



**SISTEM APLIKASI PENENTU ESTIMASI BIAYA PENGEMBANGAN  
PERANGKAT LUNAK MENGGUNAKAN METODE COCOMO II**

**TUGAS AKHIR**



**Program Studi**

**S1 SISTEM INFORMASI**

UNIVERSITAS  
**Dinamika**

**Oleh:**

**Eko Febri Harsono**

**16410100003**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2020**

**SISTEM APLIKASI PENENTU ESTIMASI BIAYA PENGEMBANGAN  
PERANGKAT LUNAK MENGGUNAKAN METODE COCOMO II**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan**

**Program Sarjana Komputer**



UNIVERSITAS  
**Dinamika**

Oleh :

**Nama : Eko Febri Harsono**

**NIM : 16410100003**

**Jurusan : SI Sistem Informasi**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2020**

**Tugas Akhir**

**SISTEM APLIKASI PENENTU ESTIMASI BIAYA PENGEMBANGAN  
PERANGKAT LUNAK MENGGUNAKAN METODE COCOMO II**

Dipersiapkan dan disusun oleh

**Eko Febri Harsono**

**NIM : 16410100003**

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas

Pada : Selasa, 04 Februari 2020

**Susunan Dewan Pembahas**

**Pembimbing:**

I. Dr. Anjik Sukmaaji, S.Kom., M.Eng.

NIDN: 0731057301

II. Arifin Puji Widodo, S.E., MSA

NIDN: 0721026801

**Pembahas:**

Tan Amelia, S.Kom., M.MT

NIDN. 0728017602



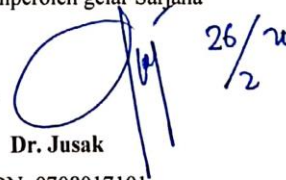
Handwritten signatures of the supervising lecturers and reviewers, dated 25/20.

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana



Fakultas Teknologi dan Informatika  
UNIVERSITAS  
**Dinamika**



Handwritten signature of Dr. Jusak, dated 26/20.

**Dr. Jusak**

NIDN: 0708017101

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

## PERNYATAAN

### PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika Surabaya, saya:

Nama : Eko Febri Harsono  
NIM : 16410100003  
Program Studi : S1 Sistem Informasi  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Tugas Akhir  
Judul Karya : **SISTEM APLIKASI PENENTU ESTIMASI  
BIAYA PENGEMBANGAN PERANGKAT LUNAK  
MENGUNAKAN METODE COCOMO II**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 04 Februari 2020

Yang menyatakan

  
**Eko Febri Harsono**  
NIM: 16410100003

*“My self it’s not my priority for few moments”*



UNIVERSITAS  
**Dinamika**



*Kupersembahkan hasil karya ini untuk  
Ibu yang selalu mendoakanku,  
Bapak yang selalu menasihati dan mendukungku,  
Para sahabat yang selalu menyemangati,  
Dan untuk semua,  
Terima kasih*

UNIVERSITAS  
Dinamika

## ABSTRAK

Estimasi biaya proyek perangkat lunak itu sangat penting bagi pihak pemerintahan sebelum proyek ditenderkan. pihak pemerintahan harus menentukan sebuah HPS (Harga Perkiraan Sendiri) dengan cara survei harga proyek pada tiap daerah. Dikarenakan tidak adanya peraturan khusus terkait HPS itu sendiri pada peraturan presiden. Perpres No.70 tahun 2012 menjelaskan bawah HPS hanya untuk pengadaan barang atau jasa dan tidak ada peraturan khusus terkait HPS untuk proyek perangkat lunak. Oleh karna itu pihak pemerintahan kesusahan dalam menentukan estimasi nilai total proyek pengembangan perangkat lunak. Dalam menghitung proyek pengembangan perangkat lunak ada beberapa perhitungan yang harus dihitung seperti lama pengerjakan proyek, jumlah orang dalam pengerjaan proyek. Untuk menghitung variabel pengembangan proyek perangkat lunak tersebut dapat digunakan pendekatan dengan metode COCOMO dan dengan parameter DFD (*Data Flow Diagram*) sebagai acuan perhitungan. Apalagi saat ini sudah dikembangkan metode COCOMO untuk Effort Multipliers yaitu metode COCOMO II.oleh karena itu metode COCOMO II sangat cocok untuk digunakan menyelesaikan masalah tersebut dan membangun sebuah aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak menggunakan metode COCOMO II. Aplikasi yang dibangun dapat menentukan estimasi nilai total proyek pengembangan perangkat lunak sebelum proyek tersebut ditenderkan dengan menggunakan metode COCOMO II. Sehingga pihak pemerintahan mudah untuk menentukan HPS pada tiap proyek pengembangan perangkat lunak.

**Kata Kunci:** COCOMO II, HPS, estimasi biaya, proyek perangkat lunak.

## KATA PENGANTAR

Puji dan rasa syukur penulis panjatkan kepada Tuhan yang Maha Esa, karena dengan rahmat, karunia, ridho dan hidayah-Nya, penulis dapat menyelesaikan laporan tugas akhir dengan topik yang diambil yaitu penilaian kinerja. “Rancang Bangun Aplikasi Penentu Estimasi Biaya Pengembangan Perangkat Lunak Menggunakan Metode COCOMO II Berbasis *Website*” merupakan salah satu judul tugas akhir yang digunakan untuk menyelesaikan program sarjana yang wajib ditempuh oleh setiap mahasiswa sebagai syarat kelulusan.

