BAB II

KAJIAN TEORITIS

2.1 Pendahuluan

Pada bab sebelumnya dijelaskan rumusan masalah pada penelitian ini adalah Bagaimana pengaruh algoritma *Particle Swarm Optimization* dalam meningkatkan akurasi metode *Fuzzy Time Series* untuk melakukan peramalan jumlah penduduk. Untuk memahami fundamental objek penelitian, maka dilakukan *literature review* terhadap jurnal, buku, dan artikel yang terkait dengan metode *Fuzzy Time Series* dan algoritma *Particle Swarm Optimization*.

2.2 Peramalan

Peramalan merupakan suatu proses pendugaan terhadap kejadian yang akan terjadi pada masa depan. Peramalan juga dapat diartikan sebagai proses yang dilakukan ketika ada kesenjangan waktu (lag) dari data aktual pada waktu tertentu dengan data yang ingin diketahui pada waktu yang akan datang. Peramalan diperlukan untuk mengetahui kapan atau bagaimana suatu peristiwa akan terjadi sehingga tidakan yang tepat dapat dilakukan. Peramalan identik dengan analisis data time series (deret waktu). Data *time series* merupakan serangkaian data yang berupa nilai pengamatan yang diukur selama kurun waktu tertentu berdasarkan interval waktu yang tetap (Susetyoko.R, 2016). Metode peramalan data

yang menggunakan prinsip-prinsip *fuzzy* sebagai dasarnya peramalan dengan menggunakan *fuzzy time series* menangkap pola dari data yang telah lalu kemudian digunakan untuk memproyeksikan data yang akan datang. Oleh karena itu perlunya untuk melakukan, peramalan jumlah penduduk sehingga dapat mengontrol pertumbuhan penduduk dan kesejahteraan penduduk secara berkala, berdasarkan hasil peramalan yang dilakukan.

Peramalan sering dilakukan untuk memprediksi dan membuat perencanaan dengan menggunakan data masa lalu dan data masa sekarang, agar dapat membuat prediksi di masa yang akan datang. Berdasarkan jangka waktunya, peramalan dapat dibagi menjadi tiga yaitu:

(1) Peramalan Jangka Panjang *(Long-Term Forecasting)*, yang memprediksikan keadaan dalam jangka waktu beberapa tahun ke depan (tahunan).

(2) Peramalan Jangka Menengah *(Mid-Term Forecasting)*, yang memprediksikan keadaan dalam jangka waktu bulanan atau mingguan.

(3) Peramalan Jangka Pendek *(Short-Term Forecasting)*, yang memprediksikan keadaan dalam jangka waktu harian hingga tiap jam.

2.1 Data Runtun Waktu (*Time Series*)

Data time series adalah data dari waktu ke waktu yang memiliki nilai tertentu untuk setiap waktunya. Data time series berisi data tentang objek

tertentu (Rizky P, 2018). Dapat dilihat dari contoh data *time series* pada data harga saham, data ekspor, data nilai tukar (*kurs*), data produksi, dan lain-lain sebagainya. Jika diamati masing-masing data tersebut terkait dengan waktu (*time*) dan terjadi berurutan. Misalnya data produksi minyak sawit dari tahun 2000 hingga 2009, data kurs Rupiah terhadap dollar Amerika Serikat dari tahun 2000 - 2006, dan lain-lain. Dengan demikian maka akan sangat mudah untuk mengenali jenis data ini. Data *time series* juga sangat berguna bagi pengambil keputusan untuk memperkirakan kejadian di masa yang akan datang. Karena diyakini pola perubahan data *time series* beberapa periode masa lampau akan kembali terulang pada masa kini.

