

**ESTIMASI PERKIRAAN BIAYA DAN WAKTU PENGEMBANGAN  
SISTEM INFORMASI KULIAH KERJA NYATA (KKN)  
UNIVERSITAS LAMPUNG MENGGUNAKAN MODEL  
*CONSTRUCTIVE COST MODEL II (COCOMO II)***

**Skripsi**

**Oleh**

**ELVA JUASTA**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2023**

**ESTIMATION OF COST AND DEVELOPMENT TIME  
REAL WORK COLLEGE INFORMATION SYSTEM (KKN)  
LAMPUNG UNIVERSITY MODEL USING  
CONSTRUCTIVE COST MODEL II (COCOMO II)**

**Abstract**

**By**

**Elva Juasta**

*Success in working on a software project must start with good planning. This research uses the Cocomo II Model (Constructive Cost Model II) to measure the estimated cost and time of software development with a case study of the University of Lampung Real Work Lecture Information System (KKN). The Field Work Lecture Executing Agency (BP-KKN) must know the resources needed in the development of the University of Lampung Field Work Information System (KKN) to be able to determine a budget plan to serve as a reference in future system development and reduce the risk of time overruns. Costs, and human resources in the development of the system. Measurements using the Cocomo II model require Unadjusted Function Point (UFP) values, Scala Factor values, Effort Multiplier values, and Effort Estimation values. By using the COCOMO II model, it is hoped that a more accurate estimate of the costs, time, and human resources needed to develop the University of Lampung's KKN information system will be obtained. The estimation results show that the cost required to develop the University of Lampung KKN information system is IDR 1,006,848,414, while the time required is 23 months, requiring 18 human resources.*

**Keywords:** *Cocomo II, Scala Factor, Effort Multiplier, Effort Estimation.*

**ESTIMASI PERKIRAAN BIAYA DAN WAKTU PENGEMBANGAN  
SISTEM INFORMASI KULIAH KERJA NYATA (KKN)  
UNIVERSITAS LAMPUNG MENGGUNAKAN MODEL  
*CONSTRUCTIVE COST MODEL II (COCOMO II)***

**Abstrak**

**Oleh**

**Elva Juasta**

Keberhasilan dalam mengerjakan suatu proyek perangkat lunak harus dimulai dengan perencanaan yang baik. Penelitian ini menggunakan Model Cocomo II (*Constructive Cost Model II*) untuk mengukur estimasi biaya dan waktu pengembangan perangkat lunak dengan studi kasus Sistem Informasi Kuliah Kerja Nyata (KKN) Universitas Lampung. Badan Pelaksana Kuliah Kerja Nyata (BP-KKN) harus mengetahui sumber daya yang diperlukan dalam pengembangan Sistem Informasi Kuliah Kerja Nyata (KKN) Universitas Lampung agar dapat menentukan rencana anggaran yang dijadikan sebagai bahan acuan dalam pengembangan sistem di masa mendatang, dan mengurangi risiko pembengkakan waktu, biaya, sumber daya manusia dalam pengembangan sistem. Pengukuran menggunakan model Cocomo II membutuhkan nilai *Unadjusted Function Point* (UFP), nilai *Scala Factor*, nilai *Effort Multiplier*, dan nilai *Effort Estimation*. Dengan menggunakan model COCOMO II, diharapkan dapat diperoleh perkiraan yang lebih akurat tentang biaya, waktu, dan sumber daya manusia yang diperlukan untuk mengembangkan sistem informasi KKN Universitas Lampung. Hasil estimasi menunjukkan bahwa biaya yang diperlukan untuk mengembangkan sistem informasi KKN Universitas Lampung adalah sebesar Rp 1.006.848.414, sedangkan waktu yang diperlukan adalah 23 bulan, dengan membutuhkan sumber daya manusia sebanyak 18 orang.

**Kata Kunci :** *Cocomo II, Scala Factor, Effort Multiplier, Effort Estimation.*

**ESTIMASI PERKIRAAN BIAYA DAN WAKTU PENGEMBANGAN  
SISTEM INFORMASI KULIAH KERJA NYATA (KKN)  
UNIVERSITAS LAMPUNG MENGGUNAKAN MODEL  
*CONSTRUCTIVE COST MODEL II (COCOMO II)***

Oleh

**ELVA JUASTA**

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar  
**SARJANA ILMU KOMPUTER**

Pada

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2023**

Judul Skripsi

: **ESTIMASI PERKIRAAN BIAYA DAN  
WAKTU PENGEMBANGAN SISTEM  
INFORMASI KULIAH KERJA NYATA  
(KKN) UNIVERSITAS LAMPUNG  
MENGUNAKAN MODEL *CONSTRUCTIVE  
COST MODEL II (COCOMO II)***

Nama Mahasiswa

: **Elva Juasta**

Nomor Pokok Mahasiswa

: 1617051007

Program Studi

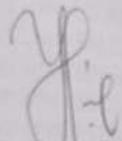
: Ilmu Komputer

Fakultas

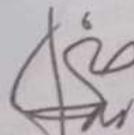
: Matematika dan Ilmu Pengetahuan Alam

**MENYETUJUI**

1. Komisi Pembimbing

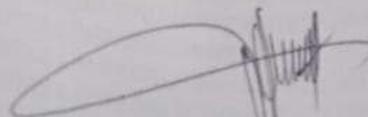


**Anie Rose Irawati, S.T., M.Cs.**  
NIP 19791031 200604 2 002



**Dewi Asiah Shofiana, S.Komp., M.Kom.**  
NIP 19950929 202012 2 003

2. Ketua Jurusan Ilmu Komputer

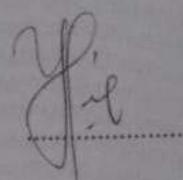


**Didik Kurniawan, S.Si., M.T.**  
NIP 19800419 200501 1 004

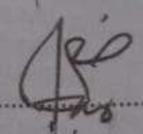
**MENGESAHKAN**

1. Tim Penguji

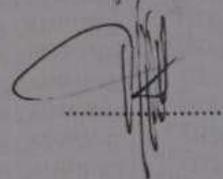
Ketua : **Anle Rose Irawati, S.T., M.Cs.**



Sekretaris : **Dewi Asiah Shoflana, S.Komp., M.Kom.**



Penguji  
Bukan Pembimbing : **Didik Kurniawan, S.Si., M.T.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Dr. Eng-Sumpto Dwi Yuwono, M.T.**  
NIP 19740705 200003 1 001

Tanggal Lulus Ujian Skripsi : **19 Januari 2023**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "**Estimasi Perkiraan Biaya dan Waktu Pengembangan Sistem Informasi Kuliah Kerja Nyata (KKN) Universitas Lampung Menggunakan Model *Constructive Cost Model II (Cocomo II)***" ini merupakan hasil karya sendiri dan bukan hasil karya orang lain. Semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar akademik yang telah saya terima.

Bandar Lampung, 19 Januari 2023



Elva Juasta  
1617051007

## **RIWAYAT HIDUP**

Penulis dilahirkan pada 2 Januari 1998 di Kota Bandar Lampung, merupakan putri kedua dari pasangan Bapak Iskandar dan Ibu Indra Puji. Penulis menyelesaikan pendidikan formal pertamanya di Taman Kanak-Kanak (TK) Kartini II Bandar Lampung pada tahun 2004, melanjutkan sekolah dasar di SD Negeri 1 Palapa Bandar Lampung dan selesai pada tahun 2010. Kemudian, penulis melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 18 Bandar Lampung yang diselesaikan pada tahun 2013, serta menyelesaikan pendidikan sekolah menengah kejuruan pada tahun 2016 di SMK Negeri 1 Bandar Lampung dengan Jurusan Administrasi Perkantoran.

Pada tahun 2016, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung dengan jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN). Selama menjadi mahasiswa, penulis mengikuti beberapa kegiatan antara lain:

1. Anggota Abacus Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2016/2017.
2. Anggota bidang Media Komunikasi Himpunan Mahasiswa Jurusan Ilmu Komputer (HIMAKOM) periode 2017.
3. Pada bulan Januari 2017 penulis melaksanakan karya wisata ilmiah di Desa Margosari, Kecamatan Pagelaran Utara, Kabupaten Pringsewu
4. Pada bulan Desember 2018 penulis melaksanakan kerja praktik di Badan Pendapatan Daerah Kabupaten Pesawaran.
5. Pada bulan Juli 2019 penulis melaksanakan KKN di Desa Kedaton, Kecamatan Abung Tengah, Kabupaten Lampung Utara.

6. Pada bulan Oktober 2020 penulis mengikuti ujian sertifikasi yang diselenggarakan oleh BNSP dengan kualifikasi atau kompetensi sebagai *Junior Office Operator*.

## **MOTTO DAN PERSEMBAHAN**

“Di tengah kesulitan terdapat kesempatan”

-Albert Einstein-

Skripsi ini saya persembahkan kepada:

Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini

Terima kasih kepada kedua orang tua, Ayah dan Bunda, yang selalu memberikan kasih sayang, dukungan, semangat, dan doa.

Terima kasih kepada Bapak dan Ibu dosen di Jurusan Ilmu Komputer yang senantiasa membagikan ilmu dan memberikan nasihat yang memotivasi.

Terima kasih kepada teman-teman Ilmu Komputer 2016 yang juga selalu mendukung dan berjuang bersama dalam meraih cita-cita.

## SANWANCANA

Puji syukur kehadirat Allah Subhanahu wa ta'ala yang telah melimpahkan rahmat, hidayah serta inayah-Nya sehingga penulis dapat menyelesaikan skripsi ini berjudul “Estimasi Perkiraan Biaya Dan Waktu Pengembangan Sistem Informasi Kuliah Kerja Nyata (Kkn) Universitas Lampung Menggunakan Model *Constructive Cost Model II (Cocomo II)*”. Tidak lupa shalawat serta salam kepada Nabi Muhammad Shallahu ‘alaihi wasallam, yang kita nanti-nantikan syafaatnya di yaumul akhir kelak.

Penulis mengucapkan terimakasih kepada semua pihak yang telah membantu dan memiliki peran besar dalam penyusunan skripsi ini, yaitu:

1. Kedua orang tuaku tercinta, ayahku Iskandar dan bundaku Indra Puji yang selalu memberikan semua dukungan, kasih sayang dan segala doa yang tiada hentinya.
2. Teteuku Sisca Mellyda, S.Pd., yang selalu mendukung dan membantu adikmu ini dalam menyelesaikan skripsi, serta adikku Isnaini Ilyah yang selalu menemani kakak dalam penyelesaian skripsi ini.
3. Ibu Anie Rose Irawati, S.T., M.Cs., selaku dosen pembimbing utama atas kesediaannya dan kesabarannya untuk memberikan dukungan, bimbingan, kritik, dan saran dalam proses penyelesaian skripsi.
4. Ibu Dewi Asiah Shofiana, S.Komp., M.Kom., selaku dosen pembimbing kedua atas kesediaannya dalam memberikan bimbingan, nasihat, juga kritik, dan saran selama proses pengerjaan skripsi.
5. Bapak Didik Kurniawan, S.Si., M.T., selaku dosen pembahas skripsi sekaligus Ketua Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, yang telah memberikan saran dan masukan guna penyempurnaan penulisan skripsi.

