Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
15/1 (2022), xx-xx. DOI: http://dx.doi.org/10.21609/jiki.v15i1.xxx

Redesigning the User Interface of the Smart Research Environment (MySRE) Platform with an Automatic Comparative Analysis Feature

Amalia Kartika Author, Rio Nurtantyana,Arya Bisri and M. Iman Wahyudi

[1,4]Informatics, Fauclty of Science & Technology, UIN SMH Banten, Banten, Indonesian
[2,3]Research Group HCI-V, Data and Information Science, Center, National Research and Innovation Agency, Bandung, West Java, Indonesian

E-mail: Kartikaamalia025@gmail.com, rion003@brin.go.id, aria005@brin.go.id, imanwahyudi@uinbanten.ac.id

Abstract

Abstract*—The Smart Research Environment (MySRE) platform is a web-based digital research ecosystem developed by the Indonesian National Research and Innovation Agency (BRIN) in collaboration with several universities under LPDP funding. This paper presents the redesign and implementation of the MySRE version 2.0 user interface, encompassing nine development areas: main page, authentication, registration, dashboard, session management, session creation, project workspace, comparative analysis feature, and supporting design system components. Development was conducted using an iterative component-driven development (CDD) approach with a technology stack comprising Next.js 15, React 19, Mantine UI v7, and Turborepo monorepo architecture, within the MBKM Internship programme at PRSDI BRIN. Key contributions include improvements to visual consistency and interface responsiveness across the entire platform, as well as the implementation of a new Automatic Comparative Analysis feature offering a multi-tab display with up to six simultaneous analysis sessions and five structured analytical aspects. Responsiveness testing was performed using Chrome DevTools across three representative device classes: mobile (360–414 px), tablet (768–1,024 px), and desktop (1,280–1,920 px). Evaluation demonstrates consistent layout response across all tested screen sizes, with uniform dark mode and light mode support across all platform components. A structured user evaluation with 36 respondents using the System Usability Scale (SUS) and User Experience Questionnaire (UEQ) yielded a mean SUS score of 58.75 ("OK/Marginal") alongside "Excellent" ratings across all six UEQ dimensions (overall mean 1.519), revealing a task-efficiency/experience-quality divergence that informs subsequent design priorities.*

***Keywords*—** *user interface design; smart research environment; comparative analysis; component-driven development; monorepo architecture*

1. Introduction

Managing an ever-expanding body of literature while simultaneously analyzing and synthesizing it into new writing is a persistent operational burden in scientific work. This burden falls hardest on researchers still building their track record: without a workflow supported by adequate tooling, postgraduate students in particular tend to lose disproportionate time to purely administrative reference-handling tasks [1]. Much of this inefficiency traces back to a single structural problem—the tools involved in a literature review are rarely the same tool twice. References live in one application, annotations in another, and the eventual manuscript in yet a third, forcing researchers to manually reconcile information across disconnected environments.

MySRE (Smart Research Environment) was conceived to close that gap: a single web-based ecosystem, built through joint work between BRIN, Universitas Brawijaya, Universitas Pasundan, UIN Sultan Maulana Hasanuddin, and Telkom University under LPDP funding, that keeps reference management, knowledge-graph visualization, AI-assisted writing, and manuscript

drafting inside one continuous environment rather than scattered across separate tools [2].

An earlier iteration of this interface is documented in Hariyanti et al. [1], who implemented five core capabilities—a text editor, an AI discussion feature, reference management, a co-pilot assistant, and a concept map—and recorded a System Usability Scale (SUS) score of 76.75 for that version. That first build, however, stopped short of one capability researchers repeatedly need: a way to systematically compare multiple reference articles against each other rather than reviewing them one at a time. Addressing that specific omission is what motivates the work reported here.

What follows documents the version 2.0 redevelopment of MySRE, carried out through the Merdeka Belajar Kampus Merdeka (MBKM) Internship programme at the Human-Computer Interaction and Visualization Research Group, Research Center for Data and Information Science (PRSDI), BRIN. The work reported contributes on two fronts: a platform-wide interface refinement spanning nine modules to raise visual consistency and responsiveness, and the introduction of an Automatic Comparative Analysis capability—an AI-driven addition that meaningfully broadens what the platform can do.

2. Related Work

2.1 Digital Research Platforms and User Interface

Tools such as Semantic Scholar, ResearchRabbit [3], and Connected Papers illustrate how much appetite exists among researchers for digital support in handling references, yet each tends to specialize narrowly—one for surfacing articles, another for mapping citations—rather than covering analysis, writing, and collaboration under one roof. That narrowness is precisely the space a broader platform like MySRE is positioned to occupy.

Designing the interface layer for a platform of this scope cannot proceed from aesthetic preference alone; it needs to start from who will use it and for what. Preece et al. [4] frame this as user-centered design (UCD)—grounding interface decisions in a working understanding of the context of use, the people doing the using, and the tasks they are actually trying to accomplish, rather than assumptions about them.