2.3 *Fuzzy Time Series*

Fuzzy Time Series adalah metode prediksi data yang menggunakan prinsip- prinsip *fuzzy* sebagai dasarnya. Sistem prediksi dengan *FTS* menangkap pola dari data yang telah lalu kemudian digunakan untuk memproyeksikan data yang akan datang (Anwary, 2011).Pertama kali dikembangkan oleh Song and Chissom (1993) kemudian disempurnakan oleh Chen dan memiliki keuntungan mengurangi waktu perhitungan dan menyederhanakan proses perhitungan (Chen, 2004). *Fuzzy Time Series* adalah sebuah konsep baru yang dapat digunakan untuk menangani masalah prediksi di mana data historis adalah nilai-nilai linguistik (Chen, 1996). Metode ini sering digunakan oleh para peneliti untuk

menyelesaikan masalah prediksi. Dalam FTS, himpunan semesta didefinisikan sesuai Persamaan 1 (Chenn, 1996)

$$U=(Dmin-D_1 Dmax+D_2) \quad \text{(II-1)}$$

Keterangan:

$Dmin$: data historis minimum

$Dmax$: data historis maksimum

$D1$ dan $D2$: bilangan positif sembarang yang ditentukan oleh peneliti untuk menentukan himpunan semesta dari himpunan data historis

Adapun metode *Fuzzy Time Series* dalam penyelesaian masalah prediksi adalah sebagai berikut (Chen, 2004):

1. Mendefinisikan himpunan *fuzzy* pada himpunan semesta. Tahap ini mengubah himpunan semesta yang telah terbagi dan masih berupa himpunan bilangan crisp menjadi himpunan *fuzzy* berdasarkan interval.

$$A_1 = a_{11}/\ U_1 + a_{12}/\ U_2 + \cdots + + a_{1n}/\ U_n$$

$$A_2 = a_{21}/\ U_1 + a_{22}/\ U_2 + \cdots + + a_{2n}/ \quad \text{(II-2)}$$

$$UA_m = a_{m1}/\ U + a_{m2}/\ U_2 + \cdots + + a_{mn}/\ U_n$$

   Melakukan fuzzifikasi pada data historis. Tahap ini menentukan nilai keanggotaan pada masing-masing himpunan *fuzzy* dari data historis, dengan nilai keanggotaan 0 sampai 1. Nilai keanggotaan ini diperoleh dari fungsi keanggotaan yg telah dibuat sebelumnya.

2. Membentuk *fuzzy relationship*, $A_i \rightarrow A_j$ berdasarkan nilai A_i yang

telah ditentukan pada langkah sebelumnya, dimana A_i adalah hari ke n dan A_j adalah hari ke n + 1. Kemudian semua *fuzzy* relasinya dikelompokkan. Contohnya sebagai berikut :

A_i mempunyai relasi relasi, yaitu $A_i \rightarrow A_j$; $A_i \rightarrow A_{j1}$; $A_i \rightarrow A_j$. Dari 3 *fuzzy*

relationship dapat dikelompokkan menjadi $A_i \rightarrow A_j, A_{j1}$.

3. Melakukan defuzzifikasi output yang diramalkan. Tahap ini menentukan nilaihasil prediksi yang berupa nilai crisp, dengan aturan sebagai berikut:

   Misalkan (t) adalah data yang akan diramalkan dimana $F(t-1) = A_i$,maka:

a. Jika hanya terdapat satu *fuzzy relationship group* dari A_i A_i yaitu $A_i \rightarrow A_s$, maka $(t) = A_s$ dimana defuzzifikasinya adalah nilai tengah dari interval dimana memiliki nilai keanggotaan maksimum pada A_s.

b. Jika A_i tidak memiliki reasi maka defuzzifikasi $F(t)$ diperoleh dari nilai tengah interval yang memiliki nilai keanggotaan maksimum pada A_i

c. jika hanya terdapat lebih dari satu *fuzzy relationship group* dari A_i yaitu

   $A_i \rightarrow A_j, A_{j1}, A_{j2}, \ldots, A_{jn}$, maka $F(t)$ diperoleh dati rata rata nilai tengah dari masing masing interval yang memiliki nilai keanggotaan maksimum pada masing masing $A_j, A_{j1}, A_{j2}, \ldots, A_{jn}$.

