

Implementasi Metode *Artificial Neural Network (ANN) Model Based On Use Case Point* Dalam Menghitung Biaya Perangkat Lunak (Studi Kasus CV. Profile Image Studio)

Ade Pratama¹, Andi Reza Perdanakusuma², Djoko Pramono³

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹adepratamaa99@gmail.com, ²andireza@ub.ac.id, ³djoko.jalin@ub.ac.id

Abstrak

Estimasi biaya perangkat lunak merupakan aspek yang penting dalam proyek teknologi informasi untuk pembuatan anggaran. Dalam proses penentuan harga suatu sistem CV. Profile Image Studio tidak menggunakan format standar ilmiah atau biasa disebut *parametric*, mereka hanya menggunakan metode *guesstimate*. Metode tersebut hanya berdasarkan perkiraan yang dilandaskan intuisi. Berdasarkan hal tersebut maka dalam penelitian ini akan mengimplementasikan metode metode Artificial Neural Network (ANN) model Based On Use Case Point untuk menghitung biaya perangkat lunak sebagai masukan kepada CV. Profile Image Studio. Estimasi biaya dapat diperoleh setelah mendapatkan nilai estimasi effort pengembangan sistem, sumber daya manusia serta estimasi waktu pengembangan sistem. Metode Artificial Neural Network (ANN) model Based On Use Case Point diimplementasikan kepada 2 sistem yang telah selesai dikembangkan oleh CV. Profile Image Studio. Pada akhir penelitian ini, dilakukan perbandingan biaya pengembangan sistem, yaitu dengan membandingkan hasil estimasi biaya yang diperoleh menggunakan metode Artificial Neural Network (ANN) model Based On Use Case Point dengan alokasi biaya yang dikeluarkan oleh CV. Profile Image Studio. Adapun hasil dari implementasi metode Artificial Neural Network (ANN) model Based On Use Case Point dalam menghitung biaya perangkat lunak didapatkan bahwa total estimasi biaya sistem DBA ticketing sebesar Rp 17.703.140 yang dikerjakan oleh 12 orang dalam waktu 1247,46 jam sedangkan untuk sistem pintu air sebesar Rp 6.436.860 yang dikerjakan oleh 12 orang dalam waktu 453,6 jam.

Kata kunci: *estimasi biaya, estimasi biaya perangkat lunak, use case point, jaringan saraf tiruan,*

Abstract

Software cost estimation is an important part of an information technology project for budgeting. In the process of determining the price of a system CV. The Profile Image Studio does not use the scientific standard format or is commonly called *parametric*, they only use the *guesstimate* method. It is only estimates based on intuition. Based on this case, the cost estimation in this study will implement Artificial Neural Network (ANN) model Based on Use Case Point method is implemented in calculating software costs as suggestion for CV. Profile Image Studio. Cost estimates can be obtained after obtaining estimated effort, number of human resources and estimated system development time. Artificial Neural Networks (ANN) Based on Use Case Point method is implemented in 2 systems that have been developed by CV. Studio Profile Image. At the end of this study, the cost comparison of system development was done by comparing estimated cost results obtained using the Artificial Neural Network (ANN) model based on the Use Case Point with the allocation of costs incurred by CV. Studio Profile Image. The results of applying the Artificial Neural Network (ANN) model Based on the Use Case Point method in calculating software costs, it was found that the total estimated cost of the DBA ticket system was Rp. 17,703,140 carried out by 12 people in 1247.46 hours while the pintu air system was Rp. 6,436,860 carried out by 12 people in 453.6 hours.

Keywords: *cost estimation, software cost estimation, use case point, artificial neural network*

1. PENDAHULUAN

Manajemen proyek Teknologi Informasi merupakan proses penerapan pengetahuan, keterampilan, alat, dan teknik untuk memproyeksikan kegiatan untuk dapat memenuhi kebutuhan dan harapan pemangku kepentingan dari sebuah proyek Teknologi Informasi (PMI, 2000). Yang mana 10 bidang pengetahuan tersebut adalah manajemen integrasi, ruang lingkup, waktu, biaya, kualitas, sumber daya manusia, komunikasi, manajemen risiko, pengadaan, dan pemangku kepentingan (Schwalbe, 2012).

Banyak perusahaan yang tidak mementingkan manajemen proyek dengan baik dan hal tersebut yang sering menjadi masalah dalam pengembangan suatu proyek IT (Maswinandar, 2016). Sebagai contohnya pada bulan November 2012 angkatan udara Amerika Serikat akhirnya menghentikan proyek ERP yang mereka kembangkan setelah biayanya melampaui US\$ 1 Milyar (Kanaracus, 2012). Menurut Kashyap *et al* (2014), *software cost estimation* bersifat penting untuk tawar-menawar, pembuatan anggaran, dan perencanaan. Sebuah perencanaan dalam mengalokasikan biaya dibutuhkan untuk bisa memproduksi perangkat lunak dalam cakupan waktu dan biaya yang terbatas.

