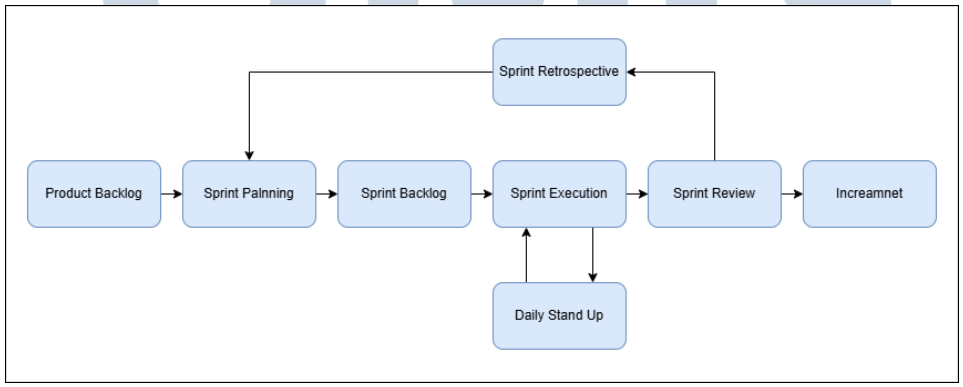
BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Posisi *internship* selama masa magang adalah sebagai *backend developer* pada divisi *Engineering Product Development* (EPD) PDDikti. Proses magang didampingi oleh mentor yaitu Ibu Annisa Nur Fadhilah dari divisi data dan Informasi. Diskusi magang dilakukan secara aktif *offline* maupun *online* dengan teman-teman satu *role* dan *role* lain yang berkaitan seperti *data engineer* dan *data analyst* menggunakan whatsapp, discord dan google meet sebagai media komunikasi.

3.2 Tugas yang Dilakukan

Tugas yang diberikan selama masa magang adalah melanjutkan *development* website PDDikti baru, dengan sistematika penulisan kode yang lebih rapi, tampilan yang lebih menarik dan fitur yang efektif dan efisien menggunakan metode scrum. Metode scrum adalah kerangka kerja pengembangan proyek yang bersifat iteraif dan inkremental. Metode ini memiliki 3 peran penting yaitu *Scrum master* yang bertugas mengawasi proses dan memastikan penerapan scrum yang benar, *product owner* bertugas memastikan bahwa visi dan kebutuhan pelanggan dipenuhi dengan prioritas yang tepat, dan tim *development* bertugas mengembangkan produk yang berkualitas sesuai dengan persyaratan dan kebutuhan yang telah ditentukan [4].



Gambar 3.1. Tahapan Metode Scrum

Tahapan yang dilakukan selama pengerjaan proyek adalah:

a) *Product Backlog*

Backlog adalah fitur atau tugas yang ingin dicapai dan diselesaikan dalam pembangunan *Website*. *Product backlog* bisa diartikan sebagai rancangan prioritas tugas yang dirumuskan untuk menyelesaikan sebuah proyek.

Pada tahap ini, *Product manager* merumuskan *backlog* untuk setiap *role* berdasarkan dari keinginan *Stakeholder* berdasarkan prioritas terntentu [5] dengan mendiskusikan terlebih dahulu kepada mentor. Pada tahap ini, kekurangan dari setiap *role* yang baru ditemukan didiskusikan untuk ditentukan menjadi *backlog*.

b) Sprint

Sprint adalah periode waktu yang ditetapkan untuk menyelesaikan *backlog* yang telah disusun pada *product backlog*. Ada beberapa tahap juga pada *sprint* yaitu:

(a) *Sprint Planning*

Sprint planning adalah pertemuan seluruh tim yang dilakukan di awal *sprint* [6]. Tujuannya untuk merencanakan dan menyampaikan pekerjaan yang dilakukan selama *sprint*, menetapkan tujuan dan berdiskusi tentang bagaimana mencapainya.

(b) *Sprint Backlog*

Sprint Backlog adalah daftar tugas yang harus diselesaikan selama periode sprint yang sudah ditentukan oleh product manager [7]. Hal ini juga termasuk pekerjaan yang perlu diselesaikan termasuk *bug fixes* dll.

(c) *Sprint Execution*

Tahap ini adalah fase dimana tim menyelesaikan pekerjaan yang sudah direncanakan pada saat *sprint planning*. pada tahap ini, seluruh *sprint backlog* dikerjakan secara sistematis dan kolaborator untuk mencapai tujuan. Pada saat eksekusi ini selama periode *sprint*, setiap harinya dilakukan *Daily Stand Up* dimana dilakukan pertemuan tim setiap hari kurang lebih selama 15 menit untuk memantau progress, mengidentifikasi hambatan, serta memastikan seluruh tim memiliki pemahaman yang sama [8]. Ketika dilakukan *daily stand up* biasanya membahas pertanyaan template untuk setiap *role* dan anggota tim. Yaitu

pertanyaan apa yang sudah dikerjakan?, apa hambatan yang ditemui?, dan apa yang dilakukan hari ini?. Hal ini dilakukan singkat namun rutin untuk memantau komitmen tiap anggota.

(d) *Sprint Review*

Kebalikan dari *sprint planning* yang dilakukan di awal *sprint* [9], *sprint review* dilakukan pada akhir *sprint*. Hal ini dilakukan untuk meninjau hasil dari pekerjaan selama periode *sprint*. Pada tahap ini setiap role dan anggota tim menyampaikan hasil pengerjaan *backlog* dengan tujuan mendapat umpan balik dari tiap *role* dan mentor.

(e) *Sprint Retrospective*

Sprint Retrospective juga dilakukan pada akhir periode sprint [10]. Tahap ini dilakukan untuk mengevaluasi prose kerja dan diskusi tentang bagaimana meningkatkan prose kerja, mencari penghambat, dan berdiskusi apa yang perlu dipertahankan, diselesaikan bahkan dipecahkan [11] .

c) *Increment*

Increment dalam metodologi scrum adalah hasil dari sebuah *sprint* [7]. Berupa versi fungsional dari produk atau perangkat lunak yang dapat digunakan atau dipresentasikan kepada pemangku kepentingan untuk mendapatkan umpan balik. Setiap *increment* harus memberikan nilai tambah yang nyata bagi pengguna atau pelanggan, dan harus dapat diuji untuk memastikan bahwa itu memenuhi harapan mereka. *Increment* ini menjadi dasar untuk perencanaan selanjutnya dalam pengembangan produk, membantu tim untuk secara bertahap meningkatkan nilai produk dan mendekati versi akhir yang diinginkan melalui iterasi berulang dalam siklus pengembangan.

Berikut beberapa fitur yang dikerjakan yaitu:

1. Fitur Perbandingan program studi dan perguruan tinggi

   (a) Membuat *endpoint* untuk layanan perbandingan program studi dan perguruan tinggi.

   (b) Membuat *Pagination* untuk layanan perbandingan program studi dan perguruan tinggi.

2. Fitur tampilan data statistik

   (a) Membuat *endpoint* program studi dan perguruan tinggi menu statistik pada *landing page*.

   (b) Membuat *CronJob* untuk menyimpan dan menampilkan data statistik

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Pengenalan lingkungan dengan mengikuti *onboarding* mitra dan nasional. Kemudian mempelajari bahasa pemrograman golang berupa *code structure*, *clean architecture*, dan *clean code* serta mempelajari *middleware* dan *object relational mapping*.
2	Mempelajari materi rest API *(Application Programming Interface)*, Sosialisasi alur kerja *Engineering product development*, pengenalan struktur directory dan review materi golang pemula.
3	Mempelajari materi swagger, mempelajari dokumentasi API menggunakan postman dan membuat dokumentasi API menggunakan postman.
4	Melanjutkan pembuatan dokumentasi API dan deskripsinya.
5	Berdiskusi dengan mentor dan *Data Analyst* untuk mempelajari *database* PDDikti.
6	Mengerjakan fitur perbandingan program studi dan perguruan tinggi.
7	Membuat *endpoint* untuk fitur perbandingan perguruan tinggi dan program studi, serta mengerjakan revisi terkait *graduation rate*, daya tampung, dan masa studi
8	Mengerjakan query untuk menampilkan graduation rate dan daya tampung pada fitur perbandingan program studi dan perguruan tinggi. kemudian mempelajari CMS dan query filter untuk akreditasi program studi
9	Membuat *handler helper* untuk *default response*
10	Membuat *endpoint* statistik untuk *landing page*
11	Membuat log untuk memantau pekerjaan cronJob dan membenarkan query yang menampilkan portofolio dosen
12	Membuat *pagination* untuk menampilkan *search* PT

Durasi kerja selama magang adalah 8 jam per hari dengan 5 hari kerja. Penerapan metode scrum dimulai pada minggu ketiga pelaksanaan magang. Setiap *sprint* berlangsung selama 5 hari kerja, yang mana satu minggu digunakan sebagai satu periode *sprint*.