2.2 Usability Heuristics and Interface Evaluation

Evaluating whether an interface actually works for its users still leans heavily on Nielsen's heuristic framework [5], and three of its ten principles map onto the MySRE redesign with particular directness: keeping the system's internal state visible to the user through timely feedback, enforcing the same visual conventions everywhere rather than letting each page invent its own, and stripping away decoration that competes with the information users came for. Shneiderman et al. [6] arrive at a compatible position from a different angle, distilling interface design down to eight recurring principles that likewise foreground consistency, clear feedback, and forgiving error handling.

Consistency, in particular, is where component-driven development (CDD) earns its keep in this project [7]. Rather than trusting every page to independently re-implement the same visual rule, CDD packages that rule once inside a reusable component—so enforcing a standard becomes a matter of using the component correctly, not remembering to apply the rule by hand each time.

2.3 AI-Assisted Literature Comparative Analysis

Comparing several reference sources side by side—working out where they agree, where they diverge, and where each one is strong or thin—is a core literature-review task that has historically been done by hand [8], one article at a time, which scales poorly once the reading list grows past a handful of papers. Large language models have begun to change that arithmetic. Alshami et al. [9] show that, given adequately framed context, an LLM can surface shared themes, flag methodological divergence, and point to open gaps across a set of articles with a level of accuracy that makes automation of this task practically viable.

2.4 Modern Frontend Development

Turborepo pushes the same component-sharing logic that underlies CDD up one level, from within a single application to across a family of applications living in one repository [10]—which is precisely the arrangement MySRE's four-application structure depends on. Framework choice matters here too: Lazuardy and Anggraini [11] point to Next.js's server-side rendering, static generation, and mature React ecosystem as concrete reasons it has become a default choice for building interfaces of this kind.

2.5 Prior Research on MySRE

The version of MySRE that preceded this work was built by Hariyanti et al. [1] through a five-phase SDLC Prototyping cycle and evaluated at a SUS score of 76.75, placing it in the "good" band. What this paper adds moves in two directions that first build did not cover: a full interface redesign built on the Mantine UI v7 design system, and an Automatic Comparative Analysis module with no counterpart in the earlier version. The value of that kind of iteration is not merely anecdotal—Setiawan et al. [12] found that redesign grounded in structured evaluation measurably improves user experience once a system is already in active use.

3. Methodology

3.1 Development Approach and Phases

Rather than a single linear build, the MySRE v2.0 interface work moved through the same four-step cycle repeatedly: read the current state of the interface and pin down what it lacked, design a component-level solution for that gap, implement it directly in the working environment, and test it independently before a supervisor evaluation closed the loop [13]. Repeating that cycle, instead of running it once, was what let the specification keep shifting without derailing the work already done.

Across the internship period that cycle unfolded in four rough stretches of time: getting oriented in the system (Weeks 1–2), pinning down the problems worth solving (Weeks 2–3), working out design solutions on paper before touching code (Weeks 3–4), and then the long tail of building and testing iteratively (Weeks 4–22). The internship itself ran under academic and field supervision at PRSDI BRIN from 18 February to 30 July 2026.

3.2 Technology Stack

No new technology environment was introduced for this work—development stayed within the stack already established in the MySRE v2.0 repository, summarized in Table 1.

Table 1. MySRE v2.0 Technology Stack

Role	Technology	Version
App Framework	Next.js (App Router)	15.4.2
Interface Library	React	19.1.1
Static Typing	TypeScript	5.8.3
Design System	Mantine UI	7.17.2
Styling	Tailwind CSS	v4.0
Monorepo	Turborepo	2.8.14
Graph Viz.	vis-network	—
PDF Viewer	react-pdf-highlighter	—
Database	Prisma + PostgreSQL (Supabase)	6.12.0
Authentication	Supabase SSR	2.52.0

3.3 Development Scope

Development encompassed four mutually integrated Next.js applications within the monorepo: (1) *main* (port 3000)—public pages and authentication portal; (2) *profile* (port 3001)—user administration dashboard; (3) *brain* (port 3002)—research session management and comparative analysis module; and (4) *writer* (port 3003)—AI-assisted scientific draft editor. Shared packages ensure component consistency and business logic uniformity across all applications.

3.4 Design System Implementation

Rather than letting each page settle its own visual decisions, colors, type, and layout rules were pulled into one Mantine UI v7 design system that every application draws from: a shared five-color palette (blue #4c6ef5, purple #6366f1, teal #14b8a6, orange #f59e0b, pink #ec4899), a single Times New Roman type hierarchy, breakpoints spanning xs through xl, and dark/light mode handled centrally through *useMantineColorScheme()* together with the CSS *light-dark()* function. The payoff of centralizing these decisions is that a single change ripples through every component that depends on it, instead of requiring a file-by-file update each time.