2.2 Penentuan Interval Berbasis Rata-rata pada *Fuzzy Time Series*

Dalam perhitungan peramalan dengan menggunakan *fuzzy time series* standar, panjang interval telah ditentukan di awal proses perhitungan. Sedangkan penentuan panjang interval sangat berpengaruh dalam pembentukan *fuzzy relationship* yang tentunya akan memberikan dampak perbedaan hasil perhitungan peramalan. Oleh karena itu, pembentukan *fuzzy relationship* haruslah tepat dan hal ini mengharuskan penentuan panjang interval yang sesuai. Kunci utama dalam penentuan panjang interval adalah tidak boleh terlalu besar dan tidak boleh terlalu kecil, karena jika interval itu terlalu besar maka tidak akan terjadi fluktuasi dalam proses perhitungan *fuzzy time series*, demikian juga jika interval tersebut terlalu kecil maka makna dari *fuzzy time series* sendiri akan hilang (karena himpunan yang terbentuk cenderung ke himpunan tegas/*crisp*).

Salah satu metode untuk penentuan panjang interval yang efektif adalah dengan metode berbasis rata-rata (*average-based*), yang memiliki algoritma sebagaimana berikut :

1. Hitung semua nilai absolute selisih antara Ai+1 dan Ai (i=1…, n-1)sehingga diperoleh rata-rata nilai absolute selisih;
2. Tentukan setengah dari rata-rata yang diperoleh dari langkah pertamauntuk kemudian dijadikan sebagai panjang interval.
3. Berdasarkan panjang interval yang diperoleh dari langkah kedua,

ditentukanbasis dari panjang interval sesuai dengan tabulasi basis berikut.

Tabel II- 1 Tabel Basis Interval

Jangkauan	**Basis**
0.1 – 1.0	0.1
1.1 – 10	1
11 – 100	10
101 – 1000	100
1001-10000	1000

4. Panjang interval kemudian dibulatkan sesuai dengan tabel basis interval.

2.3 Particle Swarm Optimization

Particle Swarm Optimization disingkat sebagai *PSO*, merupakan algoritma berbasis populasi yang mengeksploitasi individu dalam pencarian. Dalam *PSO* populasi disebut swarm dan individu disebut partikel. Tiap partikel berpindah dengan kecepatan yang diadaptasi dari daerah pencarian dan menyimpannya sebagai posisi terbaik yang pernah dicapai. *PSO* didasarkan pada perilaku sosial sekawanan burung atau sekumpulan ikan. Algoritma *PSO* meniru perilaku sosial organisme ini. Perilaku sosial terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Kata partikel menunjukkan,

misalnya, seekor burung dalam kawanan burung. Setiap individu atau partikel berperilaku secara terdistribusi dengan cara menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi perilaku kelompok kolektifnya. Dengan demikian, jika satu partikel atau seekor burung menemukan jalan yang tepat atau pendek menuju ke sumber makanan, sisa kelompok yang lain juga akan dapat segera mengikuti jalan meskipun lokasi mereka jauh di kelompok tersebut.

Metode optimasi yang didasarkan pada *swarm intelligence* ini disebut algoritma *behaviorully inspired* sebagai alternatif dari algoritma genetika, yang sering disebut *evolution-based procedures*. Algoritma *PSO* ini awalnya diusulkan oleh J. Kennedy dan R. C. Eberhart (1995). Dalam konteks optimasi multivariabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel bergerak dalam ruang / *space* tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatanmasing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut.

