



BUPATI LEMBATA
PROVINSI NUSA TENGGARA TIMUR

KEPUTUSAN BUPATI LEMBATA
NOMOR 418 TAHUN 2025

TENTANG

STANDAR TEKNIS DAN PROSEDUR PEMBANGUNAN DAN
PENGEMBANGAN APLIKASI KHUSUS DI LINGKUNGAN
PEMERINTAH KABUPATEN LEMBATA

BUPATI LEMBATA,

- Menimbang : a. bahwa untuk melaksanakan ketentuan Pasal 16 ayat (2) huruf b dan Pasal 18 Peraturan Bupati Lembata Nomor 82 Tahun 2022 tentang Sistem Pemerintahan Berbasis Elektronik di Lingkungan Pemerintah Kabupaten Lembata, perlu menetapkan Keputusan Bupati Lembata tentang Standar Teknis dan Prosedur Pembangunan dan Pengembangan Aplikasi Khusus di Lingkungan Pemerintah Kabupaten Lembata;
- Mengingat : 1. Undang-Undang Nomor 52 Tahun 1999 tentang Pembentukan Kabupaten Lembata (Lembaran Negara Republik Indonesia Tahun 1999 Nomor 180, Tambahan Lembaran Negara Republik Indonesia Nomor 3901) sebagaimana telah diubah dengan Undang-Undang Nomor 12 Tahun 2000 tentang Perubahan Atas Undang-Undang Nomor 52 Tahun 1999 tentang Pembentukan Kabupaten Lembata (Lembaran Negara Republik Indonesia Tahun 2000 Nomor 79 Tambahan Lembaran Negara Republik Indonesia Nomor 3967);
2. Undang-Undang Nomor 23 Tahun 2014 tentang Pemerintah Daerah (Lembaran Negara Republik Indonesia Tahun 2014 Nomor 244, Tambahan Lembaran Negara Republik Indonesia Nomor 5587)

sebagaimana telah diubah beberapa kali terakhir dengan Undang-Undang Nomor 9 Tahun 2015 tentang Perubahan Kedua Atas Undang-Undang Nomor 23 Tahun 2014 tentang Pemerintah Daerah (Lembaran Negara Republik Indonesia Tahun 2015 Nomor 58, Tambahan Lembaran Negara Republik Indonesia Nomor 5679);

3. Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik (Lembaran Negara Republik Indonesia Tahun 2018 Nomor 182);
4. Peraturan Menteri Komunikasi dan Digital Nomor 6 Tahun 2025 tentang Standar Teknis dan Prosedur Pembangunan dan Pengembangan Aplikasi Sistem Pemerintahan Berbasis Elektronik (Berita Negara Republik Indonesia Tahun 2025 Nomor 245);
5. Peraturan Bupati Lembata Nomor 82 Tahun 2022 tentang Sistem Pemerintahan Berbasis Elektronik di Lingkungan Pemerintah Kabupaten Lembata (Berita Daerah Kabupaten Lembata Tahun 2022 Nomor 82);

MEMUTUSKAN

- Menetapkan :
- KESATU : Standar Teknis dan Prosedur Pembangunan dan Pengembangan Aplikasi Khusus di Lingkungan Pemerintah Kabupaten Lembata dengan rincian sebagaimana tercantum dalam Lampiran Keputusan ini.
- KEDUA : Standar teknis dan prosedur sebagaimana dimaksud pada Diktum KESATU digunakan sebagai panduan penyelenggaraan sistem pemerintahan berbasis elektronik dalam pembangunan dan pengembangan aplikasi khusus secara aman, efektif, dan efisien, sehingga dapat mengoptimalkan pelayanan publik serta mendukung pembangunan berkelanjutan.
- KETIGA : Standar teknis dan prosedur sebagaimana dimaksud dalam Diktum KESATU, meliputi:
- a. prosedur pembangunan dan pengembangan aplikasi khusus;
 - b. database;
 - c. pemrograman *back end*;
 - d. pemrograman *front end*; dan desain template.
- KEEMPAT : Segala biaya yang timbul akibat ditetapkannya

Keputusan ini dibebankan pada Anggaran Pendapatan dan Belanja Daerah Kabupaten Lembata.
KELIMA : Keputusan ini mulai berlaku sejak tanggal ditetapkan.

Ditetapkan di Lewoleba
pada tanggal, 9 september 2025

BUPATI LEMBATA,



P. KANISIUS TUAQ

LAMPIRAN
KEPUTUSAN BUPATI LEMBATA
NOMOR 418 TAHUN 2025
TENTANG STANDAR TEKNIS DAN
PROSEDUR PEMBANGUNAN DAN
PENGEMBANGAN APLIKASI KHUSUS DI
LINGKUNGAN PEMERINTAH KABUPATEN
LEMBATA

STANDAR TEKNIS DAN PROSEDUR PEMBANGUNAN DAN
PENGEMBANGAN APLIKASI KHUSUS DI LINGKUNGAN PEMERINTAH
KABUPATEN LEMBATA

BAB I
PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi informatika yang sangat pesat telah berhasil mengubah kondisi sosial masyarakat. Berbagai layanan yang dapat diakses dengan nyaman tanpa berpindah tempat menjadikan masyarakat semakin paham potensi yang dimiliki teknologi informatika dalam memenuhi kebutuhan. Tentunya hal ini juga menuntut pemerintah untuk memanfaatkan potensi perkembangan teknologi informasi dan komunikasi tersebut dalam penyelenggaraan pemerintahan dan pelayanan publik.

Pemanfaatan teknologi informasi dan komunikasi oleh pemerintah tidak sekedar dilakukan digitalisasi atau elektronisasi layanan saja, melainkan mengoptimalkan layanan publik yang berkualitas dan terpercaya berbasis elektronik. Untuk itu ada banyak hal yang perlu dilakukan penyesuaian bahkan penggantian. Perubahan layanan tersebut menuntut adanya alur proses bisnis dan prosedur yang berubah pula. Hal ini berimbas pada perubahan tata kelola layanan pemerintahan, sehingga perlu adanya kejelasan hukum melalui kebijakan-kebijakan internal terkait sebagai payung hukum dalam penyelenggaraan pelayanan berbasis elektronik.

Sebagai tanggapan atas tuntutan tersebut Pemerintah telah menerbitkan Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik, yang bertujuan untuk mewujudkan tata kelola pemerintahan yang bersih, efektif, transparan dan akuntabel, serta pelayanan publik yang berkualitas dan terpercaya. Selanjutnya diterbitkan Peraturan Presiden Nomor 132 Tahun 2022 tentang Arsitektur SPBE Nasional yang dijadikan

pedoman dalam penyusunan Arsitektur SPBE bagi Instansi Pusat dan Pemerintah Daerah. Dalam Peraturan Presiden Nomor 132 Tahun 2022 tersebut mengamanatkan agar SPBE dapat diarahkan menjadi pondasi pelaksanaan *Government as a platform*, yang merupakan pondasi ekosistem digital yang memungkinkan pemerintah dapat memberikan pelayanan publik yang berkualitas dengan menciptakan pemerintahan yang bersih, efektif, efisien, netral, profesional, dan berlandaskan hukum, dengan didukung oleh pilar Aparatur Sipil Negara, kelembagaan dan proses bisnis organisasi, serta pelaksanaan akuntabilitas kinerja dan pengawasan.



Gambar 1. SPBE sebagai *Government as a Platform*.

Menindaklanjuti ketentuan Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik dan Peraturan Presiden Nomor 132 Tahun 2022 tentang Arsitektur Sistem Pemerintahan Berbasis Elektronik, Pemerintah Kabupaten Lembata telah menetapkan Peraturan Bupati Lembata Nomor 82 Tahun 2022 tentang Sistem Pemerintahan Berbasis Elektronik di Lingkungan Pemerintah Kabupaten Lembata, agar pembangunan SPBE menjadi lebih terarah, tertata, kolaboratif dan integratif, menuju terciptanya layanan publik yang handal, efektif, efisien, transparan dan akuntabel serta partisipatif.

B. Maksud dan Tujuan

1. Maksud penyusunan standar teknis dan prosedur ini adalah sebagai pedoman yang menjadi acuan dalam membangun atau mengembangkan aplikasi khusus untuk meningkatkan efisiensi, kualitas, dan keamanan serta fleksibilitas layanan.

2. Tujuan penyusunan standar teknis dan prosedur ini adalah untuk memastikan kualitas, konsistensi, efisiensi, dan keamanan dalam proses pembangunan/pengembangan aplikasi sehingga aplikasi yang dihasilkan memiliki kualitas yang baik, dapat diandalkan, dan mudah diintegrasikan dengan sistem lainnya.

C. Ruang Lingkup

Ruang lingkup standarisasi pembangunan dan pengembangan aplikasi khusus di lingkungan Pemerintah Kabupaten Lembata meliputi:

1. Prosedur pembangunan dan pengembangan aplikasi khusus;
2. *Database*;
3. Pemrograman *back end*;
4. Pemrograman *front end*; dan
5. *Desain template*.

BAB II PROSEDUR PEMBANGUNAN / PENGEMBANGAN APLIKASI KHUSUS

A. Pendahuluan

Prosedur pembangunan/pengembangan aplikasi khusus di lingkungan Pemerintah Kabupaten Lembata perlu disusun sebagai pedoman pembangunan/pengembangan aplikasi agar aman, efektif, dan efisien. Prosedur ini berlaku untuk pembangunan/pengembangan aplikasi khusus yang dilaksanakan secara internal dan/atau menggunakan pihak ketiga.

B. Tahapan

Dalam pembangunan atau pengembangan aplikasi khusus harus memenuhi pedoman Manajemen Perubahan SPBE, yang melalui serangkaian proses yaitu: perencanaan, analisis, perancangan, pengembangan, pengujian, implementasi, pemeliharaan, pemantauan dan evaluasi.

B.1. Perencanaan

Perencanaan merupakan proses identifikasi dalam menentukan prioritas pengembangan aplikasi khusus di lingkungan Pemerintah Kabupaten Lembata dalam jangka pendek, berkisar 1 (satu) tahun yang menjadi acuan dalam menentukan urutan pengerjaan pembangunan/ pengembangan aplikasi khusus. Dokumen perencanaan ditetapkan pada awal tahun sebagai pedoman dalam menentukan urutan pembangunan/pengembangan aplikasi yang akan dikerjakan dan dapat direviu pada semester pertama untuk menyesuaikan dengan kondisi dan kebutuhan yang belum terpetakan maupun kebutuhan atau kondisi khusus yang ditemukan karena adanya perubahan kebijakan maupun proses bisnis di perangkat daerah pengampu aplikasi khusus.

Daftar dokumen perencanaan dapat diperoleh melalui:

1. Peta Rencana SPBE

Daftar aplikasi didasarkan pada rencana proyek implementasi SPBE pada peta jalan SPBE mengacu pada arsitektur SPBE Kabupaten Lembata.

2. Permohonan dari Perangkat Daerah

Permohonan dari perangkat daerah yang didasarkan pada kebutuhan perangkat daerah dalam mendukung tugas pokok dan fungsinya untuk memberikan layanan secara elektronik, yang belum terpetakan dalam peta jalan SPBE.