3.5 Testing Methodology

Testing ran alongside implementation rather than after it, checked along four separate lines: *responsiveness*, verified per page through Chrome DevTools' device emulator across mobile (360–414 px), tablet (768–1,024 px), and desktop (1,280–1,920 px) breakpoints; *color-scheme parity*, toggling each component between light and dark mode to catch readability or contrast problems; *interaction correctness*, confirming each user action produced the expected response; and a *supervisor pass*, where the field supervisor reviewed every finished component as a final quality gate.

3.6 User Evaluation Instruments

Technical testing alone cannot say how the

interface actually feels to use, so a separate round of evaluation gathered that perspective directly from users, using two instruments with established track records for this purpose: the System Usability Scale (SUS) [17] and the User Experience Questionnaire (UEQ) [18]. SUS works through ten items, alternating in polarity and scored 1–5, that roll up into a single 0–100 score; UEQ instead spreads the assessment across six separate dimensions—Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty—each rated on a −3-to-+3 scale. Both were distributed online to 36 respondents, most of them undergraduates drawn from several Indonesian institutions, all of whom had used the MySRE v2.0 interface firsthand once development was complete.

4. Results and Discussion

4.1 MySRE v2.0 Interface Architecture

The MySRE v2.0 interface is organized across four integrated applications as illustrated in Fig. 1. Users experience the flow: Landing Page (*main*) → Authentication → Dashboard (*profile*) → Brainstorming Session (*brain*) → Draft Writing (*writer*). Cross-application transitions use token passing via URL parameters to maintain authentication session. Initial system condition analysis identified four primary problem categories summarized in Table 2.

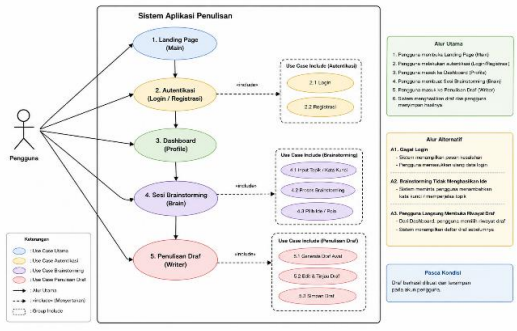
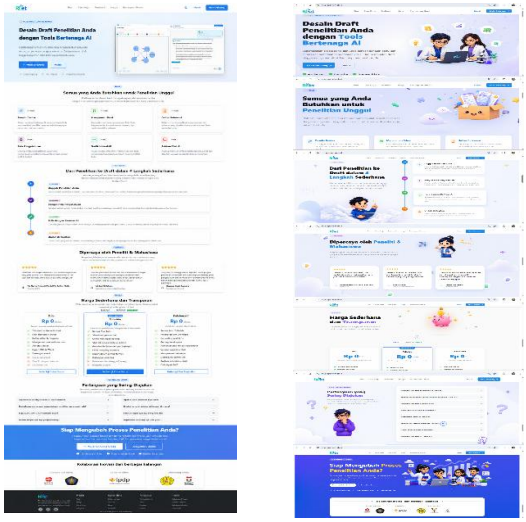


Figure 1. MySRE v2.0 integrated application architecture and user navigation flow.

Table 2. Identified Problem Categories in Initial System Analysis

Category	Description	Affected Pages
Product value communication	Main page did not effectively communicate the platform's value proposition	Homepage
Visual identity and trust	Generic authentication pages without brand identity elements	Login, Register
User engagement	Static dashboard without animation or personalization	Dashboard
Functional limitations	Absence of search, color indicators, and session context menu	Brainstorming

4.2 Main Page Development

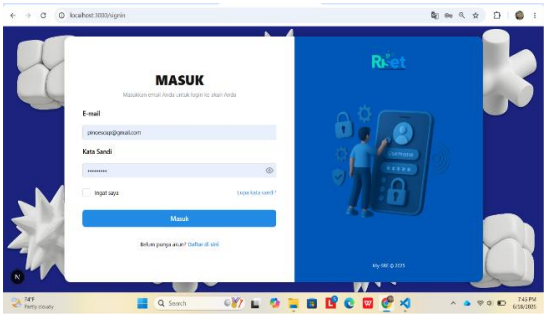


(a) Before redesign *(b) After redesign*

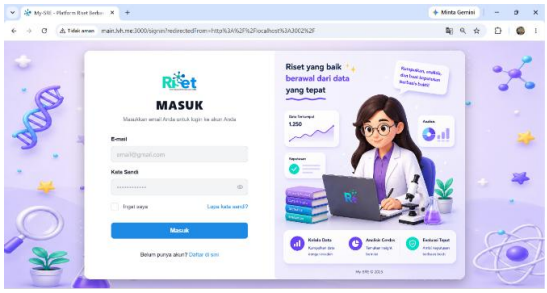
Figure 2. MySRE main page (a) before and (b) after redesign.