Pada minggu ketiga, kegiatan berfokus pada mempelajari materi swagger

dan dokumentasi API menggunakan postman. Selama *sprint planning* yang dilakukan pada awal minggu, *backlog* yang berisi tugas-tugas ini disampaikan kepada tim *development*. *Sprint planning* berlangsung selama 15-20 menit untuk memastikan semua anggota tim memahami tugas yang harus diselesaikan. Selama *sprint execution*, *daily stand-up meeting* diadakan setiap hari untuk melacak kemajuan belajar dan implementasi Swagger serta dokumentasi API, serta mengidentifikasi dan menyelesaikan hambatan yang dihadapi. Di akhir minggu, dilakukan *sprint review* dan *retrospective* untuk mengevaluasi kemajuan dan menentukan langkah selanjutnya. Pada minggu ketiga ini, *increment* berhasil dilakukan dengan pembuatan dokumentasi API yang memadai.

Pada minggu keempat, kegiatan berfokus pada melanjutkan pembuatan dokumentasi API dan deskripsinya. *Sprint planning* dilakukan untuk menetapkan backlog yang harus diselesaikan selama minggu ini. *Daily stand-up meeting* terus diadakan untuk memantau kemajuan dan mengatasi hambatan dalam pembuatan dokumentasi. Di akhir *sprint, sprint review* dan *retrospective* diadakan untuk mengevaluasi hasil dokumentasi yang telah dibuat. Pada minggu ini, increment berhasil dilakukan dengan penyelesaian lebih lanjut dari dokumentasi API.

Pada minggu kelima, kegiatan melibatkan diskusi dengan mentor dan *data analyst* untuk mempelajari database PDDikti. Sprint planning diadakan untuk menetapkan tugas diskusi dan pembelajaran yang harus diselesaikan selama minggu ini. *daily stand-up meeting* membantu memastikan bahwa kemajuan belajar terus terpantau dan hambatan dalam memahami database PDDikti dapat segera diatasi. *sprint review* dan *retrospective* di akhir minggu memungkinkan evaluasi efektivitas diskusi dan pemahaman tentang *database* PDDikti. Pada minggu ini, increment berhasil dilakukan dengan pemahaman yang mendalam tentang database yang digunakan.

Pada minggu keenam, tugas berfokus pada pengerjaan fitur perbandingan program studi dan perguruan tinggi. *Sprint planning* dilakukan untuk menetapkan backlog yang mencakup tugas-tugas terkait pengembangan fitur ini. *Daily stand-up meeting* terus mendukung pemantauan kemajuan dan penyelesaian hambatan. Di akhir *sprint*, *sprint review* dan *retrospective* membantu mengevaluasi fitur yang telah dikembangkan. Pada minggu ini, increment berhasil dilakukan dengan pengembangan fitur perbandingan yang memadai.

Pada minggu ketujuh dan seterusnya, kegiatan melibatkan pembuatan *endpoint* untuk fitur perbandingan perguruan tinggi dan program studi, serta revisi terkait *graduation rate*, daya tampung, dan masa studi. Setiap minggu dimulai

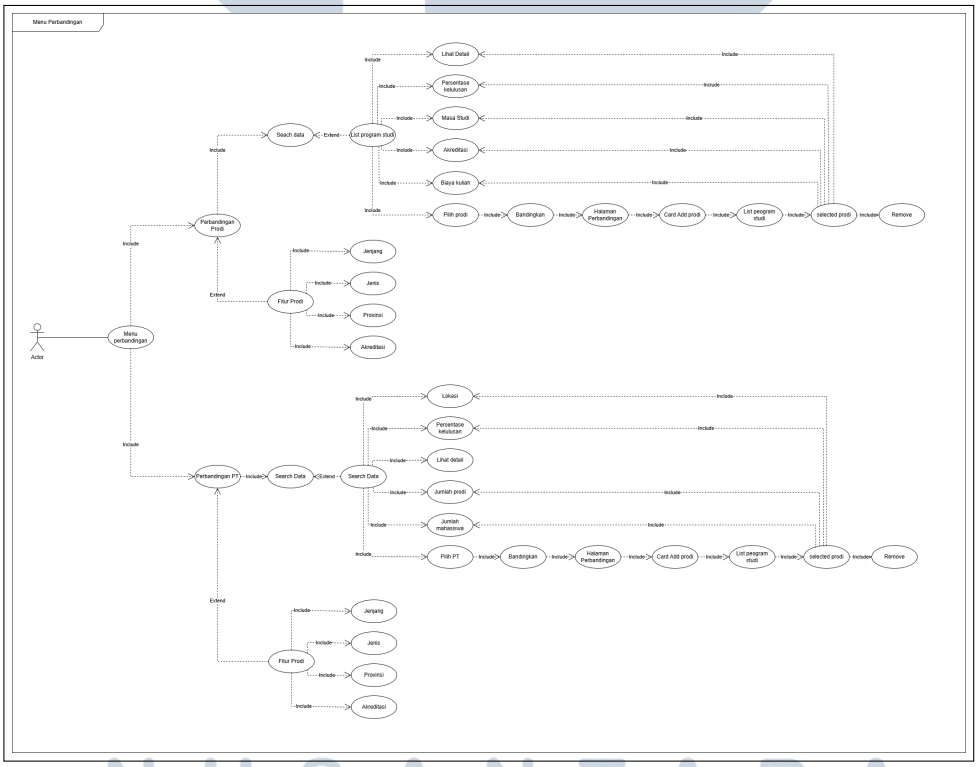
dengan *sprint planning* untuk menetapkan *backlog* dan diakhiri dengan *sprint review* dan *retrospective* untuk mengevaluasi kemajuan dan menetapkan tindakan perbaikan atau penyelesaian *backlog*. Pada setiap minggu tersebut, *increment* berhasil dilakukan dengan penyelesaian tugas-tugas yang ditetapkan dalam *backlog*, termasuk pembuatan *endpoint* dan revisi fitur yang diperlukan.

Pada Bab ini Tidak dilampirkan *script* yang dibuat karena merupakan *confidential* perusahaan.

3.3.1 Fitur Perbandingan Program Studi dan Perguruan Tinggi

a) Membuat *endpoint* untuk layanan perbandingan program studi dan perguruan tinggi.

Menu perbandingan adalah fitur baru yang disediakan Dikti untuk membantu calon mahasiswa dalam menentukan Pendidikan Tinggi (PT) dan Program Studi (Prodi) untuk melanjutkan pendidikan.



Gambar 3.2. Use case Perbandingan PT dan Prodi

Berdasarkan *use case* menu perbandingan pada Gambar 3.2, dapat dilihat bahwa dalam fitur perbandingan prodi pengguna memiliki dua menu perbandingan, yaitu perbandingan perguruan tinggi dan perbandingan

program studi. Setiap menunya memiliki satu *use case* dengan simbol *include* dan satu *use case* dengan simbol *extend*. Simbol *include* menunjukkan bahwa secara *default*, sistem akan menampilkan daftar program studi atau perguruan tinggi beserta variabel pada simbol *use case* yang dihubungkan dengan simbol *include*. Sedangkan, dengan simbol *extend*, fitur perbandingan program studi atau perguruan tinggi memungkinkan pengguna untuk menyempitkan filter dengan memasukkan input ke beberapa variabel yaitu jenjang, jenis, provinsi, dan akreditasi yang digambarkan dengan simbol *use case include* kepada fitur perbandingan PT dan Prodi.

Diagram *use case* tersebut menggambarkan interaksi pengguna dengan sistem dalam konteks fitur perbandingan perguruan tinggi dan program studi. Pengguna mengakses fitur perbandingan melalui menu utama yang disebut *Menu Perbandingan*.

Dalam menu ini, terdapat dua *use case* utama, yaitu perbandingan PT (Perguruan Tinggi) dan perbandingan prodi (Program Studi). *Use case* perbandingan PT memungkinkan pengguna untuk mencari dan membandingkan data perguruan tinggi berdasarkan berbagai kriteria. Proses ini mencakup pencarian data (*search data*), melihat detail informasi perguruan tinggi (lihat detail), yang meliputi persentase kelulusan, jumlah mahasiswa, jumlah program studi yang ditawarkan, lokasi, dan memilih perguruan tinggi untuk dibandingkan (pilih PT). Informasi ini juga dapat ditambahkan ke halaman perbandingan menggunakan fitur *card add*, dan pengguna dapat melihat daftar program studi yang telah dipilih untuk dibandingkan pada list program studi. Terdapat juga opsi untuk menghapus program studi dari daftar perbandingan dengan fitur *remove*.

Use case perbandingan prodi memberikan fungsionalitas serupa namun difokuskan pada program studi. Pengguna dapat melakukan pencarian data program studi (*search data*) dan melihat detail informasi yang mencakup persentase kelulusan, masa studi, akreditasi, dan biaya kuliah. Proses ini juga melibatkan pemilihan program studi untuk dibandingkan (pilih prodi), penambahan program studi ke dalam daftar perbandingan, serta melihat dan menghapus program studi dari daftar perbandingan.

