PEMANFAATAN PEMROSESAN BAHASA ALAMI DALAM R UNTUK VISUALISASI PROFIL KELUHAN PELANGGAN FINTECH: EVALUASI KASUS ADAKAMI

Rony Faisol[1*]
[1]*S1 Matematika, Fakultas Sains dan Teknologi, UPBJJ UT Surabaya*

**Penulis korespondensi: pakgurukalkulus83@gmail.com*

ABSTRAK

Meskipun legal dan diawasi Otoritas Jasa Keuangan (OJK), anomali operasional platform *fintech lending* kerap luput dari proses audit konvensional. Ribuan ulasan pengguna menyimpan pola tersembunyi terkait pelanggaran standar operasional, namun volume data yang masif membuat pemantauan manual tidak efisien. Penelitian ini bertujuan mengonstruksi model ekstraksi informasi berbasis *Natural Language Processing* (NLP) menggunakan bahasa R guna memvisualisasikan profil keluhan pengguna pada aplikasi AdaKami. Pendekatan kuantitatif deskriptif diterapkan melalui teknik *web scraping* ulasan di Google Play Store. Proses NLP dieksekusi menggunakan paket *tm* dan *quanteda* dalam bahasa R, mencakup tahap persiapan data, *preprocessing*, pembobotan TF-IDF, dan tokenisasi *N-gram* untuk menjaga konteks kalimat, yang kemudian divisualisasikan melalui *ggplot2*. Hasil analisis mengungkap bahwa algoritma sukses mengisolasi klaster keluhan yang mengarah pada tiga anomali operasional spesifik: (1) Klaster pertama, friksi penolakan kredit, (2) Klaster kedua, krisis kepercayaan privasi data, (3) Klaster ketiga, anomali finansial dan penagihan, dan (4) Klaster keempat, kerusakan teknis UX/UI . Temuan ini membuktikan arsitektur NLP berfungsi efektif sebagai instrumen audit forensik digital. Secara praktis, *script* pemodelan R ini dapat diadopsi langsung oleh Otoritas Jasa Keuangan (OJK) dan Satgas PASTI (Pemeberantasan Aktivitas Keuangan Ilegal) sebagai purwarupa *Early Warning System* (EWS) untuk memantau kepatuhan secara otomatis, sekaligus mendesak manajemen AdaKami mereformasi prosedur pihak ketiga dan memperbaiki transparansi pencairan dana

Kata kunci: AdaKami, Bahasa R, *scraping* , *preprocessing*, visualisasi data

1 PENDAHULUAN

Pinjaman online, pinjol, menjadi sebuah tren solusi keuangan yang instan di masyarakat karena kemudahan akses dan proses yang cepat. Sebagaimana diberitakan dalam 100 hari pertama kepemimpinan Presiden Prabowo, istilah pinjaman online resmi diubah menjadi menjadi pinjaman daring, pindar (Respati & Arif, 2023). Perubahan ini bertujuan untuk membedakan layanan pendanaan yang legal dan berizin Otoritas Jasa Keuangan (OJK) dengan yang ilegal.

Menurut Ratuwalu et al. (2025) disebutkan bahwa pinjol sebagai alternatif pembiayaan yang menjanjikan akses keuangan yang inklusif, cepat, dan mudah dijangkau, terutama bagi masyarakat yang belum bisa dilayani oleh perbankan konvensional (*unbanked and underbanked*). Seseorang bisa mendapatkan pendanaan dari pinjol dengan hanya bermodalkan KTP dan aplikasi pindar maka pendanaan diterima dalam hitungan menit. OJK mencatat oustanding pembiayaan industri pindar pada April 2026 mencapai Rp102,07 triliun. Nilai tersebut tumbuh 26,11 persen secara tahunan. Sementara itu, tingkat risiko kredit macet secara agregat (TWP90) tercatat sebesar 4,62 persen pada April 2026. Pencapaian ini lebih meningkat daripada Maret 2026 yang sekitar 4,52 persen (Setiawan, 2026).

Akan tetapi, pertumbuhan yang pesat dan manfaat kemudahan yang ditawarkan pindar memunculkan permasalahan yang serius yang mengancam konsumen. Permasalahan yang sering terjadi berbentuk *cyber criminal* misalnya ancaman penyebaran data pribadi kepada pihak ketiga. Untuk penanganan permasalahan ini platform pindar legal telah menyediakan berbagai macam layanan aduan pelanggaran melalui call center, media sosial, dan email. Selain itu pengaduan mereka juga dapat disampaikan pada ulasan aplikasi pindar di Google Play Store. Ulasan konsumen biasanya dinyatakan dalam rating bintang dan narasi. Kepuasan pengguna terbukti berkorelasi kuat dengan inovasi desain yang didorong oleh masukan langsung yang pada akhirnya memperkuat pengelolaan kepercayaan pengguna pada platform digital (Lubis, 2023). Integrasi analisis sentimen yang adaptif menjadi kunci dalam menghasilkan wawasan strategis bagi ekosistem platform pindar yang berkelanjutan

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengonstruksi dan mengevaluasi model ekstraksi informasi berbasis *Natural Language Program* (ELP) dengan penggunaan bahasa *R* untuk memvisualisasikan profil keluhan pelanggan pada ulasan aplikasi pindar AdaKami di Google Play secara empiris. Hasil yang diharapkan adalah rancangan algoritma forensik digital yang nyata dan aplikatif, bukan sekadar teori tekstual. Melalui skrip *Early Warning System* (EWS) ini, pola anomali yang direpresentasikan dari keluhan pelanggan dapat dideteksi secara komputasional guna mempercepat regulator mengambil tindakan penertiban.