Di CV. Profile Image Studio sendiri dalam melakukan estimasi biaya pembuatan perangkat lunak belum menggunakan metode dengan format standar ilmiah. Mereka memperkirakan estimasi biaya dengan melihat dari tingkat kompleksitas suatu perangkat lunak tersebut kemudian jumlah sumber daya yang diperlukan lalu mempertimbangkan dari kemampuan keuangan pelanggan atau disebut *guesstimate*. Dengan metode tersebut yang hanya dilandaskan perkiraan dapat menyebabkan hasil estimasi biaya menjadi tidak akurat. Berdasarkan hal tersebut maka dilakukan implementasi metode lain dalam menghitung biaya perangkat lunak sebagai masukan kepada CV. Profile Image Studio.

Metode *Artificial Neuron Network* (ANN) *Model Based on Use Case Points* (UCP) sendiri merupakan pengembangan dari metode *Use Case Point* yang diperkenalkan oleh Gustav Karner tahun 1993. Metode *Artificial Neuron Network* (ANN) *Model Based on Use Case*

Points (UCP) akan menghasilkan estimasi *effort*. Proses *implementasi* metode *Artificial Neuron Network* (ANN) *Model Based on Use Case Points* (UCP) akan membutuhkan tiga masukan yang akan digunakan untuk menghitung nilai *effort* atau usaha. Tiga masukan tersebut mencakup ukuran perangkat lunak, produktivitas dan kompleksitas proyek. Nilai ukuran perangkat lunak akan diperkirakan menggunakan nilai *Use Case Points* (UCP).

Metode *Artificial Neuron Network* (ANN) *Model Based on Use Case Points* (UCP) dipilih untuk digunakan dalam penelitian ini karena metode tersebut lebih baik dari dua metode berdasarkan penelitian yang dilakukan oleh Ali Bou Nassif dan Luiz Fernando Capretz yang berjudul "*Estimating Software Effort Using an ANN Model Based on Use Case Points*". Dari penelitian disimpulkan bahwa metode ANN mengungguli metode *Multiple Linier Regression* dan *Use Case Point* jika dilihat pada kriteria *Mean of Magnitude of Error Relative to the estimate* (MMER). Nilai MMER ANN adalah 0,49. Nilai tersebut lebih kecil jika dibandingkan dengan nilai MMER dari *Multiple linear regression* sebesar 0,57 dan *Use case Point* sebesar 0,99. Dengan nilai MMER yang lebih kecil maka tingkat error dalam mengestimasi lebih kecil jika menggunakan metode *Artificial Neuron Network* (ANN) *Model Based on Use Case Points* (UCP).

2. LANDASAN KEPUSTAKAAN

2.1. *Artificial Neural Network*

Artificial Neural Network (ANN) atau dalam bahasa Indonesia disebut jaringan saraf tiruan adalah jaringan yang terdiri dari neuron buatan atau kelenjar yang mengemulsi neuron biologis (A. B. Nassif, 2012). Topologi yang paling menonjol dari ANN adalah jaringan *feed-forward*. Jaringan *feed-forward* biasanya direpresentasikan sebagai *input*, *hidden*, dan *output layer*. Jika *hidden layer* tidak ada, maka jenis ANN ini disebut *perceptron*. Ada beberapa tipe jaringan *feed-forward* yang memiliki *hidden layer* didalamnya, seperti *Multilayer Perceptron* (MLP), *Radial Basis Function Neural Network* (RBFNN) dan *General Regression Neural Network* (GRNN). Di dalam MLP sendiri mengandung setidaknya satu *hidden layer* dan setiap input diwakili oleh sebuah neuron. Dalam penelitian oleh (A. B.

Nassif, 2012), jenis MLP digunakan untuk memprediksi *effort* atau usaha perangkat lunak berdasarkan ukuran perangkat lunak yang dihitung berdasarkan metode *use case point* (UCP), produktivitas tim, dan kompleksitas proyek. Gambar 1 menunjukkan bagaimana arsitektur ANN yang digunakan dalam penelitian (A. B. Nassif, 2012) dengan tiga input dan empat neuron tersembunyi.

