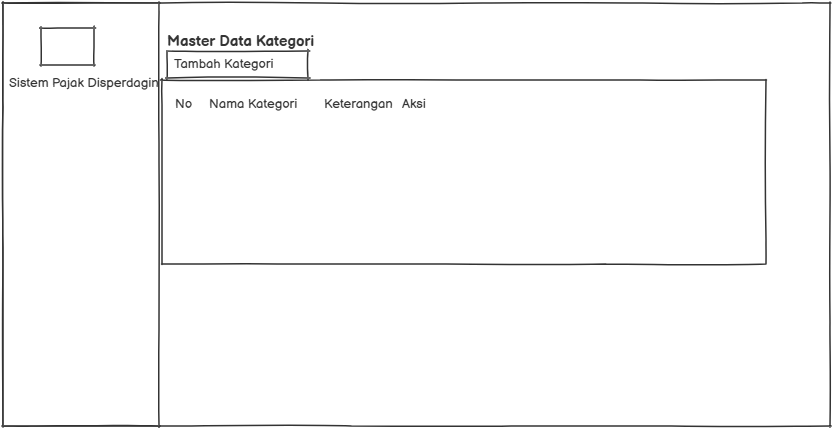
Fitur Master Data Rekanan berfungsi sebagai pusat pengelolaan informasi mitra atau perusahaan yang terlibat dalam setiap transaksi pajak. Melalui halaman ini, admin dapat dengan mudah menambahkan, memperbarui, maupun menghapus data rekanan sesuai kebutuhan. Informasi yang dikelola mencakup nama perusahaan, NPWP, status PKP (Pengusaha Kena Pajak), serta detail kontak terkait. Dengan adanya fitur ini, setiap transaksi dapat terhubung langsung dengan data mitra yang valid, sehingga membantu mengurangi potensi kesalahan dalam proses pencatatan dan pelaporan pajak.



Gambar 3. 19 Halaman Input Rekanan

1. Perancangan Halaman Kategori

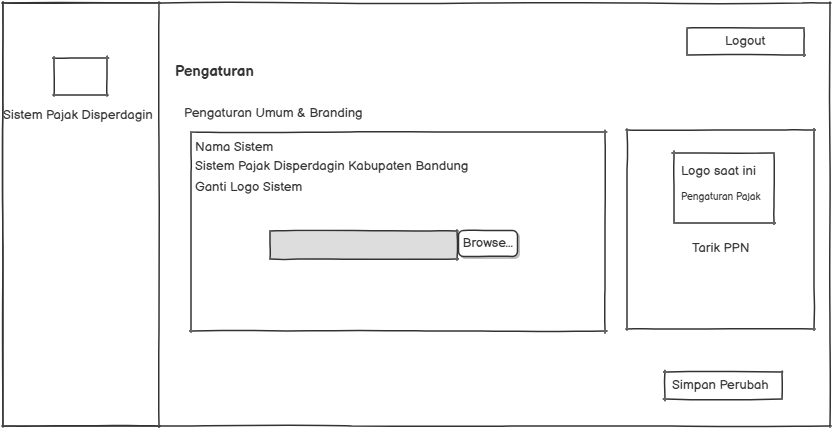
Fitur Master Data Kategori berfungsi sebagai mekanisme pengelompokan transaksi berdasarkan jenis atau kategori tertentu, seperti barang, jasa, maupun kategori lain yang relevan dengan kebutuhan sistem. Pengelompokan ini memiliki peran penting dalam meningkatkan keteraturan dan efisiensi pengelolaan data transaksi. Dengan struktur yang lebih rapi, pengguna dapat dengan mudah melakukan pencarian, penyaringan, maupun analisis data transaksi sesuai kategori masing-masing. Selain itu, fitur ini sangat mendukung proses penyusunan laporan pajak, karena laporan yang dihasilkan sering kali memerlukan klasifikasi transaksi yang jelas dan konsisten. Dengan adanya kategori yang terstandarisasi, potensi kesalahan dalam pencatatan atau pelaporan dapat diminimalkan, sekaligus mempermudah proses audit dan verifikasi data. Secara keseluruhan, Master Data Kategori tidak hanya mempermudah manajemen data internal, tetapi juga meningkatkan akurasi dan transparansi informasi pajak yang dikelola oleh system



Gambar 3. 20 Halaman Input Kategori

1. Perancangan Halaman Logout

Fitur Logout berfungsi untuk mengakhiri sesi pengguna setelah selesai mengoperasikan sistem pajak. Dengan menekan tombol ini, admin dapat keluar dari sistem secara aman, sehingga akun tidak dapat diakses oleh pihak lain tanpa melakukan login kembali. Tombol logout terletak di bagian bawah menu navigasi sebelah kiri dan ditandai dengan ikon berwarna merah untuk memudahkan identifikasi. Fitur ini sangat penting dalam menjaga keamanan sistem, khususnya untuk melindungi kerahasiaan data pajak yang dikelola.



Gambar 3. 21 Halaman Logout

#### 3.8.1 Perancangan Database

Perancangan database bertujuan untuk menyediakan struktur data yang efektif, terintegrasi, dan konsisten untuk mendukung operasional sistem informasi klasifikasi pajak daerah. Database diatur sedemikian rupa agar dapat menyimpan, mengelola, dan mengakses informasi transaksi keuangan dan pajak dengan cepat serta mengurangi redudansi data.

Berikut adalah struktur tabel yang digunakan dalam sistem:

1. Tabel Users

Tabel users digunakan untuk menyimpan data pengguna (admin dan pegawai).

Tabel 3. 13 Perancangan Table Users

| **Nama Field** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | int | Primary Key, Auto Increment |
| nama\_lengkap | varchar(100) | Nama lengkap pengguna |
| username | varchar(50) | Username untuk login (Unique) |
| password | varchar(255) | Password akun (Terenkripsi) |
| role | enum('admin','pegawai') | Hak akses pengguna |
| created\_at | timestamp | Waktu pembuatan akun |

2. Tabel rekanan

Tabel rekanan digunakan untuk menyimpan data identitas perusahaan atau pihak ketiga yang bertransaksi.

