

BAB II

TINJAUAN PUSTAKA

2.1 Merdeka Belajar Kampus Merdeka (MBKM)

Pembentukan kebijakan Merdeka Belajar – Kampus Merdeka didasarkan pada Peraturan Mendikbud No. 3 Tahun 2020, yang memberikan kesempatan bagi mahasiswa/i untuk mengasah kemampuan sesuai bakat dan minat dengan terjun langsung ke dunia kerja sebagai persiapan karier masa depan [4]. Program Kampus Merdeka memungkinkan mahasiswa untuk belajar selama 3 semester di luar program studinya, yang terdiri dari kegiatan magang di industri, studi independen, pembangunan desa, mengajar di sekolah, pertukaran mahasiswa, penelitian di lembaga riset, pengembangan kewirausahaan, proyek mandiri, dan proyek kemanusiaan [5].

Program ini bertujuan untuk memberikan kesempatan bagi mahasiswa untuk mendapat pengalaman nyata di lapangan yang sesuai dengan minat dan bakat masing-masing, agar siap menghadapi dunia kerja yang semakin kompetitif seiring dengan perkembangan zaman. Mahasiswa dapat memanfaatkan tantangan belajar di lingkungan selain di kelas atau perpustakaan untuk mengembangkan kreativitas, kemandirian, kemampuan memecahkan masalah, manajemen diri, serta interaksi sosial.

Mahasiswa yang tertarik mengikuti program Kampus Merdeka dapat mendaftar pada sistem yang telah disediakan pihak Kemendikbud-Ristek, yaitu *website* Merdeka Belajar – Kampus Merdeka. Selain itu, mahasiswa juga dapat melakukan pelaporan kegiatan dan hasil belajar. Sistem Merdeka Belajar – Kampus Merdeka telah diintegrasikan dengan data perguruan tinggi dan mahasiswa pada Pangkalan Data Pendidikan Tinggi (PDDIKTI) sehingga hanya mahasiswa aktif terdaftar di PDDIKTI yang memenuhi persyaratan dapat mendaftar dan melaporkan kegiatan pembelajaran pada sistem ini [6].

Program Kampus Merdeka yang dapat dipilih oleh mahasiswa dari *website* yaitu Indonesian International Student Mobility Awards, Kampus Mengajar, Kementerian ESDM – GERILYA, Magang, Membangun Desa (KKN Tematik), Pejuang Muda Kampus Merdeka, Pertukaran Mahasiswa Merdeka, Proyek Kemanusiaan, Riset atau Penelitian, Studi Independen, dan Wirausaha.

Di antara program-program tersebut, Membangun Desa (KKN Tematik), Proyek Kemanusiaan, Riset atau Penelitian, serta Wirausaha masih belum menerima pendaftaran mahasiswa pada periode 2020/2021 [7]. Sementara itu, program-program yang telah dilaksanakan yaitu sebagai berikut:

1. *International Student Mobility Awards*

Program ini merupakan program di mana mahasiswa Indonesia dapat menghabiskan 1 semester di universitas mitra di luar negeri untuk mempelajari dan mendalami kebudayaan negara tersebut serta meningkatkan *skill* dengan tugas-tugas yang praktikal [8].

2. Kampus Mengajar

Program ini memungkinkan mahasiswa menjadi mitra guru untuk berinovasi dalam pembelajaran literasi, numerasi serta adaptasi teknologi selama di SD dan SMP selama 1 semester dan berkesempatan untuk menjadi agen perubahan dalam dunia pendidikan Indonesia [9].

3. Kementerian ESDM – GERILYA

Program ini merupakan studi independen dari Kementerian Energi dan Sumber Daya Mineral bagi mahasiswa minimal semester 5 [10].

4. Magang

Program ini menawarkan pengalaman kerja di berbagai sektor industri seperti teknologi, kesehatan & pelayanan, BUMN, lembaga multilateral & non-profit, jasa dan lain-lain [11]. Melalui program Magang, mahasiswa dapat terlibat langsung dalam aktivitas internal institusi tempat magang dan mendapatkan ilmu yang relevan dari sana [12].

5. Pejuang Muda Kampus Merdeka

Program Pejuang Muda Kampus Merdeka menantang mahasiswa untuk belajar dari warga sekaligus berkolaborasi dengan Pemerintah Daerah, pemuka masyarakat, tokoh agama setempat serta seluruh *stakeholder* penggerak sosial di daerah [13].

6. Pertukaran Mahasiswa

Program ini adalah program di mana mahasiswa dapat mengeksplor dan mempelajari keberagaman budaya Nusantara, berteman dengan mahasiswa dari berbagai daerah, serta berkesempatan belajar di kampus lain di Indonesia [14].

7. Studi Independen

Program ini memberikan mahasiswa kesempatan untuk belajar kompetensi yang spesifik, praktis, dan dibutuhkan di masa depan. Mahasiswa juga dapat berinteraksi dengan para

pakar untuk memahami penerapannya serta mempraktekkan kompetensi tersebut dalam sebuah proyek riil [15].