PSO memiliki banyak kesamaan dengan teknik-teknik evolutionary computation yang lain, seperti *Genetic Algorithms (GA), Evolutionary Strategies (EA), Evolutionary Programming,* dan

sebagainya. *PSO* maupun GA dimulaidengan suatu populasi yang terdiri dari sejumlah individu (yang menyatakan solusi) yang dibangkitkan secara acak dan selanjutnya melakukan pencarian solusioptimum melalui perbaikan individu untuk sejumlah generasi tertentu. Tetapi berbeda dengan *GA*, *PSO* tidak menggunakan operator-operator evolusi seperti rekombinasi (*cross over*) dan mutasi. *PSO* memiliki memori untuk menyimpan solusi terbaik, sedangkan *GA* tidak punya. Setiap partikel pada *PSO* tidak pernah mati, sedangkan individu pada *GA* bisa mati dan digantikan dengan individu baru. Pada *PSO*, posisi dan kecepatan terbang partikel di-update pada setiap iterasi sehingga partikel tersebut bisa menghasilkan solusi baru yang lebih baik menurut (Ifrans, 2013).

Pada algoritma *PSO* ini, pencarian solusi dilakukan oleh suatu populasi yang terdiri dari beberapa partikel. Populasi dibangkitkan secara random dengan batasan nilai terkecil dan terbesar. Setiap partikel merepresentasikan posisi atau solusi dari permasalahan yang dihadapi. Setiap partikel melakukan pencarian solusi yang optimal dengan melintasi ruang pencarian (*search space*). Hal ini dilakukan dengan cara setiap partikel melakukan penyesuaian terhadap posisi terbaik dari partikel tersebut (*local best*) dan penyesuaian terhadap posisi partikel terbaik dari seluruh kawanan (*global best*) selama melintasi ruang pencarian. Jadi, penyebaran pengalaman atau informasi terjadi di dalam partikel itu sendiri dan antara suatu partikel dengan partikel terbaik dari seluruh kawanan selama proses pencarian solusi. Setelah

itu, dilakukan proses pencarian untuk mencari posisi terbaik setiap partikel dalam sejumlah iterasi tertentu sampai didapatkan posisi yang relatif steady atau mencapai batas iterasi yang telah ditetapkan. Pada setiap iterasi, setiap solusi yang direpresentasikan oleh posisi partikel, dievaluasi performansinya dengan cara memasukkan solusi tersebut kedalam *fitness function*.

Setiap partikel diperlakukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel (Kennedy and Eberhart, 1995).

Algoritma PSO terdiri dari tiga tahap, yaitu pembangkitan posisi serta kecepatan partikel, update velocity (*update* kecepatan), update position (*update*
posisi). Pertama posisi x_k^i dan kecepatan v_k^i dari kumpulan partikel

dibangkitkan secara random menggunakan batas atas (x_{max}) dan batas bawah (x_{min}) dari designvariable, seperti yang ditunjukkan pada persamaan (II-3) dan (II-4).

$$x^1 = x_{min} + r_1(x_{max} - x_{min}) \quad \text{(II-3)}$$

$$v^1 = x_{min} + r_2(x_{max} - x_{min}) \quad \text{(II-4)}$$

Dimana,

x^1 = posisi awal

v^1 = kecepatan awal

x_{min} = batas bawah

x_{max} = batas atas

r_1,r_2 = nilai random antara nilai 0 dan 1

Posisi dan kecepatan direpresentasikan dalam bentuk vektor di mana *n* dimensi vektor merepresentasikan jumlah dari desain variabel partikel, dengan *superscript* dan *subscript* menotasikan partikel ke *i* pada waktu ke *k* dengan proses inisialisasi ini maka kumpulan partikel dapat terdistribusi secara *random*
pada *design space*. Vektor seperti ditunjukkan di bawah ini:

$$x^i = (x^{i1}, x^{i2}, \dots, x^{in})^{\mathrm{T}} \qquad \text{(II-5)}$$

$$v^i = (v^{i1}, v^{i2}, \dots, v^{in})^{\mathrm{T}} \qquad \text{(II-6)}$$

Langkah kedua adalah *update velocity* (kecepatan) untuk semua partikel pada waktu $k + 1$ menggunakan fungsi objektif atau nilai *fitness* posisi partikel saat ini pada *design space* saat waktu ke *k*. Dari nilai *fitness* dapat ditentukan partikel mana yang memiliki nilai *global* terbaik (*global best*) pada *swarm* saat ini (p^g), dan juga dapat ditentukan posisi terbaik dari tiap partikel pada semua waktu yang sekarang dan sebelumnya (p^i). Perumusan *update velocity* menggunakan dua informasi tersebut untuk semua partikel pada kumpulan dengan pengaruh perpindahan yang sekarang (v^i), untuk memberikan arah pencarian (v^i) untuk $kk + 1$ generasi selanjutnya.

