



ClaimReview:

Panduan Media untuk
Kredibilitas Informasi



2025



CEK FAKTA



**ALIANSI
JURNALIS
INDEPENDEN**



Asosiasi
Media Siber
Indonesia



ClaimReview: Panduan Media untuk Kredibilitas Informasi

ClaimReview: Panduan Media untuk Kredibilitas Informasi

Penulis:

Catur Ratna Wulandari
Ronny Adolof Buol

Penata Isi & Desain Sampul:

Fahrul Rozi

Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindah sebagian atau seluruh isi laporan ke dalam bentuk apapun, secara elektronik maupun mekanis, termasuk fotokopi, merekam, atau dengan teknik perekaman lainnya, tanpa izin tertulis dari Penerbit. Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

Diterbitkan pertama kali oleh:



Jl. Kembang Raya No.6 Kwitang, Senen, Jakarta Pusat 10420
Telp 021-3151214, Fax 3151261
Email: sekretariat@ajindonesia.or.id
Web: www.aji.or.id

ClaimReview: Panduan Media untuk Kredibilitas Informasi
86 hlm, ; 21 x 29,7 cm
Cetakan I: Maret 2025

Apabila di dalam buku ini terdapat kesalahan cetak/produksi atau kesalahan informasi, mohon hubungi penerbit.

Daftar Isi

Pendahuluan	6
Tujuan Modul	6
Struktur Modul	7
Sesi 1: Dasar-Dasar Structured Data dan Schema.org	8
1.1: Pengenalan <i>Structured Data</i>	8
1.2: Apa itu Schema.org.....	12
1.3: Format Data JSON-LD (JavaScript Object Notation for Linked Data)	14
Sesi 2: Elemen-Elemen Penting dalam ClaimReview Markup	18
2.1: Properti Wajib ClaimReview Markup	18
2.2: Properti Rekomendasi ClaimReview Markup	29
2.3: Contoh <i>Code Snippet</i> ClaimReview Markup (Lengkap).....	35
Latihan	43
Kesimpulan	44
Pendahuluan	46
Tujuan Modul	46
Struktur Modul	47
Sesi 1: Langkah-Langkah Implementasi ClaimReview Markup	48
1.1: Persiapan: Identifikasi Klaim dan Produksi Konten Cek Fakta	48
Langkah 1: Identifikasi Klaim Spesifik	48
Langkah 2: Produksi Konten Cek Fakta Berkualitas.....	49
Sub-sesi 1.2: Implementasi Claim Review dengan Markup Tool - Google Fact Check Tool.....	50

Sub-sesi 1.3: Penyiapan Markup JSON-LD: Manual atau Otomatis?.....	56
Pendekatan 1: Penyiapan Markup Manual	56
Pendekatan 2: Penyiapan Markup Otomatis (dengan <i>Schema Markup Generator</i>).....	57
Sub-sesi 1.4: Implementasi Kode JSON-LD ke HTML di WordPress (Manual & Plugin)	58
Metode 1: Implementasi Manual dengan Insert HTML di Editor WordPress ...	58
Metode 2: Implementasi dengan Plugin WordPress (Contoh: Rank Math SEO Pro).....	62
Sub-sesi 1.5: Validasi Markup dengan Google Rich Results Test	67
Sub-sesi 1.6: Publikasi dan Monitoring	70
Sesi 2: Tools untuk Memudahkan Implementasi dan Validasi ClaimReview Markup	72
Sub-sesi 2.1: Google Rich Results Test: Uji Validasi Markup	73
Sub-sesi 2.2: Google Search Console: Monitoring Status <i>Rich Results</i>	75
Sub-sesi 2.3: Schema Markup Generator (Pihak Ketiga).....	78
Sub-sesi 2.4: Fact Check Markup Validator.....	80
Sesi 3: Praktik Implementasi Langsung (Sesi Praktik)	81
Penutup	86



CEK FAKTA

**ALIANSI
JURNALIS
INDEPENDEN**



Asosiasi
Media Siber
Indonesia



Memahami Struktur dan Elemen Teknis ClaimReview Markup

Pendahuluan

Modul ini akan membahas secara mendalam tentang struktur dan elemen teknis dari ClaimReview Markup. Modul ini dirancang untuk memberikan pemahaman yang kuat mengenai konsep-konsep dasar di balik *markup* ini, serta elemen-elemen teknis yang perlu Anda kuasai agar dapat mengimplementasikannya dengan benar dan efektif.

Pemahaman yang baik tentang aspek teknis ini akan sangat penting untuk memastikan konten cek fakta Anda tidak hanya berkualitas secara jurnalistik, tetapi juga optimal dalam hal visibilitas dan jangkauan di platform seperti Google Fact Check Explorer dan hasil pencarian Google.

Tujuan Modul

Setelah menyelesaikan modul ini, Anda diharapkan mampu:

1. **Memahami konsep dasar *structured data* dan pentingnya** dalam konteks web semantik dan mesin pencari.
2. **Memahami apa itu Schema.org** dan bagaimana *vocabulary* ini digunakan untuk mendefinisikan *structured data*.
3. **Memahami format data JSON-LD** dan alasan mengapa format ini direkomendasikan untuk implementasi ClaimReview markup.
4. **Menguasai struktur dan elemen-elemen penting dalam ClaimReview markup** berdasarkan *vocabulary* Schema.org, termasuk properti wajib dan properti yang sangat direkomendasikan.
5. **Mampu mengidentifikasi dan memetakan elemen-elemen konten cek fakta** ke dalam properti ClaimReview markup yang sesuai.

Durasi Pelajaran: 2-3 jam (dapat disesuaikan)

Struktur Modul

Modul ini terdiri dari dua sesi utama, yang masing-masing dibagi lagi menjadi beberapa sub-sesi untuk memastikan pemahaman yang mendalam:

- **Sesi 1: Dasar-Dasar Structured Data dan Schema.org**
 - Sub-sesi 1.1: Pengenalan *Structured Data*
 - Sub-sesi 1.2: Apa itu Schema.org
 - Sub-sesi 1.3: Format Data JSON-LD (JavaScript Object Notation for Linked Data)
- **Sesi 2: Elemen-Elemen Penting dalam ClaimReview Markup**
 - Sub-sesi 2.1: Properti Wajib ClaimReview Markup
 - Sub-sesi 2.2: Properti Rekomendasi ClaimReview Markup
 - Sub-sesi 2.3: Contoh *Code Snippet* ClaimReview Markup (Lengkap)



Sesi 1: Dasar-Dasar Structured Data dan Schema.org

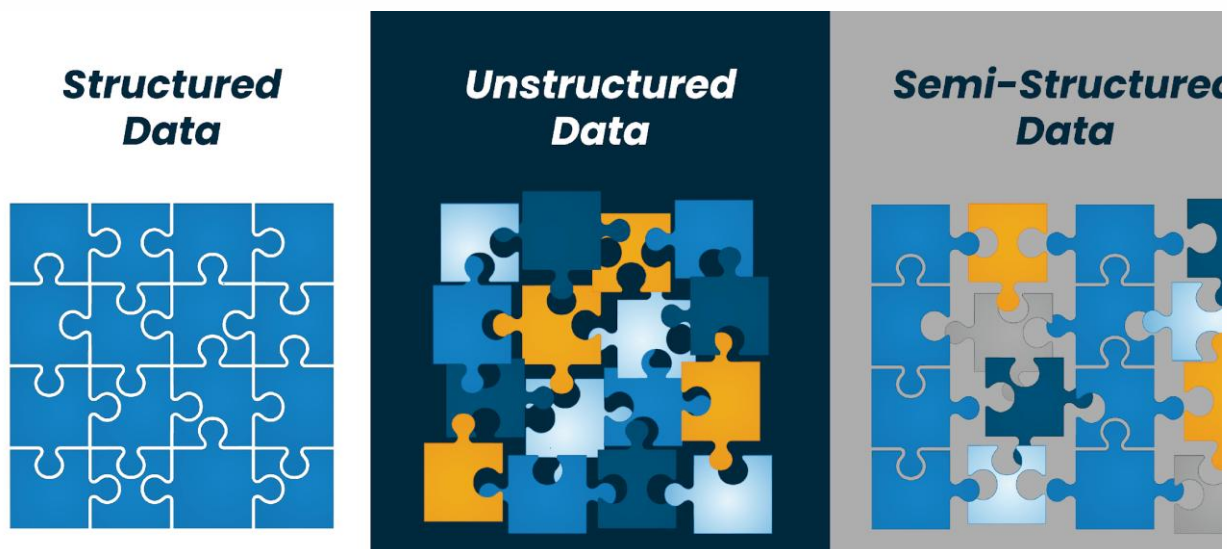
Sesi ini akan memperkenalkan konsep fundamental yang mendasari ClaimReview markup, yaitu *structured data* dan Schema.org. Tanpa memahami dasar-dasar ini, akan sulit untuk mengerti mengapa dan bagaimana ClaimReview markup bekerja.

1.1: Pengenalan *Structured Data*

Apa itu *Structured Data*?

Bayangkan internet sebagai perpustakaan raksasa yang berisi miliaran halaman web. Mesin pencari seperti Google bertugas untuk mengindeks dan mengorganisir perpustakaan ini, agar pengguna internet dapat menemukan informasi yang mereka cari dengan cepat dan relevan. Namun, bagaimana mesin pencari "membaca" dan memahami isi dari setiap halaman web?

Di sinilah peran *structured data* menjadi krusial. *Structured data* adalah cara untuk memberikan informasi tentang suatu halaman web dalam format yang terstruktur dan terorganisir, sehingga mudah dipahami oleh mesin pencari. Ini berbeda dengan *unstructured data*, seperti teks artikel berita biasa, gambar tanpa deskripsi, atau video tanpa *metadata*, yang lebih sulit diproses dan dipahami secara otomatis oleh mesin.



Gambar ilustrasi yang membandingkan contoh unstructured data seperti teks paragraf biasa dengan structured data seperti tabel atau format JSON-LD yang terstruktur. (Sumber: <https://saptatunas.com/>)

Mengapa *Structured Data* Penting?

Structured data memiliki beberapa manfaat penting, terutama dalam konteks optimasi mesin pencari (SEO) dan tampilan hasil pencarian yang lebih menarik (*rich results*):

- **Meningkatkan SEO (Search Engine Optimization):** Meskipun *structured data* bukan merupakan faktor peringkat langsung dalam algoritma pencarian Google, ia sangat membantu mesin pencari untuk lebih efektif dalam:
 - **Crawling (Perayapan):** Membantu mesin pencari lebih efisien dalam merayapi dan mengindeks halaman web Anda.
 - **Indexing (Pengindeksan):** Memungkinkan mesin pencari untuk lebih akurat dalam mengkategorikan dan menyimpan informasi dari halaman web Anda dalam indeks mereka.
 - **Understanding (Pemahaman):** Yang terpenting, *structured data* membantu mesin pencari untuk benar-benar *memahami* konten halaman web Anda. Dengan pemahaman yang lebih baik, mesin pencari dapat menilai relevansi halaman Anda dengan lebih tepat terhadap kueri pencarian pengguna.
- **Rich Results (Hasil Kaya):** Manfaat paling terlihat dari *structured data* adalah kemampuannya untuk menghasilkan *rich results*. *Rich results* adalah tampilan hasil pencarian yang lebih interaktif, menarik secara visual, dan kaya informasi dibandingkan dengan hasil pencarian teks biru standar. Contoh *rich results* termasuk:
 - **Fact Check Snippets:** Tampilan hasil cek fakta yang diperkaya dengan informasi *rating* dan sumber cek fakta, yang akan kita fokuskan dalam modul ini.
 - **Recipe Cards:** Tampilan resep makanan dengan gambar, *rating*, waktu memasak, dan informasi lainnya.
 - **Event Listings:** Daftar acara dengan tanggal, lokasi, dan harga tiket.
 - **Product Snippets:** Tampilan produk dengan harga, *rating*, dan ketersediaan.
 - Dan banyak lagi jenis *rich results* lainnya.

Wikipedia
<https://id.wikipedia.org/wiki/B...> · Translate this page

B. J. Habibie - Wikipedia bahasa Indonesia, ensiklopedia ...
 Bacharuddin Jusuf Habibie (25 Juni 1936 – 11 September 2019) adalah Presiden Indonesia ketiga yang menjabat sejak tahun 1998 sampai 1999, menggantikan Soeharto ...



Wikipedia
https://en.wikipedia.org/wiki/B._J._Habibie

B. J. Habibie
 Bacharuddin Jusuf Habibie was an Indonesian politician, engineer and scientist who served as the third president of Indonesia from 1998 to 1999.



People also ask :

- Berapa IQ BJ Habibie? ▼
- Mengapa BJ Habibie sangat penting? ▼
- Apakah Habibie pencipta pesawat? ▼
- Kenapa BJ Habibie dijuluki sebagai bapak teknologi Indonesia? ▼

Feedback

About

Bacharuddin Jusuf Habibie was an Indonesian politician, engineer and scientist who served as the third president of Indonesia from 1998 to 1999. Wikipedia

Born: June 25, 1936, Parepare

Died: September 11, 2019 (age 83 years), GATOT Subroto Army Hospital, Jakarta

Children: Ilham Akbar Habibie, Thareq Kemal Habibie

Grandchildren: Muhammad Pasha Nur Fauzan · See more

Spouse: Hasri Ainun Habibie (m. 1962–2010)

Education: RWTH Aachen University (1965) · See more

Presidential term: May 21, 1998 – October 20, 1999

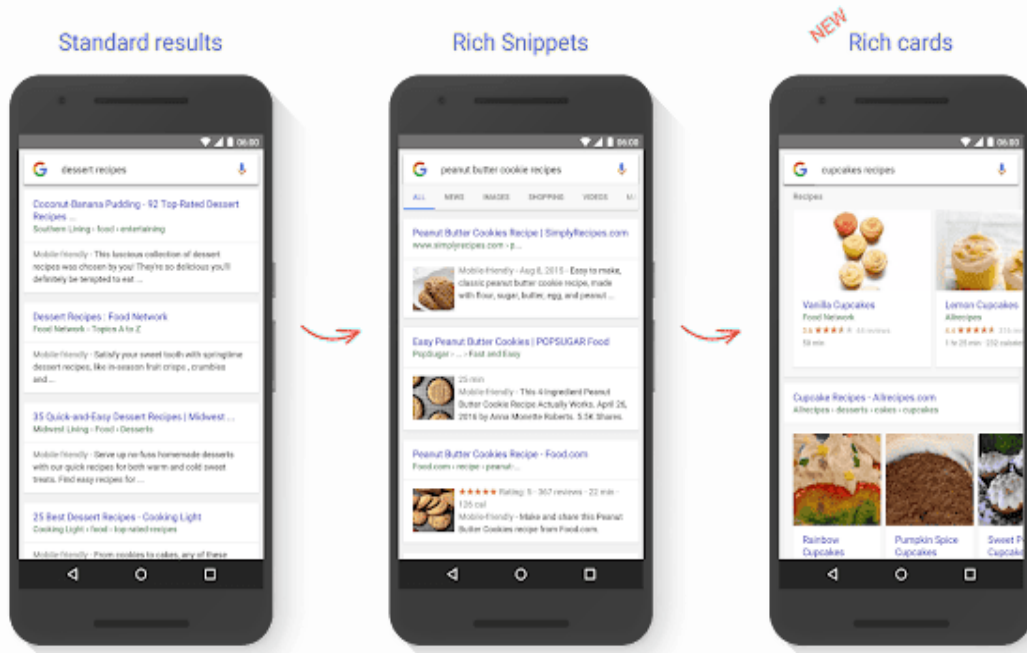
Feedback

People also search for

Hasri Ainun Habibie Ilham Akbar Habibie Suharto Abdurrahman Wahid

See more →

Contoh tampilan rich result data terstruktur hasil pencarian dengan keyword BJ Habibie di mesin telusur Google.



Perbedaan hasil pencarian terhadap resep makanan yang menggunakan data terstruktur dan tidak terstruktur.

Portkesmas
https://portkesmas.com › keliru-fo... · Translate this page ⋮
Keliru: Foto Mahasiswa UI yang Diklaim Membuktikan ...
1 day ago — Gambar itu memperlihatkan beberapa orang berfoto di sebuah ruangan, di depan logo PDIP yang menempel di dinding. Gambar kedua memperlihatkan ...

Tempo.co
https://www.tempo.co › cekfakta · Translate this page ⋮
Keliru: Video Pembagian Uang ke Mahasiswa Demonstran ...
Claim: Keliru: Video Pembagian Uang ke Mahasiswa Demonstran Indonesia Gelap
Claimed by: Tempo.co
Fact check by Tempo.co: keliru ←

Kompas.com
https://www.kompas.com › read · Translate this page ⋮
[HOAKS] Foto Mahasiswa Dibayar untuk Ikut Demo ...
3 days ago — Uang kertas yang tampak dalam foto tersebut adalah uang mainan untuk aksi unjuk rasa penolakan kenaikan PPN 12 persen pada Desember 2024.
Missing: ❌ Digerakkan PDIP

Portkesmas
https://portkesmas.com › cek-fakta... · Translate this page ⋮
Cek Fakta: Aksi Mahasiswa 'Indonesia Gelap' Ditunggangi ...
1 day ago — Keliru: Foto Mahasiswa UI yang Diklaim Membuktikan Demonstrasi 'Indonesia Gelap'
Digerakkan PDIP · Cek Fakta: Tidak Benar dalam Video ini ...

Tampilan konten cek fakta di Google Search, yang menggunakan data terstruktur (Tempo.co)

Dengan mengimplementasikan *structured data*, Anda memberikan "petunjuk" yang jelas kepada mesin pencari tentang konten halaman web Anda, sehingga halaman mesin pencari dapat menampilkannya dengan lebih baik dan menarik di hasil pencarian. Ini dapat meningkatkan visibilitas, *click-through rate* (CTR), dan akhirnya, *traffic* ke situs web Anda.

Format Implementasi *Structured Data*

Ada beberapa format yang dapat digunakan untuk mengimplementasikan *structured data*, antara lain:

- **JSON-LD (JavaScript Object Notation for Linked Data):** Format yang direkomendasikan oleh Google, dan yang akan kita gunakan untuk ClaimReview markup.
- **Microdata:** Format yang menggunakan atribut HTML untuk menambahkan *structured data* langsung di dalam konten HTML.

- **RDFa (Resource Description Framework in Attributes):** Format lain yang mirip dengan Microdata, menggunakan atribut HTML untuk *structured data*.

Meskipun ada format lain, **JSON-LD adalah format yang paling disarankan oleh Google** karena kemudahan implementasi dan struktur datanya yang bersih. Kita akan fokus menggunakan JSON-LD dalam modul ini.

1.2: Apa itu Schema.org

Schema.org: Kosakata untuk *Structured Data*

Setelah memahami konsep *structured data*, pertanyaan selanjutnya adalah: "Kosakata apa yang harus digunakan untuk membuat *structured data*?". Jawabannya adalah **Schema.org**.

Schema.org adalah *vocabulary* (kosakata) atau kumpulan skema terstruktur yang digunakan untuk mendeskripsikan berbagai jenis entitas dan properti yang ada di internet. Schema.org merupakan hasil kolaborasi antara mesin pencari besar seperti Google, Bing, Yahoo!, dan Yandex, dengan tujuan untuk menciptakan standar umum dalam *structured data*.

Tujuan Schema.org

Tujuan utama Schema.org adalah untuk menyediakan "bahasa" yang universal bagi *webmaster* untuk mendeskripsikan konten mereka. Dengan menggunakan *vocabulary* Schema.org, *webmaster* dapat memberikan informasi terstruktur tentang halaman web mereka dengan cara yang konsisten dan mudah dipahami oleh semua mesin pencari utama.

Struktur *Vocabulary* Schema.org

Schema.org terorganisir dalam struktur hirarkis yang terdiri dari **tipe** dan **properti**.

- **Tipe (Types):** Mendefinisikan kategori atau jenis entitas yang Anda deskripsikan. Contoh *tipe* dalam Schema.org antara lain:
 - Article (Artikel)
 - NewsArticle (Artikel Berita)

- BlogPosting (Postingan Blog)
 - VideoObject (Objek Video)
 - ImageObject (Objek Gambar)
 - Organization (Organisasi)
 - Person (Orang)
 - Event (Acara)
 - Recipe (Resep Makanan)
 - ClaimReview (Pemeriksaan Fakta) - yang menjadi fokus utama kita.
 - Dan banyak lagi tipe lainnya.
- **Properti (Properties):** Mendefinisikan atribut atau karakteristik dari suatu *tipe*. Setiap *tipe* memiliki daftar properti yang relevan. Contoh *properti* dalam Schema.org antara lain:
 - headline (Judul utama)
 - description (Deskripsi singkat)
 - image (Gambar)
 - author (Penulis)
 - datePublished (Tanggal Publikasi)
 - publisher (Penerbit)
 - url (URL halaman)
 - keywords (Kata kunci)
 - reviewRating (Rating Ulasan)
 - claimReviewed (Klaim yang Diperiksa) - yang spesifik untuk ClaimReview.
 - Dan banyak lagi properti lainnya.

Hierarki Tipe dan Properti

Schema.org memiliki struktur hirarkis. Tipe dan properti diorganisasikan dalam hierarki kelas, di mana tipe yang lebih spesifik mewarisi properti dari tipe yang lebih umum.