6. Bapak Dekan Dr. Eng. Suropto Dwi Yuwono, S.Si., M.T., selaku Dekan FMIPA Universitas Lampung.
7. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc., selaku Sekertaris Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer Universitas Lampung yang telah memberikan ilmu dan pengetahuan hidup selama penulis menjadi mahasiswa.
9. Ajijah, Amrina Rosada, dan Friska Daesy yang telah memberkan motivasi serta semangat dalam menyusun skripsi ini.
10. Semua pihak yang secara langsung maupun tidak langsung yang telah membantu menyelesaikan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, akan tetapi semoga skripsi ini dapat membawa manfaat dan keberkahan bagi perkembangan ilmu pengetahuan terutama bagi semua civitas Ilmu Komputer Universitas Lampung.

Bandar Lampung, 19 Januari 2023

Elva Juasta  
1617051007

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI</b> .....	<b>xii</b>
<b>DAFTAR TABEL</b> .....	<b>xiv</b>
<b>DAFTAR GAMBAR</b> .....	<b>xv</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xvi</b>
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
<b>II. TINJAUAN PUSTAKA</b> .....	<b>4</b>
2.1 Penelitian Terdahulu.....	4
2.2 Sistem Informasi.....	6
2.3 Badan Pelaksana Kuliah Kerja Nyata (BP-KKN).....	6
2.4 <i>Software Metrics</i> .....	6
2.5 Cocomo.....	7
2.6 Cocomo II.....	8
2.7 <i>Unadjusted Function Point (UFP)</i> .....	9
2.8 <i>Scala Factor</i> .....	12

2.9 <i>Effort Multiplier</i> .....	19
2.10 <i>Effort Estimation</i> .....	33
<b>III. METODE PENELITIAN .....</b>	<b>35</b>
3.1 Waktu dan Tempat Penelitian .....	35
3.2 Alat dan Data Penelitian.....	36
3.3 Tahapan Penelitian .....	37
<b>IV. PEMBAHASAN .....</b>	<b>40</b>
4.1 Analisis Perhitungan Nilai <i>Source Lines Of Code</i> (SLOC) .....	40
4.2 Analisis Perhitungan Nilai <i>Scala Factor</i> .....	44
4.3 Analisis Perhitungan Nilai <i>Effort Multiplier</i> .....	59
4.4 Analisis Perhitungan Nilai <i>Effort Estimation</i> .....	69
<b>V. SIMPULAN DAN SARAN.....</b>	<b>72</b>
5.1 Simpulan.....	72
5.2 Saran.....	72
<b>DAFTAR PUSTAKA .....</b>	<b>74</b>

## DAFTAR TABEL

Tabel	Halaman
1	Penelitian Terdahulu ..... 4
2	Komponen <i>Unadjusted Function Point</i> (UFP) ..... 9
3	<i>QSM Function Point Language</i> ..... 10
4	Atribut <i>Scala Factor</i> ..... 12
5	Tingkat Penilaian PREC ( <i>Precedentedness</i> ) ..... 13
6	Tingkat Penilaian FLEX ( <i>Development Flexibility</i> ) ..... 13
7	Tingkat Penilaian RESL ( <i>Risk Resolution</i> ) ..... 15
8	Tingkat Penilaian TEAM ( <i>Team Cohesion</i> ) ..... 16
9	Penilaian PMAT ..... 17
10	Penilaian <i>Scala Factor</i> ..... 18
11	Pengelompokan <i>Effort Multiplier</i> ..... 19
12	Atribut <i>Effort Multiplier</i> ..... 19
13	Tingkat penilai <i>effort multiplier</i> ..... 22
14	Penilaian <i>Effort Multiplier</i> ..... 30
15	Jadwal Penelitian..... 35
16	UFP Keseluruhan Fungsi Sistem Informasi Kuliah Kerja Nyata (KKN) Universitas Lampung..... 40
17	Pertanyaan dan Hasil Kuesioner <i>Scala Factor</i> ..... 46
18	Pertanyaan dan Hasil Kuesioner KPA ..... 52
19	Hasil Perhitungan KPA ..... 58
20	Pertanyaan dan Hasil Kuesioner <i>Effort Multiplier</i> ..... 61
21	Hasil Perhitungan <i>Effort Multiplier</i> ..... 68

## DAFTAR GAMBAR

Gambar	Halaman
1 Tahapan Penelitian .....	37

## DAFTAR LAMPIRAN

Lampiran	Halaman
1 Kuisisioner penelitian yang diisi secara <i>offline</i> dan dokumentasi pengisian kuisisioner penelitian oleh (Putra, 2019) .....	77
2 Tingkat penilaian KPA ( <i>Key Proses Area</i> ) .....	78

## **I. PENDAHULUAN**

### **1.1 Latar Belakang**

Kuliah Kerja Nyata (KKN) merupakan kegiatan pengabdian yang dilakukan oleh mahasiswa Universitas Lampung yang terkait dengan mata kuliah wajib mahasiswa pada jenjang S1. Kegiatan kuliah kerja nyata bagi mahasiswa Universitas Lampung dikelola oleh Badan Pelaksana Kuliah Kerja Nyata Universitas Lampung (BP-KKN). Dalam pelaksanaan kuliah kerja nyata mahasiswa perlu terjun langsung ke masyarakat dengan maksud menjadi anggota masyarakat, serta menggunakan ilmu dan pengalamannya untuk memecahkan masalah-masalah sosial.

Sistem Informasi Kuliah Kerja Nyata Universitas Lampung merupakan salah satu sistem informasi yang digunakan untuk mahasiswa yang ingin melaksanakan Kuliah Kerja Nyata yang ada di Universitas Lampung. Sistem Informasi Kuliah Kerja Nyata Universitas Lampung ini sebelumnya sudah dibuat oleh (Almaza, 2018) dan dikembangkan kembali fitur memasukan profil desa, mengubah profil desa, dan melihat ruang pembekalan oleh (Putra, 2019) dan digunakan sampai sekarang. Seiring dengan kebutuhan proses bisnis dan pengembangan organisasi, Sistem Informasi Kuliah Kerja Nyata Universitas Lampung akan terus diperbarui dan dikembangkan.

Diperlukan suatu indikator atau ukuran untuk dapat melihat dan memperkirakan sumber daya untuk mengembangkan sistem. Pengukuran perangkat lunak merupakan hal yang terpenting untuk mengetahui keberhasilan suatu pengembangan perangkat lunak. Pengukuran perangkat lunak juga dapat menjadikan sebuah evaluasi pengembangan perangkat lunak, yang dapat mengetahui berapa jumlah biaya yang dikeluarkan untuk sebuah pengembangan

perangkat lunak, serta dapat mengetahui berapa lama waktu yang dibutuhkan untuk pengembangan perangkat lunak.

Pengukuran Sistem Informasi Kuliah Kerja Nyata Universitas Lampung bertujuan untuk mengetahui estimasi waktu, biaya, dan sumber daya manusia untuk pengembangan sistem. Pengukuran ini bertujuan sebagai landasan atau tolak ukur untuk pengembangan sistem di masa mendatang. Pengukuran perangkat lunak yang digunakan adalah model *Constructive Cost Model* (Cocomo) yang merupakan algoritme estimasi biaya perangkat lunak model yang dikembangkan oleh Barry Boehm W. Model ini menggunakan Persamaan *regresi* dasar, dengan parameter yang berasal dari data historis dan karakteristik proyek-proyek saat ini. Cocomo pertama kali diterbitkan pada tahun 1981 oleh Barry Boehm W dengan judul buku ekonomi *software engineering* sebagai model untuk memperkirakan usaha, biaya, dan jadwal untuk proyek-proyek perangkat lunak (Boehm, 2000).

Cocomo II dipublikasikan pada tahun 1997 dengan tujuan untuk pendekatan yang lebih baik terhadap estimasi biaya perangkat lunak. Cocomo II adalah sebuah metode perkiraan biaya yang objektif untuk perencanaan dan pelaksanaan proyek-proyek perangkat lunak. Selain itu, model Cocomo II berfungsi untuk mengetahui estimasi perkiraan biaya dan waktu dalam pengembangan perangkat lunak.

Penelitian ini menggunakan model Cocomo II yang digunakan untuk memperkirakan estimasi biaya, waktu pengembangan, dan sumber daya manusia serta menjadikan landasan atau tolak ukur untuk pengambilan keputusan oleh Badan Pelaksana Kuliah Kerja Nyata Universitas Lampung dalam proses pembuatan perangkat lunak Sistem Informasi Kuliah Kerja Nyata Universitas Lampung di pengembangan selanjutnya.

## **1.2 Rumusan Masalah**

Rumusan masalah penelitian ini adalah bagaimana mengimplementasikan pengukuran untuk estimasi biaya dan waktu pengembangan Sistem Informasi Kuliah Kerja Nyata Universitas Lampung menggunakan model Cocomo II.

### 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini sebagai berikut:

- a. Objek penelitian ini adalah fungsi Sistem Informasi Kuliah Kerja Nyata Universitas Lampung yang dikembangkan oleh (Putra, 2019).
- b. Menggunakan model Cocomo II dengan menentukan nilai *unadjusted function point* (UFP), *scala factor*, *effort multiplier* dan *effort estimation*.
- c. Menggunakan sub model *post-architecture*.
- d. Nilai *Unadjusted Function Point* (UFP) didapatkan dari penelitian terdahulu (Anggraeni, 2021).

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk dapat mengetahui pengukuran estimasi biaya, waktu pengembangan sistem dan sumber daya manusia. Selain itu penelitian ini bertujuan untuk menjadi landasan atau tolak ukur pengembangan Sistem Informasi Kuliah Kerja Nyata Universitas Lampung di masa mendatang.

### 1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini, antara lain:

- a. Sebagai pendukung dalam pengambilan keputusan untuk pengembangan Sistem Informasi Kuliah Kerja Nyata Universitas Lampung di masa mendatang.
- b. Menjadi rekomendasi persiapan sumber daya manusia, waktu, maupun biaya untuk pengembangan Sistem Informasi Kuliah Kerja Nyata Universitas Lampung selanjutnya.

