

IMPLEMENTASI BACK-END SISTEM INFORMASI E-OFFICE STUDI KASUS BIRO ADMINISTRASI PIMPINAN PROVINSI NUSA TENGGARA BARAT

(Back-end Implementation of an E-office Information System: A Case Study of the Leadership Administration Bureau of West Nusa Tenggara Province)

Indra Maulana^[1], Raphael Bianco Huwae^[1], Tarmidzi^[2], Andy Hidayat Jatmika^[1]

^[1]Dept. Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

^[2]Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat
Jl. Pejanggik 12, Kantor Gubernur NTB, Mataram, Lombok NTB, INDONESIA

Email: indramlnx666@gmail.com, raphael.bianco.huwae@unram.ac.id, Tarmidzi17@gmail.com, andy@unram.ac.id

Abstrak

Transformasi digital dalam lingkungan pemerintahan menuntut sistem informasi yang andal, terutama pada sisi back-end yang berperan penting dalam pengelolaan data dan komunikasi antarmodul. Penelitian ini bertujuan untuk mengimplementasikan sistem back-end pada aplikasi e-office di Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat. Pengembangan dilakukan menggunakan Node.js dan Express.js sebagai framework server, serta MongoDB sebagai basis data. Metode pengembangan yang digunakan meliputi analisis kebutuhan, perancangan struktur database, dan implementasi API untuk mendukung operasi Create, Read, Update, dan Delete (CRUD). Hasil implementasi menunjukkan bahwa sistem back-end mampu mengelola data surat dan arsip dengan baik, serta mendukung pertukaran data secara efisien antara server dan klien. Dengan adanya sistem ini, proses administrasi di lingkungan biro menjadi lebih terintegrasi, cepat, dan aman. Implementasi back-end e-office ini diharapkan dapat mendukung upaya pemerintah daerah dalam mewujudkan tata kelola digital yang efektif dan transparan.

Keywords: E-office, Back-end, Node.js, Express.js, Sistem Informasi, MongoDB.

1. PENDAHULUAN

Pengelolaan data dan dokumen merupakan salah satu aspek penting dalam mendukung tata kelola pemerintahan modern. Data yang tersimpan secara terstruktur dan diolah secara efektif dapat membantu proses pengambilan keputusan, meningkatkan transparansi, serta memperkuat akurasi informasi dalam lingkungan instansi pemerintahan[1][2]. Dalam penerapan sistem informasi, back-end memiliki peran penting karena berfungsi untuk mengelola database, memproses permintaan pengguna, menjalankan logika sistem, serta menjaga konsistensi dan keamanan data yang digunakan oleh seluruh komponen aplikasi[3].

Seiring meningkatnya volume dokumen dan kompleksitas pekerjaan administrasi, instansi pemerintah membutuhkan sistem yang mampu mengelola data secara cepat, terintegrasi, dan aman. Proses administrasi yang masih dilakukan secara manual atau belum sepenuhnya terhubung dalam satu sistem dapat menimbulkan berbagai kendala, seperti keterlambatan pencatatan dokumen, kesulitan pencarian arsip, lambatnya distribusi informasi, serta risiko kehilangan atau ketidaksesuaian data. Kondisi tersebut dapat mengurangi efisiensi kerja staf administrasi dan menghambat kelancaran pelayanan internal instansi[4].

Permasalahan tersebut juga ditemukan pada Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat. Berdasarkan hasil observasi selama kegiatan PKL dan diskusi dengan staf terkait, pengelolaan dokumen administrasi seperti surat, disposisi, arsip, dan data pengguna masih memerlukan penyempurnaan agar proses kerja dapat berjalan lebih efektif. Beberapa kendala yang ditemukan meliputi pelacakan dokumen yang belum optimal, distribusi informasi yang membutuhkan waktu, pengelolaan arsip yang belum sepenuhnya terstruktur, serta kebutuhan terhadap sistem yang mampu mendukung pencatatan dan pemantauan data secara lebih konsisten.

Implementasi back-end sistem informasi e-office menjadi solusi yang ditawarkan untuk menjawab permasalahan tersebut. Back-end yang terstruktur memungkinkan proses pengelolaan data dilakukan melalui operasi Create, Read, Update, and Delete (CRUD), sehingga data surat, arsip, disposisi, dan pengguna dapat disimpan, diperbarui, dicari, serta dikelola dengan lebih baik. Dalam pengembangan sistem ini, teknologi Node.js

dan *Express.js* digunakan sebagai *server framework* karena mampu mendukung pembangunan API yang cepat, ringan, dan fleksibel. *Node.js* memiliki kemampuan *non-blocking* yang memungkinkan *server* menangani banyak permintaan pengguna secara bersamaan, sedangkan *Express.js* memudahkan pengaturan *routing*, *middleware*, dan komunikasi data antara *client* dan *server*[5][6]. *MongoDB* digunakan sebagai basis data karena memiliki struktur penyimpanan berbasis dokumen yang fleksibel dan sesuai untuk pengelolaan data administrasi. Kombinasi teknologi tersebut dinilai sesuai untuk mendukung kebutuhan sistem *e-office* yang membutuhkan kecepatan akses, skalabilitas, serta keandalan dalam pengelolaan data[7].

