JURNAL ILMIAH SISTEM INFORMASI (JUSI)
VOL 5, No. 1, Januari 2026, pp. 401 - 411
p-ISSN : 2803-1507 (media cetak) | e-ISSN : 2803-1531 (media online)
Link Publikasi: https://ejurnal.provisi.ac.id/index.php/JUISI/issue/view/169

Code Quality Analysis on My School's Web CMS Using SonarQube

Ardian Bayu Syaputra*[1], Depandi Enda[2], Muhammad Asep Subandri[3]
[1,2,3] Teknik Informatika, Politeknik Negeri Bengkalis
Jl. Bathin Alam, 28711, Kota Bengkalis, Bengkalis, Riau, Indonesia
e-mail: kaitonamikaze9@gmail.com*[1], depandienda@polbeng.ac.id[2], Subandri@polbeng.ac.id[3]

ARTICLE INFO

Article History:

Received: 17-11-2025
Revised: 21-12-2025
Accepted: 12-01-2026
Published: 21-01-2026

ABSTRACT

My School CMS is an open-source platform designed to support schools in managing content-based websites and the New Student Admission (PPDB) process. Ensuring the quality of its source code is essential for maintaining security, performance, and system stability. This study evaluates the code quality of the Sekolahku CMS using SonarQube, a continuous inspection tool that performs static analysis to identify bugs, vulnerabilities, code smells, and duplication. The analysis covered three primary project folders. The results show that all projects achieved a Quality Gate status of "Passed," indicating that minimum quality standards were met. From a security perspective, two projects are in the good category, while one project still shows security risks that need to be addressed. Additionally, all projects show low reliability and have not implemented automated testing, which could potentially affect system stability. Although the code structure is generally easy to maintain, the research results indicate that quality improvements are still needed, particularly in reliability, automated testing, and security risk control, to maintain the system's quality and sustainability in the future.

Keywords: *My School CMS, SonarQube, Code Quality Analysis, Performance, Security.*

Abstrak

CMS Sekolahku merupakan sistem open source yang digunakan untuk membantu sekolah dalam mengelola situs web berbasis konten serta proses Penerimaan Peserta Didik Baru (PPDB). Kualitas kode CMS ini sangat penting untuk menjaga keamanan, kinerja, dan stabilitas sistem. Penelitian ini bertujuan untuk menganalisis kualitas kode sumber CMS Sekolahku menggunakan *SonarQube*, sebuah platform inspeksi berkelanjutan yang melakukan analisis statis untuk mendeteksi *bug*, *code smells*, *vulnerabilities*, serta duplikasi kode. Analisis dilakukan pada tiga folder proyek utama. Hasil pengujian menunjukkan bahwa seluruh proyek memperoleh status *Quality Gate*: Passed, menandakan bahwa standar kualitas dasar telah tercapai. Dari sisi keamanan, dua proyek berada pada kategori baik, sedangkan satu proyek masih menunjukkan risiko keamanan yang perlu diperbaiki. Selain itu, seluruh proyek menunjukkan tingkat keandalan yang rendah dan belum menerapkan pengujian otomatis, yang berpotensi memengaruhi stabilitas sistem. Meskipun struktur kode secara umum mudah dipelihara, hasil penelitian mengindikasikan bahwa peningkatan kualitas masih diperlukan, khususnya pada aspek *reliability*, pengujian otomatis, dan pengendalian risiko keamanan, guna menjaga kualitas dan keberlanjutan sistem di masa mendatang.

Kata Kunci: *CMS Sekolahku, SonarQube, Analisis Kualitas Kode, Keamanan, Kinerja.*

1. PENDAHULUAN

DOI: https://doi.org/10.51903/bt753e98

p-ISSN : 2803-1507 e-ISSN : 2803-1531

Perkembangan teknologi informasi telah mendorong institusi pendidikan untuk memanfaatkan sistem informasi berbasis web dalam mendukung kegiatan akademik dan administratif [26], [27], [28]. Salah satu bentuk implementasinya adalah penggunaan *Content Management System* (CMS) yang memungkinkan pengelolaan konten website secara terstruktur, efisien, dan mudah diakses oleh berbagai pihak [18], [24]. CMS banyak digunakan pada lingkungan pendidikan karena mampu mendukung kebutuhan penyampaian informasi, profil institusi, hingga layanan pendaftaran peserta didik secara daring [22], [23].

CMS Sekolahku merupakan salah satu sistem CMS *open source* yang dirancang khusus untuk membantu institusi pendidikan, mulai dari Sekolah Dasar hingga perguruan tinggi, dalam membangun dan mengelola situs web sekolah serta mendukung proses Penerimaan Peserta Didik Baru (PPDB). CMS ini menyediakan fitur pengelolaan konten, informasi akademik, serta layanan publik yang dapat diakses oleh administrator, guru, siswa, maupun masyarakat umum. Keberadaan CMS Sekolahku menjadi penting karena mampu meningkatkan efektivitas penyampaian informasi dan kualitas layanan pendidikan berbasis web [16], [18].

Seiring dengan meningkatnya ketergantungan institusi pendidikan terhadap sistem informasi berbasis web, kualitas perangkat lunak menjadi aspek yang sangat krusial. Kualitas kode sumber berpengaruh langsung terhadap keamanan, kinerja, stabilitas, serta kemudahan pemeliharaan sistem [4], [14]. Kode sumber yang tidak memenuhi standar kualitas berpotensi menimbulkan berbagai permasalahan, seperti *bug*, *code smell*, duplikasi kode, hingga kerentanan keamanan yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab [11], [13], [17].

Berbagai penelitian menunjukkan bahwa analisis kualitas kode merupakan bagian penting dalam siklus pengembangan perangkat lunak modern, khususnya pada pendekatan *Continuous Integration* dan *Continuous Deployment* (CI/CD) [3], [7], [9]. Integrasi proses pengujian dan analisis kualitas kode secara otomatis mampu membantu pengembang dalam mendeteksi permasalahan sejak dini, sehingga risiko kegagalan sistem pada tahap produksi dapat diminimalkan [2], [5], [10]. Oleh karena itu, penerapan metode analisis kualitas kode yang sistematis menjadi kebutuhan penting dalam pengembangan dan pemeliharaan aplikasi berbasis web, termasuk CMS di lingkungan pendidikan.