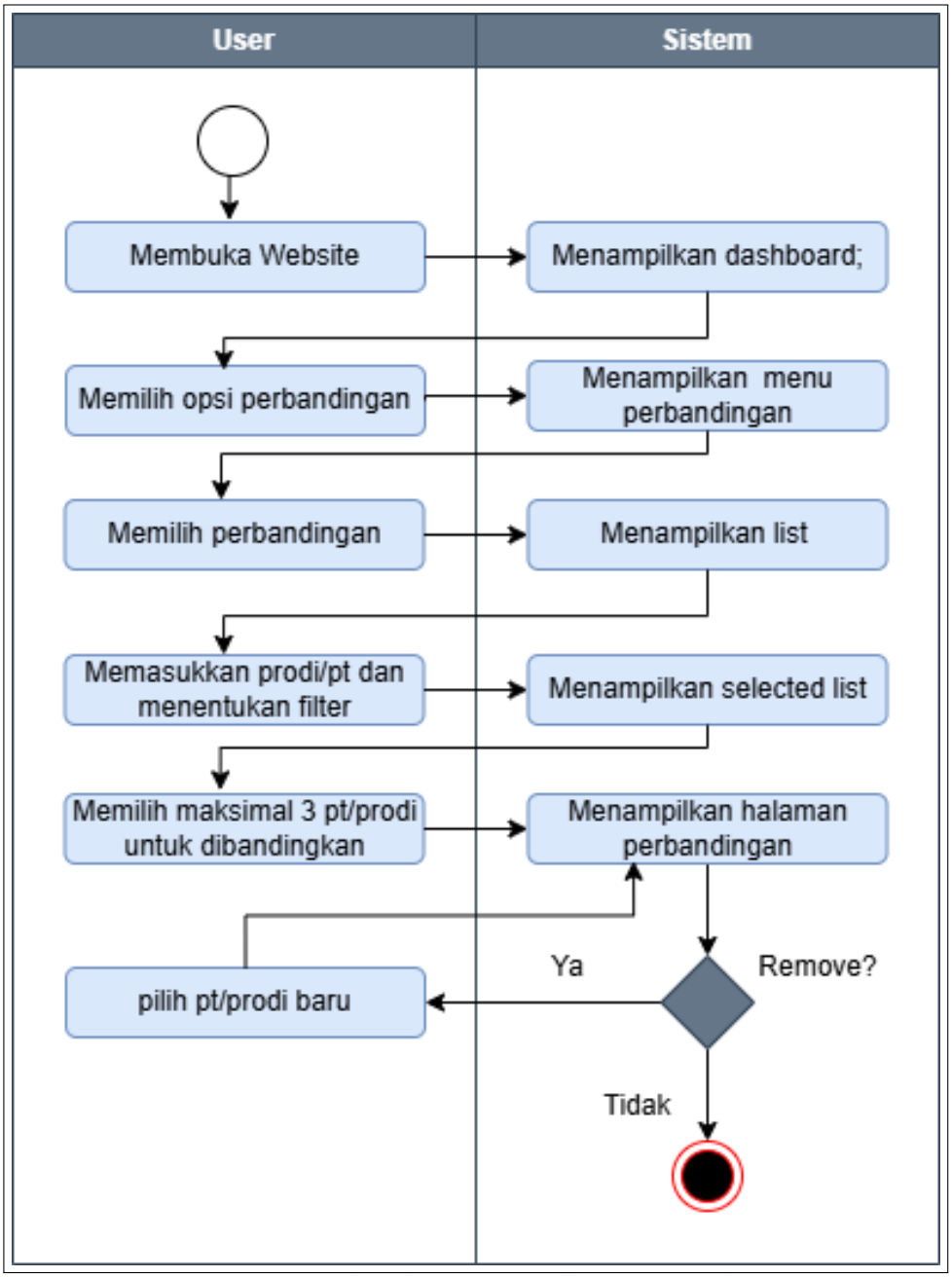
Sub-*Use case* fitur prodi dan fitur PT mendukung proses pencarian data dengan kriteria spesifik seperti jenis, jenjang, provinsi, dan akreditasi. Fitur prodi mendukung pencarian program studi, sementara fitur PT mendukung

pencarian perguruan tinggi.

Setelah pemilihan perguruan tinggi dan program studi yang ingin dibandingkan, hasil perbandingan ditampilkan di halaman perbandingan, yang menyediakan informasi terperinci untuk membantu dalam pengambilan keputusan yang lebih baik.

Keseluruhan proses ini dirancang untuk mendukung calon mahasiswa dalam mencari dan memilih pendidikan tinggi serta program studi terbaik berdasarkan informasi yang valid dan komprehensif. Fitur perbandingan ini bertujuan untuk memberikan alat yang berguna dalam menentukan pilihan pendidikan yang paling sesuai dengan kebutuhan dan preferensi akademik.

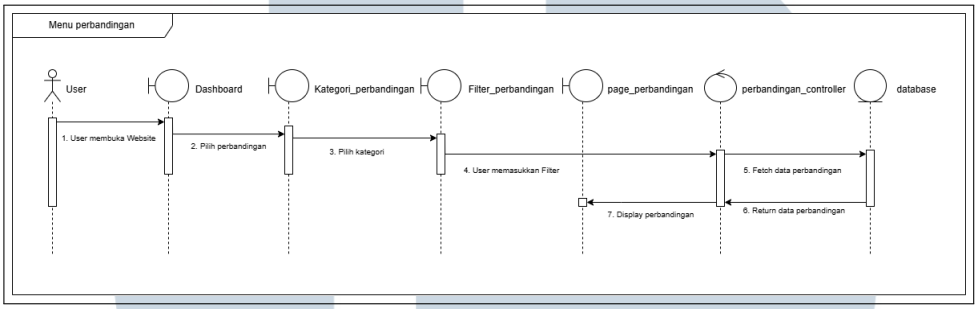
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3. *Activity Diagram* Perbandingan PT dan Prodi

Alur fitur perbandingan PT dan prodi sebagaimana tertera pada Gambar 3.3 yaitu terdapat 2 *swimlane* yang mewakili *user* dan sistem sebagai sebagai pelaku organisasi bisnis yang bersangkutan. Dimulai dengan *user* membuka website maka sistem akan menampilkan *dashboard* Kemudian *user* memilih menu perbandingan lalu sistem menampilkan opsi perbandingan yang berisi 2 opsi; perbandingan perguruan tinggi dan program studi. Ketika salah satu opsi dipilih maka akan ditampilkan list keseluruhan

PT/prodi. dilanjutkan dengan *user* memasukkan nama PT/prodi yang ingin dibandingkan dan memasukkan filter bila diinginkan, kemudian ditampilkan halaman perbandingan. Terdapat *decision* jika *user* ingin menghapus maka *user* bisa menambahkan PT/prodi baru dan selesai.



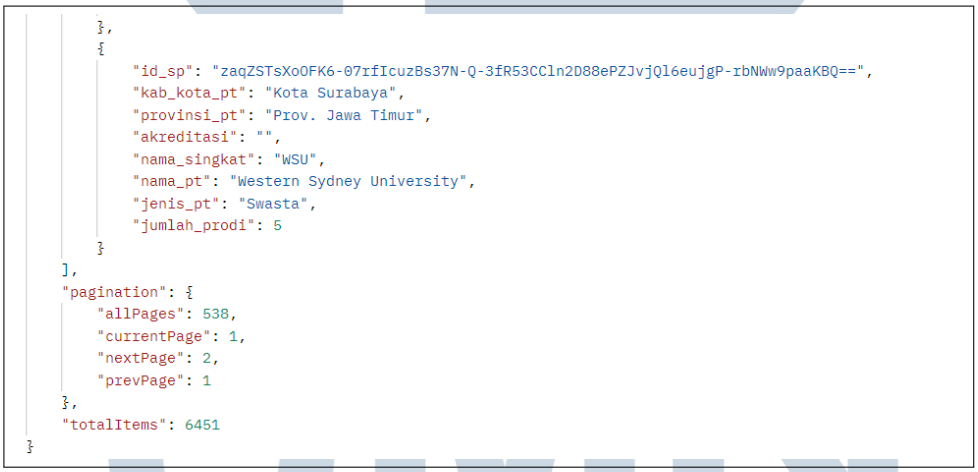
Gambar 3.4. *Sequence Diagram* Perbandingan PT dan Prodi

Terdapat 7 objek yang berinteraksi dalam fitur perbandingan sebagaimana dapat dilihat pada Gambar 3.4. Dalam fitur ini objek user berinteraksi dengan objek *interface dashboard*, kemudian berinteraksi dengan objek *interface* kategori perbandingan. Di sini *user* bisa memilih antara perbandingan perguruan tinggi atau program studi dan dilanjutkan dengan interaksi selanjutnya yaitu menampilkan objek *interface* opsi perbandingan. Pada objek ini, *user* memasukkan *input* PT/prodi beserta filter yang diinginkan. Kemudian berinteraksi dengan objek perbandingan *controller* dari objek ini akan melakukan *request* data dan berinteraksi dengan objek *database* perbandingan. Setelah berhasil melakukan *fetch* dan dari objek *database* berinteraksi lagi dengan objek *controller* perbandingan dan terakhir dilanjutkan melakukan *display* dengan berinteraksi kepada objek *interface page* perbandingan.

