PENGEMBANGAN SISTEM INFORMASI KEPEGAWAIAN UHO MODUL TENAGA KEPENDIDIKAN DENGAN ALGORITMA *B+ TREE* DAN *FULL-TEXT SEARCH*

Putri Eka Wulandari Alam. A[1], Andi Tenriawaru[*1], La Surimi[1]
[1]Program Studi Ilmu Komputer, Fakultas MIPA, Universitas Halu Oleo, Kendari, Indonesia
Penulis korespondensi: Andi Tenriawaru (andi.tenriawaru@uho.ac.id)

Pengelolaan data kepegawaian tenaga kependidikan di Bagian Kepegawaian Rektorat Universitas Halu Oleo masih mengandalkan lembar kerja dan arsip fisik, sehingga penelusuran data berjalan lambat dan rawan inkonsistensi. Penelitian ini membangun sistem informasi kepegawaian berbasis web yang memadukan algoritma B+ Tree untuk pencarian data pegawai terstruktur berdasarkan nomor induk dan nama, serta Full-Text Search untuk penelusuran isi dokumen administrasi hasil pengenalan karakter optik. Sistem dikembangkan dengan metode Waterfall menggunakan Python (Django pada lapisan antarmuka dan FastAPI pada lapisan layanan) serta MySQL. Pengujian fungsional black box terhadap 38 skenario pada tiga tingkat hak akses menunjukkan seluruh fitur berjalan valid. Pencarian B+ Tree menghasilkan precision 100% dan recall rata-rata 89,55% dengan rata-rata waktu respons end-to-end 3,338 detik, sedangkan Full-Text Search berbasis pencocokan substring pada teks hasil ekstraksi menjalankan seluruh sepuluh skenario tanpa galat dengan rata-rata waktu pemrosesan 0,0914 detik dan tingkat keberhasilan ekstraksi teks 96,88%. Pengujian penerimaan pengguna menghasilkan skor rata-rata 4,09 pada kategori Setuju, sehingga sistem dinilai layak menggantikan pengelolaan data secara manual.

Keywords - B+ Tree, Full-Text Search, Kepegawaian, Sistem Informasi, Tenaga Kependidikan.

I. PENDAHULUAN

Pemanfaatan teknologi informasi di perguruan tinggi telah bergeser dari sekadar alat bantu operasional menjadi infrastruktur penting dalam manajemen data. Tantangan utamanya kini bukan lagi pada penyimpanan data (*storage*), melainkan pada kemampuan menemukan kembali data (*retrieval*) secara cepat dan akurat. Teknologi informasi terbukti berkontribusi terhadap kinerja organisasi, terutama dalam mempercepat proses bisnis dan menjamin ketepatan pengambilan keputusan berbasis data [1]. Sistem informasi sendiri merupakan kombinasi perangkat lunak, perangkat keras, dan jaringan yang digunakan untuk mengumpulkan, mengelola, dan mendistribusikan data guna mendukung proses bisnis dan pengambilan keputusan organisasi [2]. Universitas Halu Oleo (UHO) sebagai institusi dengan aktivitas akademik yang padat membutuhkan tata kelola data sumber daya manusia yang terstruktur untuk mendukung efisiensi tersebut.

Observasi pada Bagian Kepegawaian Rektorat UHO menunjukkan bahwa pengelolaan data tenaga kependidikan masih didominasi aplikasi pengolah angka (*spreadsheet*) dan arsip fisik. Dari sudut pandang basis data, cara ini memiliki kelemahan berupa isolasi data, redundansi yang tinggi, dan risiko kesalahan manusia. Sistem informasi manajemen seharusnya berperan sebagai subsistem terintegrasi yang mengolah data menjadi informasi akurat sehingga kesalahan pengelolaan konvensional dapat ditekan [3]. Tidak adanya batasan integritas relasional pada pengelolaan berbasis berkas membuat data kepegawaian rentan terhadap anomali penyisipan, pembaruan, dan penghapusan yang berujung pada inkonsistensi informasi. Kebutuhan akan penelusuran yang cepat juga makin terasa pada proses administrasi seperti kenaikan pangkat, yang menuntut operator memeriksa surat keputusan (SK) pangkat terakhir, SK jabatan, ijazah, dan dokumen pendukung lainnya dari tumpukan arsip.

Permasalahan tersebut makin kompleks seiring pertumbuhan volume data. Laporan Kinerja UHO Tahun 2024 mencatat jumlah sumber daya manusia sebanyak 2.274 orang, termasuk 575 tenaga kependidikan yang terdiri atas 379 PNS dan 196 non-PNS [4]. Pencarian linear (*sequential search*) memiliki kompleksitas waktu terburuk $O(n)$, sehingga waktu pencarian meningkat sebanding dengan jumlah data, padahal kegagalan sistem dalam memberikan respons cepat berdampak langsung pada turunnya kepuasan pengguna [5]. Penelitian sistem informasi manajemen kepegawaian (SIMPEG) sebelumnya lebih banyak menitikberatkan pemenuhan fungsi administrasi, seperti pemberkasan terpusat, otomatisasi data karyawan, dan presensi yang dapat diakses melalui perangkat bergerak [6], namun belum banyak membahas efisiensi penelusuran data.

Struktur data *B+ Tree* dan teknik *Full-Text Search* merupakan dua pendekatan yang relevan untuk masalah tersebut. Penerapan indeks *B+ Tree* pada basis data relasional dilaporkan mampu mempercepat akses 4 sampai 5 kali dibandingkan akses sekuensial karena struktur pohonnya seimbang dan meminimalkan operasi

disk I/O [7], serta menurunkan kompleksitas pencarian menjadi O(log *n*) pada pengelolaan data numerik sekolah [8]. Pencarian dengan *B/B+ Tree* juga lebih cepat dan stabil dibandingkan *table scan* karena mengurangi jumlah akses disk secara signifikan [9], dan pengembangannya dalam bentuk B+ Tree terdistribusi mampu mempercepat pengindeksan serta kueri rentang pada data *streaming* [10].

Pada sisi pencarian teks, penggunaan *full-text index* pada MySQL terbukti menghasilkan waktu pencarian yang lebih cepat dan hasil yang lebih relevan dibandingkan pencarian tanpa indeks [11]. Perbandingan kinerja berbagai platform menunjukkan bahwa mesin pencari khusus seperti *ElasticSearch* dan *Apache Solr* paling stabil untuk data teks masif, sedangkan basis data relasional tetap kompetitif untuk skala data menengah [12]. Penelitian lain menunjukkan bahwa pengindeksan asinkron dengan *ElasticSearch* meningkatkan kecepatan pencarian tanpa mengganggu kinerja sistem pengelola konten [13], pencarian teks paralel tanpa indeks menghasilkan *throughput* tinggi dengan beban penyimpanan indeks yang lebih kecil [14], dan peringkasan teks dapat dimanfaatkan untuk memperkecil ukuran indeks pada data berskala besar [15]. Algoritma pencocokan string klasik seperti Knuth-Morris-Pratt, Boyer-Moore, dan Rabin-Karp juga memiliki keunggulan dan kelemahan masing-masing bergantung pada kondisi data [16].

Sebagian besar penelitian tersebut menguji algoritma secara terpisah pada lingkungan eksperimen, bukan di dalam sistem informasi kepegawaian yang operasional. Padahal, data kepegawaian terdiri atas dua jenis yang berbeda sifatnya, yaitu atribut terstruktur seperti Nomor Induk Pegawai (NIP) dan nama, serta isi dokumen hasil pindaian seperti SK dan ijazah. Oleh karena itu, penelitian ini bertujuan membangun Sistem Informasi Kepegawaian UHO modul tenaga kependidikan berbasis web yang mengintegrasikan *B+ Tree* untuk pencarian data pegawai terstruktur dan *Full-Text Search* berbantuan pengenalan karakter optik untuk penelusuran isi dokumen, serta menganalisis precision, recall, dan waktu respons kedua algoritma tersebut. Pengembangan sistem diselaraskan dengan Undang-Undang No. 20 Tahun 2023 tentang Aparatur Sipil Negara dan Peraturan Presiden No. 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik, dengan penerapan hak akses berjenjang (*role-based access control*) untuk menjaga kerahasiaan data.

II. METODE PENELITIAN

A. Waktu, Tempat, dan Jenis Penelitian

Penelitian dilaksanakan pada Oktober 2025 hingga Januari 2026 di Bagian Kepegawaian Biro Keuangan dan Umum, Rektorat Universitas Halu Oleo. Lokasi ini dipilih karena peneliti membutuhkan akses langsung terhadap data kepegawaian untuk menjamin ketepatan analisis kebutuhan. Penelitian ini merupakan penelitian terapan (*applied research*) dengan pendekatan deskriptif untuk menggambarkan kebutuhan pengguna dan proses bisnis yang berjalan, sekaligus penelitian pengembangan sistem (*research and development*) karena mencakup perancangan, implementasi, dan pengujian sistem informasi. Instrumen penelitian meliputi laptop dengan memori 8 GB, sistem operasi Windows 11, Visual Studio Code, peramban Chrome dan Microsoft Edge, Python, Django, MySQL, Draw.io, serta Figma untuk perancangan antarmuka.