## II. TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu sumber untuk melakukan penelitian baru. Sebagai referensi penelitian ini ada beberapa artikel jurnal tentang model Cocomo II ditunjukkan pada Tabel 1.

Tabel 1. Penelitian Terdahulu

Keterangan	Masalah	Judul	Hasil
Damayanti dkk, 2017.	Bagaimana cara analisis pembuatan perangkat lunak dengan metode Cocomo II.	Analisis Estimasi Biaya Pembuatan Perangkat Lunak Menggunakan Metode Cocomo II di Inagata Technosmith (Studi Kasus : Sistem Informasi Monitoring dan Evaluasi Penerimaan Beasiswa Berprestasi UIN Malang)	Penelitian ini menunjukkan bahwa dengan metode Cocomo II dapat mengetahui perkiraan biaya yang dibutuhkan serta waktu penyelesaian sistem yang dibuat. Dengan adanya metode Cocomo II ini dapat diketahui estimasi yang dibutuhkan sehingga tidak mengakibatkan hasil perkiraan yang berlebihan.

Tabel 1. Penelitian Terdahulu (Lanjutan)

Keterangan	Masalah	Judul	Hasil
Khoiro dkk, 2018.	Bagaimana mengevaluasi pengembangan perangkat lunak dengan Metode Cocomo II.	Evaluasi Pengembangan Perangkat Lunak dengan menggunakan Metode Cocomo II (Studi Kasus : PT DOT Indonesia).	Penelitian ini menunjukkan hasil perhitungan estimasi pembuatan perangkat lunak menggunakan Motode Cocomo II. Hasilnya yaitu dapat diketahui estimasi perhitungan SDM, perhitungan waktu pengerjaan perangkat lunak, dan estimasi biaya pengerjaan perangkat lunak.
Fatonah dkk, 2017.	Bagaimana mengevaluasi pengembangan perangkat lunak dengan Metode Cocomo II	Model estimasi biaya perangkat lunak menggunakan Cocomo II (studi kasus PT. X).	Penelitian ini menunjukkan bahwa dengan menggunakan Model Cocomo II dapat memberikan hasil prediksi, dan penelitian ini juga dapat memberikan acuan perhitungan pembuatan estimasi usaha proyek selanjutnya.

Berdasarkan beberapa penelitian terdahulu pada Tabel 1, dapat diketahui bahwa model Cocomo II dapat diterapkan dalam menentukan estimasi biaya, waktu, serta sumber daya manusia dalam pembuatan perangkat lunak. Penelitian yang sudah dilakukan bertujuan untuk mengetahui pengukuran estimasi biaya, waktu pengembangan sistem, dan sumber daya manusia, serta menjadi tolak ukur untuk pengembangan sistem dimasa mendatang.

## 2.2 Sistem Informasi

Sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, mengelola, dan menyimpan data. Selain itu sistem informasi didefinisikan sebagai cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan (Krismaji, 2015).

## 2.3 Badan Pelaksana Kuliah Kerja Nyata (BP-KKN)

Badan Pelaksana Kuliah Kerja Nyata (BP-KKN) ini diketuai oleh Bapak Sri Waluyo, Ph.D, (Periode 2022). Saat ini BP-KKN memiliki sistem informasi Kuliah Kerja Nyata guna untuk mengelola data mahasiswa Universitas Lampung yang ingin melakukan Kuliah Kerja Nyata mulai dari pendaftaran mahasiswa sampai dengan penempatan mahasiswa di kecamatan atau desa yang telah ditentukan.

Tujuan dari BP- KKN adalah sebagai berikut:

- a. Meningkatkan empati dan kepedulian mahasiswa.
- b. Melaksanakan terapan IPTEK secara *teamwork* dan *interdisipliner*.
- c. Menanamkan nilai kepribadian.
- d. Meningkatkan daya saing nasional.
- e. Menambah jiwa peneliti (BP-KKN Universitas Lampung, 2018).

## 2.4 *Software Metrics*

*Software metrics* merupakan sarana untuk mengukur aspek-aspek tertentu mengenai perangkat lunak yang berhubungan dengan faktor-faktor kualitas yang terdapat pada model kualitas perangkat lunak. Untuk mendapatkan nilai pada *metrics* ini harus dilakukan perhitungan terhadap aspek-aspek tertentu pada perangkat lunak. Akan tetapi karena kendala skala dari perangkat lunak, perhitungan manual ini menjadi sangat sulit. Oleh karena itu dibutuhkan *tools* untuk membantu *developer* mendapatkan nilai-nilai *metrics* dari perangkat lunak yang sedang dikembangkan atau akan dikembangkan (Rahensa dkk, 2015).

*Software metrics* juga dapat diartikan sebagai entitas penting di dalam seluruh siklus hidup *software*. *Software metrics* menyediakan pengukuran untuk pengembangan *software*, termasuk dokumen persyaratan *software*, desain, program, dan pengujian (Rawat dkk, 2012). Pengukuran dapat digunakan pada seluruh proyek *software* untuk membantu estimasi, kontrol kualitas, penilaian produktivitas, pengendalian proyek, membantu menilai kualitas produk, dan membantu dalam pengambilan keputusan hasil proyek (Sidabutar, 2015). Pengukuran merupakan dasar dari setiap rekayasa dan berlaku juga dalam perikayasaan perangkat lunak. Untuk mengevaluasi performa suatu sistem atau proses diperlukan suatu mekanisme untuk mengamati dan menentukan tingkat efesiansinya. Melalui pengukuran ini maka akan diperoleh tingkat pencapaian di dalam perangkat lunak yang sedang dikerjakan atau diamati. Pengukuran menurut IEEE (1990) adalah kuantitatif dari tingkat sebuah sistem, komponen atau sebuah proses yang memiliki atribut tertentu.

## 2.5 Cocomo

*Cost Constructive Model* (Cocomo) adalah teknik estimasi perangkat lunak menggunakan *source of line code* (SLOC) yang digunakan untuk mengestimasi biaya, tenaga yang diperlukan, dan jadwal dalam pengembangan perangkat lunak. Pada tahun 1981, Barry Boehm mendesain Cocomo untuk memberikan estimasi jumlah *person-month* untuk mengembangkan perangkat lunak. Referensi pada model ini disebut dengan nama Cocomo 81. Model ini memiliki tiga mode yang dibagi berdasarkan lingkup kerja, ukuran dari perangkat lunak, dan batasan-batasan yang terkait dalam pengembangan perangkat lunak berikut ini adalah mode Cocomo 81 (Boehm, 2000) :

- a. *Organic Mode* adalah proyek dengan ukuran relatif kecil, dengan anggota tim yang sudah berpengalaman dan mampu bekerja pada permintaan yang relatif fleksibel.
- b. *Semi-Detached Mode* adalah proyek yang memiliki ukuran dan tingkat kerumitan yang sedang dan tiap anggota tim memiliki tingkat keahlian yang berbeda.

- c. *Embedded Mode* adalah proyek yang dibangun dengan spesifikasi dan operasi yang ketat. Pengembangan perangkat lunak ini terikat dengan perangkat keras, perangkat lunak, regulasi, dan prosedur operasional yang kompleks (Boehm, 2000).

## 2.6 Cocomo II

Cocomo II (*Constructive Cost Model II*) adalah sebuah metode yang memperkirakan sumber daya yang objektif untuk sebuah perencanaan dan pelaksanaan perangkat lunak yang menghasilkan model estimasi yang dapat digunakan sebagai masukan atau acuan untuk estimasi pengembangan perangkat lunak berikutnya. Cocomo II adalah salah satu teknik estimasi pemodelan biaya algoritmik yang merupakan pembaharuan dari Cocomo 81 oleh Barry Boehm pada tahun 1981. Cocomo 81 dan Cocomo II memiliki kemiripan. Cocomo 81 memiliki tiga mode berdasarkan ukuran perangkat lunak, lingkup kerja, dan batasan yang terkait dengan proyek perangkat lunak (Sidabutar, 2015).

Cocomo II dipublikasikan pada tahun 1997 dengan tujuan untuk pendekatan yang lebih baik terhadap estimasi biaya perangkat lunak. Cocomo II merupakan algoritmik estimasi biaya perangkat lunak yang mempunyai sub model di antaranya yaitu:

- a. *Application Composition Model* cocok digunakan untuk proyek yang akan dikembangkan.
- b. *Early Design Model* untuk mendapatkan estimasi kasar dari biaya dan waktu sebelum ditentukan arsitektur dari seluruh perangkat lunak yang akan dibangun. Pada pemodelan ini, digunakan sekumpulan kecil dari *Cost Drivers* baru dan persamaan estimasi baru yakni berdasarkan *Unadjusted Function Point* (Khoiro dkk, 2018).
- c. *Post-Architecture Model* merupakan pemodelan paling detail dibandingkan dengan dua pemodelan lainnya, yang digunakan setelah seluruh arsitektur dari proyek telah didesain. Salah satu komponen yang digunakan adalah *function point* dan LOC. Pemodelan ini melibatkan proses pembangunan dan perawatan dari sebuah produk perangkat lunak.

Cocoma II pada sub model *Post-Architecture Model* memiliki dua aspek yang sangat penting yaitu: *5 Scala Factor* yang merupakan faktor penentu eksponen yang digunakan untuk *effort estimation* dan *effort multiplier* yang digunakan untuk menyelesaikan proyek perangkat lunak.

## 2.7 *Unadjusted Function Point (UFP)*

*Unadjusted Function Point (UFP)* digunakan untuk menghitung bobot nilai dari komponen-komponen *function point* yang dikaitkan dengan *software* yang akan dibuat atau yang sudah dibuat dengan lima komponen (Damayanti dkk, 2017) yang ditunjukkan pada Tabel 2.

Tabel 2. Komponen *Unadjusted Function Point (UFP)*

Komponen	Keterangan
<i>External Input (EI)</i>	Jumlah setiap tipe data atau masukan kontrol pengguna yang memasuki batas luar dari sistem perangkat lunak yang sedang diukur dan menambah atau mengubah data dalam berkas internal logika.
<i>External Outputs (EO)</i>	Jumlah setiap tipe data atau keluaran kontrol yang meninggalkan batas luar dari sistem perangkat lunak yang sedang diukur.
<i>Internal Logical File (ILF)</i>	Jumlah setiap grup logika utama dari data atau informasi kontrol pengguna dalam sistem perangkat lunak sebagai tipe berkas internal logika yang meliputi setiap berkas logika yang dibuat, dan digunakan atau dirawat oleh sistem perangkat lunak.
<i>External Interface Files (EIF)</i>	Berkas yang dilewatkan atau dibagikan di antara sistem perangkat lunak seharusnya dihitung sebagai tipe berkas antar muka eksternal dalam setiap sistem.
<i>External Inquiry (EQ)</i>	Jumlah setiap kombinasi masukan keluaran, yang menyebabkan tipe <i>external inquiry</i> .