UNIVERSITAS MULTIMEDIA NUSANTARA

Gambar 3.5. Hasil *Endpoint* Tampilan Prodi Menggunakan Postman

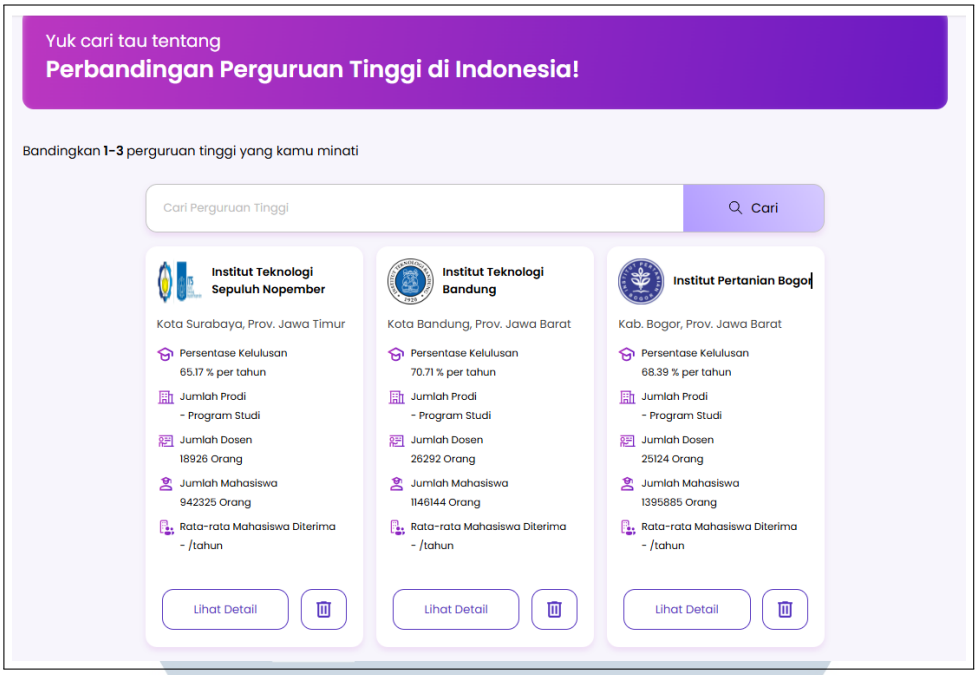
Hasil dari *endpoint* prodi yang telah berhasil dibuat adalah sebagaimana bisa dilihat pada Gambar 3.5.



Gambar 3.6. Hasil *Endpoint* Tampilan PT Menggunakan Postman

Pada Gambar 3.6 dapat dilihat juga hasil *endpoint* untuk perbandingan perguruan tinggi.

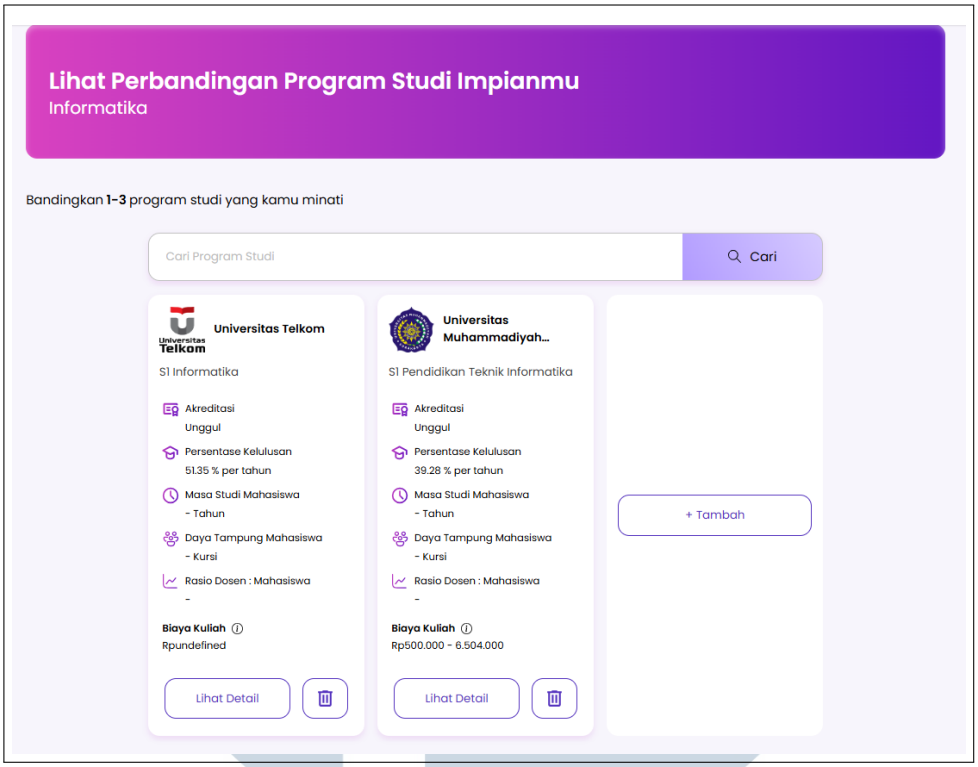
UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.7. *Interface* Perbandingan PT

Dari *endpoint* PT pada Gambar 3.6 selanjutnya akan dipakai oleh *frontend* sehingga menghasilkan tampilan akhir seperti Gambar 3.7 diatas. Syarat dan ketentuan yang berlaku ketika ingin menggunakan layanan perbandingan adalah jenjang pendidikan tinggi harus seimbang S1 dengan S1 juga, dan minimal memasukkan 2 perguruan tinggi untuk dibandingkan. user bisa menghapus dan memasukkan lagi pilihan baru untuk dibandingkan.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

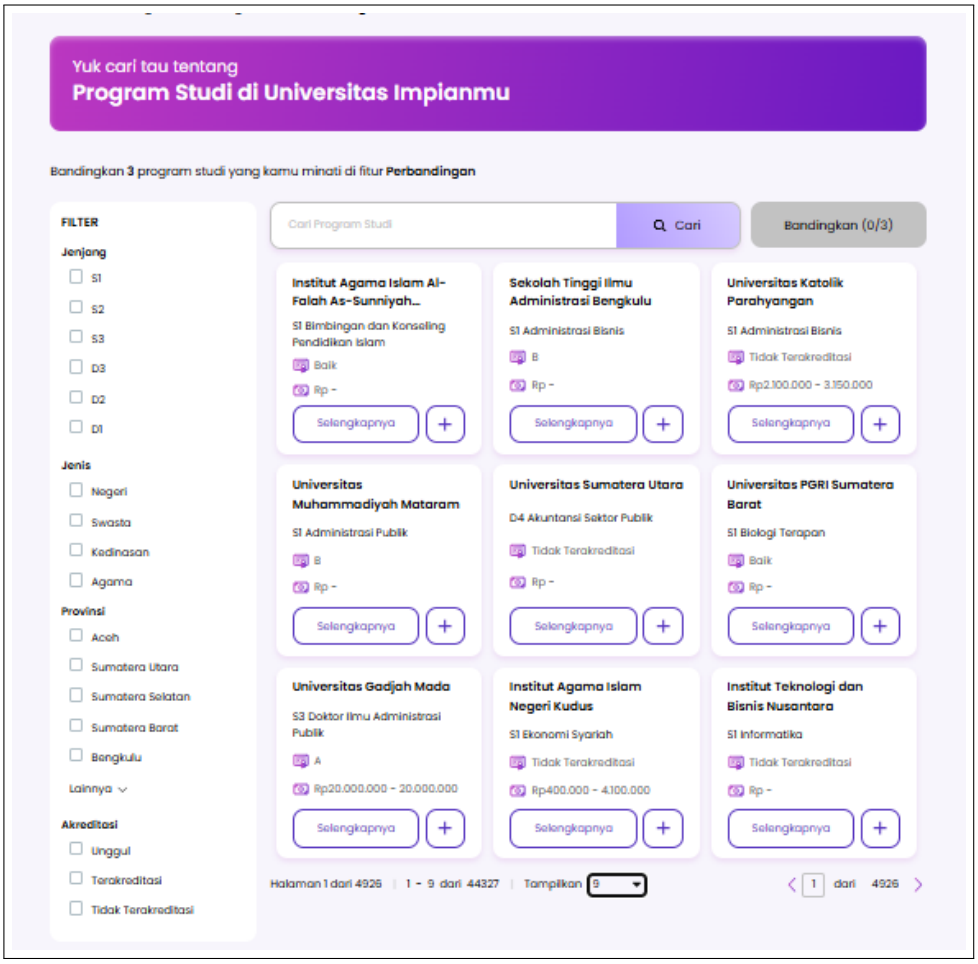


Gambar 3.8. *Interface* perbandingan Prodi

Sedangkan hasil *endpoint* prodi yang sudah dibuat pada Gambar 3.5 akan digunakan *frontend* untuk membuat tampilan perbandingan prodi pada Gambar 3.8 diatas. Pada opsi ini syarat dan ketentuan yang berlaku adalah program studi yang akan dibandingkan harus sama dan minimal memasukkan 2 prodi untuk dibandingkan.

b) Membuat *Pagination* untuk layanan perbandingan program studi dan perguruan tinggi.