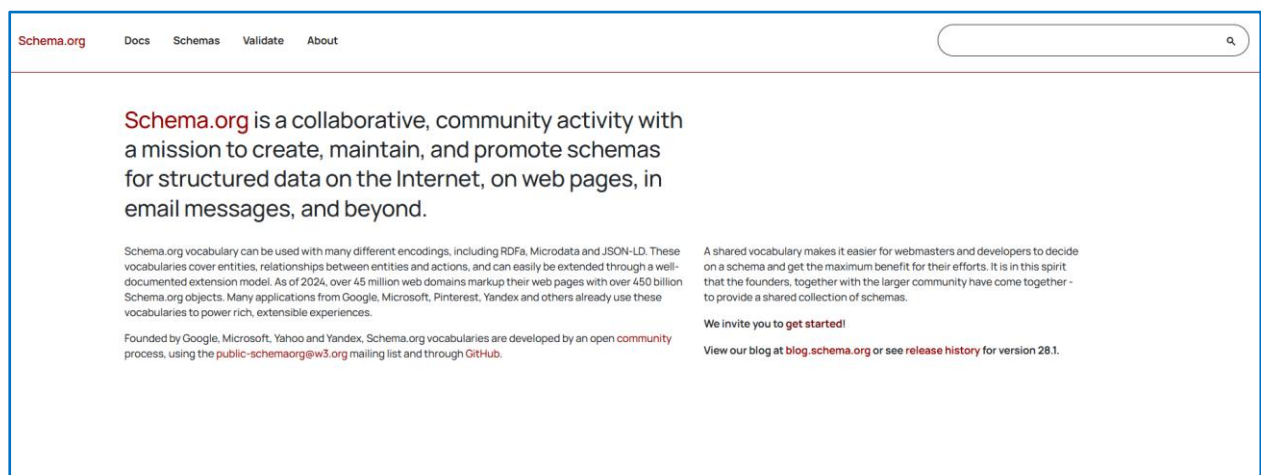
Contohnya: NewsArticle adalah tipe yang lebih spesifik dari Article. Oleh karena itu, NewsArticle akan mewarisi semua properti yang dimiliki oleh Article (seperti headline, author, datePublished), dan juga

memiliki properti tambahan yang spesifik untuk artikel berita (seperti `printEdition`).

Demikian pula, `ClaimReview` adalah tipe spesifik dalam Schema.org yang dirancang khusus untuk menandai konten pemeriksaan fakta.

Dokumentasi Schema.org

Referensi utama untuk semua tipe dan properti yang tersedia dalam Schema.org adalah situs web resmi mereka: <https://schema.org/>. Situs web ini menyediakan dokumentasi lengkap, contoh penggunaan, dan panduan untuk menggunakan *vocabulary* Schema.org. Anda sangat dianjurkan untuk menjelajahi situs web ini untuk memahami lebih dalam tentang berbagai tipe dan properti yang tersedia, dan bagaimana mereka dapat digunakan untuk mendeskripsikan berbagai jenis konten web.



(Screenshot halaman depan website Schema.org)

1.3: Format Data JSON-LD (JavaScript Object Notation for Linked Data)

JSON-LD: Format Implementasi yang Direkomendasikan

Kita telah membahas *structured data* dan *vocabulary* Schema.org. Sekarang, bagaimana cara kita *mengimplementasikan structured data* ini pada halaman web kita? Format yang direkomendasikan oleh Google untuk `ClaimReview` markup, dan secara umum untuk *structured data* Schema.org, adalah **JSON-LD (JavaScript Object Notation for Linked Data)**.

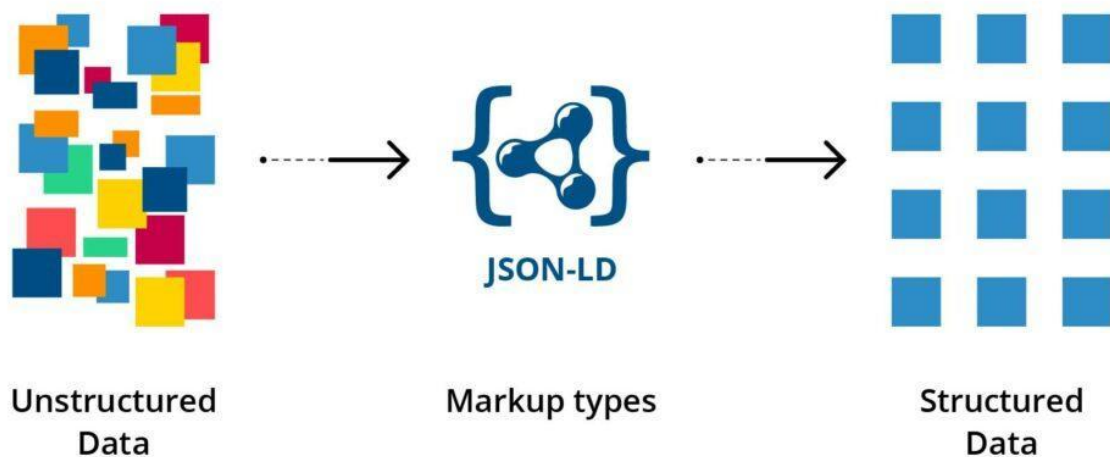
Apa itu JSON-LD?

JSON-LD adalah format untuk mengimplementasikan *structured data* yang berbasis pada format **JSON (JavaScript Object Notation)**. JSON adalah format data yang ringan, berbasis teks, dan mudah dibaca oleh manusia maupun mesin. JSON banyak digunakan dalam pengembangan web untuk pertukaran data antara *server* dan *browser*. "LD" dalam JSON-LD adalah singkatan dari "**Linked Data**", yang menekankan bahwa JSON-LD dirancang untuk membuat data yang saling terhubung di web, memfasilitasi pertukaran dan integrasi data antar aplikasi.

Mengapa Google Merekomendasikan JSON-LD?

Google merekomendasikan JSON-LD sebagai format utama untuk *structured data* karena beberapa alasan:

- **Kemudahan Implementasi:** JSON-LD relatif mudah diimplementasikan dan dikelola, bahkan bagi mereka yang tidak terlalu mendalam dalam *coding*. Kode JSON-LD biasanya ditempatkan dalam *tag* `<script>` terpisah dan disisipkan di bagian `<head>` atau `<body>` halaman HTML. Pendekatan ini memisahkan *markup structured data* dari konten HTML utama, sehingga tidak mengganggu struktur atau tampilan halaman.
- **Struktur yang Jelas dan Terbaca:** JSON-LD memiliki struktur yang sangat jelas, berbasis pada pasangan **kunci-nilai** (*key-value pairs*). Struktur ini menggunakan format teks yang mudah dibaca dan dipahami oleh manusia, sekaligus mudah diproses oleh mesin. Hal ini memudahkan dalam membuat, memodifikasi, dan *debug structured data*.
- **Fleksibilitas dan Ekspansi:** JSON-LD sangat fleksibel dan mudah diperluas. Anda dapat dengan mudah menambahkan properti baru, mengubah nilai properti, atau memodifikasi *structured data* tanpa harus mengubah struktur HTML halaman web. Ini memberikan fleksibilitas dalam mengelola dan memperbarui *structured data* seiring waktu.
- **Dukungan yang Kuat:** JSON-LD mendapatkan dukungan yang kuat dari Google dan berbagai *tools* validasi *structured data* yang mereka sediakan, seperti Google Rich Results Test dan Google Search Console. Dukungan ini memastikan bahwa *structured data* dalam format JSON-LD akan dikenali dan diproses dengan baik oleh Google.



Sumber: <https://wetech.co.za/>

Struktur Dasar JSON-LD

JSON-LD memiliki struktur dasar yang mudah dipahami. Berikut adalah elemen-elemen dasar dalam struktur JSON-LD:

- **{ dan }** (**Kurung Kurawal**): Setiap objek JSON-LD dimulai dengan kurung kurawal { dan diakhiri dengan kurung kurawal }. Objek ini merupakan kontainer utama untuk *structured data*.
- **"** (**Tanda Kutip Ganda**): *Keys* (kunci) dan *string values* (nilai teks) dalam JSON-LD *harus* diapit oleh tanda kutip ganda ". Ini adalah aturan *syntax* JSON yang penting untuk diingat.
- **:** (**Titik Dua**): Titik dua : digunakan untuk memisahkan antara *key* dan *value* dalam sebuah pasangan *key-value*. *Key* selalu berada di sebelah kiri titik dua, dan *value* berada di sebelah kanan.
- **,** (**Koma**): Koma , digunakan untuk memisahkan antara pasangan *key-value* yang satu dengan yang lain dalam sebuah objek JSON, atau untuk memisahkan elemen-elemen dalam sebuah *array* (daftar).

Contoh Struktur JSON-LD Sederhana:

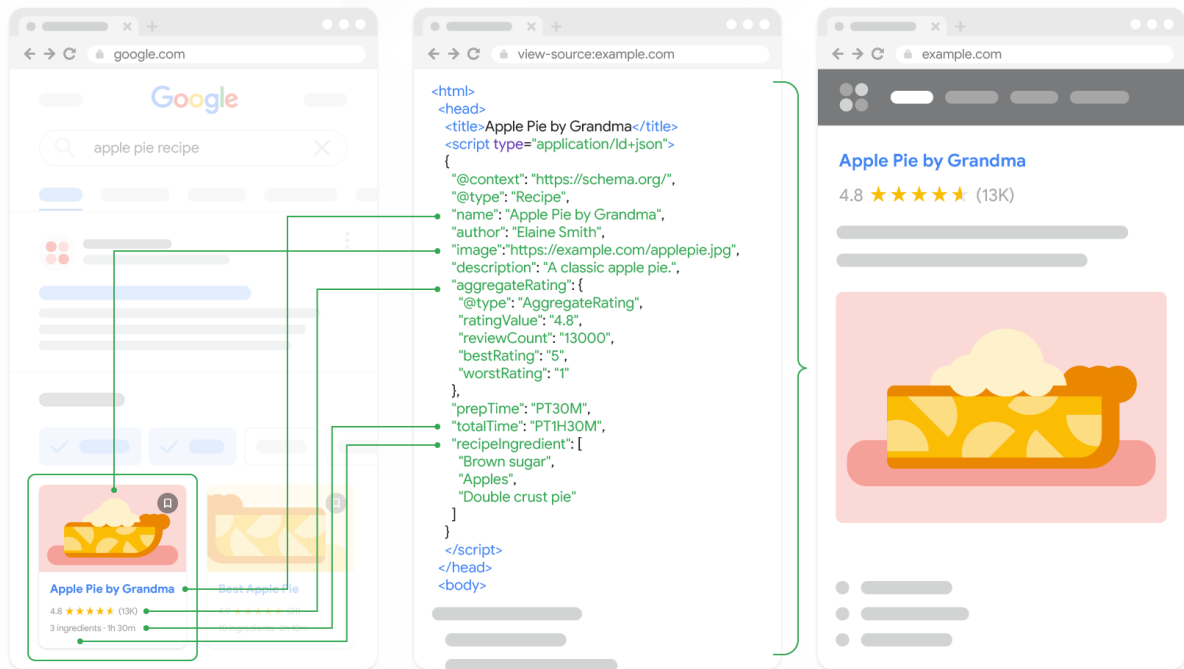
Berikut adalah contoh sederhana struktur JSON-LD untuk mendeskripsikan informasi tentang seseorang (Person):

JSON

```
{
  "@context": "https://schema.org",
  "@type": "Person",
  "name": "Ronny Buol",
  "jobTitle": "Jurnalis"
}
```

Mari kita bedah contoh di atas baris per baris:

- `{`: Memulai objek JSON-LD.
- `"@context": "https://schema.org",`: Baris ini mendefinisikan **@context**. `@context` merupakan kunci khusus dalam JSON-LD yang *wajib* ada. Nilainya adalah `"https://schema.org"`, yang memberitahu mesin pencari bahwa *vocabulary* yang digunakan dalam *markup* ini adalah Schema.org. Dengan kata lain, kita "berbicara" menggunakan "bahasa" Schema.org.
- `"@type": "Person",`: Baris ini mendefinisikan **@type**. `@type` juga merupakan kunci khusus yang *wajib* ada. Nilainya adalah `"Person"`, yang menunjukkan bahwa kita sedang mendeskripsikan entitas bertipe `"Person"` dari *vocabulary* Schema.org.
- `"name": "Ronny Buol",`: Baris ini mendefinisikan **properti name** dari tipe `"Person"`. `"name"` adalah *key*, dan `"Ronny Buol"` adalah *value*. Ini berarti nama orang tersebut adalah `"John Doe"`.
- `"jobTitle": "Jurnalis":` Baris ini mendefinisikan **properti jobTitle** dari tipe `"Person"`. `"jobTitle"` adalah *key*, dan `"Jurnalis"` adalah *value*. Ini berarti pekerjaan orang tersebut adalah `"Jurnalis"`.
- `}`: Menutup objek JSON-LD.



Contoh kode JSON-LD sederhana untuk resep makanan dan hasil tampilannya di mesin pencari. (Sumber:)

Ini adalah contoh yang sangat sederhana. Dalam ClaimReview markup, struktur JSON-LD akan lebih kompleks karena kita akan menggunakan tipe ClaimReview dan properti-properti yang spesifik untuk pemeriksaan fakta, namun prinsip dasarnya tetap sama: menggunakan struktur JSON-LD untuk mendeskripsikan konten kita dengan *vocabulary* Schema.org.

Sesi 2: Elemen-Elemen Penting dalam ClaimReview Markup

Setelah memahami dasar-dasar *structured data*, Schema.org, dan format JSON-LD, sekarang kita akan fokus pada elemen-elemen spesifik yang membentuk **ClaimReview markup**. Sesi ini akan membahas properti-properti penting yang wajib dan direkomendasikan dalam ClaimReview markup, berdasarkan *vocabulary* Schema.org.

2.1: Properti Wajib ClaimReview Markup

Properti Wajib: Pondasi ClaimReview Markup

ClaimReview markup, seperti halnya tipe *structured data* lainnya di Schema.org, memiliki sejumlah **properti** yang didefinisikan. Beberapa properti bersifat **wajib (required)**. Ini berarti properti-properti ini *harus* ada dalam *markup* Anda agar dianggap valid dan berfungsi dengan benar oleh mesin pencari, terutama Google. Jika properti wajib tidak ada, *markup* Anda mungkin tidak akan terdeteksi atau tidak akan menghasilkan *rich results* yang diharapkan.

Berikut adalah daftar **properti wajib** dalam ClaimReview markup, beserta penjelasan detail dan contoh penggunaannya:

1. `claimReviewed` (Teks): Klaim yang Diperiksa

- **Definisi:** Properti `claimReviewed` digunakan untuk menentukan **klaim atau pernyataan spesifik** yang menjadi fokus pemeriksaan fakta dalam artikel Anda. Ini adalah inti dari ClaimReview markup, karena properti ini secara langsung mengidentifikasi apa yang sebenarnya Anda periksa faktanya.
- **Tipe Data:** Properti ini harus berupa **teks (Text)**. Artinya, Anda harus memasukkan klaim tersebut dalam format teks langsung.
- **Contoh Penggunaan:**

Misalnya, jika artikel cek fakta Anda memeriksa klaim bahwa "Pemerintah mengumumkan kenaikan harga BBM sebesar 20%", maka nilai properti `claimReviewed` adalah:

JSON

```
"claimReviewed": "Pemerintah mengumumkan  
kenaikan harga BBM sebesar 20%"
```

Penting untuk diingat: Nilai `claimReviewed` harus berupa **teks klaim secara langsung**, bukan deskripsi atau interpretasi dari klaim tersebut. Hindari penggunaan kalimat seperti "Artikel ini memeriksa klaim tentang..." atau "Dalam artikel ini, kami melakukan cek fakta terhadap klaim bahwa...". Cukup tuliskan klaimnya saja.

- **Penting!** Properti `claimReviewed` adalah sangat krusial karena **Google akan menampilkan teks klaim ini secara langsung** di Google Fact Check Explorer dan potensi *rich results* di hasil pencarian. Oleh karena itu, pastikan teks klaim yang Anda masukkan **jelas, ringkas, dan akurat**, serta benar-benar merepresentasikan klaim yang Anda periksa faktanya dalam artikel.

2. **author (Organization atau Person): Organisasi atau Orang yang Melakukan Cek Fakta**

- **Definisi:** Properti `author` digunakan untuk mengidentifikasi **entitas yang melakukan pemeriksaan fakta**. Entitas ini bisa berupa **Organisasi (Organization)**, seperti media cek fakta atau lembaga pemeriksa fakta, atau **Person (Orang)**, seperti jurnalis atau pemeriksa fakta individu. Untuk media atau organisasi berita, umumnya Anda akan menggunakan tipe `Organization`.
- **Tipe Data:** Properti `author` harus berupa salah satu dari dua tipe `Schema.org: Organization` atau `Person`. Ini berarti nilai dari properti `author` harus berupa **objek JSON** yang mendeskripsikan organisasi atau orang tersebut.
- **Contoh Penggunaan (Tipe Organization):**

JSON

```
"author": {
  "@type": "Organization",
  "name": "Zonautara.com",
  "url": "https://zonautara.com/"
}
```

Dalam contoh di atas:

- `"@type": "Organization"`: Menunjukkan bahwa kita mendeskripsikan sebuah organisasi.
- `"name": "Zonautara.com"`: Nama organisasi pemeriksa fakta adalah "Zonautara.com".

- "url": "https://zonautara.com/": URL situs web organisasi pemeriksa fakta.
- **Contoh Penggunaan (Tipe Person):**

JSON

```
"author": {
  "@type": "Person",
  "name": "Ronny Buol",
  "url":
  "https://zonautara.com/author/ronnybuol"
}
```

Dalam contoh di atas:

- "@type": "Person": Menunjukkan bahwa kita mendeskripsikan seorang individu.
- "name": "Ronny Buol": Nama jurnalis atau pemeriksa fakta.
- "url": "https://zonautara.com/author/ronnybuol": URL halaman profil jurnalis di situs web media (opsional, tapi sangat dianjurkan).
- **Penting!** Properti `author` sangat penting karena berfungsi untuk **mengidentifikasi sumber kredibel** yang bertanggung jawab atas pemeriksaan fakta. Informasi seperti `name` (nama organisasi atau orang) dan `url` (situs web atau profil) sangat krusial untuk **membangun kredibilitas dan kepercayaan** di mata audiens dan mesin pencari.

3. `datePublished` (Tanggal): Tanggal Publikasi Artikel Cek Fakta

- **Definisi:** Properti `datePublished` menentukan **tanggal artikel cek fakta dipublikasikan**. Tanggal ini digunakan untuk memberikan konteks waktu pada pemeriksaan fakta.
- **Tipe Data:** Properti `datePublished` harus berupa **tanggal (Date)** dalam format **ISO 8601**. Format ISO 8601 yang paling umum digunakan adalah YYYY-MM-DD (tahun-bulan-tanggal).
- **Contoh Penggunaan:**

Jika artikel cek fakta Anda dipublikasikan pada tanggal 1 Februari 2025, maka nilai properti `datePublished` adalah:

JSON

```
"datePublished": "2025-02-01"
```

- **Penting!** Properti `datePublished` memberikan **informasi *timestamp*** yang penting untuk konteks dan *freshness* konten cek fakta. Tanggal publikasi membantu audiens dan mesin pencari memahami kapan pemeriksaan fakta tersebut dilakukan, dan apakah informasi tersebut masih relevan dan *up-to-date*.

4. `itemReviewed (Claim): Detail Klaim yang Diperiksa`

- **Definisi:** Properti `itemReviewed` digunakan untuk memberikan **detail lebih lanjut tentang klaim** yang diperiksa faktanya. Properti ini *harus* berupa **objek bertipe `Claim`** dari Schema.org. Objek `Claim` ini memiliki properti-properti tambahan yang memungkinkan Anda memberikan konteks yang lebih kaya tentang klaim tersebut.
- **Tipe Data:** Properti `itemReviewed` harus berupa **objek bertipe `Claim`**.
- **Struktur Objek `Claim` (Minimal):** Objek `Claim` minimal harus memiliki properti berikut:
 - `"@type": "Claim"`: Menandakan bahwa objek ini adalah objek `Claim`.
 - **`appearance (PropertyValue): Tempat Klaim Muncul Secara Online`**
 - **Definisi:** Properti `appearance` digunakan untuk menentukan **tempat atau lokasi di mana klaim tersebut muncul secara *online***. Ini bisa berupa URL dari artikel berita, postingan media sosial, video, atau sumber *online* lainnya di mana klaim tersebut disebarkan.
 - **Tipe Data:** `appearance` harus berupa ***array* objek `PropertyValue`**. Mengapa *array*? Karena sebuah klaim bisa muncul di beberapa tempat yang berbeda

secara *online*. Setiap elemen dalam *array* adalah objek `PropertyValue`, yang mendeskripsikan satu lokasi kemunculan klaim.

- **Struktur Objek `PropertyValue` (untuk `appearance`):** Setiap objek `PropertyValue` dalam *array* `appearance` minimal harus memiliki properti berikut:
 - `"@type"`: `"PropertyValue"`: Menandakan bahwa objek ini adalah objek `PropertyValue`.
 - `"url"` (URL): **URL halaman web** tempat klaim muncul.
- **Contoh Penggunaan Objek `itemReviewed`:**

JSON

```
"itemReviewed": {
  "@type": "Claim",
  "appearance": [
    {
      "@type": "PropertyValue",
      "url":
"https://www.situsberita-
sumberklaim.com/artikel-klaim"
    }
  ],
  "firstAppearance": [ // Properti
"firstAppearance" (tempat pertama
kali muncul) bersifat opsional, tapi
sangat direkomendasikan
    {
      "@type": "PropertyValue",
      "url": "https://www.akun-
media-sosial-
penyebarklaim.com/postingan-pertama-
klaim"
    }
  ]
}
```

```
}
```

Dalam contoh di atas:

- "itemReviewed": { ... }:
Mendefinisikan objek `itemReviewed`.
- "@type": "Claim": Menunjukkan bahwa objek ini adalah bertipe `Claim`.
- "appearance": [...]:
Mendefinisikan *array* `appearance`, yang berisi daftar tempat klaim muncul. Dalam contoh ini, hanya ada satu tempat kemunculan klaim, yaitu:
 - { "@type": "PropertyValue",
"url":
"https://www.situsberita-sumberklaim.com/artikel-klaim" }: Objek `PropertyValue` yang menyatakan bahwa klaim muncul di URL "<https://www.situsberita-sumberklaim.com/artikel-klaim>".
- "firstAppearance": [...]: Properti `firstAppearance` (tempat klaim pertama kali muncul) bersifat **opsional tapi sangat direkomendasikan** jika Anda mengetahui sumber pertama klaim tersebut. Strukturnya sama dengan `appearance`, yaitu *array* objek `PropertyValue`. Dalam contoh, diasumsikan klaim pertama kali muncul di media sosial:
 - { "@type": "PropertyValue",
"url": "https://www.akun-media-sosial-penyebarklaim.com/postingan-pertama-klaim" }: Objek `PropertyValue` yang menyatakan bahwa klaim pertama kali muncul di

URL "<https://www.akun-media-sosial-penyebarklaim.com/postingan-pertama-klaim>".