2.2 Use Case point

Use case bisa digunakan untuk mengestimasi usaha pengembangan perangkat lunak dengan metode *Use Case Point* (Anda, 2002). Kompleksitas *use case* ditunjukkan dari nilai *Unadjusted Use Case Weight* (UUCW) yang diukur dari jumlah transaksi dalam *use case*. Bobot tiap *use case* dijumlah untuk mendapatkan nilai *Unadjusted Use Case Weight* (UUCW) menggunakan persamaan 2 sebagai berikut.

$$UUCW = \sum_{i=1}^3 x * Wi \tag{2}$$

Aktor pada setiap *use case* dikategorikan sederhana, rata-rata, atau kompleks (Clemmon, 2006). Aktor dengan kategori sederhana yaitu aktor mewakili sistem lain dengan API yang telah ditentukan. Aktor dengan kategori rata-rata adalah aktor mewakili sistem lain yang berinteraksi melalui protokol seperti TCP/IP. Aktor dengan kategori kompleks merupakan orang yang berinteraksi dengan sistem melalui *Graphical User Interface* (GUI). Untuk menghitung *Unadjusted Actor Weight* (UAW) dengan cara berapa banyak aktor dari masing-masing kategori kemudian dikalikan dengan bobot dari masing-masing kategori seperti pada persamaan 3 di bawah ini.

$$UAW = \sum_{i=1}^3 m * Ci \tag{3}$$

Setelah mendapatkan nilai dari *Unadjusted Use Case Weight* (UUCW) dan *Unadjusted Actor Weight* (UAW) maka nilai *Unadjusted Use Case Points* (UUCP) dapat dihitung dari penjumlahan *Unadjusted Use Case Weight* (UUCW) dan *Unadjusted Actor Weight* (UAW) seperti pada persamaan 4.

$$UUCP = UUCW + UAW \tag{4}$$

Nilai dari *Technical Complexity Factor* (TCF) dan *Environmental Complexity Factors*

(ECF) dari proyek harus dihitung juga (Nassif Ali, 2011) karena, merupakan salah satu faktor untuk memperkirakan ukuran perangkat lunak dengan memperhitungkan pertimbangan teknis. Untuk mendapatkan nilai dari *Technical Complexity Factor* (TCF) maka nilai faktor ditentukan mulai dari 0 hingga 5 kemudian dikalikan dengan bobot dari masing-masing faktor sesuai dengan persamaan 5.

$$TCF = 0.6 + 0.01 * (\sum_{i=1}^{13} Ti * Wi) \tag{5}$$

Nilai *Environmental Factor* (EF) ditentukan dengan menetapkan nilai antara 0 sampai 5 pada setiap faktor kemudian dikalikan dengan bobot dari masing-masing faktor sesuai dengan persamaan 6 kemudian nilai dari EF digunakan untuk menghitung nilai *Environmental Complexity Factor* (ECF) menggunakan persamaan 7.

$$EF = (\sum_{i=1}^8 Ei * Wi) \tag{6}$$

$$ECF = 1.4 + (-0.303 * EF) \tag{7}$$

Nilai *Adjusted Use Case Points* (UCP) didapatkan dengan mengalikan *Unadjusted Use Case Points* (UUCP) dengan *Technical Complexity Factors* (TCF) dan *Environmental Complexity Factors* (ECF) seperti pada persamaan 8.

$$UCP = UUCP * TCF * ECF \tag{8}$$

2.3 Metode Artificial Neuron Network Model Based On Use Case point

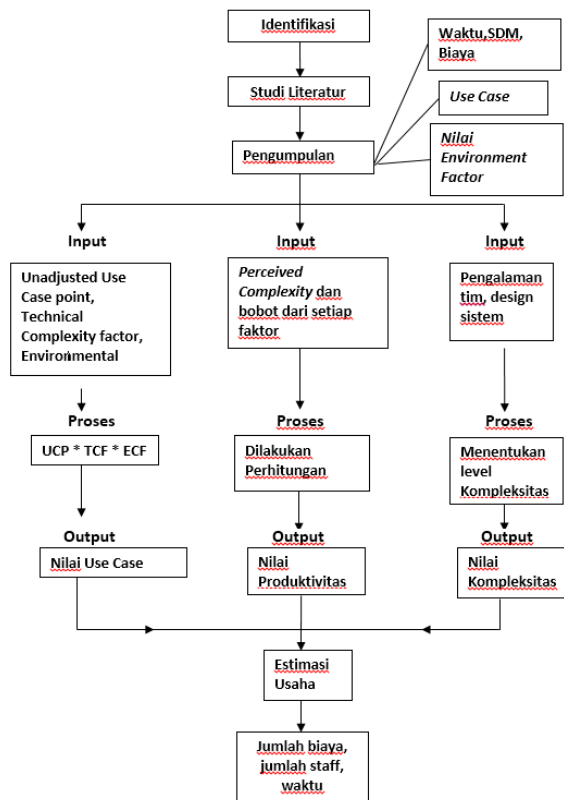
Metode *Artificial Neuron Network* (ANN) *Model Based on Use Case Points* (UCP) merupakan pengembangan dari metode *Use Case Point*. Untuk menghitung *effort* didapatkan dari tiga masukan yaitu ukuran perangkat lunak, produktivitas dan kompleksitas. Nilai ukuran perangkat lunak diperkirakan berdasarkan metode UCP, nilai produktivitas diperkirakan dengan menetapkan nilai skala antara 0 sampai 5 di setiap faktor pada *Environmental factor*, kemudian dikalikan dengan bobot dari setiap faktor. Nilai kompleksitas sendiri ditentukan berdasarkan tingkatan level pada metode ANN Model Based On use Case Point.