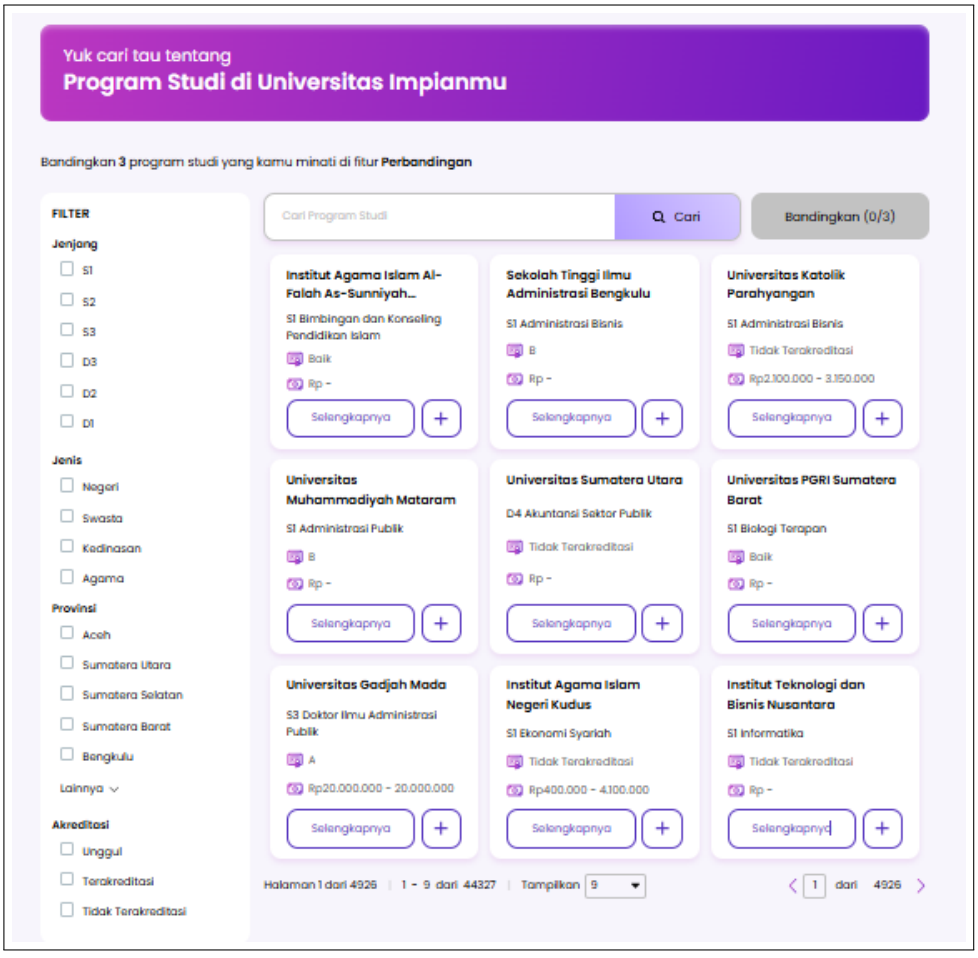
Pagination pada fitur perbandingan PT dan prodi berguna agar list PT/prodi yang ditampilkan melakukan *load* data setiap *page*, sehingga meminimalisir kemungkinan *speed* kemunculan data yang lama dan gagal *loading* setelah beberapa *page* dibuka. *Pagination* yang dibuat dapat dilihat pada Gambar 3.5 dan 3.6. *Pagination* mengembalikan semua *page*, *page* yang sedang dibuka, *page* selanjutnya dan *page* sebelumnya.



Gambar 3.9. *Interface Paginate* PT

Dari *paginate* PT yang hanya menampilkan *response* berupa angka int seperti Gambar 3.6, maka akan tampil pada *interface user* seperti pada Gambar 3.9.

UNIVERSITAS MULTIMEDIA NUSANTARA



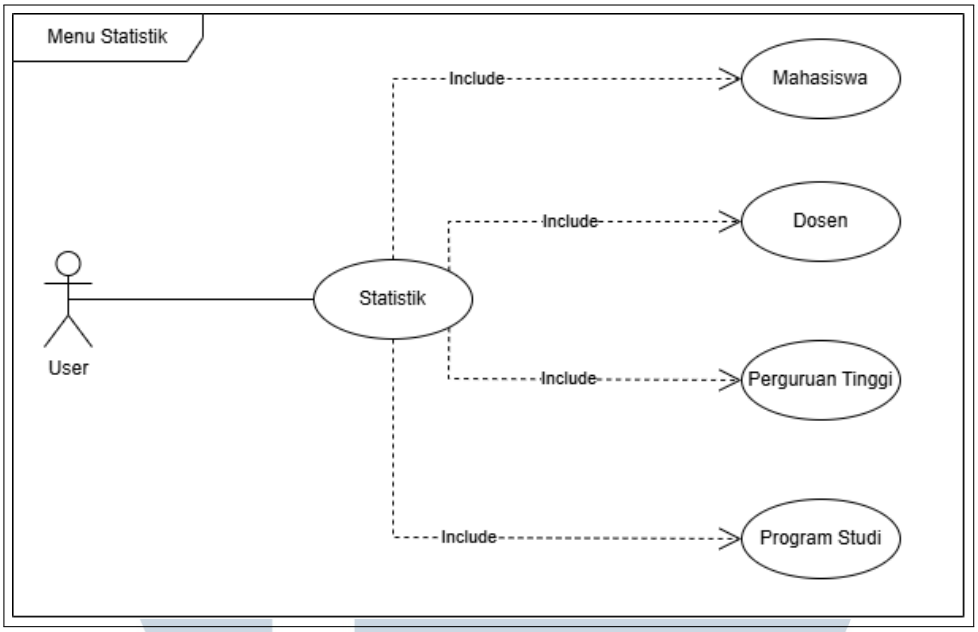
Gambar 3.10. *Interface Paginate* Prodi

Sama seperti bagian PT, *Interface Search* prodi pada Gambar 3.10, dibuat berdasarkan *paginate* pada Gambar 3.5.

3.3.2 Fitur Tampilan Data Statistik Perguruan Tinggi dan Program studi

a) Membuat *endpoint* program studi dan perguruan tinggi menu statistik pada *landing page*.

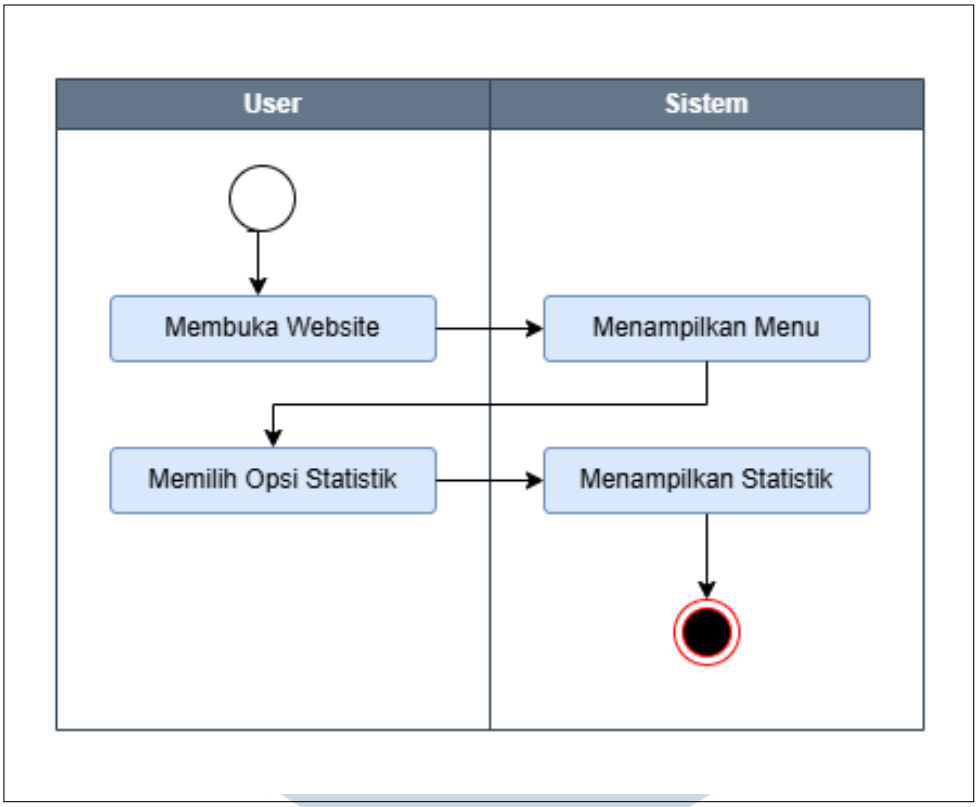
Fitur statistik dibuat dengan tujuan agar memudahkan user menganalisa dan mengetahui perbandingan dari beberapa komponen yang bisa menjadi pertimbangan menentukan perguruan tinggi atau analisis terkait pendidikan tinggi di Indonesia. Seperti jumlah PT berdasarkan provinsi, Jumlah PT berdasarkan bentuk pendidikan, jumlah PT berdasarkan kelompok pembina, Jumlah PT akreditasi, jumlah prodi berdasarkan akreditasi, jumlah prodi berdasarkan bidang ilmu, jumlah prodi berdasarkan jenjang dan jumlah prodi berdasarkan kelompok pembina.