2 METODE

2.1 Desain Eksperimen dan Pengumpulan Data

Penelitian kuantitatif deskriptif ini dilaksanakan secara komputasional dengan periode penarikan data sampai bulan Juni 2026. Objek penelitian berfokus pada data teks ulasan pengguna aplikasi AdaKami. Menurut Pratama et al. (2021) disebutkan bahwa teknik *web scraping* digunakan untuk mendapatkan korpus data yang valid dan masif. Teknik ini berguna untuk pengumpulan data secara online yang lebih efisien dan efektif. Dengan teknik ini, konten utama dari suatu halaman situs dapat diekstrak, dikoleksi dan selanjutnya dapat diproses oleh proses pengindeksan. Penarikan data ini difilter secara spesifik pada populasi pengguna yang memberikan *rating* bintang 1 dan 2 karena rentang metrik tersebut secara historis memiliki korelasi absolut terhadap sentimen keluhan dan pelaporan anomali sistemik oleh konsumen (de Lima et al.,2024).

2.2 Alat, Bahan, dan Prosedur Pemrosesan Teks

Instrumen utama yang digunakan adalah perangkat lunak *R-Studio* dengan basis bahasa pemrograman R versi 4.4.1. Perangkat lunak bahasa R mendukung analisa data teks karena bahasa R memberikan operasi sederhana dan fungsi-fungsi dasar untuk analisa vektor karakter dalam teks misalnya *tolower*, *tm*, dan *quanteda* (Benoit et al., 2018). Bahan baku berupa fail *comma-separated values* (CSV) hasil *scraping* diproses melalui serangkaian algoritma pembersihan (*preprocessing*). Proses ini merupakan tahap untuk membangun satu set fitur yang relevan dari text pada dokumen. Semua set fitur dipilih untuk koleksi dokumen yang disebut repsentional model dan setiap dokumen diwakili oleh nilai numeric vector. Tujuan dari text preprocessing adalah untuk mengekstrak vector feature berkualitas tinggi untuk setiap dokumen, sehingga values yang diambil adalah values yang bernilai tinggi atau penting (Handayanto et al., 2021). Modul utama yang digunakan meliputi paket *tm* (Text Mining) dan *quanteda* yang dieksekusi untuk mereduksi *noise* pada data. Menurut Kurniawan et al., (2017) bahwa cara kerjanya meliputi *case folding* untuk penyeragaman huruf kecil, penghapusan tautan dan *emoticon*, serta *stopword removal* menggunakan kamus leksikon bahasa Indonesia untuk membuang konjungsi. Selanjutnya, dilakukan proses *stemming* guna memangkas imbuhan sehingga variasi kata kerja kembali ke bentuk dasar (Siregar et al., 2023).

2.3 Ekstraksi Fitur dan Analisa Visual

Metode pemecahan masalah di tahap lanjutan bertumpu pada tokenisasi N-gram, difokuskan pada model *bigram* (dua kata berurutan) agar makna klausa pengaduan tidak terdistorsi. Menurut Siregar et al. (2023) bahwa matriks dokumen-istilah dihitung menggunakan pendekatan statistik *Term Frequency-Inverse Document Frequency* (TF-IDF) yang dirumuskan sebagai berikut.

$$TF_{(t,d)} = \frac{f_{(t,d)}}{\sum f_{(t,d)}} \quad (1)$$

$$IDF_{(t,d)} = log\frac{N}{DF_t} + 1 \quad (2)$$

$$TF - IDF_{(t,d)} = TF_{(t,d)} \times IDF_{(t,d)} \quad (3)$$

Dijelaskan oleh Nasution & Monika (2022) bahwa proses mengubah data teks mentah ke representasi numerik menggunakan metode TF-IDF tersebut bertujuan agar data dapat diproses oleh model komputasi matematis.

Menurut Jurafsky & Martin (2024) bahwa TF-IDF ini dapat menghitung bobot kritis suatu frasa berdasarkan frekuensi kemunculannya berbanding terbalik dengan sebarannya di seluruh dokumen korpus . Tingkat kepercayaan pengambilan keputusan anomali didasarkan pada nilai bobot dominan (p-value ekuivalen < 0.05). Pada tahap akhir, data kuantitatif ditransformasikan menjadi visualisasi grafis yang interaktif dan mudah diinterpretasi oleh auditor menggunakan paket *ggplot2*.