B.2. Analisis

Proses analisis digunakan sebagai langkah awal dalam menentukan pekerjaan yang sudah ditetapkan dalam dokumen perencanaan untuk dilaksanakan. Fase ini menggambarkan permasalahan kondisi yang ada dan memetakan masalah untuk mendapatkan pemahaman yang lebih baik dari sistem yang akan dikembangkan. Hasil dari fase ini akan menyampaikan permasalahan yang terjadi, penyebab, dampak, dan solusi yang akan diberikan untuk mengatasi permasalahan tersebut. Proses analisis paling sedikit mencakup:

1. Analisis Kelayakan

Analisis kelayakan adalah suatu proses untuk mempelajari secara mendalam daftar aplikasi khusus yang akan dibangun/dikembangkan untuk menentukan layak atau tidaknya aplikasi tersebut dikerjakan. Aspek yang perlu diperhatikan dalam analisis kelayakan meliputi:

a. Kesiapan

Bertujuan untuk memastikan kesiapan sumber daya, data, dan regulasi dari perangkat daerah pengampu aplikasi, yang akan digunakan sebagai bahan awal pembangunan/pengembangan aplikasi khusus.

b. Teknologi

Bertujuan untuk memastikan aplikasi yang akan dibangun/dikembangkan bisa dikerjakan menggunakan teknologi tersedia secara aman dan layak.

c. Hukum

Bertujuan untuk memastikan bahwa aplikasi khusus yang akan dibangun/dikembangkan sudah sesuai dan tidak menyalahi ketentuan perundang-undangan yang berlaku.

d. Dampak dan Manfaat

Bertujuan untuk memastikan bahwa aplikasi khusus yang akan dibangun/dikembangkan memiliki dampak positif

yang signifikan terhadap tugas dan fungsi perangkat daerah pengampu, serta memiliki manfaat bagi penerima layanan maupun proses tata kelola perangkat daerah pengampu layanan.

2. Analisis Kebutuhan

Analisis kebutuhan adalah proses pengumpulan kebutuhan-kebutuhan pengguna dari semua elemen sistem perangkat lunak yang akan di bangun/dikembangkan, untuk kemudian dianalisis dan dibentuk spesifikasi kebutuhan perangkat lunak, fungsi perangkat lunak yang dibutuhkan, performansi (unjuk kerja) sistem perangkat lunak, penjadwalan proyek, identifikasi sumber daya (sumber daya manusia dan perangkat keras serta perangkat lunak pendukung yang dibutuhkan) dan estimasi jangka waktu pengerjaan pembangunan/pengembangan aplikasi khusus.

B.3. Perancangan

Perancangan adalah penentuan proses dan data yang diperlukan oleh sistem baru berdasarkan data dan informasi yang sudah dikumpulkan dalam proses analisis. Perancangan bertujuan untuk memberikan gambaran rancangan bangun yang lengkap sebagai pedoman bagi *programmer* dalam mengembangkan aplikasi khusus. Perancangan dalam pembangunan/pengembangan aplikasi khusus paling sedikit meliputi perancangan *database* dan perancangan antarmuka pengguna.

B.4. Pengembangan

Pengembangan merupakan proses pengkodean hasil perancangan yang sudah dilaksanakan pada tahapan sebelumnya ke dalam wujud aplikasi khusus yang akan digunakan oleh pengguna.

B.5. Pengujian

Pengujian adalah proses untuk memeriksa aplikasi sudah memenuhi kualitas sesuai dengan kebutuhan pada perancangan tahap sebelumnya, serta memastikan bahwa produk yang telah dibuat bebas dari kesalahan.

Pengujian paling sedikit meliputi:

1. Pengujian Fungsional (*Functional Testing*)

Pengujian fungsional adalah proses pengujian *software* yang dilakukan secara manual untuk mempelajari apakah fitur dalam sebuah aplikasi berfungsi baik atau tidak. Dalam proses ini biasanya mencakup verifikasi sebuah fitur yang tertera dalam dokumen persyaratan. Metode yang digunakan adalah sebagai berikut:

- a. *White Box Testing*, yaitu pengujian yang berfokus pada komponen di dalam aplikasi khusus seperti desain, struktur *coding*, dan cara kerja dari awal hingga akhir. Penguji yang menggunakan metode *white box* dalam pengujian aplikasi khusus harus memiliki pengetahuan atau pemahaman penuh mengenai sumber kode aplikasi khusus tersebut.
- b. *Black Box Testing*, yaitu pengujian yang dilakukan pada antarmuka (*interface*) aplikasi khusus, yang bertujuan untuk mendemonstrasikan fungsi aplikasi khusus tersebut apakah *input* diterima dengan benar dan *output* yang dihasilkan adalah benar.

2. Pengujian Penerimaan Pengguna (*User Acceptance Test*)

Pengujian penerimaan pengguna dilakukan untuk mengukur penerapan dan kegunaan aplikasi khusus dalam melakukan tugas dan fungsi layanan yang didukung aplikasi khusus tersebut sesuai dengan analisis kebutuhan pada tahap sebelumnya. Proses ini menguji apa yang menurut pengguna harus dilakukan oleh aplikasi khusus tersebut dibandingkan dengan apa yang harus dilakukan oleh aplikasi khusus tersebut sesuai persyaratan terdokumentasi.

3. Pengujian Kinerja (*Performance Testing*)

Pengujian kinerja dilakukan untuk menguji performansi dari kinerja aplikasi khusus, seperti kecepatan dalam memberi respons, stabilitas, dan penggunaan sumber daya aplikasi khusus di bawah beban kerja tertentu, untuk memastikan bahwa aplikasi dapat berjalan dengan baik sampai pada batas beban yang sudah ditentukan berdasarkan potensi jumlah pengguna.

4. Pengujian Keamanan

Pengujian keamanan dilakukan untuk menguji aplikasi khusus tidak terbuka bagi peretas atau jenis kerentanan berbahaya lainnya yang mungkin dieksploitasi untuk masuk ke dalam akses layanan, atau menyebabkan aplikasi khusus tidak berfungsi dengan benar serta melindungi data dan informasi sensitif.

B.6. Implementasi

Tahap implementasi merupakan penerapan dan operasional aplikasi khusus oleh pengguna pada lingkungan kerja yang sebenarnya, setelah selesai dianalisis, dirancang, dibangun, dan diuji sesuai dengan hasil perancangan dan desain pada tahap sebelumnya. Supaya implementasi dapat berjalan dengan baik sesuai dengan fungsi dan tujuan organisasi, perangkat daerah pengampu layanan wajib memenuhi hal-hal sebagai berikut:

1. Membentuk tim pengelola aplikasi khusus yang ditetapkan dalam keputusan kepala perangkat daerah, paling sedikit terdiri dari:
 - a. Koordinator
Berasal dari salah satu pejabat struktural/fungsional yang setara dalam lingkup perangkat daerah pengampu layanan, dan memiliki tugas jabatan yang berkaitan langsung dengan layanan tersebut. Koordinator bertugas untuk merumuskan dan mereviu kebijakan serta tata kelola layanan.
 - b. Pelaksana Teknis
Terdiri dari ASN yang memiliki tugas jabatan berkaitan langsung dengan fungsi dari layanan aplikasi.
 - c. Pelaksana Sosialisasi
Berasal dari ASN yang memiliki kompetensi terkait operasionalisasi layanan aplikasi. Pelaksana sosialisasi mempunyai tugas dan tanggung jawab dalam membuat media sosialisasi dan melaksanakan sosialisasi kepada calon dan pengguna layanan, agar aplikasi khusus yang selesai dibangun/dikembangkan dapat diterima dan digunakan secara luas.
 - d. Layanan Pengguna (*Customer Service*)
Layanan pengguna bertanggung jawab menerima permasalahan dan memberikan solusi bagi pengguna

layanan yang mengalami kesulitan dalam menggunakan layanan aplikasi khusus.

2. Menyediakan panduan penggunaan bagi penerima layanan, baik dalam bentuk buku manual dan/atau video tutorial, yang dapat diakses melalui halaman awal aplikasi.
3. Menyediakan dan menyebarkan media publikasi kepada target pengguna layanan melalui forum atau media sosial elektronik untuk mensosialisasikan aplikasi agar dapat digunakan oleh penerima layanan.
4. Memberikan pelatihan kepada admin dan operator yang akan menjalankan aplikasi khusus agar dapat menjalankan fungsinya dalam memberikan layanan kepada penerima layanan.

B.7. Pemeliharaan

Pemeliharaan (*maintenance*) aplikasi khusus adalah suatu proses mengubah, memodifikasi, dan meningkatkan aplikasi khusus tersebut agar tetap sesuai dengan kebutuhan penggunanya. Proses ini dilakukan untuk meningkatkan performa aplikasi khusus secara keseluruhan serta mengidentifikasi dan memperbaiki kesalahan sistem atau *bug* yang ditemukan oleh pengguna layanan. Tujuan dari pemeliharaan adalah menjaga aplikasi khusus berjalan secara baik, optimal dan aman selama digunakan. Ketentuan tentang pemeliharaan (*maintenance*) aplikasi khusus akan diatur lebih lanjut dalam peraturan lainnya.

B.8. Pemantauan dan Evaluasi.

Pemantauan dan evaluasi aplikasi khusus dilaksanakan secara rutin untuk mengetahui keberlangsungan penggunaan aplikasi khusus, baik dari segi kebutuhan, tingkat penerimaan pengguna, performa, kendala yang dihadapi, dan potensi untuk meningkatkan nilai guna aplikasi khusus.

Monitoring dan evaluasi dapat dilakukan dengan mengamati performa aplikasi dan pertumbuhan data, serta pengamatan dan wawancara kepada stakeholder aplikasi khusus. Apabila tidak ada pertumbuhan data selama 3 (tiga) bulan terakhir, Dinas Komunikasi dan Informatika Kabupaten Lembata berhak untuk menonaktifkan aplikasi khusus dari pusat data.

C. Arsitektur

Pembangunan atau pengembangan aplikasi khusus di lingkungan Pemerintah Kabupaten Lembata menggunakan model arsitektur yang disesuaikan dengan jenis aplikasi khusus yang akan dibangun atau dikembangkan, sebagai berikut:

1. Arsitektur Monolitik

Arsitektur monolitik adalah arsitektur aplikasi khusus dimana sebuah aplikasi khusus dibangun dalam satu *codebase* yang sama menggunakan satu bahasa pemrograman untuk menghemat sumber daya, mempercepat proses pengembangan, serta mempermudah proses pemeliharannya.

Arsitektur monolitik digunakan untuk aplikasi sederhana, yaitu aplikasi yang hanya memuat kurang dari 3 (tiga) modul utama atau memiliki proses bisnis yang sederhana tanpa adanya proses dan komputasi yang kompleks.

2. Arsitektur *Backend-Frontend*

Back end merupakan bagian yang berjalan di sisi server dan tidak dapat diakses langsung oleh pengguna. Bagian ini bertanggung jawab untuk memproses permintaan, mengelola dan menyimpan data, serta melakukan logika bisnis yang kompleks. *Front end* adalah bagian dari aplikasi yang dapat dilihat dan berinteraksi langsung dengan pengguna. *Front end* terdiri dari antarmuka pengguna (*User Interface/UI*) yang meliputi tampilan, tata letak, navigasi, dan elemen interaktif lainnya seperti tombol dan formulir. Dalam proses perancangannya aplikasi khusus dapat dibuat dengan menggunakan komponen-komponen atau pelayanan yang sudah ada (modular) untuk mempersingkat waktu pembangunan atau pengembangan aplikasi.

D. Dokumentasi

Dokumentasi wajib dibuat pada semua siklus aplikasi. Dokumentasi aplikasi dapat memudahkan dalam melakukan identifikasi hingga pemeliharaan aplikasi di lingkungan Pemerintah Kabupaten Lembata. Adapun dokumentasi yang setidaknya wajib dimiliki oleh setiap tahapan/siklus dalam pembangunan/pengembangan aplikasi khusus, meliputi:

1. Perencanaan

Dokumen pada tahap perencanaan berisi daftar rencana aplikasi yang akan dibangun/dikembangkan dalam satu tahun sesuai dengan urutan prioritas, dengan sedikitnya memuat parameter sebagai berikut:

- a. Sumber: berasal dari peta rencana atau dari permohonan perangkat daerah.
- b. Tingkat kesulitan: dengan nilai rendah, sedang, atau tinggi.
- c. Urgensi dari aplikasi yang akan dibangun/dikembangkan: dengan nilai rendah, sedang, atau tinggi.