Dalam penyusunan laporan tugas akhir, penulis ingin berterimakasih kepada banyak pihak yang membantu dalam menyelesaikan laporan ini. Karena laporan ini dapat terselesaikan berkat kritik, saran, bantuan, hiburan dan motivasi yang telah diberikan kepada penulis, untuk itu penulis berterima kasih kepada:

1. Orang tua dan kakak yang selalu memberikan doa, bantuan, nasihat dan dukungan moral kepada penulis.
2. Bapak Prof. Dr. Budi Jatmiko, M.Pd. selaku Rektor Universitas DInamika.
3. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika.
4. Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng. Ketua Program Studi S1 Sistem Informasi.
5. Bapak Arifin Puji Widodo, S.E., MSA dan Bapak Sholiq selaku dosen pembimbing yang telah memberikan arahan, semangat dan dukungan dalam proses penyelesaian tugas akhir.
6. Teman-teman angkatan 2016 seperjuangan yang membantu, memberi dukungan, dan saran dari proses penyelesaian laporan tugas akhir ini.



Penulis menyadari bahwa laporan tugas akhir ini masih banyak kesalahan dan jauh dari kata sempurna. Untuk itu penulis sangat mengharapkan kritik dan saran dari semua pihak, sehingga kedepannya laporan ini dapat berguna lebih baik bagi para pembaca.

Surabaya, 04 Februari 2020

Penulis



UNIVERSITAS  
**Dinamika**

## DAFTAR ISI

	Halaman
<b>PERNYATAAN</b> .....	iv
<b>ABSTRAK</b> .....	vii
<b>KATA PENGANTAR</b> .....	vii
<b>DAFTAR ISI</b> .....	ix
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR LAMPIRAN</b> .....	xiii
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan .....	4
1.5 Manfaat .....	5
<b>BAB II LANDASAN TEORI</b> .....	6
2.1 COCOMO II .....	6
2.2 SCRUM .....	8
<b>BAB III METODOLOGI PENELITIAN</b> .....	12
3.1 Gambaran Umum Metodologi .....	12
3.2 Studi Literatur .....	13
3.3 Identifikasi Permasalahan .....	15
3.4 Desain Konsep Aplikasi.....	15
3.5 Pengembangan Perangkat Lunak.....	21
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	24
4.1 Hasil <i>Sprint</i> .....	24
4.2 Pembahasan <i>Result</i> Perhitungan COCOMO II.....	41

4.3 Evaluasi.....	47
<b>BAB V PENUTUP</b> .....	49
5.1 Kesimpulan .....	49
5.1 Saran .....	49
<b>DAFTAR PUSTAKA</b> .....	51



UNIVERSITAS  
**Dinamika**

## DAFTAR TABEL

	Halaman
Tabel 3. 1 Desain <i>Testing</i> Perhitungan Modul.....	20
Tabel 3. 2 Rancangan Pengujian Perhitungan <i>Modul</i> .....	20
Tabel 3. 3 <i>Product Backlog Item</i> .....	21
Tabel 3. 4 Tingkat Kesulitan.....	22
Tabel 3. 5 <i>Sprint Backlog</i> Iterasi Pertama.....	23
Tabel 3. 6 <i>Sprint Backlog</i> Iterasi Kedua .....	23
Tabel 3. 7 <i>Sprint Backlog</i> Iterasi Ketiga .....	23
Tabel 3. 8 <i>Sprint Backlog</i> Iterasi Keempat .....	23
Tabel 4. 1 <i>Flow of Event</i> CRUD Proyek.....	25
Tabel 4. 2 <i>Flow of Event</i> CRUD Modul .....	27
Tabel 4. 3 Tabel Sprint Review Iterasi Pertama .....	30
Tabel 4. 4 <i>Flow of Event</i> Otomatisasi Rumus COCOMO II .....	33
Tabel 4. 5 Tabel <i>Sprint Review</i> Iterasi Kedua.....	36
Tabel 4. 6 <i>Flow of Event Report Project</i> .....	38
Tabel 4. 7 Tabel <i>Sprint Review</i> Iterasi Ketiga.....	40
Tabel 4. 8 Tabel <i>Sprint Review</i> Iterasi Keempat.....	41
Tabel 4. 9 Dataset Pemerintahan.....	41
Tabel 4. 10 <i>Result input-an parameter</i> UFP .....	42
Tabel 4. 11 <i>Result</i> nilai total UFP .....	42
Tabel 4. 12 <i>Result</i> nilai total KSLOC .....	43
Tabel 4. 13 <i>Result</i> Nilai <i>Scale Factor</i> .....	43
Tabel 4. 14 <i>Result</i> Nilai <i>Effort</i> .....	43
Tabel 4. 15 <i>Result</i> Input dan Total <i>Effort Multiplier</i> .....	44
Tabel 4. 16 <i>Result</i> total PM.....	44
Tabel 4. 17 <i>Result</i> Input dan Total <i>Activity</i> .....	45
Tabel 4. 18 <i>Salary</i> .....	46
Tabel 4. 19 <i>Personel Direct Cost</i> .....	46
Tabel 4. 20 <i>Personnel With Profit</i> .....	47
Tabel 4. 21 <i>Owner Estimate Cost</i> .....	47

Tabel 4. 23 *Flow of Event Register* ..... 59  
Tabel 4. 24 *Cost per Activities* ..... 72



UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

	Halaman
Gambar 3. 1 Sistem Estimasi Biaya Penentu HPS berbasis SCRUM. ....	12
Gambar 3. 2 Diagram Estimasi Biaya Berbasis COCOMO II .....	16
Gambar 3. 3 UI <i>Login</i> .....	18
Gambar 3. 4 UI <i>Register</i> .....	18
Gambar 3. 5 UI Mengelola Proyek dan Modul .....	19
Gambar 3. 6 UI Perhitungan COCOMO II.....	19
Gambar 4. 1 <i>Usecase System</i> Iterasi Pertama.....	24
Gambar 4. 28 Perhitungan Langkah Kesepuluh.....	68
Gambar 4. 36 Tampilan <i>Save History Project</i> .....	70



UNIVERSITAS  
**Dinamika**

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Desain <i>Testing</i> .....	53
Lampiran 2 <i>Activity Diagram, Sequence</i> dan Hasil Desain I/O Iterasi Pertama..	58
Lampiran 3 Hasil Desain I/O Iterasi Kedua.....	62
Lampiran 4 <i>Activity</i> Hasil Desain I/O Iterasi Ketiga.....	69
Lampiran 5 Kebutuhan <i>Hardware</i> dan <i>Software</i> .....	71
Lampiran 6 <i>Cost Per Activity</i> .....	72
Lampiran 7 <i>Direct Cost Non Personnel</i> .....	73
Lampiran 8 Biodata Penulis .....	74