Perumusan *update velocity* mencakup beberapa parameter *random*

(*rand/r1,r2*), untuk mendapatkan cakupan yang baik pada *design space*, tiga parameter yang mempengaruhi arah pencarian, yaitu *self confidence* (c1), *swarmnconfidence* (c^2) akan digabungkan dalam satu penyajian, seperti yang ditunjukkan persamaan berikut:

$$v^i_{k+1} = {}^{i} + c^* r^*_1 {}_1(p^i - x^i) + c^* r^*_2 {}_2(p^g - x^i) \qquad \text{(II-7)}$$

Di mana,

v^i = kecepatan sekarang

x^i = posisi sekarang

c_1, c_2 = *self confidence*, *swarm confidence*, merupakan *learning rates* untuk kemampuan individu (*cognitive*) dan pengaruh sosial (*group*). Parameters c_1 dan c_2 menunjukkan bobot dari memori (*position*) sebuah partikel terhadap memori (*position*) dari kelompok (*swarm*). Nilai dari c_1 dan c_2 biasanya adalah 2 sehingga perkalian c_1r_1 dan c_2r_2 memastikan bahwa partikel-partikel akan mendekati target sekitar setengah selisihnya

r_1, r_2 = bilangan *random* yang memiliki interval 0 dan 1

p^i = *local best*, posisi terbaik dari tiap partikel pada semua waktu yang sekarang

p^g = nilai *global* terbaik (*global best*) pada *swarm* saat ini

k

Langkah terakhir dari setiap iterasi adalah *update* posisi tiap partikel dengan vektor *velocity*, seperti yang ditunjukkan pada persamaan berikut :

$$x^i_{k+1} = x^i_k + v^i_{k+1} \quad \text{(II-8)}$$

Dimana,

x^i_{k+1} = posisi pencarian

v^i_{k+1} = arah pencarian

x^i_k = posisi sekarang

Tiga tahapan di atas akan diulang sampai krtiteria kekonvergenan terpenuhi, kriteria kekonvergenan sangat penting dalam menghindari penambahanfungsi evaluasi setelah solusi optimum didapatkan, namun kriteria kekonvergenantidak selalu mutlak diperlukan, penetapan jumlah iterasi maksimal juga dapat digunakan sebagai *stopping condition* dari algoritma

Banyak cara untuk membangun kondisi berhenti, diantaranya adalah : iterasi dihentikan ketika PSO telah mencapai iterasi maksimum, atau PSO telah menemukan nilai optimum tertentu atau kesalahan minimum yang diinginkan.

2.4 Pengukuran Peramalan

Teknik peramalan tidak selamanya selalu tepat karena teknik peramalan yang digunakan belum tentu sesuai dengan sifat datanya atau disebabkan oleh kondisi di luar bisnis yang mengharuskan bisnis perlu menyesuaikan diri. Oleh karena itu, perlu diadakan pengawasan peramalan sehingga dapat diketahui sesuai atau tidaknya teknik peramalan yang digunakan. Sehingga dapat dipilih dan ditentukan teknik peramalan yang lebih sesuai dengan cara menentukan batas toleransi

peramalan atas penyimpangan yang terjadi (Jumiriani,2009).

Pada prinsipnya, pengawasan peramalan dilakukan dengan membandingkan hasil peramalan dengan kenyataan yang terjadi. Penggunaan teknik peramalan yang menghasilkan penyimpangan terkecil adalah teknik peramalan yang paling sesuai untuk digunakan (Jumiriani,2009).

