

Implementasi Arsitektur *Microservice* pada Website Business Consolidation Simulation Menggunakan Metode *Extreme Programming*

Nashar Miracle¹, Rindu Pustpita Wibawa²

^{1,3} S1 Teknik Informatika, Universitas Negeri Surabaya

¹nashar.19026@mhs.unesa.ac.id

²rinduwibawa@unesa.ac.id

Abstrak— Perkembangan teknologi website meningkat pesat, dengan penetrasi internet di Indonesia mencapai 78,19% atau 215 juta jiwa menurut survei APJII 2023. Website berperan penting dalam menyediakan informasi dan interaksi. Pengembangan website yang baik memerlukan arsitektur yang tepat untuk menghindari masalah kinerja, pengembangan, dan keamanan. Website Business Consolidation Planning and Simulation (BCS) PT Sinergi Informatika Semen Indonesia, yang dibangun dengan PHP dan Laravel, menghadapi kendala dalam pengelolaan fitur baru dan skala pengguna karena arsitektur monolitiknya. *Microservices architecture* menawarkan kemudahan pengembangan, pengujian, penerapan, serta peningkatan ketahanan dan skalabilitas. Penelitian ini mengimplementasikan *microservices* pada website BCS menggunakan metode *extreme programming* untuk meningkatkan kinerja dan kestabilan aplikasi, serta mempermudah pengembangan dan pembagian tugas bagi para developer. Hasil penelitian menunjukkan bahwa *microservices* mengurangi beban server dan mempermudah pengguna karena hanya perlu memahami modul-modul tertentu. Implementasi ini meningkatkan efisiensi dan kestabilan sistem, memudahkan pengembangan fitur baru, dan mengoptimalkan alokasi tugas bagi developer. Kesimpulannya, implementasi *microservices architecture* dan *extreme programming* pada Website BCS efektif, memecah aplikasi besar menjadi modul-modul kecil yang dapat dijalankan tanpa keseluruhan aplikasi, mengurangi beban server, dan memudahkan pengguna serta developer dalam memahami dan mengelola hanya modul yang diperlukan.

Kata Kunci—Website, *Microservices*, PHP, Laravel, Master Data, OPEX, CAPEX.

I. PENDAHULUAN

Perkembangan teknologi sangat pesat. Salah satu perkembangan teknologi yang cukup terlihat yaitu perkembangan website. Menurut Survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) mencatat, penetrasi pengguna internet di Indonesia telah mencapai 78,19% atau menembus 215 juta jiwa dari total populasi sebesar 275 juta pada Januari 2023 [1]. Menurut data tersebut Masyarakat di Indonesia banyak menggunakan internet untuk keperluan e-commerce, e-learning, video streaming, video conference. Dari hal ini kita dapat memahami bahwa peranan penting penggunaan internet terutama pada website-website yang menyediakan layanan tersebut, Menurut Aziz Solechul (2013) Website adalah sebuah sistem informasi yang terdiri dari jaringan komputer dan informasi yang saling terhubung [2]. Dengan website memungkinkan pengguna untuk mengakses

berbagai macam informasi seperti teks, gambar, audio, video. Serta memungkinkan untuk saling interaksi antar pengguna melalui platform ini.

Penyebaran informasi melalui website cukup masif terjadi dikarenakan tidak lagi dibatasi oleh tempat dan waktu. Oleh sebab itu, website memainkan peran yang penting dalam pengelolaan informasi dengan dunia luas sehingga menjadi tuntutan untuk dapat memenuhi segala keperluan dari pengguna internet. Proses pengembangan website yang baik tidak hanya ditentukan oleh aspek desain dan fungsionalitasnya, tetapi juga oleh arsitektur yang mendasari dalam penggunaannya. Dalam proses pengembangan website, pemilihan arsitektur yang tepat dapat mempermudah dalam pengembangan website. Arsitektur yang tidak tepat dapat menyebabkan masalah kinerja, pengembangan, dan keamanan yang cukup kompleks. Oleh karena itu, perlu dapat diperhatikan dalam pemilihan arsitektur yang tepat dalam membangun suatu web [3].

Website Business Consolidation Planning and Simulation (BCS) yang digunakan oleh PT Sinergi Informatika Semen Indonesia merupakan suatu website yang dikembangkan berfungsi untuk melakukan pengelolaan dan pemenuhan strategi Perusahaan PT Sinergi Informatika Semen Indonesia pada departemen BSI (Business System Integration) perlu dilakukan penyempurnaan operasional (*operational excellence*) proses bisnis dan mendukung pengelolaan penyusunan Rencana Kerja dan Anggaran Perusahaan (RKAP)/ Rencana Jangka Pendek Perusahaan (RJPP) serta monitoring kegiatan untuk pencapaian RKAP/RJPP. Berdasar hal tersebut, maka dibutuhkan pengembangan sistem informasi yang berfungsi sebagai perencanaan dan monitoring program kerja Perusahaan, pengelolaan Capital Expenditure (CAPEX), Operational Expenditure (OPEX) dan Key Performance Indicator (KPI).

Website Business Consolidation Planning and Simulation (BCS) pengguna dapat melakukan berbagai hal seperti pengadaan barang dan jasa OPEX/CAPEX. Aplikasi website BCS memiliki beragam fitur antara lain Pengelolaan master data yang dibutuhkan, Proses pengadaan OPEX, Proses pengadaan CAPEX. Dalam pengembangan website BCS menggunakan bahasa pemrograman yang digunakan yaitu PHP dengan bantuan framework Laravel, database yang digunakan adalah MySQL.