Tabel 3. 14 Perancangan Tabel Rekanan

| **Nama Field** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | int | Primary Key, Auto Increment |
| nama\_perusahaan | varchar(255) | Nama instansi atau perusahaan |
| npwp | varchar(25) | Nomor Pokok Wajib Pajak |
| status\_pkp | enum('PKP','Non-PKP') | Status Pengusaha Kena Pajak |
| kontak | varchar(50) | Nomor kontak yang bisa dihubungi |

3. Tabel kategori

Tabel kategori digunakan untuk mengelompokkan jenis transaksi atau belanja agar mudah diklasifikasikan.

Tabel 3. 15 Rancangan table kategori

| Nama Field | Tipe Data | Keterangan |
| --- | --- | --- |
| id | int | Primary Key, Auto Increment |
| nama\_kategori | varchar(100) | Nama kategori belanja |
| keterangan | text | Deskripsi tambahan mengenai kategori |

4. Tabel transaksi

Tabel transaksi adalah tabel utama untuk mencatat riwayat belanja, nilai pajak, dan status validasinya.

Tabel 3. 16 Rancangan table Transaksi

| Nama Field | Tipe Data | Keterangan |
| --- | --- | --- |
| id | int | Primary Key, Auto Increment |
| user\_id | int | Foreign Key (Tabel users) |
| rekanan\_id | int | Foreign Key (Tabel rekanan) |
| kategori\_id | int | Foreign Key (Tabel kategori) |
| tanggal\_transaksi | date | Tanggal terjadinya transaksi |
| uraian | text | Penjelasan detail transaksi |
| nilai\_transaksi | decimal(20,2) | Total nilai belanja |
| nilai\_ppn | decimal(20,2) | Nilai Pajak Pertambahan Nilai |
| nilai\_pph21 | decimal(20,2) | Nilai PPh Pasal 21 |
| nilai\_pph22 | decimal(20,2) | Nilai PPh Pasal 22 |
| nilai\_pph23 | decimal(20,2) | Nilai PPh Pasal 23 |
| nilai\_pajak\_daerah | decimal(20,2) | Nilai Pajak Daerah |
| bukti\_transaksi | varchar(255) | Nama file bukti transaksi |
| klasifikasi\_pajak | varchar(100) | Hasil penentuan jenis pajak |
| status\_validasi | enum('Pending','Valid','Tolak') | Status validasi oleh admin |

5. Tabel pengaturan

Tabel pengaturan digunakan untuk menyimpan konfigurasi sistem yang bersifat dinamis.

Tabel 3. 17 Rancangan Table Pengaturan

| **Nama Field** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| setting\_key | varchar(50) | Primary Key, Kunci pengaturan |
| setting\_value | text | Isi nilai konfigurasi |

6. Tabel log\_aktivitas (Tabel Tambahan)

Tabel log\_aktivitas digunakan untuk merekam jejak audit (audit trail) keamanan sistem.

Tabel 3. 18 Rancangan Table log\_aktivitas

| **Nama Field** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | int | Primary Key, Auto Increment |
| user\_id | int | Foreign Key (Tabel users) |
| aktivitas | varchar(255) | Deskripsi kegiatan yang dilakukan |
| ip\_address | varchar(45) | Alamat IP pengguna |
| created\_at | timestamp | Waktu aktivitas terjadi |

7. Tabel dokumen\_pendukung

Tabel dokumen\_pendukung digunakan jika satu transaksi memiliki banyak lampiran file.

Tabel 3. 19 Rancangan table dokumen\_pendukung

| **Nama Field** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | int | Primary Key, Auto Increment |
| transaksi\_id | int | Foreign Key (Tabel transaksi) |
| nama\_file | varchar(255) | Nama file yang diunggah |
| tipe\_dokumen | varchar(100) | Ekstensi dokumen (PDF/JPG) |
| uploaded\_at | timestamp | Waktu pengunggahan file |

3.9.1 Rencana Pengujian Black Box

| No | Fitur / Kelas Uji | Skenario Pengujian | Data Masukan (Input) | Hasil yang Diharapkan |
| --- | --- | --- | --- | --- |
| 1 | Login (Keamanan) | Menguji validasi akun Admin dan Pegawai | Username & Password yang salah | Sistem menolak akses dan menampilkan pesan "Login Gagal". |
|  |  | Menguji akses masuk dengan data benar | Username & Password yang benar | Sistem menerima akses dan mengarahkan ke Dashboard sesuai hak akses (Admin/Pegawai). |
| 2 | Input Transaksi Manual | Menguji kelengkapan form input | Form dibiarkan kosong lalu disimpan | Sistem menampilkan peringatan "Data tidak boleh kosong". |
|  |  | Menguji penyimpanan data valid | Data tanggal, rekanan, nominal, dan bukti diisi lengkap | Data tersimpan ke database dengan status "Pending". |
| 3 | Import Excel (Klasifikasi AI) | Menguji fungsi upload file | Mengupload file bukan format .xlsx (misal .pdf) | Sistem menolak file dan menampilkan pesan error format. |
|  |  | Menguji Akurasi Decision Tree | Mengupload file Excel berisi data transaksi dengan nominal > 2 Juta dan PKP = Ya | Sistem secara otomatis mengklasifikasikan data tersebut sebagai "PPN". |
|  |  |  | Mengupload file Excel berisi data transaksi jasa dengan nominal tertentu | Sistem secara otomatis mengklasifikasikan data tersebut sebagai "PPh 23" atau jenis lain sesuai logika algoritma. |
| 4 | Validasi Transaksi (Admin) | Menguji fitur persetujuan data | Admin menekan tombol "Valid" pada data pending | Status transaksi berubah menjadi "Valid" dan masuk ke rekap laporan. |
|  |  | Menguji fitur penolakan data | Admin menekan tombol "Tolak" | Status transaksi berubah menjadi "Ditolak" dan tidak dihitung di laporan. |
| 5 | Manajemen Rekanan | Menguji tambah data rekanan | Input Nama CV/PT, NPWP, dan Status PKP | Data rekanan baru berhasil ditambahkan ke master data. |
| 6 | Cetak Laporan | Menguji filter dan output laporan | Memilih filter rentang tanggal (Start Date - End Date) | Sistem menampilkan preview laporan hanya pada tanggal yang dipilih. |
|  |  | Menguji fungsi unduh dokumen | Menekan tombol "Print/Export" | Sistem menghasilkan file PDF/Excel laporan yang siap cetak. |