The main page (application *main*, port 3000) was rebuilt with an eight-section content architecture guiding prospective users from product introduction to action. The Hero Section was implemented using CSS Modules with two primary keyframe animations: *floatDrift* (7-second float animation) and *heroGlow* (4-second ambient animation). The *CursorBlink* component implements a custom cursor with real-time position updates via *useEffect*. The CTA Section displays logos of six partner institutions as institutional involvement evidence, reinforcing platform credibility.

4.3 Authentication Page Development



(a) Before redesign

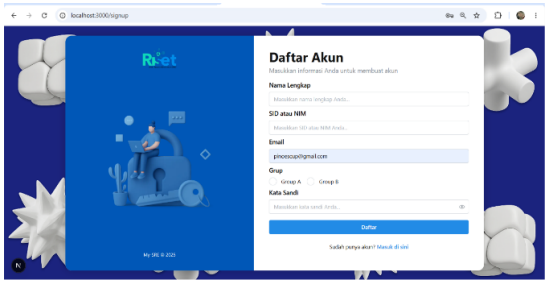


(b) After redesign

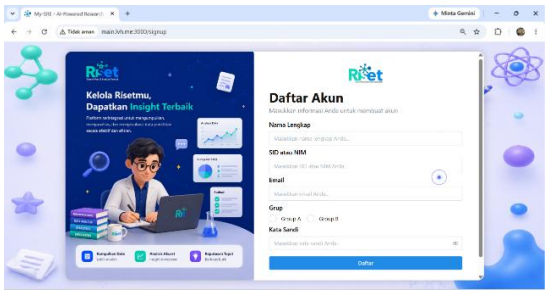
Figure 3. MySRE login page (a) before and (b) after redesign.

Login page redesign was grounded in trust-building principles in UX design [4]. The design solution implemented a split-panel layout: the left panel (55% width) contains the login form with MySRE logo, reinforced input fields, password visibility control, and a "Forgot password?" link; the right panel (45%) displays a full illustration with dark blue background (#1A237E). Password recovery is implemented through a modal dialog enabling recovery link delivery without leaving the page. The post-login routing system implements user-group-based differentiation: Group A users are directed to the *brain* application as active research feature users, while others are directed to *profile*.

4.4 Registration Page Development



(a) Before redesign

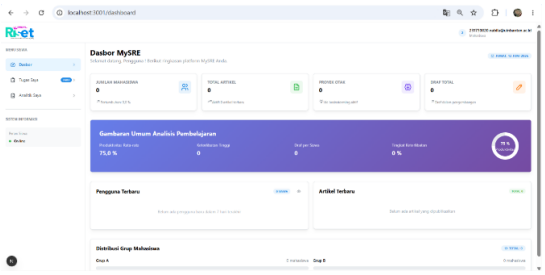


(b) After redesign

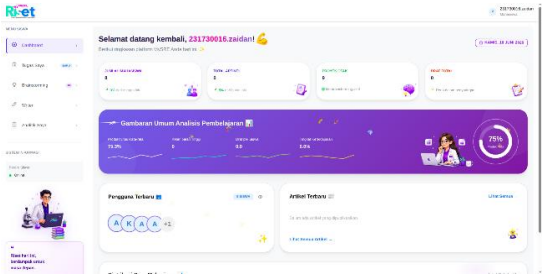
Figure 4. MySRE registration page (a) before and (b) after redesign.

The registration page maintains visual experience continuity from the login page with reversed split-panel proportions: illustration on the left (45%) and registration form on the right (55%). The form includes five fields with different functional considerations: Full Name, SID/NIM (with regex-based numeric validation), Email Address, Group Selection (Group A or B via radio button), and Password. Group selection carries direct functional implications, as Group A users are directed to the *brain* application after each successful login. Client-side validation provides immediate error feedback without awaiting network responses.

4.5 Administrative Dashboard Development



(a) Before redesign



(b) After redesign

Figure 5. MySRE dashboard (a) before and (b) after redesign.

The dashboard (application *profile*, port 3001) is the most frequently visited page by active users. The design decision was that an effective dashboard for this context should welcome users in a manner that feels personal, as an enjoyable platform tends to be used more consistently. The welcome component applies a shimmer gradient text effect on the username and a periodic emoji rotation system via *setInterval* every two seconds. The *AnimatedStatCard* component displays four key metrics with number counting animations using the *easeOutExpo* mathematical function $(1-\text{Math.pow}(2,-10\times t))$ via *requestAnimationFrame*, producing natural deceleration as the target value is approached.

4.6 Brainstorming Session List Development

Each session card on the Brainstorming page (application brain, port 3002) now features: (1) user-customizable cover color reflected as color dots and "ACTIVE" labels; (2) a three-dot context menu providing Edit, Duplicate, and Delete operations with an informative deletion

confirmation modal; and (3) skeleton loader components replacing empty screen states during data retrieval. Real-time search filters cards by session title and description without explicit submit actions. Fig. 6 shows the redesigned brainstorming interface.