3 HASIL DAN PEMBAHASAN

Berdasarkan metode yang diuraikan sebelumnya, berikut dijelaskan langkah-langkah mengonstruksi model ekstraksi informasi berbasis *Natural Language Processing* (NLP) yang disajikan dalam bentuk urutan gambar.

```
# ========================================
> # FASE 1: SCRAPING DENGAN PROTEKSI ENUM
> # ========================================
> cat("\n[1/4] Menyiapkan environment & menarik data...\n")

[1/4] Menyiapkan environment & menarik data...
> virtualenv_create("env-scraping")
virtualenv: env-scraping
> use_virtualenv("env-scraping", required = TRUE)
> py_install("google-play-scraper", envname = "env-scraping")
Using virtual environment "env-scraping" ...
Requirement already satisfied: google-play-scraper in C:\Users\RONYFA~1\On
eDrive\DOCUME~1\VIRTUA~1\ENV-SC~1\Lib\site-packages (1.2.7)
> # Import dengan convert = FALSE memblokir error Attribute 'value' dari P
ython
> scraper_raw <- import("google_play_scraper", convert = FALSE)
> # Import dengan convert = FALSE memblokir error Attribute 'value' dari P
ython
> scraper_raw <- import("google_play_scraper", convert = FALSE)
> play_scraper <- import("google_play_scraper")
> app_id <- "com.adakami.dana.kredit.pinjaman"
> jumlah_tarikan<-500L
> hasil_b1 <- play_scraper$reviews(app_id, lang='id', country='id', sort=s
craper_raw$Sort$NEWEST, count=jumlah_tarikan, filter_score_with=1L)
> hasil_b2 <- play_scraper$reviews(app_id, lang='id', country='id', sort=s
craper_raw$Sort$NEWEST, count=jumlah_tarikan, filter_score_with=2L)
> df_keluhan <- bind_rows(hasil_b1[[1]], hasil_b2[[1]]) %>% arrange(desc(a
t))
```

Gambar 1. Tahapan *Scraping*

Tahapan awal adalah tahapan pengumpulan data. Dalam penelitian ini dilaksanakan melalui teknik *web scraping* secara komputasional. Proses ini mengekstraksi data tekstual ulasan pengguna secara langsung dari repositori Google Play Store menggunakan integrasi antara antarmuka bahasa pemrograman R dan pustaka Python. Dengan teknik ini pengumpulan

data secara online menjadi lebih efisien dan efektif (Pratama et al, 2021). Adapun rincian metodologis tahapan awal ini sebagai berikut.

1. Isolasi lingkungan komputasi. Guna memastikan stabilitas eksekusi algoritma dan mencegah konflik dependensi antar-paket pada sistem operasi dilakukan konfigurasi lingkungan virtual (*virtual environment*) dengan nomenklatur *env-scraping*. Modul utama yang diinstalasi pada ruang lingkup ini adalah pustaka *scraper* fungsional. Pendekatan ini menggaransi bahwa proses penarikan data komputasional beroperasi secara terisolasi tanpa mengintervensi ruang memori global (*global environment*),
2. Integrasi lintas bahasa dan proteksi data. Penghubungan fungsionalitas antara lingkungan kerja R dan modul Python diakomodasi oleh paket reticulate. Untuk memitigasi anomali sistemik berupa *type-casting error*—seperti hilangnya atribut nilai pada objek enumerasi (*Enum*) Python saat dibaca oleh R—proses impor modul dieksekusi dengan mendeklarasikan argumen *convert = FALSE*. Konfigurasi struktural ini berfungsi mempertahankan integritas objek asal secara absolut, sehingga parameter metode pengurutan (*Sort$NEWEST*) dapat diinterpretasikan secara presisi oleh arsitektur mesin pengunduh,
3. Penetapan parameter populasi dan sampel. Target observasi diidentifikasi secara spesifik menggunakan penanda unik paket aplikasi (*Application ID*), yakni *com.adakami.dana.kredit.pinjaman*. Penarikan data kuantitatif dibatasi dengan kuota tetap (*fixed quota*) sebesar 500 observasi per metrik kepuasan. Angka ini dideklarasikan melalui sintaks 500L guna mendikte mesin kompilator bahwa nilai tersebut merupakan format *integer* murni sehingga mencegah terjadinya galat presisi (*floating-point error*) pada pemrosesan latar belakang,
4. Pengumpulan korpus data dieksekusi dalam dua iterasi penarikan terpisah untuk mengamankan validitas populasi keluhan pengguna. Iterasi pertama difokuskan pada ekstraksi parsial 500 ulasan mutakhir dengan filter absolut pada metrik nilai bintang satu (*filter_score_with = 1L*) sedangkan iterasi kedua difokuskan pada ekstraksi parsial 500 ulasan mutakhir dengan filter absolut pada metrik nilai bintang satu (*filter_score_with = 2L*). Seluruh penarikan data tersebut diurutkan secara sekuensial berdasarkan kronologi waktu publikasi terbaru, dan
5. Transformasi dan penggabungan data frame, data *wragling*. Luaran data mentah (*raw data*) dari dua proses iterasi ekstraksi yang berstruktur *list* direduksi dan ditransformasikan menjadi satu kesatuan tabel data matriks menggunakan fungsi bind_rows. Korpus yang memuat agregasi 1.000 observasi dari kedua kelompok *rating* tersebut kemudian diurutkan ulang secara menurun (*descending sort*) berdasarkan variabel temporal (at). Struktur data tabular final ini secara otomatis dialokasikan ke dalam memori komputasi sebagai *dataset* primer yang siap diteruskan ke fase prapemrosesan teks (*text preprocessing*).

```
# ========================================
> # FASE 2: PREPROCESSING (ANTI TIMEOUT & ENCODING ERROR)
> # ========================================
> cat("[2/4] Membersihkan teks dan melakukan stemming...\n")
[2/4] Membersihkan teks dan melakukan stemming...
> # ========================================
> # FASE 2: PREPROCESSING (ANTI TIMEOUT & ENCODING ERROR)
> # ========================================
> cat("[2/4] Membersihkan teks dan melakukan stemming...\n")
[2/4] Membersihkan teks dan melakukan stemming...
> korpus <- corpus(df_keluhan$content)
> # Sintaks quanteda modern (tanpa tolower di dalam tokens)
> token_awal <- tokens(korpus, remove_punct = TRUE, remove_numbers = TRUE,
remove_url = TRUE, remove_symbols = TRUE)
> # Sintaks quanteda modern (tanpa tolower di dalam tokens)
> token_awal <- tokens(korpus, remove_punct = TRUE, remove_numbers = TRUE,
remove_url = TRUE, remove_symbols = TRUE)
> token_bersih <- tokens_tolower(token_awal)
> stopword_id <- stopwords::stopwords("id", source = "stopwords-iso")
> token_tanpa_stopword <- tokens_remove(token_bersih, pattern = stopword_i
d)
> # Bypass error memory/iconv dengan standarisasi encoding UTF-8
> kata_unik <- types(token_tanpa_stopword)
> # Bypass error memory/iconv dengan standarisasi encoding UTF-8
> kata_unik <- types(token_tanpa_stopword)
> kata_unik <- iconv(kata_unik, from = "UTF-8", to = "UTF-8", sub = "")
> # Manipulasi types agar stemming sangat ringan untuk RAM
> kata_dasar <- sapply(kata_unik, katadasaR, USE.NAMES = FALSE)
> # Manipulasi types agar stemming sangat ringan untuk RAM
> kata_dasar <- sapply(kata_unik, katadasaR, USE.NAMES = FALSE)
> token_stemmed <- tokens_replace(token_tanpa_stopword, pattern = kata_uni
k, replacement = kata_dasar)
```

