

MATERI PEMBELAJARAN KODE DAN QUALITY ANALYSIS

Website Toko Perhiasan Emas Perak Monica

Panduan ringkas tetapi berbobot untuk memahami struktur file, alur kode PHP, relasi database, modul admin dan pelanggan, revisi bug utama, serta cara melakukan pengujian kualitas kode dengan SonarQube.

STACK

**PHP Native, MySQL,
HTML, CSS, JavaScript**

TANGGAL

12 Juni 2026

FOKUS

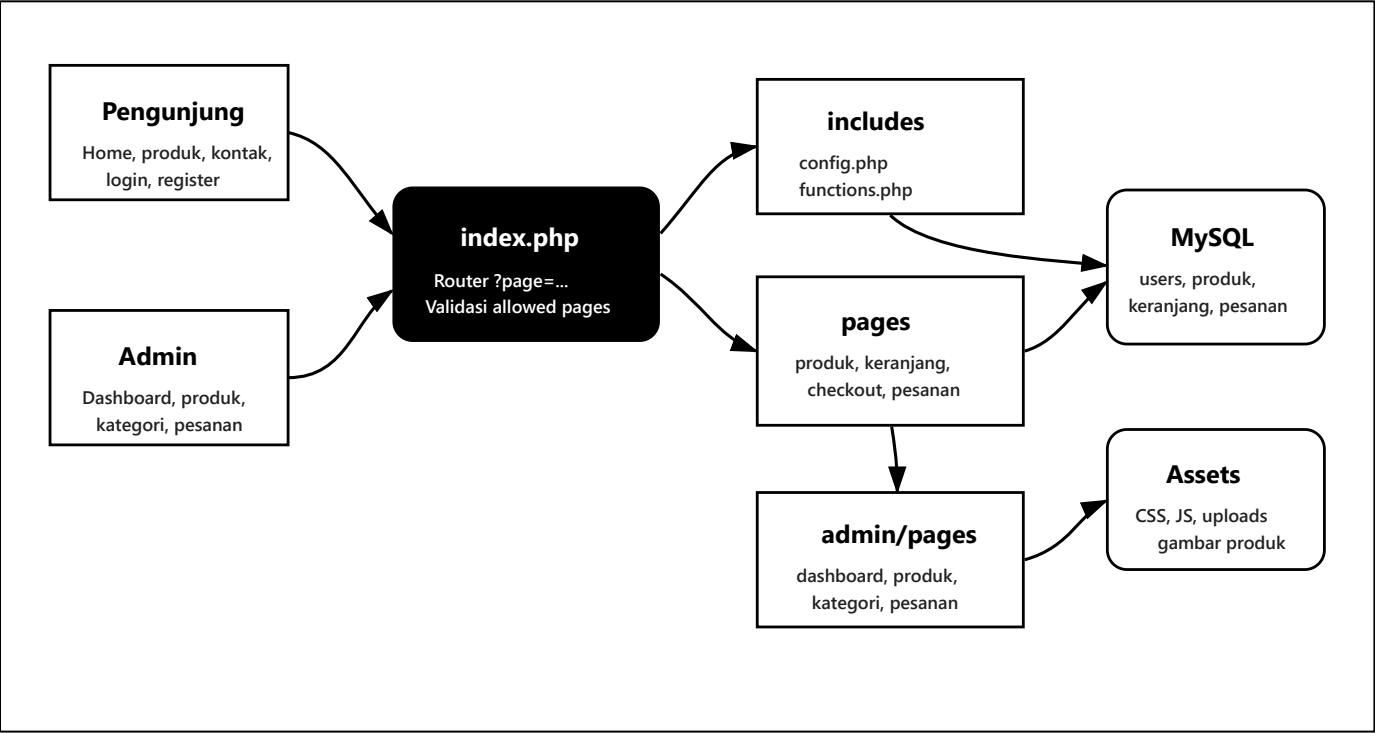
**Kode website, revisi bug,
dan SonarQube**

1. Gambaran Umum Website

Proyek ini adalah website toko perhiasan berbasis PHP native. Aplikasi tidak memakai framework besar, sehingga alurnya mudah dipelajari: satu file router, beberapa file include, halaman publik, halaman admin, aset tampilan, folder upload gambar, dan satu file SQL untuk struktur database awal yang sudah memuat data seed admin dan customer.