B. Prosedur dan Pengumpulan Data

Penelitian dilaksanakan dalam delapan tahap, yaitu studi pendahuluan, studi literatur, analisis kebutuhan, perancangan sistem, pengembangan sistem, pengujian sistem, evaluasi dan revisi, serta penyusunan laporan. Studi pendahuluan dilakukan untuk memahami proses bisnis kepegawaian yang berjalan dan mengidentifikasi masalah, sedangkan studi literatur menelusuri konsep sistem informasi kepegawaian, *B+ Tree*, *Full-Text Search*, dan metode pengembangan sistem. Tahap evaluasi dan revisi dilakukan secara iteratif hingga seluruh fungsi dinyatakan stabil dan layak digunakan.

Data dikumpulkan melalui empat teknik. Observasi dilakukan dengan mengamati langsung proses pengelolaan data tenaga kependidikan di Rektorat UHO. Wawancara dengan staf kepegawaian menggali kendala yang sering muncul, seperti kesulitan mencari data, pengarsipan berkas, dan lamanya rekapitulasi. Studi dokumentasi mengumpulkan biodata pegawai, riwayat jabatan, riwayat pendidikan, dokumen administrasi, serta format laporan manual yang selama ini digunakan sebagai dasar perancangan basis data dan keluaran sistem. Studi literatur digunakan sebagai landasan teoretis dalam merancang arsitektur sistem.

C. Analisis Kebutuhan Sistem

Analisis kebutuhan menghasilkan 18 kebutuhan fungsional yang dirangkum pada Tabel 1. Setiap kebutuhan dirumuskan sebagai perilaku sistem yang dapat diuji, misalnya sistem harus menampilkan pesan “data tidak ditemukan” apabila pencarian NIP atau nama tidak memberikan hasil. Kebutuhan tersebut melibatkan tiga aktor, yaitu SuperAdmin, Admin, dan Pegawai Tendik (tenaga kependidikan).

Tabel 1. Kebutuhan fungsional sistem

Kode	Fitur	Aktor
F1	Login	SA, A, T
F2	Menampilkan data pegawai	SA, A
F3	Menambah data pegawai	SA, A
F4	Mengubah data pegawai	SA, A
F5	Menghapus data pegawai	SA, A
F6	Kelola riwayat jabatan	SA, A
F7	Kelola riwayat pendidikan	SA, A
F8	Kelola administrasi kepegawaian (SK, cuti, mutasi)	SA, A
F9	Pencarian B+ Tree (NIP/nama)	SA, A
F10	Full-Text Search (isi dokumen)	SA, A
F11	Menampilkan profil pegawai	SA, A, T
F12	Laporan kepegawaian	SA, A
F13	Menambah akun pengguna	SA
F14	Ajuan perubahan data pegawai	T
F15	Verifikasi ajuan data	A
F16	Pembaruan data setelah verifikasi	A
F17	Unduh dokumen	SA, A, T
F18	Logout	SA, A, T

Keterangan: SA = SuperAdmin, A = Admin, T = Pegawai Tendik

Kebutuhan non-fungsional pada Tabel 2 mencakup aspek performa, keandalan, kemudahan penggunaan, kompatibilitas, dan kemudahan pemeliharaan. Ambang waktu respons pencarian maksimal 5 detik ditetapkan berdasarkan prinsip *usability engineering* yang menempatkan durasi tersebut sebagai batas toleransi pengguna dalam menunggu respons sistem sebelum muncul persepsi kegagalan [17], sejalan dengan standar interaksi manusia dan sistem yang mensyaratkan umpan balik tepat waktu [18].

Tabel 2. Kebutuhan non-fungsional sistem

Kode	Kategori	Kebutuhan
NF1	Performance	Hasil pencarian B+ Tree tampil utuh maksimal 5 detik
NF2	Performance	Full-Text Search tetap responsif pada data besar
NF3	Performance	Dasbor dimuat maksimal 3 detik pada koneksi standar
NF4	Performance	Menangani minimal 50 pengguna aktif bersamaan
NF5	Reliability	Ketersediaan minimal 99% pada penggunaan normal
NF6	Reliability	Data tetap tersimpan saat proses input/update gagal
NF7	Usability	Antarmuka mudah dipahami pengguna non-teknis
NF8	Usability	Navigasi fungsi utama tidak membingungkan
NF9	Usability	Pesan kesalahan jelas saat input tidak valid
NF10	Compatibility	Berjalan baik pada Chrome dan Microsoft Edge
NF11	Compatibility	Dapat diakses tanpa spesifikasi perangkat khusus
NF12	Maintainability	Struktur kode mudah dipelihara dan dikembangkan
NF13	Maintainability	Fitur baru dapat ditambah tanpa mengganggu fungsi lama

D. Metode Pengembangan Sistem

Sistem dikembangkan menggunakan metode Waterfall, yaitu model *software development life cycle* klasik yang bersifat linear dan berurutan sehingga setiap fase diselesaikan sebelum fase berikutnya dimulai. Model ini menekankan perencanaan dan dokumentasi yang matang sehingga cocok untuk proyek dengan kebutuhan yang terdefinisi dengan baik dan relatif stabil [19], seperti proses kepegawaian di UHO. Tahap analisis kebutuhan menghasilkan spesifikasi pada Tabel 1 dan Tabel 2. Tahap perancangan menyusun model fungsional, model proses, model data, rancangan antarmuka, serta desain penerapan algoritma pencarian. Tahap implementasi menerjemahkan rancangan ke dalam kode program, tahap pengujian memverifikasi fungsi dan performa, sedangkan tahap pemeliharaan mencakup perbaikan kesalahan (*corrective*), penyesuaian terhadap kebutuhan organisasi (*adaptive*), dan pengembangan fitur (*perfective*).

Implementasi menggunakan bahasa Python dengan kerangka kerja Django yang berarsitektur *Model-View-Template* serta menyediakan *object-relational mapping* (ORM), autentikasi, dan otorisasi bawaan [20]. Sistem ini dibangun dengan arsitektur dua lapis. Django digunakan pada lapisan antarmuka (*frontend*) untuk menyajikan halaman dan mengelola sesi pengguna, sedangkan pengelolaan data dan pencarian dijalankan oleh lapisan layanan (*backend*) berbasis FastAPI yang menyediakan *REST API* dan mengakses basis data melalui ORM SQLAlchemy dengan parameter kueri terikat (*bound parameters*). Django juga menyediakan perlindungan terhadap ancaman umum aplikasi web, seperti *SQL injection*, *cross-site scripting*, dan *cross-site request forgery* [21]. Basis data menggunakan MySQL, yaitu sistem manajemen basis data relasional yang menjamin integritas data melalui *primary key*, *foreign key*, dan transaksi yang memenuhi prinsip ACID (*atomicity*, *consistency*, *isolation*, *durability*) [22], serta mendukung akses multi-pengguna dan *storage engine* InnoDB [23].

