

Rancang Bangun Modul *Job Marketplace* di Aplikasi MyITS Connect Berdasarkan Onion Architecture dengan Paradigma Domain Driven Design

Nuralamsyah, Rizky Januar Akbar dan Hadziq Fabroyir

Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: rizky@if.its.ac.id

Abstrak—Program Merdeka Belajar yang dicetuskan oleh Menteri Pendidikan dan Kebudayaan membuat kemampuan lulusan perguruan tinggi semakin beragam. Banyaknya lulusan perguruan tinggi dengan kemampuan yang bervariasi tentunya menguntungkan dunia usaha dunia industri ketika mencari calon tenaga kerja baru. Namun, jika proses seleksi calon tenaga kerja dilakukan dengan proses manual dan konvensional, tentu proses pencariannya akan memakan banyak waktu. Institut Teknologi Sepuluh Nopember menyediakan sarana careers.its.ac.id dan bursa tenaga kerja untuk memudahkan dunia usaha dunia industri mencari calon tenaga kerja. Sayangnya fasilitas yang disediakan oleh careers.its.ac.id hanya berupa media informasi terkait lowongan yang tersedia. Selain itu, setiap kali pencari pekerjaan ingin mendaftar, mereka harus mengisi formulir dan mengunggah berkas di laman yang berbeda-beda sesuai dengan informasi lowongan. Untuk memudahkan pertemuan dunia usaha dunia industri dengan para pencari pekerjaan, maka myITS Connect Modul Job Marketplace dibuat sebagai tempat untuk mengadakan seleksi awal calon tenaga kerja. Melalui myITS Connect Modul Job Marketplace dunia usaha dunia industri dapat membuat lowongan yang mana nantinya lowongan tersebut dapat dilamar oleh pencari pekerjaan melalui myITS Connect Modul Job Marketplace pula. Selain itu, dunia usaha dunia industri juga dapat melihat profil sekilas dari pencari pekerjaan sehingga dapat menambah bahan pertimbangan demi mendapatkan tenaga kerja yang sesuai. Pembangunan myITS Connect Modul Job Marketplace dilakukan dengan menerapkan paradigma Domain Driven Design dan Onion Architecture dan berfokus pada platform web. Paradigma dan arsitektur tersebut dipilih karena keduanya mendukung kualitas *maintainability* utamanya *modularity* suatu aplikasi. Dengan kualitas *maintainability* yang tinggi, proses perbaikan dan perawatan dari perangkat lunak dapat dilakukan dengan lebih mudah. Untuk mengevaluasi nilai *maintainability*.

Kata Kunci—Domain Driven Design, Marketplace, Merdeka Belajar, myITS Connect, Onion Architecture.

I. PENDAHULUAN

MENTERI Pendidikan dan kebudayaan Nadiem Anwar Makarim mengatakan, tantangan dalam menjalankan bisnis bagi Badan Usaha Milik Negara (BUMN) dan perusahaan-perusahaan adalah memiliki sumber daya manusia yang unggul. Hal tersebut beliau sampaikan pada peluncuran Program Magang Mahasiswa Bersertifikat (PMMB) 2020 di Forum Human Capital Indonesia (FHCI) Assembly Hall Mandiri Sudirman pada hari Rabu, 12 Februari 2020. Program Kampus Merdeka merupakan salah

satu cara pemerintah dalam membantu dunia industri menghadapi tantangan tersebut. Program Kampus Merdeka berisi 4 poin utama yaitu otonomi bagi Perguruan Tinggi Negeri (PTN) dan Swasta (PTS) untuk melakukan pembukaan atau pendirian program studi (prodi) baru, program reakreditasi yang bersifat otomatis dan bersifat sukarela, kebebasan bagi PTN Badan Layanan Umum (BLU) dan Satuan Kerja (Satker) untuk menjadi PTN Badan Hukum (PTN BH), dan perpanjangan waktu magang hingga dua semester dan satu semester di luar program studi. Tiga dari empat poin tersebut yaitu kebebasan untuk menjadi PTN BH, pembukaan prodi baru, serta pemanjangan durasi magang berkaitan erat dengan dunia industri. Melalui program-program tersebut diharapkan lulusan diploma dan sarjana memiliki keterampilan yang sesuai dengan kebutuhan dunia kerja serta memiliki berbagai kemampuan yang mendukung di dunia industri.

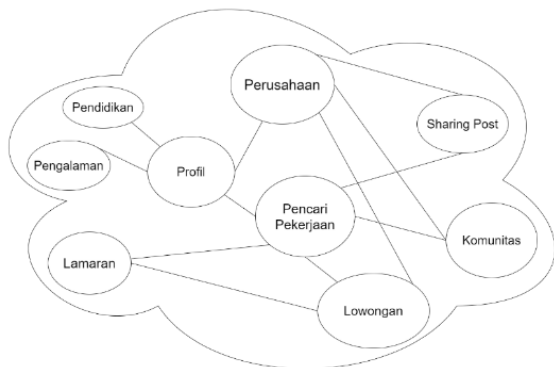
Dengan adanya program kampus merdeka, khususnya program pembukaan prodi baru dan pemanjangan durasi magang, kompetensi yang akan dimiliki oleh mahasiswa akan sangat beragam. Hal ini tentu sangat menguntungkan bagi perusahaan industri karena mereka dapat mencari tenaga kerja yang sesuai dengan kebutuhan mereka. Namun, setiap tahun, jumlah lulusan perguruan tinggi di Indonesia dapat mencapai satu juta jiwa. Mencari tenaga kerja yang spesifik dari banyaknya pilihan yang ada tentu sangat memakan waktu jika proses seleksi dilakukan secara konvensional. Hal tersebut menjadikan proses perekrutan tenaga kerja baru menjadi kurang efisien dan efektif

