

menggunakan informasi dari kolom lain maupun tabel eksternal yang diperoleh secara manual serta menggunakan Python untuk melakukan *scraping* data dari tabel HTML.

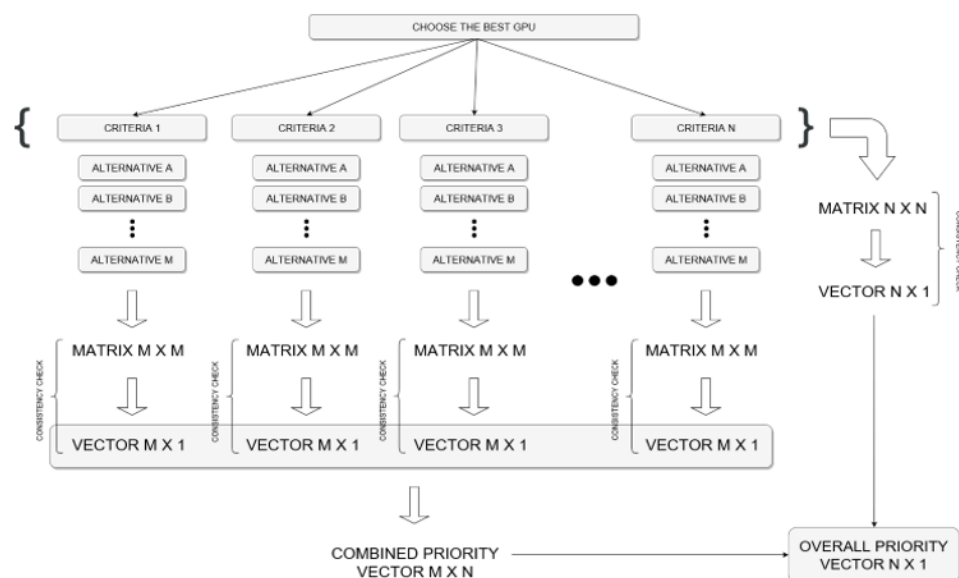
Masalah-masalah di atas membuat penulis lebih kreatif dalam memilih alternatif pemecahan masalah, lebih mandiri dalam mempelajari hal baru, lebih bertanggung jawab terhadap solusi yang diambil, dan menanggung konsekuensinya.

2.4 Penuntasan Tugas dan Penanganan Masalah

Berikut adalah dua proyek yang telah dikerjakan oleh penulis selama magang, yaitu membuat:

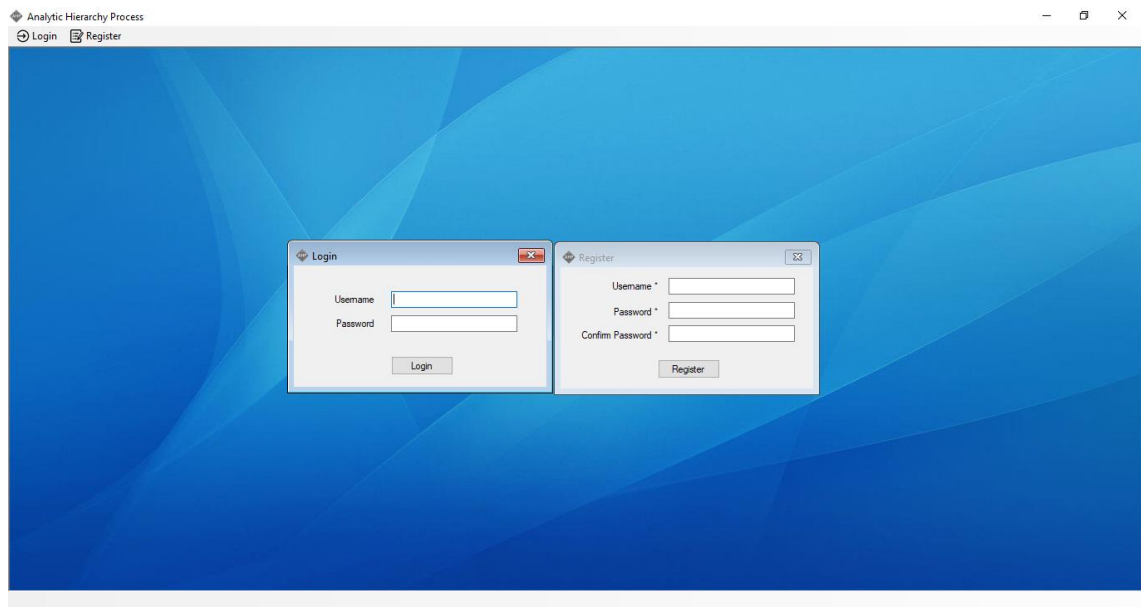
1. Aplikasi rekomendasi tipe *Graphics Processing Unit* (GPU) dengan metode *Analytic Hierarchy Process* (AHP)