# BAB 4

## HASIL DAN PEMBAHASAN

### 4.1 Hasil

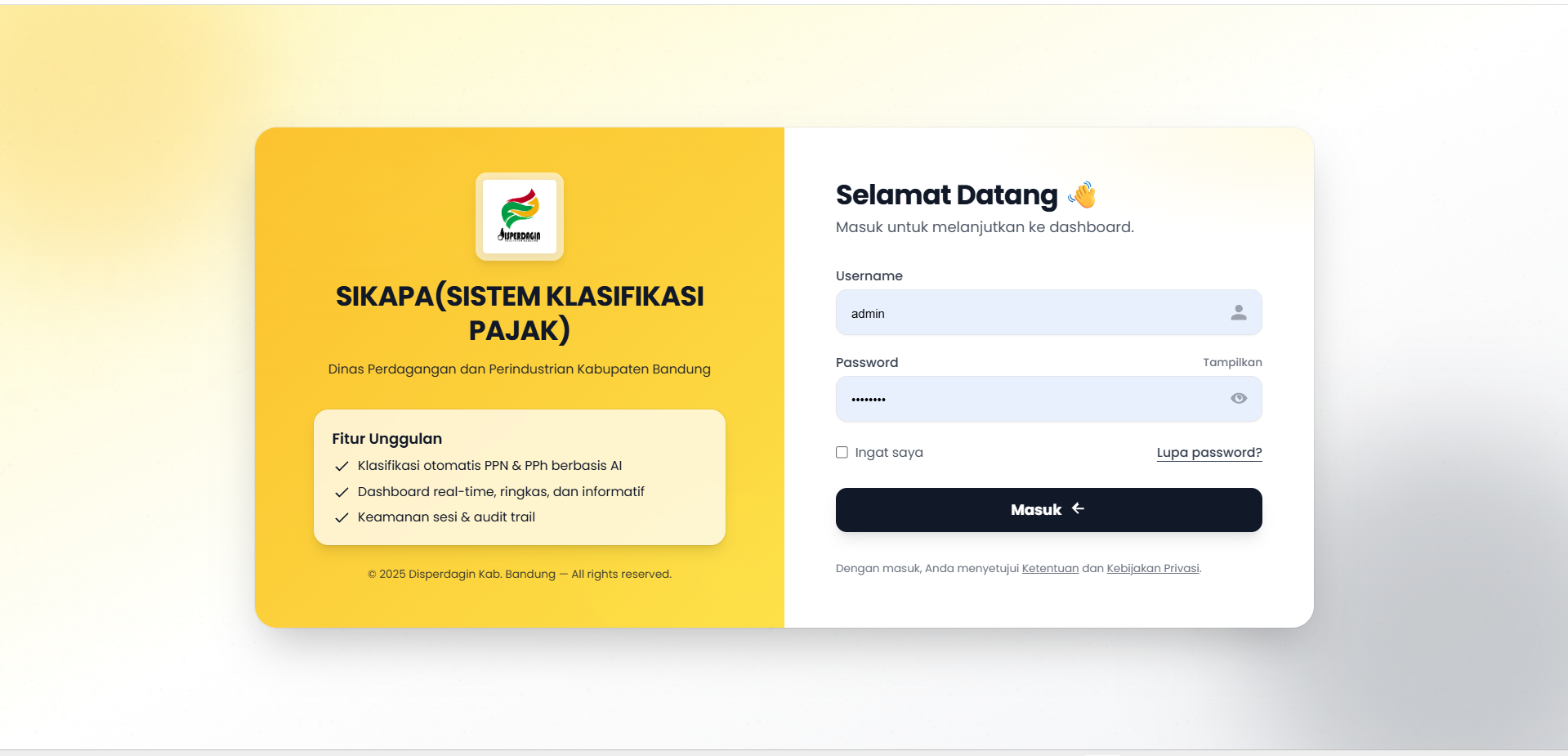
Berikut merupakan hasil dan pembahasan pada Sistem Informasi algoritma decision tree dalam klasifikasi transaksi ppn dan pph pada sistem informasi pajak digital: studi kasus pada dinas perdagangan dan perindustrian kabupaten bandung

#### 4.1.1 Halaman

Webiste Sistem Sistem Informasi algoritma decision tree dalam klasifikasi transaksi ppn dan pph pada sistem informasi pajak digital Berikut merupakan hasil dari tampilan halaman pada Sistem Informasi Manajemen Sistem Informasi algoritma decision tree dalam klasifikasi transaksi ppn dan pph pada sistem informasi pajak digital.