Setiap komponen *function point* diklasifikasi tingkat kompleksitasnya. Tingkat kompleksitas menentukan kualitas UFP, nilai bobot kompleksitas UFP

diklasifikasi menjadi 3 yaitu DET (*Data Elements Types*), RET (*Record Elements Types*), dan FTR (*Files Type References*).

- a. DET (*Data Elements Types*) dikendalikan oleh pengguna sebagai sesuatu yang unik dan tidak berulang. Elemen data yang unik untuk membedakan sebuah komponen *function point* yang satu dengan komponen *function point* yang lainnya, berupa data *input* untuk EI (*External Input*).
- b. RET (*Record Elements Types*) adalah sub group data yang dikendalikan oleh pengguna, ILF (*Internal Logical File*).
- c. FTR (*Files Type References*) diakses oleh parameter EI (*External Input*), EO (*External Outputs*), dan EQ (*External Inquiry*) untuk *insert*, *update*, dan *delete*.

Setiap bagian dari tipe fungsi diklasifikasikan oleh tingkat kerumitan. Tingkat kerumitan menentukan bobot yang akan diaplikasikan pada jumlah fungsi untuk menentukan kuantitas *Unadjusted Function Points* (Aditya dkk, 2018).

Setelah menentukan kompleksitas setiap fungsi perangkat lunak barulah diperoleh nilai UFP harus diubah ke dalam *Source Line of Code* (SLOC) menggunakan bahasa pemrograman berdasarkan QSM (*Quantitative Software Management*) sesuai bahasa pemrograman yang digunakan agar dapat disubstitusikan ke dalam Persamaan estimasi usaha sebagaimana ditunjukkan pada Tabel 3 (*QSM Function Point Language Table*, 2022)

Tabel 3. *QSM Function Point Language*

<i>Language</i>	QSM SLOC/FP			
	<i>Average</i>	<i>Median</i>	<i>Low</i>	<i>High</i>
ABAP (SAP)	28	18	16	60
ASP	51	54	15	69
Assembler*	119	98	25	320
Brio+	14	14	13	16
C*	97	99	39	333
C++*	50	53	25	80
C#*	54	59	29	70
COBOL*	61	55	23	297
Cognos Impromptu Scripts +	47	42	30	100
Croos System Products (CSP)	20	18	10	38
Cool:Gen/IEF	32	24	10	82
Datastage	71	65	31	157

Tabel 3. *QSM Function Point Language (Lanjutan)*

<i>Language</i>	<i>QSM SLOC/FP</i>			
	<i>Average</i>	<i>Median</i>	<i>Low</i>	<i>High</i>
Excel*	209	191	131	315
Focus*	43	45	45	45
FoxPro	36	35	34	38
HTML	34	40	14	48
J2EE*	46	49	15	67
Java*	53	53	14	134
JavaScript*	47	53	31	63
JCL*	62	48	25	221
LINC II	29	30	22	38
Lotus Notes*	23	21	19	40
Natural	40	34	34	53
NET	57	60	53	60
J2EE*	57	50	50	67
Java*	55	53	9	214
JavaScript*	54	55	45	63
JCL*	96	59	58	173
JSP	59	-	-	-
KML	50	50	49	50
Lotus Notes*	23	21	15	46
Maestro	30	30	30	30
Mantis	71	27	22	250
Mapper*	69	70	58	81
Natural*	51	53	34	60
NET	60	60	60	60
Oracle*	37	40	17	60
PACBASE	35	40	22	60
Perl*	24	15	15	60
PL/I*	64	80	16	80
PL/SQL	37	35	13	60
Powerbuiler*	26	28	7	40
REXX	77	80	50	80
Sabretalk*	70	66	45	109
SAS*	38	37	22	55
Siebel	59	60	51	60
SLOGAN	75	75	74	75
SQL*	21	21	13	37
VB.NET	52	60	26	60
Visual Basic*	42	44	20	60

Nilai *Source Line of Code* (SLOC) yang didapatkan dibagi dengan 1000 untuk mendapatkan nilai *Kilo Source Line Of Code* (KSLOC) yang dimasukkan dalam persamaan usaha.

## 2.8 *Scala Factor*

*Scala Factor* (SF) merupakan cara untuk menentukan usaha proyek dan juga karakteristik dari suatu proyek. *Scala factor* ini digunakan pada sub model Cocomo II *early design* dan juga *post-architecture model* (Damayanti dkk, 2017). Untuk perhitungan Persamaan nilai jumlah *factor* eksponen (E) dapat dihitung menggunakan Persamaan 1.

$$E = B + 0.01 \times SF \quad (1)$$

Keterangan:

E : Faktor Eksponen

B : Nilai Koefesien 0.91 (Untuk Cocomo II)

SF : Total Nilai *Scala Factor*

Nilai *scala factor* yang telah dikumpulkan langsung dengan cara kuesioner yang diisi oleh pengembang yaitu (Putra, 2019) terdiri dari 5 atribut *scala factor* yang ditunjukkan pada Tabel 4 (Milicic, 2004).

Tabel 4. Atribut *Scala Factor*

<i>Scala Factor</i>	Keterangan
PREC ( <i>Precedentness</i> )	Faktor skala ini merupakan faktor skala yang menggambarkan pengalaman masa lalu organisasi dengan proyek-proyek sejenis.
FLEX ( <i>Flexibility</i> )	Faktor skala ini merupakan faktor skala yang menggambarkan kemampuan <i>klien</i> dalam menentukan tujuan dan mengkomunikasikan kebutuhan perangkat lunak kepada tim pengembang.
RESL( <i>Risk Resolution</i> )	Faktor skala ini merupakan faktor skala yang menggambarkan seberapa sering perusahaan merencanakan manajemen risiko terhadap proyek yang dijalankan oleh perusahaan. Perencanaan manajemen risiko ini berguna untuk mengidentifikasi dan mencegah semua risiko yang mungkin terjadi.

Tabel 4. Atribut *Scala Factor* (Lanjutan)

<i>Scala Factor</i>	Keterangan
TEAM ( <i>Team Cohesion</i> )	Faktor skala ini merupakan faktor skala yang menggambarkan seberapa baik tim pengembangan bekerja sama.
PMAT ( <i>Process Maturity</i> )	Faktor skala yang menggambarkan kematangan proses pengembangan perangkat lunak dalam organisasi. Hal ini didasarkan pada Model Kematangan Kemampuan Rekayasa Perangkat Lunak atau <i>Capability Maturity Model</i> yang memiliki 5 tingkatan.

Tingkat penilaian *scala factor* sebagai berikut:

a. PREC (*Precedentedness*)

Faktor skala yang menggambarkan pengalaman organisasi terhadap perangkat lunak sebelumnya. PREC (*Precedentedness*) merupakan sebuah perangkat lunak yang mirip dengan beberapa yang dikembangkan sebelumnya, maka nilai *precedentedness* akan tinggi, yang ditunjukkan pada Tabel 5.

Tabel 5. Tingkat Penilaian PREC (*Precedentedness*)

<i>Feature</i>	VL	N	EH
Pemahaman organisasi tentang tujuan pengembangan perangkat lunak.	Umum	Lengkap	Terperinci
Berpengalaman bekerja dalam perangkat lunak.	Sedang	Lengkap	Luas
Pengembangan terkait <i>hardware</i> baru dan prosedur operasional.	Luas	Sedang	Beberapa
Perlu adanya data yang inovatif dan pengolahan arsitektur dan algoritme.	Lengkap	Beberapa	Minimal

Keterangan:

VL : *Very Low*, N : *Nominal*, dan EH : *Extra High*

b. FLEX (*Development Flexibility*)

FLEX (*Development Flexibility*) merupakan faktor skala yang menggambarkan kemampuan klien dalam menentukan tujuan, dan menyampaikan kebutuhan perangkat lunak kepada pengembang perangkat

lunak. *Development flexibility* menilai fleksibilitas yang menentukan tujuan sebuah perangkat lunak. Penilaian *development flexibility* dapat dinilai melalui *feature* yang ditunjukkan pada Tabel 6.

Tabel 6. Tingkat Penilaian FLEX (*Development Flexibility*)

<i>Feature</i>	VL	N	EH
Perlu adanya kesesuaian perangkat lunak dengan persyaratan yang telah ditentukan sebelumnya.	Penuh	Lengkap	Dasar
Perlu adanya kesesuaian perangkat lunak sesuai dengan spesifikasi antarmuka eksternal ( <i>mouse, keyboard, touch screen</i> , dan lainnya).	Penuh	Lengkap	Dasar
Ketidaksiesuaian waktu dan kebutuhan penyelesaian dalam pengembangan perangkat lunak.	Tinggi	Lengkap	Rendah

Keterangan:

VL : *Very Low*, N : *Nominal*, dan EH : *Extra High*

c. RESL (*Risk Resolution*)

*Risk Resolution* merupakan faktor skala yang menggambarkan identifikasi risiko yang terkait dengan perangkat lunak yang sedang dikembangkan, dan untuk mengatasi kegagalan perangkat lunak yang dapat dinilai dari *feature* ditunjukkan pada Tabel 7.

Tabel 7. Tingkat Penilaian RESL (*Risk Resolution*)

<i>Feature</i>	VL	L	N	H	VH	EH
Merencana manajemen risiko yang mengidentifikasi semua <i>item</i> risiko untuk mengatasi masalah dengan menggunakan PDR ( <i>product design review</i> ) atau LCA ( <i>life cycle architecture</i> ).	Tidak Satupun	Sedikit	Beberapa	Umum	Banyak	Sepenuhnya
Jadwal, anggaran, dan tugas yang harus melalui PDR ( <i>product design review</i> ) atau LCA ( <i>life cycle architecture</i> ) sesuai dengan rencana manajemen risiko.	Tidak Satupun	Sedikit	Beberapa	Umum	Banyak	Sepenuhnya
Persentase pengembangan jadwal perangkat lunak yang ditujukan untuk menetapkan arsitektur perangkat lunak secara umum.	5	10	17	25	33	40
Persentase dari pengembang perangkat lunak diperlukan untuk tersedianya proyek.	20	40	60	80	100	120
Ketersediaan alat untuk menyelesaikan <i>item</i> risiko, mengembangkan, dan memverifikasi spesifikasi perangkat lunak yang dibutuhkan.	Tidak Satupun	Sedikit	Beberapa	Baik	Kuat	Penuh
Tingkat ketidakpastian <i>driver</i> arsitektur utama yaitu: misi, antarmuka pengguna, COTS, <i>hardware</i> , teknologi, dan performa.	Ekstrim	Penting	Besar	Beberapa	Sedikit	Sangat Sedikit
Jumlah kekritisannya dari <i>item</i> risiko.	>10 Kritis	5-10 Kritis	2-4 Kritis	1 Kritis	>5 Non Kritis	>5 Non Kritis

Keterangan:

VL : *Very Low*, L : *Low*, N : *Nominal*, H : *High*, VH : *Very High*, dan EH : *Extra High*

d. TEAM (*Team Cohesion*)

*Team Cohesion* merupakan faktor skala yang menggambarkan kemampuan setiap anggota tim proyek dalam berkomunikasi dan bekerjasama. *Team Cohesion* juga berpengaruh pada sumber ketidakstabilan proyek dengan alasan bahwa kerumitan dalam sinkronisasi *stakeholder* proyek. Pengguna akhir, pelanggan, pengembang, pengelola, *interface*, dan semacamnya termasuk dalam kategori yang ditunjukkan pada Tabel 8.