Pada proyek ini, penulis berperan untuk membuat aplikasi berbasis *desktop* dan juga menerapkan ilmu Matematika dalam mempermudah dan mempercepat pembuatan keputusan berdasarkan data yang ada, dalam hal ini adalah menentukan GPU yang terbaik dengan menggunakan metode AHP. *Analytic Hierarchy Process* (AHP) adalah metode Matematika yang digunakan dalam membuat aplikasi rekomendasi. Gambar 2.1 menunjukkan alur perhitungan dari metode AHP yang dibuat oleh penulis berdasarkan teori pada subbab 2.3.2.4.



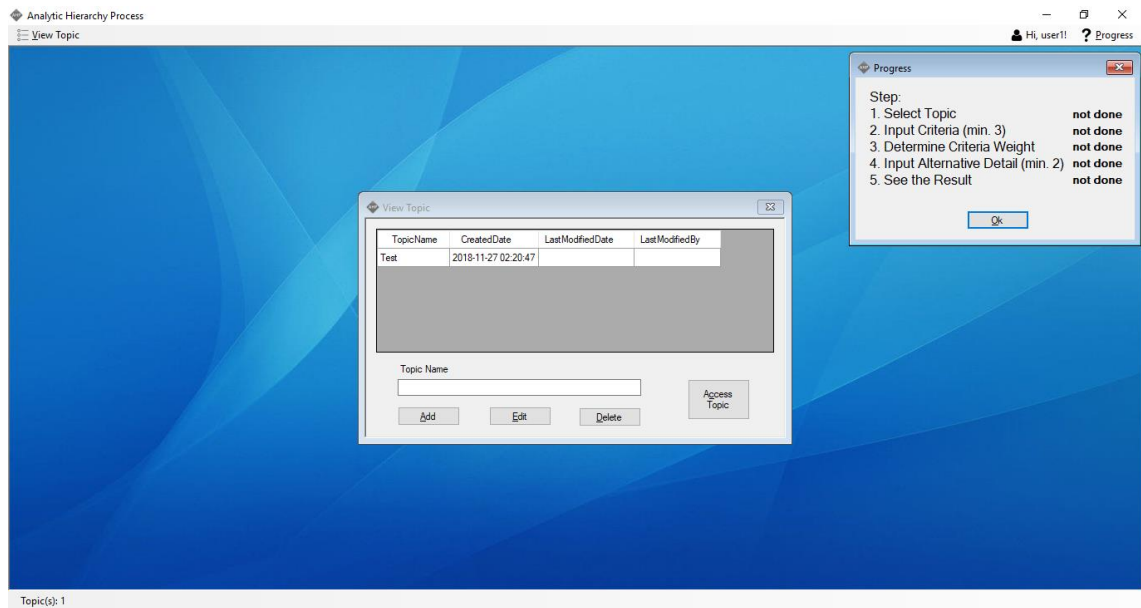
Gambar 2.1. Alur Perhitungan Metode AHP

Kriteria yang dimaksud adalah spesifikasi masing-masing GPU misalnya kapasitas memori, sedangkan alternatif yang dimaksud adalah tipe/merk GPU yang ingin dibandingkan. Gambar-gambar di bawah ini merupakan tampilan aplikasi yang telah dibuat dengan C# Windows Forms.