2. Analisis

a. Analisis Kelayakan

Dokumen pada analisis kelayakan paling sedikit memuat tentang:

- a.1. Identifikasi layanan.
- a.2. Metode yang digunakan dalam melakukan analisis.
- a.3. Hasil analisis.
- a.4. Kesimpulan dan rekomendasi.

b. Analisis Kebutuhan

Dokumen analisis kebutuhan paling sedikit memuat tentang:

- b.1. Identifikasi kondisi awal (*existing*) meliputi latar belakang aplikasi khusus yang akan dibangun/dikembangkan, sumber daya yang tersedia, dan pihak-pihak yang terkait.
- b.2. Identifikasi proses bisnis meliputi fungsi dan layanan yang dibutuhkan pengguna serta alur prosesnya.
- b.3. Identifikasi kebutuhan sistem meliputi kebutuhan fungsional dan kebutuhan non fungsional serta spesifikasi sistem.
- b.4. Identifikasi *output* meliputi tujuan yang akan dicapai dari aplikasi khusus yang akan dibangun/dikembangkan, agar sesuai dengan tujuan organisasi.
- b.5. Hasil analisis meliputi arsitektur aplikasi, proses bisnis aplikasi, entitas data, serta tatakelola penyelesaian pembangunan/ pengembangan aplikasi.

3. Perancangan

a. Perancangan Database

Perancangan database meliputi perancangan tabel serta relasi antar tabel dalam bentuk *Entity Relationship Diagram (ERD)*.

b. Perancangan Antarmuka

Perancangan antarmuka meliputi perancangan desain *mockup* aplikasi yang menunjukkan pemodelan alur proses aplikasi.

4. Pengembangan

Dokumen kode program (*source code*) antara lain berisi struktur kode program, penggunaan *library*, konfigurasi keamanan pada kode program, dan konfigurasi *framework*.

5. Pengujian

a. Pengujian Fungsional

Dokumen pengujian fungsional berisi tentang rencana pengujian aplikasi khusus, tahapan pengujian aplikasi, dan evaluasi aplikasi.

b. Pengujian Penerimaan Pengguna

Dokumen penerimaan pengguna berisi tentang rencana pengujian berdasarkan modul atau fitur dalam layanan, hasil pengujian yang dilakukan pengguna, dan kesimpulan dari hasil pengujian.

c. Pengujian Kinerja

Dokumen pengujian kinerja berisi tentang rencana pengujian aplikasi khusus, tahapan pengujian aplikasi, dan kesimpulan hasil pengujian.

d. Pengujian Keamanan

Dokumen pengujian keamanan berisi tentang rencana pengujian keamanan aplikasi khusus, tahapan pengujian keamanan aplikasi, dan hasil pengujian keamanan dari aplikasi khusus yang diuji.

6. Implementasi

Dokumen pada tahap implementasi terdiri dari berita acara serah terima dari pembuat aplikasi khusus kepada pengelola aplikasi khusus, surat keputusan kepala perangkat daerah tentang tim pengelola, dan manual penggunaan aplikasi khusus.

7. Pemeliharaan

Dokumen pemeliharaan berisi tentang perekaman kinerja aplikasi (*performance tracking*) dan dukungan penanganan insiden terhadap sumber daya aplikasi khusus.

8. Pemantauan dan evaluasi

Dokumen pemantauan dan evaluasi berisi tentang rekomendasi terhadap hasil monitoring dan evaluasi aplikasi khusus.

E. *Source Code Management* (Manajemen Kode Sumber)

Pembangunan atau pengembangan aplikasi khusus harus menggunakan layanan *source code management* melalui repository aplikasi yang dikelola oleh Dinas Komunikasi dan Informatika Kabupaten Lembata.

Penyimpanan kode sumber dipisahkan berdasarkan *branch* yang akan digunakan sebagai lingkungan pengembangan, paling sedikit terdiri dari:

1. *Beta*

Penyimpanan kode sumber pada *branch* beta digunakan untuk lingkungan pengembangan aplikasi khusus dengan tujuan tahap pengembangan, penambahan fitur-fitur baru, perbaikan, ujicoba, dan pengujian.

2. *Master*

Penyimpanan kode sumber pada *branch master* ditujukan untuk lingkungan implementasi aplikasi khusus.

Apabila diperlukan dapat ditambahkan satu *branch* lagi yaitu *branch staging*, yang difungsikan sebagai replika sesuai kondisi yang sama persis dengan lingkungan produksi, baik dari infrastruktur maupun sumber daya. Lingkungan pengembangan *staging* digunakan untuk menguji website/aplikasi sebelum diluncurkan di lingkungan produksi, atau bisa digunakan sebagai media sosialisasi kepada pengguna agar tidak mengganggu performa dan data di aplikasi khusus yang sudah berjalan.

F. Keamanan

Aplikasi yang dibangun atau dikembangkan harus memenuhi syarat keamanan sesuai Peraturan Bupati Lembata tentang Manajemen Keamanan Informasi SPBE, yang meliputi

1. Autentikasi

Autentikasi adalah tindakan menetapkan atau mengkonfirmasi seseorang (atau sesuatu) sebagai otentik dan bahwa klaim yang dibuat oleh seseorang atau tentang perangkat adalah benar, tahan terhadap peniruan identitas, dan mencegah pemulihan atau intersepsi kata sandi. Prosedur autentifikasi pada aplikasi khusus di lingkungan Pemerintah Kabupaten Lembata dengan ketentuan keamanan kata sandi, adalah sebagai berikut:

- a. Pendaftaran akun menggunakan email aktif dan menggunakan verifikasi email.
- b. Verifikasi bahwa kata sandi yang ditetapkan pengguna memiliki panjang minimal 8 (delapan) karakter, kombinasi antara huruf kecil, huruf besar, angka, dan spesial karakter (*symbol*). Dilakukan validasi dari sistem jika format tidak sesuai maka tidak bisa diproses. Berlaku untuk pembuatan maupun perubahan kata

sandi. Ditambahkan keterangan persyaratan kata sandi pada kolom input kata sandi.

- c. Menyediakan pengukur kekuatan kata sandi pada aplikasi untuk membantu pengguna dalam membuat kata sandi yang kuat.
 - d. Menyediakan fitur lihat sementara kata sandi pengguna untuk membantu pengguna dalam memasukan kata sandi.
 - e. Mengatur jumlah salah input kata sandi dengan batas maksimal 5 (lima) kali, pengguna akan tertahan dan bisa menginput kata sandi kembali setelah 3 (tiga) menit.
 - f. Mengatur masa berlaku kata sandi selama 1 (satu) tahun. Jika masa berlaku berakhir, pengguna diwajibkan untuk merubah kata sandi.
 - g. Pastikan terdapat menu ubah kata sandi yang mewajibkan memasukan kata sandi pengguna saat ini pada proses perubahan kata sandi.
 - h. Jika akun pengguna dibuat oleh pengembang/admin aplikasi, maka pengguna wajib merubah kata sandi saat pertama kali masuk ke dalam aplikasi.
 - i. Mengatur mekanisme pemulihan atau reset kata sandi pengguna. Pemulihan atau reset kata sandi wajib dilakukan mandiri oleh pengguna.
 - j. Penggunaan kata sandi bertingkat diwajibkan bagi aplikasi yang menurut hasil penilaian Tim Asesmen merupakan aplikasi yang mengelola data penting dan sangat rahasia.
 - k. Menjaga kerahasiaan kata sandi yang disimpan pada database melalui mekanisme kriptografi satu arah.
 - l. Menggunakan jalur komunikasi yang diamankan untuk proses autentikasi pada server.
 - m. Penerapan kata sandi sesuai syarat di atas juga termasuk untuk kata sandi pada server.
 - n. Pastikan akun *default* pada server diubah (contoh: root).
 - o. Menyediakan *captcha* atau operasi hitung pada halaman login dan pendaftaran.
 - p. Menyediakan *checkbox* pernyataan atas kebenaran data yang diberikan pada halaman pendaftaran.
2. Manajemen Sesi (*Session Management*)

Salah satu komponen inti dari setiap aplikasi berbasis web atau *API stateful* adalah mekanisme yang digunakan untuk mengontrol dan

mempertahankan status untuk pengguna atau perangkat yang berinteraksi dengannya. Manajemen sesi mengubah *protokol stateless* menjadi *stateful*, yang sangat penting untuk membedakan pengguna atau perangkat yang berbeda. Prosedur keamanan manajemen sesi dasar adalah sebagai berikut:

- a. Memastikan aplikasi tidak pernah mengungkapkan token sesi dalam parameter URL.
- b. Memanfaatkan pengendali sesi yang disediakan oleh *framework* aplikasi.
- c. Atur pengendali waktu habis sesi (*auto logout*) minimal 10 (sepuluh) menit dan maksimal 15 (lima belas) menit.
- d. Memastikan bahwa *logout* dan kadaluwarsa membatalkan token sesi sehingga tombol kembali pada *browser* tidak melanjutkan sesi sebelumnya.
- e. Jika Aplikasi mengizinkan pengguna untuk tetap masuk, pastikan autentikasi ulang dilakukan secara berkala dalam waktu 12 (dua belas) jam.
- f. Lakukan perlindungan terhadap duplikasi dan mekanisme persetujuan pengguna. Satu akun pengguna tidak diperbolehkan digunakan secara bersamaan di beda perangkat.

3. Kontrol Akses (*Access Control*)

Otorisasi adalah konsep yang memungkinkan akses ke sumber daya hanya bagi mereka yang diizinkan untuk menggunakannya. Prosedur kontrol akses dimaksud adalah sebagai berikut:

- a. Menyediakan *user level root* dan menerapkan user level manajemen (paling sedikit terdiri dari tiga level yaitu: super admin/admin, operator, dan pengawas).
- b. Menyediakan halaman admin untuk pengelolaan konten sesuai hak akses pengguna.
- c. Mengatur verifikasi kebenaran token ketika mengakses data dan informasi agar hanya bisa diakses oleh yang berhak.
- d. Pastikan bahwa semua atribut pengguna dan data serta informasi yang digunakan kontrol akses tidak dapat dimanipulasi oleh pengguna.
- e. Pastikan aplikasi atau *framework* memberlakukan mekanisme anti-CSRF (*Cross-Site Request Forgery* atau anti pemalsuan permintaan lintas situs) yang kuat.

- f. Pastikan bahwa penjelajahan direktori dinonaktifkan, pastikan juga aplikasi tidak boleh mengizinkan pengungkapan metadata atau direktori.

4. Validasi Input (*Input Validation*)

Kontrol validasi input yang diterapkan dengan benar menggunakan daftar allow positif dan pengetikan data yang kuat, dapat menghilangkan lebih dari 90% dari semua serangan injeksi. Pemeriksaan panjang dan jangkauan dapat mengurangi hal ini lebih lanjut. Prosedur validasi input adalah sebagai berikut:

- a. Memastikan bahwa data terstruktur divalidasi dengan kuat menggunakan penentuan jenis karakter, panjang karakter dan pola yang sesuai.
- b. Memastikan pada setiap *form* atau *field input* pengguna dilakukan sanitasi terhadap jumlah dan jenis karakter yang diizinkan (tidak boleh menerima karakter *script*), serta melakukan *setting* anti *Cross Site Scripting* (XSS).
- c. Memastikan data tidak tersimpan jika terjadi kesalahan pada proses validasi.
- d. Melakukan perlindungan dari serangan injeksi basis data, antara lain dengan *query sql* menggunakan *storage procedure, function procedure, setting environment* menjadi *production* dan tidak menampilkan *error system*.
- e. Memastikan aplikasi sudah melakukan perlindungan terhadap serangan *Local File Inclusion* (LFI) atau *Remote File Inclusion* (RFI).
- f. Memastikan aplikasi sudah melakukan perlindungan terhadap serangan *XPath injection* atau *XML injection attacks*.

