Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 21-28
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2048

TEKNIK *CODING HYBRID* ANTARA *TYPESCRIPT* DAN *JAVASCRIPT* DALAM *FRAMEWORK ANGULAR*

Tony Wijaya[1)]
[1)] *Sistem Informasi, STMIK Pontianak*

email: *tony_wijaya@stmikpontianak.ac.id*[1)]

INFO ARTIKEL

Riwayat Artikel:

Diterima April, 2025
Revisi April, 2025
Terbit Mei, 2025

ABSTRAK

TypeScript merupakan penyempurnaan dari bahasa pemrograman *JavaScript*. Namun keunggulannya tidak serta merta membuat kehadiran JavaScript tidak dibutuhkan lagi. Hal ini dikarenakan kerumitan bahasa pemrograman *TypeScript* yang masih relatif lebih tinggi dibandingkan dengan *JavaScript*. Pengembang perangkat lunak *platform web* yang sudah terlanjur mempelajari *JavaScript* terlebih dahulu cenderung ingin tetap memanfaatkannya dibanding harus menggantinya dengan *TypeScript*. *Framework Angular* dibuat dengan menggunakan bahasa pemrograman *TypeScript* sebagai bahasa pemrograman utamanya. Akan tetapi, developer masih dapat menggunakan *JavaScript* secara bersamaan. Penelitian ini akan membahas bagaimana menggunakan *JavaScript* dan *TypeScript* secara *hybrid* di dalam *framework Angular* guna meningkatkan produktivitas pengembang perangkat lunak yang sudah memiliki kemampuan *coding* di *JavaScript*.

Kata Kunci :
Kode *Hybrid*; *JavaScript*; *TypeScript*; *Angular*

ABSTRACT

TypeScript is an improvement of the JavaScript programming language. However, its advantages do not necessarily mean we do not need JavaScript anymore. This is because the complexity of the TypeScript programming language is still relatively higher compared to JavaScript. Web platform software developers who have already learned JavaScript first tend to want to continue using it rather than having to replace it with TypeScript. The Angular framework is created using the TypeScript programming language as its main programming language. However, developers can still use JavaScript simultaneously. This study will discuss how to use JavaScript and TypeScript in a hybrid manner within the Angular framework to increase the productivity of software developers who already have coding skills in JavaScript.

Penulis Korespondensi:

Tony Wijaya
Sistem Informasi, STMIK Pontianak

Email:
tony_wijaya@stmikpontianak.ac.id

***Keywords*:**
Hybrid coding; JavaScript; TypeScript; Angular

1. PENDAHULUAN

JavaScript telah lama menjadi salah satu bahasa pemrograman paling dominan dalam pengembangan perangkat lunak, terutama di ranah pengembangan *web*. Keunggulannya terletak pada sintaksisnya yang sederhana, fleksibilitasnya yang tinggi, serta kompatibilitasnya dengan berbagai *platform*. Tidak heran jika banyak pengembang, baik pemula maupun profesional, memulai perjalanan mereka dengan mempelajari *JavaScript*. Namun, meskipun telah menjadi pondasi kuat dalam pengembangan teknologi *web modern*, JavaScript memiliki beberapa keterbatasan terutama dalam hal pengelolaan kode dan skalabilitas proyek.

Seiring dengan meningkatnya kompleksitas aplikasi, muncul tantangan dalam memastikan bahwa kode tetap terstruktur, mudah dipahami, dan bebas dari kesalahan yang sulit dideteksi saat *runtime*. Salah satu kekurangan utama *JavaScript* [1] adalah ketiadaan fitur *static typing*, yang menyebabkan kesalahan dalam kode

baru terlihat saat eksekusi program. Hal ini menjadi tantangan besar dalam proyek besar, di mana pengelolaan kode yang baik sangat krusial untuk mempertahankan efisiensi pengembangan.

Untuk mengatasi keterbatasan tersebut, *TypeScript* hadir sebagai penyempurnaan dari *JavaScript* dengan menawarkan fitur *static typing*, yang memungkinkan pengembang mendeteksi dan mengurangi kesalahan sejak tahap awal pengembangan [2]. Keunggulan *TypeScript* dalam meningkatkan keandalan kode telah dibuktikan dalam berbagai penelitian sebelumnya. Studi menunjukkan bahwa proyek yang menggunakan *TypeScript* cenderung memiliki kode yang lebih terstruktur dan lebih mudah dikelola dibandingkan proyek yang hanya menggunakan *JavaScript*. Dengan adanya mekanisme pengetikan statis, *TypeScript* membantu mengurangi potensi *runtime errors* serta meningkatkan keterbacaan dan pemeliharaan kode dalam jangka panjang.