UNIVERSITAS  
**Dinamika**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam proyek pemerintahan di Indonesia, sebelum proyek ditenderkan atau diadakan ke pihak ketiga, perlu mengetahui perkiraan harga sebagai pedoman. Perkiraan harga ini disebut *Owner Estimate Cost* (OEC). Pekerjaan menentukan OEC dilakukan oleh *Committed Officers* (CO) (Admin 2019). Selain menentukan spesifikasi teknis barang / jasa yang akan ditenderkan dan menyusun kontrak. Proyek pengembangan perangkat lunak juga dapat dikategorikan sebagai pengadaan barang atau jasa sesuai dengan peraturan presiden nomor 54 pada 2010 dan nomor 70 pada 2012 (Perpres, 2012).

Setiap pelaksanaan pengadaan barang atau jasa termasuk pengadaan perangkat lunak perlu dibuatkan harga perkiraan sendiri (HPS) atau bisa disebut sebagai estimasi harga suatu pekerjaan (barang/jasa) yang akan diadakan (Barang/Jasa, 2019). Ada maksud dan tujuan disusunnya HPS adalah supaya harga atau nilai proyek tersebut dalam batas kewajaran dan untuk besaran tambahan nilai jaminan pelaksanaan bagi penawar yang dinilai terlalu rendah.

HPS merupakan total harga yang diperkirakan cukup untuk membiayai pekerjaan yang akan dilaksanakan dalam pengadaan barang atau jasa yang ditetapkan CO. Dalam penyusunannya, HPS didasarkan pada data harga setempat, yang didapatkan dari hasil survei menjelang dilaksanakan pengadaan, dengan mempertimbangkan informasi yang ada. Jika penentuan HPS terjadi kesalahan maka, pelaksanaannya tidak akan berjalan dengan baik sehingga menyebabkan



kegagalan proyek. Kesalahan dalam menentukan HPS sering terjadi karena adanya aktivitas yang diluar dugaan atau belum dimasukkan kedalam komponen penentuan HPS. Dimana aktivitas tersebut sangat berpengaruh terhadap nilai HPS akibatnya ketika barang sudah diperoleh, barang tidak bisa dioperasikan karena HPS yang disusun tidak mencakup kegiatan yang mendukung beroperasinya barang tersebut. Oleh karena itu, CO harus menyusun HPS berdasarkan sumber data yang valid sehingga menghasilkan nilai yang bisa dipertanggungjawabkan, serta dapat menjadi acuan dalam penentuan HPS untuk pengadaan barang atau jasa khususnya pengadaan perangkat lunak (Purwanto, 2017).

Penentuan HPS untuk proyek perangkat lunak selama ini masih berpedoman pada Perpres No 70. Tahun 2012, dimana hanya menjelaskan HPS untuk pengadaan barang, atau pengadaan jasa lainnya. Didalam Perpres tersebut tidak menjelaskan mengenai penentuan HPS untuk proyek perangkat lunak secara khusus. Sehingga dalam hal ini pihak pemerintah kesulitan dalam menentukan HPS proyek perangkat lunak. dalam studi literatur, penelitian menemukan beberapa hasil penelitian mengenai penentuan HPS untuk proyek perangkat lunak, akan tetapi output yang didapat hanya sampai pada pembuatan model saja (Purwanto, 2017).

Terdapat beberapa teknik dan model estimasi biaya pengembangan perangkat lunak yang diklarifikasikan menjadi algoritma dan non algoritma. Masing – masing mempunyai kelebihan dan kekurangan, tetapi tidak ada satupun teknik yang terbaik untuk semua kondisi pengembangan perangkat lunak. Dalam menghitung proyek pengembangan perangkat lunak ada beberapa perhitungan yang harus dihitung seperti lama pengerjaan proyek, jumlah orang dalam pengerjaan proyek. Untuk menghitung variabel pengembangan proyek perangkat lunak tersebut dapat

digunakan pendekatan dengan metode COCOMO (*Constructive Cost Model*) dan dengan parameter DFD (*Data Flow Diagram*) sebagai acuan perhitungan. Metode COCOMO adalah salah satu metode algoritma yang paling populer dalam memperkirakan biaya pengembangan perangkat lunak. Metode COCOMO dipilih karena terbukti menjadi model yang fleksibel dan memiliki nilai akurasi yang tepat untuk digunakan dalam estimasi biaya. COCOMO sering digunakan karena cocok untuk mengerjakan proyek dengan *scope* yang besar maupun *scope* yang kecil. Apalagi saat ini metode COCOMO sudah dikembangkan menjadi COCOMO II dengan penambahan bobot pada *effort multiplier*. COCOMO II juga terdapat 3 submodel perhitungan, submodel ini dibedakan sesuai dengan fase atau tingkat kesulitan dalam mengembangkan perangkat lunak. Ketiga submodel tersebut yaitu, *early design*, *application composition*, dan *post architecture*. Dan yang paling sesuai dengan rumusan permasalahan tersebut adalah submodel *post-architecture* karena model ini dapat mengitung estimasi biaya proyek perangkat lunak sebagai acuan HPS sebelum ditenderkan secara detail (Khoiro, 2018).

Berdasarkan permasalahan diatas maka peneliti mengajukan pembuatan aplikasi penentu estimasi biaya pengembangan perangkat lunak berbasis *website* khususnya untuk proyek pengembangan perangkat lunak dengan menggunakan metode COCOMO II. Tujuan pembuatan aplikasi berbasis *website* adalah agar aplikasi dapat diakses atau digunakan oleh seluruh pihak pemerintah, baik pemerintahan daerah ataupun pemerintahan pusat.

