

NAMA : HAPPY ADRIAN PRADANA PUTRA

NIM : 20230801539

UTS PEMWEB

Systematic Literature Review (SLR)

Topik: Konversi Template HTML5 & CSS3 Statis menjadi Aplikasi Web Dinamis berbasis Laravel (melibatkan Blade, Admin Panel seperti Filament/Nova, dan praktik migrasi)

Tujuan SLR: Mengumpulkan, menilai, dan mensintesis bukti akademik dan praktikal tentang metode, alat, tantangan, dan outcome saat mengonversi template statis ke aplikasi Laravel dinamis.

1. Pertanyaan Riset (Research Questions)

1. Metode apa saja yang dilaporkan untuk mengonversi template HTML/CSS statis menjadi views Blade Laravel yang dinamis?
 2. Peran dan efektivitas admin panel generator (Filament, Nova, Backpack, dll.) dalam mengurangi effort migrasi dan maintenance?
 3. Kendala teknis umum (routing, asset management, SEO, state management) yang muncul selama konversi, dan solusi yang diusulkan?
 4. Apakah ada pendekatan otomatis (tools/transformasi) untuk mempermudah migrasi — dan seberapa matang mereka?
 5. Gap riset dan rekomendasi penelitian/praktik lanjutan.
-

2. Strategi Pencarian (Search Strategy)

Database & Sumber: Google Scholar, Scopus, IEEE Xplore, ScienceDirect, arXiv, ResearchGate, blog teknis/guide (Dev.to, Medium), dokumentasi resmi (Laravel, Filament, Nova), forum (StackOverflow), dan video tutorial bila relevan.

Periode: Semua tahun sampai 2025 (inklusif).

Kata kunci (sample):

- "convert HTML template to Laravel", "HTML to Blade conversion",
- "Laravel Blade templating", "Laravel admin panel Filament Nova comparison",
- "migrating website to CMS", "automatic website generation systematic review".
(Pencarian contoh dan panduan konversi yang praktikal juga disertakan dari blog dan tutorial).

Catatan: sebelumnya memang ada SLR terkait *automatic website generation* dan migrasi aplikasi web yang mereview teknik otomatisasi dan metode migrasi — ini relevan untuk konteks otomasi konversi template ke aplikasi. [ScienceDirect+1](#)

3. Kriteria Seleksi (Inclusion / Exclusion)

Inklusi:

- Studi empiris, review, atau studi kasus yang membahas migrasi website, konversi template, atau penggunaan Laravel Blade untuk aplikasi dinamis.
- Artikel teknis/guide berisi langkah-langkah konversi atau tool yang nyata (blog, tutorial resmi).
- Perbandingan admin panels (Nova, Filament, Backpack, dll.).

Eksklusi:

- Konten non-Indonesia/English yang tidak dapat diakses atau tanpa isi teknis.
 - Artikel yang hanya membahas front-end (mis. murni CSS frameworks) tanpa konteks konversi ke backend.
-

4. Proses Screening & Quality Assessment

- Screening awal berdasarkan judul/abstrak → full-text reading untuk artikel terpilih.
 - Kualitas dinilai lewat checklist adaptasi: kejelasan tujuan, metodologi, bukti praktis (kode/contoh), serta validitas klaim (replikasi/hasil). Untuk artikel non-akademik (blog/tutorial) dinilai berdasarkan kelengkapan langkah, contoh kode, dan pembaruan (recency).
-

5. Hasil Pencarian (Ringkasan Bukti Utama)

Jumlah studi/artefak yang direview: (mixed sources — akademik + praktikal). Contoh temuan penting:

1. Blade sebagai engine templating yang memudahkan pemisahan view dan logika — banyak sumber praktis menjelaskan langkah konversi (memecah layout, partials, yield/section, komponen Blade) sebagai best practice. [Prateeksha Web Design+1](#)
2. Panduan step-by-step konversi di blog/tutorial (memindahkan asset, membuat layouts, route, controllers, dan migrasi konten ke DB) banyak tersedia dan menjadi referensi utama praktisi. [webcitylab.com+1](#)
3. Admin panel generators (Filament, Nova, Backpack, dll.) signifikan menurunkan effort pembuatan CRUD & manajemen konten, dengan trade-off: biaya (Nova berbayar) vs. kemudahan & komunitas (Filament open-source). Perbandingan praktis dan panduan pemilihan dijumpai di blog teknis/CTO guides. [Hybrid Web Agency+1](#)
4. Terdapat literatur yang membahas migrasi/modernisasi aplikasi web (metodologi migrasi, ADM-inspired approaches) — berguna untuk merancang proses migrasi terstruktur. [SSRN+1](#)

5. Automated approaches / tools untuk konversi statis→dinamis ada, namun sebagian besar fokus ke *automatic website generation* atau generator static-site yang memakai Blade/template; belum banyak tool yang langsung mengubah multipage HTML template menjadi Laravel project sempurna tanpa intervensi manual. [ScienceDirect+1](#)
-

