

BAB 1 PENDAHULUAN

1.1. Latar Belakang

Di Indonesia, diabetes menjadi masalah kesehatan serius dengan lonjakan kasus dari 7,3 juta pada 2011 menjadi 19,5 juta pada 2021 [1]. Gaya hidup tidak sehat dan pola makan yang kurang seimbang menjadi penyebab utama [1]. Tingginya aktivitas di kota-kota besar membuat menjaga pola makan sehat semakin sulit, meskipun kesadaran telah meningkat. Santapan hadir sebagai solusi dengan menyediakan makanan sehat yang sesuai dengan kondisi pengguna, didukung antarmuka yang ramah dan berbasis mobile untuk kemudahan akses. Berbeda dari platform seperti Yellow Fit dan Kulina [2], Santapan menawarkan pendekatan lebih personal. Untuk merancang model bisnis yang responsif terhadap kebutuhan pasar, Lean Canvas menjadi metode yang efektif. Permasalahan *startup* pada tahap awal adalah memaksakan ide yang digunakan adalah suatu ide yang baru tanpa memikirkan kebutuhan pasar [3]. Maka dari itu untuk memaksimalkan peluang dan validasi ide yaitu dengan menggunakan Lean Canvas, dapat dengan cepat merumuskan asumsi-asumsi tentang model bisnis [4][5]. Pendekatan ini memungkinkan untuk secara efisien menguji hipotesis, melakukan iterasi, dan membuat penyesuaian berdasarkan *feedback* [4]. Lean Canvas tidak hanya membantu mencapai kesesuaian masalah atau solusi, tetapi juga memungkinkan startup untuk memaksimalkan peluang dalam memenuhi kebutuhan pasar [5]. Dalam perusahaan manapun kunci dalam keberhasilan adalah menurunkan biaya [6]. Maka perancangan *cost structure* penting di Lean Canvas. Menurut [7], agar suatu bisnis memiliki strategi yang sukses, bisnis tersebut harus memproduksi produk dengan biaya serendah mungkin, termasuk fungsi produksi, pemasaran, dan dukungan non-pemasaran. Lean Canvas tidak hanya untuk mengidentifikasi masalah atau solusi tetapi juga untuk mengidentifikasi struktur biaya [5], yang penting

untuk keberlanjutan operasi bisnis. Pengurangan biaya merupakan faktor penting untuk keberhasilan bisnis apa pun [6], terutama bagi *startup* dengan sumber daya terbatas. Menurut [7], “peran biaya terutama ditujukan sebagai struktur biaya dalam kaitannya dengan pendapatan dan profitabilitas”. Banyak *startup* yang membakar uang untuk menciptakan *brand awerness* dan *loyalty* [8]. Pentingnya memahami struktur biaya yang penting untuk mengelola pengeluaran dan memastikan efisiensi operasional. Proses ini bergantung pada proses lain seperti pemilihan arsitektur yang tepat. Software architecture pada *startup* memainkan peran penting dalam mengoptimalkan kebutuhan bisnis.

Bagi *startup*, memilih *software architecture* yang tepat cukup menantang. *Startup* mengembangkan produk perangkat lunak intensif yang inovatif di bawah batasan waktu dan dengan kurangnya sumber daya [9]. Banyak organisasi, dari perusahaan teknologi besar (misalnya, Google, Amazon, Netflix) hingga perusahaan rintisan kecil, telah mengadopsi *microservice* sebagai *best practice* [10][11]. *Startup* memilih *software architecture microservice* [11] yang mungkin tidak sesuai dengan struktur biaya mereka. Ketidakcocokan ini dapat mengakibatkan tantangan operasional, termasuk peningkatan biaya. Meskipun *microservice* adalah *software architecture* yang populer menurut beberapa penelitian [10][11][12]. *Microservice* tidak selalu merupakan pilihan yang paling hemat biaya [10]. Perusahaan rintisan harus mengevaluasi apakah kompleksitas dan biaya operasional yang lebih tinggi dapat mengakomodasi *microservice* [10][12]. Arsitektur seperti *monolithic* mungkin lebih sesuai untuk kebutuhan *startup*. *Microservice* memang menawarkan keuntungan yang jelas, seperti *modularity*, *scalability*, dan kemampuan untuk mengembangkan dan menyebarkan komponen secara independen [10][12], Ini menarik bagi bisnis yang mencari pertumbuhan yang cepat. Namun, seperti yang dicatat dalam penelitian tersebut. Hal ini dapat mengakibatkan peningkatan biaya untuk infrastruktur, pemeliharaan, dan pengembangan. Untuk *startup* yang

beroperasi dengan anggaran terbatas, biaya-biaya dapat menjadi mahal, yang mengakibatkan inefisiensi dan tantangan dalam mengembangkan bisnis. Di sisi lain, *monolithic* dapat menjadi pertimbangan. *Monolithic* meskipun kurang fleksibel dari *microservice*, menawarkan keuntungan yang signifikan dalam efisiensi biaya seperti yang dibahas dalam penelitian ini [10]. *Monolithic* menggabungkan semua basis kode menjadi satu, membuatnya lebih mudah untuk dikembangkan [10][12]. Untuk pengembangan *startup* tahap awal yang kebutuhannya masih sedikit, *monolithic* merupakan pilihan yang baik. Arsitektur ini dapat mengurangi biaya operasional, meminimalkan kompleksitas, dan menyederhanakan pemeliharaan sistem, faktor-faktor utama yang penting untuk pengembangan perusahaan rintisan tahap awal.

