

IMPLEMENTASI DOMAIN DRIVEN DESIGN DAN CLEAN ARCHITECTURE DALAM PENGEMBANGAN WEB SERVICE APLIKASI ALIFARM DIGITAL

Oleh:

Rama Sakti Hafidz Fadhilah Aziz,

Irwan Alnarus Kautsar

Informatika

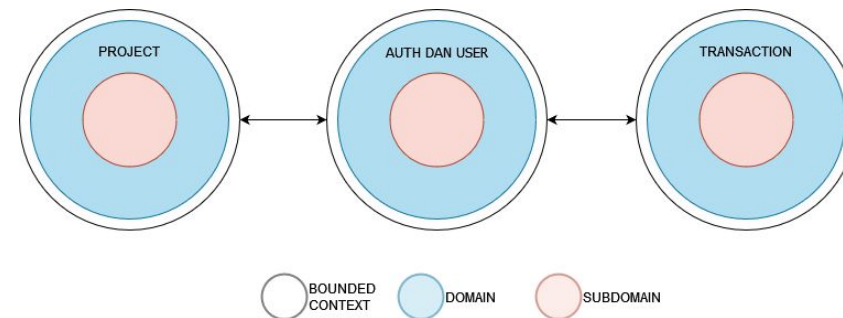
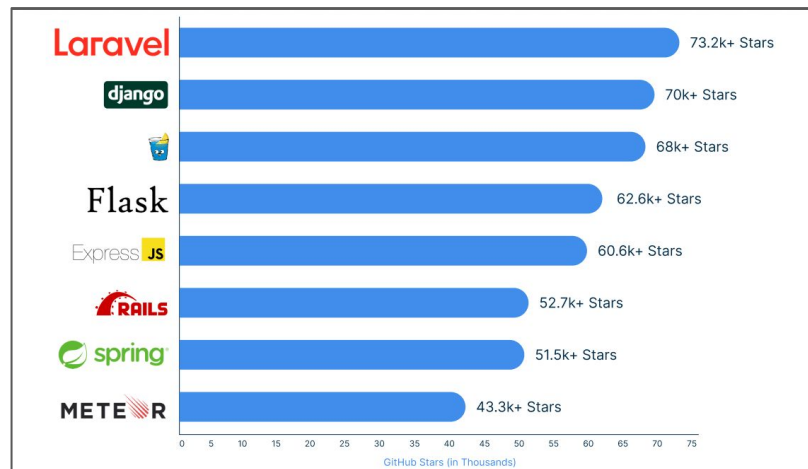
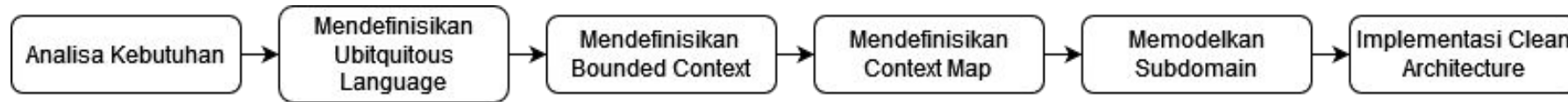
Universitas Muhammadiyah Sidoarjo

Mei 2024

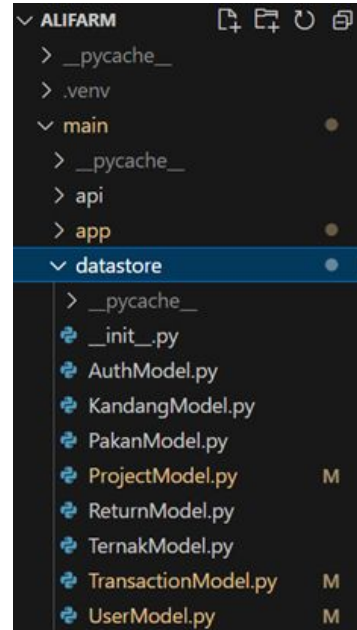
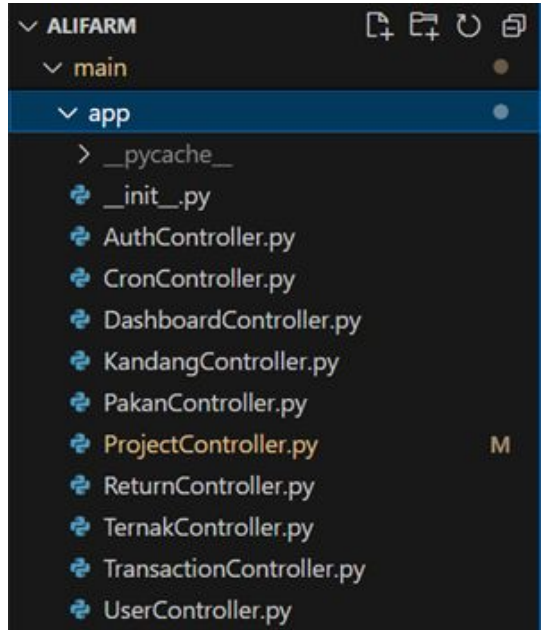
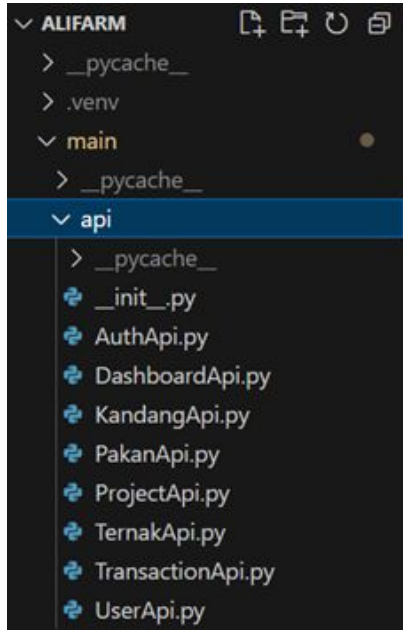
Pertanyaan Penelitian (Rumusan Masalah)

