

BAB II LANDASAN TEORI

2.1 Tinjauan Teori Umum

Teori umum merupakan sebuah teori pokok yang menjadi landasan penelitian. Teori-teori yang dibahas pada tinjauan teori umum berlaku benar secara universal dalam semua keadaan, waktu dan tempat.

2.1.1 Kredit

Asal kata kredit yaitu dari kata latin "*credo*" dengan arti "saya percaya" dan sebagai gabungan bahasa sanskerta "*cred*" dengan arti "kepercayaan" dan bahasa latin "*do*" dengan arti "saya tempatkan". Mendapatkan pinjaman berupa kredit sama artinya dengan mendapatkan rasa percaya. Rasa percaya inilah yang membuat seseorang diberi uang, barang atau jasa dengan syarat pembayaran kembali sesuai jangka waktu yang disepakati (Sibagariang & Situmorang, 2019).

Dalam UU 10/1998 tentang Perubahan Undang-Undang Nomor 7 Tahun 1992 tentang Perbankan di Pasal 1 ayat 11 kredit didefinisikan sebagai: "penyediaan uang atau tagihan yang dapat dipersamakan dengan itu berdasarkan persetujuan atau kesepakatan pinjam-meminjam antara bank dengan pihak lain yang mewajibkan pihak peminjam untuk melunasi utangnya setelah jangka waktu tertentu dengan pemberian bunga."

Indikator yang digunakan pada penelitian sebelumnya dan paling mendekati atau menyerupai analisa pembiayaan kredit pada umumnya yakni prinsip 5C

(Aisyah, 2019) menggunakan kriteria-kriteria kedisiplinan, uang muka, penghasilan, jaminan, dan status tempat tinggal dengan pemberian bobot sebagai berikut:

1. C1 = Kedisiplinan:

Riwayat kredit macet bobot 10; history kredit lunas dengan banyak tunggakan bobot 20; belum pernah kredit bobot 40; history kredit lunas dengan sedikit tunggakan bobot 80; dan history kredit lunas dan lancar bobot 100.

2. C2 = Uang Muka:

DP <15% dari harga bobotnya 20; DP 16-20% dari harga bobotnya 40; DP 21-25% dari harga bobotnya 60; DP 26-30% dari harga bobotnya 80; DP >30% dari harga bobotnya 100.

3. C3 = Penghasilan:

Penghasilan \leq 1 juta bobot 30; penghasilan antara 1 juta sampai 2 juta bobot 50; penghasilan antara 2 juta sampai 5 juta bobot 60; penghasilan antara 5 juta sampai 10 juta bobot 80.

4. C4 = Jaminan:

BPKB Motor bobot 50; BPKB Mobil bobot 75; Sertifikat Tanah bobot 100.

5. C5 = Status Tempat Tinggal:

Kost atau kontrak bobot 20; KPR bobot 40; Rumah dinas atau perusahaan bobot 60; rumah orangtua atau keluarga bobot 80; rumah milik sendiri bobot 100.

Indikator rancangan penelitian saat ini menggunakan menggunakan analisa yang menyerupai analisa 5C namun lebih lengkap yakni analisa 7P yang terdiri atas: kepribadian (*personality*), kondisi keuangan atau tunggakan yang menjadi faktor pengurang (*Party*), tujuan penggunaan dana (*Purpose*), sektor ekonomi (*Prospect*), sisa hutang atau jumlah pembayaran (*Payment*), rata-rata pendapatan (*Profitability*), dan nilai agunan (*Protection*), dengan bobot sebagai berikut:

1. P1 = Kepribadian:

Lancar atau tidak ada kredit bobot 5; dalam perhatian khusus bobot 4; kurang lancar bobot 3; diragukan bobot 2; dan macet bobot 1.

2. P2 = Persentase Tunggakan terhadap sisa hutang:

0-10% bobot 5; 11-30% bobot 4; 31-50% bobot 3; 51-70% bobot 2; diatas 71% bobot 1.

3. P3 = Tujuan Penggunaan Dana:

Konsumsi Primer bobot 5; konsumsi sekunder bobot 4; modal kerja bobot 3; investasi bobot 2, konsumsi tersier bobot 1.

4. P4 = Sektor Ekonomi:

Termasuk 10 besar sektor terbaik bobot 5; termasuk 20 besar sektor terbaik bobot 4; termasuk 50 besar sektor terbaik bobot 3; tidak termasuk sektor terbaik bobot 2; termasuk sektor yang mengalami penurunan bobot 1.