Berangkat dari hal tersebut, ditawarkan suatu aplikasi yang dapat memudahkan proses seleksi bagi perusahaan yaitu myITS Connect Modul Job Marketplace. myITS Connect Modul Job Marketplace merupakan modul dari myITS Connect, yang di dalamnya pihak yang membutuhkan tenaga kerja dapat melihat berbagai informasi para pencari pekerjaan yang berkaitan dengan dunia usaha dan dunia industri seperti kemampuan dan pengalaman yang dimiliki. Mahasiswa juga dapat memberikan informasi seputar riwayat pendidikan sehingga memudahkan industri dalam menjaring tenaga kerja. Dengan demikian, perusahaan industri dapat lebih mudah dalam melakukan seleksi calon tenaga kerja baru. Diharapkan dengan adanya myITS Connect Modul Job Marketplace, secara tidak langsung kualitas lulusan dapat ditingkatkan.

Pada Penelitian ini, paradigma Domain Driven Design dan Onion Architecture digunakan untuk membangun Modul Job

Tabel 1.
Daftar kebutuhan pengguna

Kode	Kebutuhan Fungsional	Deskripsi
F01	Mengelola lowongan perusahaan	Memfasilitasi perusahaan untuk mengelola lowongan dalam bentuk membuat lowongan, mempublish lowongan, melihat calon pekerja yang mendaftar, melihat formulir lamaran pendaftar, dan menutup lowongan.
F02	Melayani proses pencarian tenaga kerja	Memfasilitasi perusahaan untuk mencari calon tenaga kerja dengan menampilkan daftar calon tenaga kerja yang terdaftar dalam sistem dan menampilkan informasi umum dari tenaga kerja yang menarik perhatian.
F03	Melayani proses pencarian pekerjaan	Memfasilitasi pencari pekerjaan untuk melihat daftar lowongan beserta informasi detail, mendaftar lowongan, dan menampilkan informasi penerimaan.



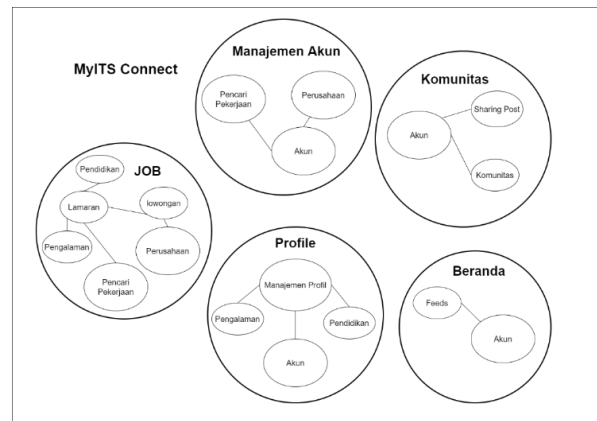
Gambar 3. Gambaran kasar Domain MyITS Connect.

Marketplace Aplikasi myITS Connect. Paradigma Domain Driven Design dan Onion Architecture digunakan karena keduanya mendukung modularitas aplikasi. Konsep modularitas menjadikan suatu aplikasi yang besar dipecah menjadi modul-modul kecil yang dapat saling berinteraksi melalui antarmuka modul. Dengan memecah aplikasi menjadi modul-modul kecil, kesalahan pada suatu modul tidak memberikan dampak yang besar kepada modul lain sehingga proses perbaikan dan pengembangan dapat lebih mudah dilakukan. Konsep modularitas juga sejalan dengan atribut kualitas maintainability. Tingginya nilai atribut kualitas maintainability menandakan bahwa aplikasi tersebut mudah untuk diperluas fungsinya atau diperbaiki galatnya. Alasan lain myITS Connect Modul Job Marketplace dibangun dengan mengedepankan modularitas karena pembangunan modul-modul lain yang ada pada myITS Connect juga mengedepankan konsep modularitas.

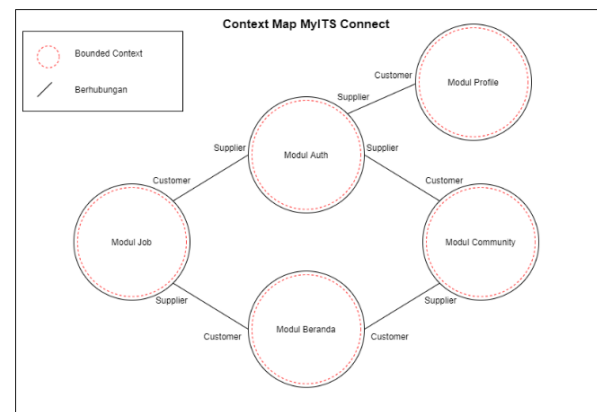
II. TINJAUAN PUSTAKA

A. Domain Driven Design

Menurut KBBI, domain memiliki makna wilayah atau ranah [1]. Dalam konteks pengembangan aplikasi, domain