Untuk memastikan bahwa arsitektur yang dipilih sesuai dengan kebutuhan operasional *startup*, diperlukan evaluasi sistematis. ATAM merupakan cara sistematis untuk menilai konsekuensi keputusan arsitektur dengan mempertimbangkan persyaratan atribut kualitas [13][14] seperti biaya, *scalability*, dan *performance*. Dengan pendekatan Analysis Tradeoff Architecture Method (ATAM), pendekatan ini dapat mengevaluasi arsitektur perangkat lunak berdasarkan skenario dan atribut kualitas yang telah ditentukan [15][16][17]. Dengan ATAM, *startup* dapat mengidentifikasi risiko dan memprioritaskan pilihan *software architecture* yang sesuai dengan tujuan bisnis dan kendala sumber daya mereka. *Startup* dapat menggunakan ATAM untuk mengevaluasi apakah *monolithic* atau *microservice* lebih sesuai dengan kebutuhan mereka dengan mempertimbangkan faktor-faktor seperti *traffic* yang diharapkan, keahlian tim, kendala anggaran, dan lainnya. ATAM merupakan metode yang melibatkan banyak *stakeholders* [17][18]. Dalam penelitian ini, ATAM juga memfasilitasi diskusi di antara *stakeholders* tentang bagaimana keputusan arsitektur menangani atribut kualitas [17]. Evaluasi tersebut merinci analisis tentang bagaimana setiap *software architecture* sesuai dengan atribut [19]. Termasuk aspek-aspek seperti struktur biaya.

Terutama bagi perusahaan rintisan untuk mencapai keseimbangan yang tepat antara *scalability* dan struktur biaya.

1.2. Rumusan Masalah

Setelah memahami latar belakang dan konteks penelitian, langkah berikutnya adalah menetapkan rumusan masalah yang menjadi fokus penelitian. Maka dari itu, rumusan masalah yang dipilih sebagai berikut:

1. Bagaimana pengembangan *cost structure* yang sesuai pada Santapan jika dirancang dengan *Lean Canvas*?
2. Bagaimana menyesuaikan *software architecture monolithic* dan *microservice* yang sesuai dengan struktur biaya menggunakan ATAM, dengan mempertimbangkan *quality attributes* serta implikasinya?

1.3. Tujuan dan Manfaat

Tujuan utama penelitian adalah untuk mengembangkan model bisnis yang efektif dengan mengintegrasikan rancangan *software architecture* yang tepat pada implementasi sistem *backend*. Poin penting dalam mencapai tujuan meliputi:

1. Menyusun *cost structure* yang sesuai dengan kebutuhan dari startup Santapan.
2. Memastikan *cost structure* yang dikembangkan tidak terjadi kesenjangan dari rancangan *software* yang ada, sehingga mendukung keberlangsungan operasional pada sistem.
3. Melakukan analisis *trade-off* menggunakan ATAM (*Architecture Tradeoff Analysis Method*) untuk mengevaluasi kebutuhan *quality attributes* serta menilai *architecture* yang sesuai antara manfaat dan risiko.

1.4. Batasan Masalah

Untuk mencegah luasnya pembahasan yang berpotensi menghilangkan fokus penelitian, maka batasan-batasan untuk penelitian dan perancangan adalah sebagai berikut:

1. Model bisnis yang sesuai hanya dibatasi dengan konsep *Lean Canvas* dan pada elemen *cost structure*.
2. Batasan jumlah *stakeholder* yang dilibatkan dalam ATAM, yang difokuskan pada pihak-pihak utama yang berperan dalam keputusan arsitektur, seperti *end-user*, *developers*, dan *management* dapat dilihat pada **Tabel 1.1**.
3. Arsitektur yang dianalisis hanya terbatas pada *monolithic* dan *microservice* dan pada tahap awal perkembangan *startup* dengan perbandingan berdasarkan *performance*, *scalability* dan *security*.

Tabel 1.1 Batasan dan Justifikasi Stakeholders

Scope	Description
Management	1. Dana Sulistyo Kusumo, ST, MT, Ph.D. sebagai Scrum Master 2. Jan Falih Fadhillah sebagai Product Owner
Developers	1. Jan Falih Fadhillah sebagai Backend Engineer 2. Pahala Fawwaz Nazmi M sebagai Mobile Application Engineer
End-Users	Pengguna aplikasi santapan

1.5. Metode Penelitian

Metode penelitian pada penelitian menggunakan Lean Canvas untuk merancang bisnis model, dan menggunakan ATAM dalam mencari kebutuhan dari *software architecture* yang digunakan untuk mencari kesenjangan dari *cost structure* pada sebuah bisnis *startup*. Dengan membandingkan *monolithic* dan *microservice* dan mencari yang mana *software architecture* lebih efektif untuk tahap awal bisnis *startup* teknologi.