5. P5 = Persentase sisa hutang:

0-10% bobot 5; 11-30% bobot 4; 31-50% bobot 3; 51-70% bobot 2; diatas 71% bobot 1.

6. P6 = Persentase Rata-Rata Pendapatan Disetahunkan dibanding sisa hutang:

Diatas 71% bobot 5; 51-70% bobot 4; 31-50% bobot 3; 11-30% bobot 1; dibawah 10% bobot 1.

7. P7 = Persentase Nilai Agunan dibanding sisa hutang:

Diatas 71% bobot 5; 51-70% bobot 4; 31-50% bobot 3; 11-30% bobot 1; dibawah 10% bobot 1.

2.1.2 Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah proses memutuskan suatu masalah dengan alat bantu teknologi berdasarkan fakta dan metode tertentu untuk mempermudah pengambilan keputusan dengan akurat serta tepat sasaran (Mubarak et al., 2019). Sesuai dengan namanya, tujuan dibuatnya sistem ini adalah sebagai pendapat alternatif dan sebagai bahan pertimbangan dalam pengambilan keputusan maupun kebijakan tertentu, dan menjadikannya dinamis sehingga membantu pengguna dalam menciptakan gagasan dan batasan permasalahan sesuai yang diinginkan (Nawawi et al., 2019). Kegunaan sistem Pendukung Keputusan yaitu memberikan pihak manajemen bantuan berupa alternatif dukungan dalam mengambil sebuah keputusan (Kartiko et al., 2021). Penggambaran ini hanya terbatas pada kegiatan yang memberikan bantuan pada atasan-atasan dalam struktur organisasi ketika melakukan penilaian dapat juga membantu saat posisi dan peran manajer sedang digantikan. Sistem Pendukung Keputusan memberikan solusi

permasalahan atau mengevaluasi peluang. Pada dasarnya sistem pendukung keputusan dibuat dengan tujuan agar segala aktivitas mulai dari langkah awal sampai dengan akhir dalam menyimpulkan maupun dalam membuat keputusan dapat terbantu (Supiyandi et al., 2020).

2.1.3 Konsep Pengambilan Keputusan

Dalam mengatasi permasalahan, manusia telah mengembangkan sistem yang membantu memecahkan masalah. Salah satunya adalah dengan membuat keputusan terhadap sebuah permasalahan (Arnomo, 2021). Seiring berjalannya pemecahan masalah terbantu dengan adanya konsep pengambilan keputusan.

Konsep pengambilan keputusan pertama kali diperkenalkan Michael Scott Morton pada awal tahun 1970-an kemudian istilah "*Management Decision System*" mulai dikenal (Dzaka et al., 2021). Konsep ini berupa suatu sistem yang basis komputer yang interaktif dalam membantu dalam pengambilan keputusan dengan menggunakan data serta model dalam menyelesaikan permasalahan serta memiliki sifat tak berstruktur maupun setengah terstruktur (Rahmah, 2021).

Sistem Pendukung Keputusan dirancang untuk mendukung seluruh tahap pengambilan keputusan sejak awal mulai yaitu tahap identifikasi permasalahan, relevansi pemilihan data, penentuan alur pengambilan keputusan hingga aktivitas evaluasi dalam memilih alternatif. Dalam prosesnya, sebuah penentuan keputusan dilakukan melalui tiga fase utama yakni inteligensi, desain serta kriteria hingga ditambahkan fase berikutnya yakni implementasi (Daulay, 2020).

2.2 Tinjauan Teori Khusus

Tinjauan teori khusus merupakan teori yang dipergunakan dalam penelitian sesuai dengan pembahasan ataupun perancangan yang dilakukan. Teori ini dikaitkan bersamaan kebenaran faktual sebuah penjelasan tertentu yang digunakan sebagai acuan penyelesaian permasalahan.

2.2.1 Logika Fuzzy

Logika *fuzzy* yaitu bagian dari sebuah komponen berbentuk *soft computing* dan diperkenalkan pada mulanya oleh Prof. Lofti A. Zadeh pada tahun 1965 (Sibagariang & Situmorang, 2019). Alasan dibalik penggunaan logika fuzzy, yaitu karena konsep logika *fuzzy* ini, yaitu digunakan konsep teori himpunan dalam logika *fuzzy* sehingga mudah dimengerti, fleksibel dan memiliki kemampuan adaptasi yang tinggi terhadap perubahan maupun pada hal-hal yang tidak pasti dalam setiap permasalahan (Syakir et al., 2021).