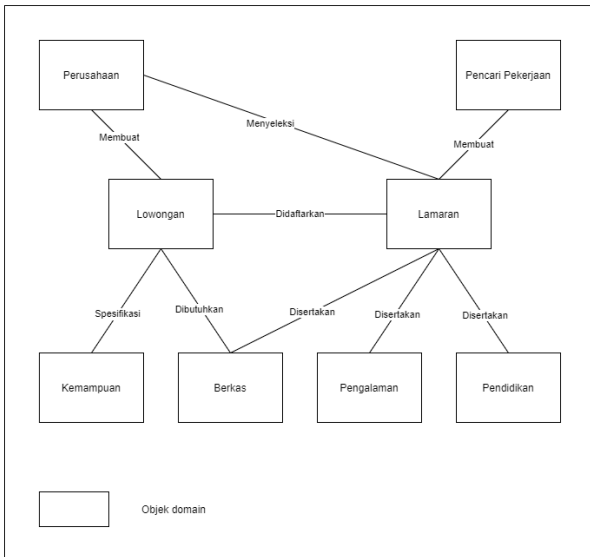


Gambar 1. Gambaran pembagian subdomain MyITS Connect.



Gambar 2. Context map myITS Connect.

adalah area atau hal yang menjadi fokus dari suatu aplikasi. Paradigma *Domain Driven Design* atau disingkat *DDD* adalah suatu metode untuk pengembangan aplikasi yang memfokuskan implementasi sistem pada penyelesaian masalah-masalah yang dihadapi dalam proses bisnis. Proses pengembangan dengan menggunakan *DDD* akan menghasilkan suatu model dari domain aplikasi yang dapat terus berkembang sesuai dengan kebutuhan. Dengan menggunakan *DDD*, diharapkan proses pengembangan aplikasi yang bersifat kompleks menjadi lebih mudah untuk dirawat dan dikembangkan. Dalam menerapkan paradigma *DDD*, terdapat dua tahap *design* yang penting yaitu *strategic design* dan *tactical design*. Tujuan dari proses *strategic design* adalah untuk menghasilkan *ubiquitous language*, *bounded context*, dan *context map* yang nantinya akan menjadi *guideline* bagi pengembang dalam membangun aplikasi atau modul. *Ubiquitous language* adalah kumpulan istilah yang digunakan untuk menjelaskan berbagai hal dalam domain yang dipahami baik oleh pengembang maupun *domain expert* sehingga memudahkan mereka untuk berkolaborasi [2]. *Bounded context* adalah suatu pembatas imajiner dalam suatu domain bisnis atau kode yang berfungsi untuk mengelompokkan komponen-komponen dan metode-metode terkait guna menghindari ambiguitas pemaknaan. *Context Map* membantu tim pengembang utamanya dalam memahami *bounded context* yang ada dalam domain serta relasi antar *bounded context* pada domain tersebut. Tahap *tactical design* adalah tahap dimana proses pembangunan model domain dilakukan, dengan menggunakan komponen-komponen yang telah didefinisikan oleh *DDD*. Beberapa komponen yang didefinisikan oleh



Gambar 5. Gambaran model domain modul job.

tactical design adalah *entity*, *aggregates*, *value object*, dan *repositories*. *Entity* adalah objek dalam suatu domain yang memiliki atribut identitas (contoh ID). *Value object* adalah objek dalam domain yang tidak memiliki atribut identitas. *Aggregates* umumnya adalah kumpulan *entity* maupun *value object* yang bisa dianggap sebagai satu unit [2]. *Repositories* merupakan istilah dalam DDD untuk komponen yang berkaitan erat dengan *storage* / penyimpanan data.

B. Onion Architecture

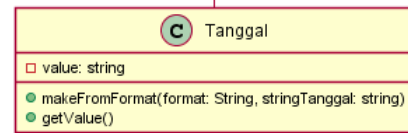
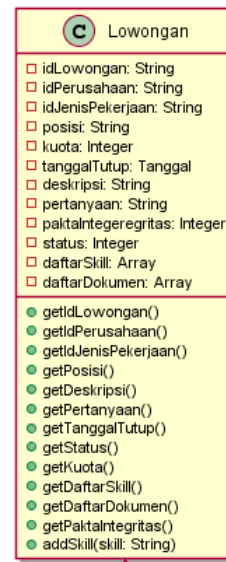
Onion Architecture adalah suatu arsitektur perangkat lunak yang terdiri dari beberapa lapisan konsentris yang hanya bergantung pada lapisan yang lebih dalam. Onion Architecture menerapkan prinsip *dependency inversion*, yaitu modul / komponen yang lebih tinggi tidak boleh bergantung pada komponen yang lebih rendah. Pada Onion Architecture, umumnya terdapat beberapa lapisan atau *layer* yaitu *Application Service Layer*, *Domain Layer*, *Infrastructure Layer*, dan *User Interface Layer*. *User Interface Layer* terletak pada bagian paling luar dari Onion Architecture. Lapisan ini berfungsi untuk menampilkan tampilan grafis pada pengguna. *Application Service Layer* berfungsi sebagai media komunikasi antara *User Interface Layer* dengan *Domain Layer*. *Domain Layer* menjadi inti dari *Onion Architecture*. *Domain Layer* mengandung berbagai objek bisnis dan perilakunya

C. PHP

Dikutip dari situs dokumentasi bahasa PHP, PHP adalah bahasa pemrograman *open source* yang dapat digunakan di berbagai hal khususnya pada bidang pengembangan aplikasi web. Sampai saat ini bahasa pemrograman PHP masih banyak digunakan utamanya untuk membangun aplikasi berbasis web. PHP juga memiliki banyak *library-library* atau *file-file* pendukung yang menambah fungsionalitas PHP. Versi PHP yang digunakan penulis adalah PHP 7.4.