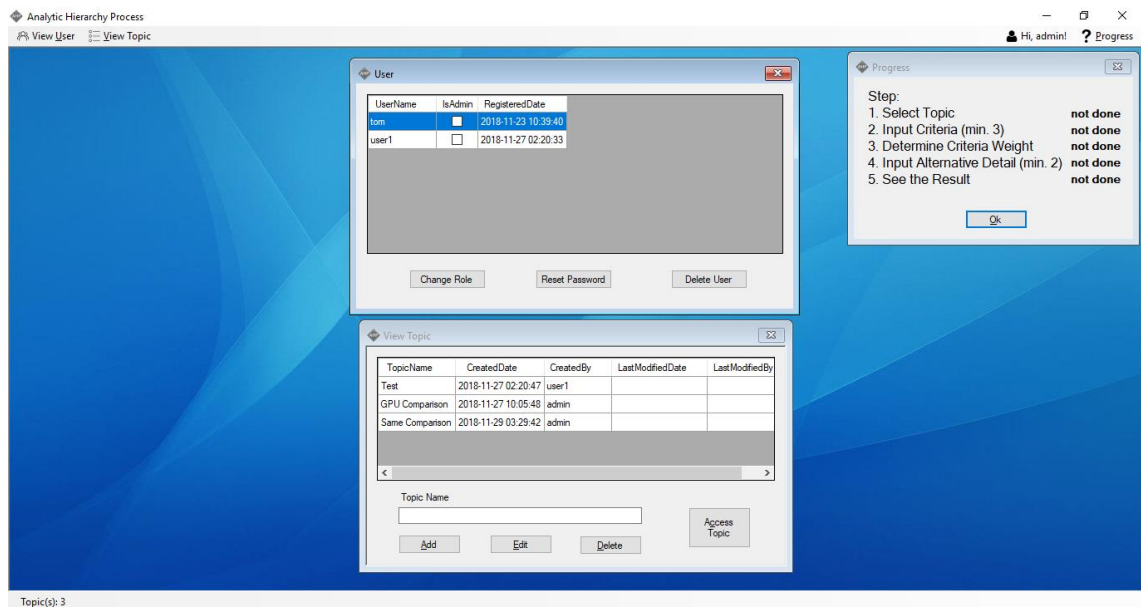
Gambar 2.2. Tampilan Aplikasi Awal Sebelum Login

Gambar 2.2 menunjukkan tampilan sebelum pengguna login ke aplikasi. Pengguna yang belum memiliki akun dapat melakukan registrasi terlebih dahulu dengan mendaftarkan *username* yang unik, *password*, serta konfirmasi *password*. Bagi pengguna yang sudah memiliki akun dapat memasukkan *username* dan *password* yang telah didaftarkan sebelumnya.



Gambar 2.3. Tampilan Aplikasi Setelah Login (Pengguna Biasa)

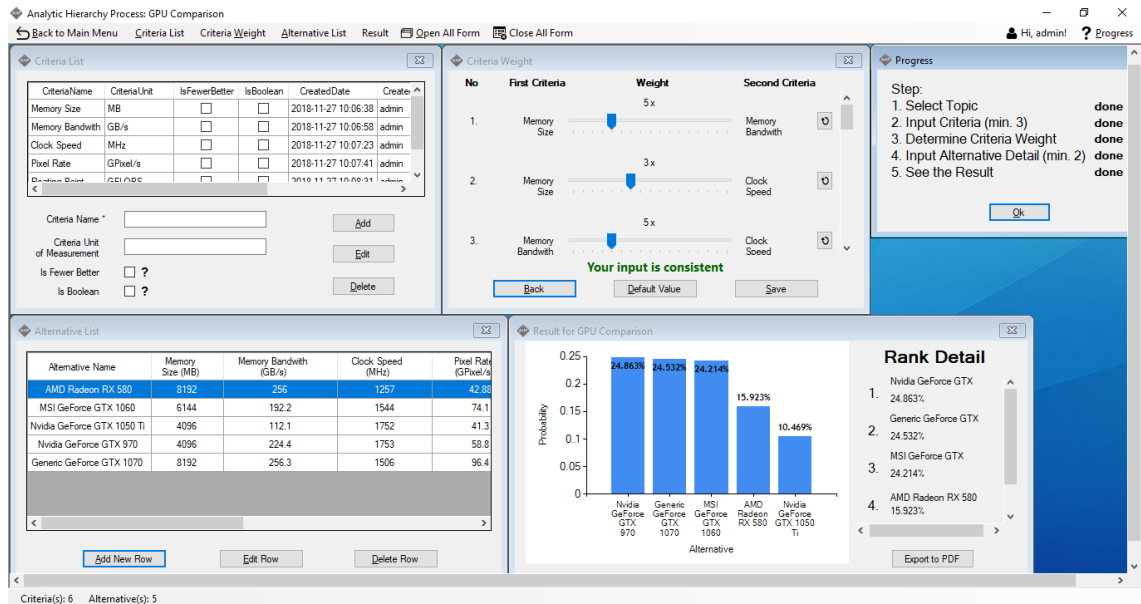
Gambar 2.3 menunjukkan tampilan aplikasi setelah login ke aplikasi sebagai pengguna biasa. Pengguna dapat menambahkan topik, yaitu objek yang ingin dibandingkan. Pengguna ini hanya dapat melihat, mengubah, dan menghapus topik yang dibuat oleh pengguna itu sendiri. Terdapat tombol “Access Topic” untuk melihat data yang tersimpan di topik tersebut.