Tabel 8. Tingkat Penilaian TEAM (*Team Cohesion*)

<i>Feature</i>	VL	L	N	H	VH	EH
Konsistensi <i>stakeholder</i> dengan tujuan dan sasaran pengembang.	Sedikit	Beberapa	Dasar	Lengkap	Kuat	Penuh
Kemampuan dan kemauan demi kepentingan untuk <i>stakeholder</i> dan tujuan lainnya.	Sedikit	Beberapa	Dasar	Lengkap	Kuat	Penuh
Pengalaman <i>stakeholder</i> dalam bekerja sama sebuah tim.	Tidak ada	Sedikit	Sedikit	Dasar	Lengkap	Luas
<i>Stakeholder</i> saling berusaha dalam mencapai tujuan, visi, dan komitmen bersama.	Tidak ada	Sedikit	Sedikit	Dasar	Lengkap	Luas

Keterangan:

VL : *Very Low*, L : *Low*, N : *Nominal*, H : *High*, VH : *Very High*, dan EH : *Extra High*

e. PMAT (*Process Maturity*)

*Process Maturity* merupakan faktor skala yang menggambarkan kematangan proses pengembangan perangkat lunak dalam organisasi. Menentukan PMAT (*Process Maturity*) adalah prosedur terstruktur, dan harus dipertimbangkan untuk jangka waktu yang memulai proyek. Khusus untuk faktor skala PMAT, sebelumnya perlu dilakukan wawancara untuk mengisi tabel KPA (*Key Proses Area*). Tabel KPA (*Key Proses Area*) mempunyai bobot mulai dari *Almost Always* hingga *Does Not Apply*. *Key KPA (Key Proses Area)* mempunyai bobot penilaian mulai dari 0%, 1%,

25%, 50%, 75%, dan 100% penentuan PMAT adalah prosedur memutuskan persentase kepatuhan di setiap KPA (*Key Proses Area*). Tingkat penilaian KPA (*Key Proses Area*) dapat dilihat pada Lampiran 2.

Kemudian KPA (*Key Proses Area*) dihitung rata-ratanya menggunakan EMPL (*Equivalent Process Maturity Level*) seperti pada Persamaan 2.

$$EMPL = \left( \sum_{i=1}^{17} KPA\%i / 100 \right) / 17$$

(2)

Keterangan:

EMPL : *Equivalent Process Maturity Level*

KPA : *Key Proses Area*

Setelah mendapatkan nilai EMPL (*Equivalent Process Maturity Level*) kemudian nilainya dicocokkan dengan penilaian PMAT (*Process Maturity*) yang ditunjukkan pada Tabel 9.

Tabel 9. Penilaian PMAT

Penilaian PMAT	<i>Maturity Level</i>	EMPL
<i>Very Low</i>	CMM Level 1 (Setengah lebih rendah)	0
<i>Low</i>	CMM Level 1 (Bagian atas setengah)	1
<i>Nominal</i>	CMM Level 2	2
<i>High</i>	CMM Level 3	3
<i>Very High</i>	CMM Level 4	4
<i>Extra High</i>	CMM Level 5	5

Setiap *scala factor* memiliki penilaian *rating* bobot yang telah ditentukan, nilai dari setiap bobot *scala factor* menggunakan sub model *early design* dan *post-architecture* yang ditunjukkan pada Tabel 10. Setiap nilai bobot *scala factor* dijumlahkan lalu dibagi dengan jumlah *scala factor* yang digunakan untuk menentukan faktor eksponen (E) (Milicic, 2004).

Tabel 10. Penilaian *Scala Factor*

<i>Scala Factor</i>	<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
PREC	<i>Thoroughly Unprecedented</i> 6.20	<i>Largely Unprecedented</i> 4.96	<i>Somewhat Unprecedented</i> 3.72	<i>Generally Familiar</i> 2.48	<i>Largely Familiar</i> 1.24	<i>Thoroughly Familiar</i> 0.00
FLEX	<i>Rigorous</i> 5.07	<i>Occasional Relaxation</i> 4.05	<i>Some Relaxation</i> 3.04	<i>General Conformity</i> 2.03	<i>Some Conformity</i> 1.01	<i>General Goals</i> 0.00
RESL	<i>Little (20%)</i> 5.48	<i>Some (40%)</i> 4.38	<i>often (60%)</i> 5.24	<i>Generally (75%)</i> 2.83	<i>Mostky (90%)</i> 1.41	<i>Full (100%)</i> 0.00
TEAM	<i>Very Difficult Interactions</i> 5.48	<i>Some Difficult Interactions</i> 4.38	<i>Basically Cooperative Interactions</i> 3.39	<i>Largely Cooperative</i> 2.19	<i>Highly Cooperative</i> 1.10	<i>Seamless Interactions</i> 0.00
PMAT	level 1 7.80	level 2 6.24	level 2+ 4.66	level 3 3.12	level 4 1.56.	level 5 0.00

## 2.9 Effort Multiplier

*Effort Multiplier* digunakan untuk mendapatkan nilai EM yang akan dijabarkan pada 17 *effort multiplier* yang digunakan dalam sub model *post architecture model* Cocomo II untuk menentukan nilai usaha nominal yang menggambarkan proyek perangkat lunak yang sedang dikembangkan (Damayanti dkk, 2017). Sebanyak 17 *effort multiplier* tersebut dikelompokkan ke dalam empat kelompok yaitu *product*, *platform*, *personnel* dan *project* yang ditunjukkan pada Tabel 11.

Tabel 11. Pengelompokan *Effort Multiplier*

Kategori	<i>Effort Multiplier</i>
<i>Product</i>	RELY, DATA, CPLX, RUSE, DOCU
<i>Platform</i>	TIME, STOR, PVOL
<i>Personnel</i>	ACAP, LTEX, APEX, PLEX, PCON, PCAP
<i>Project</i>	TOOL, SITE, SCED

*Effort multiplier* digunakan untuk menggambarkan sistem informasi yang dikembangkan. Berikut ini 17 atribut *effort multiplier* (Milicic, 2004) yang ditunjukkan pada Tabel 12.

Tabel 12. Atribut *Effort Multiplier*

Kategori	EM	Keterangan
<i>Product</i>	RELY ( <i>Required Software Reliability</i> )	Terkait sejauh mana perangkat lunak menjalankan aplikasi sesuai dengan fungsinya selama periode waktu.
	DATA ( <i>Data Base Size</i> )	Ukuran <i>database</i> yang digunakan. Ukuran dapat dihitung menggunakan D/P.
	CPLX ( <i>Product Complexity</i> )	Perangkat lunak dan perangkat keras dalam melakukan tugasnya seperti <i>platform</i> (arsitektur, sistem operasi, bahasa pemrograman dan antarmuka yang terkait), sistem manajemen <i>database</i> , <i>browser</i> yang sesuai digunakan dalam menjalankan aplikasi ini.

Tabel 12. Atribut *Effort Multiplier* (Lanjutan)

Kategori	EM	Keterangan
Platform	RUSE ( <i>Developed for Reusability</i> )	<i>Cost driver</i> terkait tingkat upaya yang diperlukan untuk mengembangkan komponen yang dimaksudkan untuk digunakan kembali pada proyek-proyek yang sedang berjalan atau proyek di masa mendatang.
	DOCU ( <i>Documentation Match to Life-Cycle Needs</i> )	Kesesuaian dokumentasi proyek terhadap kebutuhan siklus hidup perangkat lunak.
	TIME ( <i>Execution Time Constraint</i> )	Persentase kendala waktu eksekusi yang diharapkan dapat digunakan pada sistem perangkat lunak.
	STOR ( <i>Main Storage Constraint</i> )	Persentase tingkat kendala penyimpanan utama yang digunakan pada sistem perangkat lunak.
Personnel	PVOL ( <i>Platform Volatility</i> )	Perubahan yang terjadi pada <i>hardware</i> dan <i>software</i> dalam kurun waktu tertentu.
	ACAP ( <i>Analyst Capability</i> )	Kemampuan <i>personel</i> dalam analisis dan desain, secara efisien dan teliti, serta kemampuan untuk berkomunikasi dan berkerja sama. Hal ini dapat dinilai dari sertifikasi yang sudah didapat <i>personel</i> atau pengalaman kerja tim dalam sebuah proyek.
	LTEX ( <i>Language and Tool Experience</i> )	Pengalaman tim proyek dalam bahasa pemrograman yang digunakan serta memanfaatkan CASE TOOL dalam pengembangan perangkat lunak.
	APEX ( <i>Applications Experience</i> )	Pengalaman kerja tim proyek pada suatu proyek pengembangan aplikasi sistem perangkat lunak atau subsistem.
	PLEX ( <i>Platform Experience</i> )	Penilaian pemahaman tim proyek dalam menggunakan <i>platform</i> , <i>interface database</i> , jaringan, dan <i>middleware</i> .
	PCON ( <i>Personnel Continuity</i> )	Pergantian <i>personel</i> tiap tahun pada proyek. Semakin sedikit pergantian maka semakin tinggi skala.
	PCAP ( <i>Programmer Capability</i> )	Kemampuan <i>programmer</i> dalam sebuah kemampuan, efisiensi, dan ketelitian sebagai sebuah tim.

Tabel 12. Atribut *Effort Multiplier* (Lanjutan)

Kategori	EM	Keterangan
	<i>SITE (Multisite Development)</i>	Cara komunikasi yang digunakan dalam pengembangan perangkat lunak pada proyek.
	<i>SCED (Required Development Schedule)</i>	Penilaian <i>cost driver</i> terkait tingkat persentase dari percepatan atau kemunduran jadwal terhadap jadwal suatu proyek yang telah ditetapkan sebelumnya.