Kegiatan ini dilakukan melalui beberapa tahapan, yaitu identifikasi masalah, analisis kebutuhan, perancangan struktur *database*, implementasi API, integrasi sistem, dan pengujian fungsi *back-end*. Implementasi sistem difokuskan pada pengelolaan data pengguna, surat, disposisi, arsip, serta pengujian endpoint API menggunakan HTTPie. Hasil implementasi menunjukkan bahwa *back-end* sistem informasi *e-office* mampu mendukung pengelolaan data administrasi secara lebih terstruktur, mempermudah pertukaran data antara *server* dan *client*, serta membantu meningkatkan efisiensi pengelolaan dokumen di lingkungan Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat.

2. TINJAUAN PUSTAKA

2.1. Sistem Informasi

Sistem informasi merupakan kumpulan elemen yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan serta pengelolaan organisasi[8]. Elemen-elemen tersebut meliputi perangkat keras, perangkat lunak, manusia, prosedur, dan data yang terintegrasi menjadi satu kesatuan. Keberadaan sistem informasi memungkinkan organisasi mengubah data mentah menjadi informasi yang bernilai, meningkatkan efisiensi operasional, serta mendukung koordinasi lintas unit. Di era digital, sistem informasi juga berperan strategis dalam modernisasi administrasi, mempercepat akses data, dan memperkuat transparansi serta akuntabilitas organisasi[9].

2.2. E-office

E-Office merupakan sistem digital yang digunakan untuk mengelola kegiatan perkantoran secara elektronik, seperti pengelolaan surat masuk, surat keluar, disposisi, dan arsip digital. Penerapan *e-office* membantu mengurangi ketergantungan pada dokumen fisik, mempercepat distribusi surat, dan meningkatkan pelacakan administrasi. Selain efisiensi waktu, sistem ini memudahkan koordinasi antarbagian melalui akses dokumen secara *real-time*, memungkinkan proses persetujuan dan disposisi berlangsung lebih cepat[10]. Integrasi *e-office* dengan basis data dan sistem manajemen dokumen menjadikannya sarana efektif dalam meningkatkan transparansi dan akuntabilitas administrasi publik.

2.3. Back-end Development

Back-end development adalah proses pengembangan sisi server aplikasi yang berfokus pada manajemen data, logika bisnis, autentikasi, serta komunikasi dengan *front-end*. Komponen *back-end* menjadi inti dari sistem karena menangani permintaan pengguna dan menghasilkan respons yang sesuai. Tahapan pengembangan meliputi desain API, pengaturan logika bisnis, dan pengelolaan *database*. Pemilihan *framework* dan bahasa pemrograman yang tepat menentukan performa, keamanan, dan skalabilitas sistem[11]. Dalam konteks *e-office*, *back-end* berperan penting dalam pengelolaan data surat, arsip, dan autentikasi pengguna secara aman dan konsisten.

2.4. JSON (JavaScript Object Notation)

JSON merupakan format pertukaran data berbasis teks yang menampilkan struktur kunci-nilai sederhana, ringan, dan mudah dibaca oleh manusia maupun mesin. Format ini banyak digunakan dalam komunikasi antara *client* dan server karena efisien dan kompatibel dengan berbagai bahasa pemrograman. Dalam sistem *e-office*, JSON digunakan untuk mengirimkan data surat, arsip, dan status dokumen secara cepat dan konsisten. Keunggulannya meliputi fleksibilitas, kemudahan integrasi, serta dukungan terhadap data *real-time* yang membuat interaksi sistem menjadi lebih responsif dan dinamis[12].

2.5. MongoDB

MongoDB adalah basis data *non-relasional* (*NoSQL*) yang menyimpan data dalam bentuk dokumen JSON dan tidak memerlukan skema tetap. Struktur ini memungkinkan pengelolaan data yang fleksibel serta mendukung performa baca-tulis yang tinggi. Dalam sistem *e-office*, *MongoDB* digunakan untuk menyimpan surat masuk, surat keluar, arsip digital, dan data pengguna. Fitur *indexing* serta *query* kompleks memungkinkan pencarian data yang

cepat dan efisien. Selain itu, integrasinya dengan *framework* modern menjadikan *MongoDB* pilihan tepat untuk aplikasi administrasi yang membutuhkan skalabilitas dan kecepatan[13].

2.6. Node.js

Node.js adalah *platform runtime JavaScript* yang memungkinkan eksekusi kode di sisi server dengan arsitektur *event-driven* dan *non-blocking I/O*. Platform ini sangat cocok untuk aplikasi berskala besar yang memerlukan performa tinggi dan komunikasi *real-time*. *Node.js* juga memiliki ekosistem pustaka luas melalui NPM, sehingga memudahkan pengembang menambahkan modul sesuai kebutuhan. Dalam sistem *e-office*, *Node.js* berfungsi sebagai server *back-end* yang menangani API, autentikasi, serta koneksi ke *database*, menjadikan aplikasi lebih responsif dan efisien meskipun diakses secara bersamaan oleh banyak pengguna[14].