Hasil akhir yang diharapkan dari penelitian tugas akhir ini berupa aplikasi penentu estimasi biaya pengembangan perangkat lunak yang nantinya dapat

membantu CO dalam menetapkan HPS untuk proyek pemerintah dimasa yang akan datang.

### 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang permasalahan diatas, maka dapat dirumuskan permasalahan yang ada, yaitu:

Bagaimana merancang bangun sistem aplikasi penentu estimasi biaya pengembangan perangkat lunak menggunakan metode COCOMO II?

### 1.3 Batasan Masalah

Berdasarkan rumusan masalah diatas, dapat disusun batasan masalah dalam pembuatan sistem aplikasi penentu estimasi biaya pengembangan perangkat lunak menggunakan metode COCOMO II adalah sebagai berikut :

1. Aplikasi berbasis *website* dan menggunakan Bahasa pemrograman PHP *Native*.
2. Aplikasi ini menggunakan metode perhitungan menggunakan rumus COCOMO II.
3. Metode perhitungan COCOMO II fokus terhadap submodel *post architecture*.
4. Aplikasi ini fokus menentukan estimasi biaya pengembangan perangkat lunak sebagai referensi.

### 1.4 Tujuan

Berdasarkan rumusan masalah diatas, dapat disusun tujuannya adalah menghasilkan sistem aplikasi penentu estimasi biaya pengembangan perangkat lunak menggunakan metode COCOMO II.

## 1.5 Manfaat

Berdasarkan rumusan masalah diatas, dapat disusun manfaat yang diperoleh dari aplikasi tersebut, antara lain :

### 1. Manfaat Pengguna :

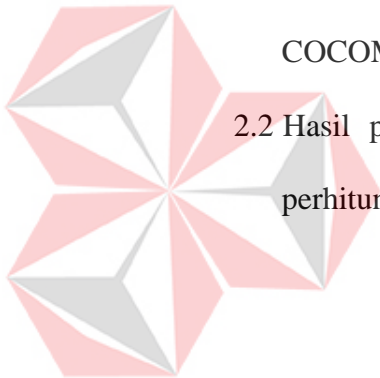
1.1 Aplikasi ini akan memberikan informasi estimasi biaya dalam proyek pengembangan perangkat lunak.

1.2 Aplikasi ini memudahkan pengguna untuk menentukan perkiraan harga sebelum proyek ditenderkan.

### 2. Manfaat Peneliti :

2.1 Menambah wawasan atau kemampuan berfikir mengenai penerapan metode COCOMO II.

2.2 Hasil penelitian dapat dijadikan sebagai acuan pengembangan sistem perhitungan perkiraan harga dengan metode COCOMO II.



UNIVERSITAS  
Dinamika

## BAB II

### LANDASAN TEORI

#### 2.1 COCOMO II

Metode COCOMO II atau *Cost Constructive Model* merupakan suatu metode untuk mengestimasi biaya dan penjadwalan dalam merencanakan sebuah proyek pengembangan perangkat lunak, dimana digunakan persamaan untuk menemukan jumlah orang, bulan dan biaya yang dibutuhkan untuk menyelesaikan suatu proyek. COCOMO II memberikan dukungan yang *up to date* untuk bisnis *software*, perangkat lunak berorientasi objek, perangkat lunak yang dibuat melalui pengembangan ulang dan perangkat lunak yang dikembangkan untuk keperluan komersial. COCOMO II dapat digunakan pada tahap awal desain proyek maupun proyek dengan arsitektur yang telah dibuat. Berikut merupakan kegunaan dari metode COCOMO II:

1. Menetapkan sebuah anggaran perusahaan terhadap pengeluaran yang digunakan untuk biaya perancangan perangkat lunak.
2. Memfasilitasi tawaran kontrak yang kompetitif.
3. Menentukan hal apa yang paling efektif bagi organisasi.

Dalam mencari estimasi biaya dengan menggunakan metode COCOMO II terdapat beberapa tahapan yang harus dilakukan yaitu:

- a. Mencari nilai UFP (*Unadjusted Function Point*) *Function point* mengukur kompleksitas sebuah sistem dengan menghitung total keseluruhan tipe berkas, *input* dan *output data* dan juga kontrol eksternal. Perhitungan dilakukan dengan cara mengkategorikan tipe fungsi pengguna yang ada pada DFD maupun pada

*database*, dimana ER diagram merepresentasikan penggunaan *database* beserta tabel dan *field* atau atribut. Data yang ada pada sistem kemudian digunakan untuk menentukan bobot kompleksitas sesuai dengan aturan DET, FTR, dan RET yang akan dipaparkan selanjutnya.

- b. Mencari Nilai *Scale Factor* Dan *Effort Multiplier* Dalam mencari nilai *scale factor* terdapat parameter penilaian untuk setiap bobot yang ada pada *scale factor*, penilaian tersebut dilakukan dengan cara memberikan kuisisioner kepada beberapa responden, responden disini merupakan tim pengembang perangkat lunak. Menurut Boehm *Scale Factor* terdiri dari 5 parameter penilaian.
- c. Untuk mencari nilai dari *Scale Factor* digunakan rumus persamaan *Scale Factor* sebagai berikut:  $E = B + 0.01 \times \sum SF_j$  Dari rumus diatas nilai B adalah 0.91, nilai tersebut merupakan nilai ketentuan dari jurnal COCOMO. Lalu SF merupakan nilai rata rata dari kelima *Scale Factor*. Lalu untuk mencari nilai *effort multiplier* dilakukan hal yang sama yaitu dengan menyebarkan kuisisioner. Kuisisioner berisi 17 point *effort adjustment multiplier* untuk menentukan usaha nominal yang menggambarkan suatu proyek perangkat lunak yang sedang dikembangkan.
- d. Mencari Estimasi Biaya Dan untuk langkah terakhir yaitu mencari estimasi nilai biaya dari proyek Sistem Informasi pariwisata kota Sawahlunto. Estimasi dilakukan melalui beberapa tahap dengan menggunakan beberapa persamaan. Pertama mencari nilai estimasi usaha (*person-month*) dengan menggunakan persamaan berikut:  $PM = A \times (size)^E \times \pi$  Dari rumus diatas dapat diketahui nilai A merupakan nilai ketentuan yang ada pada jurnal COCOMO dengan nilai sebesar 2.94, size merupakan nilai UFP yang telah dikonversikan

menjadi KLOC, E merupakan nilai dari *Scale Factor*, dan EM merupakan nilai rata rata dari *Effort Multiplier* (Prasetya Muhamad Aditya, 2018).