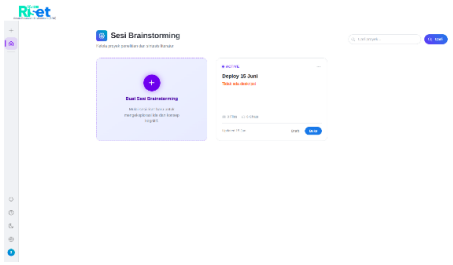


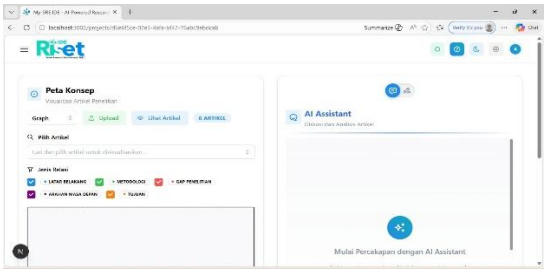
Figure 6. Redesigned brainstorming session list with color indicators and context menu.

4.7 Project Workspace Development

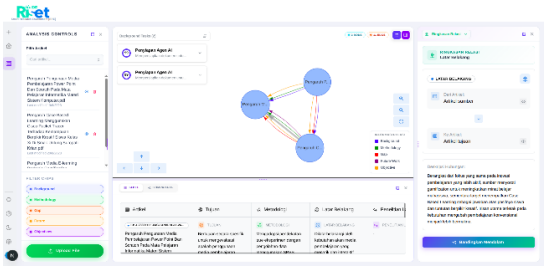
The session detail page (/projects/[id]) is the primary user workspace. A dynamically resizable multi-panel layout integrates large components operating simultaneously. Cross-panel data synchronization is managed through an *eventBus* mechanism based on the observer pattern, enabling automatic display updates without manual refresh. The workspace provides sub-routes for article management and draft writing, making it the central hub connecting the entire research workflow. Panel components and their functions are summarized in Table 3.

Table 3. Project Workspace Panel Components

Panel	Component	Function
Left	NetworkGraph (vis-network)	Knowledge graph visualization
Right	ChatPanel	AI assistant conversation
Right	NodeDetail / ArticleDetailTable	Five-pillar article summary
Right	EdgeDetail	Relation summary between linked articles
Right	AnnotationPanel	User notes, annotations, and analysis history
Overlay	ComparativeModal	Comparative analysis configuration
Floating	ProgressPanel	Article upload progress (non-blocking)



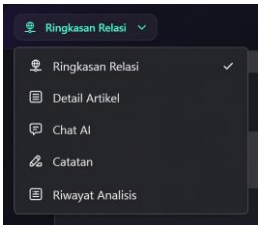
(a) Before redesign



(b) After redesign

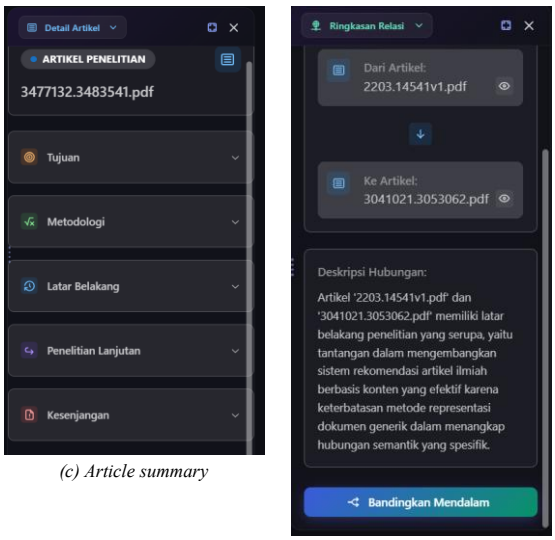
Figure 7. Project workspace (a) before and (b) after redesign.

The right-hand entry in Table 3 is not four independent panels but a single context panel that exposes its views as five selectable modes through one dropdown, so moving from a conversation with the AI assistant to a per-article summary or to the reasoning behind a graph relation happens inside the same panel rather than through a separate window. Figure 8 shows the mode selector alongside three representative modes: the AI assistant, the per-article summary, and the relation summary linking two articles. The relation-summary mode doubles as a second entry point into the Automatic Comparative Analysis feature covered next: selecting “Compare In-Depth” from within it opens a new analysis tab with both linked articles already selected, complementing the manual configuration route through ComparativeModal described in Section 4.8.