2.7. Express.js

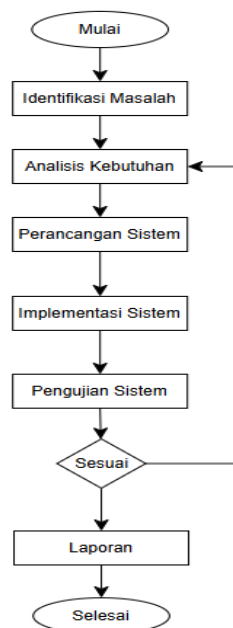
Express.js merupakan *framework* minimalis untuk *Node.js* yang mempermudah pembangunan aplikasi web dan *RESTful API*. *Framework* ini menyediakan fitur *routing*, *middleware*, dan manajemen *request-response* secara efisien. *Express.js* mendukung arsitektur *client-server* serta protokol *RESTful*, memungkinkan komunikasi data yang cepat dan aman antara *front-end* dan *back-end*. Dalam sistem *e-office*, *Express.js* digunakan untuk mengatur jalur data dan memastikan setiap permintaan diproses dengan benar, menjadikannya komponen penting dalam pengembangan sistem administrasi digital yang stabil dan terstruktur[15].

2.8. HTTPie

HTTPie merupakan alat berbasis *Command Line Interface (CLI)* yang digunakan untuk menguji API secara cepat dan mudah. Alat ini memfasilitasi pengiriman permintaan HTTP seperti GET, POST, PUT, dan DELETE dengan tampilan hasil yang terstruktur dan mudah dibaca. Dalam konteks pengembangan *e-office*, *HTTPie* digunakan untuk menguji fungsionalitas *endpoint* API, termasuk autentikasi pengguna dan pengelolaan data surat. Dengan antarmuka yang sederhana dan hasil respons yang jelas, *HTTPie* membantu memastikan keandalan serta konsistensi komunikasi antara server dan klien[16].

3. METODE PENGABDIAN MASYARAKAT

Kegiatan pengabdian masyarakat ini dilaksanakan di Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat dengan fokus pada implementasi *back-end* sistem informasi *e-office*. Metode pelaksanaan kegiatan dilakukan melalui pendekatan pengembangan sistem yang disesuaikan dengan kebutuhan instansi. Pengumpulan data dilakukan melalui observasi terhadap proses administrasi yang berjalan, diskusi dengan staf terkait, serta analisis kebutuhan terhadap pengelolaan data surat, disposisi, arsip, dan pengguna. Hasil dari proses tersebut digunakan sebagai dasar dalam merancang struktur *database*, *endpoint* API, serta fungsi utama sistem *back-end*.



Gambar 1. Diagram Alir Pelaksanaan Kegiatan

Gambar 1 menunjukkan pelaksanaan kegiatan dilakukan melalui beberapa tahapan. Tahap awal dimulai dari identifikasi masalah untuk mengetahui kendala yang dihadapi dalam proses administrasi. Selanjutnya, dilakukan analisis kebutuhan untuk menentukan fitur dan fungsi yang diperlukan dalam sistem. Hasil analisis tersebut kemudian digunakan dalam perancangan sistem, implementasi *back-end*, pengujian fungsi, dan penyusunan laporan sebagai dokumentasi akhir kegiatan. Uraian setiap tahapan pelaksanaan kegiatan adalah sebagai berikut.

3.1. Identifikasi Masalah

Tahap identifikasi masalah dilakukan untuk mengetahui kendala yang muncul dalam proses pengelolaan administrasi di Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat. Berdasarkan hasil observasi dan diskusi dengan staf, ditemukan beberapa permasalahan utama, yaitu proses pelacakan dokumen yang belum optimal, pengelolaan arsip yang masih perlu disusun lebih terstruktur, distribusi informasi yang membutuhkan waktu, serta perlunya sistem yang dapat membantu pencatatan dan pemantauan data secara lebih konsisten.

3.2. Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk menentukan kebutuhan fungsional dan *nonfungsional* sistem. Hasil analisis menunjukkan bahwa sistem membutuhkan beberapa fitur utama, yaitu pengelolaan data pengguna, pengelolaan surat, pengelolaan disposisi, pengelolaan arsip, autentikasi pengguna, serta pengaturan status data. Selain itu, sistem juga membutuhkan API sebagai penghubung antara *front-end* dan *back-end* agar proses pertukaran data dapat berjalan secara terintegrasi.

3.3. Perancangan Sistem

Tahap perancangan sistem dilakukan dengan menyusun rancangan struktur *back-end*, *database*, dan alur komunikasi data. Pada tahap ini, ditentukan entitas utama yang akan dikelola oleh sistem, seperti data pengguna, surat, disposisi, arsip, dan komunikasi internal. Setiap entitas dirancang dalam bentuk koleksi *database* agar data dapat disimpan dan dikelola secara sistematis.

3.4. Implementasi Sistem

Tahap implementasi sistem dilakukan dengan membangun *back-end* menggunakan *Node.js* dan *Express.js*. Pada tahap ini, *MongoDB* digunakan sebagai tempat penyimpanan data, sedangkan *Express.js* digunakan untuk mengatur *routing* dan *middleware*. Implementasi juga mencakup penerapan autentikasi pengguna agar sistem dapat membatasi akses berdasarkan data *login*.

3.5. Pengujian Sistem

Tahap pengujian sistem dilakukan untuk memastikan bahwa fungsi *back-end* berjalan sesuai dengan kebutuhan. Pengujian dilakukan menggunakan HTTPie dengan mengirimkan *request* ke *endpoint* API yang telah dibuat.