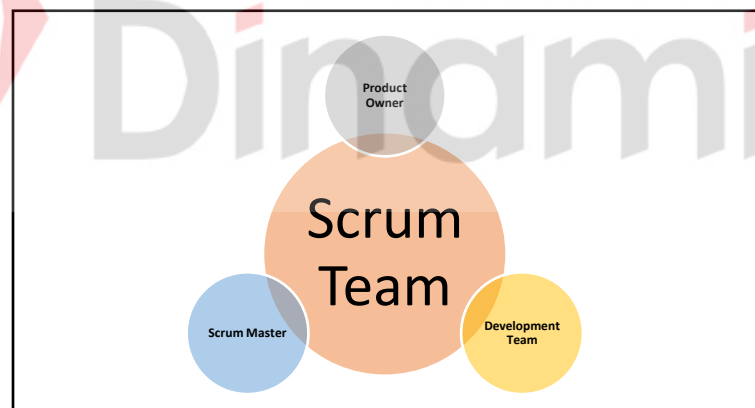
## 2.2 SCRUM

Menurut Ependi (2018) *SCRUM* merupakan model pengembangan perangkat lunak yang banyak digunakan baik pada skala industri maupun pendidikan. Kajian berkaitan dengan *SCRUM* diantaranya adalah dalam proses pengembangan sistem manajemen untuk melakukan analisis *performance indicator*, penelitian tersebut bertujuan untuk mengukur target dan pencapaian dari masing-masing karyawan berdasarkan tujuan perusahaan melalui *key performace indicator*. Pada penelitian tersebut *SCRUM* yang digunakan terdiri dari *backlog* dan *sprint*. Dimana dalam proses *backlog* aktivitas dilakukan yaitu studi literatur baik berupa hasil penelitian maupun produk sebelumnya, penyebaran kuesioner, dan wawancara sedangkan *sprint* aktivitas yang dilakukan terdiri dari *sprint*, *release planning* dan *actual sprint*. Kajian lain juga berkaitan dengan *SCRUM* yang dilakukan pada pengembangan laboratorium untuk perangkat lunak penelitian. Dalam kajian tersebut dilakukan pemanfaatan model *SCRUM* untuk perangkat lunak manajemen proyek dalam penelitian pengembangan laboratorium. Hasil dari penelitian tersebut menunjukkan model *SCRUM* dapat membuat atau mampu menyajikan informasi berkaitan dengan proyek yang dikerjakam. Pada *Scrum* sendiri merupakan bagian dari *Agile Development* yang memiliki *key scrum practices* yaitu (Alexandra, 2017):

1. Fokus kepada jadwal.
2. Bekerja sesuai sprint dengan konsisten.
3. Semua pekerjaan harus di tandai sebagai produk *backlog*.

4. Produk *backlog* dasar melakukan sprint dan tim harus memutuskan produk dapat dikembangkan atau tidak
5. *Scrum* master mempunyai tanggung jawab menerima hasil *sprint*.
6. Melakukan pertemuan setiap hari.
7. Fokus pada *sprint*, pertemuan, *review* produk dan jadwal pengerjaan.

*Scrum* memungkinkan pengerjaan pengumpulan kebutuhan, perancangan arsitektur dan antarmuka. *Scrum* juga memiliki kerangka kerja sendiri yang disebut dengan *scrum team* yang memiliki fleksibilitas, kreatif dan produktif. Pada *scrum team* mampu mengantarkan produk secara berkala, dapat memaksimalkan kesepakatan dan menerima masukan untuk dan dapat meningkatkan keberhasilan pengguna (Firdaus, 2017). Komponen *scrum team* dapat dilihat pada gambar 3 berikut:



Gambar 2. 1 *Scrum Team*  
Sumber: (Firdaus, 2017)

### 1. *Product Owner*

Pada *product owner* ini merupakan bagian dari *scrum team* yang dapat bekerjasama dengan setiap anggota tim lainnya dan memastikan bahwa pekerjaan yang dibutuhkan berhasil di selesaikan sesuai dengan kebutuhan oleh karena itu *product team* memiliki tanggung jawab untuk memaksimalkan nilai produk atau



bisnis dan hasil kerja tim pengembang. Menurut Aryadita (2018), *product owner* digunakan untuk mengumpulkan dan menyusun semua permintaan dan kebutuhan sebuah sistem, misalkan sebuah fitur-fitur yang dibutuhkan dan kebutuhan non fungsional pada sistem. Jika semua tujuan sudah ditetapkan, maka semua permintaan dan kebutuhan tersebut dibagi-bagi menjadi sebuah bagian kecil yang mana setiap bagian tersebut harus memiliki nilai serta layak untuk dipertimbangkan.