E. Perancangan Sistem

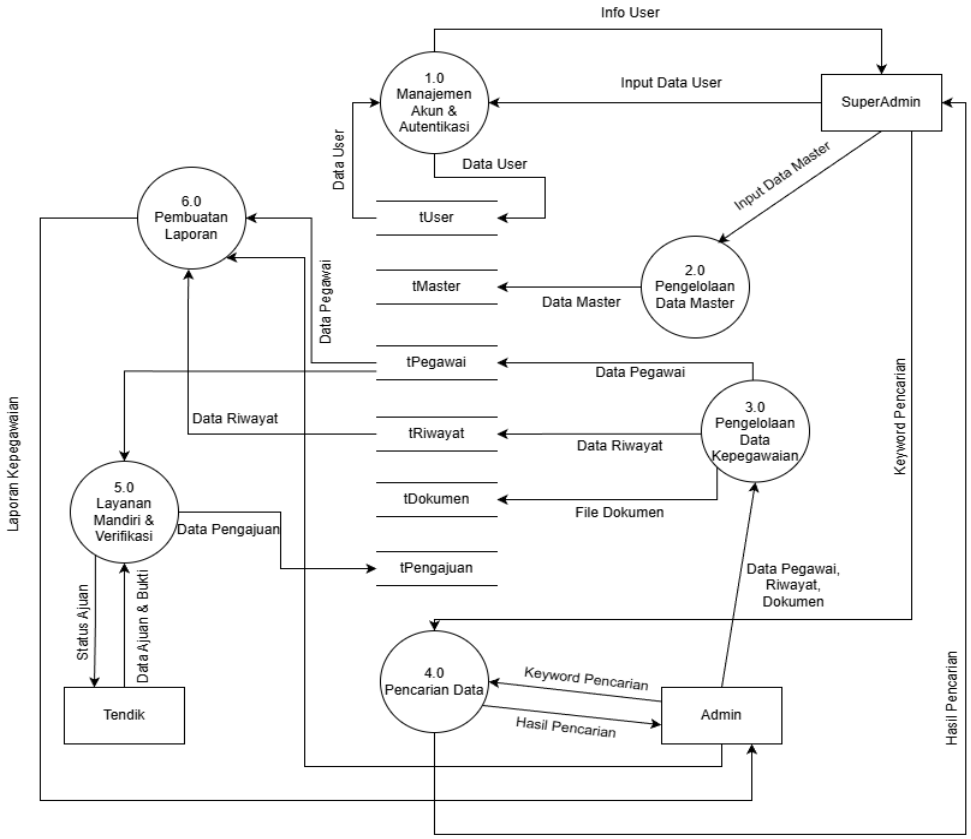
Interaksi pengguna dengan sistem dimodelkan menggunakan *use case diagram* dari *Unified Modeling Language* (UML), yang menggambarkan aktor dan fungsi utama yang dapat dijalankan setiap aktor [24]. Gambar 1 menunjukkan bahwa SuperAdmin merupakan spesialisasi (*generalization*) dari Admin, sehingga mewarisi seluruh hak akses Admin sekaligus memiliki wewenang eksklusif mengelola data master. Pegawai Tendik memiliki hak akses terbatas yang berfokus pada layanan mandiri, yaitu mengubah kata sandi dan melihat profil. Relasi <<extend>> pada *use case* Upload Dokumen menunjukkan bahwa unggah berkas bersifat opsional ketika aktor mengelola data pegawai, administrasi, atau riwayat pendidikan.



Gambar 1. *Use case diagram* sistem dengan tiga aktor: SuperAdmin, Admin, dan Pegawai Tendik

Aliran data dimodelkan dengan *data flow diagram* (DFD), yaitu diagram yang menggambarkan aliran data secara grafis melalui simbol entitas eksternal, proses, penyimpanan data, dan aliran data [25]. Gambar 2 memperlihatkan DFD level 0 yang terdiri atas enam

proses: manajemen akun dan autentikasi (1.0), pengelolaan data master (2.0), pengelolaan data kepegawaian (3.0), pencarian data (4.0), layanan mandiri dan verifikasi (5.0), serta pembuatan laporan (6.0). Seluruh proses bermuara pada penyimpanan data terpusat, yaitu tabel tUser, tMaster, tPegawai, tRiwayat, tDokumen, dan tPengajuan. Proses 3.0 dirinci lebih lanjut pada DFD level 1 menjadi pengelolaan biodata, pengelolaan riwayat dengan validasi NIP terhadap tabel induk, serta pengelolaan administrasi dan unggah dokumen.



Gambar 2. DFD level 0 Sistem Informasi Kepegawaian UHO

Struktur basis data dimodelkan dengan *entity relationship diagram* (ERD), yaitu model data konseptual yang menggambarkan entitas, atribut, dan relasinya [26]. Entitas Pegawai (tPegawai) menjadi entitas induk yang berelasi *one-to-many* dengan entitas riwayat, seperti riwayat pendidikan, keluarga, kepangkatan, jabatan, dan penghargaan, serta berelasi *many-to-one* dengan data master seperti agama dan golongan. Tabel tPegawai memiliki indeks BTREE InnoDB pada atribut NIP (idx_nip) dan nama pegawai (idx_nama); adapun pencarian B+ Tree pada sistem dijalankan pada struktur B+ Tree di tingkat aplikasi yang dijelaskan pada Bagian II.F. Atribut utama tabel tPegawai yang terdiri atas 24 kolom dirangkum pada Tabel 3.

Tabel 3. Atribut utama tabel pegawai (tPegawai)

Nama Kolom	Tipe Data	Keterangan
id	INT	Primary key
nip	VARCHAR	Nomor Induk Pegawai (indeks idx_nip)
NamaPegawai	VARCHAR	Nama lengkap (indeks idx_nama)
tempatlahir	VARCHAR	Tempat lahir
tanggallahir	DATE	Tanggal lahir
nik	VARCHAR	Nomor Induk Kependudukan
nitk / nuptk	VARCHAR	Nomor induk tenaga kependidikan
email, noTelp	VARCHAR	Kontak pegawai
alamat	TEXT	Alamat domisili
status_akun	ENUM	Status akun pengguna
role_idFK	INT	Foreign key hak akses
tAgama_idFK, tJk_idFK	INT	Foreign key agama dan jenis kelamin
created_at, updated_at	TIMESTAMP	Waktu pembuatan dan pembaruan data

Tabel riwayat menyimpan lokasi berkas digital untuk mendukung penelusuran dokumen. Tabel riwayat pangkat (rPangkatGolongan), misalnya, menyimpan nomor SK, tanggal surat, terhitung mulai tanggal (TMT), pejabat penetap, dan lokasi berkas SK, sedangkan tabel riwayat pendidikan (rPendidikan) menyimpan lembaga pendidikan, jurusan, nomor ijazah, tahun lulus, dan lokasi berkas ijazah.

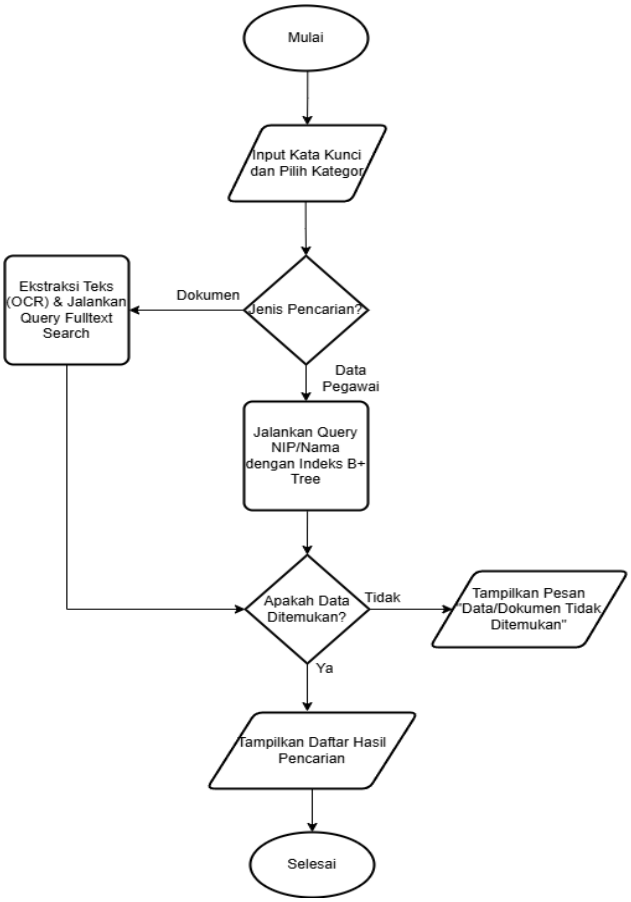
F. Perancangan Algoritma Pencarian

B+ Tree merupakan struktur indeks yang menyimpan kunci secara terurut sehingga operasi pencarian titik maupun rentang dapat dilakukan dengan kompleksitas O(log *N*) melalui penelusuran simpul yang terstruktur. Pada basis data transaksional dengan tuntutan konsistensi tinggi, *B+ Tree* memerlukan mekanisme seperti *next-key locking* yang menambah beban komputasi, terutama pada beban kerja yang didominasi pencarian titik [27]. Dalam

sistem ini, B+ Tree diimplementasikan sebagai kelas Python pada lapisan layanan dengan *order* 1000. Pada setiap permintaan pencarian, sistem membaca seluruh data pegawai (657 data), mengambil daftar dokumen SK setiap pegawai, lalu menyisipkan nama dan NIP dalam huruf kecil sebagai kunci sehingga terbentuk 1.314 kunci yang tersimpan terurut pada simpul daun. Kunci yang sama disimpan sebagai entri terpisah sehingga pegawai dengan nama identik tetap ditemukan. Pencarian dilakukan berdasarkan awalan (*prefix*): suatu data ditampilkan apabila nama atau NIP-nya diawali kata kunci. Pada implementasi ini, penelusuran awalan memeriksa kunci pada seluruh simpul daun sehingga kompleksitas pencariannya linear terhadap jumlah kunci, O(n). Penelusuran dari akar menuju daun pertama yang sesuai lalu memindai daun berikutnya, dengan kompleksitas O(log n + k), belum diterapkan. Indeks idx_nip dan idx_nama pada MySQL tidak digunakan dalam proses ini karena data pegawai dibaca seluruhnya sebelum pohon dibangun.