1. Halaman Login Admin

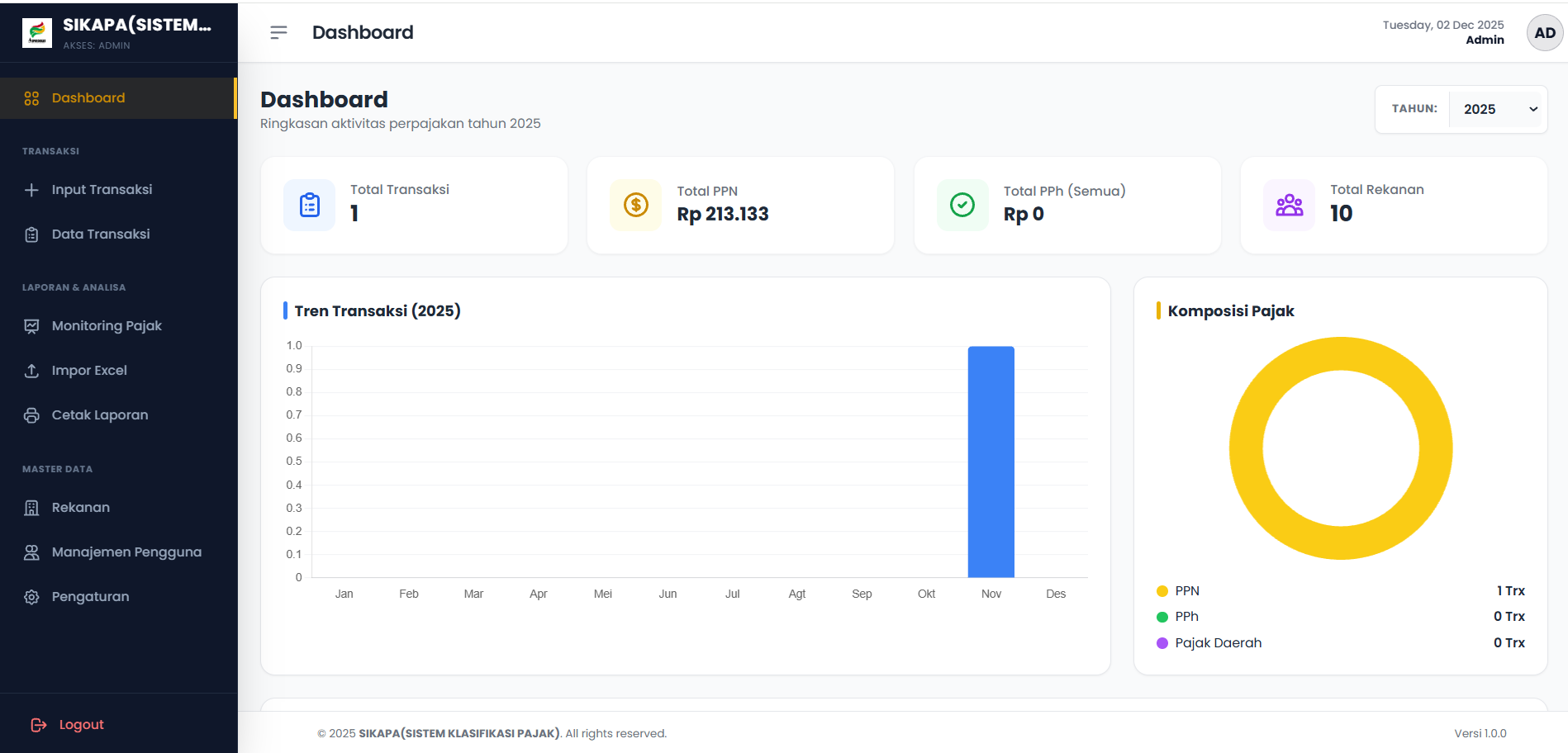
Halaman Login Admin merupakan halaman yang pertama kali di tampilkan saat admin mengunjungi website. Berikut merupakan tampilan login Admin :



*Gambar 4. 1* Halaman Login Admin

2. Halaman Dashboard Admin

Halaman Dashboard Admin Adalah halaman yang Dimana Ketika admin sudah login maka akan masuk ke halaman dashboard. Di halaman ini admin dapat melihat Total Transaksi, total ppn, total pph, dan toal rekanan, dan admin juga bisa memantau chart tren transaksi yang bisa di filter pertahun.



*Gambar 4. 2* Halaman Dashboard Admin

3. Halaman Input Transaksi

Halaman Inputr transaksi Adalah halaman untuk admin agar input transaksi baru yang Dimana bisa memilih tanggal tranaksi, rekanan, klasifikasi pajak, kategori pengeluaran, dan nilai transaksi, uraian/keterangan dan buktri transaksi.