- **Penting!** Properti `itemReviewed` memberikan **konteks penting tentang klaim yang diperiksa**. Menyertakan informasi tempat klaim muncul (`appearance`) dan tempat klaim pertama kali muncul (`firstAppearance`) membantu audiens dan mesin pencari untuk memahami **asal-usul dan penyebaran klaim**, serta memberikan konteks yang lebih lengkap terhadap pemeriksaan fakta yang Anda lakukan.

5. `reviewRating` (Rating): Hasil Penilaian atau Rating Cek Fakta

- **Definisi:** Properti `reviewRating` digunakan untuk menyatakan **hasil penilaian atau *rating* dari pemeriksaan fakta**. Ini adalah kesimpulan akhir dari analisis Anda terhadap klaim yang diperiksa. Properti ini harus berupa **objek bertipe `Rating`** dari Schema.org.
- **Tipe Data:** Properti `reviewRating` harus berupa **objek bertipe `Rating`**.
- **Struktur Objek `Rating` (Minimal):** Objek `Rating` minimal harus memiliki properti berikut:
 - `"@type": "Rating"`: Menandakan bahwa objek ini adalah objek `Rating`.
 - **`ratingValue` (Teks atau Numerik): Nilai atau Label `Rating`**
 - **Definisi:** Properti `ratingValue` adalah **nilai atau label yang merepresentasikan *rating*** Anda terhadap klaim yang diperiksa. Ini bisa berupa:
 - **Nilai Numerik:** Jika sistem *rating* Anda menggunakan skala numerik (misalnya, skala 1-5, di mana 1 berarti "Sangat Tidak Benar" dan 5 berarti "Sepenuhnya Benar"), maka `ratingValue` akan berupa angka numerik. Jika Anda menggunakan skala numerik, sangat direkomendasikan untuk juga menyertakan properti `bestRating` (nilai terbaik dalam skala) dan `worstRating` (nilai terburuk

dalam skala) untuk memperjelas rentang skala *rating* Anda.

- **Label Teks:** Jika sistem *rating* Anda menggunakan label teks (misalnya, "Benar", "Salah", "Menyesatkan", "Tidak Berbasis Fakta", dll.), maka `ratingValue` akan berupa teks label tersebut. Pastikan label teks yang Anda gunakan **konsisten** dengan sistem *rating* yang Anda gunakan di seluruh website Anda, dan **mudah dipahami** oleh audiens.
- **ratingExplanation (Teks): Penjelasan Singkat Hasil Rating**
 - **Definisi:** Properti `ratingExplanation` adalah **penjelasan singkat dan ringkas** tentang hasil *rating* atau kesimpulan pemeriksaan fakta Anda. Ini adalah **ringkasan dari kesimpulan utama** artikel cek fakta Anda.
 - **Penting untuk diingat:** `ratingExplanation` *bukanlah* artikel cek fakta lengkap. Ini hanyalah ringkasan singkat. Detail lengkap dari analisis dan bukti pendukung harus tetap ada di dalam artikel cek fakta Anda. `ratingExplanation` berfungsi untuk memberikan *preview* kesimpulan kepada audiens langsung di hasil pencarian.
- **Contoh Penggunaan Objek `reviewRating` (Menggunakan Skala Numerik):**

JSON

```
"reviewRating": {
  "@type": "Rating",
  "ratingValue": 2, // Skala 1-5, contoh:
  1=Sangat Salah, 5=Sepenuhnya Benar, 2 mungkin
  berarti "Sebagian Salah"
  "bestRating": 5, // Nilai terbaik dalam
  skala adalah 5
  "worstRating": 1, // Nilai terburuk dalam
  skala adalah 1
```

```
"ratingExplanation": "Klaim tersebut
sebagian benar, namun ada konteks penting
yang diabaikan yang jika tidak disertakan,
bisa menyesatkan."
}
```

Dalam contoh di atas, kita menggunakan skala numerik 1-5 untuk *rating*. `ratingValue`: 2 mungkin merepresentasikan label *rating* "Sebagian Salah". `bestRating`: 5 dan `worstRating`: 1 memberikan konteks tentang skala yang digunakan. `ratingExplanation` memberikan ringkasan kesimpulan.

- o **Contoh Penggunaan Objek `reviewRating` (Menggunakan Label Teks):**

JSON

```
"reviewRating": {
  "@type": "Rating",
  "ratingValue": "Salah", // Menggunakan
label teks "Salah" sebagai rating
  "ratingExplanation": "Klaim tersebut
sepenuhnya tidak benar dan tidak didukung
oleh bukti yang kredibel."
}
```

Dalam contoh ini, kita menggunakan **label teks "Salah"** sebagai `ratingValue`. Ini lebih sederhana dan lebih mudah dipahami oleh audiens umum. `ratingExplanation` tetap memberikan ringkasan kesimpulan.

- o **Penting!** Properti `reviewRating` adalah **hasil akhir** dari proses pemeriksaan fakta Anda. Informasi dalam properti ini (terutama `ratingValue` dan `ratingExplanation`) adalah yang **secara langsung ditampilkan di *rich results*** Google dan Google Fact Check Explorer. Oleh karena itu, pastikan *rating* dan ringkasan kesimpulan Anda **akurat, jelas, dan ringkas**.

ratingExplanation sangat penting untuk memberikan konteks singkat tentang *rating* yang Anda berikan.

Contoh Code Snippet ClaimReview Markup dengan Properti Wajib

Berikut adalah contoh *code snippet* JSON-LD ClaimReview markup yang hanya mencakup **properti-properti wajib** yang telah kita bahas:

JSON

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ClaimReview",
  "claimReviewed": "Pemerintah mengumumkan kenaikan
harga BBM sebesar 20%",
  "author": {
    "@type": "Organization",
    "name": "Media Cek Fakta ABC",
    "url": "https://www.mediacekfaktaabc.com/"
  },
  "datePublished": "2024-07-27",
  "itemReviewed": {
    "@type": "Claim",
    "appearance": [
      {
        "@type": "PropertyValue",
        "url": "https://www.situsberita-
sumberklaim.com/artikel-klaim"
      }
    ]
  },
  "reviewRating": {
    "@type": "Rating",
    "ratingValue": "Sebagian Salah",
    "ratingExplanation": "Klaim tersebut sebagian
benar, namun ada konteks penting yang diabaikan."
  }
}
```

</script>

Code snippet di atas adalah contoh dasar ClaimReview markup yang valid karena telah menyertakan semua properti wajib. Namun, untuk meningkatkan kualitas *rich results* dan visibilitas konten Anda, sangat disarankan untuk juga menggunakan properti-properti rekomendasi, yang akan kita bahas selanjutnya.

2.2: Properti Rekomendasi ClaimReview Markup

Properti Rekomendasi: Meningkatkan Kualitas dan Visibilitas

Setelah memahami properti-properti wajib dalam ClaimReview markup, sekarang kita akan membahas **properti-properti rekomendasi (recommended properties)**. Meskipun properti-properti ini **tidak wajib** secara teknis agar *markup* Anda valid, Google sangat **menganjurkan** untuk menggunakannya. Mengapa? Karena properti rekomendasi ini dapat secara signifikan **meningkatkan kualitas *rich results***, memberikan informasi yang lebih lengkap dan menarik kepada pengguna, serta berpotensi **meningkatkan visibilitas** konten cek fakta Anda di Google Fact Check Explorer dan platform Google lainnya.

Berikut adalah daftar **properti rekomendasi penting** dalam ClaimReview markup, beserta penjelasan detail dan contoh penggunaannya:

1. **url (URL, Sangat Direkomendasikan dan Praktisnya Wajib untuk ClaimReview Markup): URL Artikel Cek Fakta**
 - **Definisi:** Properti `url` adalah **URL halaman artikel cek fakta itu sendiri**. Ini adalah tautan langsung ke artikel lengkap di situs web Anda yang berisi pemeriksaan fakta.
 - **Tipe Data:** Properti `url` harus berupa **URL** yang valid.

- **Contoh Penggunaan:**

JSON

```
"url": "https://www.mediacekfaktaabc.com/cek-fakta/klaim-kenaikan-bbm-20-persen"
```

- **Penting! (Sangat Direkomendasikan dan Praktisnya Wajib):**

Meskipun dalam *spesifikasi* Schema.org ClaimReview, properti `url` mungkin tidak secara eksplisit disebut sebagai properti *wajib* dalam arti teknis yang ketat, dalam **praktik implementasi dan untuk tujuan mendapatkan manfaat *rich results***, properti `url` ini **sangat penting dan praktisnya dapat dianggap *wajib hukumnya***. Mengapa?

- **Menghubungkan ke Artikel Lengkap:** Properti `url` adalah **jembatan** yang menghubungkan *rich results* atau tampilan di Google Fact Check Explorer dengan **artikel cek fakta lengkap** di situs web Anda. Tanpa properti `url` ini, mesin pencari dan pengguna tidak akan bisa dengan mudah mengakses artikel lengkap Anda untuk mendapatkan detail dan bukti pendukung dari pemeriksaan fakta.
- **Persyaratan *Rich Results* dan Google Fact Check Explorer:** Google kemungkinan besar **memerlukan properti `url`** agar konten cek fakta Anda memenuhi syarat untuk ditampilkan sebagai *rich results* atau di Google Fact Check Explorer. Tanpa `url`, Google mungkin tidak dapat mengindeks dan menampilkan konten Anda dengan benar dalam konteks pemeriksaan fakta.
- **SEO dan *Canonical URL*:** Properti `url` membantu mesin pencari memahami ***canonical URL*** dari artikel cek fakta Anda. Ini penting untuk menghindari masalah konten duplikat dan memastikan bahwa *traffic* dan *link equity* diarahkan ke URL yang benar.

Oleh karena itu, **sangat penting untuk selalu menyertakan properti `url`** dalam ClaimReview markup Anda. Pastikan URL

yang Anda masukkan adalah URL yang benar dan berfungsi menuju artikel cek fakta lengkap.

2. **image** (URL Gambar): **Gambar *Thumbnail* Artikel Cek Fakta**

- **Definisi:** Properti `image` digunakan untuk menentukan **URL gambar *thumbnail* atau gambar utama** yang relevan dengan artikel cek fakta Anda. Gambar ini akan digunakan untuk mempercantik tampilan *rich results* dan di platform Google lainnya.
- **Tipe Data:** Properti `image` harus berupa **URL gambar**. Anda juga dapat menggunakan tipe `ImageObject` dari `Schema.org` untuk memberikan informasi gambar yang lebih detail (seperti *caption*, resolusi, dll.), tetapi untuk tujuan *thumbnail*, URL gambar sudah cukup.
- **Contoh Penggunaan:**

JSON

```
"image":  
"https://www.mediacekfaktaabc.com/gambar-  
thumbnail-cek-fakta-bbm.jpg"
```

- **Pentingnya:**
 - **Meningkatkan Daya Tarik Visual:** Gambar *thumbnail* membuat *rich results* **lebih menarik secara visual** dan menonjol di halaman hasil pencarian. Hasil pencarian dengan gambar cenderung lebih menarik perhatian pengguna dibandingkan hasil pencarian teks biasa.
 - **Meningkatkan *Click-Through Rate* (CTR):** Tampilan yang lebih menarik secara visual berpotensi **meningkatkan *click-through rate* (CTR)** dari hasil pencarian. Pengguna lebih mungkin untuk mengklik hasil pencarian yang memiliki gambar *thumbnail* yang relevan dan menarik. CTR yang lebih tinggi dapat berkontribusi pada peningkatan *traffic* ke situs web Anda.
 - **Rekomendasi Google:** Google secara eksplisit **merekomendasikan penggunaan gambar** dalam *structured*

data, termasuk ClaimReview markup, untuk meningkatkan kualitas *rich results*.

Saat memilih gambar untuk properti `image`, pertimbangkan hal-hal berikut:

- **Relevansi:** Pilih gambar yang **relevan dengan topik artikel cek fakta**. Gambar sebaiknya merepresentasikan klaim yang diperiksa atau hasil pemeriksaan fakta secara visual.
- **Kualitas Tinggi:** Gunakan gambar dengan **resolusi yang baik dan kualitas visual yang tinggi**. Gambar yang buram atau pecah akan memberikan kesan kurang profesional.
- **Format yang Didukung:** Gunakan format gambar yang umum didukung web, seperti **JPEG, PNG, atau GIF**.
- **Ukuran yang Optimal:** Perhatikan **ukuran file gambar**. Usahakan agar ukuran file tidak terlalu besar agar tidak memperlambat *loading* halaman.

Properti Rekomendasi Lain (Untuk Pengembangan Lebih Lanjut)

Selain `url` dan `image`, ada beberapa properti rekomendasi lain yang mungkin berguna untuk pengembangan lebih lanjut ClaimReview markup Anda, meskipun tidak sepenting dua properti yang baru saja dibahas. Properti-properti ini dapat memberikan informasi tambahan yang lebih kaya, tetapi untuk implementasi awal, fokus pada properti wajib dan `url` serta `image` sudah merupakan langkah yang baik. Beberapa properti rekomendasi tambahan tersebut antara lain:

- **reviewBody (Teks): Teks Lengkap Artikel Cek Fakta**
 - **Definisi:** Properti `reviewBody` dapat digunakan untuk menyertakan **teks lengkap dari artikel cek fakta** Anda langsung di dalam *structured data*.
 - **Tipe Data:** Teks (*Text*).
 - **Penggunaan dan Pertimbangan:**
 - Secara teknis, Anda dapat memasukkan seluruh teks artikel cek fakta Anda sebagai nilai properti `reviewBody`. Namun, ini **tidak selalu diperlukan atau direkomendasikan**. Konten utama artikel Anda seharusnya

sudah ada dalam HTML halaman web Anda, dan *markup ClaimReview* berfungsi untuk memberikan *metadata* terstruktur tentang konten tersebut.

- Penggunaan `reviewBody` mungkin lebih relevan dalam skenario di mana Anda ingin **mengindeks konten cek fakta Anda secara penuh** dalam *structured data*, atau jika Anda ingin menyediakan **akses ke teks artikel melalui API atau sistem lain yang mengkonsumsi *structured data***.
- Untuk tujuan tampilan *rich results* dan Google Fact Check Explorer, properti `reviewBody` **tidak wajib dan tidak memberikan dampak signifikan** pada tampilan. Fokus utama tetap pada properti wajib dan `url`, `image`, serta `reviewRating` dan `claimReviewed`.

- **Properti Tambahan untuk `author` (Organization atau Person): Memperkaya Informasi Penulis**

- Untuk memperkaya informasi tentang organisasi atau orang yang melakukan pemeriksaan fakta (yang didefinisikan dalam properti `author`), Anda dapat menggunakan properti-properti tambahan berikut dari tipe `Organization` atau `Person` di `Schema.org`:
 - `author.jobTitle` (Teks): Jabatan atau posisi penulis.
 - `author.image` (URL Gambar): Foto profil penulis atau logo organisasi.
 - `author.logo` (URL Gambar, Khusus untuk `Organization`): Logo organisasi.
 - `author.sameAs` (URL): Tautan ke profil media sosial atau halaman "Tentang Kami" organisasi/penulis.
 - `publisher` (`Organization`): Jika artikel cek fakta diterbitkan oleh organisasi lain (misalnya, sindikasi konten), Anda dapat menggunakan properti `publisher` untuk menentukan organisasi penerbit.
- **Contoh Penggunaan Properti Tambahan untuk `author` (`Organization`):**

JSON

```
"author": {
  "@type": "Organization",
  "name": "Media Cek Fakta ABC",
  "url": "https://www.mediacekfaktaabc.com/",
  "logo":
    "https://www.mediacekfaktaabc.com/logo-media-
    cek-fakta-abc.png",
  "sameAs":
    "https://twitter.com/mediacekfaktaabc"
}
```

Contoh ini menambahkan properti logo (URL logo organisasi) dan sameAs (URL profil Twitter organisasi) untuk memberikan informasi yang lebih lengkap tentang organisasi pemeriksa fakta.

Contoh Implementasi Properti Rekomendasi (JSON-LD *Snippet*)

Berikut adalah contoh bagaimana cara menambahkan properti url dan image (properti rekomendasi utama) ke *code snippet* ClaimReview markup yang sebelumnya hanya berisi properti wajib:

JSON

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ClaimReview",
  "url": "https://www.mediacekfaktaabc.com/cek-
  fakta/klaim-kenaikan-bbm-20-persen", // Properti URL
  ditambahkan (Rekomendasi)
  "image": "https://www.mediacekfaktaabc.com/gambar-
  thumbnail-cek-fakta-bbm.jpg", // Properti Image
  ditambahkan (Rekomendasi)
  "claimReviewed": "Pemerintah mengumumkan kenaikan
  harga BBM sebesar 20%",
  "author": {
    "@type": "Organization",
    "name": "Media Cek Fakta ABC",
```

```

      "url": "https://www.mediacekfaktaabc.com/"
    },
    "datePublished": "2024-07-27",
    "itemReviewed": {
      "@type": "Claim",
      "appearance": [
        {
          "@type": "PropertyValue",
          "url": "https://www.situsberita-
sumberklaim.com/artikel-klaim"
        }
      ]
    },
    "reviewRating": {
      "@type": "Rating",
      "ratingValue": "Sebagian Salah",
      "ratingExplanation": "Klaim tersebut sebagian
benar, namun ada konteks penting yang diabaikan."
    }
  }
</script>

```

Dengan menambahkan properti `url` dan `image`, *markup* `ClaimReview` Anda menjadi lebih lengkap dan berpotensi menghasilkan tampilan *rich results* yang lebih baik dan visibilitas yang lebih tinggi.

2.3: Contoh *Code Snippet* `ClaimReview` Markup (Lengkap)

Contoh *Code Snippet* Lengkap: Integrasi Semua Properti

Untuk memberikan gambaran yang lebih jelas, berikut adalah **dua contoh *code snippet* JSON-LD `ClaimReview` markup yang lebih komprehensif**, yang menggabungkan **semua properti wajib dan properti rekomendasi utama**, serta beberapa properti opsional lain yang mungkin relevan.

Contoh 1: *Code Snippet* Lengkap dengan *Rating* Numerik

Contoh ini menggunakan sistem *rating* numerik (skala 1-5) untuk properti `reviewRating`. Properti `itemReviewed` juga menyertakan properti

firstAppearance (tempat pertama kali klaim muncul), yang merupakan properti rekomendasi lain yang berguna untuk memberikan konteks lebih lengkap tentang klaim. Properti author juga diperkaya dengan properti logo untuk menampilkan logo organisasi.