Public catalog	Login dan register	Keranjang	Checkout	Pesanan	Admin CRUD	SonarQube scan
File PHP utama 28+	CSS publik style.css	CSS admin admin.css	Akun seed 1 admin + 3 customer			

2. Gambar Arsitektur Aplikasi



Gambar 1. Hubungan pengguna, router, modul halaman, helper, database, dan aset.

3. Struktur File dan Fungsinya

Folder/File	Peran	Yang Dipelajari
index.php	Router utama halaman publik. Nilai <code>?page=...</code> dicek ke daftar <code>\$allowed_pages</code> , lalu file halaman di-include.	Routing sederhana, session logout, proteksi halaman dengan <code>checkLogin()</code> .
includes/config.php	Konfigurasi timezone, koneksi MySQL, session, helper <code>sanitize()</code> , <code>e()</code> , upload gambar, nomor pesanan, format rupiah.	Koneksi database, helper global, escaping output, validasi upload, formatting.
includes/functions.php	Helper login/admin, prepared statement helper, ambil user, ambil produk, daftar produk, keranjang, total keranjang, dan slider.	Pemisahan logic yang sering dipakai banyak halaman dan query yang lebih aman.
includes/header.php dan footer.php	Layout global: navbar, status login untuk JavaScript, link CSS, Font Awesome, Google Fonts, footer, dan JS.	Reusable layout, tampilan konsisten, dan jembatan data PHP ke frontend.
pages/*.php	Halaman publik: home, produk, detail produk, login, register, profil, keranjang, checkout, pesanan, search, kontak, portfolio, testimoni.	MVC sederhana versi PHP native: satu file biasanya memuat query, validasi, dan HTML.
admin/index.php	Router area admin. Memanggil <code>checkAdmin()</code> , menampilkan sidebar, lalu include halaman admin.	Role-based access control sederhana.
admin/pages/*.php	Dashboard, CRUD produk, CRUD kategori, update status pesanan, daftar user, dan manajemen testimoni.	CRUD admin dan query agregasi dashboard.
assets/css/style.css	Style halaman publik: navbar, slider, card produk, form auth, keranjang, checkout, testimoni, responsif.	Design token CSS, layout grid, media query.
admin/assets/css/admin.css	Style panel admin: sidebar, dashboard cards, tabel, form, status pesanan.	Dashboard layout dan UI administrasi.
assets/js/script.js	Slider, add to cart, lightbox, validasi form sederhana, search validation, portfolio slider, rating testimoni.	Interaksi frontend tanpa framework.

Folder/File	Peran	Yang Dipelajari
database.sql	Definisi database <code>perhiasan_db</code> , tabel utama, 4 akun seed, produk, kategori, slider, galeri, dan testimoni.	Relasi tabel, seed data, dan konsistensi skema dengan kode.
uploads/	Gambar produk, slider, dan gambar owner.	File statis yang dipakai oleh halaman produk dan slider.
sonar-project.properties	Konfigurasi SonarQube scanner untuk menganalisis source code proyek.	Scope analisis, exclusions, dan identitas project.

4. Cara Membaca Alur Kode

Entry point

Mulailah dari `index.php`. File ini membaca parameter `page`, mencocokkannya dengan `$allowed_pages`, lalu memanggil file di folder `pages`. Validasi memakai mode strict agar nilai parameter URL tidak lolos karena perbandingan longgar. Contohnya:

```
$allowed_pages = ['home', 'login', 'register', 'produk', 'detail_produk', ...];
$page = isset($_GET['page']) ? $_GET['page'] : $default_page;

if (!in_array($page, $allowed_pages, true)) {
    $page = $default_page;
}
```

Pola ini mencegah user langsung memasukkan nama file bebas melalui URL. Area admin memakai pola yang mirip, tetapi file `admin/index.php` memanggil `checkAdmin()` terlebih dahulu.

Helper global

`config.php` membuat koneksi database melalui `new mysqli(...)`, memulai session, dan menyediakan helper seperti `e()`, `formatRupiah()`, dan `generateNomorPesanan()`. `functions.php` berisi fungsi yang dipakai lintas halaman, misalnya `dbQuery()`, `getProduct()`, `getCartItem()`, dan `getCartTotal()`.