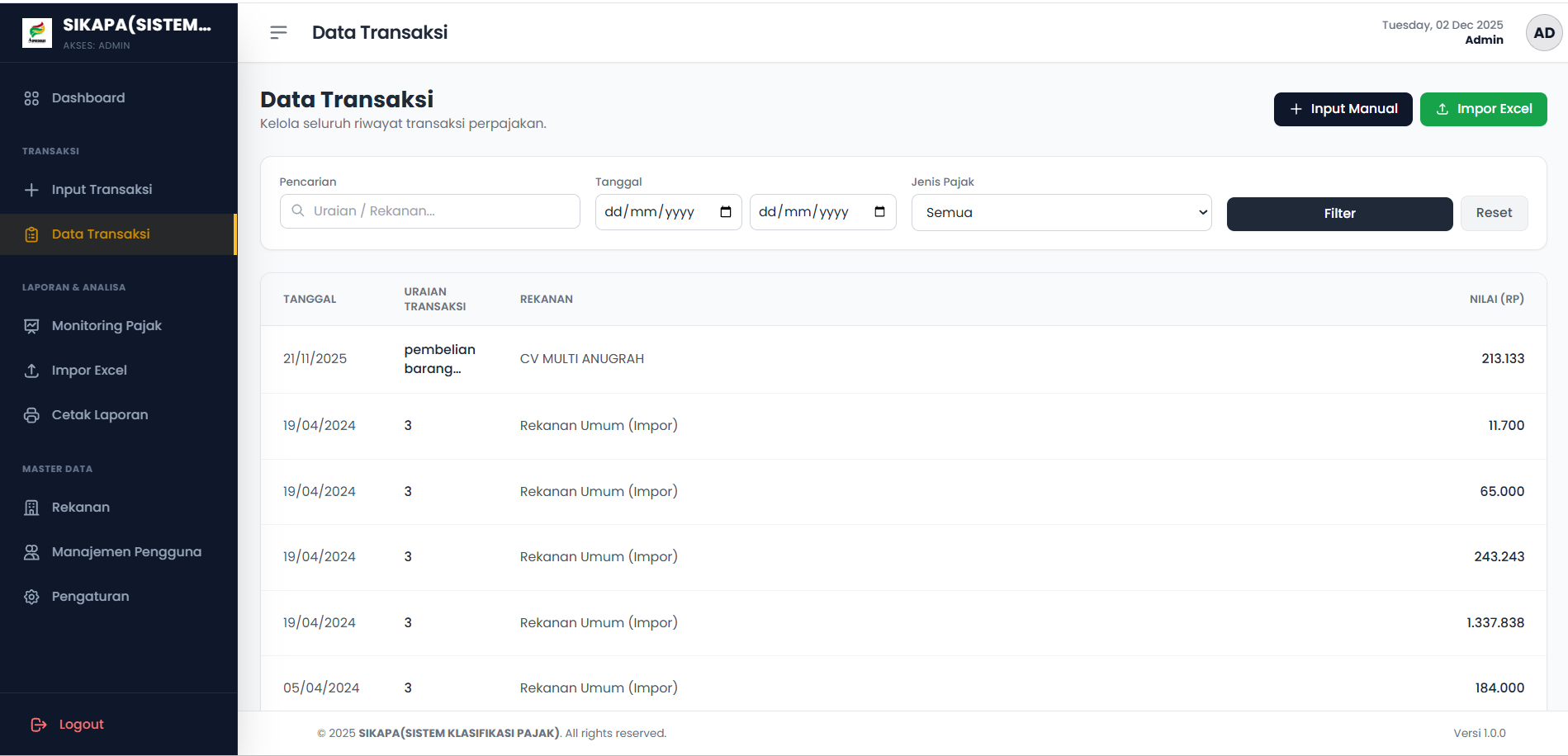
A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 3* Halaman Input Transaksi

4. Halaman Daftar Transaksi

Halaman Daftar Transaksi Adalah halaman yang Dimana admin bisa melihat dafrar transaksi yang di input atau melalui ai(excel), lalu bisa di filter sesuai pencarian, tanggal, jenis pajak, dan dapat menampilkan data tanggal, uraian transaksi, rekanan, serta total pajak yang di hitung.



*Gambar 4. 4* Halaman Daftar Transaksi

5. Halaman Monitoring Pajak

Halaman monitoring pajak Adalah halaman yang Dimana admin dapat melihat dan memonitoring daftar total pajak daerah per tahun, total ppn, total pph 21,23 dan ada chart serta rinciannya.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 5* Halaman Monitoring

6. Halaman Import transaksi

Halaman Import transaksi merupakan halaman untuk meng-import data dengan mudah tanpa harus input manual yang Dimana disini bisa di import denagn file excel yang sekali banyak data jadi admin mudah untuk input data tanpa input satu satu disini lah metode decision tree diterapkan.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 6* Halaman Import Excel(AI)

7. Halaman Cetak laporan, disini admin dapat mencetak laporan dengan memfilter dari tanggal, rekanan dan jenis pajakA screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 7* Halaman Cetak Laporan

8. Halaman Priview Cetak laporan

Halaman ini Adalah halaman untuk print out laporan yang sudah di fiter admiin tadi

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 8* Halaman Priview Cetak Laporan

9. Halaman Master data rekanan

Halaman ini Adalah halaman yang di mana admin dapat input rekanan/bisa otomatis terinput melalui excel dan dapat melihat data data rekanan.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 9* Halaman Master Data Rekanan

10. Halaman Manajemen Pengguna

Halaman ini merupakan halaman untuk menampilkan serta CRUD pengguna dan mengelola akun di system ini.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 10* Halaman Manajemen Pengguna

11. Halaman Pengaturan system

Halaman ini Adalah halaman untuk admin mengubah pengaturan umum seperti nama siste, logo system, dan pengaturann pajak.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 11* Halaman Mengelola Pengaturan

12. Halaman login staff

Halaman login pegawai merupakan halaman yang Dimana pegawai/staff masuk ke dalam system.

A screenshot of a computer screen

AI-generated content may be incorrect.

*Gambar 4. 12* Halaman Login Pegawai

13. Halaman Dashboard Staff

Halaman Dashhboard Staff Adalah yang Dimana pegawai bisa melihat data total transaksi, total ppn, total pph serta total rekanan

A screenshot of a computer

AI-generated content may be incorrect.

14. Halaman Inputr Transaksi staf

Halaman Inputr transaksi Adalah halaman untuk admin agar input transaksi baru yang Dimana bisa memilih tanggal tranaksi, rekanan, klasifikasi pajak, kategori pengeluaran, dan nilai transaksi, uraian/keterangan dan buktri transaksi.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 13* Halaman Input Transaksi Staff

15. Halaman Data Transaksi staff

Pada halaman ini staff dapat melihat dafta transaksi

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 14* Data Transaksi Staff

16. Halaman Monitoring Pajak

Halaman monitoring pajak Adalah halaman yang Dimana admin dapat melihat dan memonitoring daftar total pajak daerah per tahun, total ppn, total pph 21,23 dan ada chart serta rinciannya.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 4. 15* Halaman Monitoring Pajak

* + 1. Pengujian pada Sistem Klasifikasi Pajak Decision Tree

Pengujian sistem dilakukan untuk memastikan bahwa setiap fitur dalam sistem laporan keuangan dapat berjalan dengan baik, sesuai dengan fungsinya, serta tidak terjadi kesalahan (error) saat digunakan oleh admin. Tahap ini penting dilakukan sebelum sistem benar-benar diterapkan secara penuh dalam lingkungan kerja agar dapat meminimalisir risiko kesalahan data maupun penggunaan.