Salah satu implementasi nyata dari kekuatan *TypeScript* dalam pengembangan aplikasi *web* adalah *framework Angular* [3]. *Angular* dirancang untuk meningkatkan efisiensi dan produktivitas pengembang dengan memanfaatkan fitur-fitur unggulan *TypeScript*, seperti sistem modularisasi yang lebih kuat dan dukungan penuh terhadap konsep *Object-Oriented Programming* (*OOP*). Namun, kelebihan ini tidak serta-merta menghilangkan relevansi *JavaScript*, karena *Angular* juga mendukung penggunaan *JavaScript* dalam proses pengembangannya. Hal ini memberikan fleksibilitas bagi pengembang yang telah terbiasa menggunakan *JavaScript* untuk tetap beradaptasi tanpa harus meninggalkan keterampilan yang telah mereka kuasai.

Meskipun penggunaan *TypeScript* menawarkan banyak manfaat, transisi dari *JavaScript* ke *TypeScript* tidak selalu mudah bagi semua pengembang. Perubahan sintaksis serta konsep baru yang diperkenalkan oleh *TypeScript* dapat menjadi tantangan, terutama bagi mereka yang sudah lama menggunakan *JavaScript* sebagai alat utama. Oleh karena itu, pendekatan *hybrid* yang menggabungkan *JavaScript* dan *TypeScript* dapat menjadi solusi yang menjembatani kesenjangan antara teknologi lama dan baru. Beberapa penelitian telah menyoroti bahwa pendekatan ini memungkinkan transisi yang lebih halus serta mengurangi hambatan dalam adopsi teknologi baru bagi tim pengembang.

Penelitian ini bertujuan untuk mengeksplorasi cara penggunaan *JavaScript* dan *TypeScript* secara *hybrid* dalam *framework Angular* [4]. Dengan pendekatan ini, diharapkan pengembang yang sudah memiliki keahlian di *JavaScript* dapat tetap produktif sekaligus memanfaatkan kemampuan tambahan yang ditawarkan oleh *TypeScript*. Pendekatan ini juga berpotensi mengurangi hambatan dalam adopsi teknologi baru dan membantu pengembang mempertahankan efisiensi pengembangan dalam proyek mereka.

Melalui pembahasan ini, penelitian ini memberikan kontribusi penting dalam mempromosikan efisiensi dan produktivitas dalam pengembangan perangkat lunak. Dengan mengintegrasikan keunggulan dua bahasa ini, pengembang dapat menjembatani kesenjangan antara teknologi yang sudah dikenal dan teknologi baru yang menawarkan potensi besar dalam mendukung pengembangan aplikasi *web modern*.

2. METODOLOGI PENELITIAN

Objek penelitian ini berfokus pada Sistem Informasi Dagang *PT. Banaki Jaya Teknik* (*SID Banaki Jaya Teknik*) versi *Android*. Penelitian ini dilakukan dengan pendekatan kualitatif [5]. Instrumen penelitian yang digunakan dalam penelitian ini adalah:

a. *Smartphone Xiaomi Redmi Note 10 Pro* dengan sistem operasi *Android* 13.
b. *Android Studio* versi 2024.3.1 (*Meerkat*) [6]
c. Aplikasi *Android hybrid SID Banaki Jaya Teknik*
d. *Web service* / *web API Banaki Jaya Teknik*

Penelitian ini mengadopsi pendekatan eksperimen dengan menjadikan aplikasi *Android SID Banaki Jaya Teknik* sebagai objek utama. Dalam proses pengembangan perangkat lunak, metode *Agile* [7] diterapkan. *Agile* sendiri berlandaskan prinsip-prinsip yang dijelaskan dalam *Agile Manifesto for Software Development* [8]. Metode ini tidak hanya berfokus pada penggunaan rancangan *UML*, tetapi juga menawarkan kemampuan untuk merespons perubahan secara cepat. Pendekatan ini sesuai dengan sifat penelitian yang berbentuk eksperimen.

Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 21-28
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2048

Extreme Programming (*XP*) [9] dipilih sebagai metode dalam proses pengembangan perangkat lunak pada penelitian ini. *XP* adalah pendekatan yang berlandaskan pada prinsip-prinsip seperti kesederhanaan, komunikasi, umpan balik, dan keberanian [10]. Siklus *XP* terdiri dari lima tahapan utama yang diterapkan dalam pengembangan aplikasi *SID Banaki Jaya Teknik*:

1. ***Planning*** **(Perencanaan)**
   Pada tahap awal, dilakukan identifikasi kebutuhan pengguna dan penentuan fitur utama yang akan dikembangkan dalam aplikasi. Tim pengembang menetapkan daftar tugas berdasarkan *user stories*, yang akan menjadi pedoman dalam setiap iterasi pengembangan.

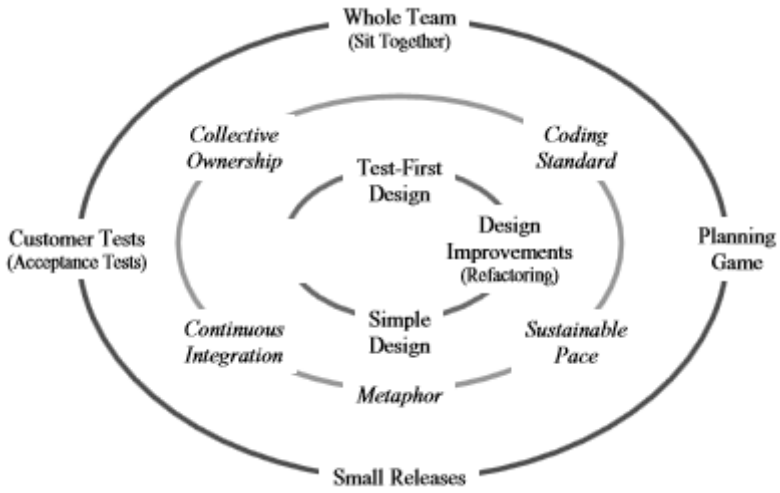
2. ***Design*** **(Desain)**
   Setelah kebutuhan ditentukan, dilakukan perancangan sistem dengan pendekatan *simple design*, yaitu desain yang cukup untuk memenuhi kebutuhan saat ini tanpa kompleksitas yang berlebihan. Diagram *UML* digunakan untuk membantu pemetaan arsitektur sistem yang dikembangkan.

3. ***Coding*** **(Pengkodean)**
   Implementasi kode dilakukan dengan konsep *pair programming*, di mana dua orang bekerja sama dalam satu perangkat untuk meningkatkan kualitas dan mengurangi kesalahan. Bahasa pemrograman yang digunakan pada sisi klien adalah *TypeScript* dan *JavaScript* dalam *framework Angular* [11], sedangkan *backend* menggunakan *PHP* dengan *framework CodeIgniter* [12]. *IDE Visual Studio Code* [14] digunakan dalam pengkodean.

4. ***Testing*** **(Pengujian)**
   Setiap fitur yang dikembangkan diuji dengan pendekatan *test-driven development* (*TDD*). Pengujian dilakukan dengan mengimplementasikan *unit testing* serta integrasi dengan *database MariaDB* yang berjalan di server berbasis *Ubuntu 20.04 Long Term Support* (*LTS*) [15]. Pendekatan ini memastikan bahwa setiap perubahan kode tidak menimbulkan kesalahan baru dalam sistem.

5. ***Release*** **(Peluncuran &** ***Feedback*****)**
   Setelah pengujian selesai, sistem diintegrasikan ke dalam lingkungan produksi menggunakan *Virtual Private Server* (*VPS*) berbasis *cloud* untuk memastikan stabilitas dan performa aplikasi. *Feedback* dari pengguna dikumpulkan dan digunakan dalam iterasi pengembangan berikutnya.

Siklus *XP* berlangsung secara iteratif, di mana setiap siklus berulang hingga fitur yang dikembangkan sesuai dengan kebutuhan pengguna. Pendekatan ini cocok untuk penelitian yang dilakukan oleh satu orang, karena *XP* memungkinkan seluruh tugas dikerjakan secara menyeluruh tanpa pembagian peran. Fleksibilitas dalam *XP* juga mempermudah respon terhadap perubahan yang mungkin terjadi selama pengembangan.



Gambar 1. Siklus Hidup *Extreme Programming*.