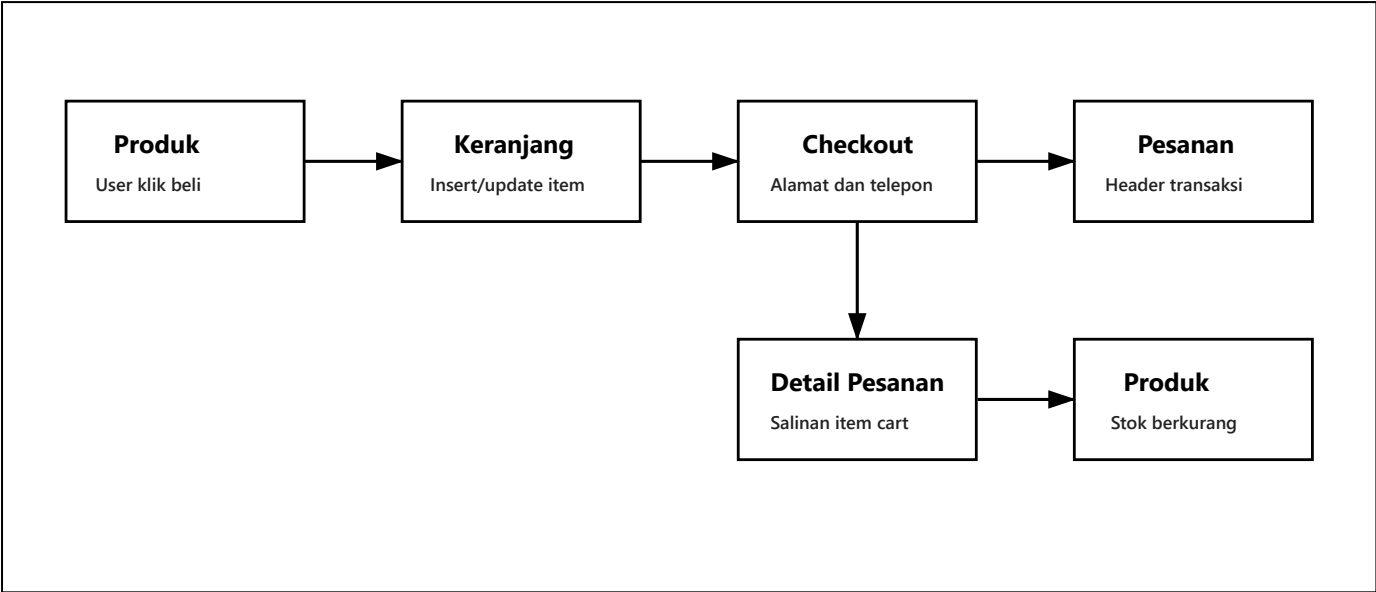
KUNCI BELAJAR

Jika bingung sebuah data berasal dari mana, cari nama fungsi atau query SQL-nya. Setelah revisi, input yang berasal dari user diprioritaskan memakai prepared statement helper seperti `dbQuery()` dan `dbStatement()`, lalu output HTML dicetak dengan `e()`.

5. Modul Publik

Modul	File	Alur Singkat
Beranda	pages/home.php	Mengambil slider dari <code>getSlider()</code> , produk unggulan dari <code>getAllProducts(6, true)</code> , lalu menampilkan card produk.
Produk	pages/produk.php	Membaca filter kategori, search, dan sort, lalu membangun query produk aktif.
Detail produk	pages/detail_produk.php	Mengambil produk berdasarkan <code>id</code> , menampilkan spesifikasi, galeri foto, harga, stok, dan produk terkait.
Auth	pages/login.php , pages/register.php	Register memakai <code>password_hash()</code> ; login memakai <code>password_verify()</code> dan mengisi <code>\$_SESSION</code> .
Profil	pages/profil.php	User login dapat memperbarui nama, nomor telepon, dan alamat.
Keranjang	pages/keranjang.php	Tambah produk, update jumlah, hapus item, validasi stok, hitung subtotal dan total plus pajak.
Checkout	pages/checkout.php	Membuat record <code>pesanan</code> , menyalin item keranjang ke <code>detail_pesanan</code> , mengurangi stok, lalu mengosongkan keranjang dalam transaksi database.
Pesanan	pages/pesanan.php	Menampilkan riwayat pesanan milik user berdasarkan <code>user_id</code> session.
Testimoni CRUD	pages/crud_demo.php	User login bisa membuat, membaca, mengubah, dan menghapus testimoni miliknya sendiri. Admin dapat mengelola lebih luas.
Kontak dan portfolio	pages/kontak.php , pages/portfolio.php	Halaman informasi, owner, dan kontak WhatsApp. Pesan kontak juga disimpan ke tabel <code>kontak</code> .