3.6. Penyusunan Laporan

Tahap penyusunan laporan dilakukan sebagai bentuk dokumentasi dari seluruh proses kegiatan, mulai dari identifikasi masalah, analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian. Laporan ini memuat hasil kegiatan pengembangan *back-end* sistem informasi *e-office* serta pembahasan mengenai fungsi-fungsi yang telah berhasil diimplementasikan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Analisis Kebutuhan

Berdasarkan hasil observasi dan diskusi dengan staf Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat, diperoleh beberapa kebutuhan utama dalam pengembangan sistem informasi *e-office*. Kebutuhan tersebut berkaitan dengan pengelolaan data pengguna, surat, disposisi, arsip, serta integrasi antara *front-end* dan *back-end*. Hasil analisis menunjukkan bahwa sistem perlu menyediakan fitur yang mampu membantu proses pencatatan, penyimpanan, pencarian, pembaruan, dan pengarsipan data secara lebih terstruktur.

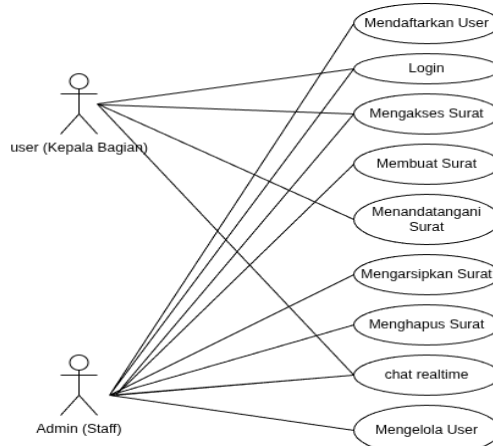
Dari sisi pengguna, sistem membutuhkan pembagian hak akses antara admin dan kepala bagian. Admin membutuhkan akses untuk mengelola data pengguna, data surat, arsip, disposisi, serta komunikasi internal. Sementara itu, kepala bagian membutuhkan akses untuk melihat, memproses, dan menindaklanjuti data surat atau disposisi sesuai kebutuhan administrasi. Pembagian hak akses ini diperlukan agar setiap pengguna hanya dapat menjalankan fungsi sesuai perannya, sehingga keamanan dan konsistensi data dapat lebih terjaga.

Dari sisi sistem, *back-end* harus mampu menyediakan layanan API yang dapat menghubungkan antarmuka pengguna dengan *database*. API tersebut digunakan untuk menjalankan proses CRUD pada data pengguna, surat, arsip, dan disposisi. Selain itu, sistem juga membutuhkan autentikasi pengguna, validasi data, serta penyimpanan

data yang fleksibel. Oleh karena itu, *Node.js* dan *Express.js* digunakan sebagai framework *back-end*, sedangkan *MongoDB* digunakan sebagai basis data untuk menyimpan data dalam bentuk dokumen.

4.2. Rancangan Sistem

Rancangan sistem dibuat sebagai dasar dalam proses implementasi *back-end* sistem informasi *e-office* dengan menggunakan *use case diagram*.



Gambar 2. Use Case Diagram Sistem Informasi E-office

Gambar 2 menggambarkan interaksi antara dua aktor utama dalam sistem informasi *e-office*, yaitu admin dan kepala bagian. Admin memiliki hak akses penuh terhadap sistem, meliputi pengelolaan pengguna, pengaturan status pengguna, pengelolaan surat, disposisi, arsip, serta komunikasi internal. Hak akses ini diberikan karena admin berperan sebagai pengelola sistem yang bertanggung jawab terhadap data utama dan pengaturan operasional aplikasi. Kepala bagian memiliki hak akses yang lebih terbatas sesuai kebutuhan administrasi, yaitu mengakses data surat, memproses disposisi, melihat arsip, serta menggunakan komunikasi internal. Kepala bagian tidak diberikan akses untuk mengelola pengguna karena fitur tersebut merupakan kewenangan admin. Pembagian hak akses ini bertujuan untuk menjaga keamanan data, membatasi perubahan data oleh pihak yang tidak berwenang, serta memastikan setiap pengguna menjalankan fungsi sesuai dengan perannya dalam sistem. Rancangan ini menggambarkan hubungan antara pengguna dengan fitur sistem, struktur data yang digunakan, serta alur komunikasi antara *front-end*, *back-end*, dan *database*.

4.3. Implementasi Pengembangan Sistem

Implementasi pengembangan sistem difokuskan pada pembuatan *back-end* yang mampu mengelola data pengguna, surat, disposisi, arsip, dan komunikasi internal. *Back-end* berfungsi sebagai pusat pemrosesan data yang menerima request dari *front-end*, memproses logika sistem, menghubungkan data dengan *MongoDB*, serta mengirimkan response kembali kepada pengguna. Implementasi ini dilakukan menggunakan *Node.js* dan *Express.js* sebagai framework server, serta *MongoDB* sebagai basis data.

Pada sistem informasi *e-office*, penyimpanan data dilakukan menggunakan *MongoDB*, yaitu basis data NoSQL yang menyimpan data dalam bentuk dokumen. Setiap data dikelompokkan ke dalam koleksi sesuai kebutuhan sistem. Koleksi utama yang digunakan meliputi data pengguna, surat, arsip, dan komunikasi internal.