6. Tema Sintesis (Thematic Synthesis)

Dari studi dan sumber praktis, muncul beberapa tema utama:

A. Praktik Konversi Manual (umum dan dominan)

- Pisahkan layout global (header/footer) menjadi layouts.app dengan @yield dan @section.
- Ubah setiap halaman statis menjadi Blade view; gunakan partials untuk ulangan (navbar, footer, cards).
- Asset pipeline: pindahkan CSS/JS ke public/ dan gunakan mix() atau vite() (tergantung setup Laravel).
- Routing: replace link statis .html dengan route() atau URL dinamis (slug), buat Controller & Model sesuai domain.
(Banyak panduan teknis mendukung cara ini). [webcitylab.com+1](#)

B. Penggunaan Blade Components & Reusable Partials

- Komponen Blade meningkatkan maintainability; cocok untuk elemen UI yang diulang (card anime, thumbnail, rating badge). [Prateeksha Web Design](#)

C. Admin Panel Generator sebagai Pengurang Effort

- CRUD cepat, validasi, file upload, dan resource management jadi lebih cepat pakai Filament/Nova/Backpack. Pilihan tergantung kebutuhan, biaya, dan preferensi stack. [Hybrid Web Agency+1](#)

D. Tantangan Teknis Umum

- Routing & permalinks: migrasi link struktural dan memastikan slug konsisten.
- Asset & dependency conflicts: versi Bootstrap/JS kadang perlu disesuaikan.
- SEO & redirects: perlu setup redirect/301 untuk menghindari broken links.
- Automasi terbatas: belum ada alat yang 100% otomatis untuk konversi penuh ke Laravel; banyak tahap tetap manual atau semi-otomatis. [ScienceDirect+1](#)

7. Gap Riset & Kesenjangan Praktis

1. Kurangnya studi empiris kuantitatif yang mengukur pengurangan maintenance effort ketika menggunakan admin panel tertentu untuk migrasi template ke Laravel (mis. metrik waktu, bug, cost).

2. Minimnya tooling otomatis yang secara khusus mengonversi multipage Bootstrap templates ke struktur Laravel (views, routes, controllers, models) tanpa banyak intervensi manual.
 3. Studi best-practice untuk SEO-preserving migration dari .html multipage ke single-route dynamic apps relatif jarang terpublikasi secara akademik.
 4. Evaluasi usability/UX pasca-migrasi (apakah UX membaik setelah dinamis?) masih sedikit bukti empiris.
-

8. Rekomendasi Praktis (untuk developer / tim migrasi)

1. Gunakan pendekatan hybrid: proses manual terstruktur + admin panel generator (Filament/Nova) untuk CRUD. Filament direkomendasikan bila mau open-source & cepat; Nova jika butuh dukungan enterprise/fitur lebih polished. [DEV Community+1](#)
 2. Checklist migrasi singkat:
 - Auditing semua file HTML & assets → buat mapping halaman → buat ERD sederhana jika ada data dinamis.
 - Setup Laravel project → buat layouts & partials → konversi halaman → buat Routes & Controllers → buat Models & Migrations → bangun Admin Panel untuk content.
 - Test redirects & SEO, setup error handling (404) & caching.
 3. Gunakan Blade Components untuk elemen UI yang berulang agar maintainability tinggi. [Prateeksha Web Design](#)
 4. Automasi parsial: gunakan script untuk memindahkan asset & beberapa markup, tapi siapkan reviewer manual untuk link, JS behavior, dan integrasi backend.
 5. Dokumentasi & backup: catat mapping URL lama ke URL baru, sediakan redirect 301 untuk mencegah 404.
-

9. Rekomendasi Riset Lanjutan (untuk akademisi)

- Studi komparatif kuantitatif antara admin panels (Filament vs Nova vs Backpack) berdasarkan metrik waktu implementasi, cost, dan bug density. [Hybrid Web Agency+1](#)
 - Pengembangan dan evaluasi tool semi-otomatis yang memetakan struktur HTML/Bootstrap ke proyek Laravel (views + resource skeleton).
 - Studi case pada SEO & user engagement sebelum/sesudah migrasi statis→dinamis.
-

10. Keterbatasan SLR ini

- Menggabungkan literatur akademik *dan* sumber praktikal (blog, tutorial) — yang terakhir punya variasi kualitas; namun tetap relevan karena topik sangat praktikal.
 - Pencarian bersifat terbatas (tidak exhaustive seperti database berbayar penuh), tapi sudah mencakup sumber-sumber kunci dan panduan industri hingga 2025.
-

11. Kesimpulan Singkat

- Praktik migrasi yang dominan adalah manual namun terstruktur (layouts, partials, routes, controllers, models) dibantu Blade components untuk reuse. webcitylab.com+1
- Admin panel generators (Filament, Nova, dll.) terbukti memotong effort CRUD dan maintenance; pilihan tergantung kebutuhan & budget. Hybrid Web Agency+1
- Gap riset nyata ada pada automasi penuh dan evaluasi kuantitatif outcome pasca-migrasi — area menarik buat penelitian ataupun pengembangan tooling.