Metode pengujian yang digunakan dalam sistem ini adalah *black box testing*. Pengujian dilakukan dengan memeriksa output dari sistem berdasarkan input yang diberikan, tanpa melihat kode program secara langsung. Dengan kata lain, fokus utama pengujian adalah pada bagaimana sistem merespon tindakan dari pengguna dan apakah setiap fungsi bekerja sebagaimana mestinya.

Pengujian dilakukan terhadap seluruh fitur inti, mulai dari proses login, input laporan, klasifikasi, hingga fitur visualisasi dan cetak laporan. Hasil pengujian kemudian dicatat dalam bentuk tabel yang memperlihatkan nama fitur, skenario pengujian, data uji yang digunakan, serta hasil pengujian. Berikut adalah ringkasan hasil pengujian sistem:

*Tabel 4. 1 Hasil Pengujian Sistem dengan Metode Black Box Testing*

| **No.** | **Fitur yang Diuji** | **Skenario Pengujian** | **Input/aksi** | **Hasil yang Diharapkan** | **Status** |
| --- | --- | --- | --- | --- | --- |
| 1 | Login Admin | Admin memasukkan username dan password | Username: admin Password: admin123 | Sistem menampilkan dashboard utama | Berhasil |
| 2 | Input Transaksi | Admin mengisi form transaksi dan menyimpandan menyimpan | Tanggal, Rekanan, Kategori, Nilai, Uraian | Data tersimpan dan muncul di halaman Data Transaksi | Berhasil |
| 3 | Data Transaksi | Menampilkan daftar transaksi | |  | | --- |   Data transaksi yang sudah diinput | |  | | --- |   Data tampil lengkap dengan opsi **Edit** dan **Hapus** | Berhasil |
| 4 | Impor Excel | Admin mengunggah file transaksi .xlsx | File Excel berisi transaksi valid | Data berhasil diimpor dan tampil di Data Transaksi | Berhasil |
| 5 | Cetak Laporan | Admin memilih filter laporan dan cetak | Rentang tanggal, Rekanan, Jenis Pajak | Laporan ditampilkan dan dapat diunduh dalam format Excel | Berhasil |
| 6 | |  | | --- |  | Rekanan | | --- | | Admin menambah dan mengelola data rekanan | Nama perusahaan, NPWP, Status PKP, Kontak | Data rekanan baru tersimpan dan muncul di daftar | Berhasil |
| 7 | Kategori | Admin menambah kategori baru | |  | | --- |  |  | | --- |   Nama kategori, Keterangan | Data kategori tersimpan dan tampil di tabel | Berhasil |
| 8 | Pengaturan | Admin mengubah nama sistem/logo/tarif PPN | Nama sistem, logo, tarif PPN | Perubahan tersimpan dan langsung diterapkan | Berhasil |
| 9 | Logout | Admin klik tombol logout | - | |  | | --- |   Sistem mengakhiri sesi dan kembali ke halaman login | Berhasil |

### 4.2 Pembahasan

Pada Sistem Informasi Klasifikasi dan Manajemen Pajak (SIKAPA) yang dikembangkan berbasis *website*, terdapat berbagai aspek yang dapat dianalisis berdasarkan hasil implementasi dan pengujian sistem. Secara umum, sistem ini telah berhasil mengakomodasi seluruh fitur utama yang dirancang untuk mengatasi permasalahan pencatatan dan pelaporan pajak yang sebelumnya dilakukan secara manual. Fitur-fitur tersebut meliputi input transaksi harian oleh pegawai, validasi data oleh admin, impor data massal dari Excel, hingga monitoring realisasi pajak secara *real-time*. Semua proses tersebut kini terintegrasi dalam satu *platform* digital yang dapat diakses oleh pihak-pihak berwenang di instansi.

Struktur sistem dirancang dengan pendekatan berbasis *role* (peran), yang memastikan keamanan dan integritas data. Pegawai sebagai staf operasional dapat melakukan pencatatan transaksi belanja dan mengunggah bukti dukung secara digital. Sementara itu, Admin memiliki wewenang lebih luas untuk memverifikasi kebenaran data melalui fitur validasi, mengelola data master (rekanan dan pengguna), serta melakukan impor data dalam jumlah besar. Pemisahan hak akses ini meminimalisir risiko manipulasi data dan kesalahan input yang tidak terverifikasi.

Salah satu keunggulan utama dari sistem ini adalah kemampuan klasifikasi pajak otomatis. Sistem dirancang untuk mendeteksi jenis pajak (PPN, PPh 21, PPh 22, PPh 23, PPh 4 ayat 2, atau Pajak Daerah) berdasarkan nominal yang diinputkan dari file Excel maupun input manual yang telah divalidasi. Hal ini secara signifikan mengurangi beban kerja administratif dalam memilah jenis pajak secara manual yang rentan terhadap *human error*.

Dari sisi antarmuka (*user interface*), sistem dirancang dengan tampilan *dashboard* yang informatif, menampilkan grafik tren transaksi dan komposisi jenis pajak secara visual. Pengujian sistem dilakukan menggunakan metode *Black Box Testing*, di mana setiap fitur diuji berdasarkan skenario fungsional. Hasil pengujian menunjukkan bahwa alur kerja mulai dari *login*, input data, validasi, hingga pencetakan laporan akhir berjalan sesuai dengan rancangan tanpa kendala fungsional yang berarti.

Secara keseluruhan, implementasi SIKAPA memberikan kontribusi nyata dalam mempercepat proses rekapitulasi pajak, meningkatkan akurasi perhitungan nominal pajak, serta mempermudah proses audit melalui ketersediaan arsip digital yang terpusat.