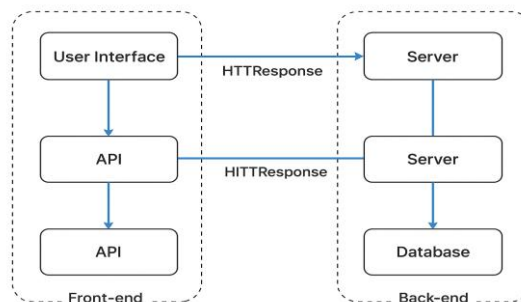
arsipschemas				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
4,10 kB	0	0 B	1	4,10 kB
chats				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20,48 kB	2	172,00 B	1	36,86 kB
suratschemas				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20,48 kB	3	476,00 B	1	36,86 kB
users				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20,48 kB	7	224,00 B	1	36,86 kB

Gambar 3. Data Koleksi Database

Pada Gambar 3 terlihat bahwa sistem memiliki beberapa koleksi utama yang digunakan untuk menyimpan data administrasi. Koleksi pengguna digunakan untuk menyimpan informasi akun dan hak akses; koleksi surat digunakan untuk menyimpan data surat yang diproses; koleksi arsip digunakan untuk menyimpan data surat yang telah diarsipkan; sedangkan koleksi komunikasi internal digunakan untuk mendukung pertukaran informasi antarpengguna. Struktur *database* ini dirancang agar data dapat tersimpan secara terorganisir dan mudah diakses melalui *endpoint* API.

4.3.1. Integrasi *front-end* dan *back-end*

Integrasi antara *front-end* dan *back-end* pada sistem informasi *e-office* dilakukan untuk memastikan aliran data yang konsisten dan responsif antara antarmuka pengguna dan *server*. Untuk mempermudah pemahaman, alur integrasi ini dapat digambarkan sebagai berikut.



Gambar 4. Alur Integrasi *Front-end* dan *Back-end* Sistem *E-office*

Gambar 4 menunjukkan alur integrasi antara *front-end* dan *back-end* pada sistem informasi *e-office*. Pengguna melakukan aksi melalui antarmuka, seperti *login*, menambahkan surat, memperbarui status, atau mengakses arsip. *Request* dari pengguna dikirimkan ke server melalui API, kemudian diproses oleh *controller* sesuai fungsi yang diminta. Setelah itu, server berinteraksi dengan *database MongoDB* untuk mengambil atau menyimpan data. Respons dari server dikirimkan kembali ke *front-end* agar hasil proses dapat ditampilkan kepada pengguna secara *real-time* dan konsisten.

4.3.2. Bukti implementasi sistem

Bukti implementasi sistem ditampilkan untuk menunjukkan bahwa rancangan *back-end* dan integrasi sistem telah diterapkan. Bukti implementasi ini meliputi tampilan *back-end* yang menunjukkan server atau endpoint API berjalan, serta tampilan *front-end* yang menunjukkan bahwa data hasil pemrosesan *back-end* dapat ditampilkan kepada pengguna.

```

trotsky@elGeeker:~/eOffice$ npm run dev
> e-office adpin pemprov ntb.@1.0.0 dev
> concurrently "npm run backend" "npm run frontend"
[1] > e-office adpin pemprov ntb.@1.0.0 frontend
[1] > npm run serve --prefix ./frontend
[1]
[0] > e-office adpin pemprov ntb.@1.0.0 backend
[0] > nodemon app.js
[0]
[1] > PLCNWKS@0.1.0 serve
[1] > vue-cli-service serve
[1]
[1] sh: 1: vue-cli-service: not found
[0] [nodemon] 2.0.22
[0] [nodemon] to restart at any time, enter `rs`
[0] [nodemon] watching path(s): *.*
[0] [nodemon] watching extensions: js,mjs,json
[0] [nodemon] starting `node app.js`
[1] npm run frontend exited with code 127
[0] (node:74916) [MONGODB] DeprecationWarning: Mongoose: the 'strictQuery' option will be switched back to 'false' by default in Mongoose 7. Use 'mongoose.set('strictQuery', false);' if you want to prepare for this change. Or use 'mongoose.set('strictQuery', true);' to suppress this warning.
[0] (Use 'node --trace-deprecation ...' to show where the warning was created)
[0] Server listening on port 3000
[0] MongoDB connected
  
```

Gambar 5. Tampilan Implementasi *Back-end* Sistem *E-office*

Gambar 5 menunjukkan hasil implementasi *back-end* sistem informasi *e-office*. Tampilan tersebut memperlihatkan bahwa server *back-end* telah berjalan. Implementasi *back-end* ini digunakan untuk menjalankan fungsi utama sistem, seperti register, *login*, pengelolaan data surat, pembaruan data pengguna, pengarsipan surat, serta komunikasi data dengan *database MongoDB*.


```
trotksy@elGeeker: $ http --session=user_session PUT :3000/api/users/68f4d9b3f6c40103a3540bf6/status status="active"
HTTP/1.1 200 OK
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: http://localhost:8080
Connection: keep-alive
Content-Length: 264
Content-Type: application/json; charset=utf-8
Date: Sun, 19 Oct 2025 12:38:15 GMT
ETag: W/"108-5u305fgMfnnVl8e3EWeR0ink4o4"
Keep-Alive: timeout=5
Vary: Origin
X-Powered-By: Express

{
  "data": {
    "_v": 0,
    "id": "68f4d9b3f6c40103a3540bf6",
    "email": "comm@mail.com",
    "jabatan": "Kepala Bagian Komunikasi",
    "password": "comm123",
    "registratonDate": "2025-10-19T12:28:42.676Z",
    "role": "pimpinan",
    "status": "active",
    "username": "kabagKomunikasi"
  },
  "message": "success"
}
```