Website BCS menggunakan arsitektur *monolithic architecture* untuk mengelola semua website. Sehingga seringkali mengalami kesulitan dalam mengelola penambahan fitur baru dan skala pengguna pengembangan yang semakin

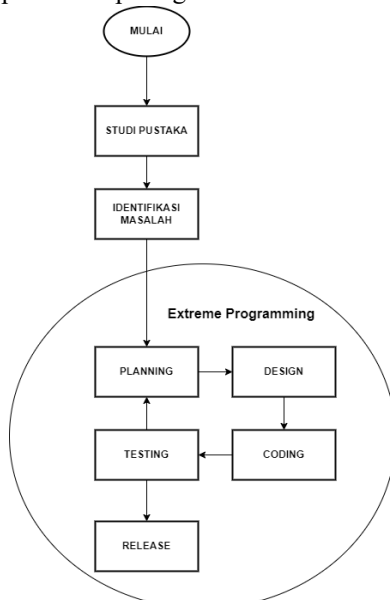
besar. Hal ini dapat menyebabkan performa dan keamanan sistem menjadi terganggu, serta dapat memperberat proses pengembangan dan pemeliharaan sistem pada masa yang akan datang.

Arsitektur microservices telah menjadi topik populer dalam pengembangan aplikasi web. Martin Fowler, seorang ahli perangkat lunak, menjelaskan bahwa arsitektur microservices adalah pendekatan untuk membangun aplikasi besar dengan memecahnya menjadi servis-servis kecil yang berdiri sendiri. Setiap servis memiliki API yang jelas dan padat, serta dapat berkomunikasi dengan servis lainnya. Pendekatan ini memberikan beberapa keuntungan, termasuk kemudahan dalam pengembangan, pengujian, dan penerapan, serta meningkatkan ketahanan dan skalabilitas sistem.

Oleh karena itu, perlu dilakukan penelitian dengan melakukan implementasi arsitektur microservices pada website BCS menggunakan metode extreme programming dengan tujuan untuk peningkatan kinerja performa aplikasi dan kestabilan sistem terutama saat akan melakukan proses pengembangan aplikasi lebih lanjut. Dengan kelebihan-kelebihan arsitektur microservices diharapkan dapat mempermudah proses development aplikasi. Selain itu, dengan menggunakan arsitektur microservices aplikasi akan dibagi dengan beberapa servis-servis untuk memudahkan dalam pembagian developer pada saat proses pengembangan sehingga akan lebih efisien saat dilakukan pengembangan lebih lanjut.

II. METODOLOGI PENELITIAN

Dalam Tahapan penelitian tentang Implementasi Arsitektur Microservices Pada Website Business Consolidation Planning and Simulation (BCS) Menggunakan metode Extreme Programming (XP), dilakukan serangkaian Langkah dan proses untuk melakukan pengumpulan data studi literatur hingga implementasi pengembangan aplikasi. Adapun tahapan penelitian dapat dilihat pada gambar 1.



Gbr 1. Diagram Aliran Proses Penelitian

A. Studi Pustaka

Langkah pertama, yaitu studi pustaka, melibatkan mencari dan membaca literatur yang relevan untuk memahami arsitektur microservices dan metode pengembangan aplikasi extreme programming. Tujuannya adalah untuk mengidentifikasi konsep, kelebihan, dan kekurangan keduanya. Langkah ini penting untuk membangun pemahaman yang kuat, dengan merujuk pada buku, jurnal ilmiah, artikel, atau sumber online yang terpercaya. Misalnya, buku "Building Microservices: Designing Fine-Grained Systems" oleh Sam Newman digunakan untuk mempelajari prinsip dan praktik terbaik dalam arsitektur microservices.

B. Identifikasi Masalah

Langkah kedua dalam penelitian ini adalah mengidentifikasi masalah dan memilih studi kasus yang relevan untuk mengimplementasikan arsitektur microservices pada website Business Consolidation Planning and Simulation (BCS) menggunakan metode Extreme Programming (XP). Fokusnya adalah pada menu Master Data dan Transaksi, yang akan menjadi Master Data Services dan Transaksi Services. Master Data mencakup data statis seperti Company, Plant, Asset Code, Costcenter, GL Account, dan Project Profile, sementara Transaksi mencakup data dinamis OPEX dan CAPEX.

Implementasi arsitektur microservices dilakukan dengan menggunakan standar REST. Setiap operasi CRUD (Create, Read, Update, Delete) diwakili oleh endpoint API yang terpisah:

1. Create (POST): Membuat entitas baru, seperti "create company".
2. Read (GET): Mengambil entitas yang sudah ada.
3. Update (PUT/PATCH): Memperbarui entitas yang ada, seperti "update company".
4. Delete (DELETE): Menghapus entitas yang ada, seperti "delete company".

Dengan adopsi arsitektur microservices dan standar REST, setiap layanan akan memiliki endpoint API khusus yang dapat diakses dengan permintaan HTTP sesuai metode yang digunakan (POST, GET, PUT, DELETE).