Gambar 3.11. *Usecase* Statistik

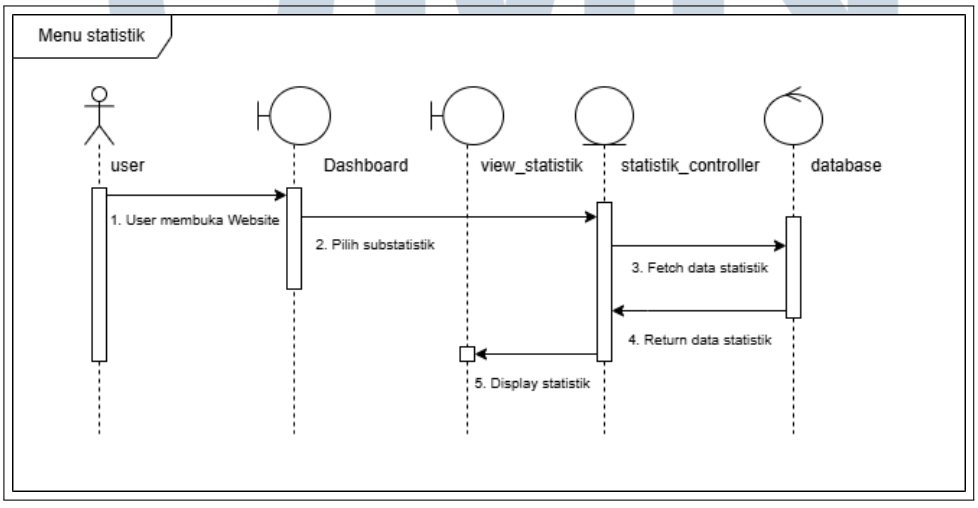
Seperti yang tertera pada Gambar 3.11, pada fitur statistik, terdapat 4 *use case* yang *Include* pada statistik. Artinya, ketika *user* membuka fitur statistik, *user* dapat mengakses 4 kategori statistik yaitu mahasiswa, dosen, perguruan tinggi dan program studi.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.12. *Activity Diagram* Statistik

Sebagaimana pada Gambar 3.12, alur fitur statistik terlihat sederhana. komunikasi diwakilkan dengan 2 *swimline* yaitu *user* dan sistem. Dimulai dengan *user* membuka website kemudian sistem menampilkan menu, dilanjutkan dengan *user* memilih 1 dari 4 opsi statistik kemudian statistik ditampilkan dan selesai.



Gambar 3.13. Sequence Diagram Statistik

Pada diagram sequence menu statistik pada Gambar 3.13, diperlihatkan terdapat 5 objek yang berkomunikasi. objek *user* berinteraksi dengan objek *interface dashboard*. Setelah *user* memilih opsi statistik kemudian berinteraksi dengan objek *controller* statistik untuk melakukan *request* data kepada objek *database*. Setelah objek *database* berhasil memberikan *return* data maka dari objek *controller* akan berkomunikasi dengan objek *interface view* statistik untuk *display* data.


```
{
    "jumlah_pt": 106,
    "provinsi_pt": "Prov. Kalimantan Timur"
},
{
    "jumlah_pt": 360,
    "provinsi_pt": "Prov. Sulawesi Selatan"
},
{
    "jumlah_pt": 542,
    "provinsi_pt": "Prov. Jawa Tengah"
},
{
    "jumlah_pt": 54,
    "provinsi_pt": "Prov. Maluku"
}
```


Gambar 3.14. Postman PT Provinsi

Gambar 3.14 menampilkan hasil *endpoint* statistik jumlah PT berdasarkan provinsi.

Gambar 3.15. *Interface* PT Provinsi

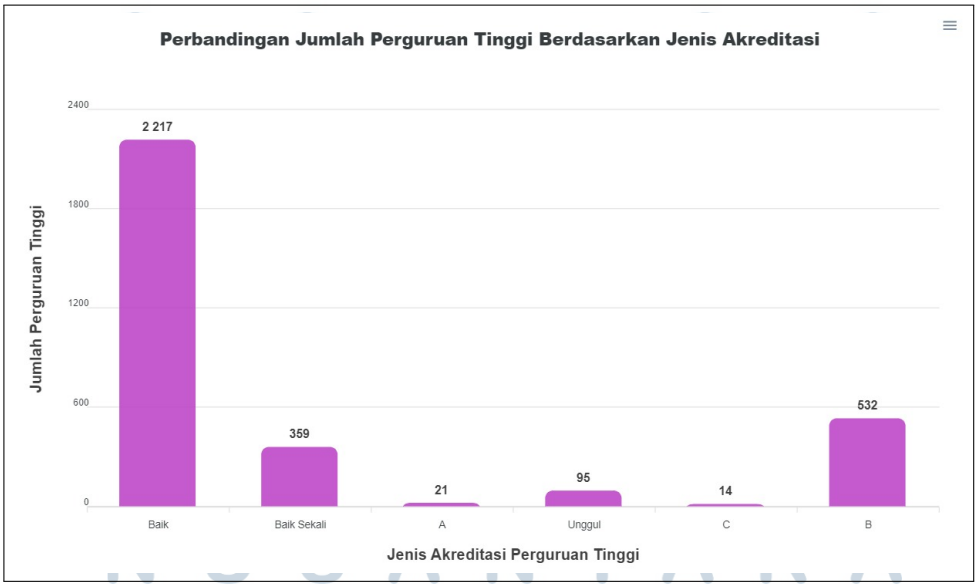
Dari *endpoint* yang dibuat dan berhasil mengembalikan *response* seperti Gambar 3.14, data ini divisualisasikan dalam bentuk *choropleth map* yang dibuat dengan geojson. Peta ini akan menampilkan distribusi perguruan

tinggi di setiap provinsi dengan skala warna untuk menggambarkan jumlah perguruan tinggi, memudahkan pemahaman persebaran geografis perguruan tinggi di Indonesia.


```
[
    {
        "jumlah_pt": 2239,
        "nm_akred": "Baik"
    },
    {
        "jumlah_pt": 370,
        "nm_akred": "Baik Sekali"
    },
    {
        "jumlah_pt": 21,
        "nm_akred": "A"
    },
    {
        "jumlah_pt": 96,
        "nm_akred": "Unggul"
    },
    {
        "jumlah_pt": 14,
        "nm_akred": "C"
    },
    {
        "jumlah_pt": 518,
        "nm_akred": "B"
    }
]
```


Gambar 3.16. Postman PT Akreditasi

Gambar 3.16 menampilkan hasil *endpoint* postman pt berdasarkan akreditasi.



Gambar 3.17. Interface PT Akreditasi

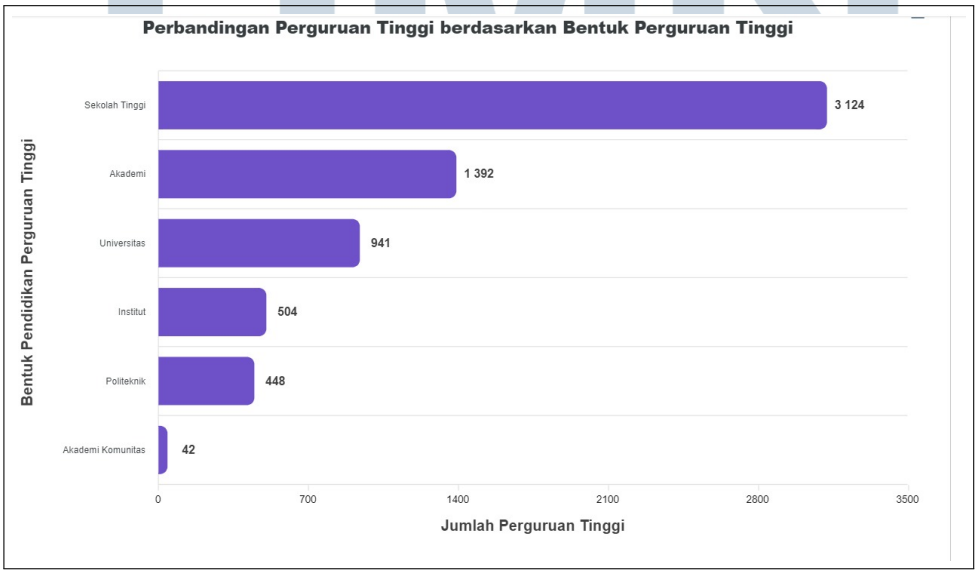
Berdasarkan data dari *endpoint* yang dibuat seperti Gambar 3.16, divisualisasikan Perbandingan Jumlah Perguruan Tinggi Berdasarkan Jenis

Akreditasi yang meliputi kategori akreditasi A, B, C, Unggul, Baik Sekali, dan Baik, penggunaan grafik kolom (column chart) dapat memberikan visualisasi yang jelas dan informatif.


```
[
    {
        "jumlah_pt": 3124,
        "nama_bentuk": "Sekolah Tinggi"
    },
    {
        "jumlah_pt": 42,
        "nama_bentuk": "Akademi Komunitas"
    },
    {
        "jumlah_pt": 449,
        "nama_bentuk": "Politeknik"
    },
    {
        "jumlah_pt": 1392,
        "nama_bentuk": "Akademi"
    },
    {
        "jumlah_pt": 504,
        "nama_bentuk": "Institut"
    },
    {
        "jumlah_pt": 941,
        "nama_bentuk": "Universitas"
    }
]
```


Gambar 3.18. Postman PT Bentuk

Berdasarkan *endpoint* yang dibuat, maka *response* data yang digunakan untuk membuat visualisasi statik jumlah pt berdasarkan bentuk pendidikan dapat dilihat pada Gambar 3.18.