Salah satu pendekatan yang banyak digunakan dalam analisis kualitas kode adalah *static code analysis*. Pendekatan ini memungkinkan evaluasi kode sumber tanpa perlu menjalankan aplikasi, sehingga mampu mendeteksi berbagai potensi masalah secara lebih cepat dan terukur dibandingkan metode *manual code review* [1], [11]. *SonarQube* merupakan *platform open source* yang populer digunakan untuk analisis statis kualitas kode. *SonarQube* mampu menghasilkan laporan komprehensif terkait *bug*, *code smells*, *vulnerabilities*, serta duplikasi kode berdasarkan standar pengembangan perangkat lunak yang telah ditetapkan [1], [13].

Beberapa penelitian sebelumnya telah membuktikan efektivitas *SonarQube* dalam menganalisis kualitas kode pada berbagai jenis aplikasi, seperti aplikasi berbasis *Laravel*, sistem informasi berbasis web, hingga aplikasi *enterprise* yang terintegrasi dengan CI/CD [1], [2], [11]. Selain itu, *SonarQube* juga dapat digunakan sebagai alat pendukung dalam peningkatan keamanan aplikasi melalui pendekatan *Static Application Security Testing* (SAST) [2]. Meskipun demikian, sebagian besar penelitian masih berfokus pada aplikasi umum, sementara kajian khusus terkait analisis kualitas kode pada CMS pendidikan, khususnya CMS Sekolahku, masih terbatas.

Penelitian terkait CMS Sekolahku sebelumnya lebih banyak menitikberatkan pada aspek fungsionalitas dan pengujian sistem, seperti *black box testing* dengan teknik *Boundary Value Analysis* [16], pengujian penerimaan pengguna, serta pengujian fungsional PPDB online [22], [25]. Penelitian-penelitian tersebut belum secara mendalam membahas kualitas kode sumber dari sisi struktur, keamanan, dan *maintainability*. Padahal, kualitas kode merupakan fondasi utama yang menentukan keberlanjutan dan keandalan sistem dalam jangka panjang [4], [14], [20].

Berdasarkan latar belakang tersebut, permasalahan utama dalam penelitian ini adalah bagaimana menganalisis kualitas kode sumber pada CMS Sekolahku untuk mengidentifikasi potensi *bug*, *code smells*, dan *vulnerabilities* yang dapat memengaruhi keamanan, kinerja, serta stabilitas sistem. Analisis kualitas kode diperlukan untuk memastikan bahwa CMS Sekolahku memenuhi standar pengembangan perangkat lunak yang baik, aman, serta mudah dipelihara [1], [11], [14]. Penelitian ini bertujuan untuk menganalisis kualitas kode pada CMS Sekolahku menggunakan *SonarQube* sebagai alat *static code analysis*. Hasil analisis diharapkan dapat memberikan gambaran komprehensif mengenai kondisi kualitas kode CMS Sekolahku,

p-ISSN : 2803-1507 e-ISSN : 2803-1531

khususnya dari aspek keamanan dan stabilitas sistem. Selain itu, penelitian ini diharapkan dapat menjadi referensi bagi pengembang dan institusi pendidikan dalam meningkatkan kualitas kode CMS serta meminimalkan risiko permasalahan di masa mendatang [1], [11], [17].

2. TINJAUAN PUSTAKA

2.1. *Analisis Kode Statis*

Beberapa penelitian menunjukkan bahwa analisis kode statis digunakan untuk mendukung proses debugging program dengan memeriksa kode sumber tanpa perlu menjalankan aplikasi. Metode ini bertujuan untuk memahami struktur kode, mendeteksi kesalahan logika, serta memastikan bahwa kode telah memenuhi standar pengkodean yang berlaku. Analisis kode statis banyak digunakan dalam bidang *software engineering*, khususnya oleh tim *software development* dan *quality assurance* untuk menjaga kualitas dan keamanan perangkat lunak sejak tahap awal pengembangan [1]. Penggunaan *tools* otomatis dalam analisis kode statis memungkinkan pengembang melakukan pemindaian seluruh kode dalam sebuah proyek secara menyeluruh guna mendeteksi potensi *bug*, *code smells*, dan kerentanan keamanan. Pendekatan ini sejalan dengan konsep *Static Application Security Testing* (SAST) yang menekankan pengujian keamanan aplikasi melalui analisis statis terhadap kode sumber sebelum aplikasi dijalankan [2], [19]. Selain itu, analisis kode statis juga berkaitan dengan pengukuran kualitas perangkat lunak berdasarkan metrik dan standar tertentu. Beberapa penelitian menunjukkan bahwa penerapan analisis kualitas kode dan *software metrics* dapat membantu mengidentifikasi kompleksitas kode serta meningkatkan *maintainability* aplikasi secara keseluruhan [4], [12], [15].

2.2. *SonarQube*

Penelitian terkait *SonarQub*e menyoroti penggunaan platform open-source ini untuk inspeksi berkelanjutan terhadap kualitas kode aplikasi, termasuk analisis otomatis berbagai aspek kode seperti bug, code smells, vulnerability, dan duplication code. *SonarQube* mampu melakukan analisis kode statis secara otomatis dan menghasilkan laporan yang mencakup berbagai aspek kualitas kode, seperti *bug*, *code smells*, *vulnerability*, dan *duplication code* [1], [11]. *SonarQube* mendukung lebih dari 30 bahasa pemrograman yang terbagi ke dalam beberapa edisi, yaitu *Community Edition*, *Developer Edition*, *Enterprise Edition*, dan *Data Center Edition*. *Platform* ini mengklasifikasikan aturan pengkodean ke dalam lima tingkat keparahan, yaitu *Blocker*, *Critical*, *Major*, *Minor*, dan *Info*. Setiap aturan yang diterapkan akan membentuk *Quality Gate* yang digunakan untuk menentukan apakah kualitas kode dinyatakan *Passed* atau *Failed* berdasarkan ambang batas tertentu [1]. Dalam praktik pengembangan perangkat lunak modern, *SonarQube* sering diintegrasikan dengan proses *Continuous Integration* dan *Continuous Deployment* (CI/CD) untuk memastikan kualitas kode tetap terjaga secara berkelanjutan. Integrasi alat analisis kode statis dalam *pipeline* CI/CD terbukti dapat meningkatkan efisiensi proses pengembangan serta mendeteksi permasalahan kode lebih dini [3], [6].