6. Gambar Alur Checkout



Gambar 2. Alur data saat user membeli produk, stok divalidasi, pesanan dibuat, lalu keranjang dikosongkan.

Pada `checkout.php` , sistem membuat nomor pesanan melalui `generateNomorPesanan()` , menyimpan total yang sudah ditambah pajak 10%, mengisi `detail_pesanan` dari isi keranjang, mengurangi stok produk, lalu menjalankan `DELETE FROM keranjang WHERE user_id = ?` . Semua langkah penting berjalan di dalam transaksi agar data tidak setengah tersimpan.

7. Modul Admin

Area admin berada di folder `admin/` . Semua akses admin melewati `admin/index.php` dan dilindungi fungsi `checkAdmin()` . Jika user belum login, diarahkan ke login. Jika role bukan `admin` , diarahkan ke halaman utama.

File Admin	Fungsi
<code>admin/pages/dashboard.php</code>	Menghitung total produk, pesanan, customer, kategori, daftar pesanan terbaru, dan produk stok rendah.
<code>admin/pages/produk.php</code>	CRUD produk. Mendukung upload gambar utama melalui helper <code>uploadGambar()</code> , validasi file, dan mencegah hapus produk yang sudah masuk pesanan.
<code>admin/pages/kategori.php</code>	CRUD kategori produk dan mencegah hapus kategori yang masih digunakan produk.
<code>admin/pages/pesanan.php</code>	Menampilkan semua pesanan, mengubah status, dan menampilkan detail item dengan JavaScript.
<code>admin/pages/testimoni.php</code>	Menampilkan testimoni dari user dan menyediakan aksi hapus.
<code>admin/pages/users.php</code>	Menampilkan daftar customer.

8. Database dan Relasi Tabel

File `database.sql` membuat database `perhiasan_db` dan tabel inti. Relasi utamanya memakai foreign key InnoDB.

Tabel	Isi Data	Relasi Penting
users	Akun admin dan customer, password hash, nama, kontak, role. Seed awal berisi 1 admin dan 3 customer.	Dipakai oleh <code>keranjang</code> , <code>pesanan</code> , dan <code>testimoni</code> .
kategori	Kategori perhiasan seperti cincin emas, kalung perak, gelang emas.	Direferensikan oleh <code>produk.kategori_id</code> .
produk	Nama produk, kategori, kadar, berat, harga, stok, gambar, status, featured.	Direferensikan oleh <code>keranjang</code> , <code>detail_pesanan</code> , dan <code>galeri_foto</code> .
galeri_foto	Foto tambahan produk.	Foreign key ke <code>produk</code> .
slider	Gambar hero/slider di beranda.	Tidak bergantung pada tabel lain.
keranjang	Produk yang akan dibeli user, jumlah, harga satuan.	Foreign key ke <code>users</code> dan <code>produk</code> .
pesanan	Header transaksi: nomor pesanan, total, status, alamat, telepon.	Foreign key ke <code>users</code> .
detail_pesanan	Item transaksi: produk, jumlah, harga, subtotal.	Foreign key ke <code>pesanan</code> dan <code>produk</code> .
kontak	Pesan dari halaman kontak.	Tidak bergantung pada tabel lain.
testimoni	Rating, judul ulasan, isi testimoni, status tampil, dan waktu edit.	Foreign key ke <code>users</code> .

STATUS REVISI DATABASE

`database.sql` sudah diselaraskan dengan kode: tabel `testimoni` dan `galeri_foto` tersedia, halaman kontak terhubung ke tabel `kontak` , dan akun awal dibuat tepat 4 akun yaitu `admin` , `siti_aminah` , `budi_santoso` , dan `rina_m` . Semua password seed disimpan sebagai hash.

9. Catatan Kualitas Kode Setelah Revisi

QUERY INPUT USER

Login, register, profil, search, keranjang, checkout, testimoni, kontak, dan beberapa aksi admin sudah memakai prepared statement helper seperti `dbQuery()` dan `dbStatement()`.