Aplikasi *Android hybrid SID Banaki Jaya Teknik* dikembangkan menggunakan *framework Angular* memanfaatkan *theme* dari *Admin LTE* versi 3 [11]. Bahasa pemrograman yang digunakan adalah *TypeScript* dan *JavaScript* [12] di sisi *client* (*framework Angular*). Sedangkan di sisi *backend* (*API*) menggunakan bahasa pemrograman *PHP* [13] dengan *framework CodeIgniter* [14] versi 3. Pengkodean menggunakan bantuan *IDE Visual Studio Code* [15]. Untuk server basis data, digunakan *Virtual Private Server* (*VPS*) yang berbasis *cloud* serta menjalankan sistem operasi *Ubuntu 20.04 Long Term Support* (*LTS*) [16]. Basis data yang dipakai adalah *MariaDB* versi 10.3.39, sebuah basis data *open source* [17]. *MariaDB* dipilih karena kompatibilitasnya yang tinggi dengan sistem operasi *Ubuntu*, sehingga mampu memberikan performa optimal.

3. HASIL DAN PEMBAHASAN

Hasil penelitian menunjukkan bahwa penggunaan pendekatan *hybrid* dengan *TypeScript* dan *JavaScript* dalam *framework Angular* meningkatkan efisiensi pengembangan aplikasi. *TypeScript*, dengan sistem pengetikannya yang statis, memungkinkan pengembang mendeteksi kesalahan kode lebih awal. Di sisi lain, fleksibilitas *JavaScript* memungkinkan penyesuaian kode yang lebih dinamis dan responsif terhadap kebutuhan proyek. Kombinasi ini menciptakan lingkungan yang seimbang antara kontrol ketat pada struktur kode dan kemampuan adaptasi cepat terhadap perubahan. Hal ini dapat dilihat pada potongan kode program pada gambar 2.

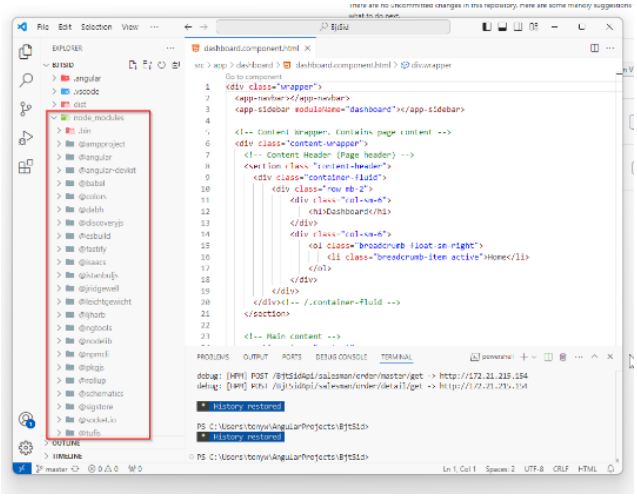
```
165  bindAmInvoiceNoText(showNotification: boolean = false): void {
166    var salesmanOrderDate = DatetimeService.toDbDate($("#am_salesmanOrderDateText").val());
167
168    this.httpClient.post(
169      CONFIG.apiBaseUrl + "salesman/order/master/invoice_no/get", {
170        "salesmanOrderDate": salesmanOrderDate
171      }).subscribe((response: any) => {
172        //console.log(response);
173        $("#am_invoiceNoText").val(response.invoiceNo);
174
175        if (showNotification) {
176          ToastrService.showSuccess("No invoice berhasil diperbaharui.");
177        }
178      }
179    );
180  }
```

Gambar 2. Potongan Kode dalam Modul "Tambah *Order Sales*".

Analisis performa menunjukkan bahwa teknik *hybrid* menghasilkan aplikasi yang stabil dan cepat. *TypeScript* memastikan konsistensi dengan struktur kode yang kuat, sedangkan *JavaScript* mendukung eksekusi *runtime* yang fleksibel untuk mengoptimalkan performa pada bagian tertentu dari aplikasi. Pendekatan ini memungkinkan pengembang untuk mengatasi tantangan performa tanpa mengorbankan integritas kode secara keseluruhan.