2.3. *CMS Sekolahku*

Beberapa penelitian terkait sistem pendidikan menunjukkan bahwa CMS Sekolahku digunakan untuk mendukung pembuatan dan pengelolaan website sekolah di Indonesia, termasuk pengelolaan Penerimaan Peserta Didik Baru (PPDB) secara daring. CMS ini ditujukan untuk jenjang SD, SMP/sederajat, SMA/sederajat, hingga perguruan tinggi, dan dikembangkan sebagai sistem yang sederhana namun fungsional untuk instansi pendidikan. Secara umum, CMS digunakan untuk mengelola konten website, seperti menambah, mengubah, dan menghapus informasi tanpa memerlukan keahlian teknis yang mendalam. CMS Sekolahku telah memenuhi standar modul yang umum dibutuhkan dalam website sekolah, antara lain modul data siswa, data guru dan staf, data alumni, serta pendaftaran siswa baru, sehingga dinilai layak digunakan sebagai media informasi sekolah [16], [18]. Beberapa penelitian menunjukkan bahwa pengembangan dan pemeliharaan sistem informasi berbasis website di lingkungan pendidikan memerlukan perhatian terhadap kualitas layanan dan keberlanjutan sistem. Oleh karena itu, evaluasi kualitas perangkat lunak, termasuk dari sisi kode program dan fungsionalitas, menjadi aspek penting untuk mendukung peningkatan kualitas layanan pendidikan berbasis teknologi informasi [21], [23].

2.4. *Penelitian Terdahulu*

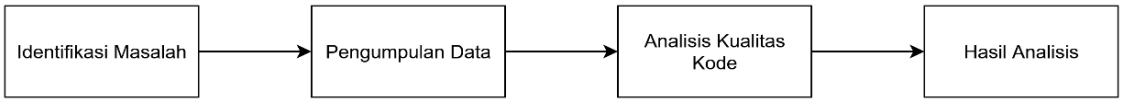
Penelitian oleh [1] menganalisis kualitas kode aplikasi berbasis *Laravel* menggunakan *SonarQube*. Hasil penelitian menunjukkan *bug* dengan 2 *issues* kategori *Major* dan 2 *bug* kategori *Minor*, kemudian ditemukan 2 *issues level low* pada *matriks security hotspots*, dan pada *matriks code smell* ditemukan 31 *issues* dengan 4

p-ISSN : 2803-1507 e-ISSN : 2803-1531

issues kategori *critical*, 16 *issues* kategori *major*, 4 *issues* kategori *minor*, 7 *issues* merupakan info. Penelitian lain oleh Shama dan Chandra (2021) menerapkan *Static Application Security Testing* menggunakan *SonarQube* dalam *pipeline* CI/CD dan menemukan bahwa integrasi analisis statis mampu mendeteksi kerentanan keamanan sejak tahap awal pengembangan [2]. Meskipun beberapa penelitian telah membahas analisis kualitas kode menggunakan *SonarQube*, sebagian besar penelitian berfokus pada aplikasi umum atau *framework* tertentu. Belum banyak penelitian yang mengkaji kualitas kode pada CMS pendidikan berbasis web, khususnya CMS Sekolahku.

3. METODOLOGI PENELITIAN

Gambar 1. adalah gambar metode penelitian yang berfokus pada analisis kualitas kode Web CMS Sekolahku. Metode penelitian mencakup beberapa tahap, yaitu identifikasi masalah, pengumpulan data, analisis kualitas kode, dan hasil analisis. Berikut adalah tahapan metode penelitian ini:



Gambar 1. Metode Penelitian

3.1. *Identifikasi Masalah*

Identifikasi masalah pada penelitian ini bagaimana caranya menganalisis kualitas kode untuk mencegah hal-hal yang tidak diinginkan terjadi seperti disalahgunakan oleh pihak yang tidak bertanggung jawab, kebocoran informasi dan resiko lainnya. Maka dilakukanlah analisis untuk memastikan keamanan, kinerja dan stabilitas dari kode sumber dengan mencari apakah terdapat *bug*, *code smells*, dan *vulnerability* yang ada pada kode sumber.

3.2. *Pengumpulan Data*

Dalam penelitian ini, pengumpulan data dilakukan dengan mengunduh kode sumber dari web CMS Sekolahku yang akan dianalisis. Kode sumber ini akan menjadi objek utama dalam penelitian untuk mengevaluasi kualitasnya menggunakan *SonarQube*.

3.3. *Analisis Kualitas Kode*

Tahap analisis kualitas kode merupakan inti dari penelitian ini, yang bertujuan untuk mengevaluasi mutu kode sumber CMS Sekolahku secara objektif menggunakan metode analisis kode statis. Pada tahap ini, kode sumber yang telah dikumpulkan dianalisis menggunakan *SonarQube* untuk mengidentifikasi berbagai permasalahan yang berkaitan dengan kualitas perangkat lunak. Analisis difokuskan pada beberapa parameter utama, yaitu *Security*, *Reliability*, *Maintainability*, *Accepted Issues*, *Coverage*, *Duplications*, dan *Security Hotspots*. Setiap parameter digunakan untuk menilai aspek yang berbeda, mulai dari potensi kerentanan keamanan, kestabilan sistem, kemudahan pemeliharaan kode, hingga keberadaan pengujian otomatis. *SonarQube* melakukan pemindaian kode secara otomatis berdasarkan aturan (*rules*) dan standar kualitas yang telah ditetapkan. Hasil pemindaian kemudian ditampilkan dalam bentuk metrik, jumlah isu (*issues*), serta peringkat kualitas dari A hingga E.