PASSWORD SEED

Fallback login `$password === 'password123'` sudah dihapus. Akun seed tetap bisa login dengan `password123`, tetapi nilai yang disimpan di database adalah hash bcrypt.

BUG ADD TO CART

JavaScript tidak lagi menebak login dari `logout.php`. Status login dikirim dari PHP melalui atribut `data-logged-in`, sehingga tombol beli bekerja konsisten.

NOMOR PESANAN

`generateNomorPesanan()` dan nama file upload sudah memakai `random_int()`. Untuk pengembangan lanjutan, nomor pesanan masih bisa dibuat retry otomatis jika bentrok dengan data lama.

10. SonarQube: Tujuan dan Cara Kerja

SonarQube digunakan untuk menganalisis kualitas source code. Scanner membaca kode di folder proyek, mengirim hasil analisis ke server SonarQube, lalu dashboard menampilkan issue seperti bug, vulnerability, security hotspot, code smell, duplications, dan coverage.

1. Server SonarQube berjalan, misalnya di `http://localhost:9000`.
2. Project dibuat di dashboard SonarQube dan token analisis dibuat.
3. File `sonar-project.properties` memberi identitas dan scope proyek.
4. `sonar-scanner` dijalankan dari root folder proyek.
5. Dashboard menampilkan Quality Gate dan daftar issue.

REFERENSI RESMI

Dokumentasi SonarSource menyebut SonarScanner CLI sebagai scanner analisis kode untuk pipeline CI/CD ketika tidak ada scanner khusus build system, dan konfigurasi project umumnya dibuat di root dengan nama `sonar-project.properties`.

11. Konfigurasi SonarQube Proyek Ini

File `sonar-project.properties` saat ini berisi:

```
sonar.projectKey=website-perhiasan-emas-perak-monica
sonar.projectName=Website Perhiasan Emas Perak Monica
sonar.projectVersion=1.0

sonar.sources=.
sonar.sourceEncoding=UTF-8

sonar.exclusions=uploads/**,tools/**,*.zip,**/*.sql,**/*.md,**/*.pdf,**/*.html
```

Properti	Makna
<code>sonar.projectKey</code>	ID unik proyek di SonarQube.
<code>sonar.projectName</code>	Nama yang tampil di dashboard.
<code>sonar.sources=.</code>	Scanner membaca source dari root proyek.
<code>sonar.sourceEncoding=UTF-8</code>	Encoding source code.
<code>sonar.exclusions</code>	Mengecualikan gambar upload, zip, SQL, dan markdown agar analisis fokus ke source code aplikasi.

Dokumen PDF, file HTML materi, dan script generator laporan dikecualikan agar SonarQube fokus pada source code website PHP/CSS/JavaScript, bukan artefak dokumentasi.

12. Langkah Pengujian SonarQube

Menjalankan SonarQube lokal

```
docker run -d --name sonarqube -p 9000:9000 sonarqube:community
```

Buka `http://localhost:9000` , login, buat project, lalu buat token analisis. Jika container sudah pernah dibuat, jalankan `docker start sonarqube` .

Menjalankan scanner dari root proyek

```
$env:SONAR_TOKEN="ISI_TOKEN_DARI_SONARQUBE"
sonar-scanner -Dsonar.host.url=http://localhost:9000
```

Token sebaiknya disimpan sebagai environment variable, bukan ditulis permanen di file project. Jika butuh debug, jalankan `sonar-scanner -X` .

Membuka hasil

```
http://127.0.0.1:9000/dashboard?id=website-perhiasan-emas-perak-monica
```

Setelah scan selesai, baca dashboard dari severity tertinggi: vulnerability, security hotspot, bug, lalu code smell. Perbaiki issue, jalankan ulang scanner, dan bandingkan hasilnya.

13. Ringkasan Hasil SonarQube dan Validasi Revisi

Berdasarkan file `HASIL_PENGUJIAN_SONARQUBE.md` , pengujian sebelumnya dilakukan pada 11 Juni 2026 memakai SonarQube Community Build 26.6.0.123539 dan SonarScanner CLI 8.0.1. Setelah revisi website, beberapa temuan yang sempat dicatat sebagai risiko sudah diperbaiki dan diverifikasi ulang dengan lint PHP, import SQL sementara, smoke test halaman, serta uji alur checkout.