Full-Text Search merupakan teknik penelusuran informasi pada kumpulan teks berukuran besar dengan memanfaatkan *inverted index* yang memetakan setiap istilah ke lokasi dokumen. Akurasi hasil ditingkatkan melalui tokenisasi dan normalisasi, seperti pemisahan kata, penghapusan *stop words*, dan *stemming*, sedangkan relevansi dapat ditentukan dengan algoritma pemeringkatan seperti TF-IDF dan BM25 [28]. Karena sebagian besar dokumen kepegawaian berupa hasil pindaian, teks dokumen terlebih dahulu diekstraksi dengan *optical character recognition* (OCR), yaitu teknik yang mengubah citra dokumen menjadi teks digital melalui tahapan praproses, segmentasi, normalisasi, ekstraksi fitur, klasifikasi, dan pascaproses, sehingga dokumen fisik dapat diindeks dan dikelola secara terorganisasi [29]. Dalam sistem ini, Full-Text Search diimplementasikan pada lapisan layanan tanpa indeks FULLTEXT MySQL. Dokumen yang ditelusuri adalah berkas ijazah, surat cuti, dan SK pangkat yang tercatat pada tabel riwayat pegawai. Teks setiap dokumen diekstraksi terlebih dahulu dengan pustaka pdfplumber; apabila berkas PDF tidak memiliki lapisan teks, setiap halaman dikonversi menjadi citra 300 dpi lalu dikenali dengan Tesseract OCR (bahasa Indonesia dan Inggris) setelah melalui konversi *grayscale*, peningkatan kontras, penajaman, dan binarisasi dengan ambang 180 pada empat orientasi rotasi, dan hasil terpanjang yang dipilih. Teks hasil ekstraksi disimpan sebagai berkas *cache* sehingga dokumen yang sama tidak diekstraksi ulang. Pencarian dilakukan dengan pencocokan *substring* tanpa membedakan huruf besar-kecil terhadap teks setiap dokumen, tanpa tokenisasi, *stemming*, maupun pemeringkatan relevansi; pada modul pengujian, spasi berlebih pada teks dan kata kunci dinormalisasi terlebih dahulu. Dengan cara ini, kata kunci pendek seperti “S1” tetap dapat ditemukan, berbeda dengan indeks FULLTEXT InnoDB yang secara bawaan hanya mengindeks token minimal tiga karakter (*innodb_ft_min_token_size* = 3).

Mekanisme pencarian dirancang secara hibrida sesuai kategori pencarian yang dipilih pengguna, sebagaimana ditunjukkan pada Gambar 3. Jika pengguna memilih kategori dokumen, sistem mengekstraksi teks melalui OCR lalu menjalankan kueri *Full-Text Search*. Jika pengguna memilih kategori data pegawai, sistem menjalankan kueri NIP atau nama dengan memanfaatkan B+ Tree. Sistem kemudian memeriksa keberadaan data sebelum menampilkan daftar hasil atau pesan bahwa data tidak ditemukan.



Gambar 3. Diagram alir pencarian hibrida: kategori dokumen diproses melalui OCR dan *Full-Text Search*, sedangkan kategori data pegawai diproses melalui B+ Tree

G.Metode Pengujian

Pengujian dilakukan dengan tiga pendekatan. Pertama, *black box testing* untuk memvalidasi fungsi sistem berdasarkan kebutuhan fungsional tanpa memperhatikan struktur internal program. Pengujian ini dapat didukung teknik *equivalence partitioning* yang mengelompokkan masukan ke dalam kelas valid dan tidak valid [30], serta *boundary value analysis* yang menguji nilai minimum, maksimum, dan nilai di sekitar batas masukan [31]. Skenario pengujian disusun dari perilaku yang dirumuskan pada Tabel 1 untuk setiap tingkat hak akses.

Kedua, pengujian performa algoritma pencarian menggunakan parameter precision, recall, waktu respons, tingkat keberhasilan ekstraksi teks, dan tingkat keberhasilan eksekusi skenario. Precision mengukur proporsi hasil yang ditampilkan sistem yang benar-benar relevan, sedangkan recall mengukur proporsi data relevan yang berhasil ditemukan [32], seperti pada

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (1)$$

TP (*true positive*) adalah data relevan yang ditampilkan, FP (*false positive*) adalah data yang ditampilkan tetapi tidak relevan, dan FN (*false negative*)

adalah data relevan yang tidak ditampilkan. Sebagai ilustrasi, apabila terdapat lima data relevan dan sistem menampilkan lima data dengan empat di antaranya benar, precision dan recall masing-masing 80%. Untuk pencarian B+ Tree, data relevan ditetapkan melalui kueri pembanding pada basis data dan diverifikasi secara manual: untuk kata kunci nama, data relevan adalah pegawai yang namanya memuat kata kunci sebagai satu kata utuh pada posisi mana pun (misalnya "Saputri" tidak dihitung relevan untuk kata kunci "Putri"), sedangkan untuk kata kunci NIP adalah pegawai yang NIP-nya diawali kata kunci karena awal NIP menyatakan tanggal lahir. Skenario dengan kata kunci yang tidak memiliki data relevan menghasilkan pembagian 0/0 pada (1), sehingga dinilai sebagai skenario hasil kosong yang berhasil apabila sistem mengembalikan nol data tanpa galat dan tidak diikutsertakan dalam perhitungan rata-rata. Waktu respons B+ Tree diukur pada lapisan antarmuka sejak permintaan dikirim ke lapisan layanan hingga respons diterima (*end-to-end*), sedangkan waktu respons pengujian Full-Text Search diukur di dalam lapisan layanan. Tingkat keberhasilan ekstraksi teks pada (2) dihitung sebagai proporsi dokumen yang menghasilkan teks tidak kosong dari seluruh dokumen; ukuran ini belum menilai ketepatan karakter hasil OCR terhadap teks asli dokumen [33]. Tingkat keberhasilan eksekusi skenario pada (3) menyatakan proporsi skenario kata kunci yang dijalankan sistem hingga selesai tanpa galat dan mengembalikan jumlah hasil, sehingga belum menilai relevansi dokumen yang ditemukan [32].

$$\text{OCR SR} = \frac{\text{Dokumen berhasil diekstraksi}}{\text{Seluruh dokumen}} \times 100\% \quad (2)$$

$$\text{Execution SR} = \frac{\text{Skenario selesai tanpa galat}}{\text{Seluruh skenario pencarian}} \times 100\% \quad (3)$$

Ketiga, *user acceptance testing* (UAT), yaitu pengujian akhir untuk memastikan sistem memenuhi kebutuhan dan ekspektasi pengguna dalam kondisi penggunaan nyata [34]. UAT dilakukan melalui kuesioner skala Likert lima poin [35], [36] dengan skor 5 (Sangat Setuju) hingga 1 (Sangat Tidak Setuju). Responden terdiri atas SuperAdmin, Admin, dan Pegawai Tendik yang mengisi sembilan pernyataan tentang fungsionalitas, efisiensi pencarian, dan kemudahan penggunaan setelah mencoba sistem secara langsung. Skor rata-rata tiap pernyataan diinterpretasikan menggunakan interval kelas

$$\text{Interval} = \frac{\text{Skor tertinggi} - \text{Skor terendah}}{\text{Jumlah kelas}} = 0{,}80 \quad (4)$$

Dengan skor tertinggi 5, skor terendah 1, dan jumlah kelas 5, sehingga diperoleh rentang kategori pada Tabel 4.