5. Kriptografi

Aset terpenting adalah data yang diproses, disimpan, atau ditransmisikan oleh aplikasi. Selalu lakukan penilaian dampak privasi untuk mengklasifikasikan kebutuhan perlindungan data dari setiap data yang disimpan dengan benar. Prosedur kriptografi adalah sebagai berikut:

- a. Memastikan bahwa data pribadi, penting dan rahasia, dienkrpsi menggunakan mekanisme enkripsi satu arah.
- b. Memastikan enkripsi tidak menggunakan metode atau algoritma yang tidak aman.

6. Penanganan *Error* dan Pencatatan Log

Tujuan utama penanganan *error* dan pencatatan log adalah untuk memberikan informasi yang berguna bagi pengguna, administrator, dan tim respons insiden. Tujuannya bukan untuk membuat log dalam jumlah besar, tetapi log yang berkualitas tinggi. Prosedur penanganan *error* dan pencatatan log adalah sebagai berikut:

- a. Memastikan log tidak mencatat informasi sensitif atau dikecualikan.
- b. Mengatur rentang waktu penyimpanan log sesuai kebutuhan aplikasi.
- c. Memastikan log mencatat aktivitas *user*, proses berhasil atau gagal login, kesalahan validasi input, dan kesalahan kontrol akses.
- d. Memastikan log menyertakan informasi tentang garis waktu hari dan jam terperinci (dd/mm/yyyy dan hh:mm:ss).
- e. Memastikan log terlindungi dari akses ilegal dan modifikasi.
- f. Memastikan zona waktu server sesuai dan benar.
- g. Memastikan pengaturan tampilan pesan saat terjadi kesalahan atau *error*.

7. Proteksi Data

Ada tiga elemen kunci untuk perlindungan data yang baik, antara lain: kerahasiaan, integritas, dan ketersediaan. Standar ini mengasumsikan bahwa perlindungan data diberlakukan pada sistem terpercaya, seperti server yang telah diperkuat dan memiliki perlindungan yang memadai. Prosedur proteksi data adalah sebagai berikut:

- a. Menyediakan prosedur *backup data* secara otomatis melalui sistem.
- b. Menyediakan prosedur *backup data* secara manual melalui super admin/admin.
- c. Memastikan data yang di-*backup* termasuk file *upload*.
- d. Memastikan data *backup* tersimpan dengan aman.
- e. Jika aplikasi memiliki fungsi *export* data dalam bentuk file (pdf, xls, doc, ppt dan sejenisnya) dimana didalamnya mengandung informasi sensitif, data pribadi, dan atau data rahasia, file keluaran wajib menggunakan *password*, yang diatur dalam SOP terpisah.

8. Keamanan Komunikasi

Prosedur yang harus diperhatikan dalam keamanan komunikasi adalah:

- a. Pastikan aplikasi menerapkan *protocol HTTPS* atau *SSL*.
- b. Komunikasi dengan server menggunakan jaringan internal Pemerintah Kabupaten Lembata.

9. Pengendalian Kode Berbahaya

Prosedur yang harus diperhatikan dalam pengendalian kode berbahaya adalah:

- a. Memastikan kode sumber aplikasi dan *library* tidak mengandung *backdoor* dan kode berbahaya lainnya.
- b. Memastikan melakukan pengaturan *Content Security Policy (CSP)* dan *Cross-Origin Resource Sharing (CORS)*.
- c. Memastikan bahwa aplikasi meminta izin ke pengguna sebelum memanfaatkan fitur dan sensor dari pengguna.
- d. Memastikan aplikasi hanya meminta izin yang diperlukan.

10. Logika Bisnis

Prosedur yang harus diperhatikan dalam logika bisnis adalah:

- a. Memastikan aplikasi hanya memproses alur bisnis sesuai dengan alur bisnis yang ada, dan tidak melewatkan langkah alur bisnis.
- b. Memastikan aplikasi hanya memproses alur logika bisnis dengan semua langkah lainnya diproses dalam waktu yang realistis.
- c. Memastikan aplikasi memiliki kontrol anti-otomatisasi.

11. File

Prosedur yang harus diperhatikan dalam keamanan file adalah:

- a. Memastikan aplikasi tidak akan menerima file dengan ukuran besar yang akan membebani penyimpanan.
- b. Memastikan aplikasi melakukan validasi terhadap *type*, ukuran dan jumlah file yang diunggah oleh pengguna.
- c. Memastikan aplikasi menerapkan pengacakan nama file yang diunggah dan tersimpan di server.
- d. Memastikan aplikasi menyimpan file unggahan diluar *root web*.
- e. Memastikan bahwa permintaan langsung ke file yang diunggah tidak akan dijalankan sebagai konten *HTML/JavaScript*.

12. Keamanan API dan Web Service

Prosedur yang harus diperhatikan dalam keamanan *API* dan *web service* adalah:

- a. *API* yang dibuat harus menggunakan token.
- b. Memastikan *URL API* tidak mengekspos informasi sensitif, seperti kunci *API* dan token sesi.
- c. *API* satuan data yang harus dijaga kerahasiaannya menggunakan token atau sistem enkripsi dan dekripsi dalam proses transaksi data.
- d. *API* dibuat menggunakan aplikasi SPLP (Sistem Penghubung Layanan Pemerintah). Proses integrasi dan interoperabilitas data yang diselenggarakan melalui SPLP Kabupaten Lembata dilaksanakan oleh unit khusus di luar ketentuan lampiran ini.

13. Keamanan Konfigurasi

Prosedur yang harus diperhatikan dalam keamanan konfigurasi adalah:

- a. Menggunakan *hardening configuration template* yang sesuai dengan standar untuk semua komponen infrastruktur aplikasi khusus, termasuk database, server dan *cloud container*.
- b. Memastikan konfigurasi server disesuaikan dengan hasil analisa kebutuhan server aplikasi.
- c. Memastikan semua data dummy, uji coba dan tidak diperlukan dihapus (termasuk *user, role, table, kolom table*, dan sebagainya).
- d. Memastikan *setting CSP (Content Security Policy)* dilakukan untuk mengurangi dampak serangan *XSS (Cross Site Scripting)*.
- e. Memastikan bahwa *header origin* yang disediakan tidak digunakan untuk keputusan autentikasi atau kontrol akses, karena *header origin* dapat dengan mudah diubah oleh penyerang.
- f. Memastikan *setting CORS (Cross-origin Resource Sharing)* menggunakan daftar domain yang terpercaya, tidak menggunakan "*null*" atau "*".

BAB III STANDARISASI DATABASE

A. Pendahuluan

Database yang dapat digunakan dalam pembangunan/pengembangan aplikasi di lingkungan Pemerintah Kabupaten Lembata harus mempertimbangkan format data, kebutuhan data, dan kompleksitas data dari aplikasi yang akan dibangun serta ketersediaan infrastruktur dan teknologi yang disediakan oleh Pemerintah Kabupaten Lembata.

B. Ruang Lingkup

Ruang lingkup standarisasi database adalah sebagai berikut:

1. Konvensi Penamaan.
2. Tabel.
3. Kolom.
4. *Primary Key*.
5. *Foreign Key*.
6. Indeks.
7. Tipe Data.
8. Keamanan Data.
9. Normalisasi.
10. Integritas Referensial.
11. Pemrograman Database.

B.1. Konvensi Penamaan

Penyusunan konvensi penamaan bertujuan untuk menetapkan standar, norma, dan kriteria penamaan hal-hal yang berkaitan dengan perancangan basis data. Hal ini dilakukan untuk mempermudah dalam membaca dan memahami skema basis data.

Objek basis data terdiri dari: basis data, skema, tabel (entitas), kolom (atribut), dan baris data.

Penamaan objek basis data dilakukan dengan ketentuan sebagai berikut:

1. Penamaan menggunakan huruf kecil (*lowercase*), dapat dikombinasikan dengan angka dan garis bawah (). Nama diawali dengan huruf (a-z) atau garis bawah. Karakter berikutnya dalam nama dapat berupa huruf, angka (0-9), atau garis bawah.

2. Tanda garis bawah digunakan sebagai pemisah apabila nama objek basis data terdiri lebih dari satu suku kata.
3. Bahasa yang digunakan untuk penamaan objek basis data diutamakan menggunakan Bahasa Indonesia.
4. Catatan: Penggunaan Bahasa Inggris diperbolehkan apabila ada pertimbangan lain, misalnya ada istilah-istilah yang maknanya menjadi berbeda jika diubah ke Bahasa Indonesia, atau tidak ditemukan padanan yang sesuai jika istilah Bahasa Inggris tersebut diterjemahkan ke Bahasa Indonesia.
5. Penggunaan singkatan diperbolehkan selama singkatan yang digunakan tersebut umum, baku, dan mudah dipahami.
6. Penamaan objek basis data menggunakan nama yang deskriptif.
7. Nama objek basis data tidak mengandung perintah *SQL*.
8. Penamaan objek basis data maksimal terdiri dari 64 (enam puluh empat) karakter.

B.2 Tabel

1. Dalam setiap tabel harus ada satu kolom yang menjadi *primary key*.
2. Apabila terdapat relasi antar tabel, tentukan dan klasifikasikan relasinya sebagai salah satu dari:
 - a. Satu ke satu (*one-to-one*).
 - b. Satu ke banyak (*one-to-many*).
 - c. Banyak ke banyak (*many-to-many*).
3. Prefix nama tabel:
 - a. **mst_** untuk master data (*mst: master*).
 - b. **ref_** untuk tabel referensi (*ref: reference*).
 - c. **trx_** untuk tabel transaksi (*trx: transaction*).
 - d. **rpt_** untuk tabel pelaporan (*rpt: report*).
 - e. **rcp_** untuk tabel rekapitulasi (*rcp: recap*).

B.3. Kolom

Setiap kolom yang ada di dalam tabel harus diberi deskripsi/penjelasan singkat mengenai isi dan fungsi kolom tersebut.

B.4. *Primary Key*

Dalam menentukan *primary key*, dapat menggunakan *natural key* atau *surrogate key*.

1. *Natural key* merupakan *primary key* yang diambil dari kolom berisi data unik untuk setiap baris. Keuntungan menggunakan *natural key* adalah datanya sudah ada sehingga tidak perlu menambah kapasitas ruang penyimpanan untuk satu kolom tambahan khusus *primary key*.
 - a. Misal di tabel pegawai terdapat kolom NIK atau NIP, maka *primary key* dapat menggunakan salah satu kolom tersebut dengan pertimbangan nilai NIK atau NIP untuk setiap orang pasti berbeda dan kemungkinan terjadi perubahan nilainya sangat kecil atau bahkan tidak akan ada perubahan.
 - b. Hal yang perlu dipertimbangkan bila ingin menentukan *primary key* berdasarkan *natural key* adalah pilih kolom yang nilainya unik untuk setiap baris dan pastikan nilainya tidak akan berubah.
2. *Surrogate key* merupakan *primary key* yang diciptakan dengan membuat kolom baru.
 - a. Misalnya menambahkan *primary key* berupa kolom id di tabel pegawai yang nilainya dikonfigurasi supaya bertambah satu (*increment*) setiap ada record baru yang dimasukkan ke dalam tabel. Kolom id ini disebut dengan istilah *surrogate key*.
 - b. Kekurangan dari *surrogate key* adalah perlu ada tambahan ruang untuk menyimpan data kolom id tersebut, juga tambahan ruang untuk membuat indeks dari *natural key* (yang seharusnya otomatis terindeks bila *natural key* menjadi *primary key*).
 - c. Jika memilih *surrogate key*, maka perlu menentukan *key generator*. Nilai dari *surrogate key* tidak berhubungan dengan data yang diwakilinya, maka bisa dilakukan optimasi dalam pemilihan algoritma untuk menghasilkan nilai baru.