4. HASIL DAN PEMBAHASAN

4.1. *Hasil Analisis Kualitas Kode*

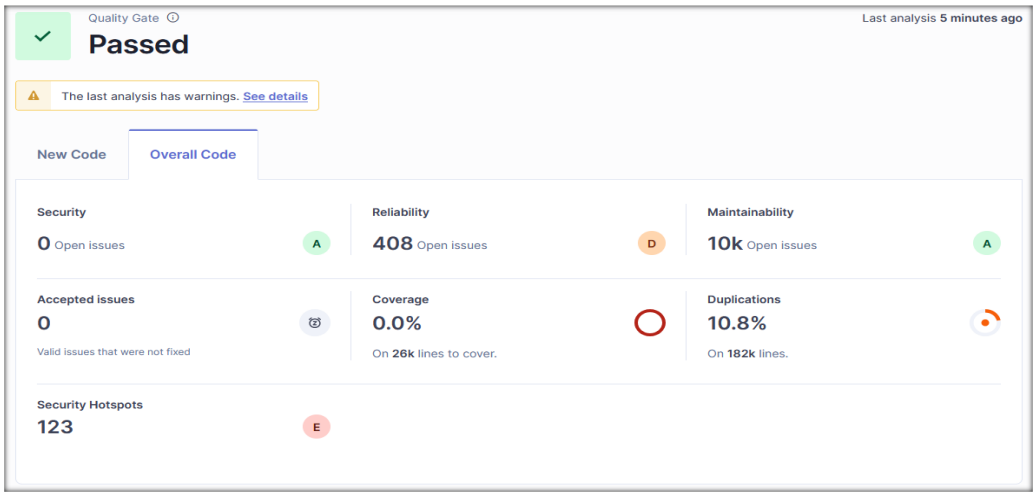
4.1.1. *Hasil Analisis Kualitas Kode Proyek 1*

Pada Gambar 3. Hasil analisis kualitas kode proyek 1, menampilkan status *Quality Gate*: *Passed* yang berarti kualitas kode tersebut telah memenuhi standar dari *SonarQube*. Gambar 3. Hasil analisis kualitas kode proyek 1 menampilkan beberapa metrik yang digunakan untuk menampilkan nilai kualitas kode yaitu:

1. *Security*: Nilai *Security* menunjukkan angka 0 *open issues* dengan peringkat A, yang berarti tidak ditemukan masalah keamanan pada kode sumber.
2. *Reliability*: Terdapat 408 *open issues* dengan peringkat D, menunjukkan adanya potensi kesalahan atau *bug* dalam kode yang dapat mempengaruhi fungsionalitas sistem.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

3. *Maintainability*: Nilai *Maintainability* menunjukkan 10k *open issues* dengan peringkat A, yang menandakan bahwa kode mudah untuk dipelihara.
4. *Accepted Issues*: Nilai 0 menunjukkan bahwa tidak ada masalah yang dibiarkan tanpa perbaikan diverifikasi.
5. *Coverage*: Nilai *Coverage* sebesar 0.0% menandakan bahwa tidak ada pengujian otomatis (*unit test*) yang dijalankan terhadap kode sumber.
6. *Duplications*: Nilai Duplications sebesar 10.8% menunjukkan ada sekitar 10.8% dari total 182.000 baris kode merupakan duplikasi.
7. *Security Hotspot*: Ditemukan 123 *Security Hotspot* dengan peringkat E, yang berarti ada potensi bagian kode yang perlu diperiksa secara manual karena berpotensi menjadi celah keamanan.



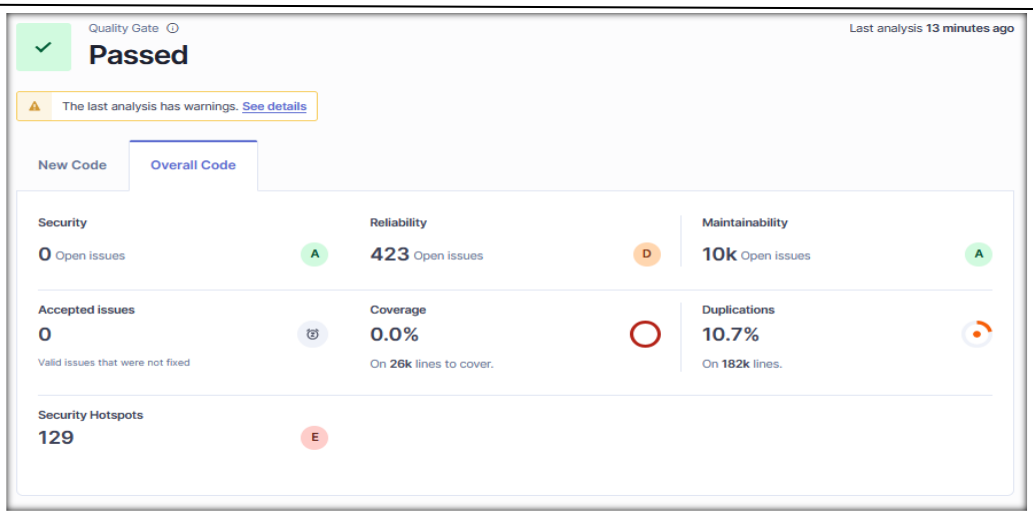
Gambar 3. Hasil Analisis Kualitas Kode Proyek 1

4.1.2. *Hasil Analisis Kualitas Kode Proyek 2*

Pada Gambar 4. Hasil analisis kualitas kode proyek 2, menampilkan status *Quality Gate*: *Passed* yang berarti kualitas kode tersebut telah memenuhi standar dari *SonarQube*. Gambar 4. Hasil analisis kualitas kode proyek 2 menampilkan beberapa metrik yang digunakan untuk menampilkan nilai kualitas kode yaitu:

1. *Security*: Nilai *Security* menunjukkan angka 0 *open issues* dengan peringkat A, yang berarti tidak ditemukan masalah keamanan pada kode sumber.
2. *Reliability*: Terdapat 423 *open issues* dengan peringkat D, menunjukkan adanya potensi kesalahan atau *bug* dalam kode yang dapat mempengaruhi fungsionalitas sistem.
3. *Maintainability*: Nilai *Maintainability* menunjukkan 10k *open issues* dengan peringkat A, yang menandakan bahwa kode mudah untuk dipelihara.
4. *Accepted Issues*: Nilai 0 menunjukkan bahwa tidak ada masalah yang dibiarkan tanpa perbaikan diverifikasi.
5. *Coverage*: Nilai *Coverage* sebesar 0.0% menandakan bahwa tidak ada pengujian otomatis (*unit test*) yang dijalankan terhadap kode sumber.
6. *Duplications*: Nilai *Duplications* sebesar 10.7% menunjukkan ada sekitar 10.7% dari total 182.000 baris kode merupakan duplikasi.
7. *Security Hotspot*: Ditemukan 129 *Security Hotspot* dengan peringkat E, yang berarti ada potensi bagian kode yang perlu diperiksa secara manual karena berpotensi menjadi celah keamanan.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