Untuk menghitung *effort* pada metode *Artificial Neuron Network Model Based On Use Case point* dengan menggunakan persamaan 1. Hasil *effort* yang didapatkan memiliki satuan *person-hours*.

$$Effort = 3.53 + (0.88 \times Size) - (0.009 \times Productivity) + (0.31 \times Complexity) \quad (1)$$

3. METODOLOGI

Metode penelitian yang dilakukan dimulai dengan mengidentifikasi masalah yang ada pada tempat penelitian, kemudian dilakukan studi pustaka untuk menentukan metode yang tepat. Tahapan berikutnya setelah masalah teridentifikasi dan menemukan metode yang tepat yaitu mengumpulkan data yang dibutuhkan untuk menentukan nilai estimasi *effort*, nilai estimasi biaya. Setelah itu dilakukan penjadwalan perencanaan waktu dan hasil estimasi biaya yang diperoleh akan dievaluasi.



Gambar 1. Diagram Alur Penelitian

3.1. Identifikasi Masalah

Berdasarkan hasil wawancara dengan direktur operasional CV. Profile Image Studio, diketahui bahwa CV. Profile Image Studio menggunakan metode *guesstimate* ketika melakukan penghitungan estimasi biaya pembuatan perangkat lunak, jadi belum memiliki metode parametric untuk menghitung estimasi biaya perangkat lunak yang mereka buat.

3.2. Studi Pustaka

Pada tahap studi pustaka akan dilakukan pengumpulan referensi dari *paper*, jurnal, maupun buku untuk menunjang penelitian ini.

3.3. Pengumpulan Data

Ketika pengumpulan data penulis melakukan pengumpulan data yang bersumber dari manajer proyek di CV. Profile Image Studio. Penulis melakukan pengumpulan data dengan cara observasi, wawancara dan membagikan lembar penilaian. Dari hasil pengumpulan data didapatkan data berupa waktu, sumber daya manusia yang digunakan, serta biaya yang dikeluarkan untuk pengembangan sistem dan juga nilai pada *environmental factor*. Data yang berhasil dikumpulkan kemudian akan digunakan untuk menghitung estimasi *effort* dengan metode *ANN model based on use case point*.

3.4. Menghitung Estimasi Effort

Data yang berhasil didapatkan pada tahap pengumpulan data lalu diolah untuk memperoleh hasil estimasi *effort* menggunakan metode *Artificial Neural Network (ANN) model based on use case point* yang akan digunakan untuk menghitung estimasi waktu, sumber daya manusia serta biaya pengembangan.

3.5. Estimasi Waktu

Estimasi waktu pengembangan sistem dihitung berdasarkan penelitian (Saleh, 2011). Hasil estimasi waktu didapatkan dengan dengan cara mengalikan *effort rate* dengan nilai use case point kemudian dibagi dengan jumlah jam kerja seperti persamaan 2.1. Hasil estimasi waktu yang didapat pada tahapan ini kemudian digunakan untuk menghitung estimasi biaya pengembangan sistem.

3.6. Estimasi Sumber Daya Manusia

Hasil dari tahapan ini akan mendapatkan jumlah estimasi sumber daya manusia yang diperlukan dalam pengembangan sistem. Jumlah estimasi sumber daya manusia didapatkan dari pembagian nilai *effort* pengembangan sistem dibagi dengan estimasi waktu pengembangan sistem yang didapatkan pada tahapan sebelumnya. Jumlah estimasi waktu dan estimasi sumber daya manusia akan digunakan untuk menghitung estimasi biaya.

3.7. Estimasi Biaya

Dari estimasi waktu pengembangan sitem dan estimasi sumber daya manusia yang telah didapatkan pada tahap sebelumnya Pada tahap ini, aktivitas pengembangan dibagi ke dalam 2 kelompok aktivitas yaitu *Software Phases* dan *Ongoing Life Cycle Activities*. Pada tahap ini, estimasi biaya diperoleh dengan mengalikan durasi pengerjaan setiap aktivitas dengan standar gaji UMK kota Malang pada tahun 2017. Biaya per aktivitas dijumlahkan untuk mendapatkan biaya total keseluruhan pengembangan sistem.