2.2.2 Fuzzy Multiple Attribute Decision Making (FMADM)

Dasar pemrosesan *Fuzzy Multiple Attribute Decision Making* ini dapat terjadi dalam tiga tahapan, yakni dengan menyusun komponen situasi terlebih dahulu, analisa, serta menyintesis informasi (Krismawati et al., 2021).

Pada tahap penyusunan komponen, komponen situasi, akan dibentuk tabel taksiran yang berisi identifikasi alternatif dan spesifikasi tujuan, kriteria dan atribut. Salah satu cara untuk menspesifikasikan tujuan situasi $|O_i, i = 1, \dots, t|$ adalah dengan cara mendaftar konsekuensi-konsekuensi yang mungkin dari alternatif yang telah

teridentifikasi $|A_i, i = 1, \dots, n|$. Selain itu juga disusun atribut-atribut yang akan digunakan $|A_k, k=1, \dots, m|$. Metode *Fuzzy Multiple Attribute Decision Making* (FMADM) adalah sebuah metode yang dipergunakan dalam menemukan alternatif terbaik dari antara alternatif lainnya dengan kriteria tertentu (Nurjaman et al., 2021).

2.2.3 *Technique for Orders Preference by Similarity to Ideal Solution (TOPSIS)*

Salah satu metode pengambilan keputusan multikriteria ini pada mulanya diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus memiliki jarak paling dekat dengan solusi ideal positif juga jarak paling jauh dengan solusi ideal negatif (Nawawi et al., 2019).

Solusi ideal positif merupakan hasil penjumlahan aspek dengan nilai tertinggi yang bisa dicapai setiap atribut begitu pula solusi ideal negatif merupakan penjumlahan aspek dengan nilai terendah yang bisa dicapai setiap atribut (Nawawi et al., 2019). Metode ini kerap diterapkan karena lebih mudah dan efisien komputasinya, juga mempunyai kemampuan pengukuran kinerja relatif dari alternatif-alternatif keputusan.

Dalam metode TOPSIS klasik, nilai bobot dari setiap kriteria telah diketahui dengan jelas. Setiap bobot kriteria ditentukan berdasarkan tingkat kepentingannya menurut pengambil keputusan.

Langkah-langkah penyelesaian masalah dengan metode TOPSIS:

1. TOPSIS dimulai dengan membangun sebuah matriks keputusan. Matriks keputusan X mengacu terhadap m alternatif yang akan dievaluasi berdasarkan n kriteria.

Matriks keputusan X dapat dilihat pada persamaan berikut:

$$X = \begin{matrix} & a_1 & \cdots & a_n \\ \begin{matrix} a_1 \\ \vdots \\ a_m \end{matrix} & \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \end{matrix} \quad \text{Rumus 2. 1 Matriks Keputusan X}$$

dimana a_i ($i = 1, 2, 3, \dots, m$) adalah alternatif-alternatif yang mungkin, x_j ($j = 1, 2, 3, \dots, n$) adalah atribut dimana performansi alternatif diukur, x_{ij} adalah performansi alternatif a_i dengan acuan attribute x_j .

2. Membuat matriks keputusan yang ternormalisasi

Persamaan yang digunakan untuk mentransformasikan setiap elemen x_{ij} adalah:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad \text{Rumus 2. 2 Matriks keputusan ternormalisasi}$$

dengan $i = 1, 2, 3, \dots, m$; dan $j = 1, 2, 3, \dots, n$; dimana r_{ij} adalah elemen dari matriks keputusan yang ternormalisasi R . x_{ij} adalah elemen matriks dari keputusan X .

3. Membuat matriks keputusan yang ternormalisasi terbobot

Dengan bobot $w_i = (w_1, w_2, w_3, \dots, w_n)$ dimana w_j adalah bobot dari kriteria ke- j dan $\sum_{j=1}^n w_j = 1$ maka normalisasi bobot matriks V adalah:

$V_{ij} = w_j r_{ij}$ **Rumus 2. 3** Matriks keputusan ternormalisasi terbobot

dengan $i = 1, 2, 3, \dots, m$; dan $j = 1, 2, 3, \dots, n$; dimana v_{ij} adalah elemen dari matriks keputusan yang ternormalisasi terbobot V.

w_{ij} adalah bobot dari kriteria ke-j

r_{ij} adalah elemen dari matriks keputusan yang ternormalisasi R

4. Menentukan matriks solusi ideal positif dan solusi ideal negatif

Solusi ideal positif dinotasikan A^+ , sedangkan solusi ideal negatif dinotasikan