Tabel 4. Rentang interpretasi skor rata-rata UAT

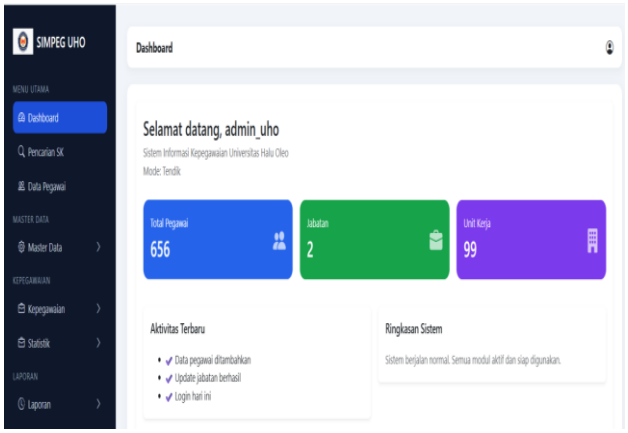
Rentang Skor Rata-rata	Kategori
4,21 – 5,00	Sangat Setuju
3,41 – 4,20	Setuju
2,61 – 3,40	Cukup Setuju
1,81 – 2,60	Tidak Setuju
1,00 – 1,80	Sangat Tidak Setuju

III. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

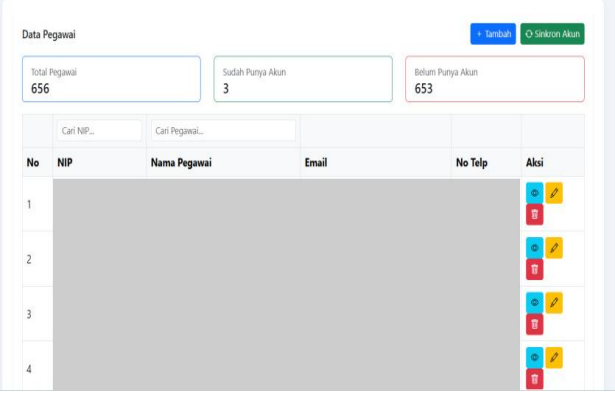
Sistem diimplementasikan sebagai aplikasi web menggunakan Python dengan Django pada lapisan antarmuka dan FastAPI pada lapisan layanan, dengan MySQL sebagai basis data yang dijalankan pada server lokal Laragon. Dari 18 kebutuhan fungsional pada Tabel 1, sistem telah mengimplementasikan 13 fitur fungsional utama, meliputi manajemen pengguna dan hak akses, data master, pengelolaan data pegawai beserta riwayatnya, laporan, serta pencarian data dan dokumen. Fitur ajuan perubahan data dan verifikasinya (F14–F16) direncanakan pada tahap pengembangan berikutnya untuk melengkapi layanan mandiri pegawai.

Setelah autentikasi berhasil, SuperAdmin diarahkan ke dasbor pada Gambar 4. Panel navigasi SIMPEG UHO mengelompokkan menu menjadi Menu Utama, Master Data, Kepegawaian, Laporan, dan Lainnya. Area konten menampilkan kartu statistik total pegawai, jabatan, dan unit kerja, disertai panel aktivitas terbaru dan ringkasan status sistem. Modul data master mengelola referensi agama, jenis kelamin, status pegawai, pangkat/golongan, gaji pokok, klasifikasi jabatan, jabatan, unit kerja, pendidikan, penghargaan, dan cuti dengan antarmuka yang seragam sehingga standar nilai data tetap konsisten di seluruh sistem.



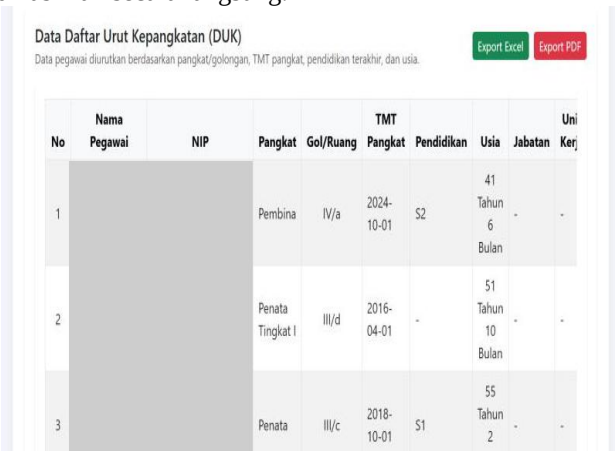
Gambar 4. Halaman dasbor SuperAdmin yang menampilkan statistik pegawai, jabatan, dan unit kerja

Halaman Pengelolaan Data Pegawai pada Gambar 5 merupakan antarmuka inti bagi Admin. Halaman ini menampilkan kartu ringkasan jumlah pegawai, pegawai yang sudah dan belum memiliki akun, kolom pencarian NIP dan nama, serta tabel pegawai dengan tombol aksi untuk melihat detail, mengubah, dan menghapus data. Data yang dikelola mencakup tenaga kependidikan berstatus aparatur sipil negara (ASN) maupun pegawai pemerintah dengan perjanjian kerja (PPPK). Halaman detail pegawai menampilkan biodata lengkap dengan kalkulasi usia otomatis serta menu riwayat pendidikan, pangkat, cuti, kenaikan gaji berkala (KGB), peninjauan masa kerja (PMK), penghargaan, keluarga, tugas belajar, jabatan, dan *impassing*.



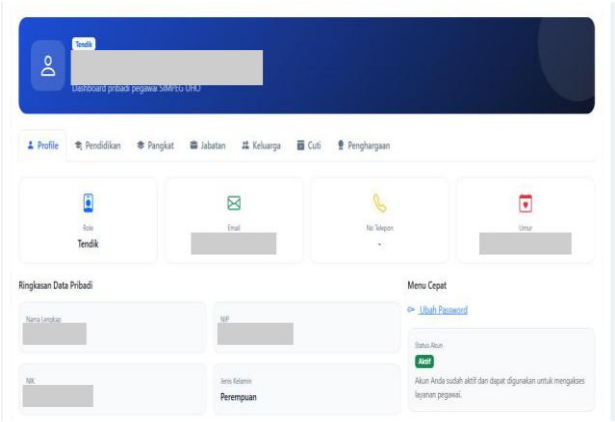
Gambar 5. Halaman Pengelolaan Data Pegawai yang memuat ringkasan status akun, kolom pencarian NIP dan nama, serta tabel data pegawai; data pribadi disamarkan

Halaman Laporan Daftar Urut Kepangkatan (DUK) pada Gambar 6 menyajikan data pegawai yang diurutkan secara otomatis berdasarkan pangkat/golongan, TMT pangkat, pendidikan terakhir, dan usia. Laporan dapat diekspor ke format Excel maupun PDF sehingga rekapitulasi yang sebelumnya disusun manual dapat dihasilkan secara langsung.



Gambar 6. Halaman Laporan DUK dengan fitur ekspor Excel dan PDF; data pribadi disamarkan

Hak akses dibedakan untuk setiap peran. Admin Tendik tidak memiliki akses ke menu data master, sedangkan Pegawai Tendik memperoleh dasbor pribadi pada Gambar 7 yang menampilkan profil, pendidikan, pangkat, jabatan, keluarga, cuti, dan penghargaan miliknya sendiri. Pegawai yang pertama kali masuk dengan kata sandi bawaan diwajibkan mengganti kata sandi sebelum mengakses dasbor demi alasan keamanan.



Gambar 7. Halaman dasbor Pegawai Tendik yang menampilkan ringkasan data pribadi; data pribadi disamarkan

B. Hasil Black Box Testing

Pengujian fungsional dilakukan pada tiga tingkat hak akses dengan total 38 skenario, sebagaimana dirangkum pada Tabel 5. Setiap skenario membandingkan hasil yang diharapkan dengan keluaran sistem, misalnya data pegawai yang ditambahkan harus tersimpan ke basis data, dan penekanan tombol logout harus mengembalikan pengguna ke halaman login. Seluruh skenario menghasilkan keluaran yang sesuai dengan hasil yang diharapkan.

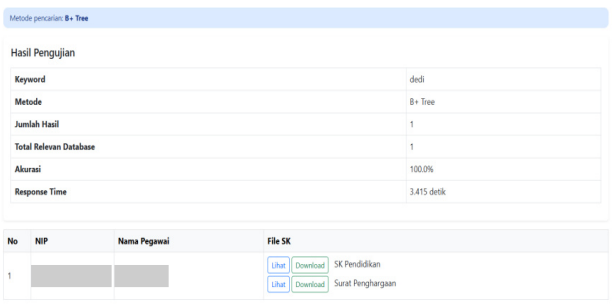
Tabel 5. Rangkuman hasil pengujian black box

Hak Akses	Cakupan Fitur yang Diuji	Jumlah Skenario	Status
SuperAdmin	Login, pilih kategori, dasbor, data pegawai, data master, pencarian B+ Tree dan Full-Text Search, laporan DUK, ekspor laporan, logout	15	Valid
Admin Tendik	Login, dasbor, data dan riwayat pegawai, pencarian B+ Tree dan Full-Text Search, laporan DUK, ekspor laporan, logout	12	Valid
Pegawai Tendik	Login pertama, ubah kata sandi, dasbor pegawai, profil, riwayat, dokumen pribadi, logout	11	Valid
Total		**38**	**Valid**

Pengaturan hak akses berjalan akurat karena setiap pengguna hanya dapat mengakses menu, memanipulasi data, dan melihat informasi sesuai kewenangannya. Tidak ditemukan kebocoran akses data maupun kegagalan sistem saat memproses masukan, sehingga fungsionalitas sistem dinyatakan layak untuk dioperasikan.