4. PENGUMPULAN DATA

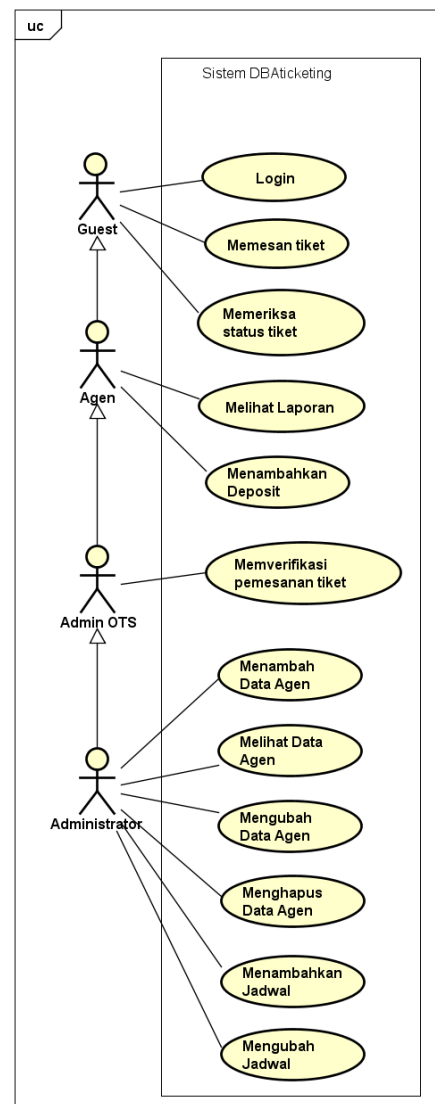
Bab ini menjelaskan data yang telah dikumpulkan adapun data tersebut yaitu berupa waktu pengembangan sistem, sumber daya manusia yang digunakan, dan biaya pengembangan milik perusahaan. Data yang lan berupa *use case diagram* dan *use case skenario* dari sistem DBA ticketing maupun sistem pintu air.

4.1. Waktu, Sumber Daya Manusia, Biaya

Waktu yang diperlukan untuk pengembangan sistem DBA ticketing berdasarkan data milik perusahaan adalah 4 bulan, sedangkan untuk sistem pintu air selama 4 bulan. Sumber daya manusia yang yang digunakan ketika mengembangkan sistem DBA ticketing adalah sebanyak 3 orang sedangkan sumber daya manusia yang digunakan ketika mengembangkan sistem pintu air sebanyak 2 orang. Biaya pengembangan sistem DBA ticketing adalah sebesar Rp 25.000.000 sedangkan biaya pengembangan sistem pintu air adalah sebesar Rp 35.000.000.

4.2. Use Case Diagram

Gambar 2 merupakan *use case diagram* sistem DBA *ticketing*. Sistem divisualisasikan ke dalam *use case diagram* sesuai dengan hasil wawancara terhadap manajer proyek di CV. Profile Image Studio. Langkah berikutnya adalah mengidentifikasi aktor untuk dieklompokkan berdasarkan karakteristik yang dimiliki, kemudian menentukan tujuan aktor saat menggunakan sistem, dan menspesifikasikan setiap *use case*.



Gambar 2. Use Case Diagram

Tabel 1 berisi tentang deskripsi tiap-tiap aktor yang ada pada use case diagram sistem DBA *ticketing* :

Tabel 1. Deskripsi Aktor

Nama Aktor	Deskripsi
Guest	Aktor yang diperankan oleh orang yang dapat mengakses halaman awal sistem DBA <i>ticketing</i> secara langsung. Aktor bisa menggunakan sistem dan mengakses informasi sesuai akses yang dimiliki.
Agen	Aktor yang diperankan oleh pihak yang bekerjasama dengan pihak pembuat film untuk membantu menjual tiket, yang akan menggunakan sistem untuk proses pemesanan tiket dan mengatur deposit.

Admin OTS	Aktor yang diperankan oleh pegawai yang ditunjuk pihak pembuat film yang mana dapat menggunakan fungsi di dalam sistem tersebut seperti pemesanan tiket dan mengatur deposit.
Adminstrator	Aktor yang diperankan oleh pegawai yang ditunjuk menjadi admin oleh pihak pembuat film yang mana dapat menggunakan seluruh fungsi di dalam sistem tersebut seperti mengatur laporan, mengatur deposit, mengatur data agen dan mengatur jadwal.

Pada use case skenario akan dijelaskan lebih rinci mengenai aktifitas di dalam sebuah *use case*. Use case skenario tersusun dari deskripsi mengenai use case tersebut kemudian bagaimana kondisi sebelum *use case* dijalankan, aktor dan kondisi setelah *use case* dijalankan, dan urutan langkah-langkah yang harus dilalui para aktor guna menyelesaikan sebuah use case.