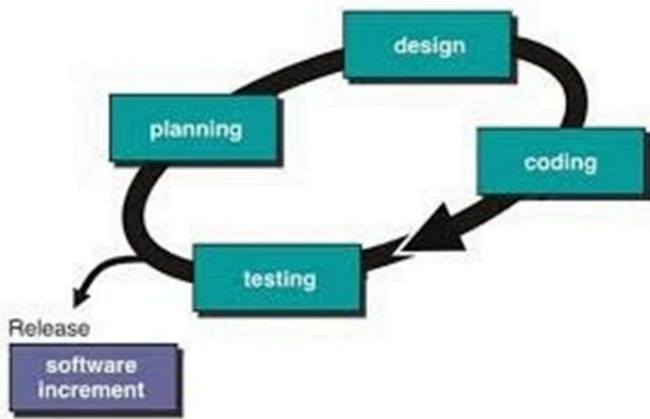
C. Penerapan metode Extreme Programming

Dalam penelitian ini proses pengembangan aplikasi dilakukan dengan menggunakan metode Extreme Programming. Metode ini dilakukan serangkaian langkah dan proses yang dilakukan secara iterasi. Menurut Putu Hartawan (2021) dalam International Journal of Engineering and Emerging Technology bahwa Extreme Programming (XP) adalah metode pengembangan perangkat lunak dan merupakan salah satu metode agile yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham [4]. XP merupakan metode agile yang paling banyak digunakan dan diuraikan secara rinci selama fase perencanaan. Hal ini dilakukan untuk mendapatkan umpan balik yang lebih cepat dari klien.

Menurut jurnal internasional yang ditulis Rostislav Fojtik dan dikeluarkan oleh Procedia Computer Science pada tahun 2010 dengan judul "Extreme Programming in Development of

Specific Software” bahwa Metodologi XP didesain utamanya untuk tim yang lebih kecil dengan dua hingga sepuluh anggota yang akan mengerjakan tugas tersebut. Sehingga cukup efisien jika dikerjakan oleh tim yang kecil. XP adalah metodologi agile yang fleksibel yang menekankan interkoneksi proposal dan tahap implementasi [5].

Adapaun tahapan tersebut dapat dilihat pada gambar 2 dibawah ini.



Gbr 2. Diagram Proses Metode Extreme Programing

1. Langkah 1 : Planning

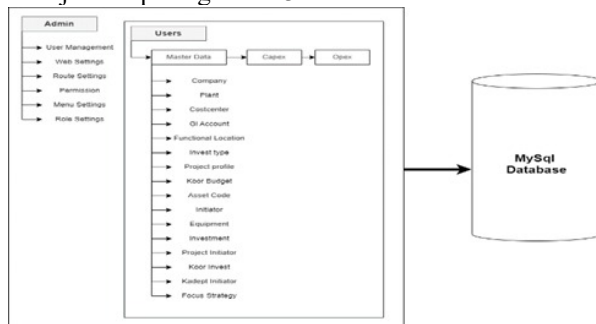
Langkah pertama adalah membuat rancangan proses bisnis untuk sistem dan fitur-fitur pada aplikasi BCS. Tujuannya adalah memberikan gambaran jelas mengenai fitur utama, fungsionalitas, dan keluaran aplikasi. Proses planning ini penting sebagai dasar untuk membangun pemahaman yang kuat dan mengacu pada kebutuhan perusahaan.

2. Langkah 2 : Design

Langkah kedua dalam penelitian ini adalah membuat desain kebutuhan aplikasi seperti desain struktur aplikasi, diagram struktur penyimpanan, dan diagram Use case.

a. Desain Struktur Aplikasi

Adapun struktur aplikasi website BCS yang ditunjukkan pada gambar 3 dibawah ini.



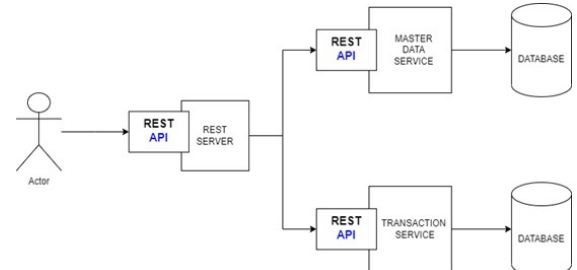
Gbr 3. Struktur aplikasi website BCS

Gambar di atas menunjukkan bahwa file-file dalam website BCS tersebar di beberapa direktori tanpa standar tetap dan terhubung ke satu database. Untuk menambah fitur dan skala, diperlukan pemahaman

menyeluruh tentang keterikatan file. Tanpa pemahaman ini, aplikasi bisa gagal karena masalah integrasi antar file. Ini adalah contoh arsitektur monolitik, yang kelemahannya menghambat pengembangan aplikasi website BCS.

b. Desain Struktur Penyimpanan

Adapun struktur penyimpanan website BCS yang ditunjukkan pada gambar 4 dibawah ini