Bagaimana implementasi *Domain Driven Design* dan *Clean Architecture* dalam Pengembangan Web Service Aplikasi Alifarm Digital?

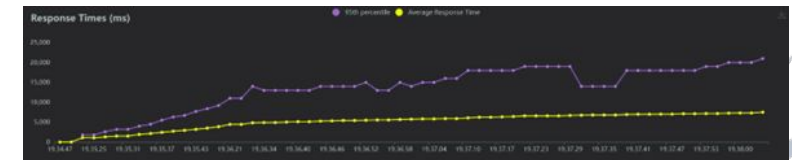
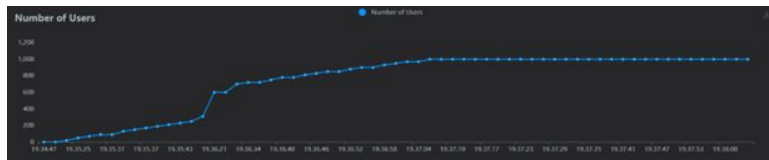
Metode



Hasil



```
ProjectApi.py X
main > api > ProjectApi.py > -
1 from ..app import *
2
3 @app.route('/project', methods=['GET'])
4 async def project():
5     return await ProjectController.project()
6
7 @app.route('/project/all', methods=['GET'])
8 async def allProjects():
9     return await ProjectController.allProjects()
10
11 @app.route('/project/invest/<id_project>', methods=['GET'])
12 async def projectInvest(id_project):
13     return await ProjectController.projectInvest(id_project)
14
15 @app.route('/project/create', methods=['GET'])
16 async def createProject():
17     return await ProjectController.create()
18
19 @app.route('/project/store', methods=['POST'])
20 async def storeProject():
21     return await ProjectController.store()
22
23 @app.route('/project/detail/<id_project>')
24 async def detailProject(id_project):
25     return await ProjectController.detail(id_project)
26
27 @app.route('/project/status/pending', methods=['GET'])
28 async def pending():
29     return await ProjectController.pending()
```



Pembahasan

- Pemisahan route merupakan bagian dari penerapan Clean Architecture, di mana setiap route bertanggung jawab untuk menangani permintaan (request) dari klien dan mengarahkan permintaan tersebut ke controller yang sesuai.
- Domain pada web service aplikasi Alifarm Digital dibagi menjadi 3 domain dengan masing masing domain memiliki beberapa subdomain.
- Dari 3 pengujian yang dilakukan dengan lingkungan pengujian yang relatif minimum, aplikasi Alifarm Digital kesulitan dalam menangani beban yang tinggi.

Temuan Penting Penelitian

- Dengan mengimplementasikan Domain Driven Design pengembangan aplikasi secara kolektif lebih mudah dan lebih cepat
- Dengan mengimplementasikan Clean Architecture, kode program terlihat lebih ringkas dengan masing masing file kode program hanya 20 - 180 baris kode
- Lonjakan waktu respons pada persentil ke-99 yang tinggi dan fluktuasi dalam jumlah request yang gagal
- Grafik request per second yang menunjukkan peningkatan tajam dalam jumlah total request per detik

Manfaat Penelitian

- Dengan diimplementasikannya Domain Driven Design dalam pengembangan web service Alifarm Digital dapat mempermudah dan mempercepat proses pengembangannya.
- Dengan diimplementasikannya Clean Architecture pengembangan web service Alifarm Digital dapat mempermudah programmer dalam memelihara sistem.
- Dengan adanya web service aplikasi Alifarm Digital dapat membantu peternak dalam memperoleh modal dari investor.
- Dengan adanya web service aplikasi Alifarm Digital dapat membantu masyarakat umum dalam berinvestasi dengan akad syariah dan terhindar dari penipuan.