Jilani, Burney, dan Ardil (2007) menggunakan metode AFER(*Average Forecasting Error Rate*) dan MSE (*Mean Square Error*) untuk mengetahui besarnya penyimpangan yang terjadi pada data hasil peramalan terhadap data riil.

a. RMSE (*Root Mean Square Error*)

Root Mean Squared Error (RMSE) adalah besarnya tingkat kesalahan hasil prediksi, dimana semakin kecil (mendekati 0) maka hasil RMSE semakin akurat. Adapun rumus perhitungan RMSE dapat dilihat pada II-9.

$$\text{RMSE} = \left(\frac{\sum(y_{i-}\hat{y})}{n}\right)^{1/2} \qquad \text{(II-9)}$$

Keterangan:

y = nilai hasil observasi

ŷ = nilai hasil prediksi

n = jumlah data

i = urutan data pada database

b. AFER (*Average Forecasting Error Rate*)

AFER (*Average Forecasting Error Rate*) adalah metode lain untuk mengevaluasi peramalan berdasarkan rata-rata kesalahan peramalan. Adapun

perhitungan AFER dapat dilihat pada 2.10.

$$\text{AFER} = \frac{\sum| \ (actual(t) - forecast(t))| \ / actual(t)}{n} \times 100\% \qquad \text{(II-10)}$$

Keterangan :

At = nilai hasil observasi/ data actual

Ft = nilai hasil peramalan data ke-i

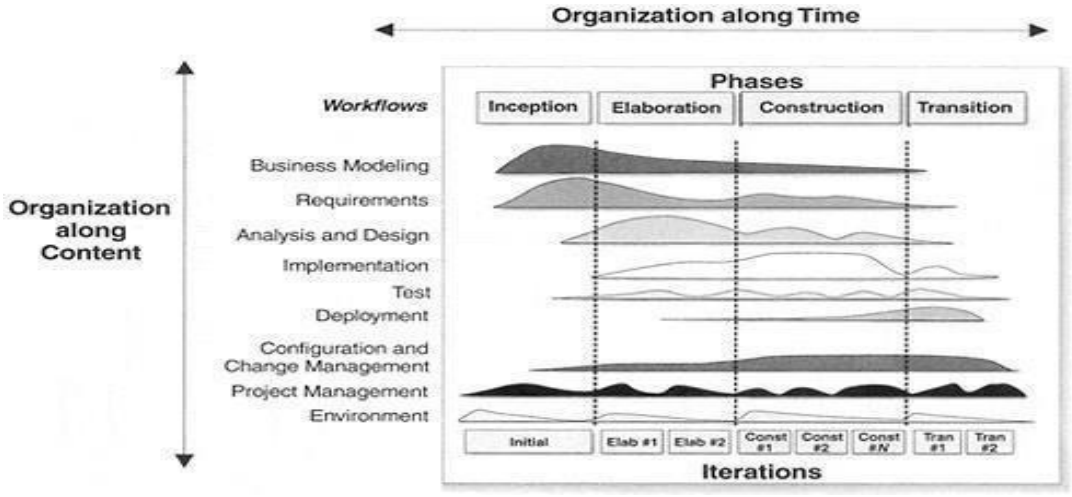
n = banyaknya data

2.5 Rational Unified Process (RUP)

Pada penelitian ini menggunakan metode pengembangan perangkat lunak *Rational Unified Process* (RUP). RUP merupakan suatu pendekatan yang mengatur tugas-tugas serta tanggung jawab pada organisasi pengembang perangkat lunak (Krutchen, 2004). RUP menggunakan konsep berorientasi objek, dengan aktifitas yang berfokus pada pengembangan model menggunakan *Unified Model Language* (UML).

1. Struktur Proses RUP

Arsitektur RUP memiliki dua dimensi herizontal dan vertikal yang dapat dilihat pada Gambar Gambar II-7 dibawah ini.