Gambar 8. Implementasi HTTPie Update Status User

Pada Gambar 8 terlihat bahwa *request* PUT dikirimkan ke *endpoint* status user dengan parameter status baru, misalnya *active* atau *inactive*. Server kemudian memperbarui data pengguna di *database* dan mengembalikan respons berupa data yang telah diperbarui. Hasil ini menunjukkan bahwa sistem mampu mengelola status pengguna secara dinamis melalui *endpoint* API.

d. Tambah surat

Implementasi HTTPie pada fitur tambah surat digunakan untuk menguji proses penambahan data surat ke dalam *database*. Data surat yang dikirimkan dapat meliputi nomor surat, pengirim, penerima, sifat surat, lampiran, tanggal surat, disposisi, dan catatan.

```
trotksy@elGeeker: $ http --session=user_session -F POST :3000/api/submit/surat \
  lampiran@/home/trotksy/Downloads/Test.pdf \
  noSurat="001/INT/X/2025" \
  surat_dari="Bagian Umum" \
  diteruskanKepada="Kepala Bagian Protokol" \
  perihal="Pernohonan Rapat" \
  sifatSurat="Penting" \
  tglSurat="2025-10-19"
HTTP/1.1 201 Created
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: http://localhost:8080
Connection: keep-alive
Content-Length: 372
Content-Type: application/json; charset=utf-8
Date: Sun, 19 Oct 2025 12:51:16 GMT
ETag: W/"174-S2GKrV0Huzum00gfyC12q7c6rZU"
Keep-Alive: timeout=5
Vary: Origin
X-Powered-By: Express

{
  "data": {
    "_v": 0,
    "id": "68f4dec4f6c40103a3540c1b",
    "disposisi": [
      null
    ],
    "diteruskanKepada": [
      "Kepala Bagian Protokol"
    ],
    "lampiran": [
      "Test-1760878276436.pdf"
    ],
    "noSurat": "001/INT/X/2025",
    "parafLembarDisposisi": null,
    "sifatSurat": "Penting",
    "statusSurat": "dikirim",
    "surat_dari": "Bagian Umum",
    "tglSurat": "2025-10-19",
    "timestamp": "2025-10-19T12:51:16.463Z"
  },
  "message": "Surat created"
}
```

Gambar 9. Implementasi HTTPie Tambah Surat

Pada Gambar 9 terlihat bahwa *request* POST dikirimkan ke *endpoint* surat dengan data lengkap. Server memproses data tersebut dan menyimpannya ke dalam *database*. Jika proses berhasil, server mengembalikan respons berupa status berhasil dan data surat yang baru dibuat. Hasil ini menunjukkan bahwa fungsi penambahan surat pada *back-end* telah berjalan sesuai kebutuhan sistem.

4.4. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan seluruh fitur *back-end* sistem informasi *e-office* berjalan sesuai rancangan dan mampu mengelola data administrasi, surat, serta arsip secara akurat. Pengujian dilakukan

menggunakan *test case* yang dirancang berdasarkan kebutuhan fungsional sistem. Data simulasi digunakan untuk mempermudah verifikasi hasil pengujian. Setiap *test case* memuat langkah pengujian, hasil yang diharapkan, serta hasil aktual yang diperoleh dari *database*. Hasil implementasi pengujian pada fungsionalitas untuk sistem informasi *e-office* adalah sebagai berikut.