Tabel 2. Use Case Skenario

Brief Description	<i>Use Case</i> ini menjelaskan bagaimana proses aktor pengguna ketika melakukan pemesanan tiket.
Actor	<i>Guest</i>
Precondition	<ul style="list-style-type: none"> Aktor telah berada di halaman awal.
Postcondition	Aktor telah berhasil mendapatkan kode pemesanan tiket.
Basic Flow	<ol style="list-style-type: none"> <i>Use case</i> dimulai ketika aktor memilih untuk memesan tiket Sistem menampilkan jadwal tayang. Aktor memilih jadwal tayang. Sistem menampilkan <i>form</i> pengisian identitas pemesan. Aktor mengisi identitas pada <i>form</i>. Sistem menampilkan form jumlah tiket yang akan dibeli. Aktor mengisi jumlah tiket yang akan dibeli. Sistem menampilkan pilihan nomor rekening yang akan digunakan untuk membayar.

	<ol style="list-style-type: none"> Aktor memilih nomor rekening. Sistem melakukan proses pemesanan tiket oleh aktor Sistem menampilkan pesan bahwa proses pemesanan telah berhasil. Sistem menampilkan kode pemesanan tiket. <i>Use case</i> selesai.
Alternative Flow	<p>AF1 : Menangani kegagalan dalam proses pemesanan tiket</p> <p>Jika sistem gagal dalam memproses pemesanan tiket maka sistem menampilkan pesan proses pemesanan tiket gagal dilakukan, dan <i>use case</i> selesai.</p>

5. PEMBAHASAN

5.1. Menghitung *Unadjusted Use Case Point*

Penjumlahan *Unadjusted Use Case Weight* (UUCW) dengan *Unadjusted Actor Weight* (UAW) akan menghasilkan nilai *Unadjusted Use Case Point* seperti pada persamaan 2. Diperlukan jumlah transaksi dalam setiap *use case* guna menghitung *Unadjusted Use Case Weight* (UUCW). Diperlukan jenis antarmuka yang digunakan aktor untuk berinteraksi diperlukan guna menghitung *Unadjusted Actor Weight* (UAW). Aktor akan dikategorikan berdasarkan jenis antarmuka yang digunakan saat berinteraksi dengan sistem.

Tabel 3. Deskripsi Aktor

No	Nama Aktor	Kategori Aktor	Bobot Aktor
1	Agen	Kompleks	3
2	Administrat or	Kompleks	3
3	Admin OTS	Kompleks	3
Total UAW			9

Menurut tabel 3, aktor agen, administrator dan admin OTS masuk kedalam kategori kompleks dan memiliki nilai bobot 3. Hal tersebut dikarenakan aktor agen, administrator dan admin OTS berinteraksi dengan sistem melalui *Graphical User Interface* (GUI). Total nilai *Unadjusted Actor Weight* (UAW) pada sistem DBA *ticketing* adalah 9.

5.2. Transaksi Use case

Transaksi dapat dihitung dengan cara menghitung jumlah langkah yang ada pada *basic flow* dan *alternative flow* yang ada didalam sebuah use case (Kristen Ribu, 2001). Terdapat tiga kategori yang berdasarkan jumlah dari transaksi yang terjadi dalam *use case*. Kategori tersebut yaitu sederhana, medium atau rata-rata, atau kompleks. Berikut merupakan transaksi yang ada pada sistem DBA *ticketing*.

Tabel 4. Transaksi Use Case

Basic flow :	
1.	Use case dimulai ketika aktor pengguna memilih untuk memesan tiket
2.	Sistem menampilkan jadwal tayang.
3.	Aktor memilih jadwal tayang.
4.	Sistem menampilkan form pengisian identitas pemesan.
5.	Aktor mengisi identitas pada form.
6.	Sistem menampilkan form jumlah tiket yang akan dibeli.
7.	Aktor mengisi jumlah tiket yang akan dibeli.
8.	Sistem menampilkan pilihan nomor rekening yang akan digunakan untuk membayar.
9.	Aktor memilih nomor rekening.
10.	Sistem melakukan proses pemesanan tiket oleh actor
11.	Sistem menampilkan pesan bahwa prose pemesanan telah berhasil.
12.	Sistem menampilkan kode pemesanan tiket.
Alternative Flow :	
1.	Menangani kegagalan dalam proses pemesanan tiket
Jumlah transaksi : 13	

Berdasarkan tabel 4 jumlah transaksi yang terjadi pada *use case* memesan tiket adalah 13 transaksi, hal tersebut dihitung berdasarkan 12 langkah yang ada pada *basic flow* dan 1 langkah yang ada pada *alternative flow* pada *use case* memesan tiket.