## **2. Scrum Master**

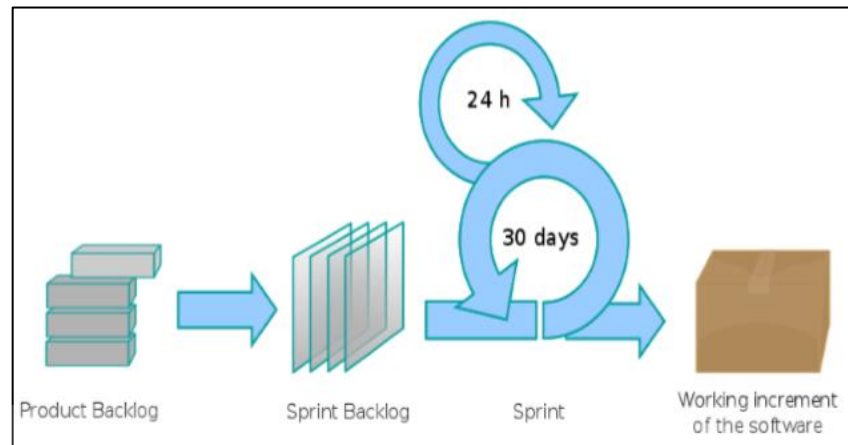
Pada *scrum master* ini bekerja dengan memastikan bahwa tim dapat melewati segala hambatan yang ditemukan dan sudah meminta bantuan sebelumnya, karena *scrum master* juga memiliki peran yang berhubungan dengan *product owner*. Menurut Partogi (2016), seseorang yang mengajarkan cara yang lebih kolaboratif dan menyenangkan dalam mengembangkan sebuah perangkat lunak. Jika berbicara topik mengenai scrum, master scrum bertanggung jawab dalam memahami scrum secara keseluruhan hingga mengenai pola pikir serta nilai dari scrum itu sendiri (*openness, focus, courage, commitment*).

## **3. Development Team**

Pada *development team* memiliki tanggung jawab dalam menyusun dan mengelola pekerjaan yang telah diberikan. Oleh karena itu pada development team terdiri dari para ahli yang sesuai dengan bidangnya, yang nantinya jika pekerjaan tersebut terselesaikan, maka akan dirilis disetiap akhir *Sprint*.

*SCRUM* juga digunakan tidak hanya untuk mengembangkan perangkat lunak namun juga dapat digunakan untuk melakukan evaluasi penerapan teknologi, kondisi tersebut terjadi ketika melakukan manajemen proyek untuk mengevaluasi

kemungkinan penerepan sebuah metodologi bagi sebuah perusahaan. Evaluasi dilakukan untuk melihat kemungkinan penerapan dan pengembangan proyek berdasarkan karakteristik perusahaan. Konseptual *SCRUM* dapat dilihat pada gambar 4 berikut:



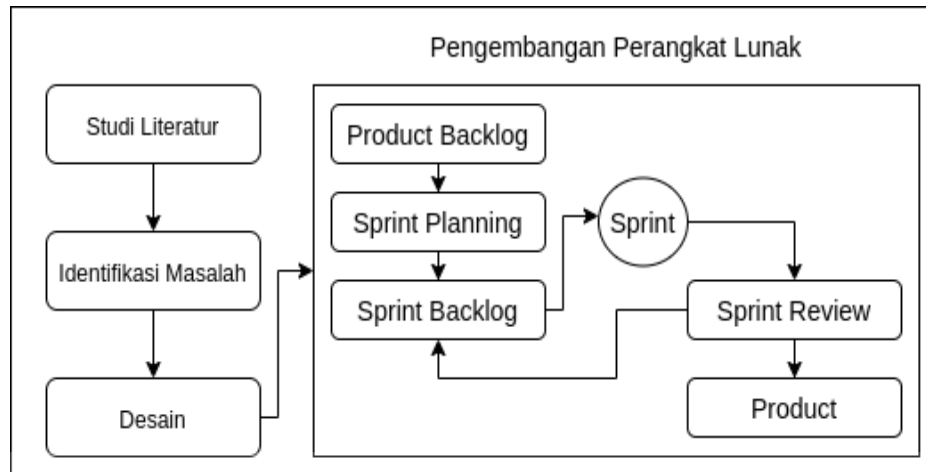
Gambar 2. 2 Konseptual *SCRUM*

Sumber: (Firdaus, 2017)

Menurut Fajrin (2018), *development team* dalam scrum memiliki kewenangan penuh dalam melakukan eksekusi sebuah *product backlog* yang telah dibuat dan disepakati oleh tim agar menghasilkan sebuah *increment* yang selesai dalam setiap *sprint*. Selain wewenang yang telah dijelaskan sebelumnya, *development team* memiliki nilai-nilai seperti nilai komitmen, keberanian, saling menghargai, keterbukaan dan fokus.

## BAB III METODOLOGI PENELITIAN

### 3.1 Gambaran Umum Metodologi



Gambar 3. 1 Sistem Estimasi Biaya Penentu HPS berbasis *SCRUM*.

Dalam metodologi penelitian ini yaitu berbasis *SCRUM* yang diawal dengan melakukan sebuah studi literatur COCOMO II, Harga Perkiraan Sendiri (HPS), dan estimasi biaya proyek pengembangan perangkat lunak dari beberapa jurnal. Kemudian melakukan identifikasi permasalahan untuk mengetahui permasalahan yang ada, solusi dan tujuan dari penelitian ini. Hasil dari studi literatur dan identifikasi permasalahan akan dibuatkan sebuah desain terlebih dahulu yang nantinya akan dimasukkan ke *Product Backlog Item* (PBI). Kemudian melakukan tahap selanjutnya yaitu *sprint planning* yang harus dilakukan pada setiap awal *sprint* yang bertujuan untuk menentukan selesai tidak *sprint* tersebut serta memasukkan PBI yang telah dipilih untuk setiap *sprint* yang akan dimasukkan kedalam *sprint backlog*. Setelah melakukan *sprint planning* kemudian masuk ke fase *sprint* untuk *development* aplikasi. Jika telah selesai melakukan *development* kemudian harus melakukan

*sprint review* yang bertujuan untuk melakukan *review* terhadap PBI yang dilakukan pada tahap *sprint*. Jika PBI telah dianggap selesai, maka PBI tersebut dimasukkan kedalam *product* dan jika belum selesai maka PBI masuk lagi kedalam *sprint planning* untuk dikerjakan kembali.