Referensi

- [1] A. Deljouyi and R. Ramsin, "MDD4REST: Model-Driven Methodology for Developing RESTful Web Services," presented at the 10th International Conference on Model-Driven Engineering and Software Development, May 2024, pp. 93–104. Accessed: May 28, 2024. [Online]. Available: <https://www.scitepress.org/Link.aspx?doi=10.5220/0011006300003119>
- [2] Q. Li, W. Sun, and R. Ma, "Sharing platform of digital specimen of wood canker based on WebGIS in Xinjiang province: architecture, design and implementation," in 2022 International Conference on Computers, Information Processing and Advanced Education (CIPAE), Aug. 2022, pp. 102–106. doi: 10.1109/CIPAE55637.2022.00029.
- [3] E. A. Ishlahuddin, R. J. Akbar, and H. Fabroyir, "Rancang Bangun Modul Komunitas di Aplikasi MyITS Connect Berdasarkan Onion Architecture dengan Paradigma Domain Driven Design," JTITS, vol. 10, no. 2, pp. A201–A208, Dec. 2021, doi: 10.12962/j23373539.v10i2.66674.
- [4] D. Alulema, J. Criado, L. Iribarne, A. J. Fernández-García, and R. Ayala, "SI4IoT: A methodology based on models and services for the integration of IoT systems," Future Generation Computer Systems, vol. 143, pp. 132–151, Jun. 2023, doi: 10.1016/j.future.2023.01.023.
- [5] K. E. Blond, A. L. Clark, and T. H. Bradley, "A Decision-Making Framework for the KC-46A Maintenance Program," in 2023 Annual Reliability and Maintainability Symposium (RAMS), Jan. 2023, pp. 1–6. doi: 10.1109/RAMS51473.2023.10088176.
- [6] A. Chessa et al., "Data-Driven Methodology for Knowledge Graph Generation Within the Tourism Domain," IEEE Access, vol. 11, pp. 67567–67599, 2023, doi: 10.1109/ACCESS.2023.3292153.
- [7] A. D. Prasetyo, I. A. Kautsar, and N. L. Azizah, "Rancang Bangun Aplikasi Pelaporan Fasilitas Umum Berbasis Web Service Dalam Rangka Menuju Sidoarjo Smart City Dan Open Data," jipi. jurnal. ilmiah. penelitian. dan. pembelajaran. informatika., vol. 7, no. 4, pp. 1271–1280, Nov. 2022, doi: 10.29100/jipi.v7i4.3259.

- [8] D. Mufidah and H. Setiawan, "Analisis Framing Berita Nasib Aset Indra Kenz Akibat Kasus Binomo Media Detik dan Tirto," vol. 6, 2022.
- [9] S. Darma, "Crowdfunding Pada Teknologi Keuangan Islam," *SOSMANIORA: Jurnal Ilmu Sosial dan Humaniora*, vol. 1, no. 2, Art. no. 2, Jun. 2022, doi: 10.55123/sosmaniora.v1i2.441.
- [10] C. Setya Budi, "Implementasi Arsitektur Microservices Pada Backend Comrades," diploma, Universitas Komputer Indonesia, 2018. Accessed: May 28, 2024. [Online]. Available: <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-cahyantose-40046>
- [11] M. F. Ramadhan and Z. Zukhri, "PENGEMBANGAN REST API SISTEM UIIADMISI DENGAN MENGGUNAKAN PENDEKATAN DOMAIN DRIVEN DESIGN," *JURNAL ILMIAH INFORMATIKA*, vol. 11, no. 02, Art. no. 02, Sep. 2023, doi: 10.33884/jif.v11i02.8017.
- [12] H. W. Prayoga, R. J. Akbar, and H. Fabroyir, "Rancang Bangun Sistem MyITS Dorm Menggunakan Metode Domain Driven Design dan Onion Architecture," *JTITS*, vol. 10, no. 2, pp. A298–A305, Dec. 2021, doi: 10.12962/j23373539.v10i2.69815.
- [13] A. R. Fajri, "Penerapan Design Pattern Mvvm Dan Clean Architecture Pada Pengembangan Aplikasi Android (Studi Kasus: Aplikasi Agree)," Aug. 2022, Accessed: May 28, 2024. [Online]. Available: <https://dspace.uii.ac.id/handle/123456789/40624>
- [14] F. F. Anhar and F. T. Anggraeny, "IMPLEMENTASI CLEAN ARCHITECTURE MVVM DAN REPOSITORY PATTERN UNTUK PENGEMBANGAN APLIKASI ANDROID JUAL BELI BARANG BEKAS 'SECONDHAND,'" 2022.
- [15] T. Saifulloh, A. P. Kharisma, and D. W. Brata, "Pengembangan Aplikasi Perangkat Bergerak Pencarian Partner Lomba berbasis Android menggunakan Clean Architecture," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 4, Art. no. 4, Jun. 2023.