5.2. Nilai Technical Complexity Factor

Penelitian Ani dan Basri memberikan nilai 3 terhadap *technical factor*. Ani dan Basri menjelaskan nilai 3 diberikan pada *technical factor* adalah karena berdasarkan jumlah *use case* yang ada kurang dari 50. Jumlah *use case* yang kurang dari 50 berarti sistem tersebut tidak terlalu rumit. Jumlah *use case* pada sistem DBA *ticketing* adalah 11 *use case*. Menurut

penelitian Ani dan Basri (2013), penulis memberikan nilai 3 pada *technical factor* yang berarti faktor memiliki pengaruh yang biasa saja terhadap pengembangan sistem.

Tabel 5 berikut ini adalah penghitungan *Technical Complexity Factor* Sistem DBA *ticketing* yang didapatkan melalui perkalian bobot setiap *technical factor* dengan nilai *perceived complexity* yang sudah ditetapkan.

Tabel 5. Penghitungan TCF

Total Technical factor	42
TCF	$0,6 + (0,01 * 42) = 1,02$

Dari hasil perhitungan pada tabel 5 nilai *Technical factor* adalah 42 dan nilai *Technical Complexity Factor* (TCF) pada sistem DBA *ticketing* adalah **1,02**.

5.3. Nilai Environmental Complexity Factor

Terdapat Terdapat 8 *Environmental factor* pada lembar penilaian yang mana setiap faktor memiliki skala nilai yang telah ditentukan. Penilaian pada *Environmental factor* dilakukan oleh manajer proyek. Pada tabel 6 berikut merupakan hasil penghitungan *Environmental factor* sistem DBA *ticketing*.

Tabel 6. Penghitungan ECF

Total Environmental factor	15
Environmental Complexity Factor (ECF)	$1.4 + (-0,03 * 15) = 0,95$

Dari hasil perhitungan pada tabel 6 nilai *Environmental factor* adalah 15 dan nilai *Environmental Complexity Factor* (TCF) pada sistem DBA *ticketing* adalah 0,95.

5.4. Nilai Use Case Point

Tabel 7. Penghitungan ECF

Perhitungan	Hasil
UUCP	157
TCF	1,02
ECF	0,95
Use Case point (UCP)	$157 * 1,02 * 0,95 = 152,133$

Pada Tabel 7 di atas, dijelaskan bahwa nilai dari *Unadjusted Use Case Point* (UUCP) sistem DBA *ticketing* adalah 157, nilai dari *Technical Complexity Factor* (TCF) adalah 1,02, dan nilai dari *Environmental Complexity*

Factor (ECF) adalah 0,95 dan nilai *use case point* (UCP) sebesar 152,133.

5.5. Nilai Produktifitas

Nilai produktifitas pada metode *ANN model based on Use Case Point* menggunakan nilai dari *Environmental complexity factor* (ECF) (B. Nassif, 2011). Sebelumnya pada penelitian ini nilai *Environmental complexity factor* (ECF) sudah didapatkan karena nilai *Environmental complexity factor* (ECF) adalah komponen yang digunakan untuk menghitung nilai *Use Case Point* (UCP). Nilai produktifitas sistem *DBA ticketing* adalah 0,95.

5.5. Bobot Kompleksitas

Nilai kompleksitas diukur melalui 5 level pada metode *ANN model based on Use Case Point* (A. B. Nassif, 2012) dengan masing – masing level memiliki karakteristik tersendiri. Berdasarkan hasil wawancara dengan pihak CV. Profile Image Studio diketahui bahwa sistem *DBA ticketing* masuk kedalam level 2, karena diketahui bahwa sekitar 10% dari desain atau bagian proyek sistem berasal dari proyek yang serupa sehingga, bobot kompleksitas sistem *DBA ticketing* adalah 2.

5.6. Menghitung Effort dengan ANN model based on UCP

Tabel 8 merupakan penghitungan effort sistem *DBA ticketing* menggunakan *ANN model based on use case point*.

Tabel 8. Rancangan Analisis Komputasi

Perhitungan	Hasil
UCP	152,133
Produktifitas	0,95
Kompleksitas	2
Nilai effort	$3.53 + (0.88 \times 152,133) - (0.009 \times 0,95) + (0.31 \times 2) = 138,01$

Pada tabel 8 dijelaskan bahwa nilai dari UCP sebesar 152,133, kemudian nilai Produktifitas adalah 0,95 dan nilai kompleksitas adalah 2 dan hasil effort sistem *DBA ticketing* adalah 138,01 dalam satuan *person hours*.