### 3.2 Studi Literatur

Pada tahap ini penulis melakukan kajian dan mencari referensi teori yang sesuai dengan topik atau kasus yang ditemukan. Referensi yang dimaksud adalah sebagai berikut:

1. COCOMO II
2. Estimasi biaya proyek pengembangan perangkat lunak
3. HPS (Harga Perkiraan Sendiri)

Referensi diatas dapat ditemukan melalui buku, jurnal, artikel laporan penelitian dan situs-situs di internet. Sehingga luaran yang dapat diambil dari studi literatur ini dapat digunakan untuk memperkuat permasalahan serta sebagai dasar teori dalam melakukan studi. Selain itu, menjadi dasar untuk menerapkan metode COCOMO II untuk menentukan estimasi biaya proyek pengembangan perangkat lunak.

#### 3.2.1 Hasil Studi Literatur

Model COCOMO II merupakan model terbaru dari COCOMO. Model tersebut dipilih untuk penelitian ini karena dibandingkan dengan metode-metode estimasi biaya yang lain, COCOMO II memiliki kemampuan analisis yang lebih sempurna yaitu mampu digunakan untuk memprediksi biaya, waktu, dan jumlah pegawai yang dibutuhkan dalam menyelesaikan proyek pembuatan perangkat lunak skala kecil maupun besar (Khoiro, 2018).

Estimasi biaya perangkat lunak merupakan bagian tak terpisahkan dari pengembangan perangkat lunak. Estimasi biaya perangkat lunak sangat dibutuhkan dalam membuat usulan proposal, negosiasi kontrak, penjadwalan, kontrol dan monitoring. Untuk itu akurasi biaya perangkat lunak sangat dibutuhkan karena:

- A. Dapat membantu untuk mengklasifikasi dan menentukan prioritas pengembangan proyek sehubungan dengan perencanaan bisnis secara keseluruhan.
- B. Dapat digunakan untuk menentukan resources yang sesuai dengan proyek yang dikerjakan dan seberapa baik resources ini akan digunakan.
- C. Dapat digunakan untuk menilai pengaruh terhadap perubahan dan dukungan dalam perencanaan ulang.
- D. Dapat memudahkan untuk mengatur dan mengontrol proyek ketika resources benar-benar sesuai dengan yang dibutuhkan (Dillak, 2012).

Harga Perkiraan Sendiri (HPS) adalah bagian dari proses perencanaan pengadaan barang /jasa yang disusun dan ditetapkan oleh Pejabat Pembuat Komitmen (PPK). HPS harus memini malisir kekeliruan dalam proses penyusunannya karena bisa berakibat gagal lelang apabila Harga penawaran berada di atas HPS dan sebaliknya harga HPS terlalu tinggi membuat harga penawaran menjadi tidak wajar. Pemerintah sebagai penyelenggara proses pengadaan barang/jasa dalam pelaksanaannya masih sering mengalami kesulitan dalam proses penyusunan HPS. Menyusun HPS harus memiliki keahlian karena selain harus memahami proses pengadaan juga harus bisa mengidentifikasi secara rinci spesifikasi teknis barang/jasa yang di adakan (Putera, 2019).

### 3.3 Identifikasi Permasalahan

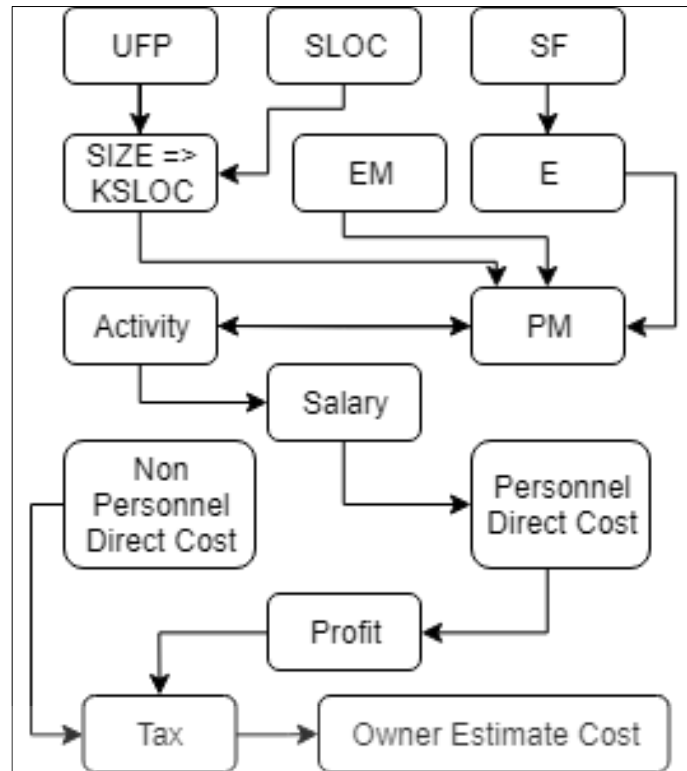
Tahap ini dilakukan untuk mengidentifikasi permasalahan solusi dan tujuan dengan cara observasi terhadap beberapa jurnal yang membahas tentang COCOMO II, HPS, dan estimasi biaya proyek pengembangan perangkat lunak. Hasil observasi tersebut dimasukkan kedalam *Product Backlog Item*.

Berdasarkan studi literatur yang telah dilakukan dapat mengambil kesimpulan bahwa di negara Indonesia belum ada penentuan secara khusus terkait HPS tersebut. Pemerintahan hanya melakukan survei harga pasar yang berubah terus menerus bahkan setiap daerah mempunyai harga pasar yang berbeda beda untuk proyek pengembangan aplikasi. Sehingga pihak pemerintahan kesulitan untuk menetapkan HPS untuk sebuah proyek pengembangan perangkat lunak sebelum ditenderkan.

### 3.4 Desain Konsep Aplikasi

#### 3.4.1 Desain Sistem COCOMO II

Desain sistem *base* COCOMO II adalah proses sebuah proses perhitungan menggunakan rumus COCOMO II. Dan parameter apa saja yang dibutuhkan dalam menggunakan rumus COCOMO II untuk menghasilkan sebuah nilai total *owner estimate cost*.