TABEL 1. IMPLEMENTASI PENGUJIAN DENGAN *TEST CASE*

<i>Test case ID</i>	Fitur	Deskripsi	<i>Input/Data Simulasi</i>	<i>Expected Result</i>	<i>Actual Result</i>
TC01	<i>Login User</i>	Memastikan <i>user</i> dengan status <i>active</i> bisa <i>login</i>	<i>Email: admin@mail.com</i> <i>Password: admin</i>	<i>Login</i> berhasil, sistem mengembalikan data <i>user</i> dan token	Sesuai
TC02	<i>Login User</i>	<i>User inactive</i> mencoba <i>login</i>	<i>Email: inactive@mail.com</i> <i>Password: admin</i>	<i>Login</i> gagal, muncul pesan akun <i>inactive</i>	Sesuai
TC03	<i>Logout User</i>	Menghapus sesi <i>login</i>	-	Cookie/session dihapus, <i>user logout</i> berhasil	Sesuai
TC04	Register <i>User</i>	Menambahkan <i>user</i> baru	<i>Username: testUser</i> <i>Email: test@mail.com</i> <i>Password: test123</i> <i>Role: kepala_bagian</i> <i>Jabatan: Kepala Bagian</i>	<i>User</i> tersimpan di <i>database</i> , status <i>default 'inactive'</i>	Sesuai
TC05	Ambil Semua <i>User</i>	Mengambil daftar seluruh <i>user</i>	-	Response berisi list <i>user</i>	Sesuai
TC06	Ambil <i>User By Status</i>	Mengambil <i>user</i> dengan status tertentu	Status: <i>active</i>	Response berisi list <i>user</i> aktif	Sesuai
TC07	<i>Update Status User</i>	Mengubah status <i>user active/inactive</i>	ID <i>user</i> , Status: <i>active</i>	Status <i>User</i> berhasil diperbarui	Sesuai
TC08	Hapus <i>User</i>	Menghapus <i>user</i> berdasarkan ID	ID <i>user</i>	<i>User</i> dihapus dari <i>database</i>	Sesuai
TC09	Buat Surat Baru	Menambahkan surat baru ke <i>database</i>	Surat_dari: Admin NoSurat: 001/ADM/X/2025 TglSurat: 2025-10-05 DiteruskanKepada: [Kepala Bagian] Disposisi: [Diperiksa] Catatan: Contoh surat	Surat tersimpan, status surat = 'dikirim', lampiran & paraf disimpan jika ada	Sesuai
TC10	Ambil Semua Surat	Mengambil daftar seluruh surat	-	Response berisi semua surat	Sesuai
TC11	Ambil Surat <i>By ID</i>	Mengambil satu surat berdasarkan IDE	ID surat	Response berisi data surat sesuai ID	Sesuai
TC12	<i>Update Surat</i>	Mengubah <i>field</i> tertentu pada surat	ID surat, Catatan: <i>update pengujian</i>	<i>Field</i> catatan terUpdate di <i>database</i>	Sesuai
TC13	Hapus Surat	Menghapus surat berdasarkan IDE	ID surat	Surat dihapus, tidak muncul di <i>database</i>	Sesuai
TC14	<i>Upload Lampiran Surat</i>	Menambahkan <i>file</i> lampiran untuk surat	ID surat, <i>file</i> lampiran (PDF/PNG)	Lampiran tersimpan di folder <i>uploads</i> , nama <i>file</i> tersimpan di <i>database</i>	Sesuai
TC15	Upload Paraf Lembar Disposisi	Menambahkan <i>file</i> paraf lembar disposisi	ID surat, <i>file</i> paraf (PDF)	File tersimpan di folder <i>uploads</i> , <i>field</i> parafLembarDisposisi terUpdate	Sesuai

<i>Test case ID</i>	Fitur	Deskripsi	<i>Input/Data Simulasi</i>	<i>Expected Result</i>	<i>Actual Result</i>
TC16	Buat Arsip	Menambahkan arsip baru untuk surat tertentu	ID surat, Status: diarsipkan	Arsip tersimpan, data surat terhubung (<i>populate</i>)	Sesuai
TC17	Ambil Semua Arsip	Mengambil daftar seluruh arsip	-	<i>Response</i> berisi daftar arsip beserta data surat terkait	Sesuai
TC18	Ambil Arsip By ID	Mengambil arsip tertentu	ID arsip	<i>Response</i> berisi data arsip sesuai ID	Sesuai
TC19	<i>Update Status Lampiran</i>	Mengubah status lampiran surat	ID surat, StatusLampiran: lengkap	<i>Field</i> statusLampiran terUpdate di <i>database</i>	Sesuai

Tabel I merupakan ringkasan *test case* pengujian sistem informasi *e-office*. Tabel ini menampilkan berbagai skenario pengujian pada fitur *back-end*, mulai dari pengelolaan data surat, arsip, hingga manajemen *user*.

4.5. Dokumentasi Pengabdian



Gambar 10. Dokumentasi Implementasi *Back-end* Sistem *E-office*

Gambar 10 menunjukkan dokumentasi dari proses identifikasi masalah, implementasi dan pengujian sistem yang mencakup penerapan logika bisnis pada *back-end*, integrasi API dengan database, serta pengujian konektivitas antara komponen sistem.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan kegiatan pengabdian masyarakat yang dilakukan di Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat, hasil analisis menunjukkan bahwa sistem membutuhkan fitur utama berupa pengelolaan pengguna, pengelolaan surat, disposisi, arsip, autentikasi pengguna, serta integrasi data antara *front-end* dan *back-end*. Berdasarkan kebutuhan tersebut, dilakukan perancangan sistem yang meliputi perancangan struktur *database*, perancangan *endpoint* API, serta alur komunikasi antara antarmuka pengguna, *server*, dan *database*.

Implementasi sistem dilakukan dengan mengembangkan *back-end* menggunakan *Node.js* dan *Express.js* sebagai *framework server*, serta *MongoDB* sebagai basis data. *Back-end* yang dikembangkan mampu menjalankan fungsi *Create, Read, Update, and Delete* (CRUD) pada data pengguna, surat, dan arsip. Selain itu, sistem juga mendukung proses autentikasi pengguna, pembaruan status *user*, penyimpanan data surat, serta pengarsipan dokumen. Integrasi antara *back-end* dan *database* memungkinkan data dikelola secara lebih konsisten dan mudah diakses melalui *endpoint* API.

Hasil pengujian menggunakan HTTPie dan *test case* menunjukkan bahwa fitur-fitur utama sistem telah berjalan sesuai dengan kebutuhan yang dirancang. Fitur *register*, *login*, *update status user*, tambah surat, ambil data surat, *update* surat, hapus surat, *upload* lampiran, serta pengelolaan arsip menunjukkan hasil yang sesuai dengan *expected result*. Dengan demikian, implementasi *back-end* sistem informasi *e-office* berhasil mendukung proses pengelolaan data administrasi secara lebih efektif, terstruktur, dan aman.