Gambar 2. Tahapan *Prepocessing*

Tahapan kedua, *processing*. Tahapan yang paling penting ini menurut Ratnaningsih & Hakim (2025) disebut juga dengan tahapan normalisasi yaitu tahapan dengan menghilangkan bagian-bagian yang tidak relevan. Adapun rincian metodologis tahapan kedua ini sebagai berikut.

1. Pembentukan korpus dan pembersihan derau (noise removal). Teks mentah dari hasil *scrapping* dikonversi menjadi objek korpus (struktur data khusus analisis linguistik). Setelah itu, kalimat dipecah menjadi unit kata tunggal (*tokenization*) sembari menghapus tanda baca, angka, simbol, dan tautan URL. ulasan Play Store sangat berantakan (penuh dengan *emoticon*, salah ketik, atau *spam link*). Mesin tidak bisa memahami sentimen dari angka atau simbol. Tahapan ini membuang semua "sampah" digital tersebut agar algoritma murni hanya berfokus pada analisis bahasa manusia.
2. Penyeragaman teks (*Case Folding*). Menggunakan sintaks tokens_tolower, seluruh huruf kapital dalam dokumen ditransformasikan menjadi huruf kecil secara absolut. Komputer membaca karakter secara kaku. Kata **"Kecewa"**, **"KECEWA"**, dan **"kecewa"** akan dianggap sebagai tiga entitas data yang berbeda karena perbedaan huruf kapital. Penyeragaman ini memastikan ketiganya dikelompokkan dan dihitung sebagai satu keluhan yang sama,
3. Pembuangan kata nir-makna (*Stopword Removal*). Mengaplikasikan leksikon (*dictionary*) bahasa Indonesia untuk mengeliminasi kata hubung, preposisi, dan kata ganti (misal: *dan, di, ke, yang, ini, itu*). Kata-kata tersebut mendominasi frekuensi kemunculan dalam kalimat, tetapi tidak memiliki nilai emosional atau informasi (*noise words*). Dengan membuangnya, data menjadi jauh lebih ramping sehingga mesin bisa langsung menyoroti kata kunci kritis (seperti *gagal, tolak, error)*.
4. Standarisasi encoding (*Encoding Bypass*). Penggunaan fungsi *iconv* untuk memaksa standarisasi karakter teks ke dalam format murni UTF-8, serta menyapu bersih sisa karakter residu (*sub = ""*). ni adalah benteng pertahanan krusial dalam sistem pemrosesan. Pengguna ponsel cerdas sering memakai *keyboard* dengan aksara atau *emoji* aneh yang tidak dikenali sistem statistik. Karakter inilah yang sering membuat sistem R *crash*, *hang*, atau memicu

batas waktu (*timeout*). Fungsi ini menyederhanakan karakter tersebut agar proses komputasi tidak terputus di tengah jalan.