D. Laravel

Laravel adalah PHP *web framework* gratis dan *open source* yang banyak dibuat oleh Taylor Otwell. Laravel menyediakan berbagai fitur-fitur yang berkaitan dengan pengembangan web yang memudahkan pengembang dalam



Gambar 4. Aggregate lowongan.

membangun aplikasi berbasis web. Versi Laravel yang digunakan oleh penulis adalah Laravel 8 .

E. McCabe's Cyclomatic Complexity

Cyclomatic Complexity adalah metrik untuk menghitung jumlah maksimum jalur independen linear dalam *graph* kontrol program [3]. *Cyclomatic Complexity* berguna untuk menentukan nilai *maintainability* suatu perangkat lunak. Rumus untuk menghitung *cyclomatic complexity* adalah sebagai berikut.

$$v(G) = e - n + 2p$$

Dengan G adalah *graph* dari program yang dimaksud, e adalah jumlah *edges* pada *graph*, n adalah jumlah *nodes* pada *graph*, dan p adalah jumlah *component* pada *graph*. Untuk perhitungan *cyclomatic complexity* pada Penelitian ini, digunakan nilai batas / *threshold* 10 dimana nilai *complexity* program dianggap baik jika di bawah 10.

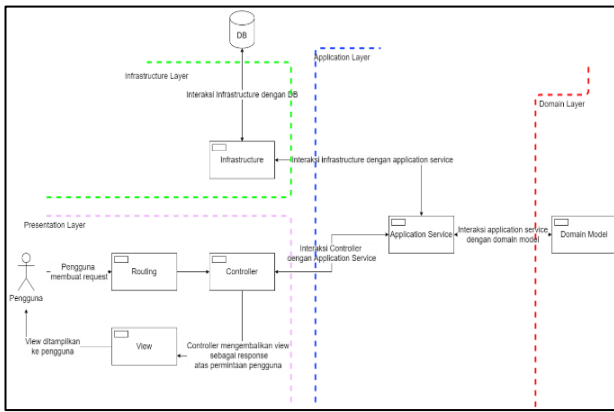
III. ANALISIS DAN PERANCANGAN SISTEM

A. Daftar Kebutuhan Pengguna

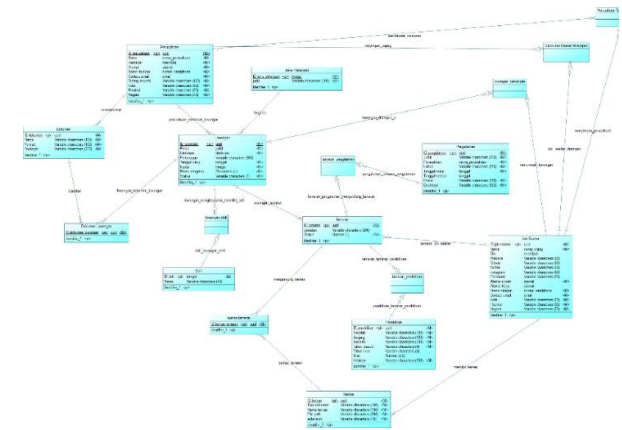
Kebutuhan fungsional mendefinisikan fitur yang harus difasilitasi oleh perangkat lunak. Kebutuhan fungsional dari perangkat lunak untuk aplikasi ini dijelaskan pada Tabel 1.

B. Strategic Design

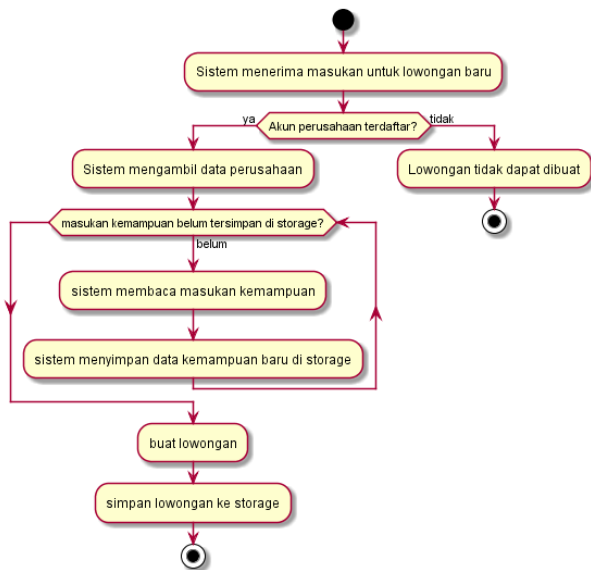
Pada tahap ini, tim pengembang menentukan *bounded context* dari aplikasi myITS Connect. Pada proses *strategic design*, pengembang melakukan diskusi dengan *domain expert* seperti pihak SAC ITS untuk membahas lebih lanjut terkait domain dari aplikasi yang dibangun. Dari hasil diskusi awal didapatkan gambaran kasar dari domain myITS Connect yang tertera pada Gambar 1. Penjelasan dari gambaran kasar domain MyITS Connect sebagai berikut :