C. Hasil Pengujian B+ Tree

Pengujian *B+ Tree* dijalankan melalui halaman Pencarian SK dengan tombol “Cari dengan *B+ Tree*”. Gambar 8 memperlihatkan contoh pengujian dengan kata kunci “dedi”. Sistem secara otomatis menampilkan panel hasil pengujian yang berisi kata kunci, metode, jumlah hasil, total data relevan dalam basis data, akurasi, dan waktu respons, disertai tabel pegawai beserta tautan dokumen SK yang terkait untuk dilihat atau diunduh. Nilai akurasi pada panel tersebut dihitung sistem sebagai rasio jumlah hasil terhadap jumlah data yang diawali kata kunci, sehingga untuk pencarian awalan nilainya selalu 100%. Oleh karena itu, evaluasi pada penelitian ini menggunakan precision dan recall terhadap data relevan yang ditetapkan secara terpisah sebagaimana dijelaskan pada Bagian II.G.



Gambar 8. Pengujian pencarian B+ Tree dengan kata kunci “dedi” dengan waktu respons 3,415 detik; data pribadi disamarkan

Pengujian dilanjutkan dengan sepuluh variasi kata kunci berupa NIP lengkap, penggalan NIP, dan nama pegawai untuk menguji kestabilan sistem dalam menangani permintaan yang beragam. Hasilnya disajikan pada Tabel 6.

Tabel 6. Recall dan waktu respons pencarian *B+ Tree*

No	Kata Kunci (NIP/Nama)	Jumlah Hasil	Relevan	Recall	Waktu (detik)
1	dedi	1	1	100%	3,4150
2	1984**********05	1	1	100%	3,3822
3	Ahmad	3	4	75%	3,5825
4	1968**********04	1	1	100%	2,9992
5	Muhammad	17	21	80,95%	3,1539
6	1976	16	16	100%	3,2354
7	Darwin	0	0	–*	2,9985
8	1968**********01	1	1	100%	3,9200
9	Putri	2	4	50%	3,3849
10	19720	11	11	100%	3,3067
Rata-rata recall (9 skenario dengan data relevan)				**89,55 %**	
Precision (9 skenario dengan data relevan)				**100%**	
Rata-rata waktu respons (10 skenario)					**3,338**

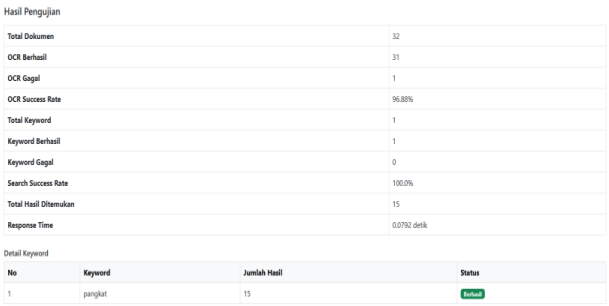
Catatan : Skenario hasil kosong (total data relevan = 0); dinilai berhasil karena sistem mengembalikan nol data tanpa galat. NIP lengkap disamarkan untuk melindungi data pribadi. Relevan = jumlah data relevan sesuai definisi pada Bagian II.G; precision dan recall dihitung dengan (1). Waktu respons diukur secara *end-to-end* pada lapisan antarmuka.

Tabel 6 menunjukkan bahwa seluruh data yang ditampilkan pada sembilan skenario dengan data relevan memang memuat kata kunci, sehingga precision mencapai 100%. Untuk kata kunci NIP, recall juga mencapai 100% karena pencarian awalan sesuai dengan struktur NIP. Untuk kata kunci nama, recall turun pada "Ahmad" (75%), "Muhammad" (80,95%), dan "Putri" (50%) karena terdapat pegawai yang namanya memuat kata kunci tersebut di tengah nama, misalnya setelah gelar atau nama depan lain, sehingga tidak terjangkau pencarian awalan. Rata-rata recall pada sembilan skenario adalah 89,55%. Pada percobaan ke-7, kata kunci "Darwin" tidak memiliki data relevan dan sistem mengembalikan hasil kosong tanpa galat, sehingga dinilai sebagai skenario hasil kosong yang berhasil. Waktu respons *end-to-end* berkisar antara 2,9985 dan 3,9200 detik dengan rata-rata 3,338 detik sehingga memenuhi kebutuhan NF1, yaitu maksimal 5 detik. Waktu tersebut mencakup permintaan HTTP ke lapisan layanan, pembacaan 657 data pegawai beserta daftar dokumen SK masing-masing (sekitar 2.628 kueri), pembangunan pohon, dan penelusuran awalan.

D. Hasil Pengujian Full-Text Search

Pengujian *Full-Text Search* mengevaluasi kemampuan sistem menelusuri kata kunci di dalam isi dokumen digital pegawai, seperti SK pangkat, ijazah, dan dokumen administrasi lainnya. Fungsi "Cari dengan *Full-Text Search*" menampilkan daftar dokumen beserta cuplikan isi (*snippet*) yang menyoroti letak kata kunci di dalam dokumen. Sebelum ekstraksi karakter oleh mesin Tesseract OCR, setiap dokumen yang diunggah diproses secara otomatis melalui konversi keabuan (*grayscale*), peningkatan kontras, dan binarisasi (*thresholding*) untuk meningkatkan kualitas pindaian. Hasil ekstraksi disimpan melalui mekanisme *caching* di direktori khusus sehingga dokumen yang sama tidak perlu diproses ulang.

Modul Uji *Full-Text Search* menjalankan pencarian terhadap seluruh 32 dokumen dalam basis data. Gambar 9 memperlihatkan bahwa 31 dokumen berhasil diekstraksi dan 1 dokumen gagal, sehingga tingkat keberhasilan ekstraksi teks sebesar 96,88%. Kegagalan tersebut disebabkan oleh kualitas fisik pindaian yang rendah sehingga karakter teks tidak dapat diekstraksi dengan presisi yang memadai. Dokumen dihitung gagal apabila teks yang dihasilkan kosong atau hanya berisi tanda air aplikasi pemindai. Label *OCR Success Rate* dan *Search Success Rate* pada Gambar 9 berturut-turut sesuai dengan (2) dan (3).



Gambar 9. Hasil pengujian metrik Full-Text Search dengan kata kunci "pangkat" terhadap 32 dokumen

Konsistensi performa kemudian diuji dengan sepuluh variasi kata kunci berupa frasa, kata tunggal, nama, dan NIP. Hasilnya disajikan pada Tabel 7.

Tabel 7. Hasil pengujian *Full-Text Search* berdasarkan variasi kata kunci

No	Kata Kunci	Jumlah Hasil	Status	Waktu (detik)
1	kenaikan pangkat	15	Berhasil	0,0653
2	S1	5	Berhasil	0,1976
3	ijazah	10	Berhasil	0,0830
4	Program Studi	13	Berhasil	0,0686
5	Desy	2	Berhasil	0,0841
6	terhitung mulai	14	Berhasil	0,0778
7	1985**********08	1	Berhasil	0,0834
8	tugas	1	Berhasil	0,0674
9	Darwin	0	Berhasil (data tidak ada)	0,1053
10	mutasi	0	Berhasil (data tidak ada)	0,0812
Tingkat keberhasilan eksekusi skenario (10 dari 10)			**100%**	
Rata-rata waktu respons (10 skenario)				**0,0914**

Catatan : NIP lengkap disamarkan untuk melindungi data pribadi. Status "Berhasil" menyatakan skenario dijalankan hingga selesai tanpa galat.

Seluruh sepuluh skenario pada Tabel 7 dijalankan hingga selesai tanpa galat, termasuk dua skenario tanpa hasil ("Darwin" dan "mutasi"), sehingga tingkat keberhasilan eksekusi skenario menurut (3) adalah 100%.

Angka ini menunjukkan keandalan eksekusi, bukan ketepatan hasil, karena relevansi dokumen yang ditemukan belum dibandingkan dengan isi dokumen asli. Kata kunci "S1" menghasilkan lima dokumen karena pencocokan *substring* tidak dibatasi panjang token. Sebaliknya, pencocokan *substring* berpotensi menghasilkan kecocokan di dalam kata lain, misalnya "tugas" pada "petugas", dan kesalahan pengenalan karakter dapat membuat dokumen relevan tidak ditemukan. Waktu pemrosesan berkisar antara 0,0653 dan 0,1976 detik dengan rata-rata 0,0914 detik. Waktu ini diukur di dalam lapisan layanan dengan teks dokumen yang dibaca dari *cache*, sehingga tidak mencakup waktu ekstraksi OCR awal.

E. Hasil User Acceptance Testing

Rekapitulasi penilaian responden disajikan pada Tabel 8, dengan kategori setiap pernyataan ditentukan berdasarkan rentang pada Tabel 4.