5. Rekayasa reduksi imbuhan (*Optimized Stemming*). Sistem mengekstraksi daftar kosakata unik (*types*) terlebih dahulu. Pemotongan imbuhan (mengembalikan kata seperti *menyusahkan* menjadi *susah*) hanya dilakukan pada daftar kosakata unik tersebut, lalu ditimpa kembali ke teks asalnya (*tokens_replace*). Jika metode *stemming* konvensional diterapkan pada ribuan baris data, perangkat lunak akan memakan waktu berjam-jam hingga RAM komputer penuh. Dengan metode manipulasi kamus unik ini, algoritma bekerja secara eksponensial lebih efisien, memangkas waktu pemrosesan dari ukuran jam menjadi hanya hitungan detik.

```
# ========================================
> # FASE 3: KALKULASI TF-IDF & BIGRAM
> # ========================================
> cat("[3/4] Menghitung matriks TF-IDF...\n")
[3/4] Menghitung matriks TF-IDF...
> # ========================================
> # FASE 3: KALKULASI TF-IDF & BIGRAM
> # ========================================
> cat("[3/4] Menghitung matriks TF-IDF...\n")
[3/4] Menghitung matriks TF-IDF...
> token_bigram <- tokens_ngrams(token_stemmed, n = 2, concatenator = " ")
> dfm_bigram <- dfm(token_bigram)
> # Pembobotan proporsional
> dfm_tfidf_bobot <- dfm_tfidf(dfm_bigram, scheme_tf = "prop", scheme_df =
"inverse")
> # Pembobotan proporsional
> dfm_tfidf_bobot <- dfm_tfidf(dfm_bigram, scheme_tf = "prop", scheme_df =
"inverse")
> # Ekstraksi 15 nilai keluhan dominan
> tabel_anomali <- textstat_frequency(dfm_tfidf_bobot, force = TRUE) %>%
+   rename(Frasa_Bigram = feature, Bobot_Kritis = frequency) %>%
+   arrange(desc(Bobot_Kritis)) %>%
+   head(15)
> # Ekstraksi 15 nilai keluhan dominan
> tabel_anomali <- textstat_frequency(dfm_tfidf_bobot, force = TRUE) %>%
+   rename(Frasa_Bigram = feature, Bobot_Kritis = frequency) %>%
+   arrange(desc(Bobot_Kritis)) %>%
+   head(15)
```

Gambar 3. Tahapan Kalkulasi TF-IDF dan Bigram

Tahapan komputasi pada tahapan ketiga, kalkulasi TF-IDF & Bigram, merepresentasikan inti dari algoritma *text mining*, data linguistik kualitatif ditransformasikan secara matematis menjadi ukuran statistik kuantitatif. Menurut Qaiser & Ali (2008) bahwa TF-IDF sebagai jantung evaluasi statistikal dalam algoritma *Text Mining*, mengonversi teks mentah kualitatif yang tak terhitung (*uncountable*) menjadi metrik frekuensi. Adapun rincian metodologis tahapan kedua ini sebagai berikut.

1. Pembentukan asosiasi konteks (*Bigram Tokenization*). Eksekusi fungsi *tokens_ngrams(n = 2)* mensintesis kata-kata tunggal (*unigram*) menjadi urutan dua kata yang saling berdampingan secara sekuensial (*bigram*), disatukan dengan spasi pemisah (concatenator = " "). Menganalisis kata tunggal sangat rawan mendistorsi makna asli keluhan. Pemecahan kata tunggal "tidak" dan "bisa" akan menghasilkan interpretasi bias. Pemodelan *bigram* mengamankan konteks klausa tersebut agar tetap utuh dan relevan, memunculkan frasa nyata seperti "tidak bisa", "aplikasi error", atau "pinjam tolak".
2. Kontruksi ruang vektor. Sintaks *dfm()* mengonversi kumpulan *bigram* tekstual tersebut ke dalam struktur matriks aljabar dua dimensi yang bersifat *sparse* (jarang)Sistem komputasi tidak bisa menghitung alfabet. Konversi DFM berfungsi memetakan teks ke dalam format koordinat angka (vektor matriks), di mana setiap ulasan pengguna diubah menjadi representasi baris, dan setiap entitas frasa *bigram* menjadi kolom fitur. Ini adalah jembatan mutlak sebelum masuk ke perhitungan statistik.
3. Kalkulasi bobot signifikasi (*Proportional TF-IDF Calculation*). Fungsi *dfm_tfidf* mengkalkulasi matriks bobot menggunakan parameter spesifik. Pengaturan *scheme_tf = "prop"* melakukan normalisasi panjang teks secara proporsional (mencegah ulasan panjang mendominasi skor), sedangkan *scheme_df = "inverse"* menekan secara algoritmik bobot frasa yang muncul terlalu umum di seluruh populasi dokumen. Jika hanya menghitung

jumlah kata (frekuensi), frasa "aplikasi ini" mungkin mendominasi meskipun tidak berarti apa-apa. Formulasi statistik TF-IDF bekerja secara akurat untuk meredam *noise* tersebut dan menaikkan bobot frasa spesifik yang mengindikasikan kerusakan sistem (seperti "data hapus" atau "bunga tinggi").

4. Isolasi anomali prioritas (*Feature Ranking and Extraction*). Rangkaian metode *piping* (%>%) mengeksekusi ekstraksi nilai frekuensi statistik (*textstat_frequency*), memetakan ulang labelnya (rename), mengurutkan matriks secara menurun berdasarkan bobot tertinggi (*arrange(desc)*), dan memangkas limit data absolut pada 15 observasi teratas (*head(15)*). Langkah ini adalah produk akhir dari otomatisasi audit. Ketimbang mengharuskan tim analis membaca ribuan ulasan secara manual, skrip secara otomatis menyaring seluruh dimensi *database* dan mengumpukannya menjadi 15 akar masalah (prioritas utama) yang memiliki impak keluhan paling destruktif.

```
# ========================================
> # FASE 4: RENDER VISUALISASI GRAFIK
> # ========================================
> cat("[4/4] Membangun visualisasi akhir...\n")
[4/4] Membangun visualisasi akhir...
> tabel_anomali$Frasa_Bigram <- factor(tabel_anomali$Frasa_Bigram, levels
= rev(tabel_anomali$Frasa_Bigram))
> plot_akhir <- ggplot(tabel_anomali, aes(x = Frasa_Bigram, y = Bobot_Krit
is)) +
+   geom_segment(aes(x = Frasa_Bigram, xend = Frasa_Bigram, y = 0, yend =
Bobot_Kritis), color = "gray") +
+   geom_point(size = 4, color = "firebrick") +
+   coord_flip() +
+   labs(
+     title = "Deteksi Anomali Sistemik: Top Keluhan Pengguna",
+     subtitle = "Fokus Metrik Rating 1 & 2 (Analisis TF-IDF Bigram)",
+     x = "Frasa Pengaduan (Bigram)",
+     y = "Bobot Kritis (TF-IDF)"
+   ) +
+   theme_minimal() +
+   theme(plot.title = element_text(face = "bold"))

> print(plot_akhir)
> cat("SELESAI! Grafik berhasil ditampilkan.\n")
SELESAI! Grafik berhasil ditampilkan.
```