JSON

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ClaimReview",
  "url": "https://www.mediacekfaktaabc.com/cek-fakta/klaim-kenaikan-bbm-20-persen",
  "image": "https://www.mediacekfaktaabc.com/gambar-thumbnail-cek-fakta-bbm.jpg",
  "claimReviewed": "Pemerintah mengumumkan kenaikan harga BBM sebesar 20%",
  "author": {
    "@type": "Organization",
    "name": "Media Cek Fakta ABC",
    "url": "https://www.mediacekfaktaabc.com/",
    "logo": "https://www.mediacekfaktaabc.com/logo-media-cek-fakta-abc.png" // Properti tambahan untuk author (Organization)
  },
  "datePublished": "2024-07-27",
  "itemReviewed": {
    "@type": "Claim",
    "appearance": [
      {
        "@type": "PropertyValue",
        "url": "https://www.situsberita-sumberklaim.com/artikel-klaim"
      }
    ],
    "firstAppearance": [ // Properti rekomendasi (firstAppearance) ditambahkan
      {
        "@type": "PropertyValue",
```

```

        "url": "https://www.akun-media-sosial-
penyebarklaim.com/postingan-pertama-klaim"
    }
]
},
"reviewRating": {
    "@type": "Rating",
    "ratingValue": 2,
    "bestRating": 5,
    "worstRating": 1,
    "ratingExplanation": "Klaim tersebut sebagian
benar, namun ada konteks penting yang diabaikan."
}
}
</script>

```

Contoh 2: Code Snippet Lengkap dengan Rating Label Teks

Contoh ini menggunakan sistem *rating* label teks ("Salah") untuk properti `reviewRating`. Struktur *markup* lainnya sama dengan contoh sebelumnya, menyertakan properti rekomendasi `url`, `image`, `firstAppearance`, dan `author.logo`.

JSON

```

<script type="application/ld+json">
{
    "@context": "https://schema.org",
    "@type": "ClaimReview",
    "url": "https://www.mediacekfaktaabc.com/cek-
fakta/klaim-kenaikan-bbm-20-persen",
    "image": "https://www.mediacekfaktaabc.com/gambar-
thumbnail-cek-fakta-bbm.jpg",
    "claimReviewed": "Pemerintah mengumumkan kenaikan
harga BBM sebesar 20%",
    "author": {
        "@type": "Organization",
        "name": "Media Cek Fakta ABC",
        "url": "https://www.mediacekfaktaabc.com/",

```

```

        "logo": "https://www.mediacekfaktaabc.com/logo-
media-cek-fakta-abc.png"
    },
    "datePublished": "2024-07-27",
    "itemReviewed": {
        "@type": "Claim",
        "appearance": [
            {
                "@type": "PropertyValue",
                "url": "https://www.situsberita-
sumberklaim.com/artikel-klaim"
            }
        ],
        "firstAppearance": [
            {
                "@type": "PropertyValue",
                "url": "https://www.akun-media-sosial-
penyebarklaim.com/postingan-pertama-klaim"
            }
        ]
    },
    "reviewRating": {
        "@type": "Rating",
        "ratingValue": "Salah", // Menggunakan label teks
"Salah" sebagai rating
        "ratingExplanation": "Klaim tersebut sepenuhnya
tidak benar dan tidak didukung oleh bukti yang
kredibel."
    }
}
</script>

```

Analisis *Code Snippet* Lengkap: Bedah Baris per Baris

Mari kita bedah salah satu contoh *code snippet* lengkap di atas (misalnya contoh dengan *rating* numerik) baris per baris untuk memastikan Anda memahami setiap bagian kode:

- `<script type="application/ld+json">` : Baris ini memulai *tag* `<script>` dengan atribut `type="application/ld+json"`. Ini adalah cara standar untuk menyisipkan *structured data* JSON-LD ke dalam halaman HTML. *Browser* dan mesin pencari akan mengidentifikasi dan memproses konten dalam *tag* ini sebagai *structured data* JSON-LD.
- `{` : Memulai objek JSON-LD utama, yang merupakan *container* untuk semua *markup* ClaimReview.
- `"@context": "https://schema.org",` : Menentukan `@context` sebagai "<https://schema.org>", yang memberitahu bahwa kita menggunakan *vocabulary* Schema.org. **Wajib ada.**
- `"@type": "ClaimReview",` : Menentukan `@type` sebagai "ClaimReview", yang menunjukkan bahwa kita sedang membuat *markup* untuk konten bertipe pemeriksaan fakta. **Wajib ada.**
- `"url": "https://www.mediacekfaktaabc.com/cek-fakta/klaim-kenaikan-bbm-20-persen",` : Properti url (URL artikel cek fakta). **Sangat Direkomendasikan/Praktisnya Wajib.**
- `"image": "https://www.mediacekfaktaabc.com/gambar-thumbnail-cek-fakta-bbm.jpg",` : Properti image (URL gambar *thumbnail*). **Sangat Direkomendasikan.**
- `"claimReviewed": "Pemerintah mengumumkan kenaikan harga BBM sebesar 20%",` : Properti `claimReviewed` (teks klaim yang diperiksa). **Wajib.**
- `"author": { ... },` : Objek `author`, mendeskripsikan organisasi pemeriksa fakta. **Wajib.** Di dalam objek `author`:
 - `"@type": "Organization",` : Menentukan tipe `Organization` untuk penulis. **Wajib dalam objek author.**
 - `"name": "Media Cek Fakta ABC",` : Properti `name` organisasi. **Wajib dalam objek author.**
 - `"url": "https://www.mediacekfaktaabc.com/",` : Properti `url` organisasi. **Wajib dalam objek author.**
 - `"logo": "https://www.mediacekfaktaabc.com/logo-media-cek-fakta-abc.png"` : Properti `logo` organisasi. **Rekomendasi untuk memperkaya info author.**

- "datePublished": "2024-07-27", : Properti datePublished (tanggal publikasi artikel cek fakta). **Wajib.**
- "itemReviewed": { ... }, : Objek itemReviewed, mendeskripsikan detail klaim yang diperiksa. **Wajib.** Di dalam objek itemReviewed:
 - "@type": "Claim", : Menentukan tipe Claim untuk objek itemReviewed. **Wajib dalam objek itemReviewed.**
 - "appearance": [...], : Array appearance, daftar tempat klaim muncul. **Wajib dalam objek itemReviewed.** Di dalam *array* appearance:
 - { "@type": "PropertyValue", "url": "https://www.situsberita-sumberklaim.com/artikel-klaim" } : Objek PropertyValue yang mendeskripsikan satu tempat kemunculan klaim (URL). **Wajib dalam *array* appearance.**
 - "firstAppearance": [...] , : Array firstAppearance, daftar tempat klaim pertama kali muncul. **Rekomendasi untuk memperkaya info itemReviewed.** Struktur sama dengan appearance.
- "reviewRating": { ... } : Objek reviewRating, mendeskripsikan hasil *rating* cek fakta. **Wajib.** Di dalam objek reviewRating:
 - "@type": "Rating", : Menentukan tipe Rating untuk objek reviewRating. **Wajib dalam objek reviewRating.**
 - "ratingValue": 2, : Properti ratingValue (nilai numerik *rating*). **Wajib dalam objek reviewRating.**
 - "bestRating": 5, : Properti bestRating (nilai terbaik skala *rating*). **Rekomendasi jika menggunakan skala numerik.**
 - "worstRating": 1, : Properti worstRating (nilai terburuk skala *rating*). **Rekomendasi jika menggunakan skala numerik.**
 - "ratingExplanation": "Klaim tersebut sebagian benar, namun ada konteks penting yang diabaikan." : Properti ratingExplanation (penjelasan singkat *rating*). **Wajib dalam objek reviewRating.**

- } : Menutup objek `reviewRating`.
- } : Menutup objek `itemReviewed`.
- } : Menutup objek `author`.
- } : Menutup objek JSON-LD utama (`ClaimReview`).
- `</script>` : Menutup *tag* `<script>`.

Pilih Sistem *Rating* yang Tepat: Numerik atau Label Teks?

Dalam *markup* `ClaimReview`, Anda memiliki fleksibilitas untuk menggunakan sistem *rating* numerik atau label teks. Pilihan mana yang lebih baik tergantung pada sistem *rating* yang sudah Anda gunakan di website Anda dan preferensi Anda.

- **Rating Numerik:**
 - **Kelebihan:** Lebih presisi dan memungkinkan skala *rating* yang lebih *granular*. Cocok jika Anda memiliki sistem *rating* yang kompleks dengan banyak tingkatan *rating*.
 - **Kekurangan:** Mungkin kurang intuitif dan sulit dipahami oleh audiens awam jika skala *rating* tidak dijelaskan dengan baik. Penting untuk selalu menyertakan `bestRating` dan `worstRating` untuk memperjelas skala.
 - **Contoh:** Skala 1-5, skala 0-10, dll.
- **Rating Label Teks:**
 - **Kelebihan:** Lebih mudah dipahami dan intuitif bagi audiens umum. Label teks seperti "Benar", "Salah", "Menyesatkan" lebih langsung dan jelas maknanya.
 - **Kekurangan:** Kurang presisi dibandingkan skala numerik. Mungkin kurang fleksibel jika Anda ingin sistem *rating* yang lebih kompleks.
 - **Contoh:** "Benar", "Salah", "Sebagian Benar", "Sebagian Salah", "Menyesatkan", "Tidak Berbasis Fakta", dll.

Rekomendasi: Untuk kemudahan pemahaman dan konsistensi dengan praktik umum media cek fakta, **sistem *rating* label teks seringkali lebih disarankan**, terutama untuk implementasi awal. Pastikan label teks yang Anda gunakan

konsisten di seluruh website Anda dan mudah dipahami oleh audiens. Jika Anda menggunakan skala numerik, pastikan untuk **menjelaskan skala *rating*** di website Anda dan selalu sertakan `bestRating` dan `worstRating` dalam *markup*.



Latihan

Untuk memperdalam pemahaman Anda, Anda dapat melakukan latihan berikut:

1. **Studi Kasus Artikel Cek Fakta:** Pilih contoh artikel cek fakta dari website media cek fakta (bisa website Anda sendiri atau contoh dari internet).
2. **Identifikasi Elemen-Elemen:** Baca artikel tersebut dan identifikasi elemen-elemen yang diperlukan untuk ClaimReview markup:
 - Klaim yang diperiksa (*claimReviewed*)
 - Organisasi/Penulis cek fakta (*author*)
 - Tanggal publikasi (*datePublished*)
 - Tempat klaim muncul (*appearance, firstAppearance*)
 - Hasil *rating* dan penjelasannya (*reviewRating*)
 - URL artikel (*url*)
 - Gambar *thumbnail* yang relevan (*image*)
3. **Buat Code Snippet JSON-LD:** Berdasarkan elemen-elemen yang Anda identifikasi, coba buat *code snippet* JSON-LD ClaimReview markup lengkap menggunakan salah satu contoh *template* yang telah kita bahas. Pilih sistem *rating* yang sesuai (numerik atau label teks).
4. **Validasi Code Snippet:** Gunakan *tool* validasi Google Rich Results Test (yang akan kita bahas di Modul lain) untuk memeriksa apakah *code snippet* yang Anda buat valid dan tidak ada *error*.

Latihan ini akan membantu Anda mempraktikkan pemahaman tentang properti-properti ClaimReview markup dan bagaimana menerapkannya pada konten cek fakta yang nyata.

Kesimpulan

Dalam sesi ini, kita telah membahas secara mendalam tentang elemen-elemen penting dalam ClaimReview markup, termasuk properti wajib dan rekomendasi. Pemahaman yang kuat tentang struktur dan properti ini adalah kunci untuk mengimplementasikan ClaimReview markup dengan benar dan memaksimalkan manfaatnya untuk konten cek fakta Anda.

Di sesi selanjutnya, kita akan membahas implementasi praktis ClaimReview markup pada konten cek fakta, termasuk langkah-langkah implementasi, *tools* validasi, dan praktik terbaik.





CEK FAKTA

**ALIANSI
JURNALIS
INDEPENDEN**



Asosiasi
Media Siber
Indonesia



Implementasi Praktis Claim Review Markup pada Konten Cek Fakta

Pendahuluan

Pada Modul sebelumnya kita mempelajari struktur dan elemen teknis ClaimReview markup, kini saatnya kita membahas fase praktisnya. Modul ini akan memandu Anda langkah demi langkah dalam mengimplementasikan ClaimReview markup pada konten cek fakta yang Anda produksi.

Pembahasan akan fokus pada cara-cara implementasi yang paling umum dan mudah diterapkan, terutama bagi pengguna platform WordPress, baik menggunakan plugin maupun metode manual.

Tujuan Modul

Setelah menyelesaikan modul ini, Anda diharapkan mampu:

1. **Memahami langkah-langkah implementasi ClaimReview markup** secara sistematis pada artikel cek fakta.
2. **Mampu mengimplementasikan ClaimReview markup secara manual** dengan menyisipkan kode JSON-LD ke dalam HTML artikel.
3. **Mampu menggunakan plugin WordPress (khususnya Rank Math SEO Pro)** untuk mempermudah implementasi ClaimReview markup.
4. **Menggunakan *tools* validasi** yang disediakan Google untuk memastikan *markup* ClaimReview valid dan berfungsi dengan baik.
5. **Memahami proses publikasi dan monitoring** implementasi ClaimReview markup pada konten Anda.

Durasi Pembelajaran: 2-3 jam (dapat disesuaikan, sesi praktik mungkin memerlukan waktu lebih lama)

Struktur Modul

Modul ini terdiri dari tiga sesi utama yang akan membimbing Anda melalui proses implementasi praktis:

- **Sesi 1: Langkah-Langkah Implementasi ClaimReview Markup**
 - Sub-sesi 1.1: Persiapan: Identifikasi Klaim dan Produksi Konten Cek Fakta
 - Sub-sesi 1.2: Implementasi Claim Review dengan Markup Tool – Google Fact Check Tool
 - Sub-sesi 1.3: Penyiapan Markup JSON-LD: Manual atau Otomatis?
 - Sub-sesi 1.4: Implementasi Kode JSON-LD ke HTML di WordPress (Manual dan Plugin)
 - Sub-sesi 1.5: Validasi Markup dengan Google Rich Results Test
 - Sub-sesi 1.6: Publikasi dan Monitoring
- **Sesi 2: Tools untuk Memudahkan Implementasi dan Validasi ClaimReview Markup**
 - Sub-sesi 2.1: Google Rich Results Test: Uji Validasi Markup
 - Sub-sesi 2.2: Google Search Console: Monitoring Status *Rich Results*
 - Sub-sesi 2.3: Schema Markup Generator (Pihak Ketiga)
 - Sub-sesi 2.4: Fact Check Markup Validator
- **Sesi 3: Praktik Implementasi**



Sesi 1: Langkah-Langkah Implementasi ClaimReview Markup

Sesi ini akan menjabarkan langkah-langkah konkret yang perlu Anda lakukan untuk mengimplementasikan ClaimReview markup pada artikel cek fakta Anda, mulai dari persiapan hingga publikasi dan monitoring.

1.1: Persiapan: Identifikasi Klaim dan Produksi Konten Cek Fakta

Langkah 1: Identifikasi Klaim Spesifik

Langkah pertama sebelum mengimplementasikan ClaimReview markup adalah **mengidentifikasi klaim spesifik** yang akan Anda periksa faktanya dalam artikel Anda. Klaim ini akan menjadi nilai dari properti `claimReviewed` dalam *markup* nanti.

- **Proses Identifikasi Klaim:**
 - **Baca dan pahami sumber klaim:** Perhatikan sumber asli klaim (artikel berita, postingan media sosial, video, dll.) dengan seksama.
 - **Ekstraksi klaim secara verbatim:** Salin klaim tersebut persis seperti yang tertulis atau diucapkan. Usahakan untuk tidak mengubah atau menginterpretasi klaim asli.
 - **Fokus pada satu klaim utama:** Jika artikel Anda fokus pada satu klaim utama, identifikasi klaim utama tersebut.
 - **Jika artikel memeriksa beberapa klaim, pilih klaim yang paling signifikan:** Jika artikel Anda memeriksa beberapa klaim, pilih klaim yang paling penting, paling banyak dibicarakan, atau paling berpotensi menyesatkan untuk dijadikan `claimReviewed`. Untuk artikel yang memeriksa banyak klaim, mungkin lebih tepat untuk membuat artikel cek fakta terpisah untuk setiap klaim utama.
 - **Pastikan klaim *fact-checkable*:** Klaim yang dipilih harus merupakan pernyataan faktual yang dapat diperiksa kebenarannya melalui bukti dan data. Opini atau pernyataan subjektif tidak cocok untuk ClaimReview markup.

- **Contoh Identifikasi Klaim:**

Misalkan Anda menemukan artikel berita dengan judul: "Pemerintah Umumkan Kenaikan Harga BBM 20% Mulai Minggu Depan".

Klaim yang dapat Anda identifikasi dari judul tersebut adalah: **"Pemerintah mengumumkan kenaikan harga BBM sebesar 20% mulai minggu depan"**. Ini adalah klaim faktual yang dapat diperiksa kebenarannya.

Langkah 2: Produksi Konten Cek Fakta Berkualitas

Langkah kedua, dan yang paling krusial, adalah **memproduksi konten cek fakta yang berkualitas tinggi**. ClaimReview markup hanyalah *markup*—ia membantu mesin pencari menemukan dan memahami konten Anda, tetapi *markup* tidak dapat menggantikan kualitas konten itu sendiri.

- **Kriteria Konten Cek Fakta Berkualitas:**

- **Akurasi dan Ketelitian:** Informasi yang disajikan harus akurat, berdasarkan bukti dan data yang valid.
- **Transparansi Metodologi:** Jelaskan secara transparan metodologi pemeriksaan fakta yang Anda gunakan, termasuk sumber-sumber Anda dan proses analisis yang Anda lakukan.
- **Imparsialitas dan Objektivitas:** Sajikan informasi secara imparsial dan objektif, hindari bias atau keberpihakan.
- **Klaritas dan Kemudahan Pemahaman:** Tulis artikel dengan bahasa yang jelas, ringkas, dan mudah dipahami oleh audiens umum.
- **Konteks yang Cukup:** Berikan konteks yang cukup untuk klaim yang diperiksa, termasuk latar belakang klaim, sumber klaim, dan implikasinya.
- **Kesimpulan yang Jelas (Rating):** Sajikan kesimpulan atau *rating* pemeriksaan fakta Anda secara jelas dan ringkas. Sistem *rating* yang konsisten dan mudah dipahami sangat dianjurkan.

- **Integrasikan Elemen-Elemen Cek Fakta dalam Artikel:** Pastikan artikel cek fakta Anda secara jelas memuat elemen-elemen berikut:
 - **Klaim yang diperiksa:** Nyatakan klaim yang Anda periksa di awal artikel.
 - **Proses pemeriksaan fakta:** Jelaskan langkah-langkah yang Anda lakukan untuk memeriksa klaim tersebut, termasuk sumber-sumber yang Anda gunakan dan analisis yang Anda lakukan.
 - **Bukti dan data pendukung:** Sajikan bukti dan data yang mendukung kesimpulan Anda.
 - **Kesimpulan atau *rating*:** Nyatakan kesimpulan atau *rating* pemeriksaan fakta Anda secara jelas.

Sub-sesi 1.2: Implementasi Claim Review dengan Markup Tool - Google Fact Check Tool

Memahami Markup Tool

Markup tool merupakan salah satu fitur yang terdapat di Google Fact Check Tools. Alat ini digunakan untuk menambahkan data terstruktur ke artikel hasil pemeriksaan fakta. Langkah ini untuk menandai klaim yang telah ditinjau kebenarannya.

Secara sederhana, Markup Tool pada Claim Review berarti memberi tanda pada sebuah artikel berita, sehingga pembaca dan mesin pencari bisa memahami dan memverifikasi klaim yang ada di dalamnya dengan mudah.

Sebelumnya, markup terstruktur (schema.org/ClaimReview) hanya bisa ditambahkan langsung ke kode HTML artikel, yang rumit dan rawan salah. Alat ini mempermudah dengan formulir web, tanpa perlu mengubah artikel. Markup yang dihasilkan sama dengan yang ditanamkan langsung.

Sederhananya, Markup Tool adalah alat bantu untuk "menandai" artikel cek fakta di internet. Fungsinya seperti stabilo warna-warni untuk menandai bagian penting dalam buku. Markup Tool ini tak ubahnya seperti stabilo digital yang membantu mesin pencari seperti Google untuk memahami artikel cek fakta.

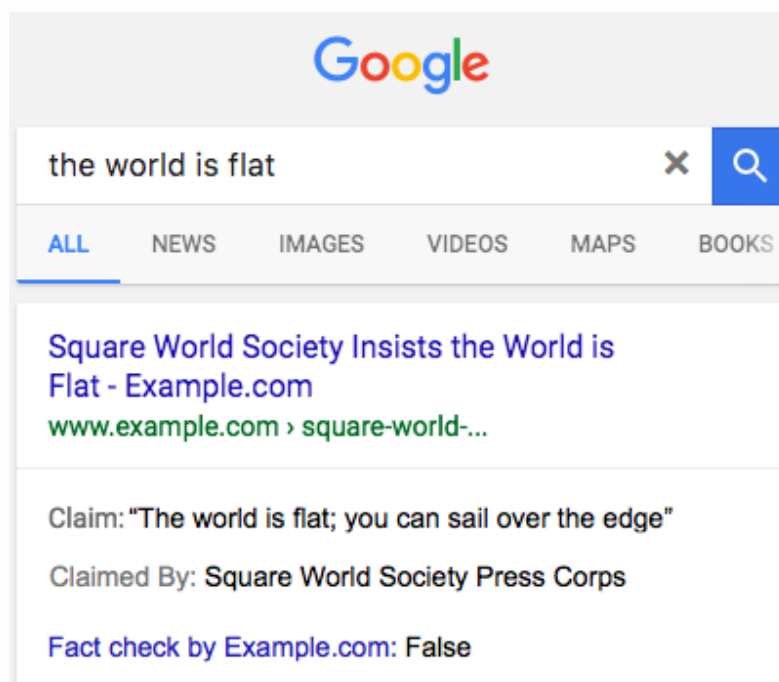
Dengan Markup Tool, media pemeriksa fakta bisa memberi tahu Google bahwa artikelnya berisi hasil pemeriksaan fakta atas klaim tertentu. Pada dasarnya,

markup tool bisa memberitahu siapa yang membuat klaim, kapan klaim itu muncul, dan apakah klaim itu benar, salah, atau sebagian benar. Informasi ini kemudian akan ditampilkan oleh Google di hasil pencarian, sehingga pembaca bisa langsung melihat ringkasan hasil pemeriksaan fakta tanpa harus membaca keseluruhan artikel.

Memberi informasi yang lebih detail

Dengan menggunakan Markup Tool, label yang disematkan pada artikel berita dilengkapi dengan informasi tambahan yang memudahkan untuk membuat penilaian atas klaim tersebut. Informasi penjelas itu antara lain:

1. Siapa yang membuat klaim
2. Kapan klaim tersebut dibuat
3. Apakah klaim tersebut benar, salah, atau sebagian benar
4. Tautan ke artikel yang menjelaskan hasil verifikasi



Claim: merujuk pada pernyataan yang dibuat oleh seseorang atau sekelompok orang dan belum terbukti kebenarannya.

Claimed by: pihak pertama yang membuat klaim yang diperiksa faktanya.

Fact check by: pihak yang memeriksa fakta

Contoh Pemeriksaan Fakta dan Tampilan Claim Review

TEMPO Langganan

Keliru: Video Raffi Ahmad Promosikan Situs Judi Online Miliknyanya

Sebuah video beredar di Facebook yang diklaim memperlihatkan selebriti Raffi Ahmad mempromosikan situs judi online (judol) miliknya.

6 Maret 2025 | 19:50 WIB

KELIRU
Berdasarkan semua bukti yang ada, pernyataan ini tidak akurat.

WAL KLARIFIKASI RAFFI AHMAD

Tempo.co
https://www.tempo.co › cekfakta

Keliru: Video Raffi Ahmad Promosikan Situs Judi Online ...

Claim: Keliru: Video Raffi Ahmad Promosikan Situs Judi Online Miliknya
Claimed by: Tempo.co
Fact check by Tempo.co: keliru

Catatan: pada bagian “Claimed by” masih ambigu karena tertulis Tempo.co. Semestinya adalah pembuat klaim yang pertama. Hal ini bisa terjadi karena data yang dimasukkan tidak sesuai dengan struktur yang benar.

Cara Claim Review Menggunakan Markup Tool

1. Identifikasi klaim yang sudah beredar lewat website atau media sosial.
2. Lakukan pemeriksaan fakta menggunakan berbagai teknik cek fakta yang bisa membuktikan klaim tersebut. Lalu tayangkan di website atau kanal cek fakta yang dimiliki.
3. Pastikan telah memiliki akses ke Google Search Console
4. Akses Markup Tool di <https://toolbox.google.com/factcheck/markuptool>
5. Masukkan URL yang dijadikan penanda
 - Masukkan URL artikel hasil pengecekan fakta yang telah ditayangkan di website
 - Pilih opsi “Go”

Google Fact Check Tools

Explorer

- Markup Tool
- APIs
- FAQ
- Report Issue
- Privacy
- Terms

Fact Check Markup Tool

Enter article URL to create or edit markup

Match one of the following: digitalqueen.digitalmama.id, https://digitalmama.id/, https://digitalqueen.digitalmama.id/, screencore.digitalmama.id

We've made updates to the Fact Check Markup Tool. [Learn more about what's new.](#)

A place for you to add and view your ClaimReview markup, allowing easy access to Google, Bing, Facebook, researchers and other interested parties. [Learn more.](#)

6. Isi informasi klaim

☰ Google Fact Check Tools 📱

Fact Check Markup Tool

Create markup

Fact-check article (readonly)
 → **URL artikel hasil cek fakta**

URL of your fact-check article to create markup for.

Fact-check article date → **tanggal penayangan artikel**

When your fact-check article was or will be published.

▲ Required by: Facebook

Fact-check organization name → **nama media**

Name of your fact-checking organization.

7. Jika klaim tidak hanya berupa teks

Review #1 - Media Remove MediaReview

Image Image with overlaid text Audio Video → **Pilih sesuai media yang membuat klaim**

Original
No evidence the image has been misleadingly altered or manipulated, though it still may contain false or misleading claims.

Missing Context
Presenting unaltered images in an inaccurate manner to misrepresent the image and mislead the viewer.

Edited or cropped
Presenting part of an image from a larger whole to mislead the viewer.

Transformed
Adding or deleting visual elements to give the image a different meaning with the intention to mislead.

Staged
An image that was created using actors or similarly contrived, such as a screenshot of a fake tweet.

Satire
An image that was created as a political or humorous commentary and is presented in that context.

▲ At least one manipulation rating must be selected.

URL of the image → **URL foto/video atau URL unggahan di medsos**

When possible provide the URL of the image file that was fact checked rather than the page URL. For social media posts, the URL of the post is acceptable if the image URL is not accessible.

URL for the original version of the image → **URL laman yang mengandung foto/video tersebut**

This can be the original page containing the image or the original image itself.

Informasi tentang klaim

Review #1 - Claim

Claim reviewed → **Apa klaimnya, ex: bumi itu datar**

What the person or entity claimed to be true.
▲ Required by: Google, Facebook, Bing

Claim date → **tanggal klaim ditayangkan**

When the person or entity made the claim.

Claim appearance → **URL konten yang berisi klaim tersebut**

URL for a document where this claim appears.
+ Add another claim appearance

Original appearance

Claim author name → **Nama orang/entitas yang membuat klaim tersebut**

Name of the person or entity who made the claim.

Rating text → **Penilaian atas klaim, contoh: salah, sebagian besar salah, separuh benar, sebagian besar benar, benar**

Your written assessment of the claim.
▲ Required by: Google, Facebook, Bing

Show more fields

+ Add claim review + Add media review

Create Cancel

This tool allows you to generate Schema.org-compliant [ClaimReview markup](#) and to submit your markup to Google and to add your

8. Periksa apakah struktur sudah benar

Pemeriksaan struktur data menggunakan Google Rich Result Test - Google Search Console

Bisa diakses di:

https://search.google.com/test/rich-results?utm_campaign=devsite&utm_medium=jsonld&utm_source=facebookcheck

Contoh hasil uji:

Rich Results Test

Does your page support rich results?

URL <> CODE

<https://digitalmama.id/2025/02/cek-fakta-awas-hoaks-info-loker-badan-gizi-nasional/>

Google Inspection Tool smartphone TEST URL

Test results

2 valid items detected
Valid items are eligible for Google Search's rich results. [Learn more](#)

VIEW TESTED PAGE PREVIEW RESULTS

Details

Crawl

✓ Crawled successfully on Feb 27, 2025, 2:28:25 PM

Detected structured data

✓ Articles	1 valid item detected	>
✓ Breadcrumbs	1 valid item detected	>

Explorer

Tidak ada jaminan hasil cek fakta bisa masuk ditemukan di Explorer, tapi setidaknya ada beberapa hal yang bisa dijadikan panduan agar Claim Review muncul di Explorer:

- Akurasi dan Transparansi: Analisis cek fakta harus dapat dilacak dan transparan mengenai sumber dan metode yang digunakan.
- Data Terstruktur yang Tepat: Penggunaan data terstruktur Claim Review harus benar dan sesuai dengan panduan Google.
- Kredibilitas Penerbit: Google cenderung menampilkan hasil cek fakta dari penerbit yang dianggap kredibel dan memiliki reputasi baik.
- Konsistensi: Penerapan markup yang konsisten membantu Google memahami dan menampilkan hasil cek fakta dengan lebih baik.

Terdapat opsi: English dan All Languages

Google Fact Check Tools

Explorer

Markup Tool

APIs

FAQ

Report Issue

Privacy

Terms

More results in other languages

Language filter
All Languages

Search fact checks about a topic or person

Search by image

Recent fact checks

Claim by Publications TikTok et X:
Les frais de succession vont augmenter en mars 2025

TF1 Info rating: Faux
[VÉRIF - Des droits de succession bientôt en hausse ? Attention à ces vidéos trompeuses](#)
23 hours ago

Inheritance
TF1 Info

Pencarian berdasar topik, gambar, dan situs

Sub-sesi 1.3: Penyiapan Markup JSON-LD: Manual atau Otomatis?

Ada dua pendekatan utama untuk menyiapkan kode JSON-LD ClaimReview markup:

Pendekatan 1: Penyiapan Markup Manual

- **Menggunakan *Template* atau Contoh *Code Snippet*:**
 - Anda dapat menggunakan *template* atau contoh *code snippet* ClaimReview markup yang telah kita bahas di Modul sebelumnya sebagai titik awal.
 - Salin *template code snippet* JSON-LD.
 - **Sesuaikan nilai properti:** Ganti nilai-nilai *placeholder* dalam *template* dengan informasi yang sesuai dari artikel cek fakta Anda:
 - `url`: URL artikel cek fakta Anda.
 - `image`: URL gambar *thumbnail* artikel Anda.
 - `claimReviewed`: Teks klaim yang Anda identifikasi.
 - `author.name`, `author.url`, `author.logo`: Informasi organisasi/penulis cek fakta Anda.
 - `datePublished`: Tanggal publikasi artikel Anda.
 - `itemReviewed.appearance.url`: URL sumber klaim yang Anda periksa.
 - `itemReviewed.firstAppearance.url` (jika ada): URL sumber pertama klaim (opsional).
 - `reviewRating.ratingValue`, `reviewRating.ratingExplanation`, `reviewRating.bestRating`, `reviewRating.worstRating` (sesuai sistem *rating* Anda): Hasil *rating* dan penjelasannya.
- **Menggunakan Teks Editor:**
 - Anda dapat menggunakan teks editor sederhana (seperti Notepad, Notepad++, TextEdit, atau *code editor* seperti VS Code, Sublime Text) untuk mengedit *code snippet* JSON-LD. Anda dapat mendownload Notepad++ disini: <https://notepad-plus-plus.org/downloads/>

- Pastikan untuk **memperhatikan *syntax* JSON** dengan benar (tanda kutip, kurung kurawal, titik dua, koma). *Code editor* biasanya memiliki fitur *syntax highlighting* yang dapat membantu Anda menghindari kesalahan *syntax*.

Pendekatan 2: Penyiapan Markup Otomatis (dengan *Schema Markup Generator*)

- **Menggunakan *Schema Markup Generator Online*:**

- Ada berbagai *tools Schema Markup Generator online* gratis yang dapat membantu Anda membuat *code snippet* JSON-LD secara otomatis. Salah satunya adalah **Easy Schema Markup Generator**: <https://easyschema.com/factcheck/>

- **Cara Menggunakan *Generator*:**

1. **Isi formulir dengan data artikel:** Isi setiap kolom properti yang disediakan oleh *generator* dengan informasi artikel cek fakta Anda (klaim, penulis, tanggal, *rating*, URL, dll.).
2. **Generate *code snippet*:** Saat mengisi kolom dengan *value* yang sesuai, *Generator* secara otomatis membuat *code snippet* JSON-LD berdasarkan data yang Anda masukkan.

The screenshot displays the 'Easy Schema / Fact Check' interface. On the left, there is a form with several sections: 'Fact Check' (with Date Published, URL, and claimReviewed fields), 'Item Reviewed' (with Author Name, sameAs, and Date Published fields), and 'Appearance' (with Image, URL, and Headline fields). Below these are additional fields for Date Published, Author Name, Publisher Name, Logo URL, Author Name, Rating Value, Best Rating, Worst Rating, and Alternate Name. On the right side, a code editor shows the generated JSON-LD code snippet, which includes metadata like @context, @type, and @type for ClaimReview, OpinionNewsArticle, and Person.

3. **Salin *code snippet*:** Salin *code snippet* JSON-LD yang dihasilkan.

- **Keuntungan Menggunakan *Generator*:**

- **Mempermudah proses pembuatan *markup*:** *Generator* dapat menghemat waktu dan mengurangi risiko kesalahan *syntax*, terutama bagi mereka yang kurang familiar dengan *coding* JSON-LD.
- **Memandu pengisian properti:** Formulir dalam *generator* membantu Anda memastikan semua properti penting (terutama yang wajib) terisi dengan benar.

- **Keterbatasan Menggunakan *Generator*:**

- **Mungkin kurang fleksibel untuk kasus yang kompleks:** *Generator* mungkin memiliki keterbatasan dalam menangani kasus implementasi yang lebih kompleks atau properti *custom*.
- **Perlu diperiksa ulang:** Meskipun *generator* membantu, tetap penting untuk **memeriksa ulang *code snippet* yang dihasilkan** untuk memastikan semua informasi sudah benar dan sesuai dengan artikel Anda.

Sub-sesi 1.4: Implementasi Kode JSON-LD ke HTML di WordPress (Manual & Plugin)

Setelah Anda memiliki *code snippet* JSON-LD ClaimReview markup, langkah selanjutnya adalah **mengimplementasikannya ke halaman artikel cek fakta Anda di WordPress**. Ada dua cara utama untuk melakukan ini di WordPress:

Metode 1: Implementasi Manual dengan Insert HTML di Editor WordPress

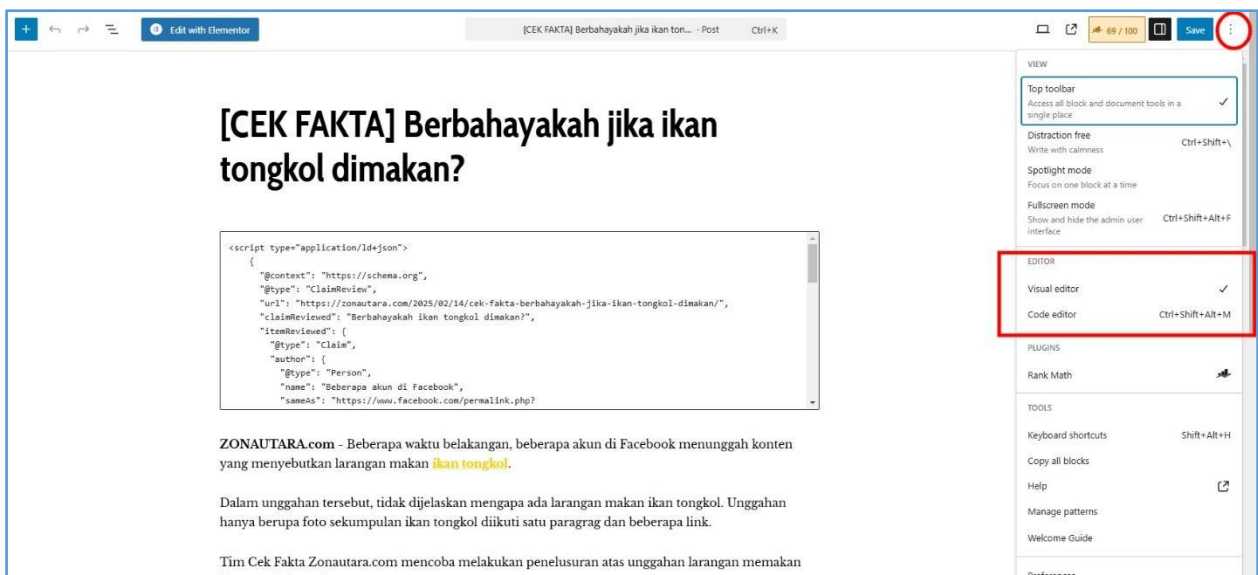
Metode ini cocok untuk pengguna WordPress yang ingin **kontrol penuh** atas *markup* dan tidak ingin bergantung pada plugin. Metode ini juga berguna jika Anda hanya perlu menambahkan *markup* pada beberapa artikel tertentu.

- **Langkah-Langkah Implementasi Manual (Insert HTML di Editor WordPress):**

1. **Buka Artikel di Editor WordPress:** Buka artikel cek fakta yang ingin Anda tambahkan ClaimReview markup di *editor* WordPress (baik *Block Editor* atau *Classic Editor*).

2. **Pilih Mode "Text" atau "Code Editor":**

- **Block Editor (Gutenberg):** Klik menu "Options" (ikon tiga titik vertikal) di pojok kanan atas *editor*, lalu pilih "Code editor". Atau, di *block* mana pun, klik ikon tiga titik vertikal, pilih "Edit as HTML".
- **Classic Editor:** Klik *tab* "Text" di sebelah *tab* "Visual" di atas *editor*. Ini akan menampilkan *code editor* HTML.



Pilihan "Code editor" di WordPress Block Editor

3. **Sisipkan Code Snippet JSON-LD:**

- **Cari tag <head> atau <body>:** Dalam *code editor* HTML, cari *tag* <head> atau <body>. Anda dapat menyisipkan *code snippet* JSON-LD di dalam *tag* <head> atau <body>. **Direkomendasikan untuk menyisipkan di dalam <head>** karena lebih terstruktur dan tidak mengganggu konten utama halaman.
- **Tempelkan code snippet JSON-LD:** Tempelkan *code snippet* JSON-LD ClaimReview markup yang telah Anda

siapkan langsung di bawah tag `<head>` atau di awal tag `<body>`. Atau Anda dapat menyisipkannya di bagian paling atas sebelum karakter apapun di dalam editor.

Contoh Penyisipan di dalam `<head>` (Block Editor atau Classic Editor):

HTML

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Judul Artikel Cek Fakta</title>
  <meta name="description" content="Deskripsi
singkat artikel cek fakta">
  <script type="application/ld+json">
  {
    "@context": "https://schema.org",
    "@type": "ClaimReview",
    "url": "...",
    "image": "...",
    "claimReviewed": "...",
    "author": {
      "@type": "Organization",
      "name": "...",
      "url": "..."
    },
    "datePublished": "...",
    "itemReviewed": {
      "@type": "Claim",
      "appearance": [
        {
          "@type": "PropertyValue",
          "url": "..."
        }
      ]
    }
  }
</script>
</head>
<body>
```