Ada 2 (dua) opsi yang biasa digunakan, yaitu:

 - c.1. *Auto increment/sequence*.

Cara ini disebut dengan berbagai istilah, misalnya *AUTO_INCREMENT* di *MySQL*, *SEQUENCE* di *Oracle*, *SERIAL* di *PostgreSQL*, *IDENTITY* di *MS SQL Server*, *AutoNumber* di *MS Access*, dan lainnya. Intinya adalah penggunaan tipe data integer yang nilainya bertambah terus.

c.2. *UUID/GUID*.

Tipe datanya 32 (tiga puluh dua) karakter alfanumerik, bisa diwakili di basis data dengan *CHAR* atau *VARCHAR*.

Prioritas pemilihan *primary key*:

1. Diutamakan menggunakan *natural key* yang memiliki tipe data numerik.
2. Apabila ada pertimbangan lain terkait dengan performa atau kecepatan pencarian data, serta penambahan ruang untuk membuat satu kolom baru (berisi id) tidak menjadi masalah, maka dapat menggunakan *surrogate key*.

B.5. *Foreign key*

Kriteria penamaan *foreign key* sebagai berikut:

1. Menggunakan huruf kecil.
2. Apabila nama *foreign key* terdiri lebih dari satu suku kata, maka gunakan tanda garis bawah sebagai pemisah antar kata.
3. Prefiks *foreign key* diawali dengan **fk_[nama kolom]**.

B.6. Indeks

Penerapan indeks dilakukan untuk dapat mempercepat proses pencarian (*query*) data. Kriteria penamaan indeks adalah sebagai berikut:

1. Menggunakan huruf kecil (*lowercase*)
2. Apabila nama indeks terdiri lebih dari satu suku kata menggunakan tanda garis bawah (_) sebagai pemisah antar kata.
3. Prefix indeks diawali dengan **idx_[nama kolom yang dijadikan indeks]**

B.7. Tipe data

Setiap data disimpan sesuai kebutuhan dengan mempertimbangkan tipe dan besaran data. Hal ini dilakukan untuk menghemat ruang penyimpanan dan mempertimbangkan performa basis data.

1. Untuk data non numerik dengan panjang karakter tetap, sedapat mungkin menggunakan tipe data *CHAR*.
2. Untuk data non numerik dengan panjang karakter tidak tetap, menggunakan tipe data *VARCHAR* dengan memperhatikan kemungkinan panjang maksimal karakter.

3. Untuk data numerik sedapat mungkin menggunakan TINYINT untuk range data yang masih memungkinkan.
4. Disarankan mengubah ukuran panjang tipe data yang disesuaikan dengan kebutuhan maksimal.

B.8. Keamanan Data

1. Metode penyimpanan data dapat dilakukan dengan enkripsi tertentu dengan mempertimbangkan keamanan data sensitif dan performa basis data saat mengolah data tersebut.
2. Tidak diperbolehkan untuk menyimpan kata sandi/*password* menggunakan *plain text* yang terbaca, wajib menggunakan enkripsi.
3. Diperbolehkan menggunakan enkripsi yang berbeda untuk setiap kolom dalam tabel yang sama.
4. Tidak diperkenankan menggunakan enkripsi yang telah diketahui atau mudah untuk dibobol (misalnya md5).
5. Penggunaan enkripsi dikombinasikan dengan SALT dengan panjang 128 (seratus dua puluh delapan) bit untuk memperkuat keamanan data yang terenkripsi.

B.9. Normalisasi

Normalisasi dilakukan sebagai upaya untuk meminimalkan redundansi data (pengulangan data yang sama di beberapa tempat). Normalisasi dilakukan minimal sampai bentuk ketiga (1NF, 2NF, dan 3NF).

B.10. Integritas Referensial

Integritas referensial (*referential integrity*) adalah seperangkat aturan yang mengatur hubungan antara kunci primer dengan kunci tamu pada tabel-tabel yang berada dalam suatu basis data relasional untuk menjaga konsistensi data. Tujuan integritas referensial sendiri adalah untuk menjamin dan memastikan agar entitas dalam suatu tabel yang menunjuk ke suatu pengenal unik pada suatu baris di tabel lain benar-benar menunjuk pada nilai yang memang ada. Berdasarkan operasi yang dilakukan, integritas referensial dapat dibedakan menjadi: penambahan (*insert*), penghapusan (*delete*) dan peremajaan (*update*).

Integritas referensial membuat ketiga operasi di atas dapat dilaksanakan pada tabel yang memiliki relasi. Sehingga proses

penghapusan ataupun peremajaan suatu kolom juga akan terjadi pada kolom tabel lain yang mempunyai referensi dengannya. Integritas referensial merupakan cara untuk menjaga agar kunci asing (*foreign key*) suatu tabel dan kunci primer (*primary key*) milik tabel yang direferensi oleh kunci asing selalu konsisten.

B.11. Pemrograman Basis Data

1. *Stored Procedure*

- a. Prefix: **usp** [nama *stored procedure*].
'ups' merupakan singkatan dari *user-defined stored procedure*.
- b. Penamaan *stored procedure* menggunakan huruf kecil dan hanya boleh terdiri dari huruf (a-z).
- c. Pemisah antar kata menggunakan garis bawah.
- d. Penamaan *stored procedure* diberi keterangan operasi yang dikerjakan seperti 'Add', 'Insert', 'Update', 'Change', 'Set', 'Delete', 'Remove', 'Select' atau 'Get'.

Contoh: **usp_update_data_pegawai**

Dalam penulisan *stored procedure* harus menyertakan komentar yang berisi detail pembuatan *stored procedure* serta riwayat perubahannya, sebagai berikut:

- a. Detail pembuatan *stored procedure* harus berisi informasi:
 - a.1. *Stored Procedure Name*: nama *stored procedure*.
 - a.2. *Purpose*: penjelasan deskripsi singkat tujuan *stored procedure* tersebut dibuat.
 - a.3. *Author*: nama pembuat *stored procedure*.
 - a.4. *Created On*: tanggal *stored procedure* dibuat.
- b. Riwayat Perubahan *stored procedure* harus berisi informasi:
 - b.1. *Revision No.*: nomor revisi *stored procedure*.
 - b.2. *Changed On*: tanggal perubahan.
 - b.3. *Changed By*: *author* yang melakukan perubahan *code*.
 - b.4. *Change Description*: deskripsi perubahan.

Berikut contoh *template* yang dapat digunakan untuk penulisan detail dan riwayat perubahan *stored procedure*.

```

CREATE PROCEDURE [dbo].[usp_stored_procedure_name]
(
    @parameter1 INT
    ,@parameter2 VARCHAR(50)
    ,@return_value1 INT OUTPUT
    ,@error_code INT = 0 OUTPUT
) AS
/*
*****Creation Details*****
Stored Procedure Name : [dbo].[usp_stored_procedure_name]
Purpose : Insert a new record in [table_name]
Author : Author Name
Created On : 2025/06/01
*****Revision Details*****
Project/
Revision No. Changed On Changed By Change Description
-----
1234 2018/01/10 Mr. ABC New column 'new_column_name'
added.
1235 2018/03/16 Mr. XYZ Column 'column_name' deleted.
*/

BEGIN
    SET NOCOUNT ON
    BEGIN TRY
        BEGIN TRANSACTION
            --INSERT statement and other logic go here.
        COMMIT TRANSACTION
        RETURN 0
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION
        RETURN 1
    END CATCH
END

```

2. *Trigger*

- a. Prefix: **trg_**
- b. Penamaan *trigger* menggunakan huruf kecil dan hanya boleh terdiri dari huruf (a-z).
- c. Pemisah antar kata menggunakan garis bawah.
- d. Format penamaan *trigger*: **trg_[nama tabel]_[singkatan trigger]**.

Contoh: **trg_pegawai_au**

Waktu Trigger	Singkatan Trigger	Keterangan Trigger
BEFORE INSERT	bi	<i>Trigger</i> dijalankan sebelum <i>record</i> dimasukkan ke basis data
AFTER INSERT	ai	<i>Trigger</i> dijalankan sesudah <i>record</i>

		dimasukkan ke basis data
BEFORE UPDATE	bu	<i>Trigger</i> dijalankan sebelum <i>record</i> diubah di basis data
AFTER UPDATE	au	<i>Trigger</i> dijalankan sesudah <i>record</i> diubah di basis data
BEFORE DELETE	bd	<i>Trigger</i> dijalankan sebelum <i>record</i> dihapus di basis data
AFTER DELETE	ad	<i>Trigger</i> dijalankan sesudah <i>record</i> dihapus di basis data

Dalam penulisan *trigger* harus menyertakan komentar yang berisi detail pembuatan *trigger* serta riwayat perubahannya, sebagai berikut:

- a. Detail pembuatan *trigger* harus berisi informasi:
 - a.1. *Trigger Name*: nama *trigger*.
 - a.2. *Purpose*: tujuan *trigger* tersebut dibuat.
 - a.3. *Author*: nama pembuat *trigger*.
 - a.4. *Created On*: tanggal dibuat.
- b. Riwayat perubahan *trigger* harus berisi informasi:
 - b.1. *Revision No.*: nomor revisi.
 - b.2. *Changed On*: tanggal perubahan.
 - b.3. *Changed By*: *author* yang melakukan perubahan *code*.
 - b.4. *Change Description*: deskripsi perubahan.

Berikut contoh *template* yang dapat digunakan untuk penulisan detail dan riwayat perubahan *trigger*.

```

CREATE FUNCTION [dbo].[udf_function_name]
(
    @Param1 INT,
    @Param2 INT,
    @Param3 VARCHAR(50)
) RETURNS BIT AS
/*
    *****Creation
Details*****
Function Name : udf_function_name
Purpose : Calculates and returns the value.
Author : Author Name
Created On : 2025/06/01
    *****Revision
Details*****
Project/`
Revision No. Changed On Changed By Change Description
-----
1234 2018/01/01 Mr. ABC Changed the calculation part.
1235 2018/02/08 Mr. XYZ Revert the previous change.
*/
BEGIN
    --Function Body goes here.
    RETURN (0)
END

```

BAB IV PENGEMBANGAN APLIKASI

A. Pendahuluan

Sesuai dengan arsitektur aplikasi khusus, pengembangan merupakan proses pengkodean dari sisi *back end*, untuk membuat *web service* yang akan digunakan oleh *front end*, menggunakan bahasa pemrograman.

B. Ruang Lingkup

Ruang lingkup dalam tahap pengembangan aplikasi khusus ini meliputi: 1) Bahasa Pemrograman;

- 2) Konvensi Penamaan URL;
- 3) Dokumentasi; dan
- 4) Pencatatan Log.