A^- . Berikut ini adalah persamaan dari A^+ dan A^- :

$$\begin{aligned} \text{a. } A^+ &= \{(\max v_{ij} | j \in J), (\min v_{ij} | j \in J'), i = 1, 2, 3, \dots, m\} \\ &= \{v_1^+, v_2^+, v_3^+, \dots, v_n^+\} \end{aligned}$$

$$\begin{aligned} \text{b. } A^- &= \{(\min v_{ij} | j \in J), (\max v_{ij} | j \in J'), i = 1, 2, 3, \dots, m\} \\ &= \{v_1^-, v_2^-, v_3^-, \dots, v_n^-\} \end{aligned} \quad \text{Rumus 2. 4 Matriks solusi ideal positif}$$

dan solusi ideal negatif

$J = \{j = 1, 2, 3, \dots, n \text{ dan } J \text{ merupakan himpunan kriteria keuntungan}\}$

$J' = \{j = 1, 2, 3, \dots, n \text{ dan } J' \text{ merupakan himpunan kriteria biaya}\}$

Dimana v_{ij} adalah elemen dari matriks keputusan yang ternormalisasi terbobot V.

v_j^+ ($j = 1, 2, 3, \dots, n$) adalah elemen matriks solusi ideal positif.

v_j^- ($j = 1, 2, 3, \dots, n$) adalah elemen matriks solusi ideal negatif.

5. Menghitung seperasi

a. S^+ adalah jarak alternatif dari solusi ideal positif didefinisikan sebagai:

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad \text{Rumus 2. 5 Menghitung jarak alternatif ke-i dari}$$

solusi ideal positif dengan $i = 1, 2, 3, \dots, m$

- b. S_i^- adalah jarak alternatif dari solusi ideal negatif didefinisikan sebagai:

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad \text{Rumus 2. 6 Menghitung jarak alternatif ke-i dari}$$

solusi ideal negatif dengan $i = 1, 2, 3, \dots, m$

6. Menghitung kedekatan relatif terhadap solusi ideal positif

Kedekatan relatif dari setiap alternatif terhadap solusi ideal positif dapat dihitung dengan persamaan berikut:

$$C_i^+ = \frac{s_i^-}{(s_i^- + s_i^+)}, 0 \leq c_i^+ \leq 1 \quad \text{Rumus 2. 7 Menghitung kedekatan relatif terhadap solusi ideal positif}$$

dengan $i = 1, 2, 3, \dots, m$ dimana C_i^+ adalah kedekatan relatif dari alternatif ke-i terhadap solusi ideal positif.

7. Menentukan nilai preferensi setiap alternatif

Alternatif diurutkan dari nilai C^+ terbesar ke nilai terkecil. Alternatif dengan nilai C^+ terbesar merupakan solusi yang terbaik.

2.2.4 Konsep *Low Code*

Low Code atau kode rendah adalah pendekatan pengembangan perangkat lunak yang memungkinkan pembuatan aplikasi lebih cepat dengan sedikitnya pengkodean. Platform kode rendah adalah kumpulan alat yang memungkinkan pengembangan visual aplikasi melalui pemodelan dan antarmuka grafis. Kode

rendah memungkinkan pengembang untuk mempercepat proses penyelesaian sebuah aplikasi meski tanpa memakai pengkodean.

Platform kode rendah adalah rangkaian alat yang memungkinkan pengembangan dan pengiriman aplikasi lengkap secara visual. Antarmuka *drag-and-drop* adalah inti dari platform kode rendah. Tanpa menulis ribuan baris kode dan sintaks yang kompleks, kode rendah digunakan dalam membangun aplikasi lengkap dengan antarmuka pengguna modern, integrasi, data, dan logika dengan cepat dan visual.

Fitur-fitur merupakan alat bersaing untuk membedakan suatu produk perusahaan dengan pesaingnya (Arnomo & Hendra, 2019). Salah satu fitur unggulan serta keuntungan dari kode rendah adalah kecepatan pengembangan perangkat lunak lebih cepat sepuluh kali lipat dari metode tradisional dan memiliki kemudahan dalam pembuatan sebuah aplikasi *web* maupun *mobile* karena tidak mengharuskan penguasaan kode program. Salah satu Platform kode rendah yang digunakan saat ini adalah Outsystem versi 11.14.1.