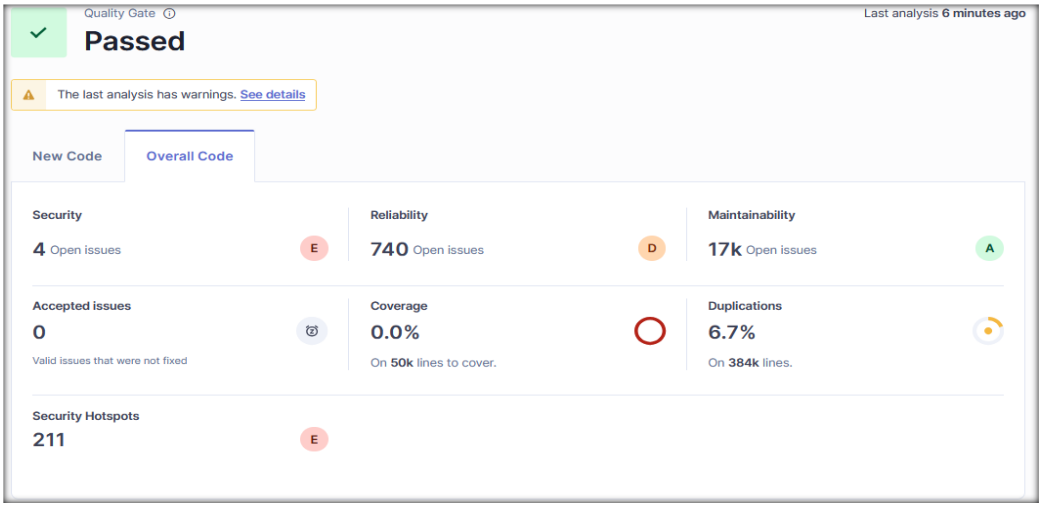


Gambar 4. Hasil Analisis Kualitas Kode Proyek 2

4.1.3. *Hasil Analisis Kualitas Kode Proyek 3*

Pada Gambar 5. Hasil analisis kualitas kode proyek 3, menampilkan status *Quality Gate*: *Passed* yang berarti kualitas kode tersebut telah memenuhi standar dari *SonarQube*. Gambar 5. Hasil analisis kualitas kode proyek 3 menampilkan beberapa metrik yang digunakan untuk menampilkan nilai kualitas kode yaitu:

1. *Security*: Nilai *Security* menunjukkan angka 4 *open issues* dengan peringkat E, nilai ini menunjukkan bahwa terdapat masalah 4 masalah keamanan pada kode sumber yang perlu diperbaiki. Masalah ini bisa mencakup potensi celah keamanan pada sistem.
2. *Reliability*: Terdapat 740 *open issues* dengan peringkat D, menunjukkan adanya potensi kesalahan atau *bug* dalam kode yang dapat mempengaruhi fungsionalitas sistem.
3. *Maintainability*: Nilai *Maintainability* menunjukkan 17k *open issues* dengan peringkat A, yang menandakan bahwa kode mudah untuk dipelihara.
4. *Accepted Issues*: Nilai 0 menunjukkan bahwa tidak ada masalah yang dibiarkan tanpa perbaikan diverifikasi.
5. *Coverage*: Nilai *Coverage* sebesar 0.0% menandakan bahwa tidak ada pengujian otomatis (*unit test*) yang dijalankan terhadap kode sumber.
6. *Duplications*: Nilai *Duplications* sebesar 6.7% menunjukkan ada sekitar 6.7% dari total 384.000 baris kode merupakan duplikasi.
7. *Security Hotspot*: Ditemukan 211 *Security Hotspot* dengan peringkat E, yang berarti ada potensi bagian kode yang perlu diperiksa secara manual karena berpotensi menjadi celah keamanan.



Gambar 5. Hasil Analisis Kualitas Kode Proyek 3

p-ISSN : 2803-1507 e-ISSN : 2803-1531

4.2. *Evaluasi*

Evaluasi kualitas kode dilakukan berdasarkan hasil analisis *SonarQube* terhadap tujuh metrik utama, yaitu *Security*, *Reliability*, *Maintainability*, *Accepted Issues*, *Coverage*, *Duplications*, dan *Security Hotspots*. Evaluasi ini bertujuan untuk menilai tingkat risiko serta implikasinya terhadap kualitas sistem secara keseluruhan. Pada aspek *Security*, Proyek 1 dan Proyek 2 memperoleh peringkat A tanpa temuan isu, yang menunjukkan bahwa tidak terdapat celah keamanan kritis. Sebaliknya, Proyek 3 memperoleh peringkat E dengan sejumlah isu keamanan, sehingga dikategorikan berisiko dan memerlukan perbaikan untuk mencegah potensi kebocoran data dan akses tidak sah.

Pada aspek *Reliability*, ketiga proyek memperoleh peringkat D dengan jumlah isu yang relatif tinggi. Kondisi ini mengindikasikan adanya banyak potensi *bug* atau kesalahan logika yang dapat menurunkan stabilitas sistem dan berpotensi menyebabkan kegagalan fungsi saat aplikasi dijalankan. Evaluasi pada metrik *Maintainability* menunjukkan bahwa seluruh proyek memperoleh peringkat A, yang menandakan struktur kode masih relatif mudah dipelihara. Namun, tingginya jumlah *code smells* berpotensi menurunkan kualitas pemeliharaan dalam jangka panjang apabila tidak dilakukan perapihan kode.

Pada metrik *Accepted Issues*, seluruh proyek menunjukkan nilai 0, yang berarti tidak terdapat isu yang diabaikan tanpa proses peninjauan. Hal ini mencerminkan praktik pengelolaan kualitas kode yang baik dan telah berada dalam batas aman. Nilai *Coverage* pada ketiga proyek sebesar 0.0%, yang menunjukkan belum diterapkannya pengujian otomatis. Kondisi ini meningkatkan risiko kesalahan yang tidak terdeteksi pada tahap pengembangan, sehingga penerapan *unit testing* diperlukan untuk meningkatkan keandalan sistem.