B.1. Bahasa Pemrograman

Pengembangan aplikasi khusus di lingkungan Pemerintah Kabupaten Lembata menggunakan beberapa standar bahasa pemrograman yang umum digunakan, serta wajib menggunakan *framework*. Bahasa pemrograman yang bisa digunakan antara lain (tidak terbatas pada):

- a. PHP
- b. Javascript
- c. Java
- d. Python
- e. Go

B.2. Konvensi Penamaan URL

Penyusunan konvensi penamaan bertujuan untuk menetapkan standar, norma, dan kriteria penamaan pada hal-hal yang berkaitan dengan penggunaan *web service*. Hal ini dilakukan untuk mempermudah dalam membaca dan memahami fungsi dari *web service* tersebut.

a. *Uniform Resource Locator* (URL)

Uniform Resource Locator (URL) merupakan rangkaian karakter huruf atau angka yang disusun berdasarkan strukturnya untuk menunjukkan alamat *web service* dalam memperoleh atau memproses data/informasi yang digunakan dalam aplikasi khusus.

Alamat URL *web service* untuk *backend* berada di bawah sub domain lembatakab.go.id dengan konvensi penamaan sebagai berikut:

1. Nama Aplikasi

Konvensi penamaan aplikasi khusus menunjukkan nama utama yang digunakan untuk membedakan aplikasi khusus satu dengan yang lainnya. Penggunaan nama aplikasi menunjukkan fungsi utama dari layanan aplikasi khusus tersebut. Apabila diperlukan menggunakan dua kata atau lebih, dipisahkan dengan karakter *under line* (_). Contoh yang diperkenankan adalah sebagai berikut:

- a. <https://sidastik.lembatakab.go.id/>
- b. <https://kilat.lembatakab.go.id/>
- c. https://nama_aplikasi.lembatakab.go.id/

2. Module

Konvensi penamaan module harus menunjukkan nama modul yang dikelola dalam aplikasi. Penamaan modul diletakkan setelah URL web service aplikasi dan harus memenuhi ketentuan: menggunakan huruf kecil dan dipisahkan dengan tanda strip (-) apabila terdiri dari 2 (dua) kata atau lebih.

Contoh penamaan modul adalah sebagai berikut:

- a. <https://kilat.lembatakab.go.id/layanan>
- b. <https://kilat.lembatakab.go.id/tentang-aplikasi>

3. Aksi

Konvensi penamaan aksi disesuaikan dengan modul yang sedang dikerjakan, dan tidak wajib untuk ditampilkan di URL web aplikasi khusus tersebut.

Contoh penamaan aksi adalah sebagai berikut:

https://kilat.lembatakab.go.id/berita?page_hal=2

b. Respon

Konvensi penamaan respon dari web service ketika memberikan jawaban dari klien paling sedikit memuat parameter sebagai berikut:

1. Success

Parameter ini berisi nilai **true** atau **false**, untuk menunjukkan proses berhasil atau gagal.

2. Status

Parameter ini berisi kode status HTTP terstandar yang merupakan 3 (tiga) digit respon dari web service kepada klien peminta layanan.

3. Message

Parameter ini berisi pesan, baik pesan berhasil maupun pesan gagal yang diberikan oleh web service untuk menjelaskan hasil permintaan *request* dari klien dan bisa ditampilkan kepada pengguna. Pesan harus jelas dan singkat serta mudah dipahami.

4. Data

Parameter ini berisi data yang diminta oleh klien kepada web service. Parameter ini bersifat generik, menyesuaikan format yang dibutuhkan oleh aplikasi khusus.

Contoh dari respon yang diberikan adalah sebagai berikut:

```
{
  "success": true,
  "status": 200,
  "message": "Pengaduan baru berhasil disimpan."
  "data": null
}
```

B.3. Dokumentasi

Dokumentasi digunakan untuk memahami kemampuan web service tanpa akses langsung ke kode sumber. Dokumentasi pengembangan kode sumber *backend* di lingkungan Pemerintah Kabupaten Lembata menggunakan *swagger* sebagai standar yang diacu.

B.4. Pencatatan Log

Aplikasi khusus harus menyimpan log untuk setiap aksi yang berkaitan dengan transaksi yang krusial dalam proses bisnis. Parameter yang dicatat paling sedikit meliputi:

1. Pengguna yang melakukan aksi.
2. Waktu dengan format mulai tanggal sampai dengan detik.
3. Jenis aksi yang dilakukan atau data yang dimodifikasi.

BAB V ANTARMUKA PENGGUNA (USER INTERFACE)

A. Pendahuluan

Antarmuka pengguna (*user interface*) adalah tampilan visual pada layar yang menghubungkan sistem dengan pengguna (*user*) dalam melaksanakan interaksi penggunaan aplikasi. Antarmuka pengguna yang meliputi tata letak, warna, tipografi, dan grafik, harus dirancang untuk memberikan kemudahan bagi pengguna dan memiliki visualisasi yang menarik.

B. Ruang Lingkup

Ruang Lingkup standarisasi antarmuka pengguna dalam proses pembangunan atau pengembangan aplikasi khusus di lingkungan Pemerintah Kabupaten Lembata antara lain:

1. Antarmuka pengguna aplikasi berbasis web.
2. Antarmuka pengguna aplikasi berbasis mobile.

C. *User Interface* Aplikasi Berbasis Web

1. Bahasa Pemrograman

Aplikasi berbasis website yang akan dikembangkan oleh Pemerintah Kabupaten Lembata menggunakan beberapa standar bahasa pemrograman yang umum digunakan, antara lain (tidak terbatas pada):

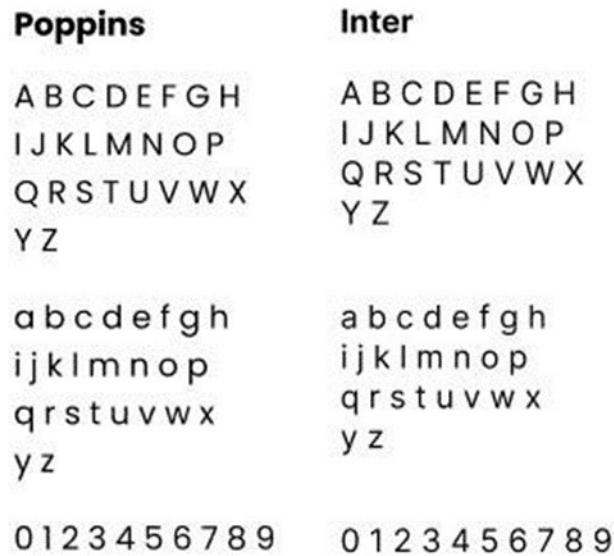
- a. *React*.
- b. *Vue*.
- c. *Bootstrap*.

2. Desain *Template*

a. Tipografi

Tipografi bertujuan agar desain, konten, dan navigasi terbaca dan terlihat dengan jelas, menarik, dan mudah dipahami oleh user.

Font yang digunakan dapat menggunakan *Poppins* atau *Inter*.



Gambar 5.1. Contoh Tipografi.

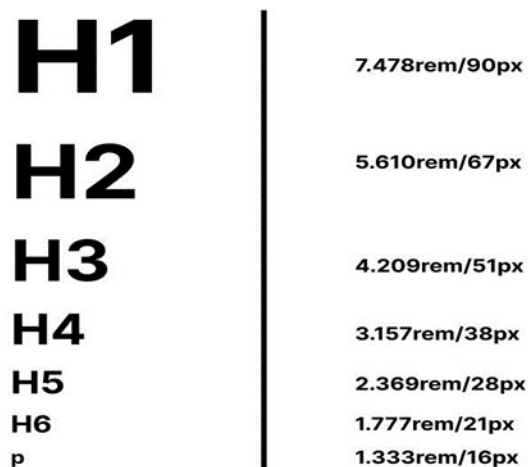
Dengan Height mulai dari light, normal, semibold, bold, dan bolder:



Gambar 5.2. Ketebalan Tipografi.

b. *Type Scale*

Type Scale bertujuan untuk memberikan keserasian terhadap hierarki tipografi pada konten.



Gambar 5.3. *Type Scale*

c. *Line Spacing*

Line Spacing bertujuan untuk memberikan *space* kosong/jarak antar tulisan agar dapat terlihat dan terbaca dengan baik. *Golden Ratio* adalah perbandingan proporsional antara dua bagian. Berikut cara menghitung *line spacing* yang sesuai dengan kebutuhan berdasarkan peraturan *Golden Ratio* yakni dengan mengalihkan besaran *font* dengan 1.618.

Berikut sudah terdapat *line height* untuk *Heading* yang digunakan:

Heading	ukuran	Line Height
H1	7.478rem/90px	146px
H2	5.610rem/67px	108px
H3	4.209rem/51px	82px
H4	3.157rem/38px	61px
H5	2.369rem/28px	45px
H6	1.777rem/21px	34px
p	1.333rem/16px	26px

Gambar 5.4. Perhitungan *Line Spacing* menggunakan *Golden Ratio*.

d. *Body Text*

Ukuran font untuk desktop minimal sebesar 16px agar mudah terbaca dengan jelas.

PEMERINTAH KABUPATEN LEMBATA
Kabupaten Lembata berada di Pulau Lembata yang beriklim tropis-kering dan memiliki gunung berapi aktif yaitu Ite Lewotolok. Masyarakat setempat juga mempunyai tradisi perburuan ikan paus secara tradisional.

Ukuran font 16 px akan terbaca jelas di tampilan desktop

PEMERINTAH KABUPATEN LEMBATA
Kabupaten Lembata berada di Pulau Lembata yang beriklim tropis-kering dan memiliki gunung berapi aktif yaitu Ite Lewotolok. Masyarakat setempat juga mempunyai tradisi perburuan ikan paus secara tradisional.

Ukuran font terlalu kecil dan akan sulit terbaca di tampilan desktop

Gambar 5.5. Contoh penggunaan ukuran font yang tepat.

Hindari penggunaan huruf kapital dengan kalimat yang panjang di dalam *body text* agar informasi dapat terbaca dengan jelas.

Kabupaten Lembata berada di Pulau Lembata yang beriklim tropis-kering dan memiliki gunung berapi aktif yaitu Ile Lewotolok. Masyarakat setempat juga mempunyai tradisi perburuan ikan paus secara tradisional.

✓ Penggunaan huruf kapital yang baik dan tepat

KABUPATEN LEMBATA BERADA DI PULAU LEMBATA YANG BERIKLIM TROPIS-KERING DAN MEMILIKI GUNUNG BERAPI AKTIF YAITU ILE LEWOTOLOK. MASYARAKAT SETEMPAT JUGA MEMPUNYAI TRADISI PERBURUAN IKAN PAUS SECARA TRADISIONAL.

✗ Penggunaan huruf kapital yang tidak tepat

Gambar 5.6. Contoh penggunaan huruf kapital yang tepat.

Ketebalan text dapat digunakan untuk meng-*highlight* informasi yang terdapat pada suatu kalimat atau paragraf.

Kabupaten Lembata berada di Pulau Lembata yang Beriklim tropis-kering dan memiliki gunung berapi aktif yaitu yaitu Ile Lewotolok. Masyarakat setempat juga mempunyai tradisi **perburuan ikan paus secara tradisional**.

✓ Penggunaan ketebalan huruf untuk meng-highlight informasi

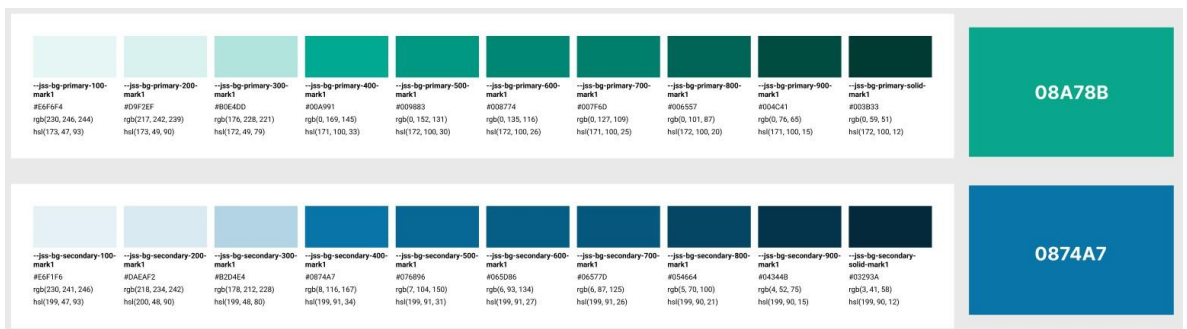
Kabupaten Lembata berada di Pulau Lembata yang Beriklim tropis-kering dan memiliki gunung berapi aktif yaitu Ile Lewotolok. Masyarakat setempat juga mempunyai tradisi *perburuan ikan paus secara tradisional*.