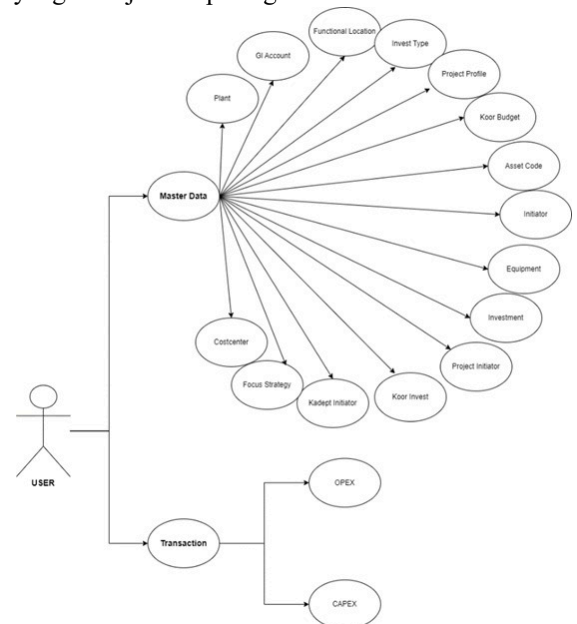


Gbr 4. Struktur Penyimpanan Website BCS

Sesuai gambar tersebut. Website BCS akan berinteraksi dengan masing-masing service dengan dihubungkan oleh API . dan hasil inputan akan di simpan pada database yang digunakan (MySQL).

c. Desain Diagram Use Case

Adapun diagram use case pada website BCS yang ditunjukkan pada gambar 5 dibawah ini.



Gbr 5. Diagram Use Case Website BCS

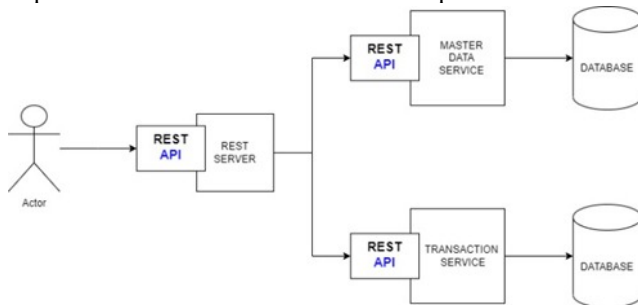
Pada gambar tersebut user dapat berinteraksi dengan 2 menu utama yaitu Master Data dan Transaksi. User dapat melakukan CRUD (Create, Read, Delete, Update) pada tiap-tiap menu.

3. Langkah 3 :Coding

Pada langkah ini, dilakukan perancangan sistem dengan mempertimbangkan prinsip dan praktik terbaik dari arsitektur. Arsitektur microservices adalah pendekatan

desain perangkat lunak di mana aplikasi dibangun sebagai kumpulan servis mandiri yang dapat dikembangkan, dikelola, dan diterapkan secara independen. Dalam perancangan sistem, komponen-komponen utama yang akan menjadi servis mandiri dalam arsitektur microservices akan diidentifikasi. Aspek-aspek seperti desain API yang jelas, komunikasi antara server-klien melalui protokol REST, dan pemisahan tanggung jawab fungsional antara servis-servis tersebut akan dipertimbangkan.

Gambar berikut ini menunjukkan rancangan dari implementasi architecture microservices pada website BCS.



Gbr 6. Struktur Website BCS Microservices

4. Langkah 4 : Testing

Setelah merancang sistem microservices website BCS, akan dilakukan testing aplikasi dengan menggunakan metode blackbox. Jika sistem memiliki kesalahan maka akan dilakukan perbaikan pada tahap coding.

III. HASIL DAN PEMBAHASAN

Pada bagian ini, dilakukan pembahasan hasil dari penelitian yang sudah dilaksanakan menurut metode penelitian yang sudah dijelaskan dalam bagian sebelum ini

A. Tahapan Pengembangan Aplikasi

Pengembangan website BCS menggunakan metode Extreme Programming dengan melalui 4 tahapan yaitu:

1. Planning

Dalam langkah ini dilakukan asesmen terkait proses bisnis, fitur aplikasi, dan spesifikasi sistem. Proses bisnis website BCS meliputi Master Data dan Menu Transaksi. Master data ini akan berisi 17 menu master yang ditunjukkan pada tabel 1 berikut ini.

TABEL I
LIST MENU MASTER

No.	Menu Master	No.	Menu Master
1.	Company	10.	Initiator
2.	Plant	11.	Equipment
3.	InvestType	12.	Investment
4.	Costcenter	13.	Investgroup
5.	Glaccount	14.	Projectinisiator
6.	Functional-location	15.	Coorinvest
7.	Projectprofile	16.	Kadeptiniator
8.	Koorbudget	17.	Focusstrategy

9.	Assetcode		
----	-----------	--	--

Selain master data, website BCS juga akan dilengkapi dengan menu Transaksi untuk melakukan transaksi data. Menu ini akan berisi 2 menu yaitu:

a. Opex (Operational Expenditure)

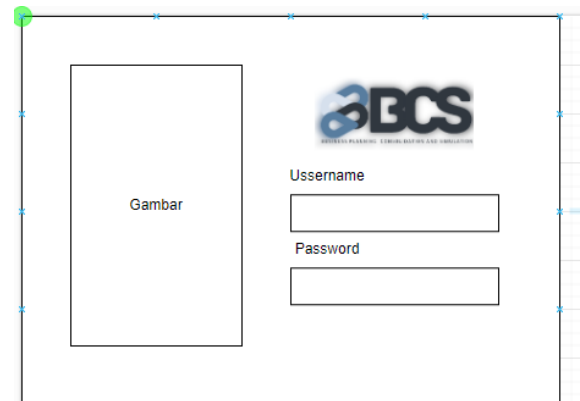
Operational expenditure (OPEX) adalah biaya yang digunakan secara langsung maupun tidak langsung untuk produk dan kegiatan operasional perusahaan sehari-hari [6].