Pada aspek *Duplications*, tingkat duplikasi kode pada ketiga proyek masih berada di atas batas ideal. Kondisi ini berpotensi menyulitkan proses pemeliharaan dan meningkatkan risiko kesalahan ketika dilakukan perubahan kode, sehingga diperlukan penerapan modularisasi dan penggunaan kembali fungsi. Terakhir, evaluasi *Security Hotspots* menunjukkan bahwa seluruh proyek memperoleh peringkat E dengan jumlah *hotspot* yang tinggi. Hal ini mengindikasikan adanya bagian kode yang perlu ditinjau secara manual karena berpotensi menimbulkan risiko keamanan, sehingga diperlukan penerapan praktik pengkodean yang aman secara konsisten.

5. DISKUSI

Hasil analisis menunjukkan bahwa seluruh proyek CMS Sekolahku memperoleh status *Quality Gate: Passed*, yang mengindikasikan bahwa standar kualitas dasar menurut SonarQube telah terpenuhi. Temuan ini memberikan gambaran objektif mengenai kondisi kualitas kode sumber CMS Sekolahku dari sisi keamanan, keandalan, dan kemudahan pemeliharaan sesuai dengan tujuan penelitian. Meskipun demikian, tingginya jumlah isu pada metrik *reliability* serta tidak adanya penerapan *automated testing* menunjukkan bahwa stabilitas sistem dalam jangka panjang masih menghadapi potensi risiko. Kondisi ini menggambarkan bahwa pemenuhan *quality gate* lebih merefleksikan kelayakan awal kode dibandingkan kesiapan sistem untuk pemeliharaan dan pengembangan berkelanjutan.

Jika dibandingkan dengan penelitian terdahulu, temuan penelitian ini sejalan dengan [1], [11] yang menunjukkan bahwa aplikasi berbasis web dapat memenuhi standar kualitas SonarQube meskipun masih memiliki isu signifikan pada aspek *reliability* dan *code smells*. Hasil ini juga konsisten dengan temuan [2] yang menekankan peran *static code analysis* dalam mengidentifikasi potensi masalah kode sejak tahap awal pengembangan. Berbeda dengan sebagian penelitian sebelumnya yang berfokus pada aplikasi umum atau framework tertentu, penelitian ini mengkaji CMS pendidikan berbasis web dengan karakteristik pengguna dan kebutuhan sistem yang spesifik. Perbedaan konteks ini menunjukkan bahwa sistem informasi pendidikan memiliki tantangan tersendiri dalam menjaga kualitas kode yang stabil dan mudah dipelihara.

Analisis pada metrik *security* dan *security hotspots* menunjukkan adanya variasi tingkat risiko antar proyek, di mana satu proyek masih memiliki isu keamanan dengan tingkat keparahan tinggi. Variasi ini dapat dijelaskan oleh perbedaan kompleksitas kode, struktur modul, dan jumlah baris kode yang dianalisis, yang berpengaruh terhadap potensi munculnya celah keamanan. Temuan ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa peningkatan kompleksitas sistem cenderung meningkatkan risiko *vulnerability* apabila tidak diimbangi dengan praktik pengkodean yang aman [14], [17]. Kondisi tersebut menggambarkan bahwa penggunaan CMS yang telah digunakan secara luas tidak selalu menjamin tingkat keamanan yang seragam pada setiap implementasi.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

Dari sisi teoretis, hasil penelitian ini memperkuat pandangan bahwa kualitas perangkat lunak merupakan konstruksi multidimensi yang mencakup *maintainability*, *reliability*, dan *security* secara terintegrasi. Dari sisi praktis, temuan penelitian memberikan gambaran bagi pengembang CMS Sekolahku mengenai area kode yang perlu mendapatkan perhatian lebih, seperti penerapan *unit testing*, pengurangan duplikasi kode, dan penanganan *security hotspots*. Namun, penelitian ini masih memiliki keterbatasan karena hanya mengandalkan pendekatan *static code analysis* tanpa melibatkan pengujian dinamis atau evaluasi performa sistem secara langsung. Keterbatasan tersebut menunjukkan bahwa hasil analisis lebih merepresentasikan kondisi struktural kode dibandingkan perilaku sistem pada lingkungan operasional nyata.

6. KESIMPULAN DAN SARAN

6.1. *Kesimpulan*

Berdasarkan hasil analisis kualitas kode pada folder proyek 1, proyek 2, dan proyek 3 menggunakan *SonarQube* terhadap CMS Sekolahku, dapat disimpulkan bahwa ketiga proyek tersebut telah memenuhi standar kualitas dengan status *Quality Gate*: *Passed*. Hal ini menunjukkan bahwa seluruh proyek telah sesuai dengan standar kualitas yang ditetapkan oleh *SonarQube*. Berdasarkan hasil pengujian, diperoleh temuan sebagai berikut:

1. *Security*: proyek 1 dan proyek 2 mendapatkan peringkat A dengan 0 *open issues*, yang menunjukkan tidak ditemukan masalah keamanan dengan tingkat risiko berarti. Namun, proyek 3 mendapatkan peringkat E dengan 4 *open issues*, yang menunjukkan adanya kerentanan keamanan dengan tingkat risiko tinggi dan perlu segera diperbaiki.
2. *Reliability*: proyek 1 memiliki 408 *open issues*, proyek 2 memiliki 423 *open issues*, dan proyek 3 memiliki 740 *open issues*, seluruhnya dengan peringkat D. Peringkat ini menunjukkan bahwa banyak temuan berada pada level risiko *Medium* hingga *High*, yang berpotensi memengaruhi stabilitas dan menyebabkan *bug* saat sistem dijalankan.
3. *Maintainability*: Ketiga proyek memperoleh peringkat A, walaupun ditemukan banyak *code smells* (proyek 1 dan proyek 2 sekitar 10k *issues* serta proyek 3 sekitar 17k *issues*). Hal ini menunjukkan bahwa struktur kode masih tergolong mudah dipahami dan dipelihara, meskipun tetap diperlukan perapihan kode.
4. *Accepted Issues*: Ketiga proyek menunjukkan angka 0, yang berarti tidak ada masalah yang diabaikan atau dibiarkan tanpa proses peninjauan.
5. *Coverage*: Ketiga proyek memiliki nilai 0.0%, yang mengindikasikan bahwa belum dilakukan pengujian otomatis (*unit testing*), sehingga belum dapat diketahui secara optimal tingkat keandalan kode dari sisi pengujian.
6. *Duplications*: proyek 1 memiliki tingkat duplikasi kode sebesar 10.8%, proyek 2 sebesar 10.7%, dan proyek 3 sebesar 6.7%. Nilai ini menunjukkan adanya pengulangan kode yang pada level tertentu dapat menurunkan kualitas pemeliharaan sistem.
7. *Security Hotspots*: proyek 1 ditemukan 123 *hotspots*, proyek 2 ditemukan 129 *hotspots*, dan proyek 3 ditemukan 211 *hotspots*, seluruhnya dengan peringkat E. Hal ini menunjukkan adanya bagian kode yang perlu ditinjau secara manual karena berpotensi memiliki risiko keamanan pada level *Medium* hingga *High*.