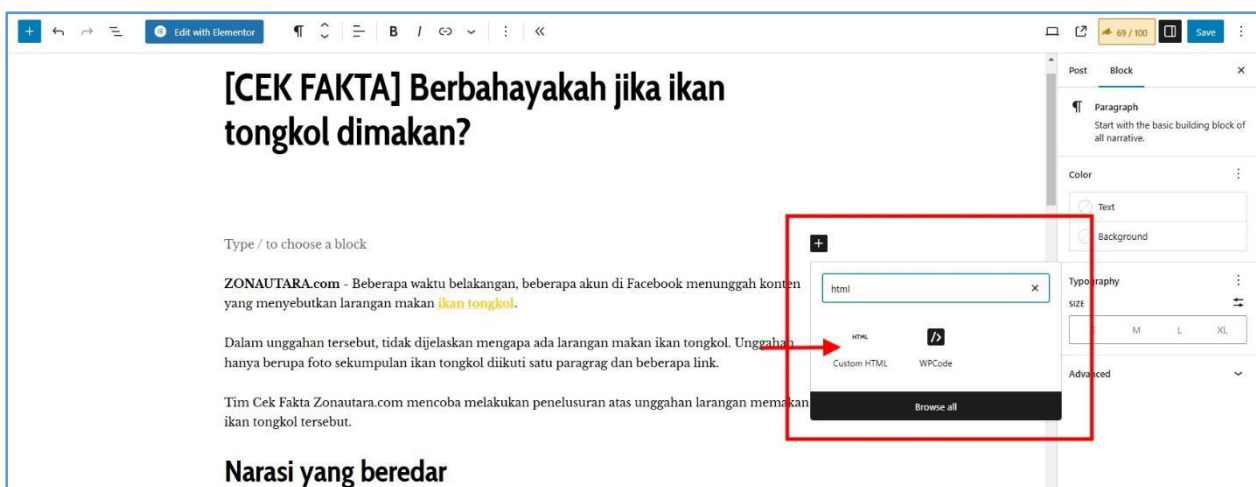
```

    }
  ]
},
"reviewRating": {
  "@type": "Rating",
  "ratingValue": "...",
  "ratingExplanation": "..."
}
}
</script>
<link rel="stylesheet" href="style.css">
</head>
<body>
  </body>
</html>

```

4. **Perbarui atau Publikasikan Artikel:** Setelah menyisipkan *code snippet*, klik "Update" (jika artikel sudah dipublikasikan) atau "Publish" (jika artikel baru) untuk menyimpan perubahan.
5. **Validasi Markup (Langkah Selanjutnya):** Setelah publikasi, lakukan validasi *markup* menggunakan Google Rich Results Test (akan dibahas di sub-sesi 1.4).

TIPS: Anda juga dapat menyisipkan *code snippet* JSON-LD melalui *insert block custom HTML*, di bagian paling atas isi konten Anda (*body*).



[CEK FAKTA] Berbahayakah jika ikan tongkol dimakan?