b. Capex (Capital Expenditure)

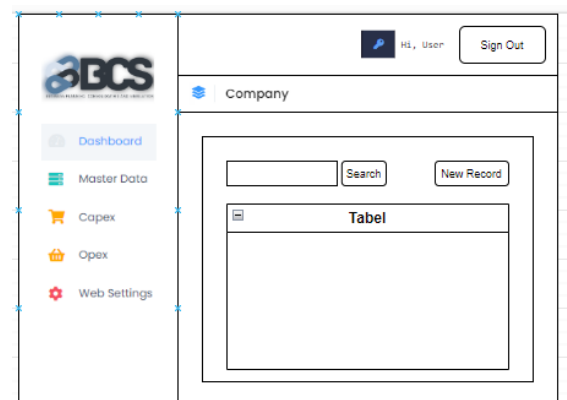
Capital expediture merupakan biaya-biaya yang dikeluarkan dalam rangka memperoleh aset tetap, meningkatkan efisiensi operasional dan kapasitas produktif aset tetap [7].

2. Design

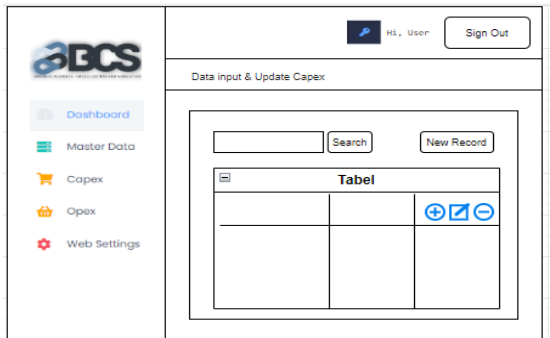
Website BCS menggunakan didesain dengan mengutamakan User Experience dan Responsifitas dengan berbagai macam tipe device. Berikut hasil desain Website BCS:



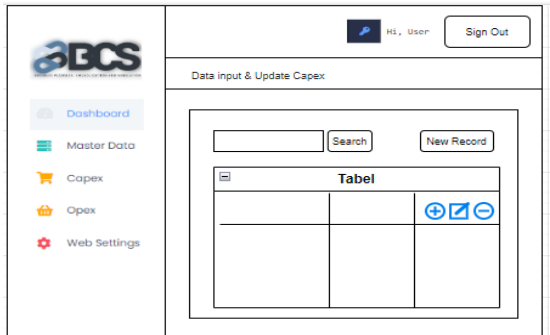
Gbr 7. Halaman Login



Gbr 8. Halaman Master Data



Gbr 9. Halaman Transaksi

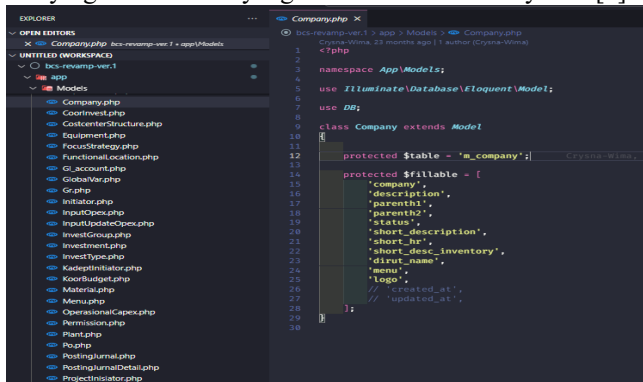


Gbr 10. Halaman Web Setting

3. Coding

Website BCS menggunakan bahasa pemrograman PHP dengan framework Laravel. Dalam pengembangannya website BCS menggunakan konsep MVC Yaitu Model, View, Controller. Dengan menggunakan pattern ini sangat memudahkan dalam melakukan develop website BCS.

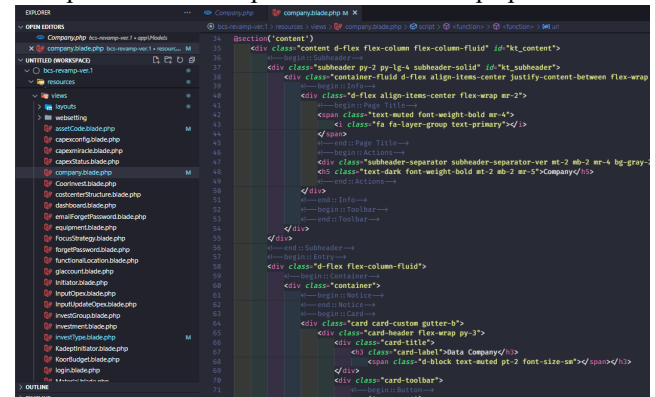
Menurut Wijaya & Christian, Model adalah bagian yang bertugas untuk menyiapkan, mengatur, memanipulasi dan menyinkronkan data yang ada dalam database system [8].