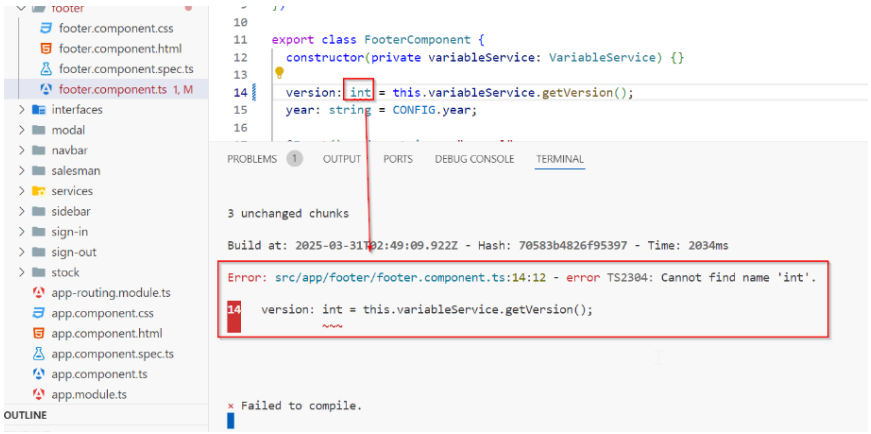
Ketika diterapkan pada proyek berskala besar, teknik *hybrid* ini menawarkan banyak keuntungan. *TypeScript* menyediakan struktur kode yang terorganisasi dengan baik, sehingga pengembang dapat dengan mudah menambahkan atau memodifikasi komponen tanpa menyebabkan masalah kompatibilitas. *TypeScript* menyediakan sangat banyak *library built-in* yang siap pakai dalam *folder node_modules*. Di sisi lain, *JavaScript* memberi ruang untuk improvisasi yang memungkinkan integrasi cepat pada elemen-elemen tertentu yang membutuhkan adaptasi langsung. *JavaScript* memiliki banyak *library* dari luar *framework Angular* yang dapat dimanfaatkan dalam *project* pengembangan perangakat lunak, seperti *MomentJS* (*library* untuk tanggal dan waktu), *ChartJS* (*library* untuk menampilkan grafik), *DataTables* (*library* untuk menampilkan data dalam bentuk tabel yang responsif dan dinamis) dan banyak lagi *library-library* lainnya. Hal ini sangat memudahkan pengembang perangkat lunak dalam mempercepat *time-to-market project* yang sedang dikerjakan.

Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 21-28
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2048



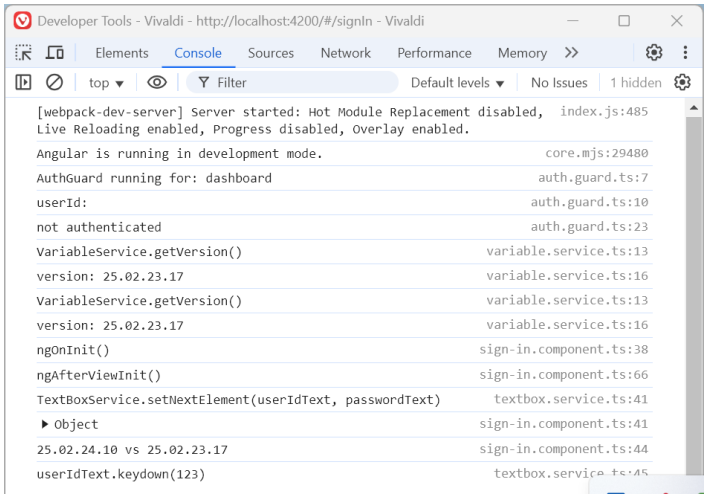
Gambar 3. *Folder node_modules* dalam *Framework Angular*.

Penggunaan *TypeScript* dan *JavaScript* dalam *framework Angular* didukung oleh komunitas pengembang yang luas dan alat bantu yang terus diperbarui. Hal ini menjadi keunggulan besar karena pengembang dapat mengakses dokumentasi dan sumber daya untuk menyelesaikan permasalahan teknis. Pengembang yang terbiasa dengan *JavaScript* beserta *library-library* pendukungnya, tidak perlu belajar lagi di *TypeScript*. Karena ilmu tersebut masih dapat diterapkan pada *project* di *framework Angular* ini. Jadi mereka tidak butuh terlalu banyak waktu untuk menyesuaikan diri pada *framework* ini. Selain itu, perpaduan kedua teknologi ini mendapat respons positif dari komunitas yang mendorong adopsi teknik *hybrid* secara lebih luas.



Gambar 4. *TypeScript Debugging* pada *Framework Angular*.