Gambar II-1 Arsitektur RUP (Sumber: Kruchten,2004)

Dilihat pada Gambar II-1 arsitektur RUP menunjukkan proses RUP yangmemiliki dua dimensi tahapan, yaitu:

a. Dimensi Horizontal

Dimensi ini menggambarkan waktu dan menjelaskan aspek dinamis dari proses pengembangan perangkat lunak meliputi; siklus, fase, iterasi, dan *milestones (*batas pencapaian yang menandakan akhir dan awal dari fase selanjutnya). Dimensi horizontal ini terdiri atas fase *Inception*, *Elaboration*, *Construction*, dan *Transition*.

b. Dimensi Vertikal

Dimensi ini menjelaskan aspek-aspek statis dari proses pengembangan perangkat lunak (RUP) meliputi; aktivitas dan alur pekerjaan. Alur kerja pada dimensi vertikal ini terdiri atas; *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration, Project Management,* dan *nvironment.*

2. Fase-Fase Pengembangan RUP

a. Fase Insepsi (*Inception*) merupakan tahapan pertama dari 4 tahapan RUP. Fase ini berfokus pada permodelan bisnis. Dalam tahap ini memahami tentang semua proyek dan ruang lingkup tujuan. Permodelan bisnis tersebut mencakup antara lain; mendefinisikan masalah, membatasi ruang lingkup proyek, dan *initial bussiness case*.

b. Fase Elaborasi *(Elaboration)* merupakan tahapan kedua RUP. Tujuannya adalah untuk menganalisis permasalahan, menentukan arsitektur dasar dari sistem untuk menyediakan dasar yang stabil sebagian besar dari desain dan pelaksanaan upaya dalam tahap konstruksi.

c. Fase Konstruksi (*Construction*), merupakan tahapan ketiga RUP.fase ketiga yang merupakan tahap untuk mengimplementasikan hasil desain ke *coding* dan melakukan pengujian hasil implementasi. Di fase ini semua komponen dan fitur aplikasi dibangun dan disatukan. Fase ini berupa sebuah proses *manufacturing* yang berfokus pada pengontrolan operasi untuk mengoptimalkan biaya, jadwal dan kualitas.

d. Fase Transisi *(Transition),* merupakan tahap keempat sekaligus tahap terakhir tahapan pelaksanaan lifecycle RUP yaitu membuat apa yang sudah dimodelkan sebelumnya menjadi suatu produk jadi. Fase ini terfokus untuk memastikan bahwa produk layak untuk pengguna

akhir untuk itu penulis menentukan *tools* pengujian yang diperlukan yaitu perangkat keras yang sama saat digunakan untuk pengembangan perangkat lunak.

2.6 Penelitian Lain Yang Relevan

1. Penelitian sebelumnya juga mengenai Tentang pentuan *base trancivier system* menggunakan *Partical Swarm Optimization (PSO),* Salah satu jaringannya adalah jaringan 4GLTE. Jaringan 4G ini merupakan suatu sistem berbasis IP dengan kecepatan 100Mbps/detik dan 1Gb/detik. Teknologi pada jaringan 4G menggunakan menara untuk memproses transmini yang disebut eNode B (Sri, 2014) Frekuensi jaringan 4G yang ada di kota Malang saat ini 2300 MHz dan jangkauannya masih kurang memenuhi kebutuhan masyarakat (Sri, 2014). Maka penentuan lokasi BTS sangatlah penting. Sehingga diperlukan suatu metode dalam penentuan lokasi BTS yang akan dipasang. Satu Base Tranceiver System (BTS) akan melayani sebuah sel, dimana satu Base Tranceiver System akan mengatur sebuah BCCH (Broadcast Control Channel) dengan jumlah kanal pembawa maksimum 8 kanal. Setiap transceiver mentransmisikan dengan daya yang sama. Fungsi internal dari Base Tranceiver System (BTS) adalah sebagai protokol dari jalur sinyal radio, jalur sinyal informasi antara BSC dan MS, maupunprotocol interface BSC (Ningsih dkk, 2015).