5.2. Saran

Adapun saran yang diberikan penulis agar lebih baik di masa yang akan datang adalah sebagai berikut:

- Menambahkan fitur autentikasi dan otorisasi yang lebih lengkap, misalnya pembatasan akses berdasarkan jabatan atau peran, untuk meningkatkan keamanan data.
- Melakukan pengujian sistem lebih luas, termasuk pengujian beban (*load testing*) dan pengujian keamanan, agar sistem lebih stabil dan aman digunakan pada kondisi operasional nyata.
- Membuat dokumentasi teknis dan manual penggunaan *back-end*, sehingga memudahkan pegawai dalam memahami alur sistem dan pengoperasiannya secara mandiri.

UCAPAN TERIMA KASIH

Terima kasih kepada Program Studi Teknik Informatika Universitas Mataram dan seluruh staf/pimpinan Biro Administrasi Pimpinan Provinsi Nusa Tenggara Barat atas fasilitas dan kerja samanya selama pelaksanaan kegiatan.

DAFTAR PUSTAKA

- M. J. Aw, "Dampak E-Government dalam Meningkatkan Efisiensi dan Efektivitas Tata Kelola Pemerintahan," *Politika Progresif: Jurnal Hukum, Politik dan Humaniora*, vol. 1, no. 4, pp. 204–226, 2024.
- D. P. Cantica, A. H. Jatmika, R. B. Huwae, and D. Ratnasari, "Analisis sistem informasi Dinas Perpustakaan dan Kearsipan Provinsi Nusa Tenggara Barat menggunakan System Usability Scale," *Jurnal Begawe Teknologi Informasi (JBegaTI)*, vol. 5, no. 1, pp. 76–84, 2024.
- M. D. Zaidan, "Pengembangan Back End Aplikasi Sistem Informasi Terpadu BSI Pinter menggunakan Spring Framework," Universitas Islam Indonesia, 2024.
- E. Maiyana and others, "Arsip yang Berbicara: Implementasi Sistem Informasi Pengelolaan Surat Berbasis Laravel untuk Efisiensi Organisasi di SMA Negeri 2 Lubuk Basung," *Petik: Jurnal Pendidikan Teknologi Informasi Dan Komunikasi*, vol. 11, no. 2, pp. 149–161, 2025.
- M. D. C. Nugraha, R. Afwani, and A. Y. Husodo, "PENGEMBANGAN BACKEND SCALABLE UNTUK APLIKASI DIET BERBASIS MICROSERVICES DENGAN KUBERNETES".
- R. Lisgiani and S. Nurmajid, "Implementasi Autentikasi Dari Sisi Backend Pada Arsitektur Microservices Menggunakan Express JS," *Infotronik: Jurnal Teknologi Informasi dan Elektronika*, vol. 7, no. 1, pp. 27–32, 2022.
- V. Azkarin, R. G. Guntara, and O. Herdiana, "Development of a rest api for human resource information system for employee referral management domain using the express js framework and node. js," *Journal of Scientific Research, Education, and Technology (JSRET)*, vol. 2, no. 3, pp. 1085–1094, 2023.
- H. Nopriandi, "Perancangan sistem informasi registrasi mahasiswa," *Jurnal Teknologi Dan Open Source*, vol. 1, no. 1, pp. 73–79, 2018.
- B. D. Abdillah, "Peran Teknologi Informasi dalam Transparansi dan Akuntabilitas Pemerintahan," *literacy notes*, vol. 2, no. 1, 2024.
- M. Raihan and Y. Fatimah, "IMPLEMENTASI APLIKASI 'E-OFFICE' DALAM MENINGKATKAN KINERJA PEGAWAI PADA DINAS KOMUNIKASI INFORMASI DAN STATISTIK KABUPATEN BARRU PROVINSI SULAWESI SELATAN," Institut Pemerintahan Dalam Negeri, 2025.
- A. Mubariz, D. Nur, E. Tungadi, and M. N. Y. Utomo, "Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node. JS (Studi Kasus: Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang)," in *Seminar Nasional Teknik Elektro Dan Informatika (SNTEI)*, 2020, pp. 72–77.
- A. Ramadhan, "Implementasi Json Web Token dalam Pengembangan Basis Data Monografi Desa Berbasis Rest Api," Universitas Islam Indonesia, 2024.
- S. T. Lizda Iswari, "Implementasi Mongo Db, Express Js, React Js Dan Node Js (Mern) Pada Pengembangan Aplikasi Formulir, Kuis, Dan Survei Online," 2022.
- A. Ramadhani, N. Iriadi, and R. Hidayat, "Implementasi Teknologi Rest API Dengan Node Js Untuk Aplikasi Rekomendasi Destinasi Wisata," *Indonesian Journal Computer Science*, vol. 4, no. 1, pp. 22–29, 2025.
- M. Siahaan and R. W. Wijaya, "Performance comparison between Laravel and ExpressJs framework using apache JMeter," *Journal Of Informatics And Telecommunication Engineering*, vol. 7, no. 2, pp. 545–554, 2024.
- J. Roztocil, "HTTPie CLI Docs," HTTPie Documentation [Daring]. Tersedia pada: <https://httpie.io/docs/cli> [Diakses: 03-Feb-2026].