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ClaimReview",
  "url": "https://zonautara.com/2025/02/14/cek-fakta-berbahayakah-jika-ikan-tongkol-dimakan/",
  "claimReviewed": "Berbahayakah ikan tongkol dimakan?",
  "itemReviewed": {
    "@type": "Claim",
    "author": {
      "@type": "Person",
      "name": "Beberapa akun di Facebook",
      "sameAs": "https://www.facebook.com/permalink.php?"
    }
  }
}
```

ZONAUTARA.com - Beberapa waktu belakangan, beberapa akun di Facebook menunggah konten yang menyebutkan larangan makan **ikan tongkol**.

Dalam unggahan tersebut, tidak dijelaskan mengapa ada larangan makan ikan tongkol. Unggahan hanya berupa foto sekumpulan ikan tongkol diikuti satu paragraf dan beberapa link.

Tim Cek Fakta Zonautara.com mencoba melakukan penelusuran atas unggahan larangan memakan ikan tongkol tersebut.

- **Keuntungan Metode Manual:**

- **Kontrol penuh:** Anda memiliki kontrol penuh atas *markup* yang diimplementasikan.
- **Tidak perlu plugin:** Tidak memerlukan instalasi plugin tambahan.
- **Gratis:** Tidak ada biaya tambahan.

- **Kekurangan Metode Manual:**

- **Memerlukan *coding* HTML dasar:** Memerlukan pemahaman dasar tentang HTML dan cara mengedit *code editor* WordPress.
- **Potensi kesalahan *syntax*:** Risiko kesalahan *syntax* JSON-LD lebih tinggi jika dilakukan manual (teliti saat mengedit dan selalu validasi).
- **Tidak terotomatisasi:** Prosesnya manual dan perlu diulang untuk setiap artikel yang ingin di-*markup*.

Metode 2: Implementasi dengan Plugin WordPress (Contoh: Rank Math SEO Pro)

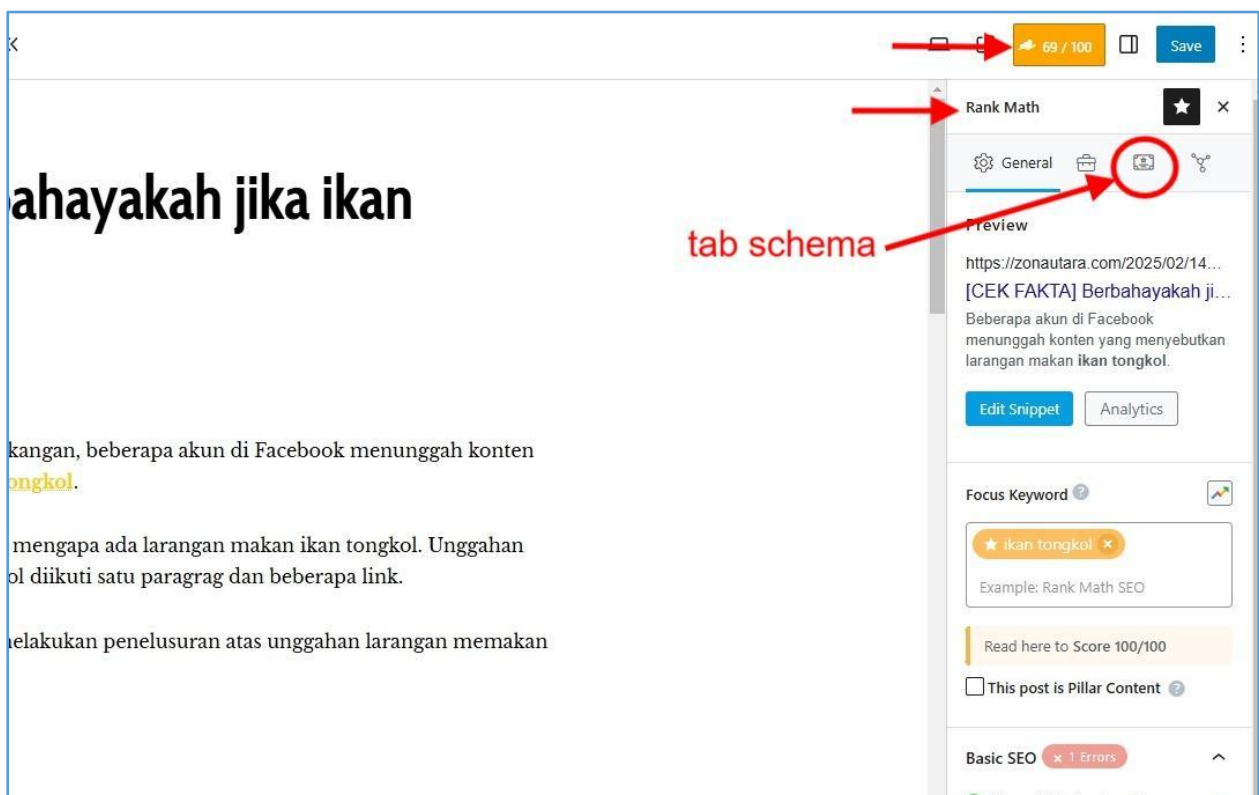
Metode ini **lebih mudah dan praktis** untuk implementasi ClaimReview markup dalam skala besar atau jika Anda ingin mengotomatiskan proses *markup*. Plugin SEO seperti **Rank Math SEO Pro** menyediakan fitur untuk menambahkan *structured data* Schema dengan mudah, termasuk tipe ClaimReview.

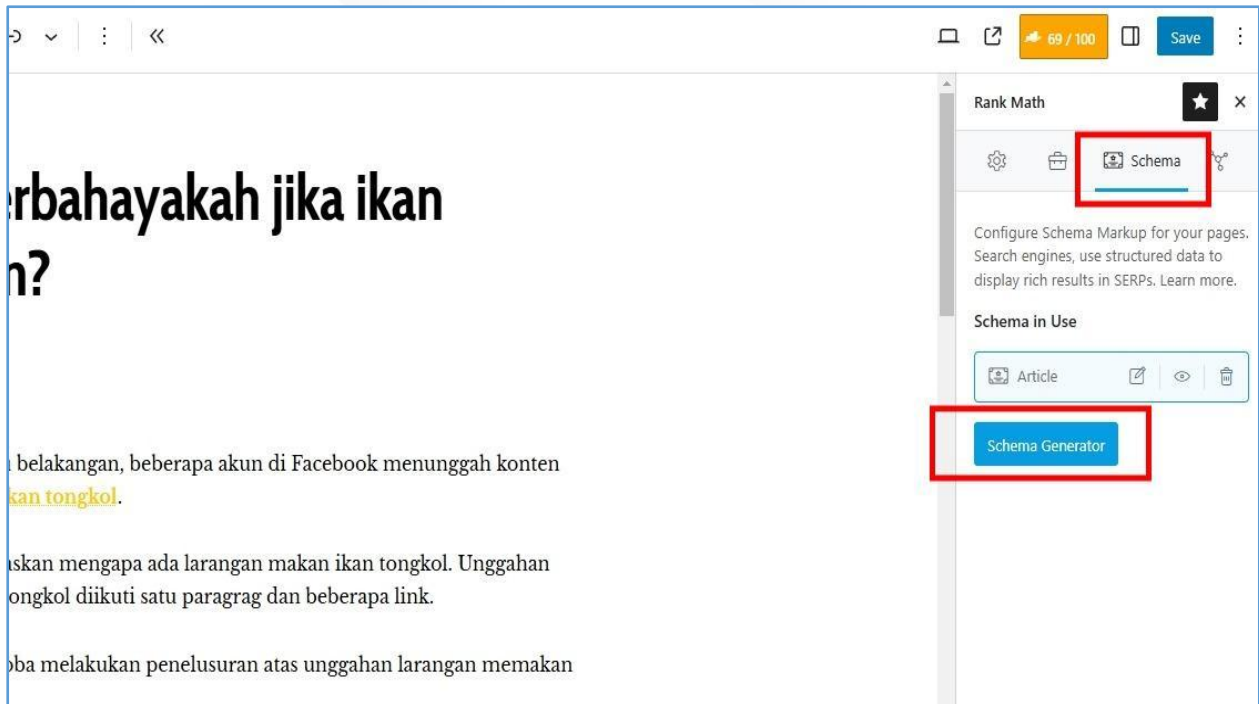
- **Persyaratan:**

- **Plugin Rank Math SEO Pro terinstal dan aktif:** Pastikan Anda telah menginstal dan mengaktifkan plugin Rank Math SEO Pro (versi *Pro* diperlukan untuk fitur *Custom Schema* yang fleksibel). Jika Anda menggunakan versi gratis Rank Math, fitur *Schema* mungkin terbatas.

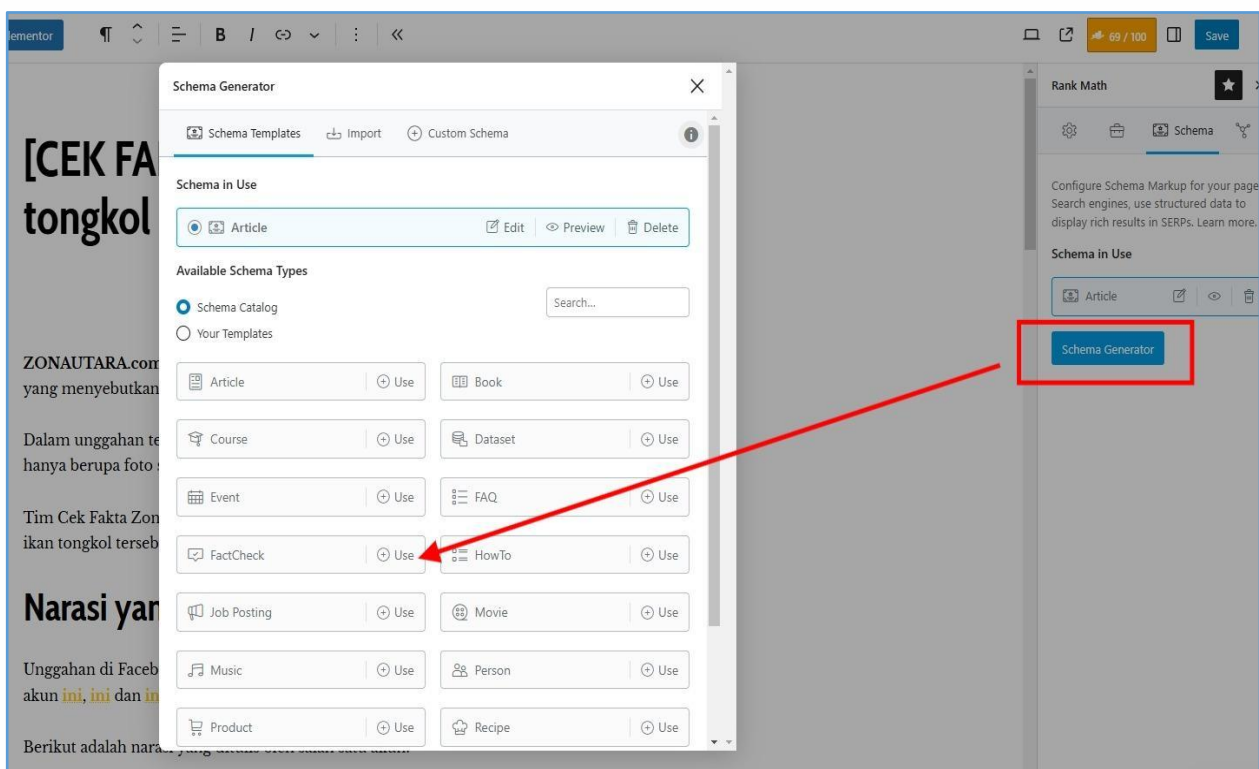
- **Langkah-Langkah Implementasi dengan Rank Math SEO Pro:**

0. **Buka Artikel di Editor WordPress:** Buka artikel cek fakta yang ingin Anda tambahkan ClaimReview markup di *editor* WordPress.
1. **Buka Meta Box Rank Math SEO:** Di bawah atau disamping *editor* artikel, cari meta box "**Rank Math SEO**".
2. **Buka Tab "Schema":** Di dalam meta box Rank Math SEO, klik *tab* "**Schema**".
3. **Pilih "Schema Generator":** Di *tab* "Schema", Anda akan melihat opsi "**Schema Generator**" atau "**Schema Markup**". Klik tombol "**Schema Generator**" atau "**Add Schema**".



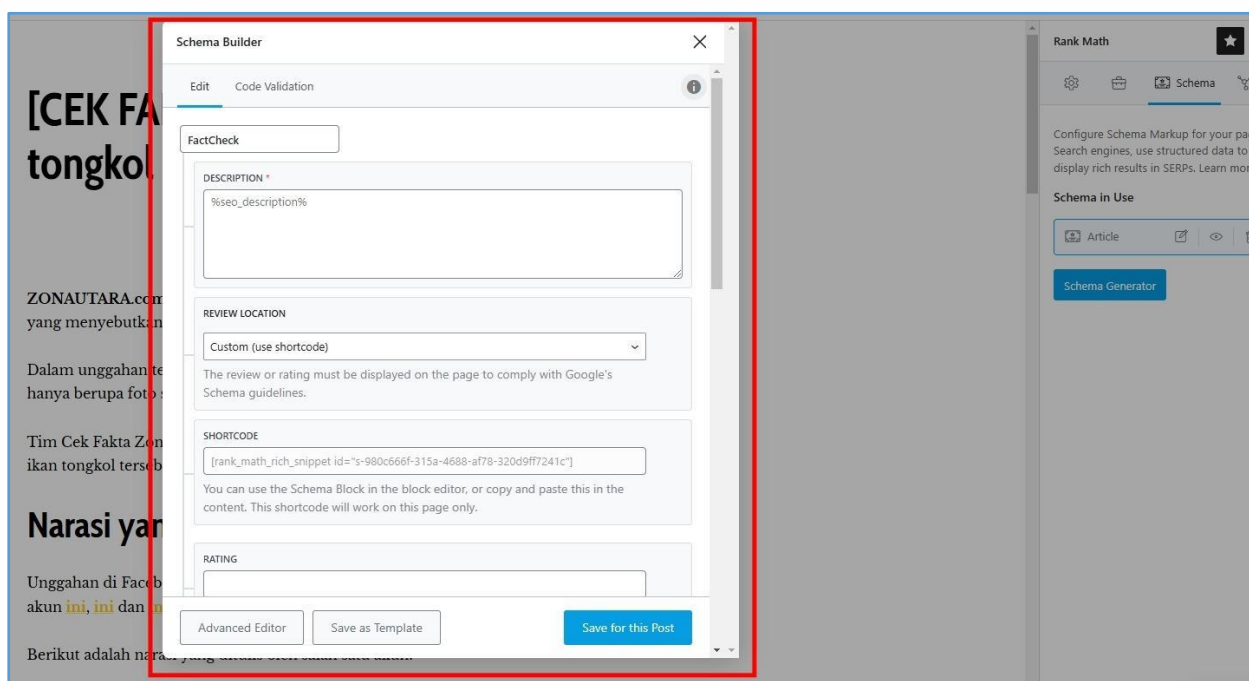


4. **Pilih Tipe Schema "FactCheck"**: Dalam jendela "Schema Generator" yang muncul, cari dan pilih tipe Schema "FactCheck" dari daftar yang tersedia.



5. **Isi Field Properti ClaimReview:** Setelah memilih "FactCheck", Rank Math SEO akan menampilkan **formulir dengan field properti-properti ClaimReview markup. Isi field ini dengan informasi yang sesuai dari artikel cek fakta Anda:**

- **Claim Reviewed:** Masukkan teks klaim yang Anda periksa.
- **Fact-Check Author Name:** Masukkan nama organisasi atau penulis cek fakta.
- **Fact-Check Author URL:** Masukkan URL website organisasi atau profil penulis.
- **Date Published:** Pilih tanggal publikasi artikel.
- **Item Reviewed - Appearance URL:** Masukkan URL sumber klaim.
- **Rating Value:** Pilih label *rating* atau nilai numerik *rating* dari *dropdown* atau *field* yang disediakan (sesuai sistem *rating* Anda).
- **Rating Explanation:** Masukkan penjelasan singkat *rating* Anda.
- **URL Artikel Cek Fakta (Canonical URL):** Field ini mungkin sudah terisi otomatis dengan URL artikel saat ini. Pastikan URL sudah benar.
- **Image URL:** Masukkan URL gambar *thumbnail* artikel Anda.

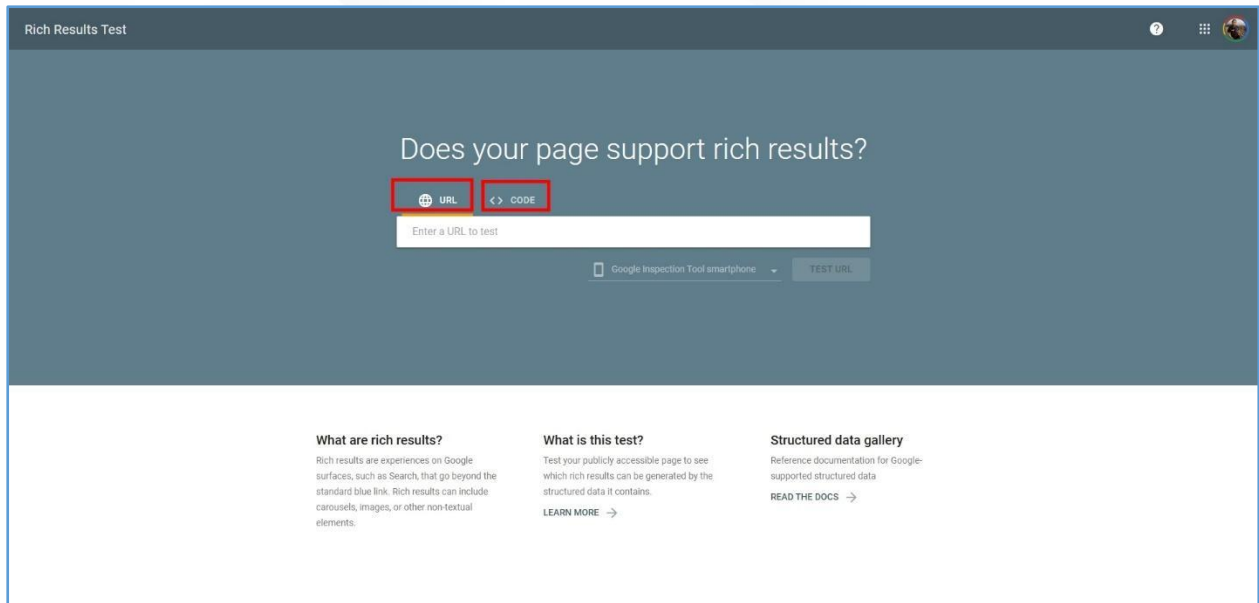


6. **Simpan dan Sisipkan Schema:** Setelah semua *field* terisi, klik tombol "Save for this Post" atau "Insert" (tergantung versi Rank Math). Rank Math SEO akan otomatis **menghasilkan *code snippet* JSON-LD ClaimReview markup dan menyisipkannya ke dalam *source code* halaman artikel Anda.**
 7. **Perbarui atau Publikasikan Artikel:** Klik "Update" atau "Publish" untuk menyimpan perubahan.
 8. **Validasi Markup (Langkah Selanjutnya):** Setelah publikasi, lakukan validasi *markup* menggunakan Google Rich Results Test (akan dibahas di sub-sesi 1.4).
- **Keuntungan Metode Plugin (Rank Math SEO Pro):**
 - **Sangat mudah dan cepat:** Proses implementasi *markup* menjadi sangat mudah dan cepat, cukup mengisi formulir.
 - **Tidak perlu *coding* HTML:** Tidak memerlukan pengetahuan *coding* HTML atau JSON-LD.
 - **Mengurangi risiko kesalahan *syntax*:** Plugin otomatis menghasilkan kode, mengurangi risiko kesalahan *syntax*.
 - **Terintegrasi dengan SEO:** Rank Math SEO adalah plugin SEO yang kuat, implementasi *structured data* menjadi bagian dari optimasi SEO keseluruhan.
 - **Kekurangan Metode Plugin (Rank Math SEO Pro):**
 - **Memerlukan plugin berbayar:** Rank Math SEO Pro adalah plugin berbayar. Versi gratis mungkin memiliki fitur *Schema* yang terbatas.
 - **Ketergantungan pada plugin:** Anda menjadi bergantung pada plugin. Jika plugin dinonaktifkan atau ada masalah dengan plugin, *markup* mungkin tidak berfungsi.
 - **Kurang fleksibel untuk kasus *custom* (mungkin):** Meskipun Rank Math SEO Pro cukup fleksibel, untuk kasus implementasi yang sangat *custom* atau kompleks, metode manual mungkin memberikan kontrol yang lebih besar.

Sub-sesi 1.5: Validasi Markup dengan Google Rich Results Test

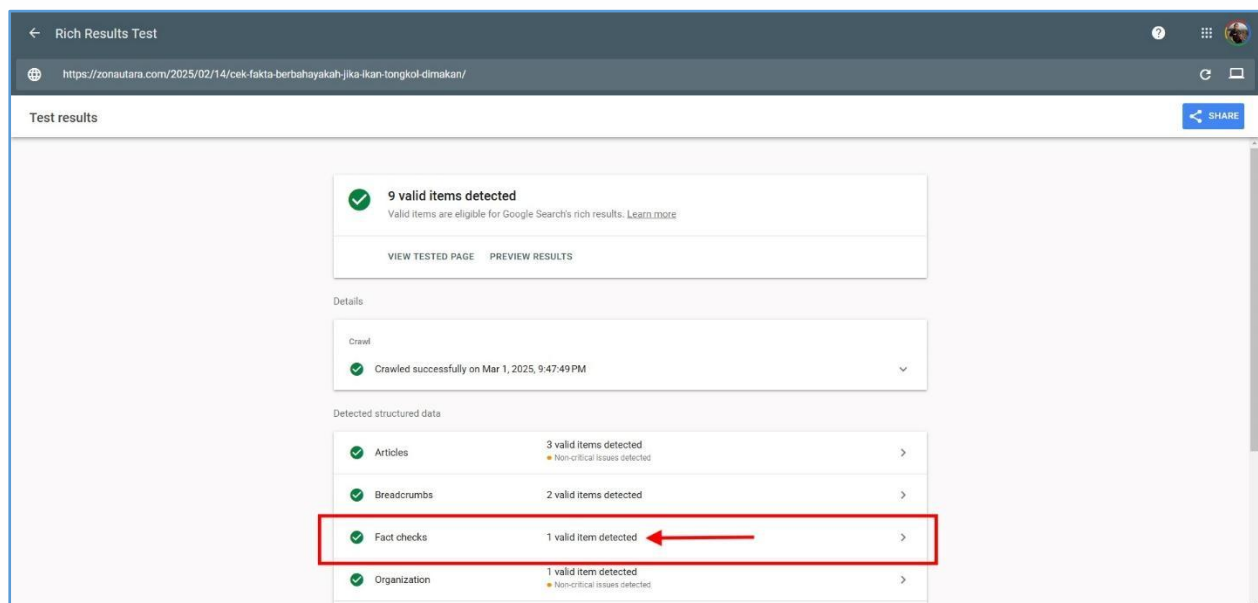
Setelah mengimplementasikan ClaimReview markup, langkah **sangat penting** selanjutnya adalah melakukan **validasi markup** menggunakan *tool* yang disediakan oleh Google, yaitu **Google Rich Results Test**. Validasi ini bertujuan untuk:

- **Memastikan markup valid:** Memeriksa apakah *code snippet* JSON-LD yang Anda implementasikan **valid secara syntax** dan **memenuhi persyaratan Google** untuk *rich results* dan ClaimReview markup.
- **Mendeteksi error dan warnings:** Jika ada kesalahan atau peringatan dalam *markup* Anda, *tool* validasi akan **memberitahukan secara detail**, sehingga Anda dapat memperbaikinya.
- **Memastikan memenuhi syarat untuk rich results:** Memastikan bahwa *markup* Anda **memenuhi syarat** untuk ditampilkan sebagai *rich results* (termasuk *fact check snippets*) di hasil pencarian Google.
- **Cara Menggunakan Google Rich Results Test:**
 1. **Buka Google Rich Results Test:** Buka *browser* web dan kunjungi URL: <https://search.google.com/test/rich-results>
 2. **Pilih Opsi "URL" atau "Code":** Anda memiliki dua opsi untuk melakukan validasi:
 - **"URL":** Jika artikel cek fakta Anda sudah dipublikasikan secara *online*, pilih opsi "URL" dan masukkan **URL artikel cek fakta** ke dalam kolom yang tersedia, lalu klik **"Test URL"**. Google Rich Results Test akan merayapi halaman tersebut dan memeriksa *structured data* yang ada.
 - **"Code":** Jika Anda ingin menguji *code snippet* JSON-LD sebelum mempublikasikan artikel atau jika Anda ingin menguji *code snippet* dari *template* atau *generator*, pilih opsi "Code". **Salin code snippet JSON-LD ClaimReview markup** Anda dan tempelkan ke dalam kolom "Code", lalu klik **"Test Code"**.



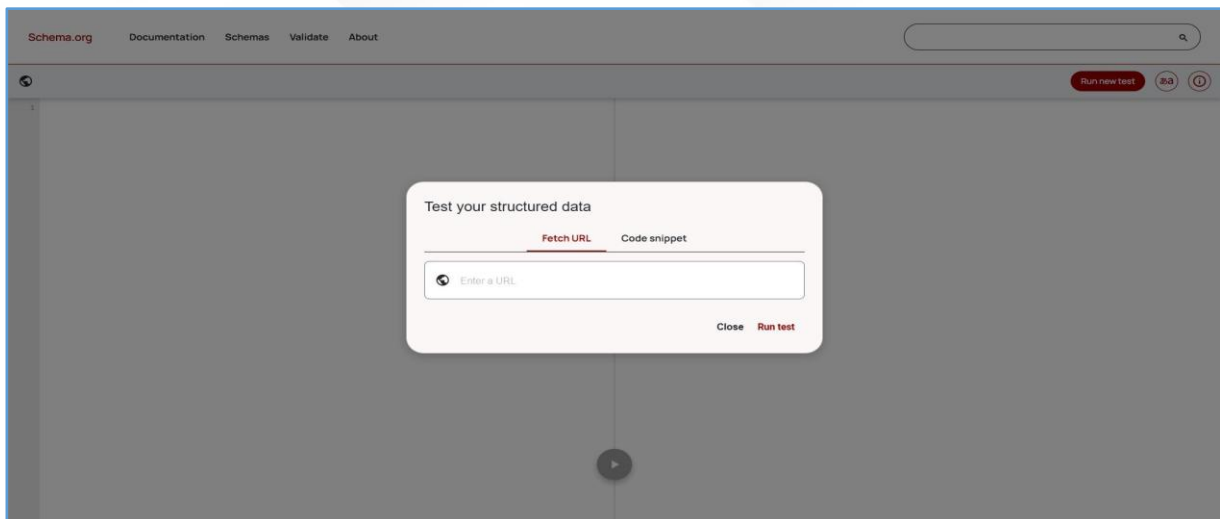
3. **Interpretasi Hasil Validasi:** Setelah pengujian selesai, Google Rich Results Test akan menampilkan hasil validasi. Perhatikan hal-hal berikut:

- **"Valid page" atau "Valid code":** Jika hasil validasi menunjukkan **"Valid page"** atau **"Valid code"**, ini berarti *markup* ClaimReview Anda **valid** dan **berhasil terdeteksi** oleh Google. Anda mungkin akan melihat tipe *rich result* yang terdeteksi, seperti "Fact Check".



- **"Invalid page" atau "Invalid code", atau "Errors"**: Jika hasil validasi menunjukkan **"Invalid page"** atau **"Invalid code"**, atau menampilkan **daftar error**, ini berarti ada **kesalahan dalam markup Anda**. Perhatikan detail *error* yang ditampilkan. *Error* biasanya menunjukkan kesalahan *syntax* JSON-LD (misalnya tanda kutip hilang, kurung kurawal tidak seimbang) atau properti wajib yang hilang atau tidak valid. **Perbaiki error yang ditemukan dan uji validasi ulang** hingga hasil validasi menjadi "Valid".
 - **"Warnings"**: Hasil validasi mungkin juga menampilkan **"Warnings"** (peringatan). *Warnings* biasanya menunjukkan **properti rekomendasi yang tidak ada** atau *markup* yang kurang optimal, meskipun secara teknis masih valid. *Warnings* **tidak akan mencegah rich results** ditampilkan, tetapi **sangat disarankan untuk memperhatikan dan memperbaiki warnings** untuk meningkatkan kualitas dan kelengkapan *markup* Anda. Klik *warnings* untuk melihat detail dan saran perbaikan.
4. **Perbaiki Error dan Warnings (Jika Ada)**: Jika hasil validasi menunjukkan *error* atau *warnings*, **kembali ke code snippet atau plugin Rank Math SEO Anda, perbaiki kesalahan yang terdeteksi berdasarkan pesan error dan saran perbaikan dari Google Rich Results Test**. Setelah memperbaiki, **uji validasi ulang** hingga hasilnya "Valid page" atau "Valid code" tanpa *error* dan *warnings* (atau minimal *warnings* yang dapat Anda abaikan setelah mempertimbangkan saran perbaikan).

TOOL TAMBAHAN: Selain Google Rich Result Test, anda juga dapat menggunakan tool validator dari Schema.org. Silahkan akses di sini: <https://validator.schema.org/>. Cara kerjanya mirip dengan Google Rich Result Test, anda dapat memasukkan url konten cek fakta yang sudah dipublish, atau memasukkan code snippet JSO-LD untuk memvalidasi sebelum dipublish.



Sub-sesi 1.6: Publikasi dan Monitoring

Publikasikan Artikel Cek Fakta yang Sudah Di-*Markup*

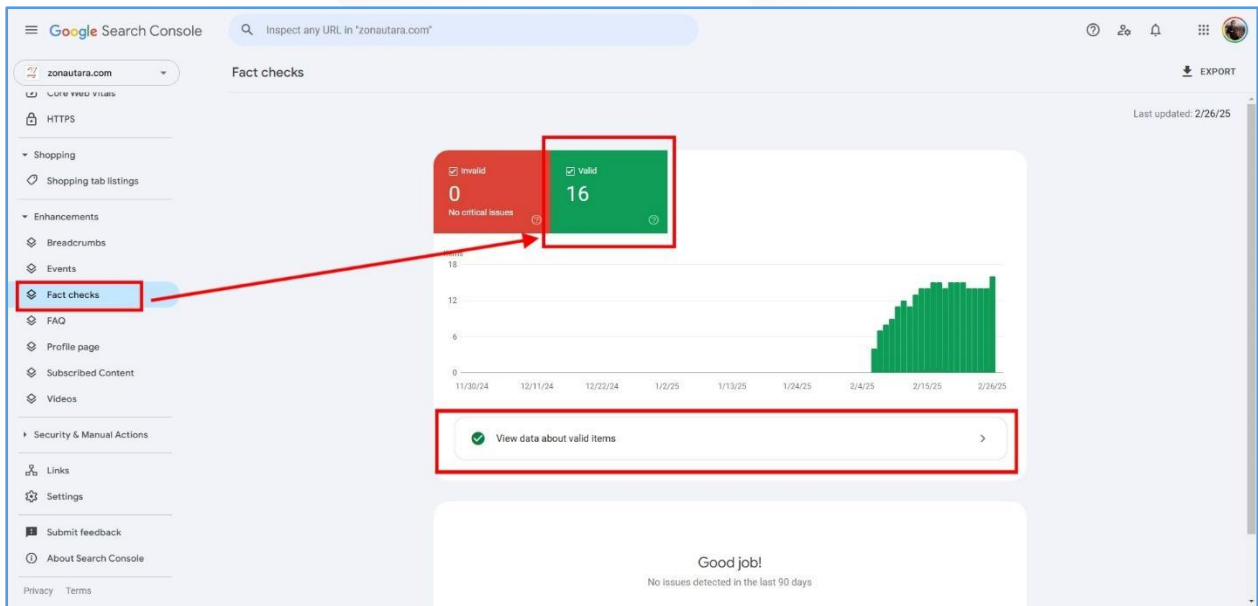
Setelah Anda yakin *markup* ClaimReview Anda valid dan tidak ada *error* atau *warnings* signifikan, langkah terakhir adalah **mempublikasikan atau memperbarui artikel cek fakta Anda di website Anda**. Pastikan artikel tersebut sudah tayang secara *online* dan dapat diakses publik.

Monitoring Implementasi dan Performa

Setelah publikasi, penting untuk **memantau implementasi ClaimReview markup dan memeriksa performa konten cek fakta Anda** di Google Search dan Google Fact Check Explorer. *Tools* yang dapat Anda gunakan untuk monitoring:

- **Google Search Console:**

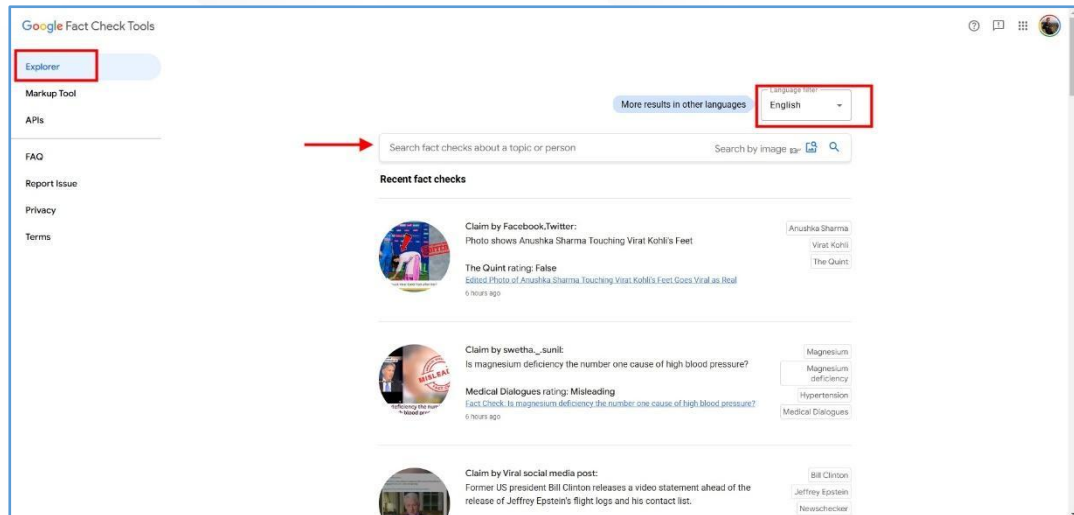
Laporan *Rich Results Status*: Di Google Search Console, periksa laporan "**Rich Results Status**" (biasanya ada di bagian "Enhancements" atau "Peningkatan"). Laporan ini akan menunjukkan **jumlah halaman valid yang terdeteksi memiliki *rich results* ClaimReview**, serta *error* atau *warnings* jika ada masalah dengan implementasi *markup* di seluruh website Anda. Laporan ini memberikan gambaran *overall* tentang status implementasi *markup* Anda.