2. Penelitian tentang peramalan penduduk sebelumnya sudah dilakukan dengan *Optimasi Fuzzy Time Series Menggunakan Algoritma ParticleSwarm Optimization Untuk Peramalan Jumlah Penduduk Di Kabupaten Probolinggo* oleh (Cahyo Adi Prasojo, Budi Darma Setiawan, Marji 2018). Hasil penelitian menunjukan metode *PSO*berhasil di implementasikan pada backpropagation untuk mengoptimalkan bobot. *PSO* Metode *Fuzzy Time Series* (FTS) dapatdigunakan untuk melakukan peramalan. peramalan yang dilakukanmenggunakan metode *FTS* memeberikan nilai MSE, yaitu: 52,05198333. Metode *Particle Swarm Optimization (PSO)* dapat digunakan untuk melakukan optimasi metode *FTS.* Langkah pertamadengan melakukan optimasi interval *FTS* dengan *PSO*. Interval pada *FTS* tersebut akan digunakan sebagai partikel pada *PSO* yang nantinyamemiliki fitness yang tinggi sebagai penentu partikel yang optimal, dimana semakin tinggi fitness semakin baik nilai akurasinya. Nilai interval dilakukan dengan mengambil nilainya secara acak denganbatasan tertentu. Hasil dari pengujian yangdilakukan dengan menggunakan data jumlah penduduk Kabupaten Probolinggo pada setiap kecamatan mulai tahun 2013 sampai 2016, didapatkan nilaiparamater terbaik yaitu: w = 0,6, $c1$= 1,8 dan $c2$= 2,4. Sehingga diperoleh nilai fitness terbaik dari peramalan tersebut, yaitu: 0,445334. Pada penelitian jumlah penduduk dengan menggunakan metode *Fuzzy Time Series* yang

diptimasi dengan metode *Particle Swarm Optimization* masih terdapat banyak kekurangan. Kekurangan tersebut dapat dikembangkan dengan melakukan penelitian yang lebih baik lagi.

3. Penelitian terkait dengan *Fuzzy Time Series* dan *Particle Swarm Optimization* dilakukan oleh Qiu et al., (2015) yang melakukan penelitian pada data shanghai stock exchange composite index dan data pendaftaran University of Alabama. Penelitian ini menerapkan metode *Generalized Fuzzy Time Series Forecasting Model Enhanced* dengan *Particle Swarm Optimization*. Dalam percobaan menunjukkan bahwa hasil *Root Mean Squared Error* (*RMSE*) sebesar 2.59, *Mean Absolute Error* (*MAE*) sebesar 0,64 serta *Mean Absolute Percentage Error* (*MAPE*) sebesar 0.0004
4. Selanjutnya penelitian pada metode *Fuzzy Time Series* dan *Particle Swarm Optimization* yang dilakukan oleh Dwi, Rifandi, & Setiawan (2018) dengan penelitiannya Peramalan Permintaan Darah. Berdasarkan hasil dari serangkaian pengujian didapatkan solusi optimum bernilai *cost* (*MSE*) sebesar 60435.685 dengan jumlah partikel sebanyak 40, jumlah dimensi sebanyak 30, kombinasi nilai $c1$ dan $c2$ masing-masing 1.5 dan 1.5, bobot inersia sebesar 0.3, danumlah iterasi maksimum sebesar 950. Tingkat kesalahan dari sistemini (*MAPE*) sebesar 7.50330% dari 12 data uji yang digunakan.

2.7 Kesimpulan

Pada bab ini telah dibahas dasar – dasar teori terkait yang dibutuhkan dalam melakukan penelitian ini. Selanjutnya pada bab III akan dibahas mengenai metodologi penelitian yang akan digunakan pada penelitian kali ini.