#### 4.2.1 Kelebihan Sistem Informasi Klasifikasi dan Manajemen Pajak (SIKAPA)

Berikut merupakan beberapa kelebihan dari sistem yang telah dibangun:

1. Klasifikasi Pajak Otomatis dan Akurat: Sistem mampu mengidentifikasi dan mengelompokkan jenis pajak (PPN, PPh, Pajak Daerah) secara otomatis, terutama pada fitur impor Excel, sehingga mengurangi kesalahan perhitungan manual.
2. Efisiensi Input Data Massal: Fitur impor data dari format Excel (.xlsx) memungkinkan admin untuk memasukkan ratusan data transaksi sekaligus dalam waktu singkat, menggantikan proses input satu per satu yang memakan waktu.
3. Mekanisme Validasi Berjenjang: Adanya fitur validasi transaksi oleh Admin memastikan bahwa data yang diinput oleh Pegawai telah diperiksa kebenaran nominal dan kelengkapan buktinya sebelum masuk ke dalam laporan keuangan resmi.
4. Monitoring Real-Time: Tersedianya *dashboard* monitoring yang menyajikan grafik tren transaksi bulanan dan proporsi jenis pajak membantu pimpinan atau admin dalam memantau realisasi pajak kapan saja.
5. Penyimpanan Bukti Transaksi Digital: Sistem memfasilitasi pengunggahan bukti transaksi (nota/faktur) dalam format digital, sehingga memudahkan proses pencarian kembali (*retrieval*) saat dibutuhkan untuk keperluan audit.
6. Pemisahan Hak Akses (Role-Based): Pembagian peran yang jelas antara Admin dan Pegawai menjaga keamanan data, di mana Pegawai tidak dapat mengubah pengaturan sistem atau memvalidasi data mereka sendiri.
7. Laporan Siap Cetak: Sistem dapat men-generate laporan rekapitulasi pajak secara otomatis dengan format standar (dilengkapi kop surat dan kolom tanda tangan), siap untuk dicetak atau diekspor ke PDF.
8. Fleksibilitas Filter Data: Fitur pencarian dan filter berdasarkan rentang tanggal, nama rekanan, atau jenis pajak memudahkan pengguna dalam menelusuri riwayat transaksi spesifik.
9. Manajemen Data Master Terpusat: Pengelolaan data rekanan (vendor) dan pengguna sistem dilakukan secara terpusat, memastikan konsistensi data di seluruh transaksi.
10. Antarmuka Responsif: Tampilan aplikasi dapat menyesuaikan dengan berbagai ukuran layar, memudahkan akses baik melalui komputer *desktop* maupun perangkat *mobile*.

#### 4.2.2 Kekurangan Sistem Informasi Klasifikasi dan Manajemen Pajak (SIKAPA)

Meskipun telah memenuhi kebutuhan utama, sistem ini masih memiliki beberapa kekurangan yang dapat dikembangkan di masa mendatang:

1. Belum Terintegrasi dengan Sistem DJP Online: Sistem saat ini masih berdiri sendiri (*standalone*) dan belum terhubung secara langsung (API) dengan sistem Direktorat Jenderal Pajak (DJP) untuk pelaporan e-Bupot atau e-Faktur secara otomatis.
2. Ketergantungan pada Format Excel: Fitur impor data sangat bergantung pada format *template* Excel yang baku. Jika pengguna mengubah struktur kolom pada file Excel, proses impor berpotensi mengalami kegagalan.
3. Notifikasi Belum Real-Time (Email/WA): Sistem belum dilengkapi dengan notifikasi via Email atau WhatsApp kepada Pegawai jika transaksi yang mereka ajukan ditolak atau disetujui oleh Admin, sehingga pengecekan status masih harus dilakukan dengan *login* ke sistem.

# BAB 5

## KESIMPULAN DAN SARAN

### 5.1 Simpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada Sistem Informasi Klasifikasi dan Manajemen Pajak (SIKAPA), dapat ditarik beberapa kesimpulan sebagai berikut:

1. Sistem informasi berbasis *website* untuk manajemen perpajakan telah berhasil dikembangkan dan diimplementasikan guna menggantikan proses pencatatan manual. Sistem ini mencakup seluruh alur kerja mulai dari input transaksi harian oleh pegawai, impor data massal melalui Excel, validasi data oleh admin, hingga pencetakan laporan rekapitulasi pajak.
2. Penerapan logika klasifikasi otomatis dalam sistem mampu mendeteksi dan mengelompokkan jenis pajak (PPN, PPh 21, PPh 22, PPh 23, PPh 4 ayat 2, dan Pajak Daerah) secara akurat berdasarkan nilai nominal dan data yang diinput. Hal ini terbukti efektif mengurangi risiko kesalahan perhitungan (*human error*) yang sering terjadi pada metode konvensional.

### 5.2 Saran

Demi peningkatan kualitas dan fungsionalitas sistem di masa mendatang, penulis mengajukan beberapa saran pengembangan sebagai berikut:

1. Disarankan agar sistem dapat diintegrasikan langsung (*API Integration*) dengan layanan DJP Online atau e-Faktur. Hal ini bertujuan agar data pajak yang telah tervalidasi di sistem SIKAPA dapat langsung dilaporkan ke Direktorat Jenderal Pajak tanpa perlu input ulang di aplikasi terpisah.
2. Pengembangan fitur notifikasi *real-time* melalui Email atau WhatsApp Gateway sangat disarankan. Fitur ini berguna untuk memberikan informasi langsung kepada pegawai apabila transaksi yang mereka ajukan telah disetujui atau ditolak oleh Admin, sehingga mempercepat respons perbaikan data.
3. Sistem dapat dikembangkan dengan menambahkan teknologi *Optical Character Recognition* (OCR) pada fitur *upload* bukti transaksi. Dengan teknologi ini, sistem dapat memindai foto nota atau kuitansi dan mengisi formulir input (tanggal, nominal, uraian) secara otomatis, sehingga semakin mengefisiensikan waktu kerja pegawai.
4. Penambahan fitur *Audit Trail* atau log aktivitas pengguna yang lebih mendetail disarankan untuk meningkatkan aspek keamanan, sehingga admin dapat melacak siapa yang melakukan perubahan data sensitif dan kapan perubahan tersebut dilakukan.
5. Pengembangan aplikasi dalam versi *mobile* (Android/iOS) dapat dipertimbangkan untuk memudahkan pegawai lapangan dalam mengunggah bukti transaksi secara langsung dari lokasi pembelian barang atau jasa.