(a) Mode selector

(b) AI assistant



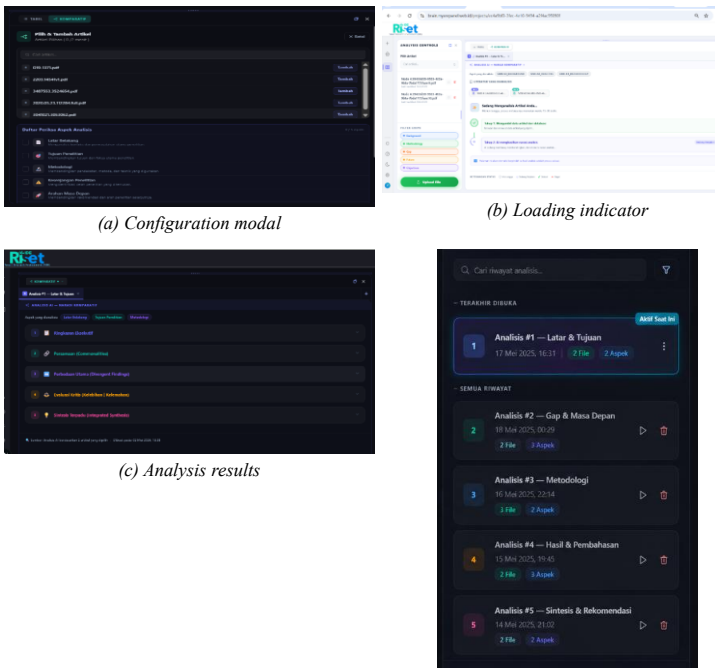
(c) Article summary *(d) Relation summary*

Figure 8. Right-hand context panel: (a) mode selector, (b) AI assistant, (c) article summary, (d) relation summary.

4.8 Automatic Comparative Analysis Feature

Of everything added in version 2.0, the Automatic Comparative Analysis feature carries the most weight. Before it existed, putting several articles side by side for comparison was a manual exercise that could stretch from a few hours to several days depending on how many articles were involved—work this feature now compresses substantially. It is worth being precise about scope here: the contribution documented in this paper is the frontend interface only; the AI logic performing the analysis itself was built by a separate backend team.

The feature interface was implemented through three primary components. First, *ComparativeModal*: a configuration dialog in which users select articles and analysis aspects. Five aspects with unique color codes are available: Background (purple #6366f1), Research Objectives (teal #14b8a6), Methodology (blue #3b82f6), Research Gaps (orange #f59e0b), and Future Directions (pink #ec4899). Second, *AnalysisTabBar*: a multi-tab system supporting up to six simultaneous analyses with automatic labels and slide-in animation. Figure 10(a) shows the tab bar with two concurrent analyses open. Third, *ComparativeAnalysisResult*: a result display component (1,543 lines of code) presenting analysis in structured collapsible panels per aspect, with each article assigned an alphabetical label (A–F) and a unique color from the ARTICLE_PALETTE.

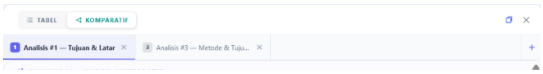


(a) Configuration modal *(b) Loading indicator*

(c) Analysis results *(d) History panel*

Figure 9. Automatic Comparative Analysis feature: (a) configuration modal, (b) loading indicator, (c) analysis results, (d) history panel.

The primary UX innovation is a save-quote-to-annotation mechanism: users can highlight text from analysis results to reveal a “Save as Note” floating action button, using *window.getSelection()* within the *onMouseUp* event handler to detect selection and calculate button position. Saved quotes are displayed with permanent highlighting using a *mark* element with background-color: rgba(255,255,0,0.95). Figure 10(b) shows a saved quote retrieved from the notes panel, linked back to the analysis tab it came from. The *AnalysisLoadingSteps* component displays step-by-step progress, consistent with Nielsen's system status visibility heuristic [5], providing assurance that the system is working rather than failing.



(a) AnalysisTabBar

(b) Saved annotation

Figure 10. Supporting comparative-analysis components: (a) AnalysisTabBar with two concurrent sessions, (b) a saved quote retrieved from the notes panel.

4.9 Implementation Evaluation

In the responsiveness dimension, layout consistency across all components was verified using Chrome DevTools across three device classes. All components utilizing Mantine Grid and Flex respond to screen size changes automatically through responsive props such as *col={{ base: 1, sm: 2, md: 3 }}*. In the color scheme consistency dimension, testing was conducted by manually switching between light and dark mode for each component. Several components required color value adjustments because elements well-rendered in one mode exhibited contrast issues in the other. A concrete example was the Analytics Banner experiencing text readability degradation when switching from dark to light mode, resolved by defining critical color values using CSS variables with the *light-dark()* function.

4.10 User Evaluation via SUS and UEQ

Across the 36 respondents, SUS scores averaged 58.75 (SD = 12.23), with a Cronbach's α of 0.793 indicating the instrument held together reliably. Placed against Bangor et al.'s [17] adjective rating scale, that average lands in the "OK" band at "Marginal" acceptability—not a failing grade, but a clear signal that the interface is usable rather than polished, with navigation consistency the most obvious place to improve. Breaking the score down by item, Q5 (how well features tie together) and Q7 (how easy the system is to pick up) drew the strongest agreement (M = 3.72), while Q6 (inconsistency) and Q8 (confusion) pulled the average down (2.83 and 2.81). That particular pairing—strong marks for integration, weaker marks for learnability—echoes what Gould and Lewis [14] argued decades ago: interfaces become learnable through continuous user feedback loops, not through integration quality alone.