Gambar 3.19. *Interface* PT Bentuk

Data *endpoint* jumlah PT pada Gambar 3.19 berdasarkan bentuk pendidikan divisualisasikan menggunakan *bar chart*, di mana setiap bar mewakili satu dari beberapa kategori bentuk pendidikan, yaitu universitas, institut, sekolah tinggi, akademi, akademi komunitas, dan politeknik. *Bar chart* memungkinkan perbandingan langsung antar kategori, menunjukkan distribusi jumlah perguruan tinggi berdasarkan bentuk pendidikannya.


```
[
    {
        "jumlah_pt": 241,
        "kelompok": "PTK"
    },
    {
        "jumlah_pt": 1526,
        "kelompok": "PTA"
    },
    {
        "jumlah_pt": 4557,
        "kelompok": "PTS"
    },
    {
        "jumlah_pt": 127,
        "kelompok": "PTN"
    }
]
```


Gambar 3.20. Postman PT Kelompok Pembina

Hasil *endpoint* jumlah PT berdasarkan kelompok pembina dapat dilihat pada Gambar 3.20.

Gambar 3.21. *Interface* PT Kelompok Pembina

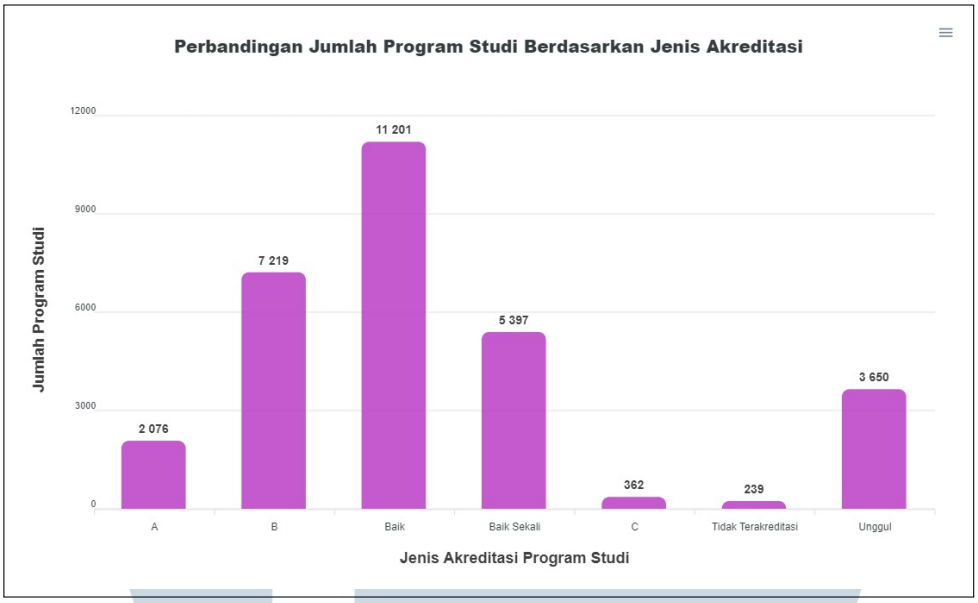
Jumlah PT berdasarkan kelompok pembina divisualisasikan menggunakan *pie chart* karena memungkinkan *user* untuk melihat pembagian proporsional dari jumlah perguruan tinggi berdasarkan kelompok pembina PTN (perguruan tinggi negeri), PTS (perguruan tinggi swasta), PTA (perguruan tinggi agama), dan PTK (perguruan tingi kedinasan). Setiap *slice* dari *pie chart* akan mewakili kelompok pembina yang berbeda, dengan ukuran *slice* yang berbeda menunjukkan jumlah perguruan tinggi dalam masing-masing kelompok. Segmentasi warna yang berbeda dalam *pie chart* membantu dalam mengidentifikasi kelompok pembina dengan mudah.


```
    {
        "jumlah_prodi": 11259,
        "nm_akred": "Baik"
    },
    {
        "jumlah_prodi": 5469,
        "nm_akred": "Baik Sekali"
    },
    {
        "jumlah_prodi": 366,
        "nm_akred": "C"
    },
    {
        "jumlah_prodi": 237,
        "nm_akred": "Tidak Terakreditasi"
    },
    {
        "jumlah_prodi": 3721,
        "nm_akred": "Unggul"
    }
]
```


Gambar 3.22. Postman Prodi Akreditasi

Pada Gambar 3.22 dapat dilihat hasil *endpoint* dari jumlah prodi berdasarkan akreditasi.

UNIVERSITAS MULTIMEDIA NUSANTARA



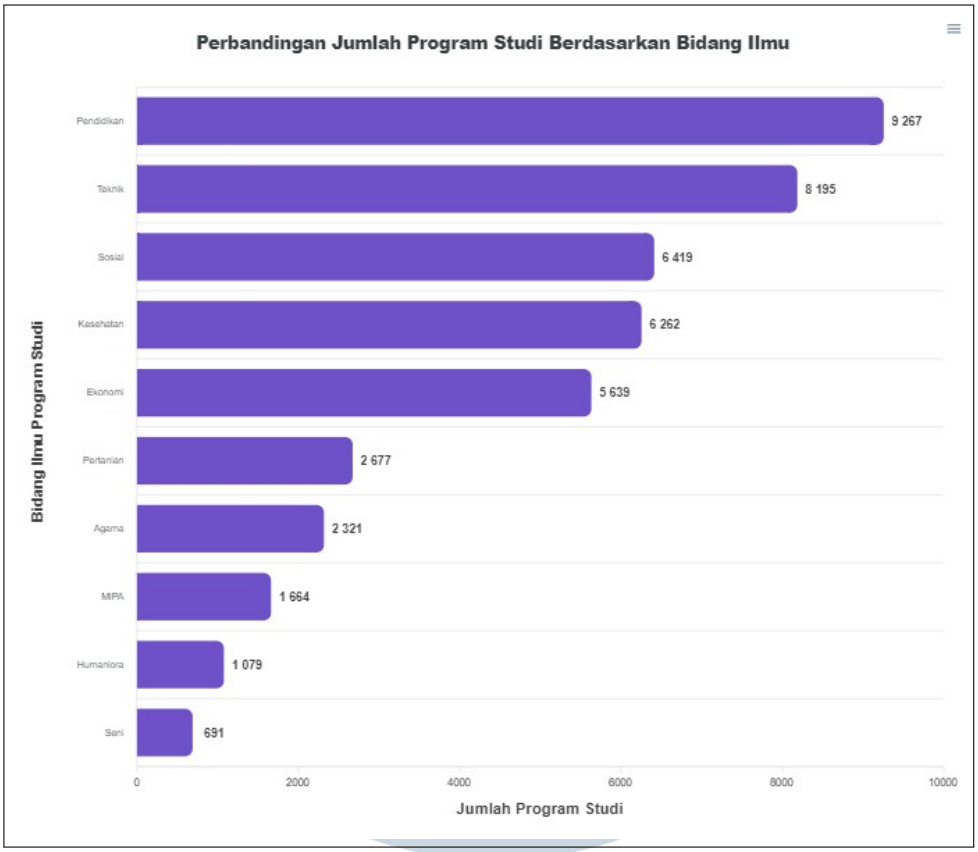
Gambar 3.23. *Interface* Prodi Akreditasi

Grafik kolom bisa digunakan untuk visualisasi ini pada Gambar 3.23. Setiap kolom mewakili jenis akreditasi (A, B, C, Unggul, Baik Sekali, Baik), yang memudahkan pemahaman distribusi akreditasi program studi.


```
    {
        "jumlah_prodi": 9267,
        "kelompok_bidang": "Pendidikan"
    },
    {
        "jumlah_prodi": 2677,
        "kelompok_bidang": "Pertanian"
    },
    {
        "jumlah_prodi": 691,
        "kelompok_bidang": "Seni"
    },
    {
        "jumlah_prodi": 6419,
        "kelompok_bidang": "Sosial"
    },
    {
        "jumlah_prodi": 8196,
        "kelompok_bidang": "Teknik"
    }
]
```


Gambar 3.24. Postman Prodi Bidang Ilmu

Gambar 3.24 menampilkan hasil uji *endpoint* jumlah prodi berdasarkan bidang ilmu.