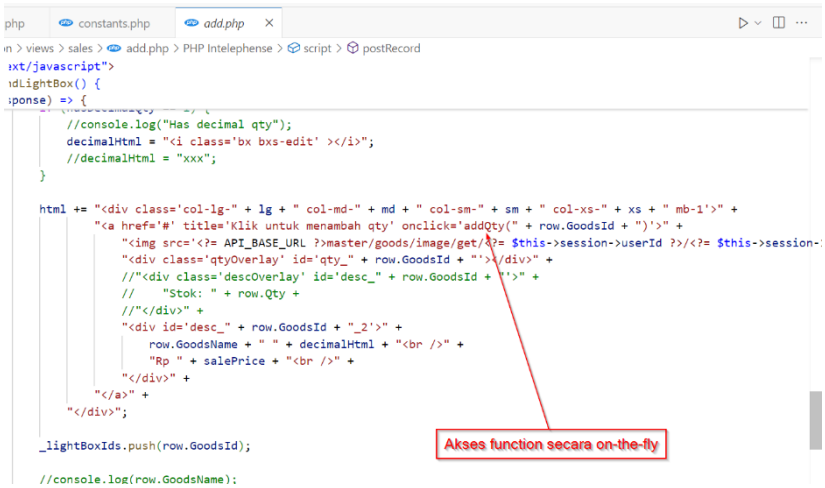
Teknik *hybrid* ini juga membawa manfaat signifikan pada proses *debugging*. Dengan fitur *type-checking* yang dimiliki *TypeScript*, kesalahan kode dapat diidentifikasi selama tahap kompilasi. Sebaliknya, kemampuan *JavaScript* dalam eksekusi *runtime* sangat membantu dalam menyelesaikan masalah yang muncul selama aplikasi dijalankan. Hal ini bisa dicapai dengan modul *Developer Tools* yang telah disediakan oleh tiap-tiap peramban yang ada. Pada *tools* ini, kita bisa melihat perubahan kode *HTML* dan *JavaScript,* penyimpanan sementara seperti *cookies*, *session storage*, *local storage* dan *IndexedDB*, penggunaan memori pada laman yang aktif, lama waktu pemuatan laman (pengukuran performa aplikasi *web*), *file-file* yang diunduh oleh laman yang bersangkutan, pesan-pesan dan log aplikasi *web* (*log*) dan lain-lainnya. Kombinasi ini memberikan pendekatan *debugging* yang lebih komprehensif dan efisien.



Gambar 5. *Developer Tools* pada *Peramban Vivaldi.*

Framework Angular yang berorientasi pada modularitas sangat mendukung pendekatan *hybrid* ini. *TypeScript* digunakan untuk membuat komponen yang terstruktur dengan baik, sementara *JavaScript* memungkinkan fleksibilitas dalam proses integrasi antar-komponen. Hasilnya adalah aplikasi yang lebih solid dengan koneksi antar-komponen yang lebih lancar.

Walaupun memiliki banyak manfaat, penggunaan teknik *hybrid* juga membawa tantangan. Pengembang harus memahami dan menguasai kedua bahasa, yang bisa menjadi hambatan bagi mereka yang belum berpengalaman. Selain itu, kombinasi *TypeScript* dan *JavaScript* terkadang menciptakan kompleksitas tambahan yang memerlukan strategi pengelolaan kode yang matang agar proyek tetap efisien. Hal ini dikarenakan kode *TypeScript* tidak bisa serta-merta diakses secara *on-the-fly* oleh *HTML* dan *JavaScript* konvensional. Oleh karena itu, pengembang aplikasi *web* yang sudah terbiasa mengkode secara *on-the-fly* secara *JavaScript*, baik dengan bentuk *Server-Side Rendering* (*SSR*) ataupun tidak, mereka butuh lebih banyak adaptasi di *framework Angular* ini.

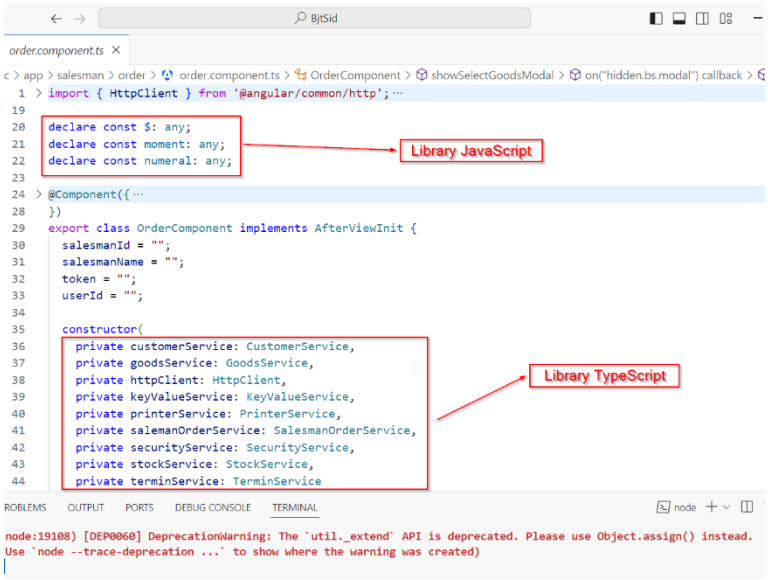


Gambar 6. Akses *Function* Secara *on-the-fly* dengan *JavaScript*.