UEQ told a noticeably different story. Benchmarked against 468 products [18], all six dimensions landed in the "Excellent" band, led by Attractiveness (M = 1.704), then Novelty (1.625), Perspicuity (1.528), Efficiency (1.472), Dependability (1.403), and Stimulation (1.292), for an overall mean of 1.519. Table 4 lays out both sets of results side by side.

Table 4. SUS and UEQ Evaluation Results (N = 36)

Metric	Value	Category
SUS Mean Score (N=36)	58.75 (SD=12.23)	OK / Marginal
Cronbach's α (SUS)	0.793	Acceptable
UEQ Attractiveness	1.704	Excellent
UEQ Perspicuity	1.528	Excellent
UEQ Efficiency	1.472	Excellent
UEQ Dependability	1.403	Excellent
UEQ Stimulation	1.292	Excellent
UEQ Novelty	1.625	Excellent
UEQ Overall Mean	1.519	Excellent

Reading a "Marginal" SUS score alongside an "Excellent" UEQ score side by side might look contradictory at first, but the two instruments are simply not measuring the same thing [16]. SUS asks how efficiently a task gets done; UEQ asks how the whole experience felt. Norman [16] draws essentially the same line between behavioral usability and reflective quality, and treats it as unremarkable that the two can pull apart. A platform as feature-dense as MySRE plausibly asks more of a first-time user before its workflow clicks—which is exactly what the SUS number is picking up—while its visual polish and interactive richness are what the UEQ number is rewarding. Weichbroth's [15] systematic review reports this same split showing up repeatedly in feature-dense applications during early adoption, which suggests the pattern here is not idiosyncratic to MySRE. Section 4.12 takes this finding and turns it into concrete refinement priorities.

4.11 Development Challenges and Solutions

Four obstacles stood out over the course of development. *Monorepo architecture complexity* came first: cross-application routing and authentication token handoffs were not documented anywhere, so the only way through was to map how the pieces connected by hand and treat the existing codebase itself as the documentation. *Keeping light and dark mode in sync* turned out to need more than good intentions—it held only once critical color values were pinned down through CSS variables using *light-dark()* and dual-mode checks became a required step in every build cycle, not an afterthought. *Synchronizing data across pages* was solved with an *eventBus* built on the observer pattern, letting a change on one page push an automatic update to a session list on another without pulling in any new library. Last, *aligning on what "good design" meant* proved harder than the technical problems: instructions like "make it more modern," given without a visual reference, routinely led to rework. The fix was procedural rather than technical—pin down a concrete visual reference before implementation starts, not after a first attempt misses the mark.

4.12 Discussion

Taken as a whole, the MySRE v2.0 build makes a reasonably strong case that a Turborepo monorepo can hold interface consistency together across several integrated applications at once [10]. Components meant to be reused—*AnimatedStatCard*, *ProjectCard*, the toast notification system—did get reused consistently across the platform rather than reimplemented per page, which is where the code-duplication savings and the uniform feel both come from [7]. The Automatic Comparative Analysis feature is the clearest step beyond what a literature-review tool typically offers, moving past single-article review into structured comparison across several at once. Pairing its multi-tab layout with the save-quote-to-annotation mechanism ties analysis directly into note-taking rather than leaving them as separate activities—an integration that lines up with what Alshami et al. [9] found about LLMs delivering their real value only when paired with an interface built to use them well, not from the model alone.

Set against Hariyanti et al.'s [1] original build, this iteration covers noticeably more ground—contextual animation, knowledge-graph visualization, multi-tab comparative analysis—and specifically targets two things the first version left unresolved: keeping the interface visually consistent across pages and making it hold up across device sizes. That the redesign moved the needle at all is consistent with what Setiawan et al. [12] report more generally: structured evaluation, fed back into iterative redesign, is what actually produces measurable user-experience gains, rather than redesign for its own sake.

5. Conclusion and Future Works

The work reported here redesigned and rebuilt the MySRE platform's user interface for version 2.0, carried out through the MBKM Internship programme at PRSDI BRIN. It contributes in two concrete ways: nine interface modules were refined for visual consistency, multi-device responsiveness, and overall experience quality, and an Automatic Comparative Analysis feature was added that lets researchers work through five analytical aspects across several reference articles at once inside a single multi-tab interface, rather than one article at a time.