Tabel 8. Rekapitulasi hasil penilaian UAT

No	Pernyataan	Rata-rata Skor	Kategori
1	Sistem mudah digunakan dalam pengelolaan data kepegawaian	4,3	Sangat Setuju
2	Tampilan antarmuka sistem mudah dipahami	4,4	Sangat Setuju
3	Fitur pengelolaan data pegawai berjalan sesuai kebutuhan	4,2	Setuju
4	Informasi riwayat kepegawaian ditampilkan dengan baik	4,2	Setuju
5	Fitur pencarian B+ Tree memudahkan pencarian data	4,3	Sangat Setuju
6	Fitur Full-Text Search membantu menemukan dokumen	4,3	Sangat Setuju
7	Laporan yang dihasilkan sesuai kebutuhan pengguna	4,0	Setuju
8	Sistem membantu pengelolaan data secara keseluruhan	4,1	Setuju
9	Penggunaan sistem lebih efisien dibandingkan pengelolaan manual dengan spreadsheet	3,0	Cukup Setuju
Rata-rata keseluruhan		**4,09**	**Setuju**

Skor rata-rata keseluruhan sebesar 4,09 berada pada rentang 3,41–4,20 sehingga termasuk kategori Setuju. Skor tertinggi diperoleh pernyataan tentang kemudahan memahami antarmuka (4,4), sedangkan fitur pencarian B+ Tree dan Full-Text Search masing-masing memperoleh skor 4,3 dengan kategori Sangat Setuju. Hal ini menunjukkan bahwa fitur pencarian yang menjadi fokus penelitian dinilai paling membantu oleh pengguna. Pernyataan tentang efisiensi sistem dibandingkan pengelolaan manual dengan *spreadsheet* memperoleh skor terendah (3,0; Cukup Setuju), yang menunjukkan bahwa manfaat efisiensi belum sepenuhnya dirasakan pengguna dan perlu dievaluasi kembali setelah sistem digunakan secara rutin. Fitur laporan (4,0) dan dukungan sistem terhadap pengelolaan data secara keseluruhan (4,1) juga masih dapat ditingkatkan.

F. Evaluasi Pemenuhan Kebutuhan Non-Fungsional

Pemenuhan kebutuhan non-fungsional pada Tabel 2 dievaluasi melalui hasil pengujian dan tinjauan arsitektur teknis, sebagaimana dirangkum pada Tabel 9. Pengukuran melalui *Network Tab* pada *Developer Tools* menunjukkan halaman dasbor dimuat dalam 1,21 detik, sehingga memenuhi NF3. Aspek *usability* diperkuat oleh hasil UAT, sedangkan aspek *maintainability* didukung oleh pemisahan lapisan antarmuka (Django) dan lapisan layanan (FastAPI).

Tabel 9. Hasil evaluasi kebutuhan non-fungsional

Kode	Status	Dasar Pemenuhan
NF1	Terpenuhi	Rata-rata waktu B+ Tree 3,338 detik (< 5 detik)
NF2	Sebagian	Rata-rata 0,0914 detik pada 32 dokumen; belum diuji pada data besar
NF3	Terpenuhi	Waktu muat dasbor 1,21 detik (< 3 detik)
NF4	Terpenuhi*	Endpoint sinkron FastAPI dijalankan pada thread pool untuk permintaan konkuren
NF5	Terpenuhi*	*Error handling* pada level aplikasi dan server
NF6	Terpenuhi*	*Atomic transaction* melalui sesi SQLAlchemy untuk integritas data
NF7	Terpenuhi	UAT: antarmuka dinilai mudah dipahami (4,4)
NF8	Terpenuhi	UAT: navigasi fitur dinilai efektif
NF9	Terpenuhi	Pesan validasi otomatis saat input salah
NF10	Terpenuhi	Berjalan stabil pada Chrome dan Microsoft Edge
NF11	Terpenuhi	Berbasis web dan tidak bergantung perangkat
NF12	Terpenuhi*	Pemisahan lapisan antarmuka (Django) dan layanan (FastAPI)
NF13	Terpenuhi*	Pengembangan fitur secara modular

Catatan: status bertanda bintang didasarkan pada desain arsitektur dan standar teknis kerangka kerja, belum melalui pengujian beban atau pemantauan ketersediaan

G. Pembahasan

Hasil pengujian menunjukkan bahwa kedua mekanisme pencarian berjalan sesuai perannya masing-masing. B+ Tree menampilkan hasil yang tepat (precision 100%) dan sangat sesuai untuk NIP, tetapi pencarian awalan membatasi recall pencarian nama. Keunggulan kecepatan B+ Tree yang dilaporkan pada penelitian sebelumnya [7], [9] belum tercermin pada implementasi ini karena pohon dibangun ulang pada setiap permintaan dan penelusuran awalan memeriksa seluruh simpul daun. Full-Text Search berbasis pencocokan *substring* menjalankan seluruh skenario tanpa galat dengan waktu pemrosesan yang rendah pada 32 dokumen. Pendekatan ini berbeda dari *full-text index* pada MySQL yang dilaporkan mempercepat pencarian berbasis teks [11]; tanpa indeks, waktu pencarian akan meningkat seiring bertambahnya jumlah dan panjang dokumen, sehingga untuk skala data yang lebih besar penggunaan indeks teks atau mesin pencari khusus perlu dipertimbangkan [12]. Pemisahan jalur pencarian berdasarkan jenis data membuat pengguna dapat mencari pegawai maupun isi dokumen dari satu antarmuka.

Waktu respons B+ Tree (3,338 detik) dan Full-Text Search (0,0914 detik) tidak dapat dibandingkan secara langsung karena diukur pada titik yang berbeda. Waktu B+ Tree diukur pada lapisan antarmuka dan mencakup pembacaan seluruh data pegawai, sekitar 2.628 kueri dokumen SK, serta pembangunan pohon pada setiap permintaan. Dengan hanya 1.314 kunci, penelusuran pohon diperkirakan berkontribusi kecil terhadap total waktu, sehingga waktu respons lebih ditentukan oleh pembacaan data dan pembangunan pohon. Waktu Full-Text Search diukur di dalam lapisan layanan dan memanfaatkan teks hasil ekstraksi yang telah tersimpan dalam *cache*.

Penelitian ini memiliki beberapa keterbatasan. Pengujian dilakukan pada server lokal dengan sepuluh skenario kata kunci untuk setiap algoritma dan 32 dokumen, sehingga perilaku sistem pada volume data dan jumlah pengguna yang jauh lebih besar masih perlu diuji. Pada sisi B+ Tree, pohon dibangun ulang pada setiap permintaan, simpul daun belum saling terhubung, dan penelusuran awalan belum memanfaatkan urutan kunci, sehingga kompleksitas pencarian masih linear. Pada sisi Full-Text Search, tingkat keberhasilan ekstraksi teks dan eksekusi skenario belum menilai ketepatan karakter OCR maupun relevansi dokumen yang ditemukan, sehingga evaluasi precision dan recall terhadap isi dokumen asli masih perlu dilakukan. Perbandingan langsung dengan pencarian tanpa indeks pada data yang sama juga belum dilakukan. Selain itu, sebagian kebutuhan non-fungsional (NF4, NF5, NF6, NF12, dan NF13) dinyatakan terpenuhi berdasarkan arsitektur, sehingga pengujian beban dan pemantauan ketersediaan menjadi agenda penelitian berikutnya.

IV. KESIMPULAN DAN SARAN

Sistem Informasi Kepegawaian UHO modul tenaga kependidikan berhasil dibangun berbasis web dengan mengintegrasikan *B+ Tree* untuk pencarian data pegawai terstruktur dan *Full-Text Search* berbantuan OCR untuk penelusuran isi dokumen. Pendekatan hibrida ini sesuai dengan karakteristik data kepegawaian yang terdiri atas atribut terstruktur dan dokumen hasil pindaian. Seluruh 38 skenario black box berjalan valid sesuai pembagian hak akses. Pencarian B+ Tree menampilkan hasil yang tepat dengan waktu respons *end-to-end* di bawah 5 detik, tetapi pencarian berbasis awalan belum menemukan nama yang memuat kata kunci di tengah nama. Full-Text Search berhasil mengekstraksi teks dari 31 di antara 32 dokumen (96,88%) dan menjalankan seluruh skenario pencarian tanpa galat, dengan kualitas pindaian sebagai faktor penentu keberhasilan ekstraksi. Hasil UAT menunjukkan bahwa sistem dapat diterima pengguna dan layak menggantikan pengelolaan data berbasis *spreadsheet*.

Pengembangan selanjutnya disarankan untuk mengintegrasikan modul tenaga pendidik (dosen) agar pengelolaan sumber daya manusia UHO lebih menyeluruh, menambah kapasitas penyimpanan server secara berkala seiring bertambahnya dokumen digital, menambahkan kustomisasi parameter laporan, serta melengkapi fitur ajuan perubahan data dan verifikasinya. Pada sisi algoritma, B+ Tree sebaiknya dibangun sekali dan diperbarui hanya saat data berubah, penelusuran awalan diubah agar turun dari akar menuju daun pertama yang sesuai dan memindai daun yang saling terhubung, serta pencarian nama diperluas agar mencakup kata di tengah nama; Full-Text Search perlu dilengkapi indeks teks serta dievaluasi dengan precision dan recall terhadap isi dokumen asli. Penelitian lanjutan juga perlu melakukan pengujian beban, pengujian pada volume data yang lebih besar, serta perbandingan langsung dengan pencarian tanpa indeks untuk mengukur peningkatan kinerja secara kuantitatif.