Penelitian ini membuktikan teknik *hybrid* mampu meningkatkan stabilitas aplikasi dan mempercepat waktu respon. Hal ini menunjukkan bahwa pendekatan ini sangat efektif, terutama untuk proyek yang membutuhkan kecepatan tinggi tanpa mengorbankan kualitas.

Meskipun teknik ini menunjukkan banyak keunggulan, ada beberapa kekurangan yang perlu diperhatikan. Salah satu tantangan utama adalah pengelolaan kompleksitas tambahan yang muncul akibat penggunaan dua bahasa secara bersamaan. Selain itu, tidak semua pengembang merasa nyaman dengan transisi antara TypeScript dan JavaScript, yang dapat menghambat proses pengembangan.

Information System Journal (INFOS)
Vol. 8, No. 1, Mei 2025, pp. 21-28
E-ISSN : 2655-142X , P-ISSN : 2655-190X, DOI:https://doi.org/10.24076/infosjournal.2025v8i01.2048



Gambar 7. Kombinasi *Library JavaScript* dan *TypeScript* dalam Satu Laman yang Sama.

Temuan dalam penelitian ini memberikan pandangan bahwa teknik *hybrid* memiliki potensi besar untuk menjadi metode standar dalam pengembangan aplikasi berbasis *Angular*. Dengan dukungan alat bantu yang lebih baik dan pelatihan bagi pengembang, pendekatan ini dapat diadopsi secara luas. *Microsoft Visual Studio Code* merupakan salah satu *tools* terbaik untuk mengembangkan aplikasi dengan *framework Angular*. Salah satu fitur yang dipakai oleh sebagian besar pengembang adalah fitur *hot-reload* yang memberikan kemudahan *auto-refresh* pada laman *web* di peramban seketika setelah pengembang menyimpan *file*. Fitur lain yang juga sangat membantu pengembang adalah extension. Dengan fitur ini pengembang dapat semakin mempermudah pengkodean perangkat lunak dalam segala bahasa pemrograman dan *framework*. Ada banyak *extension* yang tersedia di *market*, bahkan pengembang sendiri pun bisa membuat *extension* dan mempublikasikannya di *market* tersebut. Di masa depan, diperlukan integrasi teknologi yang lebih mendalam untuk mengoptimalkan efektivitas pendekatan *hybrid* ini.

4. KESIMPULAN

Penelitian ini berawal dari permasalahan yang dihadapi oleh banyak pengembang dalam transisi dari *JavaScript* ke *TypeScript*. Meskipun *JavaScript* telah lama menjadi standar utama dalam pengembangan aplikasi *web*, keterbatasannya dalam pengelolaan kode yang skalabel dan deteksi kesalahan *runtime* sering kali menjadi tantangan, terutama dalam proyek dengan skala besar. Ketiadaan *static typing* membuat *debugging* menjadi lebih sulit dan meningkatkan risiko kesalahan yang baru terdeteksi saat eksekusi program.

Sebagai solusi, penelitian ini mengeksplorasi penerapan teknik *coding hybrid* antara *TypeScript* dan *JavaScript* dalam *framework Angular*. Pendekatan ini memungkinkan pengembang untuk tetap menggunakan *JavaScript* sambil memanfaatkan fitur unggulan *TypeScript*, seperti *static typing* dan alat pengembangan yang lebih canggih. Hasil penelitian menunjukkan bahwa pendekatan *hybrid* ini memberikan fleksibilitas yang signifikan, sehingga pengembang dapat menyesuaikan penggunaan bahasa sesuai dengan kebutuhan proyek. Selain itu, teknik ini mempermudah transisi bagi tim pengembang yang sudah mahir dalam *JavaScript* tanpa mengorbankan produktivitas.