Quality Gate OK	Bugs 25	Vulnerabilities 3	Code Smells 148
STATUS PASCA-REVISI Angka SonarQube di atas adalah baseline scan 11 Juni 2026. Perbaikan kode setelahnya sudah mengatasi bug database testimoni, fallback password default, deteksi login add-to-cart, validasi stok checkout, dan beberapa raw query input user. Untuk memperoleh angka SonarQube terbaru, scanner perlu dijalankan ulang pada commit final.			

Metrik	Hasil	Makna Belajar
Coverage	0.0%	Belum ada laporan unit test atau coverage PHPUnit yang diimpor ke SonarQube.

Metrik	Hasil	Makna Belajar
Duplicated Lines Density	0.0%	Tidak terdeteksi duplikasi baris signifikan pada hasil scan tersebut.
Reliability Rating	C	Ada bug yang perlu diprioritaskan.
Security Rating	B	Ada vulnerability atau hotspot yang perlu ditinjau.
Maintainability Rating	A	Walau ada code smell, maintainability masih baik menurut threshold saat scan.

14. Cara Membaca Temuan SonarQube

1. **Vulnerability:** potensi celah keamanan yang biasanya perlu diperbaiki lebih dulu. Contoh di laporan: resource CDN tanpa integrity attribute.
2. **Security Hotspot:** bukan otomatis bug, tetapi wajib ditinjau manual. Contoh: penggunaan random yang belum aman. Pada revisi terbaru, helper nomor pesanan dan upload sudah memakai `random_int()`.
3. **Bug:** pola kode yang bisa membuat aplikasi berperilaku salah. Contoh praktis pada revisi ini adalah tabel `testimoni` yang awalnya belum ada di skema dan deteksi login add-to-cart yang mengarah ke route logout lama.
4. **Code Smell:** masalah maintainability, misalnya CSS `!important`, selector duplikat, atau struktur kode yang sulit dirawat.
5. **Coverage:** persentase kode yang diuji oleh test suite. Nilai 0.0% berarti scanner belum menerima laporan coverage, bukan berarti SonarQube gagal berjalan.

15. Rekomendasi Pengujian Manual Website

Skenario	Langkah	Hasil yang Diharapkan
Register customer	Buka <code>?page=register</code> , isi data valid, submit.	User baru tersimpan dan diarahkan ke login.
Login customer	Login dengan <code>siti_aminah</code> , <code>budi_santoso</code> , atau <code>rina_m</code> .	Session terisi dan user masuk ke marketplace.
Tambah ke keranjang	Dari produk/detail, klik beli atau tambah.	Item masuk ke tabel <code>keranjang</code> dan jumlah tidak melebihi stok.
Checkout	Isi alamat dan telepon, klik pesan.	Record pesanan dan <code>detail_pesanan</code> terbentuk, stok produk berkurang, keranjang kosong.
Admin produk	Login admin, tambah/edit/hapus produk.	Data produk berubah dan gambar upload tersimpan.
Status pesanan	Di admin, ubah status pesanan.	Status tersimpan dan muncul pada halaman pesanan user.
Testimoni	User login membuat testimoni, edit, lalu hapus.	CRUD berjalan karena tabel <code>testimoni</code> sudah tersedia dan berelasi ke <code>users</code> .

16. Rekomendasi Lanjutan Setelah Revisi

1. Jalankan SonarQube ulang pada versi final agar angka bugs, vulnerabilities, dan code smells benar-benar terbaru.
2. Lengkapi prepared statement pada query statis yang nanti berubah menjadi menerima input user.
3. Tambahkan PHPUnit untuk fungsi murni seperti `formatRupiah()` , nomor pesanan, validasi helper, dan batas stok keranjang.
4. Tambahkan CSRF token untuk form login, register, keranjang, checkout, kontak, testimoni, dan admin.
5. Tambahkan retry nomor pesanan jika terjadi bentrok unik di database.
6. Tambahkan Subresource Integrity untuk CDN eksternal atau hosting asset secara lokal.

17. Referensi Resmi

- SonarScanner CLI - SonarQube Server Documentation
- Analysis Parameters - SonarQube Server Documentation
- PHP Test Coverage - SonarQube Server Documentation
- Test Coverage Parameters - SonarQube Server Documentation