Holding four integrated applications to a consistent look and feel turned out to be a genuinely tractable problem once Turborepo and the Mantine UI v7 design system were doing that work centrally, in keeping with what component-driven development would predict [7]. Tying the save-quote-to-annotation mechanism directly into note management keeps analysis and documentation in the same ecosystem rather than treating them as separate concerns, which is close to the point of building MySRE as a unified workbench in the first place. Responsiveness held up consistently across every screen size tested, which speaks well of the multi-breakpoint approach taken. The 36-respondent evaluation added a layer beyond what technical testing alone could show: a SUS score of 58.75 ("Marginal") points to task efficiency as unfinished business, while all six UEQ dimensions coming back "Excellent" (overall mean 1.519) makes clear that users find the redesigned interface attractive and engaging even where its learning curve still asks too much of them.

One limitation worth naming directly: the 36 respondents were mostly undergraduates rather than the active researchers MySRE is ultimately built for, so how far these usability findings generalize to that primary audience remains an

open question. Three directions follow naturally from that gap: running a formal usability evaluation with active researchers to check whether the present findings hold for that population; measuring, in concrete time-on-task terms, how much the automated comparative analysis actually saves over doing the same comparison by hand; and testing the interface on physical devices rather than only through browser emulation, since real hardware can behave differently than an emulator predicts.

6. Submission

Please submit the manuscript to the Open Journal System at http://jiki.cs.ui.ac.id/ The communication between the authors and the editors will be done through Open Journal System and email. Should you experience problems in the submission, please contact us at jiki@cs.ui.ac.id .

Acknowledgement

The authors gratefully acknowledge the academic and field supervisors for their guidance throughout the MBKM Internship at PRSDI BRIN. This research was supported by the Indonesian Education Fund (LPDP) through the MySRE platform development program.

References

[1] U. Hariyanti, D. Jayadi, T. Afirianto, R. Nurtantyana, and P. Zulvarina, "Front-end development of the MySRE application: A literature review tool for early-career researchers," Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), vol. 12, no. 4, 2025.

[2] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: A literature survey," Int. J. Digit. Libr., vol. 17, no. 4, pp. 305–338, 2016. doi: 10.1007/s00799-015-0156-0.

[3] V. Cole and M. Boutet, "ResearchRabbit," J. Can. Health Libr. Assoc. / J. Assoc. Bibl. Santé Can. (JCHLA/JABSC), 2023. doi: 10.29173/jchla29699.

[4] J. Preece, Y. Rogers, and H. Sharp, Interaction Design: Beyond Human-Computer Interaction, 5th ed. Hoboken, NJ, USA: Wiley, 2019.

[5] J. Nielsen, Usability Engineering. San Diego, CA, USA: Academic Press, 1993.

[6] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos, Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th ed. Hoboken, NJ, USA: Pearson, 2016.

[7] C. Szyperski, D. Gruntz, and S. Murer, Component Software: Beyond Object-Oriented Programming, 2nd ed. London, U.K.: Addison-Wesley / ACM Press, 2002.

[8] Y. Xiao and M. Watson, "Guidance on conducting a systematic literature review," J. Plan. Educ. Res., vol. 39, no. 1, pp. 93–112, 2019. doi: 10.1177/0739456X17723971.

[9] A. Alshami, M. Elsayed, E. Ali, A. E. E. Eltoukhy, and T. Zayed, "Harnessing the power of ChatGPT for automating systematic review process: Methodology, case study, limitations, and future directions," Systems, vol. 11, no. 7, Art. 351, 2023. doi: 10.3390/systems11070351.

[10] Vercel, "Turborepo documentation: Monorepo orchestration," 2024. [Online]. Available: https://turbo.build/repo/docs

[11] M. F. S. Lazuardy and D. Anggraini, "Modern front end web architectures with React.js and Next.js," Int. Res. J. Adv. Eng. Sci., vol. 7, no. 1, pp. 132–141, 2022.

[12] W. F. Setiawan, A. Amirullah, and S. Rochimah, "Enhancing user experience through UI redesign using the UEQ+ method," JOIV: Int. J. Inform. Vis., vol. 9, no. 6, pp. 2674–2682, 2025. doi: 10.62527/joiv.9.6.3596.

[13] I. Sommerville, Software Engineering, 10th ed. Harlow, England: Pearson Education, 2016.

[14] J. D. Gould and C. Lewis, "Designing for usability: Key principles and what designers think," Commun. ACM, vol. 28, no. 3, pp. 300–311, 1985. doi: 10.1145/3166.3170.

[15] P. Weichbroth, "Usability of mobile applications: A systematic literature study," IEEE Access, vol. 8, pp. 55563–55577, 2020. doi: 10.1109/ACCESS.2020.2981892.

[16] D. Norman, The Design of Everyday Things, Rev. and Expanded Ed. New York, NY, USA: Basic Books, 2013.

[17] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," J. Usability Stud., vol. 4, no. 3, pp. 114–123, 2009.

[18] A. Hinderks, M. Schrepp, F. J. Domínguez Mayo, M. J. Escalona, and J. Thomaschewski, "Developing a UX KPI based on the user experience questionnaire," Comput. Stand. Interfaces, vol. 65, pp. 38–44, 2019. doi: 10.1016/j.csi.2019.01.007.