6.2. *Saran*

Berdasarkan hasil analisis, terdapat beberapa saran untuk pengembangan selanjutnya:

1. Pengembang disarankan untuk memperbaiki kode yang terdeteksi memiliki potensi *bug* pada metrik *Reliability*, agar sistem berjalan lebih stabil dan bebas dari kesalahan logika.
2. Melakukan perbaikan kode pada bagian yang terdeteksi duplikasi agar struktur kode lebih efisien dan mudah dikelola.
3. Menambahkan pengujian otomatis (*unit test*) agar metrik *Coverage* dapat terukur dan membantu dalam mendeteksi kesalahan.
4. Melakukan pemeriksaan manual pada *Security Hotspots*, untuk mengurangi potensi resiko keamanan.
5. Melakukan pemantauan dan analisis ulang menggunakan *SonarQube* secara berkala untuk memastikan bahwa kualitas kode tetap terjaga meskipun terjadi pembaruan sistem di masa mendatang.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

DAFTAR PUSTAKA

[1] A. F. Rahmawati and Y. A. Susetyo, "Analisis Quality Code Menggunakan Sonarqube dalam Suatu Aplikasi Berbasis Laravel," *IT-Explore J. Penerapan Teknol. Inf. dan Komun*., vol. 2, no. 2, pp. 99–103, 2023, doi: 10.24246/itexplore.v2i2.2023.pp99-103.

[2] A. M. Shama and D. W. Chandra, "Implementasi Static Application Security Testing Menggunakan Jenkins Ci/Cd Berbasis Docker Container pada PT. Emporia Digital Raya," *J. Ilm. Inform*., vol. 9, no. 02, pp. 95–99, 2021, doi: 10.33884/jif.v9i02.3769.

[3] A. P. Wahyu and I. Guna Noviantama, "Implementasi Contionous Integration dan Continous Deployment pada Aplikasi Learning Management System di PT. Millennia Solusi Informatika," *J. Ilm. Teknol. Infomasi Terap*., vol. 8, no. 1, pp. 183–186, 2021, doi: 10.33197/jitter.vol8.iss1.2021.744.

[4] G. N. M. Nata, I. M. B. Adnyanan, and G. N. Mega Nata, "Usulan Model Mengukur Kompleksitas Kode Program Pada Perangkat Lunak," *J. Manaj. Inform. dan Sist. Inf*., vol. 4, no. 2, pp. 123–130, 2021, doi: 10.36595/misi.v4i2.335.

[5] A. D. Setyoko and A. Zahra, "Efficiency Comparison of Multi-Environment CI/CD Processes Through Parallel and Sequential Implementations," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 911–925, 2024, doi: 10.57152/malcom.v4i3.1334.

[6] A. Ci and C. Dan Dev-Ops untuk KlodId, "Pengembangan Back-End Service Aplikasi Automatisasi Ci/Cd dan Dev-Ops untuk Klod.Id," *eProceedings of Applied Science*, vol. 9, no. 6, p. 3058, 2023. [Online]. Available: http://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/21657

[7] R. A. Parama, H. Studiawan, and R. J. Akbar, "Implementasi Continuous Integration dan Continuous Delivery pada Aplikasi myITS Single Sign On," *J. Tek. ITS*, vol. 11, no. 3, pp. 264-269, 2022, doi: 10.12962/j23373539.v11i3.99436.

[8] I. Akyun, R. Januari Akbar, and H. Fabroyir, "Otomasi Pengujian Antarmuka Pengguna pada Aplikasi Web myITS HumanCapital di dalam Implementasi CI/CD," *Jurnal Teknik ITS*, vol. 13, no. 1, pp. A20–A25, 2024. [Online]. Available: https://doi.org/10.12962/j23373539.v13i1.125359

[9] A. Alpery and M. A. F. Ridha, "Implementasi Ci/Cd Dalam Pengembangan Aplikasi Web Menggunakan Docker dan Jenkins," *9th Applied Business and Engineering Conference*, pp. 287–296, 2021. [Online]. Available: https://abecindonesia.org/proceeding/index.php/abec/article/view/191

[10] A. D. Setyoko and A. Zahra, "Efficiency Comparison of Multi-Environment CI/CD Processes Through Parallel and Sequential Implementations," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 911–925, Jul. 2024. [Online]. Available: doi 10.57152/malcom.v4i3.1334.

[11] N. Sitorus, A. Sirait, E. Lumbantoruan, F. Panjaitan, H. S. H. Harahap, and C. Y. Simangunsong, "Analisis Code Review Menggunakan Sonarqube Terhadap Aplikasi Rumah Kreatif Toba Berbasis Website," *JCBD: Journal of Computers and Digital Business*, vol. 4, no. 1, pp. 16–23, Jan. 2025. doi 10.56427/jcbd.v4i1.606

[12] N. Sitorus, A. Sirait, E. Lumbantoruan, F. Panjaitan, H. S. H. Harahap, and C. Y. Simangunsong, "Quality Analysis of Low-Code No-Code Application Development Using Iso/Iec 25010 Standard: A Systematic Literature Review," *Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer*, vol. 5, no. 3, pp. 266–273, Nov. 2025. doi 10.55606/juisik.v5i3.1633.