✗ Highlight informasi yang kurang tepat sehingga tidak terbaca

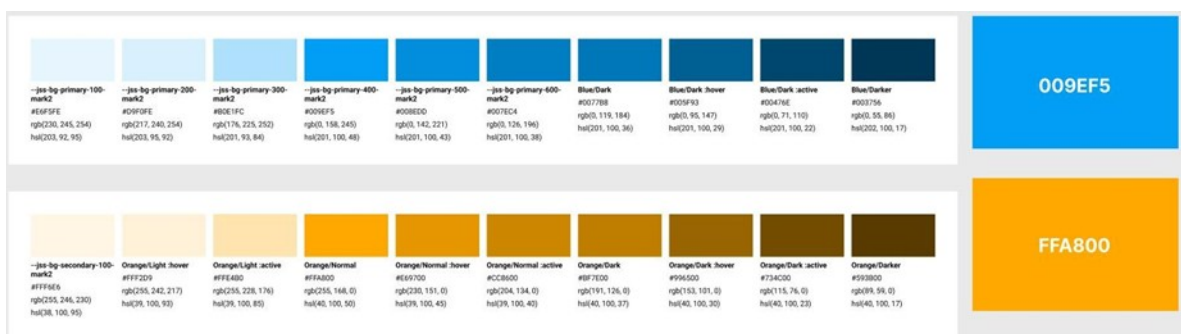
Gambar 5.7. Contoh penggunaan ketebalan *font* yang tepat untuk meng- *highlight* informasi.

e. *Colour*

Warna memiliki peran besar terhadap pengalaman dan kenyamanan pengguna dalam menggunakan aplikasi. Sistem warna dibagi menjadi 3 kelompok, yaitu primer, sekunder dan netral. Berikut beberapa pilihan warna yang dapat digunakan oleh desainer.



Gambar 5.8. Contoh pilihan kombinasi warna pertama.



g. *Layouting*

Layouting adalah proses merancang dan menyusun tata letak visual dari elemen-elemen grafis, teks, dan media lainnya dalam sebuah desain. Tujuan dari *layouting* adalah untuk menciptakan tata letak yang menarik, mudah dipahami, dan efektif dalam menyampaikan pesan atau informasi kepada audiens target.

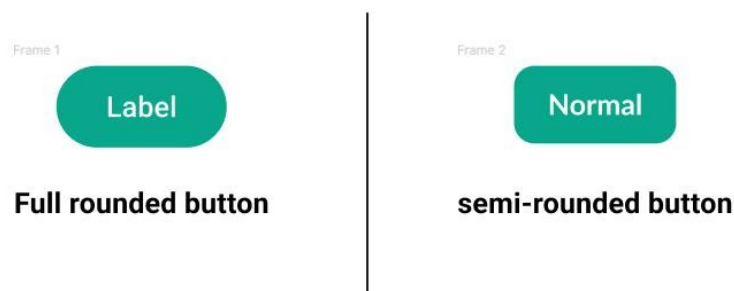
Layout breakpoints menandai titik di mana tata letak (*layout*) harus disesuaikan agar konten tetap terlihat baik dan mudah diakses pada berbagai ukuran layar.

Value (px)	Breakpoints	Column	Gutter	Margin
0 - 320	xsmall (xs)	4	16	16
320 - 575	xsmall (xs)	4	16	16
576 - 767	small (sm)	4	16	16
768 - 991	medium (md)	8	24	24
992 - 1199	large (lg)	12	24	24
1200 +	extra-large (xl)	12	24	24

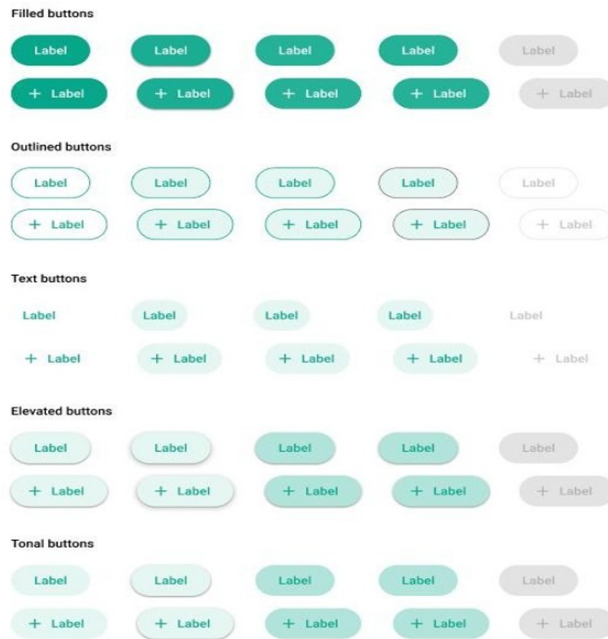
Gambar 5.13. Contoh ukuran *layout breakpoint*.

h. *Button*

Button digunakan untuk membantu pengguna dalam mengambil tindakan atau memberi perintah tertentu. Ada 2 pilihan *button* yang dapat digunakan yakni *pill button* dengan ukuran *rounded*-nya sebesar 100 px dan *button semi-rounded* dengan ukuran *rounded*-nya sebesar 10px, seperti bentuk di bawah berikut:



Gambar 5.14. Contoh pilihan *button*.



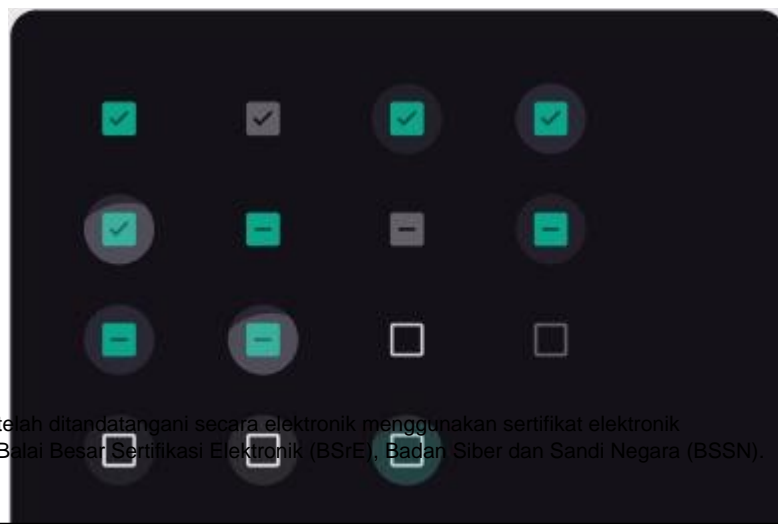
Gambar 5.15. Contoh jenis *button* yang dapat digunakan.

a. *Checkbox*

Checkbox pada web adalah elemen antarmuka pengguna yang digunakan untuk memungkinkan pengguna untuk memilih atau mengaktifkan opsi yang tersedia. Ketika sebuah *checkbox* diberi tanda centang, artinya opsi tersebut telah dipilih atau diaktifkan.



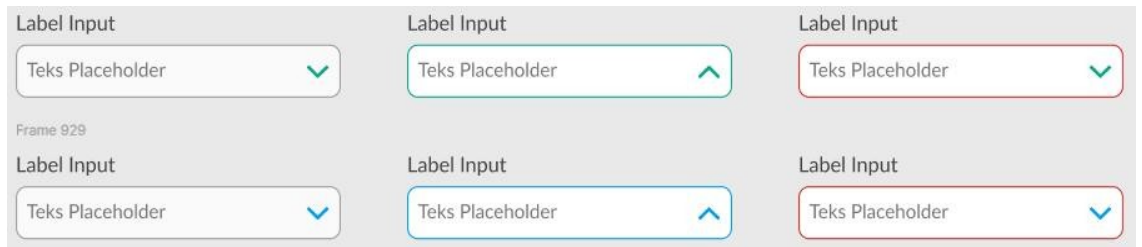
Gambar 5.16. Contoh *template checkbox* di *light mode*.



Gambar 5.17. Contoh *template checkbox* di *dark mode*.

i. *Select*

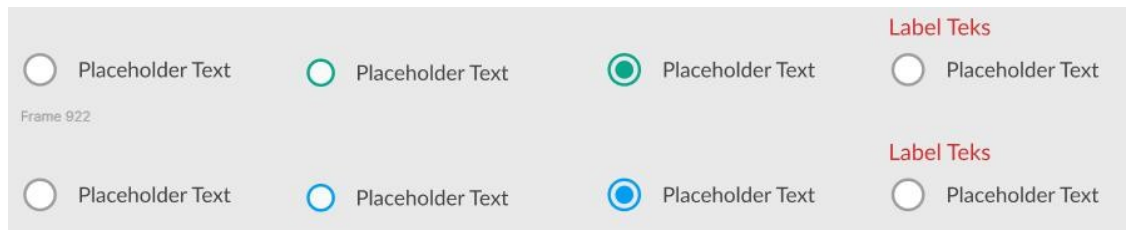
Select box adalah salah satu elemen penting dalam pembuatan formulir web dan memungkinkan pengguna untuk memberikan input dengan cara yang mudah dipahami dan dioperasikan. Elemen ini sering digunakan dalam formulir web untuk mengumpulkan input dari pengguna.



Gambar 5.18. Contoh *template desain select box*.

j. *Radio Button*

Radio button umumnya digunakan ketika hanya satu pilihan yang boleh dipilih oleh pengguna dari sekumpulan opsi yang saling terkait. Setiap *radio button* memiliki dua keadaan yaitu terpilih (*selected*) atau tidak terpilih. Ketika satu *radio button* dipilih, opsi lain dalam grup *radio button* yang sama akan secara otomatis tidak terpilih.