Berikut ini Persamaan yang digunakan untuk menghitung sumber daya yang digunakan dalam pengembangan sistem. Perhitungan usaha (*person month*), ditunjukkan pada Persamaan 3.

$$PM = A \cdot (Size)^E \prod_{i=1}^{i=17} EM$$

(3)

Keterangan:

PM : *Person-Month*

A : Nilai koefisien 2.94 (Pada Cocomo II)

*Size* : Nilai KSLOC

E : Nilai faktor eksponen (*Scala Factor*)

EM : *Effort Multiplier*

Tingkat penilai *effort multiplier* ditunjukkan pada Tabel 13 (Milicic, 2004).

Tabel 13. Tingkat penilai *effort multiplier*

Kategori	EM	Keterangan	Kriteria Penilaian					
			<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
<i>Product</i>	RELY	Terkait sejauh mana perangkat lunak menjalankan aplikasi sesuai dengan fungsinya selama periode waktu.	Jika terjadi kegagalan hanya berdampak kepada ketidak nyamanan pada pengguna.	Jika terjadi kegagalan rendah, kerugian dapat diperoleh kembali	Jika terjadi kegagalan seang, kerugian dapat diperoleh kembali.	Jika terjadi kegagalan yang merugikan dalam finansial.	Jika terjadi kegagalan yang berisiko dalam kehidupan manusia.	n/a
	DATA	Ukuran <i>database</i> yang digunakan.	n/a	Jika DB bytes/pgm SLOC<10	Jika DB bytes/pgm SLOC 10 ≤ D/P <100	Jika DB bytes/pgm SLOC 100 ≤ D/P <1000	Jika DB bytes/pgm SLOC D/P ≥ 1000	n/a

Tabel 13. Tingkat penilai *effort multiplier* (Lanjutan)

Kategori	EM	Keterangan	Kriteria Penilaian					
			<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
	CPLX	Perangkat lunak dan perangkat keras dalam melakukan tugasnya seperti <i>platform</i> (arsitektur, sistem operasi, bahasa pemrograman dan antarmuka yang terkait), sistem manajemen <i>database, browser</i> yang sesuai digunakan dalam menjalankan aplikasi ini.	Perangkat lunak dan perangkat keras tidak berjalan dengan baik.	Terjadi kesalahan atau <i>error</i> dalam menjalankan perangkat lunak dan perangkat keras.	Terjadi kesalahan atau <i>error</i> dalam menjalankan perangkat lunak.	Perangkat lunak berjalan dengan baik sebagaimana mestinya.	Perangkat lunak berjalan dengan baik dan tidak ada kesalahan atau <i>error</i> sama sekali.	Perangkat lunak dijalankan dengan baik dan tidak ada kesalahan atau <i>error</i> sama sekali, dan saat pengujian tidak ada <i>error</i> .

Tabel 13. Tingkat penilai *effort multiplier* (Lanjutan)

Kategori	EM	Keterangan	Kriteria Penilaian					
			<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
	RUSE	<i>Cost driver</i> terkait tingkat upaya yang diperlukan untuk mengembangkan komponen yang dimaksudkan untuk digunakan kembali pada proyek-proyek yang sedang berjalan atau proyek di masa mendatang.	n/a	<i>Cost driver</i> tidak digunakan dalam proyek, program, dan produk lain.	<i>Cost driver</i> akan digunakan dalam proyek lain.	<i>Cost driver</i> akan digunakan dalam program lain.	<i>Cost driver</i> akan digunakan dalam produk lain.	<i>Cost driver</i> akan digunakan dalam produk ini maupun produk lain.
	DOCU	Kesesuaian dokumentasi proyek terhadap kebutuhan siklus hidup perangkat lunak.	Hanya beberapa siklus hidup perangkat lunak yang didokumentasikan.	Hanya sebagian kebutuhan siklus hidup perangkat lunak yang didokumentasikan.	Dokumentasi sesuai dengan siklus hidup perangkat lunak.	Berlebihan dalam dokumentasi siklus hidup perangkat lunak.	Sangat berlebihan dalam dokumentasi siklus perangkat lunak.	n/a
<i>Platfrom</i>	TIME	Persentase kendala waktu eksekusi yang diharapkan dapat digunakan pada sistem perangkat lunak.	n/a	n/a	50% dari waktu yang telah tersedia.	75% dari waktu yang telah tersedia.	85% dari waktu yang telah tersedia.	95% dari waktu yang telah tersedia.

Tabel 13. Tingkat penilai *effort multiplier* ( Lanjutan)

Kategori	EM	Keterangan	Kriteria Penilaian					
			<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
	STOR	Persentase tingkat kendala penyimpanan utama yang digunakan pada sistem perangkat lunak.	n/a	n/a	50% dari waktu yang telah tersedia.	75% dari waktu yang telah tersedia.	85% dari waktu yang telah tersedia.	95% dari waktu yang telah tersedia.
	PVOL	Perubahan yang terjadi pada <i>hardware</i> dan <i>software</i> dalam kurun waktu tertentu.	n/a	Jika terjadi perubahan besar dalam waktu 12 bulan, dan perubahan kecil dalam waktu 1 bulan.	Jika terjadi perubahan besar dalam waktu 6 bulan, dan perubahan kecil dalam waktu 2 minggu.	Jika terjadi perubahan besar dalam waktu 6 bulan, dan perubahan kecil dalam waktu 1 minggu.	Jika terjadi perubahan besar dalam waktu 2 minggu, dan perubahan kecil dalam 1 hari.	n/a

Tabel 13. Tingkat penilai *effort multiplier* (Lanjutan)

Kategori	EM	Keterangan	Kriteria Penilaian					
			<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
<i>Personnel</i>	ACAP	Kemampuan personel dalam analisis dan desain, secara efisien dan teliti, serta kemampuan untuk berkomunikasi dan berkerja sama. Hal ini dapat dinilai dari sertifikasi yang sudah didapat personel atau pengalaman kerja tim dalam sebuah proyek.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 2 kali.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 4 kali.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 6 kali.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 8 kali.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 9 kali.	n/a
	LTEX	Pengalaman tim proyek dalam bahasa pemrograman yang digunakan serta memanfaatkan CASE TOOL dalam pengembangan perangkat lunak.	2 Bulan.	6 Bulan.	1 Tahun.	3 Tahun.	6 Tahun.	n/a

Tabel 13. Tingkat penilai *effort multiplier* (Lanjutan)

Kategori	EM	Keterangan	Kriteria Penilaian					
			<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
	APEX	Pengalaman kerja tim proyek pada suatu proyek pengembangan aplikasi sistem perangkat lunak atau subsistem.	2 Bulan.	6 Bulan.	1 Tahun.	3 Tahun.	6 Tahun.	n/a
	PLEX	Penilaian pemahaman tim proyek dalam menggunakan <i>platform, interface database, jaringan, middleware</i> .	2 Bulan.	6 Bulan.	1 Tahun.	3 Tahun.	6 Tahun.	n/a
	PCON	Pergantian personel tiap tahun pada proyek. Semakin sedikit pergantian maka semakin tinggi skala.	48% pertahun.	24% pertahun.	12% pertahun.	8% pertahun.	3% pertahun.	n/a

Tabel 13. Tingkat penilai *effort multiplier* (Lanjutan)

Kategori	EM	Keterangan	Kriteria Penilaian					
			<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
	PCAP	Kemampuan <i>programmer</i> dalam efisiensi penulisan kode program, ketelitian, dan kemampuan untuk berkomunikasi dan berkerja sama sebagai tim.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 2 kali.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 4 kali.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 6 kali.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 8 kali.	10 proyek terakhir, analisis proyek ini pernah berpartisipasi <i>minimal</i> 9 kali.	n/a
<i>Project</i>	TOOL	Tool merupakan penilaian penggunaan <i>cost driver</i> terkait penggunaan CASE TOOL dalam pengembangan perangkat lunak pada proyek, seperti dari mengubah kode yang sederhana menjadi terintegrasi.	Edit, kode, debug, komunikasi melalui telepon, dan <i>email</i> .	Sederhana <i>backend case</i> , integrasi dalam hal kecil dan komunikasi melalui telepon maupun fax.	<i>Basic lifecycle tools</i> , integrasi yang cukup, dan komunikasi melalui <i>email</i> pribadi.	Kuat, <i>mature lifecycle tools</i> , integrasi yang cukup, dan komunikasi melali <i>wide band cummunication (smartphone)</i> .	Kuat, matang, integrasi yang cukup, komunikasi melalui <i>wide band cummunication (smartphone)</i> , dan <i>video conference</i> .	n/a

Tabel 13. Tingkat penilai *effort multiplier* (Lanjutan)

Kategori	EM	Keterangan	Kriteria Penilaian					
			<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
	SITE	Cara komunikasi yang digunakan dalam pengembangan perangkat lunak pada proyek.	Komunikasi yang digunakan melalui telepon, dan <i>email</i> .	Komunikasi yang digunakan melalui telepon, dan fax.	Komunikasi yang digunakan melalui <i>email</i> pribadi.	Komunikasi yang digunakan melalui <i>wide band cummunication (smartphone)</i> .	Komunikasi yang digunakan melalui <i>wide band cummunication (smartphone)</i> , dan <i>video conference</i> .	Komunikasi yang digunakan melalui <i>interactive multimedia</i> .
	SCED	Penilaian <i>cost driver</i> terkait tingkat persentase dari percepatan atau kemunduran jadwal terhadap jadwal suatu proyek yang telah ditetapkan sebelumnya.	Kompresi 75% dari jadwal yang telah ditetapkan.	Kompresi 85% dari jadwal yang telah ditetapkan.	Kompresi 100% dari jadwal yang telah ditetapkan.	Kompresi 130% dari jadwal yang telah ditetapkan.	Kompresi 160% dari jadwal yang telah ditetapkan.	n/a

Setiap *effort multiplier* memiliki bobot penilaian yang telah ditentukan (Milicic, 2004) ditunjukkan pada Tabel 14.