Gbr 11. Code Model

View merupakan bagian yang bertugas untuk menampilkan informasi dalam bentuk Graphical User Interface (GUI). Bagian ini berfungsi mengatur tampilan ke pengguna. Dalam website BCS yang menggunakan framework Laravel akan dibantu dengan Blade Laravel. Fitur ini disediakan laravel untuk proses templating

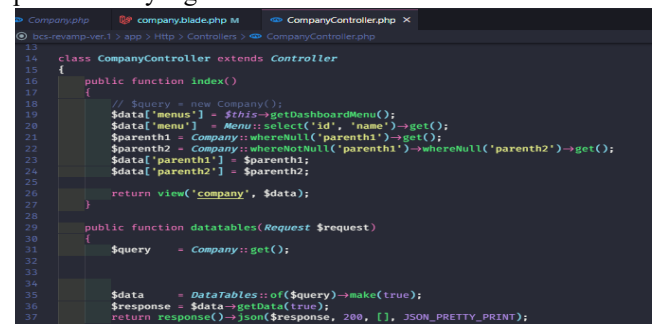
sederhana. Dengan Blade Templating developer tidak perlu menuliskan sebuah kode berulang kali, sehingga dapat mempermudah developer untuk mendevlop aplikasi.



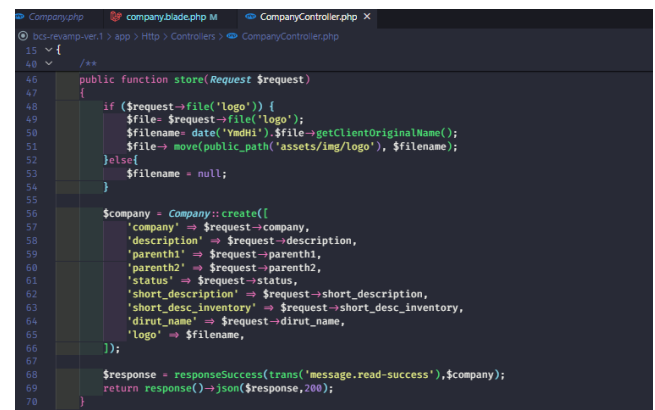
Gbr 12. Code View

Controller Merupakan bagian yang bertugas untuk menghubungkan serta mengatur model dan view agar dapat saling terhubung. Controller akan memproses data dan permintaan yang masuk, kemudian memberitahu model apa saja yang harus bekerja dan menampilkan hasil nya lewat view.

Pada controller terdapat function yang berfungsi sebagai CRUD (Create Read Update Delete). Tiap- tiap function akan memberitahu Model untuk melakukan CRUD data pada kolom yang dimaksud.



Gbr 13. Function Indeks dan datatables



Gbr 14. Function Store


```
public function update($id, CompanyRequest $request)
{
    $company = Company::find($id);
    $data = $this->findDataWhere(Company::class, ['id' => $id]);
    if ($request->file('logo')) {
        $file = $request->file('logo');
        $filename = date('YmdHis').$file->getClientOriginalName();
        $file->move(public_path('assets/img/logo'), $filename);
    } else {
        $filename = $company->logo;
    }
    // dd($data);exit();
    DB::beginTransaction();
    try {
        $data->update([
            'company' => $request->company,
            'description' => $request->description,
            'parenth1' => $request->parenth1,
            'parenth2' => $request->parenth2,
            'status' => $request->status,
            'short_description' => $request->short_description,
            'short_desc_inventory' => $request->short_desc_inventory,
            'dirut_name' => $request->dirut_name,
            'logo' => $filename,
        ]);
        DB::commit();
        $response = responseSuccess(trans('messages.update-success'), $data);
        return response()->json($response, 200, [], JSON_PRETTY_PRINT);
    } catch (Exception $e) {
        DB::rollback();
        $response = responseFail(trans('messages.update-fail'), $e->getMessage());
        return response()->json($response, 500, [], JSON_PRETTY_PRINT);
    }
}
```

Gbr 15. Function Update

```
public function destroy($id)
{
    Company::destroy($id);
    $response = responseSuccess(trans('message.delete-success'));
    return response()->json($response, 200);
}
```

Gbr 16. Function Destroy

4. Testing

Metode testing akan menggunakan Blackbox. Menurut Taufik, M (2023) Metode black box testing merupakan metode yang menguji perangkat lunak yang telah dibangun, baik pengujian pada unit-unit kecil maupun hasil yang telah terintegrasi untuk menguji fungsional perangkat lunak [9]. Testing akan dilakukan dengan cara melakukan sampling Create data, Read Data, Upload Data, dan Delete data pada service Master data dengan port 8001 dan Transaksi dengan port 8000.

a. Testing menu master data

• Create data

Pada tahap testing *create* pada master data didapatkan hasil *Queueing* sebesar 2.61 ms yang artinya proses tersebut memakan waktu 2.61 ms. Hasil testing *create* pada master data ditunjukkan pada gambar 17 dibawah ini.

Queued at 0	
Started at 2.61 ms	
Resource Scheduling	
Queueing	DURATION
2.61 ms	
Connection Start	
DURATION	
Stalled	0.39 ms
DNS Lookup	0.70 ms
Initial connection	1.07 ms
Request/Response	
DURATION	
Request sent	0.48 ms

Gbr 17. Hasil Testing Create pada Master Data

• Update data

Pada tahap testing *update* pada master data didapatkan hasil *Queueing* sebesar 4.83 ms yang artinya proses tersebut memakan waktu 4.83 ms. Hasil testing *update* pada master data ditunjukkan pada gambar 18 dibawah ini.