Gambar 4. Tahapan Visualisasi Grafik

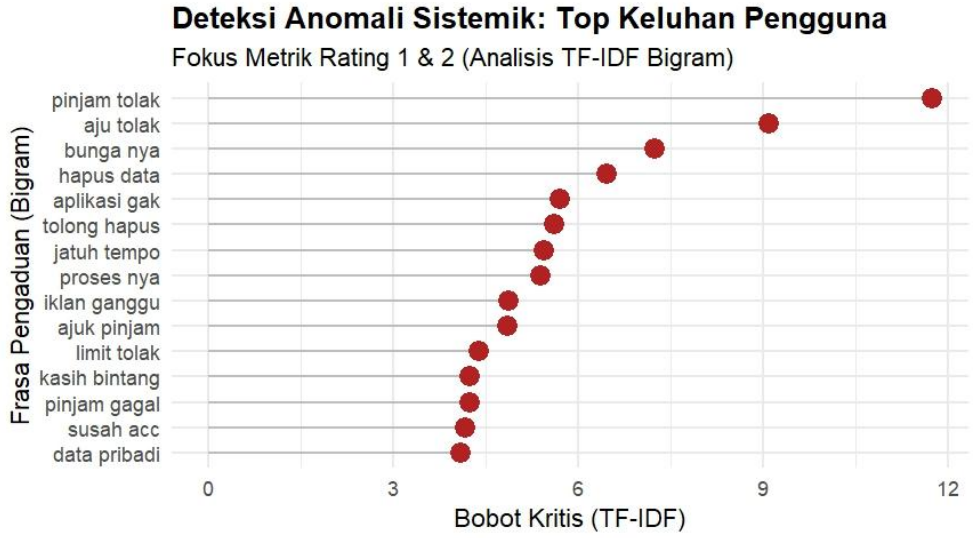
Tahapan komputasi sebagai proses yang terakhir ini merupakan proses transformasi pemetaan visual (*Data Visualization Rendering*). Menurut Borner et al. (2019) bahwa secara algoritmik, matriks data kuantitatif yang masih berupa angka statistik diubah menjadi representasi grafis guna mengekstraksi *insight* strategis untuk pengambilan keputusan operasional. Adapun rincian metodologis tahapan terakhir ini sebagai berikut.

1. Penguncian sumbu faktorial (*Factor Re-Levelling*). Pengeksekusian *factor(..., levels = rev(...))* merekayasa ulang struktur level dari variabel kategorikal (*Frasa_Bigram*) secara terbalik (*reverse*). Tanpa perintah ini, *R* secara bawaan (*default*) akan mengurutkan keluhan berdasarkan abjad (A hingga Z). Intervensi *re-leveling* ini memaksa grafik untuk secara presisi menampilkan bobot anomali paling kritis di posisi paling atas. Hal ini memandu auditor dan pihak manajemen untuk langsung menyoroti masalah paling mendesak pada detik pertama mereka melihat laporan.
2. Kontruksi *Lollipop Chart* (*Spatial Geometry Mapping*). Pemetaan arsitektur grafis menggunakan kombinasi geom_segment (garis interval abu-abu) dan geom_point (titik plot merah) di dalam ekosistem ggplot2. Pemilihan *Lollipop Chart* digunakan untuk mengoptimalkan *Data-Ink Ratio*. Dibandingkan menggunakan diagram batang (*Bar Chart*) berbentuk balok-balok tebal yang membuat laporan terlihat *cluttered* (Abdul et al., 2020). Garis dan titik pada grafik ini meminimalisasi beban kognitif pembaca saat membandingkan belasan variabel secara bersamaan sehingga visualisasi terlihat jauh lebih jelas dan profesional.
3. Inversi sumbu kartesian (*Coordinate Flipping*). Implementasi fungsi *coord_flip()* mentransformasi bidang kartesian dengan menukar posisi sumbu X dan sumbu Y. Ini adalah

parameter mutlak dalam visualisasi *text mining*. Label teks (seperti "pinjam tolak" atau "iklan ganggu") membutuhkan ruang horizontal. Jika dibiarkan di sumbu X standar (bawah), teks tersebut akan saling menindih dan mustahil dibaca. Dengan memutarnya ke sumbu Y (vertikal), label teks mendapat ruang baca alami (dari kiri ke kanan) secara sempurna tanpa pemotongan kata.