- **Google Fact Check Explorer:**

Cari artikel Anda di Google Fact Check Explorer:

Kunjungi <https://toolbox.google.com/factcheck/explorer/> dan **cari artikel cek fakta Anda** menggunakan judul artikel atau klaim yang diperiksa sebagai kata kunci pencarian. Jika implementasi ClaimReview markup Anda berhasil, **artikel Anda seharusnya muncul** di hasil pencarian Google Fact Check Explorer. Perlu diingat bahwa mungkin perlu waktu beberapa saat (beberapa jam hingga beberapa hari) setelah publikasi agar konten Anda terindeks dan muncul di Google Fact Check Explorer.



Catatan Penting:

- **Waktu Indeksasi dan Tampilan:** Perlu diingat bahwa **Google membutuhkan waktu untuk merayapi, mengindeks, dan memproses markup ClaimReview** yang Anda implementasikan. Jangan berharap konten Anda langsung muncul di Google Fact Check Explorer atau *rich results* segera setelah publikasi. Proses ini bisa memakan waktu beberapa jam hingga beberapa hari.
- **Kesabaran dan Monitoring Berkelanjutan:** Implementasi *structured data* adalah proses berkelanjutan. **Lakukan monitoring secara berkala** untuk memastikan *markup* tetap valid, tidak ada *error* baru, dan periksa apakah konten Anda tetap terindeks dan ditampilkan dengan baik di Google Fact Check Explorer dan *rich results*. Jika ada masalah, segera perbaiki dan validasi ulang *markup* Anda.

Sesi 2: *Tools* untuk Memudahkan Implementasi dan Validasi ClaimReview Markup

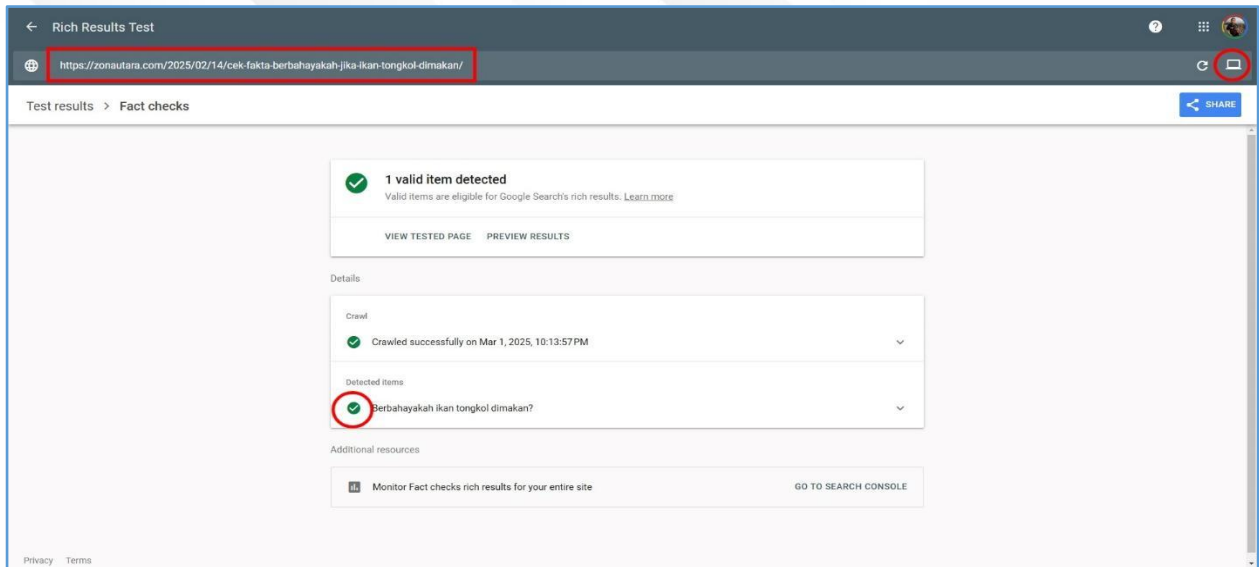
Sesi ini akan membahas lebih detail *tools* yang telah kita sebutkan sebelumnya, yang akan sangat membantu Anda dalam proses implementasi dan validasi ClaimReview markup.

Sub-sesi 2.1: Google Rich Results Test: Uji Validasi Markup

Kita telah memperkenalkan Google Rich Results Test (sebelumnya dikenal sebagai *Structured Data Testing Tool*) di sub-sesi 1.4. *Tool* ini adalah *tool wajib* yang disediakan oleh Google untuk **memvalidasi *structured data***, termasuk ClaimReview markup. Ini adalah *tool* pertama dan terpenting yang harus Anda gunakan setiap kali Anda mengimplementasikan atau memodifikasi *markup*.

- **Fungsi Utama Google Rich Results Test:**
 - **Validasi *Syntax* dan Struktur *Markup*:** Memastikan *code snippet* JSON-LD Anda **benar secara *syntax*** (tidak ada kesalahan penulisan, tanda kutip, kurung kurawal, dll.) dan **sesuai dengan struktur yang diharapkan** oleh Google untuk tipe *rich result* yang Anda targetkan (dalam hal ini, ClaimReview).
 - **Deteksi *Error* dan *Warnings*:** Mengidentifikasi **kesalahan (*errors*)** yang akan mencegah *rich results* ditampilkan, dan **peringatan (*warnings*)** yang menunjukkan potensi masalah atau area untuk perbaikan *markup*.
 - **Pratinjau Potensi *Rich Results*:** Menunjukkan **pratinjau bagaimana *rich results* Anda berpotensi tampil** di hasil pencarian Google jika *markup* valid dan memenuhi syarat. Untuk ClaimReview, pratinjau *rich results* mungkin tidak selalu ditampilkan secara visual di *tool* ini, tetapi validasi yang berhasil tetap menunjukkan potensi untuk *fact check snippet*.
 - **Uji Berdasarkan URL atau *Code Snippet*:** Memungkinkan validasi berdasarkan **URL halaman web yang sudah *live*** atau langsung dari *code snippet* JSON-LD yang Anda buat.
- **Cara Mengakses dan Menggunakan Google Rich Results Test (Ringkasan):**
 1. **Buka URL:** Kunjungi di *browser* web.
 2. **Pilih Opsi Uji:** Pilih "**URL**" (untuk menguji halaman *live*) atau "**Code**" (untuk menguji *code snippet*).
 3. **Masukkan URL atau Tempelkan Kode:** Masukkan URL artikel cek fakta atau tempelkan *code snippet* JSON-LD ClaimReview markup Anda.

4. **Klik "Test URL" atau "Test Code":** Mulai proses validasi.
 5. **Interpretasi Hasil:** Periksa hasil validasi, perhatikan pesan "**Valid page**"/"**Valid code**", *error*, dan *warnings*.
 6. **Perbaiki Kesalahan dan Validasi Ulang:** Jika ada *error* atau *warnings*, perbaiki *markup* dan validasi ulang hingga hasilnya valid.
- **Tips Menggunakan Google Rich Results Test:**
 - **Validasi Secara Berkala:** Validasi *markup* setiap kali Anda membuat perubahan atau modifikasi. Ini akan membantu Anda mendeteksi dan memperbaiki kesalahan sejak dini.
 - **Perhatikan Detail *Error* dan *Warnings*:** Baca pesan *error* dan *warnings* dengan seksama. Google Rich Results Test biasanya memberikan informasi yang cukup jelas tentang letak kesalahan dan properti yang bermasalah. Klik pada *error* atau *warning* untuk melihat detail lebih lanjut.
 - **Gunakan Opsi "Code" untuk Pengembangan Awal:** Saat Anda sedang membuat *markup* atau bereksperimen dengan berbagai properti, gunakan opsi "**Code**" untuk validasi *code snippet* secara langsung sebelum diimplementasikan di website. Ini memungkinkan Anda melakukan *iterasi* dan perbaikan *markup* dengan cepat tanpa harus mempublikasikan setiap perubahan ke website.
 - **Uji "URL" Setelah Publikasi:** Setelah *markup* diimplementasikan di website dan artikel sudah *live*, selalu uji dengan opsi "**URL**" untuk memastikan *markup* terimplementasi dengan benar di halaman *live* dan dapat dideteksi oleh Google.
 - **Referensi ke Dokumentasi Schema.org:** Jika Anda mengalami kesulitan memahami *error* atau *warnings* terkait properti Schema.org, merujuklah ke dokumentasi resmi Schema.org (<https://schema.org/ClaimReview>) untuk memastikan Anda menggunakan properti dengan benar dan sesuai dengan spesifikasi.



Sub-sesi 2.2: Google Search Console: Monitoring Status *Rich Results*

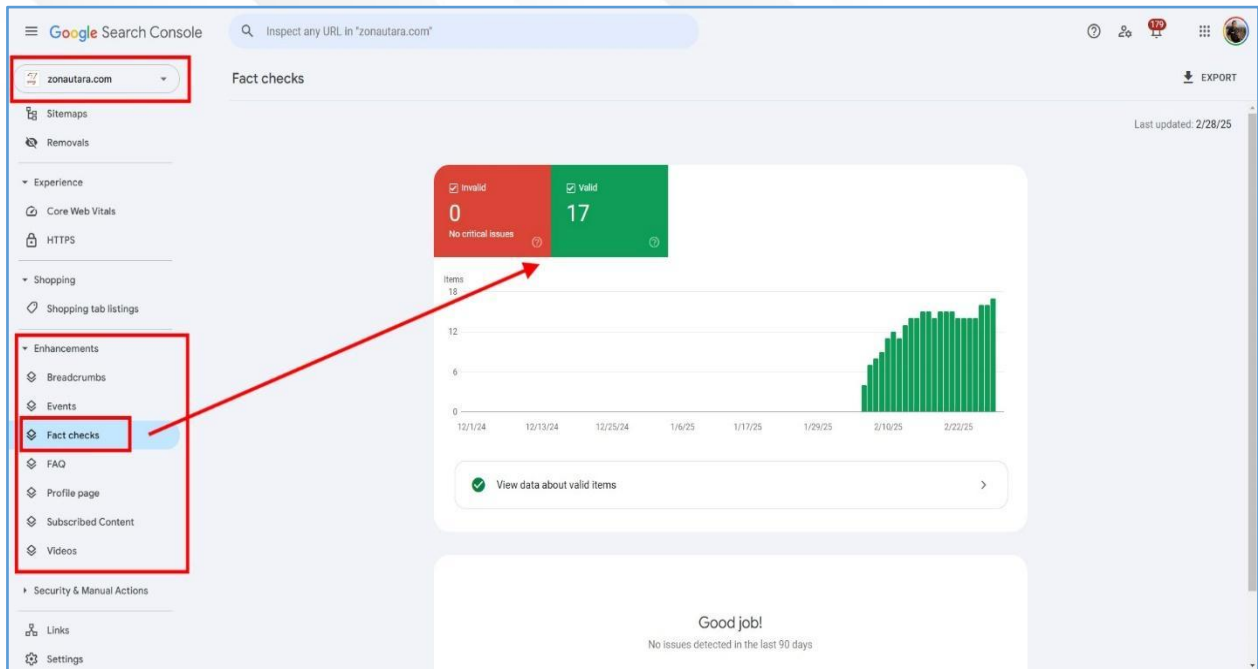
Google Search Console: Pantau Implementasi *Rich Results* Secara Keseluruhan

Google Search Console adalah *tool* gratis dan sangat penting dari Google untuk *webmaster* dan pemilik website. Selain untuk memantau performa pencarian website Anda secara umum, Google Search Console juga menyediakan **laporan khusus untuk memantau status *rich results***, termasuk ClaimReview markup.

- **Fungsi Utama Google Search Console untuk *Rich Results*:**
 - ***Rich Results Status Reports***: Google Search Console menyediakan **laporan status *rich results*** untuk berbagai tipe *rich results* yang terdeteksi di website Anda, termasuk laporan khusus untuk "**Fact Check**" (yang merepresentasikan ClaimReview markup). Laporan ini memberikan informasi *agregat* tentang implementasi *rich results* Anda secara keseluruhan di website.
 - **Deteksi *Error* dan *Warnings* di Skala Website**: Laporan *status rich results* di Search Console **mendeteksi *error* dan *warnings* terkait *structured data* di seluruh website Anda**, bukan hanya per halaman seperti Google Rich Results Test. Jika ada masalah *markup* yang terjadi secara konsisten di banyak halaman, laporan ini akan membantu Anda mengidentifikasinya.
 - **Validasi Perbaikan Massal (Setelah Memperbaiki *Markup*)**: Setelah Anda memperbaiki *error markup* yang terdeteksi oleh Search Console (misalnya, *error* yang terjadi di banyak halaman), Anda dapat menggunakan Search Console untuk **meminta Google**

untuk *re-crawl* dan *re-index* halaman-halaman yang diperbarui setelah perbaikan. Ini mempercepat proses Google mengenali *markup* yang sudah diperbaiki.

- **Performa Rich Results (Khusus untuk ClaimReview):** Untuk beberapa tipe *rich results* (seperti *Product*, *Recipe*, *Event*), Search Console juga menyediakan laporan performa yang lebih detail (misalnya, *impressions*, *clicks*, *CTR*) terkait *rich results* tersebut. Namun, untuk **ClaimReview**, **laporan performa yang spesifik mungkin belum tersedia secara langsung** di Search Console saat ini. Meskipun demikian, laporan *status rich results* tetap sangat berguna untuk memantau implementasi ClaimReview secara keseluruhan.
- **Cara Mengakses Laporan Rich Results Status di Google Search Console:**
 1. **Buka Google Search Console:** Login ke akun Google Search Console Anda (pastikan Anda sudah memverifikasi kepemilikan website Anda di Search Console). Kunjungi <https://search.google.com/search-console/>.
 2. **Pilih Properti Website:** Pilih properti website (domain website) yang ingin Anda periksa.
 3. **Navigasi ke Laporan Rich Results Status:** Di menu navigasi sebelah kiri, cari bagian "**Enhancements**" (Peningkatan) atau "**Experience**" (Pengalaman) (nama bagian dan lokasi menu mungkin sedikit berbeda tergantung *update* Search Console terbaru). Di dalam bagian tersebut, Anda akan menemukan daftar laporan *status rich results* untuk berbagai tipe *markup* yang terdeteksi oleh Google. Cari dan klik laporan "**Fact Check**".



4. Interpretasi Laporan: Laporan *status rich results* "Fact Check" akan menampilkan informasi berikut:

- **Grafik Tren Validitas:** Grafik yang menunjukkan tren validitas *rich results* ClaimReview di website Anda seiring waktu (misalnya, jumlah halaman valid, halaman dengan *error*, halaman dengan *warnings*).
- **Daftar *Error* dan *Warnings*:** Daftar *error* dan *warnings* yang terdeteksi di *markup* ClaimReview di seluruh website Anda. Klik pada *error* atau *warning* untuk melihat detail lebih lanjut dan contoh halaman yang terpengaruh.
- **Contoh Halaman dengan *Error* dan *Warnings*:** Daftar contoh URL halaman yang memiliki *error* atau *warnings*. Klik pada URL untuk melihat detail masalah pada halaman tersebut dan *code snippet* yang bermasalah.
- **Opsi "Validate Fix" (Validasi Perbaikan):** Setelah Anda memperbaiki *error markup* berdasarkan laporan Search Console, Anda dapat menggunakan opsi "Validate Fix" untuk meminta Google *re-crawl* dan *re-index* halaman-halaman yang diperbaiki. Google akan memprioritaskan *crawling* ulang halaman-halaman yang Anda validasi perbaikannya.

- **Tips Menggunakan Google Search Console untuk Monitoring *Rich Results*:**
 - **Periksa Laporan Secara Berkala: Periksa laporan *status rich results* "Fact Check" secara berkala** (misalnya, mingguan atau bulanan) untuk memantau kesehatan implementasi *markup* Anda secara keseluruhan dan mendeteksi potensi masalah sejak dini.
 - **Prioritaskan Perbaikan *Error*: Fokus utama adalah memperbaiki *errors* (kesalahan)** yang terdeteksi di laporan Search Console. *Error* akan mencegah *rich results* ditampilkan. Gunakan informasi *error* dari Search Console dan Google Rich Results Test untuk membantu Anda memperbaiki *markup*.
 - **Perhatikan *Warnings* (Sebagai Peningkatan):** Setelah *error* diperbaiki, perhatikan **peringatan** (*warnings*). *Warnings* menunjukkan area untuk **potensi peningkatan *markup***. Memperbaiki *warnings* dapat meningkatkan kualitas dan kelengkapan *rich results* Anda, meskipun *warnings* tidak selalu mencegah *rich results* ditampilkan.
 - **Gunakan "Validate Fix" Setelah Perbaikan Massal:** Setelah Anda melakukan perbaikan *markup* secara massal (misalnya, memperbaiki *template* website yang menghasilkan *markup* bermasalah di banyak halaman), **gunakan opsi "Validate Fix"** di Search Console untuk mempercepat proses Google mengenali perbaikan Anda.
 - **Kombinasikan dengan Google Rich Results Test:** Gunakan Google Search Console untuk **monitoring *overall*** dan deteksi masalah di skala website. Gunakan Google Rich Results Test untuk **validasi detail dan *debug markup* per halaman atau *code snippet***. Kedua *tool* ini saling melengkapi.