Gambar 2.4. Tampilan Aplikasi Setelah Login (Pengguna Admin)

Gambar 2.4 menunjukkan tampilan aplikasi setelah login ke aplikasi sebagai admin. Fitur pada pengguna biasa juga terdapat pada admin. Perbedaannya adalah admin dapat melihat dan memanipulasi semua topik yang

tersimpan pada *database*, melihat semua pengguna yang terdaftar, mengganti *role* dari pengguna biasa ke admin maupun sebaliknya, mengganti *password* serta menghapus pengguna.



Gambar 2.5. Tampilan Aplikasi Setelah Memilih Topik

Gambar 2.5 menunjukkan tampilan aplikasi setelah pengguna memilih salah satu topik. Terdapat empat *form* utama yang ditampilkan:

- Criteria List*, digunakan untuk menambahkan, mengubah, dan menghapus kriteria pada topik. Yang dimaksud kriteria adalah parameter dalam pengambilan keputusan. Dalam kasus ini, kriteria adalah spesifikasi GPU. Untuk setiap topik, banyaknya kriteria yang *diinput* harus berada di antara 2 sampai 6. Berikut adalah data yang *diinput*:
 - Criteria Name*, yaitu nama kriteria. Wajib diisi
 - Criteria Unit of Measurement*, yaitu satuan kriteria. Misalnya *megabyte* (MB). Tidak wajib diisi, hanya untuk mempermudah pengguna.
 - Is Fewer Better*, dicentang apabila semakin kecil nilai pada kriteria tersebut, maka semakin tinggi kemungkinan alternatif untuk dipilih. Misalnya harga GPU, semakin murah biasanya akan lebih diprioritaskan.
 - Is Boolean*, dicentang apabila nilai pada kriteria tersebut hanya berupa ya/tidak.

- b. *Criteria Weight*, digunakan untuk melakukan pembobotan secara subjektif terhadap masing-masing perbandingan berpasangan dari kriteria yang telah diinput pada *Criteria List*. Banyaknya perbandingan yang dilakukan adalah sebanyak $\frac{n(n-1)}{2}$, dimana n adalah banyaknya kriteria. Nilai yang dapat diberikan adalah bilangan bulat antara 1 sampai 9 beserta nilai resiprokalnya. Semakin besar bobot yang diberikan, semakin kriteria tersebut diprioritaskan dalam perhitungan.
- c. *Alternative List*, digunakan untuk menambahkan, mengubah, dan menghapus alternatif pada topik. Yang dimaksud alternatif adalah spesifikasi dari masing-masing objek yang ingin dibandingkan. Dalam kasus ini, alternatif adalah nilai dari masing-masing spesifikasi GPU. Untuk setiap topik, banyaknya alternatif yang diinput harus berada di antara 2 sampai 11. Nilai yang diinput harus lebih besar sama dengan nol.
- d. *Result*, yaitu menampilkan hasil perhitungan AHP dari data yang telah diinput pada tiga *form* sebelumnya. Hasil disajikan dalam bentuk diagram batang, diurutkan dari alternatif yang paling baik ke buruk. Nilai yang dihasilkan berupa probabilitas alternatif tersebut dipilih sebagai solusi dari permasalahan. Pada *form* ini juga terdapat fitur *export* hasil perhitungan ke file pdf, berisi data yang diinput pengguna, diagram batang, serta kesimpulan yang dapat diambil. Contoh hasil *export* dapat dilihat pada Lampiran V.