Queued at 0	
Started at 4.83 ms	
Resource Scheduling	
Queueing	DURATION
4.83 ms	
Connection Start	
DURATION	
Stalled	1.81 ms
DNS Lookup	0.84 ms
Initial connection	1.07 ms

Gbr 18. Hasil Testing Update pada Master Data

• Delete data

Pada tahap testing *delete* pada master data didapatkan hasil *Queueing* sebesar 2.45 ms yang artinya proses tersebut memakan waktu 2.45 ms. Hasil testing *delete* pada master data ditunjukkan pada gambar 19 dibawah ini.

Queued at 0	
Started at 2.45 ms	
Resource Scheduling	
Queueing	DURATION
2.45 ms	
Connection Start	
DURATION	
Stalled	2.11 ms
DNS Lookup	0.71 ms
Initial connection	0.87 ms

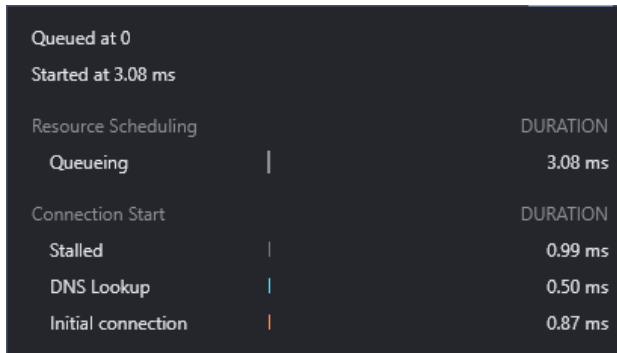
Gbr 19. Hasil Testing Delete pada Master Data

b. Testing menu transaksi

• Create data

Pada tahap testing *create* pada menu transaksi didapatkan hasil *Queueing* sebesar 3.08 ms yang artinya

proses tersebut memakan waktu 3.08 ms. Hasil testing *create* pada menu transaksi ditunjukkan pada gambar 20 dibawah ini.

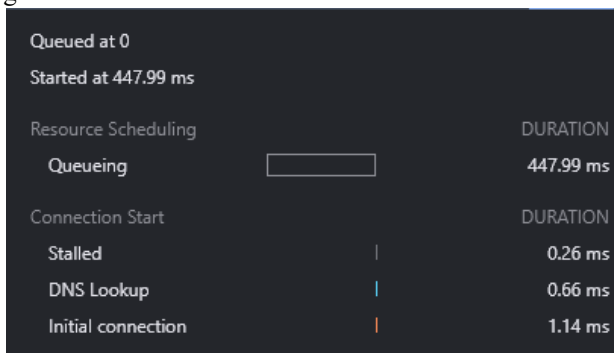


Queued at 0	
Started at 3.08 ms	
Resource Scheduling	DURATION
Queueing	3.08 ms
Connection Start	DURATION
Stalled	0.99 ms
DNS Lookup	0.50 ms
Initial connection	0.87 ms

Gbr 20. Hasil Testing Create pada Menu Transaksi

- Update data

Pada tahap testing *update* pada menu transaksi didapatkan hasil *Queueing* sebesar 447.99 ms yang artinya proses tersebut memakan waktu 447.99 ms. Hasil testing *update* pada menu transaksi ditunjukkan pada gambar 21 dibawah ini.

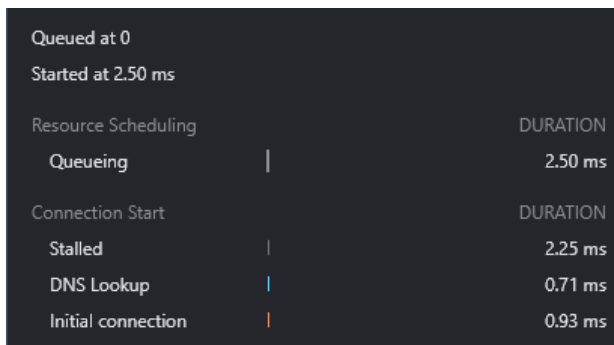


Queued at 0	
Started at 447.99 ms	
Resource Scheduling	DURATION
Queueing	447.99 ms
Connection Start	DURATION
Stalled	0.26 ms
DNS Lookup	0.66 ms
Initial connection	1.14 ms

Gbr 21. Hasil Testing Update pada Menu Transaksi

- Delete data

Pada tahap testing *delete* pada menu transaksi didapatkan hasil *Queueing* sebesar 2.50 ms yang artinya proses tersebut memakan waktu 2.50 ms. Hasil testing *delete* pada menu transaksi ditunjukkan pada gambar 22 dibawah ini.



Queued at 0	
Started at 2.50 ms	
Resource Scheduling	DURATION
Queueing	2.50 ms
Connection Start	DURATION
Stalled	2.25 ms
DNS Lookup	0.71 ms
Initial connection	0.93 ms

Gbr 22. Hasil Testing Delete pada Menu Transaksi

B. Microservices

Penerapan Microservice pada website BCS dilakukan dengan cara membuat Route API yang diwakili dengan endpoint GET, POST, PUT, DELETE. Berikut merupakan code untuk menautkan route API. API ini dibuat di folder routes/api.php pada aplikasi website BCS port:8001 yang berisi list alamat API yang akan di tuju pada aplikasi website BCS port:8000.