Sub-sesi 2.3: Schema Markup Generator (Pihak Ketiga)

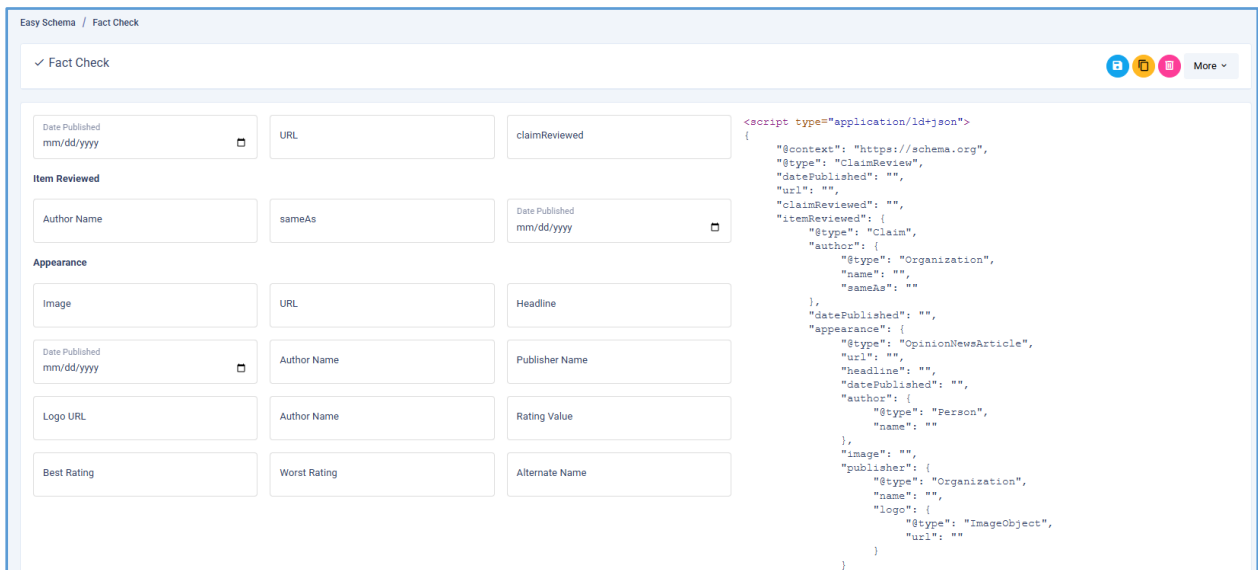
Schema Markup Generator: Mempercepat Pembuatan *Markup* (Hati-Hati Memilih *Tool*)

Seperti yang telah dibahas di sub-sesi 1.2, *Schema Markup Generator online* (pihak ketiga) dapat menjadi *tool* yang berguna untuk **mempercepat proses**

pembuatan *code snippet* JSON-LD, terutama jika Anda kurang familiar dengan *coding* JSON-LD atau ingin membuat *markup* dengan cepat.

- **Keuntungan Menggunakan *Schema Markup Generator*** (sudah dibahas di sub-sesi 1.2):
 - Mempermudah proses pembuatan *markup*.
 - Memandu pengisian properti.
- **Keterbatasan dan Peringatan Memilih *Generator*** (sudah dibahas di sub-sesi 1.2, dan perlu dipertegas):
 - Mungkin kurang fleksibel untuk kasus kompleks.
 - **Perlu diperiksa ulang:** *Output generator* harus selalu divalidasi dengan Google Rich Results Test.
 - **Kualitas dan Keandalan *Generator* Bervariasi:** Tidak semua *Schema Markup Generator* sama. Kualitas dan keandalan *generator* dari pihak ketiga bisa bervariasi. Beberapa *generator* mungkin menghasilkan kode yang kurang optimal, tidak lengkap, atau bahkan tidak valid.
- **Rekomendasi Memilih *Schema Markup Generator*:**
 - **Pilih *Generator* yang Terpercaya dan Populer:** Pilih *generator* yang direkomendasikan oleh komunitas SEO atau *web development*, dan memiliki *review* positif. Contoh *generator* yang cukup bagus dan direkomendasikan adalah **EasySchema Fact Check**.
 - **Periksa Fitur Dukungan Tipe "ClaimReview":** Pastikan *generator* yang Anda pilih mendukung tipe *Schema* "ClaimReview" atau "FactCheck". Tidak semua *generator* mendukung semua tipe *Schema*.
 - **Uji Coba dan Bandingkan Hasil:** Coba beberapa *generator* yang berbeda dan bandingkan *output code snippet* JSON-LD yang dihasilkan. Periksa apakah *code snippet* terlihat lengkap, sesuai dengan *template* Schema.org, dan valid saat divalidasi dengan Google Rich Results Test.

- **Jangan Terlalu Bergantung pada Generator:** *Schema Markup Generator* adalah *tool* pembantu, bukan pengganti pemahaman tentang *structured data* dan ClaimReview markup. **Gunakan generator sebagai titik awal**, tetapi **tetap pelajari dan pahami struktur markup**, dan selalu **validasi hasilnya** serta **sesuaikan jika diperlukan** untuk memastikan kualitas dan keakuratan *markup* Anda.



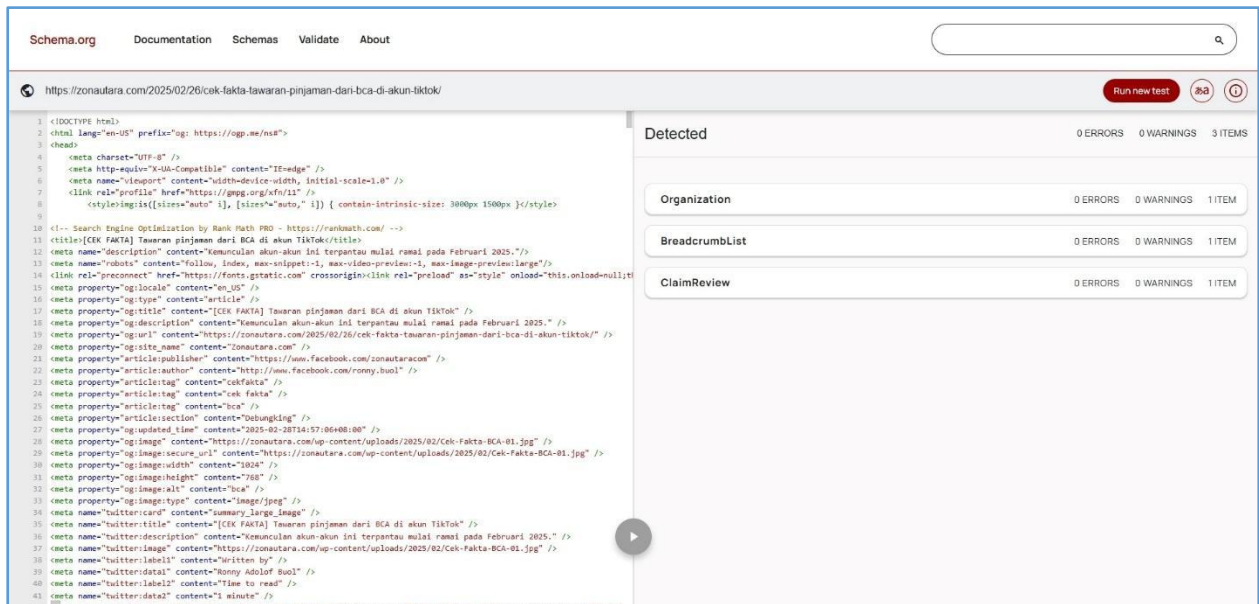
Sub-sesi 2.4: Fact Check Markup Validator

Saat ini, **tidak ada *tool* validasi *official* dari Google yang secara khusus ditujukan hanya untuk ClaimReview markup**, selain dari Google Rich Results Test yang bersifat umum untuk semua tipe *rich results*.

Beberapa *website* pihak ketiga atau *developer* mungkin pernah atau sedang mengembangkan *tool* validasi *online* yang lebih spesifik untuk ClaimReview markup. Jika Anda menemukan *tool* seperti itu, Anda dapat mencobanya, tetapi **selalu prioritaskan validasi dengan Google Rich Results Test** sebagai *tool* validasi utama dan terpercaya, karena *tool* ini adalah *tool* resmi dari Google dan paling akurat dalam mencerminkan bagaimana Google memproses *structured data*.

Anda juga dapat menggunakan *tool* validator dari Schema.org, yang dapat diakses pada link ini: <https://validator.schema.org/>

Jika di masa depan Google atau pihak ketiga merilis *tool* validasi yang lebih spesifik untuk ClaimReview markup, informasi tentang *tool* tersebut akan ditambahkan di sini. Untuk saat ini, **Google Rich Results Test** adalah *tool* validasi utama yang direkomendasikan untuk ClaimReview markup.



Contoh hasil uji dengan Validator Schema.org

Sesi 3: Praktik Implementasi Langsung (Sesi Praktik)

Sesi ini adalah sesi praktik *hands-on* di mana Anda akan mempraktikkan langsung implementasi ClaimReview markup pada contoh artikel cek fakta.

Tujuan Sesi Praktik:

Sesi praktik ini bertujuan untuk memberikan **pengalaman *hands-on*** kepada peserta dalam mengimplementasikan ClaimReview markup. Dengan praktik langsung, diharapkan peserta akan lebih **memahami alur kerja implementasi**, **mengatasi potensi kesulitan**, dan **memvalidasi pemahaman** mereka tentang properti-properti ClaimReview markup dan *tools* validasi.

Langkah-Langkah Sesi Praktik:

1. Studi Kasus: Artikel Cek Fakta Contoh

- **Trainer Menyediakan Contoh Artikel Cek Fakta:** Sediakan contoh artikel cek fakta (misalnya, dalam format *print out* atau *link* ke artikel *online*). Artikel ini akan menjadi studi kasus untuk sesi praktik.
- **Contoh Artikel (Ilustrasi):** Misalkan artikel contoh berjudul: "Cek Fakta: Benarkah Minum Air Rebusan Daun Sirsak Bisa Menyembuhkan Kanker?". Artikel ini berisi pemeriksaan fakta terhadap klaim tentang khasiat air rebusan daun sirsak untuk menyembuhkan kanker, dengan kesimpulan bahwa klaim tersebut "**Salah** atau **Tidak Berbasis Bukti Ilmiah**" (atau *rating* sejenisnya).

2. Identifikasi Elemen-Elemen ClaimReview dalam Artikel

- **Instruksi:** Secara individu atau berkelompok, **baca artikel contoh** dengan seksama dan **identifikasi elemen-elemen berikut** yang diperlukan untuk ClaimReview markup:
 - **Klaim yang Diperiksa (*claimReviewed*):** Tuliskan teks klaimnya.
 - **Organisasi/Penulis Cek Fakta (*author*):** Identifikasi nama organisasi/penulis dan URL website/profil (jika ada di artikel).
 - **Tanggal Publikasi (*datePublished*):** Cari tanggal publikasi artikel.
 - **Tempat Klaim Muncul (*appearance*):** Identifikasi URL sumber klaim yang diperiksa (jika disebutkan dalam artikel). Jika ada, identifikasi juga tempat pertama kali klaim muncul (*firstAppearance*).
 - **Hasil Rating dan Penjelasan (*reviewRating*):** Tentukan *rating* yang diberikan artikel dan ringkasan penjelasannya.
 - **URL Artikel Cek Fakta (*url*):** URL artikel contoh.
 - **Gambar Thumbnail (*image*):** Pilih gambar *thumbnail* yang relevan untuk artikel (bisa gambar ilustrasi, *screenshot* dari artikel, atau logo media).

- **Diskusi Hasil Identifikasi:** Setelah peserta selesai mengidentifikasi elemen-elemen, lakukan **diskusi bersama** untuk membandingkan hasil identifikasi antar peserta. Bahas jika ada perbedaan interpretasi atau elemen yang terlewat.

3. Buat *Code Snippet* JSON-LD ClaimReview Markup

- **Instruksi:** Berdasarkan elemen-elemen yang telah diidentifikasi, **buat *code snippet* JSON-LD ClaimReview markup lengkap** menggunakan:
 - **Opsi 1: Metode Manual:** Gunakan *template code snippet* yang telah dipelajari di Modul sebelumnya, dan **isi nilai properti secara manual** dengan informasi yang telah diidentifikasi dari artikel contoh. Gunakan teks editor untuk mengedit kode.
 - **Opsi 2: Plugin Rank Math SEO Pro (Jika Tersedia):** Jika peserta memiliki akses ke website WordPress dengan plugin Rank Math SEO Pro terinstal, **gunakan plugin Rank Math SEO Pro** untuk membuat ClaimReview markup dengan mengisi formulir *Schema Generator*.
- **Pilihan Sistem Rating:** Peserta dapat memilih untuk menggunakan sistem *rating label teks* (misalnya, “salah” atau "Tidak Berbasis Bukti Ilmiah") atau **skala numerik** (misalnya, skala 1-5 dengan "1=Salah atau 1=Tidak Berbasis Bukti Ilmiah"). Pastikan sistem *rating* yang dipilih **konsisten** dengan kesimpulan artikel contoh.

4. Implementasikan *Code Snippet* di WordPress (Simulasi)

- **Simulasi Implementasi di WordPress:** Karena ini sesi praktik, implementasi di WordPress dapat disimulasikan, tanpa harus benar-benar diimplementasikan di website *live*:
 - **Simulasi Manual:** Jika menggunakan metode manual, **simulasikan penyisipan *code snippet* JSON-LD** ke dalam *code editor* HTML artikel WordPress (seperti yang ditunjukkan di sub-sesi 1.3). Dapat memilih dalam mode "Text" atau "Code Editor" WordPress.
 - **Simulasi Plugin Rank Math SEO Pro:** Jika menggunakan plugin Rank Math SEO Pro, **simulasikan pengisian**

formulir *Schema Generator* di meta box Rank Math SEO di halaman *editor* artikel WordPress (seperti yang ditunjukkan di sub-sesi 1.3).

5. Validasi *Code Snippet* dengan Google Rich Results Test

- **Instruksi: Validasi *code snippet* JSON-LD yang telah dibuat** menggunakan **Google Rich Results Test** (opsi "Code").
- **Periksa Hasil Validasi:** Interpretasi hasil validasi. Pastikan **tidak ada *error*** dan **perhatikan *warnings* (jika ada)**. Jika ada *error*, **perbaiki *code snippet*** berdasarkan pesan *error* dan validasi ulang hingga valid.

6. Diskusi dan *Troubleshooting*

- **Diskusi Hasil Validasi:** Bahas hasil validasi bersama-sama. Bandingkan *code snippet* yang dihasilkan oleh peserta. Diskusi jika ada perbedaan, *error*, atau *warnings* yang dialami peserta.
- ***Troubleshooting* dan Perbaikan:** Jika ada peserta yang mengalami *error* atau *warnings* saat validasi, bantu mereka untuk mengidentifikasi penyebab kesalahan dan cara memperbaikinya. *Trainer* dapat memberikan panduan dan *tips troubleshooting*.



Catatan untuk *Trainer*:

- **Siapkan Contoh Artikel Cek Fakta yang Relevan:** Pilih contoh artikel cek fakta yang **representatif** dan memiliki elemen-elemen yang jelas untuk di-*markup*.
- **Sediakan *Template* dan Contoh *Code Snippet*:** Sediakan *template code snippet* JSON-LD dan contoh kode lengkap dari Modul 2 sebagai referensi bagi peserta. Simpan dalam bentuk file.txt dan bagikan ke peserta pelatihan.
- **Pandu Peserta Langkah demi Langkah:** Pandu peserta langkah demi langkah selama sesi praktik, berikan instruksi yang jelas, dan berikan bantuan jika peserta mengalami kesulitan.
- **Tekankan Pentingnya Validasi:** Tekankan **pentingnya validasi *markup* dengan Google Rich Results Test** sebagai langkah *wajib* setelah implementasi.
- **Fasilitasi Diskusi dan *Sharing* Pengalaman:** Dorong peserta untuk berdiskusi, berbagi pengalaman, dan saling membantu dalam proses praktik dan *troubleshooting*.

Dengan sesi praktik ini, Modul tentang implementasi praktis ClaimReview markup diharapkan dapat memberikan bekal yang cukup bagi peserta untuk mulai menerapkan *markup* ini pada konten cek fakta mereka sendiri.

Penutup

Selamat, Anda telah menyelesaikan Modul 3 tentang implementasi praktis ClaimReview markup! Dalam modul ini, Anda telah mempelajari langkah-langkah implementasi, berbagai metode (manual dan plugin), *tools* validasi, dan bahkan mempraktikkannya langsung melalui sesi praktik.

Dengan bekal pengetahuan dan keterampilan modul ini dan modul sebelumnya, Anda sekarang memiliki pemahaman yang kuat tentang ClaimReview markup dan siap untuk mengimplementasikannya pada konten cek fakta Anda di website, terutama yang dibuat dengan WordPress.

Selanjutnya, Anda dapat terus memantau performa implementasi *markup* Anda menggunakan Google Search Console dan terus meningkatkan kualitas konten cek fakta Anda agar semakin bermanfaat bagi audiens dan semakin optimal dalam visibilitas di platform Google.

Selamat menerapkan ClaimReview markup dan berkontribusi dalam memerangi disinformasi secara *online*!





ClaimReview:

Panduan Media untuk
Kredibilitas Informasi