Tabel 14. Penilaian *Effort Multiplier*

Kategori	EM	<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
<i>Product</i>	RELY	<i>slight inconvenience</i>	<i>low, easily recoverable losses</i>	<i>moderate, easily recoverable losses</i>	<i>high financial loss</i>	<i>risk to human life</i>	n/a
		0.82	0.92	1.00	1.10	1.26	n/a
	DATA	n/a	<i>Testing DB bytes/pgm</i>	<i>Testing DB bytes/pgm</i>	<i>Testing DB bytes/pgm</i>	<i>Testing DB bytes/pgm</i>	n/a
			SLOC <10	SLOC $10 \leq D/P < 100$	SLOC $100 \leq D/P < 1000$	SLOC $D/P \geq 1000$	
		n/a	0.90	1.00	1.14	1.28	n/a
	CPLX	0.73	0.87	1.00	1.17	1.34	1.74
	RUSE	n/a	0.95	1.00	1.07	1.15	1.25
	DOCU	<i>Many lifecycle needs uncovered</i>	<i>Some lifecycle needs uncovered</i>	<i>Right sized to lifecycle needs</i>	<i>Excessive for life cycle needs</i>	<i>Very excessive for life cycle needs</i>	n/a
		0.81	0.91	1.00	1.11	1.23	n/a
<i>Platform</i>	TIME	n/a	n/a	$\leq 50\%$ use of execution time	70% use of execution time	85% use of execution time	95% use of execution time
		n/a	n/a	1.00	1.11	1.29	1.63

Tabel 14. Penilaian *Effort Multiplier* (Lanjutan)

Kategori	EM	<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
	STOR	n/a	n/a	$\leq 50\%$ use of available storage	70% use of available storage	85% use of available storage	95% use of available storage
		n/a	n/a	1.00	1.05	1.17	1.46
	PVOL	n/a	Major change every 12 month ; Minor change every 1 month	Major: 6 month ; Minor: 2 week	Major: 2 month ; Minor 1 week	Major: 2week ; Minor 1 days	n/a
<i>Personnel</i>		n/a	0.87	1.00	1.15	1.30	n/a
	ACAP	15th Percentile	35th Percentile	55th Percentile	75th Percentile	90th Percentile	n/a
		1.42	1.19	1.00	0.85	0.71	n/a
	LTEX	$\leq 2$ month	6 month	1 year	3 year	6 years	n/a
		1.20	1.09	1.00	0.91	0.84	n/a
	APEX	$\leq 2$ month	6 month	1 year	3 year	6 years	n/a
		1.22	1.10	1.00	0.88	0.81	n/a
	PLEX	$\leq 2$ month	6 month	1 year	3 year	6 years	n/a
		1.19	1.09	1.00	0.91	0.85	n/a
	PCON	48% /year	24% /year	12% /year	6% /year	3% /year	n/a
	1.29	1.12	1.00	0.90	0.81	n/a	
	PCAP	15th Percentile	35th Percentile	55th Percentile	75th Percentile	90th Percentile	n/a

Tabel 14. Penilaian *Effort Multiplier* (Lanjutan)

Kategori	EM	<i>Very Low</i>	<i>Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very High</i>	<i>Extra High</i>
		1.34	1.15	1.00	0.88	0.76..	n/a
<i>Project</i>	TOOL	<i>edit, code, debug</i>	<i>simple, fronted, backend case, little integration</i>	<i>basic lifecycle tools, moderately integrated</i>	<i>strong, mature lifecycle tools, moderately integrated</i>	<i>strong, mature, proactive life cycle tools, well integrated with processes, methods, reuse</i>	n/a
	SITE	1.17 <i>International</i>	1.09 <i>Multicity and multicompany</i>	1.00 <i>Multicity or multicompany</i>	0.90 <i>Same city or netro area</i>	0.78 <i>Same building or complex</i>	n/a <i>Fully collocated</i>
	SCED	1.22 <i>75% of nominal</i>	1.09 <i>85% of nominal</i>	1.00 <i>100% of nominal</i>	0.93 <i>130% of nominal</i>	0.86 <i>160% of nominal</i>	0.80 n/a
		1.43	1.14	1.00	1.00	1.00	n/a

Setiap nilai parameter yang didapatkan dari pengisian kuesioner nilai *effort multiplier* akan dijumlahkan dan dibagi sebanyak parameter *effort multiplier*, lalu hasilnya nanti digunakan untuk perhitungan nilai *effort estimation*.

## 2.10 Effort Estimation

*Effort Estimation* didapatkan dari 22 atribut parameter yang ada dalam model Cocomo II yang terdiri 5 atribut parameter yang diperoleh dari *scala factor*, dan 17 atribut yang diperoleh dari *effort multiplier* yang digunakan untuk menghitung estimasi biaya, waktu, dan sumber daya manusia (Maqдум dkk, 2019).

### a. Perhitungan estimasi waktu

Perhitungan estimasi waktu sering juga disebut TDEV (*Time Development*) atau waktu (*bulan*) yang digunakan untuk estimasi penyelesaian pengembangan perangkat lunak ditunjukkan pada Persamaan 4.

$$TDEV = C \times (PM)^{(D+0,2 \times (E-B))} \quad (4)$$

Keterangan:

TDEV : Waktu (Bulan)

PM : *Person Month*

C : 3,67 (Pada Cocomo II)

D : 0,28 (Pada Cocomo II)

B : 0,91 (Pada Cocomo II)

E : Nilai faktor eksponen (*scala factor*)

### b. Perhitungan Orang (*staff*)

Perhitungan jumlah orang (*staff*) untuk melakukan pengembangan perangkat lunak dapat menggunakan Persamaan 5.

$$Average\ Staff = \frac{PM}{TDEV} \quad (5)$$

Keterangan:

*Average staff* : Jumlah orang yang dibutuhkan

PM : *Person Month*

TDEV : Waktu (Bulan)

c. Perhitungan Biaya per-Bulan

Perhitungan biaya estimasi perbulan dapat dilakukan dengan perhitungan menggunakan Persamaan 6.

$$\text{Biaya perbulan} = \text{Average staff} \times \text{UMR} \quad (6)$$

Keterangan:

Biaya perbulan : Biaya yang harus dikeluarkan setiap bulan

*Average staff* : Jumlah orang yang dibutuhkan

UMR : Upah minimum regional daerah Lampung

(Rp. 2.432.001 Menurut Surat Keputusan Gubernur Lampung Nomor: G/483/V.08/HK/2020)

d. Perhitungan Biaya Total

Perhitungan biaya total ada biaya keseluruhan yang dikeluarkan dalam pengembangan perangkat lunak menggunakan Persamaan 7.

$$\text{Biaya total} = \text{Biaya perbulan} \times \text{TDEV} \quad (7)$$

Keterangan:

TDEV : Waktu (Bulan)



### 3.2 Alat dan Data Penelitian

Alat dan data penelitian yang dibutuhkan untuk melakukan penelitian sebagai berikut:

#### a. Alat Penelitian

Penelitian ini dilakukan dengan menggunakan alat untuk mendukung dan menunjang pelaksanaan penelitian, yaitu sebagai berikut:

##### 1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan adalah satu unit laptop dengan spesifikasi *Processor* Intel® Celeron®, CPU N3060 @ 1.60 GHz 1.60Ghz, dan RAM 4GB.

##### 2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

2.1 Windows 10, digunakan untuk menjalankan aplikasi yang dibutuhkan.

2.2 Google Chorme, digunakan untuk mencari studi literatur.

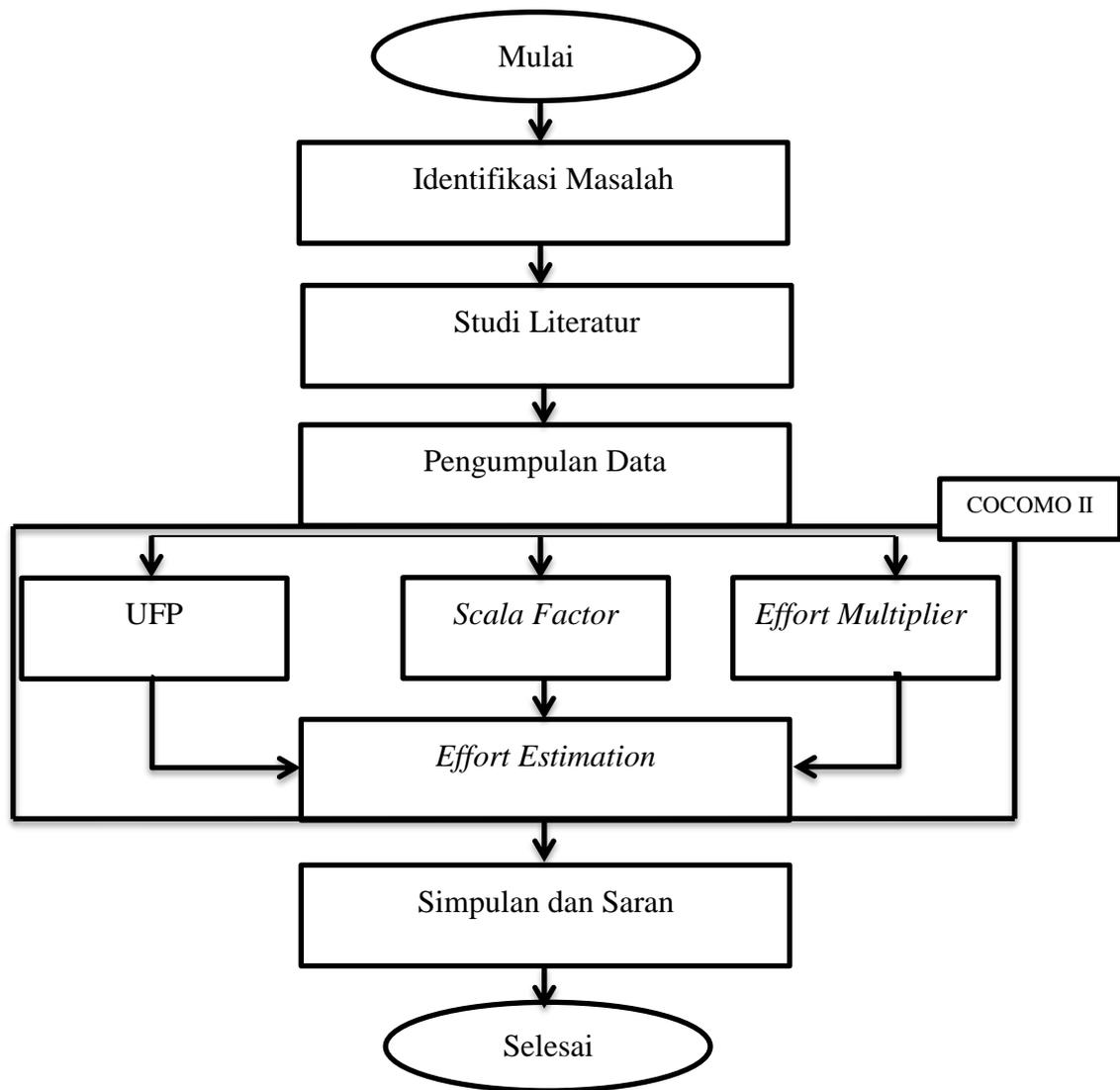
2.3 *Microsoft Excel*, digunakan untuk mengelolah data.