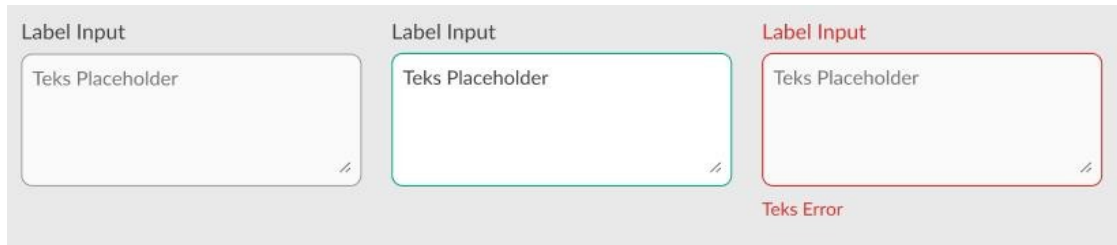


Gambar 5.19. Contoh *template radio button*.

k. *Text Area*

Text area pada web adalah elemen antarmuka pengguna yang memungkinkan pengguna untuk memasukkan teks dalam bentuk

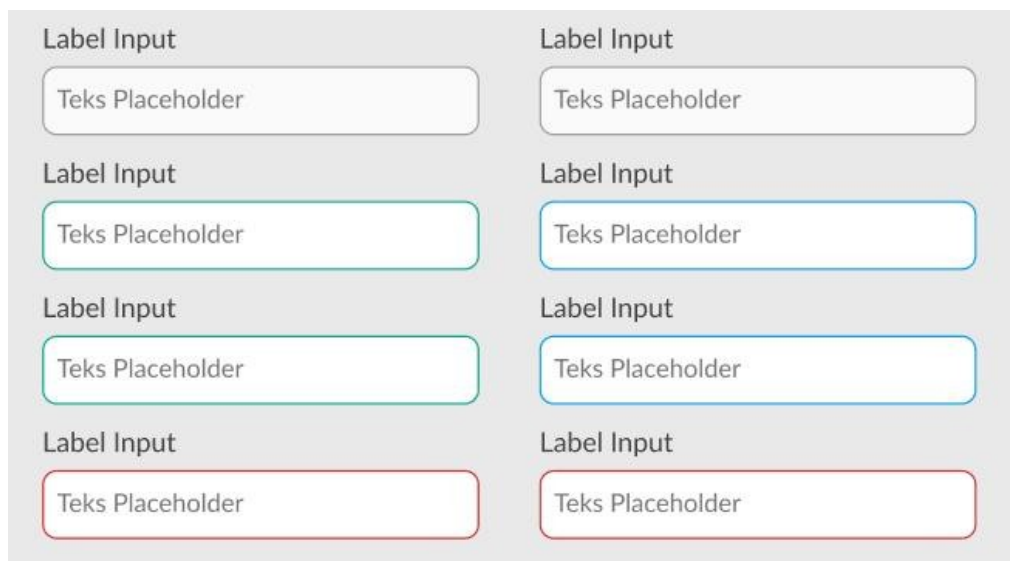
area yang lebih besar daripada *text field* biasa. *Text area* biasanya digunakan ketika pengguna perlu memasukkan teks dalam jumlah yang lebih besar, seperti untuk menulis komentar, mengisi pesan, atau memasukkan konten artikel.



Gambar 5.20. Contoh *template text area*.

1. *Text Field*

Text field (atau kadang disebut juga sebagai *input field*) pada web adalah elemen antarmuka pengguna yang memungkinkan pengguna untuk memasukkan teks atau data lainnya. Disini kita menggunakan *text field* dengan *style outline* seperti gambar dibawah ini:



Gambar 5.21. Contoh *template text field*.

D. *User Interface* Aplikasi Berbasis Mobile

1. Bahasa Pemrograman

Aplikasi berbasis *mobile platform* yang akan dikembangkan oleh Pemerintah Kabupaten Lembata menggunakan beberapa standar

bahasa pemrograman yang umum digunakan, antara lain (tidak terbatas pada):

- a. Java
Digunakan untuk pengembangan aplikasi pada gawai dengan sistem operasi android.
- b. Kotlin
Digunakan untuk pengembangan aplikasi pada gawai dengan sistem operasi android.
- c. Swift
Digunakan untuk pengembangan aplikasi pada gawai dengan sistem operasi iOS.
- d. Objective-C
Digunakan untuk pengembangan aplikasi pada gawai dengan sistem operasi iOS.
- e. Flutter
Dapat digunakan untuk pengembangan aplikasi pada gawai dengan sistem operasi android dan iOS.

2. Desain Template

a. Tipografi

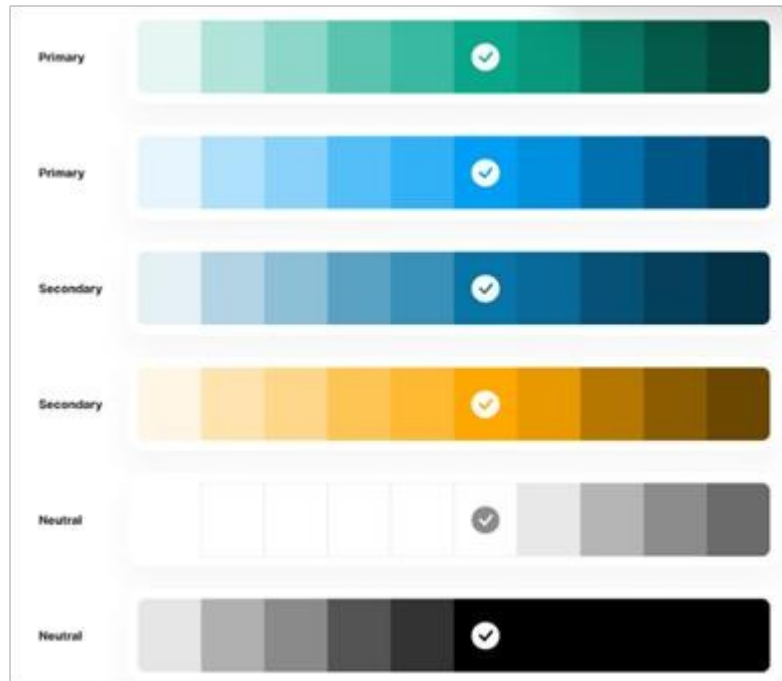
Pemilihan font yang tepat sangat penting agar isi konten yang ditampilkan dapat menarik perhatian tetapi tetap mudah dan jelas terbaca. Untuk itu, font yang direkomendasikan adalah jenis font *Sans Serif*, seperti berikut ini:



Gambar 5.22. Contoh *template typografi*.

b. Colour

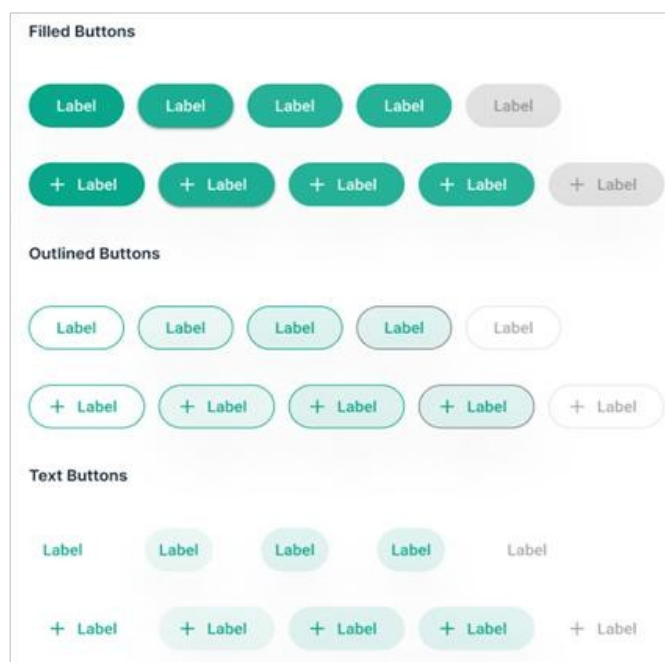
Warna memiliki peran besar terhadap pengalaman dan kenyamanan pengguna dalam menggunakan aplikasi. Sistem warna dibagi menjadi 3 kelompok, yaitu primer, sekunder dan netral.



Gambar 5.23. Contoh *template colour*.

c. *Button*

Button digunakan untuk membantu pengguna dalam mengambil tindakan atau memberi perintah tertentu.



Gambar 5.23. Contoh *template button*.

d. *Checkbox*

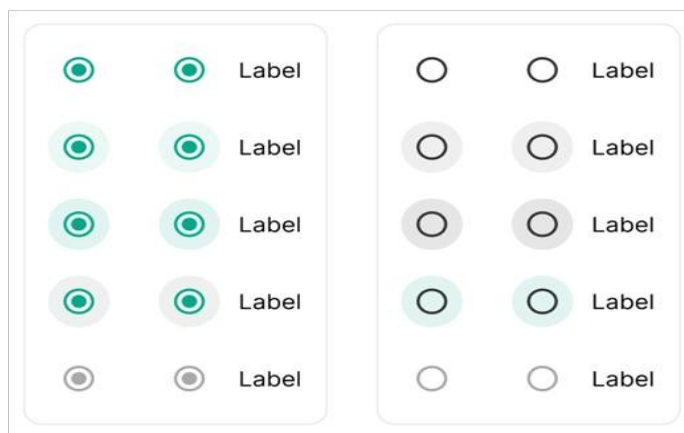
Checkbox digunakan supaya pengguna dapat memilih satu item atau lebih dari opsi yang tersedia. Selain itu juga dapat digunakan untuk mengaktifkan atau menonaktifkan item.



Gambar 5.24. Contoh *template checkbox*.

e. *Radio Button*

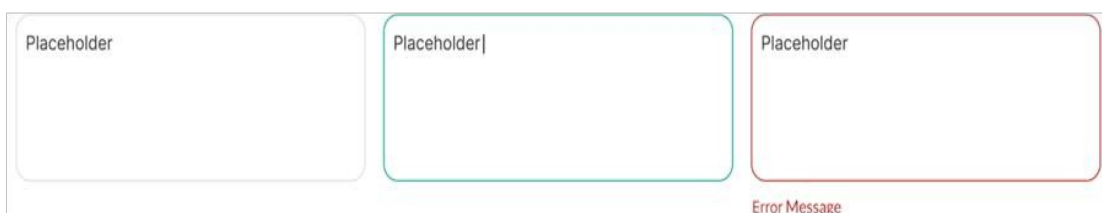
Radio button digunakan supaya pengguna dapat memilih satu opsi dari serangkaian pilihan yang tersedia.



Gambar 5.25. Contoh *template radio button*.

f. *Text Area*

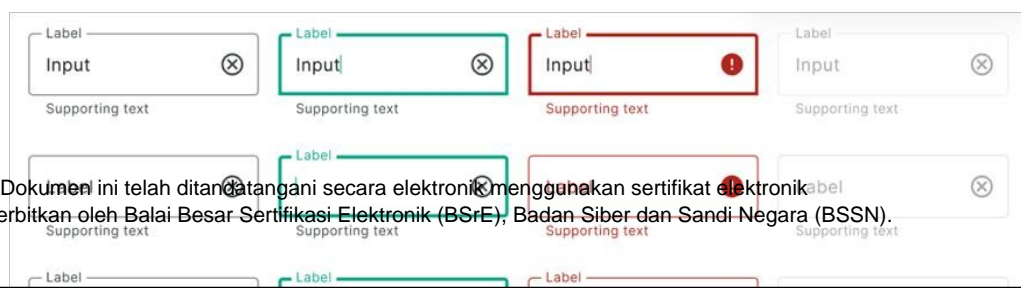
Text area digunakan supaya pengguna dapat memasukkan teks yang panjang.



Gambar 5.25. Contoh *template text area*.

g. *Text Field*

Text field digunakan supaya pengguna dapat memasukkan teks yang pendek atau tidak lebih dari satu baris.



Gambar 5.26. Contoh *template text field*.

BAB VI PENUTUP

Seluruh elemen pelaksana pembangunan/pengembangan aplikasi khusus di Lingkungan Pemerintah Kabupaten Lembata wajib mematuhi standarisasi teknis ini dengan menerapkan prinsip efisiensi, efektifitas, transparansi, dan akuntabilitas serta partisipatif dalam melaksanakan proses pembangunan/pengembangan aplikasi khusus untuk mencapai tujuan dan sasaran organisasi.

Standarisasi teknis ini merupakan salah satu alat untuk memastikan perencanaan, pengendalian dan evaluasi terhadap aplikasi yang dibangun/dikembangkan di lingkungan Pemerintah Kabupaten Lembata. Adanya standarisasi teknis ini akan memudahkan pelaksanaan, pengendalian dan evaluasi proses pembangunan/pengembangan aplikasi khusus dengan tetap memperhatikan tugas pokok dan fungsi serta tanggung jawab masing-masing sesuai ketentuan peraturan perundang-undangan yang berlaku.

BUPATI LEMBATA,



P. KANISIUS TUAQ