[13] D. B. D. Hanggoro, L. Kurniawati, and Y. Rianto, "Quality Analysis of Low-Code No-Code Application Development Using Iso/Iec 25010 Standard: A Systematic Literature Review," *Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer*, vol. 5, no. 3, pp. 266–273, Nov. 2025. doi 10.35143/jkt.v10i1.6211.

p-ISSN : 2803-1507 e-ISSN : 2803-1531

[14] M. Fadhli, Y. Fitrisia, and D. Nurmalasari, "Pengujian Kualitas Kode Program Aplikasi Bank Sampah DLHK Kota Pekanbaru Menggunakan Code Smell Tools," *Jurnal Komputer Terapan*, vol. 10, no. 1, pp. 86–97, May 2024. doi 10.35143/jkt.v10i1.6211.

[15] U. Sa'adah, D. I. Permatasari, F. F. Hardiansyah, A. A. Yunanto, J. A. N. Hasim, I. Wulandari, M. R. Pahlevi, and D. Q. Shihab, "Tool Refactoring Otomatis untuk Menangani Lazy Class Code Smell dengan Pendekatan Software Metrics," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 9, no. 4, pp. 743–750, Aug. 2022. doi 10.25126/jtiik.2022934646.

[16] S. D. S. Saian, N. L. Kakihary, and T. Wahyono, "Pengujian Content Management System (CMS) Sekolahku Menggunakan Metode Black Box Testing dengan Teknik Boundary Value Analysis," *IT-Explore: Jurnal Penerapan Teknologi Informasi dan Komunikasi*, vol. 1, no. 2, pp. 100–113, 2022. doi 10.24246/itexplore.v1i2.2022.pp100-113.

[17] A. Farisi, D. Priyawati, and F. R. Firdaus, "Pengembangan Sistem Informasi Sekolah Berbasis Website dengan Pengujian Website Vulnerability dan Acceptance Testing," *Indonesian Journal of Computer Science*, vol. 13, no. 4, 2024. doi 10.33022/ijcs.v13i4.4088.

[18] W. Pramusinto, K. Harsanto, and M. D. Syavira, "Perancangan Content Management System (CMS) Website Profil Sekolah dengan Model Cloud Computing SaaS," *IDEALIS: Indonesia Journal of Information System*, vol. 7, no. 1, pp. 1–10, 2024. doi 10.36080/idealis.v7i1.3088.

[19] M. Z. Rayhan and M. A. F. Ridha, "Implementasi DevSecOps dengan Metode Static Application Security Testing (SAST) menggunakan SNYK pada Aplikasi Berbasis Container," *11th Applied Business and Engineering Conference*, pp. 76–83, 2023. https://abecindonesia.org/proceeding/index.php/abec/article/view/342

[20] R. Abdillah, R. Hermawan, W. Hermawansyah, D. P. Agustin, I. Adkha, and N. Alam, "Analisis dan Pengujian Perangkat Lunak Sistem Informasi Pembayaran Sekolah dengan Metode Pengukuran Kualitas SQuaRE," *Jurnal Penelitian Rumpun Ilmu Teknik*, vol. 4, no. 1, pp. 208–217, 2025. doi 10.55606/juprit.v4i1.4834.

[21] J. T. H. Karyono, O. A. Dhewa, F. Y. Baktiar, D. E. Wahyu, and C. Widyastuti, "Pembaruan Fitur Aplikasi pada Sistem Informasi Berbasis Website SMP Negeri 12 Yogyakarta, Daerah Istimewa Yogyakarta, untuk Meningkatkan Kualitas Layanan Pendidikan," *Jurnal Abdi Masyarakat Indonesia*, vol. 5, no. 6, pp. 2699–2708, 2025. doi 10.54082/jamsi.2207.

[22] D. A. Ayu and R. Pambudi, "Implementasi Black Box Testing dan Usability Testing pada Website PPDB Online," *ATCSJ: Applied Technology and Computer Science Journal*, vol. 3, no. 2, 2024. doi 10.33086/atcsj.v7i2.8362.

[23] N. I. Utama, "Pengembangan Aplikasi CMS dan Dashboard Monitoring untuk Website Pendidikan," *Jurnal Altifani Penelitian dan Pengabdian kepada Masyarakat*, vol. 5, no. 2, pp. 11-17, 2025. doi 10.58794/altifani.v5i2.

[24] W. Pramusinto, K. Harsanto, and M. D. Syavira, "Perancangan Content Management System (CMS) Website Profil Sekolah dengan Model Cloud Computing SaaS," *IDEALIS: Indonesia Journal of Information System*, vol. 7, no. 1, pp. 1–10, 2024. doi 10.36080/idealis.v7i1.3088.

[25] Stevanu D. Pratama, Lasimin Lasimin, and M. N. Dadaprawira, "Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence dan Boundary Value," *Jurnal Teknologi Sistem Informasi dan Sistem Komputer (TGD)*, vol. 6, no. 2, 2023, doi 10.53513/jsk.v6i2.8166.

[26] M. Word and D. Silalahi, "Mandiri UNBK Try Out Information System Web Based Using Linear Congruent Method at Madrasah Working Group (KKM) MTs N 2 Demak," *Journal of Technology Informatics and Engineering*, vol. 2, no. 1, pp. 56–73, 2024, doi: 10.51903/jtie.v2i1.176.

[27] K. Khaeriyah, I. Amirullah, A. A. Rahman, N. Natsir, and N. F. Suyuthi, "Management of Entry and Outcomes of General Sub-Sections of Employment and Legal at the Education Office of South Sulawesi," *Journal of Management and Informatics*, vol. 1, no. 3, pp. 22–26, Dec. 2022. [Online]. Available: https://jmi.stekom.ac.id/index.php/jmi/article/view/12

p-ISSN : 2803-1507 e-ISSN : 2803-1531

[28] S. Vivi, M. A. L. Londa, and R. Togo, "Design and Build an Academic Information System at SMPN 1 Wolowaru Based on the Web," *Jurnal Ilmiah Sistem Informasi*, vol. 4, no. 2, pp. 288–307, May 2025, doi: 10.51903/ytgedh17.