# DAFTAR PUSTAKA

Ambareesh, & Augustine. (2023). *Overfitting and Pruning in Decision Tree Algorithms*. International Journal of Computer Science.

Aswiputri. (2022). *Konsep Dasar Database dan Pengelolaannya*. Jurnal Teknologi Informasi.

Budiman, dkk. (2023). *Pemanfaatan Laragon sebagai Web Server Lokal*. Jurnal Sistem Komputer.

Chen, dkk. (2024). *Persiapan dan Pembersihan Data dalam Data Mining*. Jurnal Informatika Global.

Direktorat Jenderal Pajak. (2021). *Ketentuan Umum dan Tata Cara Perpajakan*. Kementerian Keuangan Republik Indonesia.

Direktorat Jenderal Pajak. (2022). *Mekanisme Transaksi Perpajakan di Indonesia*. Kementerian Keuangan Republik Indonesia.

Fitri Khoiry Tamami Salam, & Septanto. (2024). *Penerapan Metode Waterfall dalam Pengembangan Perangkat Lunak*. Jurnal Rekayasa Perangkat Lunak.

Gunawan, & Astuti. (2022). *Konsep Dasar Algoritma Decision Tree*. Jurnal Informatika dan Komputer.

Hanafia Lubis, & Saragih. (2024). *Analisis Klasifikasi Kesadaran Wajib Pajak PBB*. Jurnal Ekonomi dan Pajak.

Haris, M. (2022). *Evaluasi Digitalisasi Perpajakan Studi Kasus di KPP Pratama Magelang*. DSpace UII (Universitas Islam Indonesia).

Intan Diah Hardyatman, & Hasan. (2025). *Analisis Sentimen Publik Terkait Kebijakan Kenaikan Pajak Menggunakan Decision Tree*. Jurnal Riset Opini Publik.

Jakse. (2024). *Struktur dan Logika Percabangan pada Decision Tree*. Journal of Data Science.

Katsevich. (2022). *Metode Minimal Cost-Complexity Pruning (MCCP) pada CART*. Statistical Learning Journal.

Kementerian Keuangan Republik Indonesia. (2022). *Kebijakan Pajak Digital dan Ekonomi Berbasis Teknologi*.

Kirmani, dkk. (2024). *Metode Pembelajaran Rekursif dalam Algoritma Klasifikasi*. International Journal of AI.

Kurniawan, & Romzi. (2022). *Pemanfaatan Visual Studio Code dalam Pengembangan Aplikasi Web*. Jurnal Pemrograman Web.

Marlina, et al. (2023). *Penggunaan Gini Index dan Entropy dalam Algoritma Data Mining*. Jurnal Teknologi Informasi.

Maskur, & Wibowo. (2024). *Taxpayer Awareness Classification Using Decision Tree and Naïve Bayes Methods*. Jurnal JAIC (Polibatam).

Mitchell. (2021). *PHP Programming: Developing Dynamic Web Applications*. O'Reilly Media.

Mohammadi-Pirouz, dkk. (2024). *Evaluasi Model Klasifikasi Menggunakan Matriks Akurasi dan ROC-AUC*. Journal of Machine Learning Research.

Orounla. (2023). *Optimasi Pemisahan Node pada Algoritma CART*. Journal of Computational Statistics.

Rafadi, & Industri. (2021). *Simbol dan Notasi dalam Unified Modeling Language (UML)*. Jurnal Teknik Industri dan Sistem Informasi.

Ramadhan, & Mukhaiyar. (2020). *Pengelolaan Database Menggunakan MySQL dan phpMyAdmin*. Jurnal Teknologi dan Sistem Informasi.

Ramadhon. (2024). *Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank*. Universitas Djuanda.

Rasiban, dkk. (2024). *Analisis dan Perancangan Sistem Menggunakan UML (Use Case, Activity, dan Class Diagram)*. Jurnal Rekayasa Sistem.

Renganathan. (2021). *Machine Learning for Data Scientists*. Packt Publishing.

Roberts. (2020). *Pure PHP Development Handbook*.

Rokhimakhumullah, Ningsih, Firmansyah, & Hanafiyah. (2024). *Analisis potensi e-commerce melalui implementasi data mining dalam perpajakan: Sebuah studi komparasi*. Juremi: Jurnal Riset Ekonomi.

Sari, dkk. (2023). *Penerapan Metode Prototype dalam Perancangan Sistem Informasi*. Jurnal Sistem Informasi.

Sasongko, dkk. (2021). *Pengujian Perangkat Lunak Menggunakan Metode Blackbox*. Jurnal Teknologi Informasi.

Septaraja, Joannes, Radhi, & Parhusip. (2024). *Implementasi Algoritma Decision Tree untuk Prediksi Efisiensi Biaya Bensin Kendaraan Bermotor Parenggean Menuju Palangkaraya*. Informatech: Jurnal Ilmiah Informatika dan Komputer.

Setiawan. (2021). *Konsep SDLC (Software Development Life Cycle) dalam Pengembangan Sistem*. Jurnal Manajemen Informatika.

Sharma, & Iqbal. (2023). *Algoritma Classification and Regression Tree (CART): Teori dan Implementasi*. International Journal of Data Science.

Siking, dkk. (2023). *Pemodelan Sistem Berorientasi Objek dengan UML*. Jurnal Informatika.

Supiyandi, dkk. (2022). *Tahapan Pengembangan Sistem Menggunakan Metode Waterfall*. Jurnal Sains dan Teknologi.

Tjandrida, & Dermawan. (2025). *Implementasi Algoritma Decision Tree pada Sistem Informasi Manajemen Inventarisasi Fakultas Vokasi UNESA*. Universitas Negeri Surabaya.

Undang-Undang Republik Indonesia Nomor 7 Tahun 2021 tentang Harmonisasi Peraturan Perpajakan.