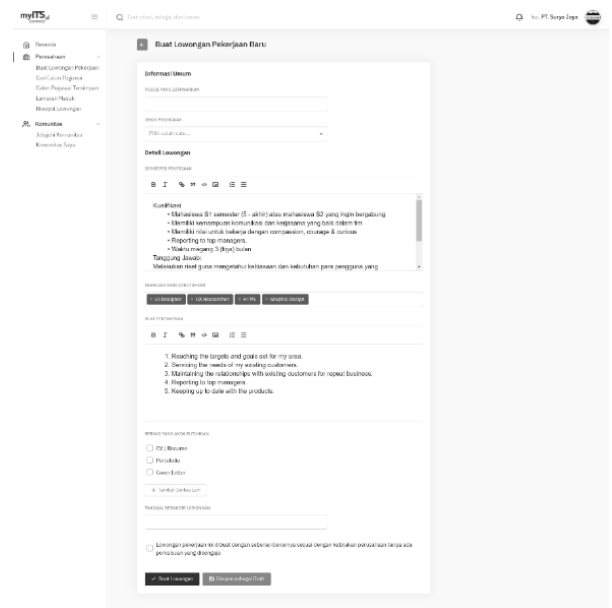
Gambar 8. Gambaran arsitektur modul job myITS Connect.



Gambar 6. CDM modul job marketplace myITS Connect.



Gambar 9. Rancangan algoritma membuat lowongan.



Gambar 7. Rancangan antarmuka laman membuat lowongan.

1. Perusahaan adalah jenis pengguna yang merepresentasikan para pencari tenaga kerja.
2. Pencari pekerjaan adalah jenis pengguna yang merepresentasikan para pencari pekerjaan.
3. Baik perusahaan maupun pencari pekerjaan memiliki profil, namun isinya berbeda. Pada profil pencari pekerjaan terdapat bagian pengalaman dan pendidikan yang berfungsi sebagai bahan pertimbangan saat seleksi oleh perusahaan.
4. Perusahaan berhak membuat lowongan, sedangkan pencari pekerjaan hanya boleh melihat lowongan dan melamar selama lowongan masih dibuka.
5. Baik perusahaan maupun pencari tenaga kerja berhak untuk membagikan *post* atau *feed*.

Dengan mempertimbangkan modularitas dari aplikasi dimana setiap modul diharapkan menerapkan *Single Responsibility Principle (SRP)* yaitu suatu modul harus hanya memiliki satu fungsi tertentu maka dibentuk pembagian *domain* menjadi *subdomain* tertera pada Gambar 2. Penjelasan dari Gambar 2 secara detail sebagai berikut :

1. Subdomain Manajemen Akun bertugas untuk mengatur segala hal yang berkaitan dengan akun pengguna (pendaftaran, autentikasi, dan lain sebagainya).
2. Subdomain *Job* bertugas untuk memfasilitasi segala kebutuhan perangkat lunak yang berkaitan dengan lowongan pekerjaan dan pencarian tenaga kerja.
3. Subdomain komunitas berkaitan dengan membuat komunitas dan berbagi hal melalui *post/feed*.

4. Subdomain Profile berkaitan dengan pengelolaan profil pengguna.
5. Subdomain Beranda bertugas untuk menampilkan informasi-informasi terbaru dari subdomain Komunitas maupun subdomain *Job*.

Masing-masing subdomain pada myITS Connect kemudian diubah menjadi modul yang memiliki *bounded context* dan *ubiquitous language* masing-masing. Subdomain *Job* menjadi Modul *Job*, Subdomain Komunitas menjadi Modul *Community*, Subdomain Profile menjadi Modul Profile, Subdomain Manajemen Akun menjadi Modul *Auth*, dan Subdomain Beranda menjadi Modul Beranda. Modul-modul tersebut jika digambarkan dalam suatu *context map* dapat dilihat pada Gambar 3.

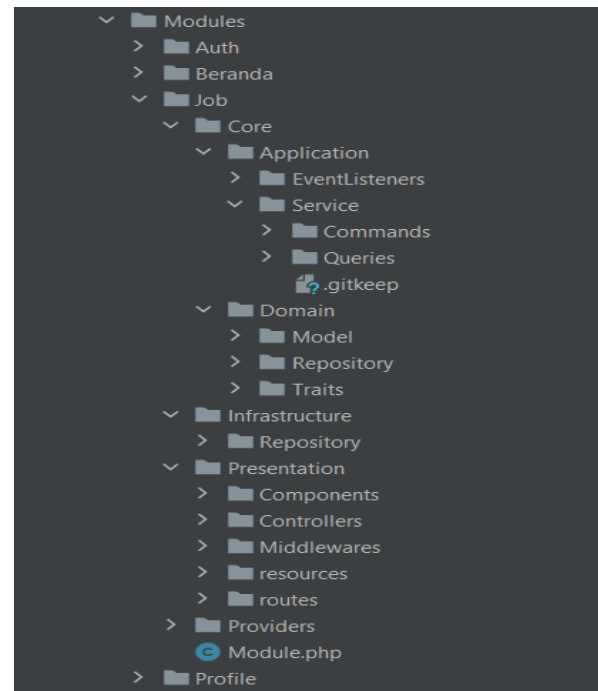
Terdapat hubungan diantara *bounded context* pada myITS Connect. Modul *Job* dan Modul *Community* memiliki hubungan dengan modul *Auth* dengan tipe hubungan *supplier-customer*. Begitu pula modul Beranda dengan modul *Job* dan Modul *Community*. Hubungan *supplier-customer* dipilih karena jenis hubungan tersebut difasilitasi oleh *framework* Laravel.