2.2 Universitas Mikroskil

Universitas Mikroskil Medan adalah institusi pendidikan tinggi di bidang ilmu Komputer dan Ekonomi [16]. Universitas Mikroskil bermula dari lembaga kursus pendidikan komputer bernama IMC yang didirikan oleh Yayasan Bina Pertiwi pada tahun 1992. Berkat dukungan masyarakat dan perkembangan teknologi informasi yang pesat, IMC kemudian ditingkatkan menjadi Microskills – program pendidikan Diploma 1 (D-1) – pada tanggal 20 Maret 1994. Setelah itu, peningkatan jumlah mahasiswa dan permintaan perusahaan-perusahaan terhadap tenaga kerja lulusan Microskills mendorong perkembangan Microskills menjadi Sekolah Tinggi pada tanggal 1 Agustus 1997 dengan nama Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Mikroskil. Awalnya, STMIK Mikroskil hanya membuka program sarjana (S-1) Teknik Informatika, program diploma 3 (D-3) Manajemen Informatika dan Komputerisasi Akuntansi, serta program D-1 sebelumnya. Pada tahun 2008, Yayasan Bina Pertiwi Medan baru mendirikan Sekolah Tinggi Ilmu Ekonomi (STIE) Mikroskil. STMIK Mikroskil dan STIE Mikroskil telah terakreditasi BAN-PT untuk semua program studi yang ada [17].

Kemudian, pada tanggal 27 Agustus 2021, STMIK Mikroskil dan STIE Mikroskil bergabung menjadi Universitas Mikroskil dengan dua fakultas. Pertama yaitu Fakultas Informatika dengan program studi Magister Teknologi Informasi (S2), Teknik Informatika (S1), Sistem Informasi (S1) dan Teknologi Informasi (S1) serta kedua yaitu Fakultas Bisnis dengan program studi Manajemen (S1) dan Akuntansi (S1) [18]. Program S-1 pada Fakultas Informatika memiliki beban kredit 144 s.d. 150 SKS yang dijadwalkan selesai dalam 8 semester dan memberikan hak menggunakan gelar Sarjana Komputer (S.Kom.) sesuai ketentuan yang berlaku bagi lulusannya. Sedangkan program studi S-1 pada Fakultas Bisnis juga memiliki beban kredit 144 s.d. 150 SKS yang dijadwalkan selesai dalam 8 semester serta memberikan hak bagi lulusan untuk menggunakan gelar Sarjana Ekonomi (S.E.) sesuai ketentuan yang berlaku [19].

Dalam pelaksanaan perkuliahan setiap mahasiswa dan dosen Universitas Mikroskil dapat mengakses fitur MIKA. MIKA merupakan portal akademik Universitas Mikroskil yang dimanfaatkan sebagai tempat bagi para mahasiswa Universitas Mikroskil melakukan

pengecekan status pembayaran uang kuliah, pengisian KRS, melihat hasil rekapitulasi nilai, melihat daftar kehadiran absensi, dan informasi-informasi lainnya seputar lingkungan Universitas Mikroskil. MIKA juga dimanfaatkan oleh dosen dalam melaksanakan aktivitas akademik, seperti melihat berita acara dan absensi perkuliahan mahasiswa, melihat jadwal mengajar dan ujian, memberikan nilai akademik mahasiswa dan lain sebagainya [20].

2.3 Integration Test

Integration test adalah sebuah pengujian yang memeriksa interaksi yang terjadi antar komponen pada sebuah perangkat lunak yang akan dibangun. Tujuan dari *integration test* ini adalah memeriksa kinerja dari aplikasi apakah sudah berjalan dengan baik. Tugas utama *integration testing* adalah untuk menganalisis parameter seperti persyaratan fungsional, persyaratan kinerja dan persyaratan keandalan yang ditempatkan pada item desain utama [31].

Pendekatan *integration test* terbagi atas:

1. Pendekatan *Big Bang*

Pengujian dengan pendekatan dengan pendekatan *Big Bang* dilakukan ketika semua modul telah dikerjakan dan digabungkan. Pengujian ini dilakukan secara satu kesatuan ketika semua modul sudah menjadi sebuah perangkat lunak. Kelebihan dari pendekatan *big bang* adalah cocok untuk sistem yang kecil, sedangkan kekurangan dari pendekatan ini adalah sumber daya yang kita miliki tidak produktif dikarenakan harus menunggu semua modul diselesaikan terlebih dahulu, baru pengujian dapat dilakukan. Selain itu kekurangan dari pendekatan ini adalah sulit menemukan sumber masalah ketika ada kesalahan dalam perangkat lunak.