#### b. Data Penelitian

Data yang digunakan untuk melakukan penelitian ini adalah nilai *Unadjusted Function Point* (UFP) Sistem Informasi KKN Universitas Lampung yang didapatkan dari penelitian (Anggraeni, 2021), serta pengisian kuesioner yang diisi oleh pengembang (Putra, 2019).

### 3.3 Tahapan Penelitian

Tahapan penelitian dirancang untuk pengukuran estimasi biaya dan waktu pengembangan Sistem Informasi Kuliah Kerja Nyata Universitas Lampung menggunakan model Cocomo II. Penelitian dimulai dari identifikasi masalah, sampai dengan analisis dan pengujian data yang ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Gambar 1 menunjukkan tahapan-tahapan yang dilalui dalam penelitian. Berikut ini adalah penjabaran dari masing-masing tahapan penelitian yang dilakukan sebagai berikut:

1. Identifikasi Masalah

Tahapan ini adalah salah satu tahapan yang paling penting untuk memecahkan masalah dan memberikan solusi alternatif. Penelitian ini menganalisis masalah-masalah yang ada seperti dalam pengembangan perangkat lunak dan sering terjadi permasalahan estimasi perkiraan waktu pengerjaan pembuatan perangkat lunak. Selain itu, terdapat juga masalah terkait total biaya yang harus dikeluarkan untuk membuat perangkat lunak, serta berapa banyak sumber daya manusia yang dibutuhkan dalam pembuatan perangkat lunak.

2. Studi Literatur

Tahapan penelitian ini adalah studi literatur tahapan ini digunakan untuk pengumpulan data, jurnal, dan artikel terkait dengan model Cocomo II dan teori-teori serta konsep yang akan diteliti dari berbagai sumber.

3. Pengumpulan Data

Tahapan ini merupakan tahapan pengumpulan data yang digunakan untuk menganalisis estimasi perkiraan biaya, waktu pengembangan, dan sumber daya manusia Sistem Informasi Kuliah Kerja Nyata Universitas Lampung menggunakan model Cocomo II yaitu dengan cara melakukan pengisian kuesioner kepada tim pengembang untuk mendapatkan nilai *scala factor* dan *effort multiplier* nantinya data yang dikumpulkan akan digunakan untuk menghitung *effort estimation*.

4. *Unadjusted Function Point (UFP)*

Tahapan yang digunakan untuk mencari nilai UFP yang telah ditentukan oleh *Function Point Internasional User Group (IFPUG)*. Tahapan penelitian ini didapatkan dari penelitian (Anggraeni, 2021). Setelah nilai UFP sudah diketahui nilai UFP tersebut akan diubah ke SLOC berdasarkan QSM selanjutnya nilai SLOC diubah ke KSLOC dengan cara dibagi dengan angka 1000, kemudian akan dihasilkan nilai *size* untuk digunakan dalam persamaan *effort multiplier*.

5. *Scala Factor*

Tahapan ini digunakan untuk mendapatkan nilai *scala factor* dengan menentukan 5 *scala factor* yaitu PREC, FLEX, RESL, TEAM, dan PMAT.

Tahapan ini yang dikumpulkan secara langsung dengan cara pengisian kuesioner oleh (Putra, 2019) yang dilakukan secara *offline* maupun *online* melalui *Goggle Sheets* mengenai Sistem Informasi Kuliah Kerja Nyata Universitas Lampung. Setelah didapatkan hasilnya lalu dimasukkan ke dalam Persamaan 1 untuk menentukan faktor eksponen.

6. *Effort Multiplier*

Tahapan ini digunakan untuk mendapatkan nilai yang dijabarkan 17 *effort multiplier*. Tahapan *effort multiplier* sama dengan tahapan *scala factor* yang dikumpulkan secara langsung dengan cara pengisian kuesioner oleh (Putra, 2019) yang dilakukan secara *offline* maupun *online* melalui *Goggle Sheets* mengenai Sistem Informasi Kuliah Kerja Nyata Universitas Lampung. Setelah didapatkan hasilnya lalu dimasukkan ke dalam Persamaan 3 perhitungan usaha (*person month*).

7. *Effort Estimation*

Tahapan ini digunakan untuk menghitung estimasi biaya, waktu, dan sumber daya manusia dalam pengembangan sistem informasi. Setelah dilakukan perhitungan nilai *scala factor* dan *effort multiplier*, lalu dilakukan perhitungan *effort estimation* perhitungan estimasi waktu menggunakan Persamaan 4, perhitungan orang (*staff*) menggunakan Persamaan 5, perhitungan biaya per-bulan menggunakan Persamaan 6, dan Persamaan 7 untuk perhitungan biaya total.

8. Simpulan dan Saran

Tahapan ini digunakan untuk menyimpulkan berapa lama proses pengembangan sistem informasi, berapa biaya yang dibutuhkan, dan sumber daya manusia yang diperlukan untuk mengembangkan Sistem Informasi Kuliah Kerja Nyata Universitas Lampung. menggunakan Cocomo II, serta saran untuk penelitian selanjutnya.

## V. SIMPULAN DAN SARAN

### 5.1 Simpulan

Berdasarkan hasil penelitian yang telah dilakukan, estimasi biaya, waktu pengerjaan, dan sumber daya manusia untuk pengembangan Sistem Informasi Kuliah Kerja Nyata Universitas Lampung, menggunakan Cocomo II dapat disimpulkan memerlukan waktu selama 23 bulan. Estimasi sumber daya manusia yang dibutuhkan adalah 18 orang, dan membutuhkan biaya sebanyak Rp. 43.776.018 setiap bulannya. Biaya total untuk pengembangan sistem 23 bulan sebanyak Rp.1.006.848.414.

Besaran jumlah sumber daya dipengaruhi oleh jumlah fungsi dari *use case* pada Sistem Informasi Kuliah Kerja Nyata Universitas Lampung, *scala factor*, *effort multiplier*, dan *effort estimation*. Jumlah fungsi dari *use case* menentukan seberapa besar proyek yang harus dikerjakan, *scala factor* menentukan tingkat kompleksitas proyek, *effort multiplier* menentukan faktor-faktor lingkungan yang mempengaruhi proyek, dan *effort estimation* menentukan besaran sumber daya manusia yang dibutuhkan.

### 5.2 Saran

Berdasarkan penelitian yang telah dilakukan untuk menghitung estimasi biaya, waktu pengerjaan, sumber daya manusia mengerjakan proyek pengembangan Sistem Informasi Kuliah Kerja Nyata Universitas Lampung, maka dapat diberikan saran sebagai berikut:

1. Penelitian selanjutnya dapat mengembangkan penelitian ini dengan cara menambahkan metode estimasi lainnya sebagai pembandingan dengan metode penelitian ini.

2. Penelitian selanjutnya dapat menambahkan jumlah responden untuk memberikan penilaian pada bagian *scala factor* dan *effort multiplier* Cocomo II, tidak hanya pengembang agar estimasi sumber daya yang didapatkan tingkat validitasnya lebih tinggi dan lebih objektif.

## DAFTAR PUSTAKA

- Aditya, P. M., Rachmadi, A., & Wicaksono, S. A. 2018. Perhitungan Biaya Pembangunan Sistem Informasi Pariwisata Kota Sawahlunto Menggunakan Metode COCOMO II. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, 2(11), 4785–4793.
- Anggraeni, R. D. 2021. Mengukur Nilai Fungsionalitas Sistem Informasi Kuliah Kerja Nyata (E-Kkn) Universitas Lampung Menggunakan Metrik Function Point. Skripsi Mahasiswa S1 Ilmu Komputer. Bandar Lampung: Universitas Lampung.
- Boehm, B. 2000. “COCOMO II Model Definition Manual Version 2.1. s.l”, *University of Southern California*.
- BP-KKN Universitas Lampung. 2018.” Badan Pelaksana Kuliah Kerja Nyata - Universitas Lampung”. Diakses melalui <https://kkn.unila.ac.id>. Pada 20 Desember 2022.
- Damayanti, D., Suprpto., & Perdanakusuma, A. Z. 2017. Analisis Estimasi Biaya Pembuatan Perangkat Lunak Menggunakan Metode COCOMO II di Inagata Technosmith (Studi Kasus : Sistem Informasi Monitoring dan Evaluasi Penerimaan Beasiswa Santri Berprestasi UIN Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, 1(10).
- Fatonah, O. S., & Afrizal, Y. 2013. *Model Estimasi Biaya Perangkat Lunak Menggunakan Cocomo II (Studi Kasus PT. X)*. 40–46.
- Gubernur Lampung. 2019. “Keputusan Gubernur Lampung Nomor: G/776/V.07/HK/2019 Tentang Penetapan Upah Minimum Provinsi (UMP) Lampung Tahun 2020”.

- Khoiro, M. N., Herlambang, A.D., & Saputra, M. C. 2018. *Evaluasi Biaya Pengembangan Perangkat Lunak Dengan Menggunakan Metode Cocomo II (Studi Kasus: PT DOT Indonesia)*. hlm. 3220-3229
- Krismaji. 2015. *Sistem Informasi Akuntansi edisi ketiga*. Yogyakarta: Unit penerbit dan sekolah Tinggi Ilmu YKPN.
- Maqdam, A. N., Perdanakusuma, A. R., & Putra, W. H. N. 2019. *Implementasi Metode COCOMO II untuk Estimasi Biaya Pengembangan Perangkat Lunak di CV . Profile Image Studio*. 3(6).
- Milicic, D. 2004. *Applying COCOMO II*. 1–55. Retrieved from <http://ejournal.unitomo.ac.id/index.php/inform/article/view/215/118>.
- Putra, G. L. 2019. *Sistem Informasi Pelaporan Pelaksanaan KKN dan Profil Desa Berbasis Web Menggunakan Framework Laravel*. Skripsi Mahasiswa S1 Ilmu Komputer. Bandar Lampung: Universitas Lampung.
- Rahensa, I., Fitriasia, Y., & Ananda. 2015. *Pengembangan Objek Oriented Software Metric Tools Untuk Evaluasi Maintainability*. 3<sup>rd</sup> *Applied Business and Engineering Conference*. Batam.
- Rawat, M. S., Mittal, A., & Dubey, S. K. 2012. Survey on Impact of Software Metrics on Software Quality. *International Journal of Advanced Computer Science and Applications*. Vol 3., 137–141.
- Sidabutar, J. C. P. 2015. *Estimasi Biaya Perangkat Lunak Menggunakan Metode COCOMO yang Dimodifikasi*.