Lalu pada folder view aplikasi website BCS port:8000 kita perlu menambahkan alamat API pada function yang bertugas melakukan CRUD.

```
@can(['investType-C', 'investType-U'])
$('#formmenu').submit(function (e) {
    e.preventDefault();
    var formData = new FormData($('#formmenu')[0]);
    // var formData = $('#formmenu').serializeArray(); // our data object
    var method = "POST";
    let menuID = $('#saveMenu').data("id");

    if (typeof menuID == "undefined" || menuID == "") {
        var url = "http://127.0.0.1:8001/api/investType";
    } else {
        var url = "http://127.0.0.1:8001/api/investType/${menuID}/update";
    }
});
```

Gbr 23. Contoh Function Update yang telah diberi alamat API.

Hal ini diterapkan pada seluruh menu view pada port:8000 agar dapat memberikan petunjuk alamat pengiriman API.

IV. KESIMPULAN

Penelitian ini bertujuan untuk mengetahui hasil implementasi microservices architecture pada website BCS dan Mengetahui hasil implementasi metode extreme programming dalam pengembangan aplikasi website. Penelitian ini berhasil dilakukan dengan menggunakan bahasa pemrograman PHP dengan framework Laravel. Penelitian ini menerapkan metode extreme programming untuk pengembangan website BCS. Tahapan meliputi analisis bisnis dan kebutuhan sistem, desain alur aplikasi, use case, dan UI, penulisan clean code dalam coding, serta pengujian Blackbox untuk CRUD. Hasil pengukuran Queueing menunjukkan performa service Master data dan Transaksi dalam satuan ms. Penelitian ini menyimpulkan bahwa microservices architecture dan extreme programming pada Website BCS memungkinkan pemecahan service dalam aplikasi besar tanpa perlu menjalankan keseluruhan aplikasi, disebut sebagai "Modulasi". Pendekatan ini mengurangi beban server dan mempermudah pengguna serta pengembang dalam pengelolaan dan pemahaman aplikasi hanya dengan memfokuskan pada modul-modul yang diperlukan.

V. SARAN

Semua *hypertext link* dan bagian *bookmark* akan dihapus. Jika paper perlu merujuk ke alamat email atau URL di artikel, alamat atau URL lengkap harus diketik dengan font biasa.

Berdasarkan pada hasil penelitian yang telah dilakukan, disarankan agar penelitian berikutnya terkait pengembangan *microservices* dapat menambah jumlah servis untuk mendapatkan hasil yang lebih maksimal, eksplorasi bahasa pemrograman server-side seperti Node.js, Java, dan Go untuk mendapatkan wawasan yang lebih luas, variasi parameter

dalam pengujian aplikasi dengan tools seperti Apache JMeter dan Clobbr untuk pengujian API Load dan Speed test yang lebih komprehensif, serta eksplorasi metode pengembangan aplikasi yang baru dan terbaru untuk meningkatkan pemahaman terhadap kecepatan dan keakuratan dalam pengembangan aplikasi.

REFERENSI

- [1] “Survei APJII Pengguna Internet di Indonesia Tembus 215 Juta Orang.” [Online]. Available: <https://apjii.or.id/berita/d/survei-apjii-pengguna-internet-di-indonesia-tembus-215-juta-orang>
- [2] S. Aziz, *Gampang dan Gratis Membuat Website: Web Personal, Organisasi dan Komersil*. Lembar Langit Indonesia, 2013, 2013.
- [3] S. Waruwu, I. Kadek, and D. Nuryana, “Implementasi Arsitektur Monolitik Pada Rancang Bangun Sistem Informasi,” *J. Informatics Comput. Sci.*, vol. 4, no. 4, pp. 399–404, 2023.
- [4] P. N. Hartawan, M. Sudarma, and Widyantara, “Extreme Programming for Developing Additional Employee Income System(Case Study: Karangasem Regency Government),” *Int. J. Eng. Emerg. Technol.*, vol. 6, no. 2, pp. 117–121, 2021, [Online]. Available: <https://ojs.unud.ac.id/index.php/ijeet/article/view/IJEET.2021.v06.i01.p21/39900>
- [5] R. Fojtik, “Extreme programming in development of specific software,” *Procedia Comput. Sci.*, vol. 3, pp. 1464–1468, 2011, doi: 10.1016/j.procs.2011.01.032.
- [6] L. D. Ibu and I. K. Gunarta, “Analisis Pengambilan Keputusan Investasi Infrastruktur Rantai Pasok PT X dengan Skenario Sewa atau Bangun,” *J. Tek. ITS*, vol. 11, no. 3, 2022, doi: 10.12962/j23373539.v11i3.97609.
- [7] I. N. Fitri, “Analisis Pengaruh Tingkat Kenaikan Penggunaan Capital Expenditure Terhadap Kinerja Perusahaan,” *J. Mhs. Univ. Negeri Surabaya*, 2014.
- [8] K. Wijaya and A. Christian, “Implementasi Metode Model View Controller (MVC) Dalam Rancang Bangun Website SMK Yayasan Bakti Prabumulih,” *Paradig. - J. Komput. dan Inform. Univ. Bina Sarana Inform.*, vol. 21, no. 1, pp. 95–102, 2019, doi: 10.31294/p.v21i1.5092.
- [9] M. T. Abdillah, I. Kurniastuti, F. A. Susanto, and F. Yudianto, “Implementasi Black Box Testing dan Usability Testing pada Website Sekolah MI Miftahul Ulum Warugunung Surabaya,” *J. Comput. Sci. Vis. Commun. Des.*, vol. 8, no. 1, pp. 234–242, 2023, doi: 10.55732/jikdiskomvis.v8i1.897.