Berikutnya penulis akan lebih membahas terkait modul *Job* karena fokus penulis pada Penelitian ini adalah modul *Job*. Beberapa *ubiquitous language* pada *bounded context* yang didapatkan dari proses *strategic design* untuk modul *Job* adalah sebagai berikut:

Tabel 2.

Penjelasan rancangan antarmuka halaman buat lowongan pekerjaan

No	Nama Atribut Antarmuka	Jenis	Keterangan
1	Input posisi yang ditawarkan	Input field	Memasukkan posisi yang ditawarkan dalam lowongan
2	Dropdown jenis pekerjaan	Dropdown	Memilih jenis pekerjaan dari posisi yang dilowongkan
3	Input deskripsi pekerjaan	Input field	Memasukkan deskripsi dari posisi yang ditawarkan
4	Dropdown keahlian yang dibutuhkan	Dropdown	Memilih keahlian yang dibutuhkan untuk posisi yang dilowongkan
5	Input pertanyaan	Input field	Memasukkan pertanyaan yang harus dijawab oleh pelamar
6	Checkbox berkas	Checkbox	Memilih berkas yang harus disertakan oleh pelamar
7	Input tanggal berakhir lowongan	Input field	Memasukkan tanggal dimana lowongan ditutup.
8	Checkbox pakta integritas	Checkbox	Menyatakan bahwa lowongan yang dibuat siap dipertanggungjawabkan



Gambar 10. Struktur folder myITS Connect.

- 1) Lowongan adalah objek yang berisi informasi terkait perekrutan tenaga kerja.
- 2) Lowongan hanya dibuat oleh perusahaan.
- 3) Lowongan yang telah dipublikasikan dapat ditutup oleh perusahaan.
- 4) Lamaran adalah objek yang berisi informasi terkait calon tenaga kerja yang bersedia mengikuti seleksi dari suatu proses perekrutan tenaga kerja.
- 5) Lamaran dapat diterima maupun ditolak.

C. Tactical Design

Setelah menghasilkan *context map* dari proses *strategic design*, penulis masuk ke tahap *tactical design*. Untuk tahap ini, penulis berfokus pada *bounded context* Modul *Job*. Dengan mempertimbangkan kebutuhan fungsional yang harus dipenuhi oleh modul *Job* dan memperhatikan batasan *bounded context* (segala komponen harus berkaitan dengan proses pengelolaan lowongan dan pencarian tenaga kerja) maka dihasilkan model domain yang akan digunakan sebagai acuan dalam implementasi. Gambar 4 merupakan gambaran model domain dari *bounded context* Modul *Job*.

Pada modul *job*, terdapat dua objek aktor yaitu perusahaan dan pencari pekerjaan. Perusahaan dapat membuat lowongan dan pencari pekerjaan yang ingin mendaftar lowongan tersebut harus membuat lamaran. Pada lowongan, terdapat daftar kemampuan yang harus dimiliki dimana daftar tersebut dapat menjadi bahan pertimbangan bagi seorang pencari pekerjaan untuk melamar atau tidak. Selain daftar kemampuan, lowongan juga mengharuskan pelamar menyertakan beberapa berkas pendukung yang disertakan pada lamaran. Ketika membuat lamaran, pencari pekerjaan dapat menyertakan beberapa riwayat pengalaman dan pendidikan sehingga dapat menjadi bahan pertimbangan perusahaan saat menyeleksi pelamar.

Demi mewujudkan domain model tersebut, beberapa komponen penting *tactical design* DDD digunakan. Komponen-komponen tersebut adalah *aggregate*, *value object*, dan *repositories*. Contoh *aggregate* yang digunakan dalam modul *Job* adalah *aggregate* Lowongan. *Aggregate*

terdiri dari *entity* lowongan yang mengandung *value object* tanggal. *Aggregate* lowongan tertera pada Gambar 5.

D. Perancangan Arsitektur

Ketika pengguna *request* data, *input* akan diterima oleh *routing* kemudian diarahkan ke *controller*. *Controller* kemudian memanggil *application service* yang sesuai untuk menjawab permintaan pengguna. *Application service* kemudian melakukan interaksi baik dengan *domain model* untuk menyelesaikan proses bisnis atau dengan *infrastructure* jika berkaitan dengan proses penyimpanan data. Hasil dari eksekusi *application service* dikembalikan ke *controller*. *Controller* kemudian memberikan *response* kepada pengguna melalui *view*. Gambar 6 merupakan gambaran arsitektur model *job* MyITS Connect.