UCAPAN TERIMA KASIH

Penulis P.E.W.A. mengucapkan terima kasih kepada Bagian Kepegawaian Biro Keuangan dan Umum Rektorat Universitas Halu Oleo yang telah memberikan data dan informasi yang diperlukan dalam penelitian ini, serta kepada Program Studi Ilmu Komputer FMIPA Universitas Halu Oleo atas dukungan fasilitas selama penelitian

DAFTAR PUSTAKA

[1] D. Juliana, Irmayunita, and E. Susanto, "Peran teknologi informasi dalam meningkatkan kinerja organisasi," *Jurnal Mahasiswa Ekonomi Bisnis (JMEB)*, Vol. 1, No. 3 (2024) 181–182.

[2] L. Zemmouchi-Ghomari, "Basic concepts of information systems," in *Contemporary Issues in Information Systems – A Global Perspective*. London: IntechOpen (2022).

[3] S. Khadijah and M. I. P. Nasution, "Peran sistem informasi manajemen dalam pengambilan keputusan strategis di organisasi," *SURPLUS: Jurnal Ekonomi dan Bisnis*, Vol. 2, No. 2 (2024) 141–148.

[4] Universitas Halu Oleo, *Laporan Kinerja Universitas Halu Oleo Tahun 2024*. Kendari: Universitas Halu Oleo (2025). [Online]. Available: https://uho.ac.id/wp-content/uploads/2025/11/LAKIN-UHO-2024-Fix.pdf

[5] D. Markuci and C. Prianto, "Analisis perbandingan penggunaan algoritma sequential search dan binary search pada aplikasi surat perjalanan dinas," *JATI (Jurnal Mahasiswa Teknik Informatika)*, Vol. 6, No. 1 (2022) 110–119.

[6] M. Z. Iksan and F. A. Abdussalaam, "Design of a web-based personnel administration management information system at Politeknik Piksi Ganesha," *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, Vol. 7, No. 1 (2023) 128–140.

[7] K. Bali and S. Maggu, "RDBMS indexing using B+ trees," *International Journal of Advances in Engineering and Management*, Vol. 2, No. 11 (2020) 250–252.

[8] I. A. Adeleke, A. D. Gbadebo, and A. O. Dawodu, "A B+-tree-based indexing and storage of numerical records in school databases," *Asian Journal of Research in Computer Science*, Vol. 16, No. 4 (2023) 418–427.

[9] A. Garg, A. Rai, A. Verma, A. Srivastava, and A. Aggarwal, "Searching using B/B+ tree in database management system," *International Journal for Research in Applied Science and Engineering Technology*, Vol. 9, No. 7 (2021) 3702–3706.

[10] S. Safaee, M. Mirabi, A. M. Rahmani, and A. A. Safaei, "A distributed B+Tree indexing method for processing range queries over streaming data," *Cluster Computing*, Vol. 27, No. 2 (2024) 1251–1274.

[11] I. Thoib, B. P. Candra, N. Sururi, and D. S. Nugraha, "Perbandingan performa pencarian data berbasis teks dengan dan tanpa full-text index pada basis data MySQL," *INSOLOGI: Jurnal Sains dan Teknologi*, Vol. 3, No. 6 (2024) 663–673.

[12] G. Fotopoulos, P. Koloveas, P. Raftopoulou, and C. Tryfonopoulos, "Comparing data store performance for full-text search: To SQL or to NoSQL?," in *Proc. 12th Int. Conf. Data Science, Technology and Applications (DATA 2023)* (2023) 406–413.

[13] H. V. Khodiakova, N. V. Khodiakova, and V. A. Pozdeev, "Full text search setup on a website," *Control Systems and Computers*, No. 5–6 (2021) 55–60.
[14] B. Kieu-Do-Nguyen, T.-K. Dang, N. The Binh, C. Pham-Quoc, H. Phuc Nghi, N.-T. Tran, K. Inoue, C.-K. Pham, and T.-T. Hoang, "A high-performance non-indexed text search system," *Electronics*, Vol. 13, No. 11 (2024) 2125.
[15] W. Iqbal, W. Ilyas Malik, F. Bukhari, K. M. Almustafa, and Z. Nawaz, "Big data full-text search index minimization using text summarization," *Information Technology and Control*, Vol. 50, No. 2 (2021) 375–389.
[16] K. K. Dukhnovska and I. L. Myshko, "Analysis and comparison of full-text search algorithms," *Control Systems and Computers*, No. 3 (2024) 45–52.
[17] J. Nielsen, *Usability Engineering*. Morgan Kaufmann Publishers (1994).
[18] *Ergonomics of Human-System Interaction — Part 11: Usability: Definitions and Concepts*, ISO Standard 9241-11 (2018).
[19] A. Gupta, A. Rawal, and Y. Barge, "Comparative study of different SDLC models," *International Journal for Research in Applied Science and Engineering Technology*, Vol. 9, No. 11 (2021) 73–80.
[20] M. M. Tkhabisimova, U. G. Baymuradov, and A. Sh. Izrailova, "Development and implementation of web applications in Python using the Django framework," *Ekonomika i Upravlenie: Problemy, Resheniya*, Vol. 12, No. 8 (2024) 53–63.
[21] B. K. P and Deepamala, "Leveraging Django: A web framework for efficient infrastructure monitoring tools," *International Journal for Research in Applied Science and Engineering Technology*, Vol. 11, No. 10 (2023) 407–413.
[22] S. bin Uzayr, *Mastering MySQL for the Web*. Boca Raton: CRC Press (2022).
[23] I. Suster and T. Ranisavljevic, "Optimization of MySQL database," *Journal of Process Management and New Technologies*, Vol. 11, No. 1–2 (2023) 141–151.
[24] S. Al-Fedaghi, "UML modeling to TM modeling and back," *IJCSNS International Journal of Computer Science and Network Security*, Vol. 21, No. 1 (2021) 84–96.
[25] F. Syahputra, A. Dalimunthe, and Z. Matondang, "Data flow diagram design for multimedia laboratory management information systems at Universitas Negeri Medan," in *Proc. 5th Int. Conf. Innovation in Education, Science, and Culture (ICIESC 2023)*, Medan (2024).
[26] S. M. Pulungan, R. Febrianti, T. Lestari, N. Gurning, and N. Fitriana, "Analisis teknik entity-relationship diagram dalam perancangan database," *Jurnal Ekonomi Manajemen dan Bisnis (JEMB)*, Vol. 1, No. 2 (2023) 143–147.
[27] S. Nakazono, Y. Bessho, H. Kawashima, and T. Nakamori, "Griffin: Fast transactional database index with hash and B+-tree," in *Proc. 2024 IEEE 20th Int. Conf. e-Science* (2024) 1–10.
[28] A. Ayyagiri, O. Goel, and S. Jain, "Innovative approaches to full-text search with Solr and Lucene," *Universal Research Reports*, Vol. 11, No. 1 (2024) 209–224.
[29] K. Kushavaha, A. Kumar, and A. Chakrawati, "OCR: Optical character recognition," *International Journal for Research in Applied Science and Engineering Technology*, Vol. 12, No. 4 (2024) 3854–3856.
[30] J. Shadiq, A. Safei, and R. W. R. Loly, "Pengujian aplikasi peminjaman kendaraan operasional kantor menggunakan black box testing," *Information Management for Educators and Professionals*, Vol. 5, No. 2 (2021) 97–110.
[31] C. D. Megawati, N. D. Miwa, and B. R. P. D. Palevi, "Black box testing of the "Hybrid Engine" application using boundary value analysis technique," *SinkrOn*, Vol. 8, No. 2 (2023) 923–938.
[32] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press (2008).
[33] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)," *IEEE Access*, Vol. 8 (2020) 142642–142668.
[34] H. Thabibi, S. F. A. Wati, and T. P. Rinjeni, "Implementasi user acceptance testing (UAT) pada website e-commerce UMKM BBhealthy," *Adopsi Teknologi dan Sistem Informasi (ATASI)*, Vol. 4, No. 1 (2025) 19–26.
[35] Wulandari, Nofiyani, and H. Hasugian, "User acceptance testing (UAT) pada electronic data preprocessing guna mengetahui kualitas sistem," *Jurnal Mahasiswa Ilmu Komputer*, Vol. 4, No. 1 (2023) 20–27.
[36] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta (2019).