Gambar 3.25. *Interface* Bidang Ilmu

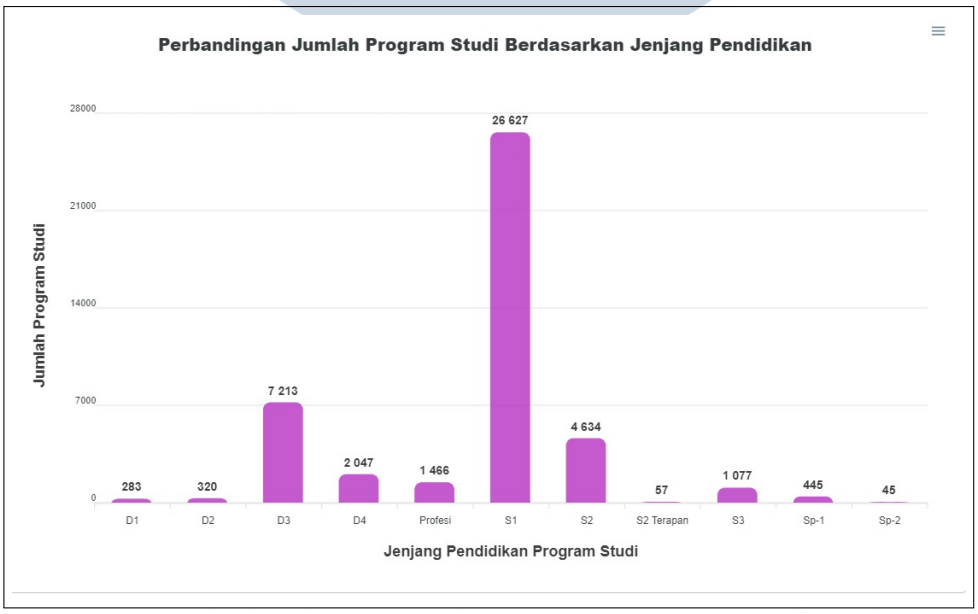
Grafik batang digunakan untuk menunjukkan sebaran jumlah program studi berdasarkan bidang ilmu (seperti Pendidikan, Teknik, dan lainnya). Ini akan menunjukkan bidang ilmu apa yang paling banyak program studinya di perguruan tinggi Indonesia, memberi gambaran tentang fokus pendidikan tinggi.

UNIVERSITAS MULTIMEDIA NUSANTARA

```
{
    "jumlah_prodi": 7215,
    "jenjang_prodi": "D3"
},
{
    "jumlah_prodi": 2049,
    "jenjang_prodi": "D4"
},
{
    "jumlah_prodi": 1467,
    "jenjang_prodi": "Profesi"
},
{
    "jumlah_prodi": 26629,
    "jenjang_prodi": "S1"
},
{
    "jumlah_prodi": 4635,
    "jenjang_prodi": "S2"
},
{
    "jumlah_prodi": 57,
    "jenjang_prodi": "S2 Terapan"
},
```


Gambar 3.26. Postman Prodi Jenjang

Response uji *endpoint* jumlah prodi berdasarkan jenjang pendidikan dapat dilihat pada Gambar 3.26.



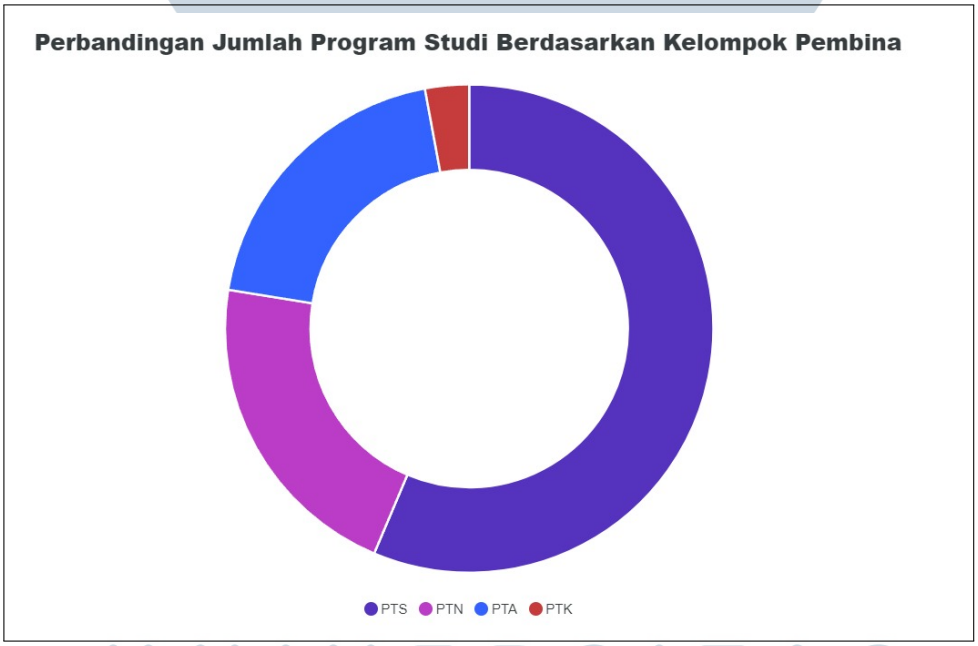
Gambar 3.27. *Interface* Prodi Jenjang

Visualisasi perbandinganjumlah program studi berdasarkan jenjang pendidikan yang mencakup jenjang seperti D1, D2, D3, D4, S1, S2, S3, Profesi menggunakan *column chart.*

```
[
    {
        "jumlah_prodi": 8646,
        "kelompok": "PTA"
    },
    {
        "jumlah_prodi": 1286,
        "kelompok": "PTK"
    },
    {
        "jumlah_prodi": 9365,
        "kelompok": "PTN"
    },
    {
        "jumlah_prodi": 24925,
        "kelompok": "PTS"
    }
]
```


Gambar 3.28. Postman Prodi Kelompok Pembina

Melalui Gambar 3.26 dapat dilihat hasil *response endpoint* yang dibuat untuk jumlah prodi berdasarkan kelompok pembina.



Gambar 3.29. *Interface* Prodi Kelompok Pembina

Sama seperti Jumlah PT berdasarkan kelompok pembina, Jumlah PT berdasarkan kelompok pembina digunakan untuk menampilkan jumlah perguruan tinggi per kelompok pembina PTN (perguruan tinggi negeri), PTS (perguruan tinggi swasta), PTA (perguruan tinggi agama), dan PTK (perguruan tingi kedinasan). Setiap bagian dari *pie chart* menunjukkan kelompok pembina, dan ukurannya mencerminkan jumlah perguruan tinggi di kelompok tersebut. Warna berbeda membantu mengenali kelompok pembina.

b) Membuat *CronJob* untuk menyimpan dan menampilkan data statistik

Data statistik yang ditampilkan pada layar *dashboard* merupakan data yang diambil dari *databse* PDDikti. Banyaknya data yang ada pada PDDikti mempengaruhi kecepatan dalam menampilkan data. Masalah akan bertambah jika data yang akan ditampilkan adalah data yang tidak statis. Ketika data terus bertambah maka akan bertambah pula waktu untuk mengolah data tersebut. Solusi yang digunakan untuk menangani masalah kecepatan ini adalah digunakannya *cronjob* atau penjadwalan yang melakukan *update* data sesuai dengan waktu yang ditentukan.

Dalam penanganan *cronjob* untuk *update* statistik juga menggunakan postgreSQL. Jadi, jika ketika tidak menggunakan *cronjob* data diambil langsung ke *database* pusat, maka ketika menggunakan *cronjob* postgreSQL digunakan untuk menampung data yang telah diupdate kemudian mengembalikannya ke *controller* baru akan ditampilkan.

3.4 Perangkat Yang Digunakan

Berikut adalah *software* dan *hardware* yang digunakan selama pelaksanaan magang.

3.4.1 *Software*

Berikut adalah *Software* yang digunakan:

- Windows 11 Home 23H2 (OS Build 22631.3593)
- Google Chrome 124.0.6367.207 (Official Build) (64-bit)
- Microsoft Edge 124.0.2478.105 (Official build) (64-bit)
- Visual Studio Code 1.89.1
- Github Desktop 3.3.12 (x64)
- Git 2.43.0.windows.1
- Postman 11.1.3
- PostgreSQL 8.2

3.4.2 Hardware

- Processor 12th Gen Intel(R) Core(TM) i5-12450H (12 CPUs), 2.0GHz
- RAM 16 GB DDR4 SDRAM 1330.1 MHz
- GPU NVIDIA GeForce RTX 3050
- Storage SSD V1.4 NVMe 512 GB

3.4.3 Tech Stack

- Go Version go1.21.6 windows/amd64
- GORM Version v1.25.5

3.5 Kendala dan Solusi yang Ditemukan

3.5.1 Kendala

Seringkali query menghasilkan data yang kurang *update* karena Terdapat beberapa isi tabel yang memiliki isi yang sama tetapi dari *database* yang berbeda membuat sering kali harus melakukan revisi query.

3.5.2 Solusi

Berdiskusi dengan para mentor untuk menentukan *database* yang digunakan dan terus melakukan *update* jika terdapat perubahan agar menghasilkan data yang akurat.

UNIVERSITAS MULTIMEDIA NUSANTARA