Dalam arsitektur yang tertera pada Gambar 6, sesuai dengan Onion Architecture, *domain model* berada di dalam *domain layer* (lapisan paling dalam), *application service* berada di dalam *application layer* (jembatan antara *domain layer* dengan *infrastructure* dan *presentation layer*), *infrastructure* berada pada *infrastructure layer* (lapisan paling luar), dan *controller*, *routing*, *view* berada pada *presentation layer* (lapisan paling luar).

E. Perancangan Algoritma

Pada bagian ini dibahas rancangan algoritma untuk salah satu kasus penggunaan yaitu membuat lowongan. Gambar 7 merupakan gambar flowchart dari proses pembuatan lowongan.

F. Perancangan Basis Data

Pada bagian ini dibahas rancangan basis data. Gambar 8 merupakan CDM modul *job* marketplace MyITS Connect.

G. Perancangan Antarmuka

Gambar 9 merupakan contoh rancangan antarmuka myITS Connect Modul *Job*. Penjelasan rancangan antarmuka halaman untuk lowongan dijelaskan dalam Tabel 2.

```

<?php
namespace App\Modules\Job\Core\Domain\Model;

class Lowongan
{
    private $idLowongan;
    private $idJenisPekerjaan;
    private $posisi;
    private $kuota;
    private $tanggalTutup;
    private $deskripsi;
    private $pertanyaan;
    private $paktaIntegritas;
    private $status;
    private $daftarSkill;
    private $daftarDokumen;

    public function __construct (...){
        ...
    }
    ...
    public function getStatus (){
        return $this->status;
    }
    ...
}
    
```

Gambar 11. Potongan kode kelas lowongan.

```

<?php
namespace
App\Modules\Job\Core\Application\Service\Commands\Vacancy\AddVacancy
....
class AddVacancyService
{
    private $jobRepository;

    public function __construct (JobRepository $jobRepository)
    {
        ...
    }
    ...
    public function execute (AddVacancyRequest $request)
    {
        $perusahaan = $this->jobRepository->getPerusahaan ($request->idPerusahaan);

        if ($perusahaan == null) throw new \Exception ('Perusahaan yang dimaksud tidak terdaftar. ');
        ...
        $lowongan = $perusahaan->makeLowongan (
            ...
        );
        ...
        $idLowonganBaru = $this->jobRepository->addVacancy ($lowongan);
        return $idLowonganBaru;
    }
}
    
```

Gambar 12. Potongan kode AddVacancyService.

IV. IMPLEMENTASI

A. Implementasi Struktur Proyek Laravel

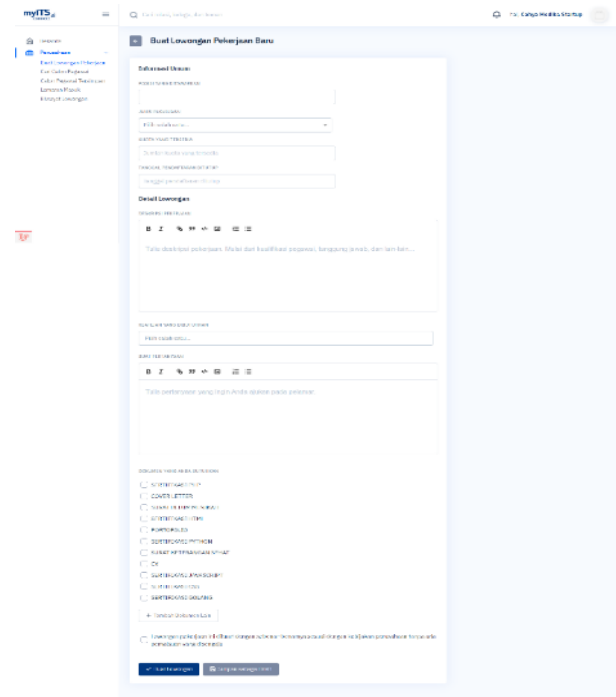
Dengan menggunakan Onion Architecture sebagai acuan, maka bentuk struktur *directory* dari myITS Connect Modul *Job* tertera pada Gambar 10.

B. Implementasi Pengetahuan Domain

Pada bagian ini, dijabarkan bagaimana mengimplementasikan pengetahuan domain ke dalam bagian kode. Pada Gambar 11 merupakan potongan kode kelas *lowongan* akan dijadikan contoh implementasi.

C. Implementasi Kasus Penggunaan

Implementasi kasus penggunaan dalam kode juga mengikuti *ubiquitous language* yang telah didiskusikan dengan *domain expert*. Di bagian ini, kasus penggunaan untuk membuat lowongan akan digunakan sebagai contoh. Proses pembuatan lowongan dimulai dengan sistem



Gambar 13. Implementasi antarmuka membuat lowongan.