Masalah yang dihadapi untuk proyek ini adalah penulis sempat bingung dalam menentukan bahasa pemrograman yang digunakan untuk membuat aplikasi *Graphical User Interface* (GUI), yaitu framework Python Kivy atau C# Windows Forms. Dengan Python Kivy, perhitungan matematis AHP lebih mudah dibandingkan di C#. Namun, hasil pendistribusian file .exe dan pembuatan tampilan aplikasi di Python Kivy lebih sulit dibandingkan di C#. Pada akhirnya, penulis memutuskan untuk menggunakan C# Windows Forms sebagai framework dalam mengembangkan aplikasi.

2. Dashboard mengenai data inventori *hardware* dan *software* dengan Microsoft PowerBI

Dashboard yang dimaksud adalah grafik interaktif yang bersifat deskriptif, menampilkan data mengenai inventori *hardware* dan *software*. Data yang digunakan adalah data yang tersedia di Microsoft *System Center Configuration Manager* (SCCM), yaitu sebuah sistem *management* yang digunakan untuk mengatur *lifecycle* dari semua perangkat yang menjalankan sistem operasi dari Microsoft. Fungsi utamanya adalah untuk meningkatkan produktifitas dan efisiensi dengan mengurangi tugas manual dalam mengatur sekumpulan user dan *device*. Salah satu fitur dari Microsoft SCCM adalah inventori, yang meliputi *hardware*, *software*, dan *asset intelligence*. Selama periode magang ini, penulis menyelesaikan dua *dashboard*:

- a. Menampilkan data inventori *hardware* berdasarkan tipe, merk, dan modelnya di masing-masing kantor BCA, dapat dilihat pada Lampiran VI.
- b. Menampilkan persentase pengguna aplikasi Microsoft Office terbaru berdasarkan masing-masing kantor BCA, dapat dilihat pada Lampiran VII.

Penulis juga menerapkan fitur *role-based access* pada kedua *dashboard*, yaitu membuat validasi agar grafik dan data yang disajikan sesuai dengan lokasi dimana user mengakses *dashboard* PowerBI. Misalnya, ketika pengguna di Kantor Pusat mengakses *dashboard* inventori *hardware*, maka hanya data di Kantor Pusat yang ditampilkan.

Masalah yang dihadapi untuk proyek ini adalah tidak adanya dokumentasi mengenai atribut dan tabel yang ada di Microsoft SCCM, sehingga penulis kesulitan untuk memulai proyek. Solusinya, penulis menanyakan kepada mentor tabel apa saja yang sering digunakan, eksplorasi secara mandiri dengan eksperimen terhadap data yang ada, dan melihat format penamaan tabel di internet agar mempermudah pencarian. Selain itu, data yang tersedia di Microsoft SCCM tidak dapat digunakan langsung untuk membuat *dashboard* karena belum sesuai kebutuhan dan tidak komplit, sehingga penulis harus melakukan penyiapan data sebelum melakukan visualisasi data, sebagai berikut:

- a. Melakukan pemetaan (*mapping*) secara manual, yaitu melihat kode perusahaan BCA dan memetakannya menjadi nama kantor wilayah atau departemen.
- b. Melakukan *scripting* Python di PowerBI, yaitu melakukan manipulasi data dengan menggunakan kode dan library Python.
- c. Melakukan *scrapping* tabel *telephone directory* dari web internal BCA yang berbentuk HTML dengan Python agar tidak perlu *mapping* secara manual. Kode Python dapat dilihat pada Lampiran VIII serta hasil dari *scrapping* dapat dilihat pada Lampiran IX.