Implikasi dari hasil penelitian ini cukup luas. Salah satu temuan utama adalah peningkatan efisiensi *debugging*, karena fitur *TypeScript* membantu mendeteksi kesalahan sejak tahap awal pengembangan, sehingga mengurangi potensi *runtime errors*. Selain itu, integrasi kedua bahasa ini juga memungkinkan kolaborasi yang lebih baik dalam tim yang terdiri dari pengembang dengan tingkat pengalaman yang beragam. Pengembang yang lebih berpengalaman dalam *JavaScript* dapat tetap bekerja dengan sintaks yang mereka kuasai, sementara mereka yang sudah familiar dengan *TypeScript* dapat berkontribusi dalam meningkatkan kualitas kode dengan fitur *static typing*.

Namun, penelitian ini juga mengidentifikasi beberapa tantangan yang perlu diperhatikan dalam penerapan teknik *coding hybrid* ini. Salah satu tantangan utama adalah kebutuhan akan pelatihan tambahan bagi pengembang yang belum familiar dengan konsep *TypeScript*. Selain itu, penggunaan dua bahasa dalam satu proyek dapat menambah kompleksitas dalam pengelolaan kode, yang perlu diatasi dengan strategi manajemen proyek yang lebih matang.

Dengan temuan-temuan ini, pendekatan *hybrid* dapat menjadi strategi transisi yang efektif bagi pengembang yang ingin beralih dari *JavaScript* ke *TypeScript* tanpa kehilangan produktivitas. Penelitian ini memberikan wawasan tentang bagaimana kombinasi kedua bahasa ini dapat dioptimalkan untuk meningkatkan efisiensi pengembangan aplikasi berbasis *web*, sekaligus mengurangi hambatan dalam adopsi teknologi baru.

DAFTAR PUSTAKA

[1] T. Wijaya, "Penerapan AJAX dalam Aplikasi Mobile Berbasis Web Untuk Meningkatkan Efisiensi Bandwidth," Techno.com, vol. 17, no. 2, pp. 197-207, 2018.

[2] Microsoft, "TypeScript : JavaScript with Syntax for Types," Microsoft, 2022. [Online]. Available: https://www.typescriptlang.org/. [Accessed 5 May 2022].

[3] "Introduction to Angular," [Online]. Available: https://angular.dev/. [Accessed 24 3 2024].

[4] "Using JavaScript and TypeScript Together," Google, [Online]. Available: https://angular.io/guide/js-to-ts. [Accessed 24 3 2025].

[5] W. Purhantara, Metode Penelitian Kualitatif untuk Bisnis, Yogyakarta: Graha Ilmu, 2010.

[6] "Download Android Studio & App Tools - Android Developers," Google, 2023. [Online]. Available: https://developer.android.com/studio. [Accessed 18 05 2023].

[7] A. A. Albarqi, "The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development," I.J. Information Engineering and Electronic Business, vol. 3, p. 13, 2018.

[8] M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, K. Schwaber, J. Sutherland and D. Thomas, "Signatories: The Agile Manifesto," 2001. [Online]. Available: http://agilemanifesto.org/. [Accessed 8 September 2018].

[9] M. A. F. Taftazani, E. Darwiyanto and R. Nurtantyana, "The development of Network Power System Website using Extreme Programming Method," Indo-JC, vol. 9, no. 2, pp. 154-167, Agustus 2024.

[10] L. Lindstrom and R. Jeffries, "Extreme Programming and Agile Software Development Methodologies," Information Systems Management, pp. 41-52, 2004.

[11] "Free Bootstrap Admin Template," AdminLTE, 2022. [Online]. Available: https://adminlte.io/. [Accessed 20 Mei 2022].

[12] M. Y. A. K. Yakti, A. Martono and L. Sunarya, "Rancang Bangun Website Berbasis Konvergensi Teknologi Informasi," Eksplora Informatika, vol. 1, no. 2, pp. 102-111, 2012.

[13] "PHP: Hypertext Preprocessor," The PHP Group, 2023. [Online]. Available: https://www.php.net/. [Accessed 01 05 2023].

[14] "CodeIgniter Web Framework," CodeIgniter Foundation, [Online]. Available: https://codeigniter.com. [Accessed 27 10 2020].

[15] Microsoft, "Visual Studio Code - Code Editing, Redefined," Microsoft, 2021. [Online]. Available: https://code.visualstudio.com/. [Accessed 6 Agustus 2021].

[16] "The leading operating system for PCs, IoT devices, servers and the cloud," Ubuntu, [Online]. Available: https://ubuntu.com. [Accessed 15 09 2019].

[17] "Supporting continuity and open collaboration," MariaDB, 2019. [Online]. Available: https://mariadb.org. [Accessed 15 09 2019].