4. Standarisasi laporan korporat (*Theming and Annotation*). Integrasi metainformasi kontekstual melalui labs() dipadukan dengan fungsi theme_minimal() untuk memangkas kebisingan visual (*visual noise*). Hasil akhirnya berupa grafik indikator masalah yang bersih, lugas, murni berfokus pada titik anomali, dan langsung siap disisipkan ke dalam dokumen *Executive Summary* tanpa memerlukan penyuntingan tambahan.

Keluaran akhir mulai dari tahapan *scrapping* hingga visualisasi disajikan dalam bigram sebagai berikut.



Gambar 5. TF-IDF Bigram

Berdasarkan visualisasi grafik *Lollipop Chart* TF-IDF tersebut, dapat dipetakan keluhan pengguna secara presisi dan terukur sebagai berikut.

1. Klaster pertama, friksi penolakan kredit, terdapat lima frase yaitu *pinjam tolak* (skor tertinggi = 11.8, *aju tolak* = 9.1, *limit tolak*, *pinjam gagal*, dan *susah acc* masing-masing mendekati = 3. Dominasi mutlak di puncak grafik menunjukkan bahwa pemicu utama *rating* buruk adalah tingginya angka penolakan pengajuan pinjaman (*rejection rate*). Terdapat indikasi bahwa kampanye pemasaran (yang berhasil mengundang banyak pengguna untuk mengunduh) tidak sejalan dengan kriteria *Credit Scoring* yang ditetapkan oleh sistem *Risk Management*. Pengguna merasa proses persetujuannya sangat dipersulit ("susah acc") atau ditolak tanpa transparansi alasan yang jelas.
2. Klaster kedua, krisis kepercayaan privasi data, terdapat tiga frase yaitu *hapus data* = 6.3, *tolong hapus* = 5.7, dan *data pribadi* = 4.0. Hal ini adalah efek domino (*secondary impact*) dari klaster pertama. Pengguna yang pengajuannya ditolak mengalami kepanikan dan menuntut penghapusan rekam jejak KTP atau data diri mereka dari *database* aplikasi. Tingginya kemunculan frasa ini menandakan adanya defisit kepercayaan (*trust deficit*). Pengguna memiliki ketakutan yang nyata bahwa data pribadi mereka akan disalahgunakan atau disebar (gagal bayar/pinjol ilegal) meskipun status pinjaman mereka tidak disetujui.
3. Klaster ketiga, anomali finansial dan penagihan, terdapat dua frase yaitu *bunga nya* = 7.2 dan *jatuh tempo* = 5.4. Bagi segmen pengguna yang pinjamannya berhasil dicairkan, beban finansial menjadi *pain point* utama. Keluhan "bunga nya" mengindikasikan ketidakpuasan terhadap besaran suku bunga yang mungkin dianggap memberatkan atau adanya biaya

tersembunyi (*hidden fees*) yang tidak transparan di awal. Sementara itu, "jatuh tempo" mengindikasikan adanya masalah dalam siklus penagihan, seperti notifikasi penagihan dari *desk collection* yang terlalu agresif sebelum waktunya.

4. Klaster keempat, kerusakan teknis UX/UI, terdapat tiga frase yaitu *aplikasi gak* = 5.8, *proses nya* = 5.4, dan *iklan ganggu* = 4.8. Infrastruktur aplikasi itu sendiri memiliki kendala teknis yang merusak pengalaman pengguna (*User Experience*). "Aplikasi gak" merujuk pada *bug* sistem (seperti aplikasi tidak bisa dibuka/keluar sendiri). "Proses nya" menunjukkan adanya jeda waktu (*latency*) yang lamban saat proses verifikasi. Ironisnya, untuk sebuah platform finansial, keberadaan "iklan ganggu" menjadi anomali fatal karena memicu rasa frustrasi dan menjatuhkan kredibilitas/profesionalitas aplikasi di mata nasabah.

4 KESIMPULAN

Pengkonstruksian model ekstraksi informasi berbasis *Natural Language Processing* (NLP) menggunakan bahasa R guna memvisualisasikan profil ulasan pengguna, fokus kepada bintan 1 dan 2, pada aplikasi AdaKami melalui teknik *web scraping* ulasan di Google Play Store. Teknik ini dilakukan melalui empat tahapan yaitu scrapping, preposising, kalkulasi TF-IDF & Bigram, dan visualisasi grafik. Berdasarkan algoritma tahapan tersebut dapat disimpulkan empat klaster ulasan yaitu (1) Klaster pertama, friksi penolakan kredit, (2) Klaster kedua, krisis kepercayaan privasi data, (3) Klaster ketiga, anomali finansial dan penagihan, dan (4) Klaster keempat, kerusakan teknis UX/UI.

Berdasarkan hal tersebut maka sebaiknya pihak developer AdaKami dan OJK sebagai lembaga pengawasan keuangan Indonesia memperhatikan ketiga hal berikut.