Tabel 3. Hasil Pengujian Kasus Penggunaan Sisi Perusahaan

Kode	Kasus Penggunaan	Terpenuhi	
		YA	TIDAK
UJ-01	Membuat lowongan	✓	
UJ-02	Melihat detail lowongan yang dibuat	✓	
UJ-03	Melihat daftar pelamar	✓	
UJ-04	Melihat detail pelamar	✓	
UJ-05	Mengubah status pelamar	✓	
UJ-06	Menutup lowongan	✓	
UJ-07	Melihat daftar pencari pekerjaan yang terdaftar di myITS Connect	✓	
UJ-08	Melihat profil sekilas dari pencari pekerjaan yang terdaftar di myITS Connect	✓	
UJ-09	Menyimpan pencari pekerjaan yang terdaftar di myITS Connect	✓	
UJ-10	Melihat daftar pencari pekerjaan yang disimpan	✓	
UJ-11	Melihat daftar riwayat lowongan yang pernah dibuat	✓	

memeriksa apakah ID perusahaan sudah terdaftar dalam sistem. Jika perusahaan terdaftar, sistem akan menambahkan *skills* yang belum ada ke dalam *storage* (dalam hal ini *database*). Kemudian, sistem akan membuat objek lowongan berdasarkan input yang dimasukkan oleh pengguna. Terakhir sistem menyimpan lowongan yang baru saja dibuat ke dalam *storage*. Potongan kode *AddVacancyService* tertera pada Gambar 12.

D. Implementasi Antarmuka

Berikut contoh implementasi antarmuka untuk membuat lowongan yang tertera pada Gambar 13.

V. PENGUJIAN DAN EVALUASI

Bagian ini membahas tentang uji coba dan evaluasi terhadap perangkat lunak yang telah dikembangkan dari implementasi kasus penggunaan

Tabel 4.

Hasil pengujian kasus penggunaan sisi pencari pekerjaan			
Kode	Kasus Penggunaan	Terpenuhi	
		YA	TIDAK
UJ-12	Melihat daftar lowongan	✓	
UJ-13	Melihat detail lowongan	✓	
UJ-14	Melamar lowongan	✓	
UJ-15	Menyimpan lowongan	✓	
UJ-16	Melihat daftar lowongan yang disimpan	✓	
UJ-17	Melihat daftar lamaran yang pernah dibuat	✓	
UJ-18	Melihat detail lamaran yang pernah dibuat	✓	
UJ-19	Melihat daftar perusahaan yang terdaftar pada myITS Connect	✓	
UJ-20	Mengikuti perusahaan yang terdaftar pada myITS Connect	✓	

Tabel 5.

Hasil pengukuran cyclomatic complexity			
Modul	Banyak Kelas	Total semua kelas	Nilai
<i>Job</i>	5	73	0,931507

A. Pengujian Kasus Penggunaan

Pengujian kasus penggunaan memiliki 2 skenario utama yaitu pengujian kasus penggunaan sisi perusahaan dan pengujian kasus penggunaan sisi pencari pekerjaan. Rekapitulasi hasil pengujian kasus penggunaan tertera pada Tabel 3. Sedangkan hasil pengujian kasus penggunaan sisi pencari pekerjaan tertera pada Tabel 4.

B. Pengukuran Kualitas Perangkat Lunak

Untuk pengukuran kualitas perangkat lunak digunakan *Cyclomatic Complexity (MMo-2-S)* untuk mengukur kualitas *maintainability* perangkat lunak, dengan menggunakan nilai 10 sebagai batas. Hasil perhitungan *Cyclomatic Complexity* dengan bantuan PHPMetrics tertera pada Tabel 5.

VI. KESIMPULAN

Bab ini membahas mengenai kesimpulan yang diperoleh selama pengerjaan penelitian dan saran mengenai pengembangan yang dapat dilakukan terhadap Penelitian ini di masa yang akan datang.

A. Kesimpulan

Berdasarkan hasil uji coba perangkat lunak yang dibahas dalam Penelitian ini, penulis mengambil kesimpulan sebagai berikut, proses desain proses bisnis myITS Connect Modul *Job* menggunakan paradigma Domain Driven Design diawali dengan proses *strategic design* yang menghasilkan *bounded context*, *ubiquitous language*, dan *context map* kemudian dilanjutkan dengan *tactical design* yang menghasilkan model domain yang dijadikan acuan implementasi, sebagaimana yang dibahas pada bab 3.

Proses implementasi myITS Connect Modul *Job* menggunakan Onion Architecture berhasil dilakukan menggunakan *framework laravel* dan bahasa PHP dengan mengikuti acuan dari *domain model*, sebagaimana yang dibahas pada bab 4.

Modul *Job* myITS Connect mendapatkan nilai *maintainability cyclomatic complexity* sebesar 0,931507 sehingga dapat dikatakan modul *Job* MyITS Connect memiliki nilai *maintainability* yang tinggi, sebagaimana yang tertera pada bab 5.

B. Saran

Saran yang dapat diberikan guna meningkatkan hasil Tugas Akhir ini antara lain proses autentikasi myITS Connect dapat lebih dikembangkan dengan menerapkan myITS SSO pada modul *Auth*. Mekanisme penyimpanan file dapat lebih dikembangkan dengan menggunakan API Storage ITS. Disarankan untuk melakukan pengembangan integrasi data myITS Student Connect dengan data pengalaman myITS Connect.

DAFTAR PUSTAKA

- [1] Departemen Pendidikan Nasional, *Kamus Besar Bahasa Indonesia*. Jakarta: Pusat Bahasa, 2008.
- [2] E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston: Addison Wesley, 2003.
- [3] A. H. Watson, D. R. Wallace, and T. J. McCabe, *Structured Testing: A Testing Methodology Using The Cyclomatic Complexity Metric*. Columbia: US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1996.