5.6. Menghitung effort, waktu, staf

Nilai effort yang sudah didapatkan kemudian didistribusikan pada masing-masing aktivitas, kemudian total waktu didistribusikan pada masing-masing aktivitas juga sehingga

dapat ditemukan jumlah sumber daya manusia pada masing-masing aktivitas sesuai pada tabel 9.

Tabel 9. Rancangan Analisis Komputasi

Aktivitas	% Effort	Effort	hours	Staff
Software Phases				
Requirements	7,5	10.35	93,5	1
Specifications	7,5	10.35	93,5	1
Design	10	13.8	124,7	1
Implementation	10	13.8	124,7	1
Integration Testing	7,5	10.35	93,5	1
Acceptance & Deployment	7,5	10.35	93,5	1
Ongoing lifecycle activities				
Project Management	8,34	11.5	104	1
Configuration Management	4,16	5.7	51,8	1
Quality Assurance	8,34	11.5	104	1
Documentation	4,16	5.7	51,8	1
Training and support	4,16	5.7	51,8	1
Evaluation and Testing	20,84	28.7	259,9	1
Total		138,01	1247,46	12

5.6. Menghitung total biaya

Standar gaji mengacu pada standar gaji UMK Kota Malang pada tahun 2017. Membagi standar gaji per bulan dengan jumlah jam kerja pada perusahaan selama satu bulan akan didapatkan standar gaji pegawai per jam. Jumlah hari kerja pada CV. Profile Image Studio adalah 20 hari setiap bulannya dan jumlah jam kerja adalah 8 jam. Total jumlah jam kerja CV. Profile Image Studio adalah sebesar 160 jam. Maka standart gaji per bulan dibagi dengan 160, maka akan didapatkan hasilnya yaitu Standar Gaji Per Jam.

Tabel 10. Menghitung Biaya

Fase	Total Estimasi Biaya
<i>Software Phases</i>	Rp 8.852.280
<i>Ongoing life-cycle activities</i>	Rp 8.850.860
Total	Rp 17.703.140

Pada Tabel 10 didapatkan bahwa total biaya pada fase *software phase* sebesar Rp 8.852.280 dan pada fase *ongoing life cycle* Rp 8.850.860 sehingga total biaya keseluruhan adalah Rp 17.703.140.

6. KESIMPULAN

Dari hasil implementasi metode *Artificial Neural Network model based on use case point* pada penelitian ini kesimpulannya adalah:

1. Estimasi effort atau usaha yang diperoleh dengan menggunakan metode *ANN model based on use case point* pada Sistem DBA ticketing adalah 138,02 (*person hours*). Sedangkan estimasi effort atau usaha yang diperoleh dengan menggunakan metode *ANN model based on use case point* pada Sistem Pintu Air adalah 53,14 (*person hours*).
2. Estimasi jumlah sumber daya manusia (staff) untuk pengembangan sistem DBA ticketing berjumlah 12 orang dan estimasi jumlah sumber daya manusia (staf) untuk pengembangan sistem Pintu Air berjumlah 12 orang.
3. Estimasi durasi waktu pengembangan sistem DBA ticketing adalah 1247,46 jam sedangkan untuk estimasi durasi waktu pengembangan sistem Pintu Air adalah 453,6 jam.
4. Estimasi biaya pengembangan Sistem DBA ticketing dengan mengimplementasikan metode *ANN model based on Use Case Point* adalah sebesar Rp 17.703.140. Sedangkan estimasi biaya pengembangan Sistem Pintu Air dengan mengimplementasikan metode *ANN model based on Use Case Point* adalah sebesar Rp 6.436.860.

DAFTAR PUSTAKA

Anda, B., 2002. Comparing effort estimates based on use cases with expert estimates. *Empirical Assessment in SoftwareEngineering* (EASE), (p. 13). Keele UK.

Marchewka, J., 2003. *Information Technology Project Management*. Hoboken, NJ Wiley.

Nassif, A., Capretz, L.F., dan Ho, D., 2012. *Estimating Software Effort Using an ANN Model Based on Use Case Points. 11th International Conference on Machine Learning and Applications*.

Nassif, A., Capretz, L.F., dan Ho, D., 2011. *Regression Model for Software Effort Estimation Based on the Use Case Point Method. 2011 International Conference on Computer and Software Modeling IPCSIT vol.14*.

Ribu, K. 2001. Estimating Object-Oriented Software Projects with *Use cases*. *Master of Science Thesis*. University of Oslo Department of Informatics.

Saleh, K., 2011. *Effort and Cost Allocation in Medium to Large Software Development Projects*. *International Journal of Computers* (1), 74-79.

Saleh, K. 2011. *Effort and Cost Allocation in Medium to Large Software Development Projects*. *International Journal of Computers* (1), 74-79.

Schwalbe, K., 2012. *Information Technology Project Management, 7th edition*. Course Technology, Cengage Learning.