1. Transparansi Status yaitu perbaikan antarmuka (UI) penolakan dengan memberikan alasan spesifik (misal: "Skor kredit belum cukup" alih-alih hanya "Ditolak") untuk meredam frustrasi.
2. SOP Penghapusan Data, Penyediaan fitur "Hapus Akun & Data" yang mudah diakses dan dilakukan secara transparan (sesuai regulasi OJK/PDP) untuk mengembalikan rasa aman pengguna.
3. Audit Infrastruktur, pengurangan intensitas iklan (*ads*) di dalam aplikasi dan iklan di media sosial yang cenderung persuasif-masif. Selain itu perbaikan *bug* teknis yang menghambat proses operasional juga perlu dilakukan.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Ibu Ida Mujathidah, M.Si, pengampu tutorial online mata kulia STMA 4313, Pemodelan dan Visualisasi Data 4, masa ujian 2025/2026 Genap yang sangat menginspirasi ide penyusunan tulisan ini. Terima kasih juga disampaikan kepada ketua Jurusan S1 Matematika Fakultas Sains dan Teknologi Universitas Terbuka, Ibu Elin Herlinawati, M.Si dan Ibu Tri Wijayanti Septiarini, M.Si.

DAFTAR PUSTAKA

Abdul, AshrAf., Weth, Christian von der Weth., Kankanhalli, Mohan Kankanhalli., Lim, Brian Y. 2020. COGAM: Measuring and Moderating Cognitive Load inMachine Learning Model Explanations. *CHI '20: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI 2020, April 25–30, 2020, Honolulu, HI, USA.* https://dl.acm.org/doi/epdf/10.1145/3313831.3376615

Börner, K., Andreas, Bueckle., Ginda, Michael. 2019. "Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments". *Proceedings of the National Academy of Sciences (PNAS)*. https://www.pnas.org/doi/10.1073/pnas.1807180116

Bennoit, Kenneth., Watanabe, Kohei., Wang, Haiyan., Nulty, Paul., Obeng, Adam., Müller, Stefan., Matsuo, Akitaka. 2018. quanteda: An R package for the quantitative analysis of

textual data. *Journal of Open Source Software*, *3*(30), 774. https://doi.org/10.21105/joss.00774

De Lima, V.M.A., Barbosa, J.R. & Marcacini, R.M. Issue detection and prioritization based on mobile application reviews. *Software Qual J **33**, 6* (2025). https://doi.org/10.1007/s11219-024-09703-2

Handayanto, Rahmadya Trias., Herlawati., Atika, Prima Dina., Khasanah, Fata Nidaul., Yusuf, Ajif Yunizar Pratama., Septia, Dwi Yoga. 2021. Analisis Sentimen Pada Situs Google Review dengan Naïve Bayes dan Support Vector Machine. *Journal Komtika (Komunikasi dan Informatika)* Vol. 5 No. 2, Nov 2021. https://doi.org/10.31603/komtika.v5i2.6280

Jurafsky, D., Martin, J. H. 2024. *Speech and Language Processing* (3rd ed. Draft). web.stanford.edu/~jurafsky/slp3/

Lubis, Fauzi Arif. 2023. Analysis of User Review on The Use of Fintech Dana Sariah Internat onal. Journal of Science and Society, 5(2):70-79. https://doi.org/10.54783/ijsoc.v5i2.671

Nasution, Arbi Haza., Monika, Winda. 2023. *Ilmu Data*. https://repository.uir.ac.id/27803/1/Ilmu%20Data.pdf

Pratama, Dionis Balan., Sofwan, Aghus., Soetrisno, Yosua Alvin Adi. 2021. Implementasi Te knik Web Scrapping dan Fitur Data Internal Pada Sistem Informasi Dosen Penelitian da n Pengabdian Dosen Fakultas Teknik Universitas Diponegoro. Jurnal Transient, 10(2), J uni 2021. https://doi.org/10.14710/transient.v10i2.284-291

Qaiser, Shahzad., Ali, Ramsha. 2018. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *Internasional Journal of COmputer Applications*, 181(1):25-29 . https://doi.org/10.5120/ijca2018917395

Ratnaningsih, Dewi Juliah, & Hakim, R. Bagus Fajriya. (2025). *BMP MSIM4310/2 SKS: Analsis dan Visualisasi Data (BMP)*. Tangerang Selatan: Universitas Terbuka

Ratuwalu, Richard., Komsatun., Dewayani, Sanny. 2026. Perlindungan Hukum Konsumen Fintech P2P Lending Akibat Penagihan oleh Pihak Ketiga Sesuai POJK No. 22 Perlindungan Konsumen. *Jurnal Riset Rumpun Ilmu Sosial Politik dan Humaniora*, 5(1):554-568. https://doi.org/10.55606/jurrish.v5i1.6918

Respati, Agustinus Rangga., Arief, Teuku Muhammad Valdy. 2025. 100 Hari Prabowo-Gibran, di Balik Perubahan Nama Pinjol Jadi Pindar. Kompas. https://money.kompas.com/read/2025/01/21/135037726/100-hari-prabowo-gibran-di-balik-perubahan-nama-pinjol-jadi-pindar

Setiawan, Sakinah Rakhmah Diah. 2026. Laba Industri Pindar Bangkit, Pendanaan Bank Tetap Dominan. *Kompas*. https://money.kompas.com/read/2026/06/08/181800226/laba-industri-pindar-bangkit-pendanaan-bank-tetap-dominan?page=1

Siregar, Dania., Ladayya, Faroh., Albaqi, Naufal Zhafran., Wardana, Bintang Mahesa. 2023. Penerapan Metode Support Vector Machines (SVM) dan Metode Naïve Bayes Classifier (NBC) dalam Analisis Sentimen Publik terhadap Konsep Child-free di Media Sosial Twitter. Jurnal Statistika dan Aplikasinya, 7(1):93-104, Juni 2023. https://doi.org/10.21009/JSA.07109