2. Pendekatan *Bottom Up*

Pendekatan *Bottom Up* dilakukan dari pengujian terhadap modul yang tekecil dahulu. Setelah pengujian modul kecil selesai maka akan diintegrasikan terhadap modul yang lebih tinggi supaya dapat dilakukan pengujian lebih lanjut. Proses ini dilakukan secara berulang sampai tahap pengujian selesai dilakukan. Kelebihan dari pengujian dengan pendekatan *bottom up* adalah tidak ada waktu yang terbuang ketika menunggu modul lain diselesaikan terlebih dahulu dan penemuan sumber masalah lebih mudah ditemukan, sedangkan kekurangan dari pendekatan ini adalah tidak dapat menyediakan *prototype* awal.

3. Pendekatan *Up Bottom*

Pendekatan *Up Bottom* adalah pendekatan kebalikan dari pendekatan *Bottom Up*. Pengujian dengan Pendekatan *Up Bottom* dimulai dengan modul yang tertinggi dahulu. Setelah modul tertinggi diuji, maka selanjutnya dilakukan pengujian terhadap modul yang lebih rendah. Kelebihan dari pendekatan ini adalah dimungkinkan munculnya *prototype* awal, kemudahan dalam melacak masalah ketika terjadi kegagalan di sistem yang dirancang, sedangkan kekurangan dari pendekatan ini adalah adanya banyak versi desain yang dibutuhkan.

2.4 Test Driven Development

TDD (*Test Driven Development*) merupakan pendekatan pengembangan perangkat lunak yang bertujuan untuk menghasilkan kode pemrograman yang bersih dengan *bug* minimal. Hal ini dapat dicapai dengan cara membuat kode test terlebih dahulu sebelum memulai pengembangan, sehingga kode yang dibangun harus dapat menyelesaikan test yang telah dibuat sebelumnya [32]. Oleh karena itu, diperlukan *requirements* dan desain yang jelas sebelum penulisan kode dimulai, sekaligus dapat menyesuaikan kode dengan mudah jika terjadi perubahan requirement. Cara untuk melakukan *test driven development* adalah pertama membuat *unit testing* terlebih dahulu. Selanjutnya, membuat kode program. Setelah itu, menjalankan semua *testing* yang sudah anda rancang sebelumnya dan lihat apakah ada *testing* yang gagal. Jika ada *testing* yang gagal, maka lakukan perubahan pada kode program kita. Lakukan hal ini berulang sampai semua hasil *testing* tidak ada yang salah [33].

2.5 User Requirements

User requirements merujuk pada fitur atau atribut yang harus dimiliki atau dilakukan oleh produk menurut sudut pandang pengguna [34]. Proses pembuatan *user requirements* merupakan bagian dari proses *User-centered Design* (UCD), yaitu pendekatan pengembangan produk atau *software* yang berfokus pada *end user* produk tersebut. Pendekatan ini berprinsip bahwa produk akhir harus disesuaikan dengan *user*, bukan *user* yang harus disesuaikan bagi produk tersebut. Oleh karena itu, setiap teknik, proses, dan metode pada *lifecycle* pengembangan produk berfokus pada *user*.

Terdapat beberapa metode pengumpulan *user requirements*, dengan lama pelaksanaan dan biaya yang berbeda. Metode-metode tersebut adalah sebagai berikut:

1. Wawancara

Wawancara merupakan pengumpulan informasi mendalam dari setiap *user*, membutuhkan pewawancara handal agar dapat mengumpulkan lebih banyak informasi, serta membutuhkan waktu lama hingga sedang.

2. Survei

Dengan survei, data kuantitatif dapat dikumpulkan dari user yang banyak dalam waktu singkat dan biaya relatif murah.

3. Analisis kebutuhan

Analisis kebutuhan adalah pengumpulan daftar prioritas mengenai keinginan dan kemauan para *user*.

4. *Card sort*

Metode ini mengidentifikasi bagaimana *user* mengelompokkan informasi atau objek agar dapat mengetahui arsitektur informasi produk. *Card sort* relatif simpel untuk dilakukan, tergantung pada *tool*, banyak *card*, dan banyak partisipan.

5. *Group task analysis*

Analisis dilakukan untuk memahami bagaimana *user* menyelesaikan suatu *task* dan apa saja kendala yang dihadapi sehingga menghasilkan *task flow* yang dapat bekerja pada berbagai *user*. Metode ini membutuhkan biaya cukup rendah dan mudah dilakukan.

6. Grup fokus

Cara pelaksanaan metode ini yaitu dengan mengumpulkan data dari beberapa *user* sekaligus untuk menaksir pendapat dan kesan *user* terhadap produk. Biaya yang dibutuhkan cukup rendah, tetapi kemungkinan sulit mengajak cukup *user* agar diskusi kelompok lebih efektif.

7. Kunjungan lapangan

Dengan melakukan kunjungan lapangan, pemahaman mengenai apa yang diinginkan *user* lebih mudah karena pengembang dapat melihat sendiri *user*, lingkungan mereka, serta *task* yang ada dalam konteks. Akan tetapi, kemungkinan akan sulit mengatur kunjungan ke lapangan sekaligus mengatur analisis data yang didapatkan.