Gambar 3. 2 Diagram Estimasi Biaya Berbasis COCOMO II

Gambar 3.2 adalah sebuah diagram alur COCOMO II dan parameter apa saja yang dibutuhkan untuk menghasilkan nilai total proyek. Diawali dengan parameter UFP (Unjusted Function Point). UFP memiliki 5 parameter tersendiri yaitu *Internal Logic File (ILF)*, *External Interfaces File (EIF)*, *External Input (EI)*, *External Output (EO)* dan *External Inquiry (EQ)* (Febrina Putri Laksamana, 2018). Setelah menemukan nilai total UFP. Kemudian menentukan bahasa pemrogramman yang akan digunakan bisa disebut dengan size in SLOC. Nilai *size in SLOC* akan diubah menjadi Kilo SLOC (KSLOC). Kemudian mencari nilai total *eksponen E*. nilai E didapat dari parameter *Scale Factor (SF)*. Nilai SF mempunyai 5 parameter tersendiri yaitu *Precedentedness (PREC)*, *Flexibility of development (FLEX)*, *Risk resolution (RESL)*, *Team of cohesion (TEAM)*, dan *Project maturity (PMAT)*. Setelah mendapatkan nilai total eksponen E. Kemudian mencari nilai total *Effort*

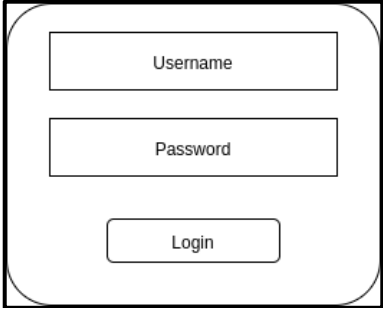
*Multiplier* (EM). EM mempunyai 16 parameter tersendiri. Setelah mendapatkan nilai total EM. Kemudian menghitung nilai total PM. nilai total membutuhkan nilai total KSLOC, Eksponen E dan EM. Nilai PM akan dibandingkan nilai total *activity* dan harus sama. Untuk mencari nilai total *activity* dibutuhkan parameter *waterfall* atau metode pengembangan yang lain. Setiap nilai dari parameter *activity* akan dikalikan dengan PM. Dan ketika dijumlah nilai total *activity* harus sama, jika tidak sama *user* harus menambahkan beberapa nilai sedikit pada salah satu *activity*. Setelah nilai PM dan total *activity* sama *user* harus mengisi personil yang dibutuhkan dan gaji yang telah ditetapkan. Jumlah gaji tersebut akan di kalikan dengan hasil per *activity* sesuai dengan *jobdecs* tiap personil. Dan akan jika ditotal akan mendapatkan gaji total personil. Dari total gaji personil tersebut akan ditambahkan profit yang diinginkan oleh user dalam hitungan satuan (%). Kemudian menghitung berapa total biaya non personil. Dan akan dijumlahkan dengan biaya personil. Setelah menjumlahkan biaya total personil dan non personil kemudian menghitung pajak proyek tersebut dalam satuan persen (%) dari total biaya terakhir. Kemudian pajak dan total biaya keseluruhan akan dijumlahkan untuk mendapatkan nilai total *owner estimate cost*.

### **3.4.2 Desain I/O**

#### **A. Desain UI Login**

Berikut merupakan desain *interface* dari tampilan *login* sesuai dengan gambar 3.3. Desain tersebut digunakan untuk pengguna dalam memasukkan username dan password yang dimiliki, sehingga dapat memasuki halaman utama aplikasi penilaian estimasi biaya perangkat lunak menggunakan metode COCOMO II.



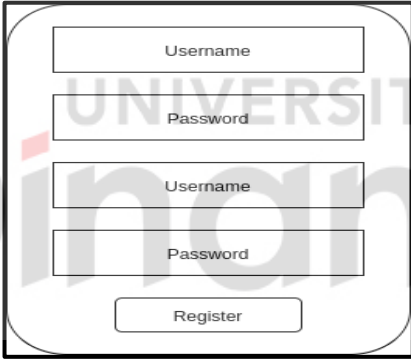
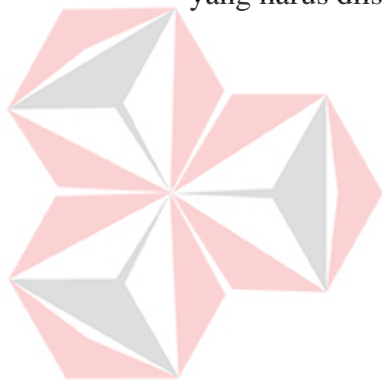


A wireframe diagram of a login form. It consists of a rounded rectangular container with a black border. Inside, there are three rectangular input fields stacked vertically. The top field is labeled 'Username', the middle field is labeled 'Password', and the bottom field is a button labeled 'Login'.

Gambar 3. 3 UI *Login*

### B. Desain UI *Register*

Desain UI *Register* digunakan untuk melakukan perencanaan desain tampilan *register*. Desain tersebut akan menampilkan *form register* dengan beberapa *textfield* yang harus diisi oleh *user* seperti nama, *email*, *username* dan *password*.



A wireframe diagram of a register form. It consists of a rounded rectangular container with a black border. Inside, there are four rectangular input fields stacked vertically. The first field is labeled 'Username', the second is labeled 'Password', the third is labeled 'Username', and the fourth is labeled 'Password'. Below these fields is a button labeled 'Register'.

Gambar 3. 4 UI *Register*

### C. Desain UI Mengelola Proyek dan Modul

Desain UI mengelola proyek dan modul digunakan sebagai rancangan tampilan halaman mengelola proyek dan modul. Desain tampilan tersebut *user* dapat melakukan tambah nama modul dan simpan nama proyek.

Nama Proyek		Simpan
Nama Modul		Simpan
Value 1	Value 2	Value 3
Value 4	Value 5	Value 6
Value 7	Value 8	Value 9

Gambar 3. 5 UI Mengelola Proyek dan Modul

#### D. Desain UI Perhitungan

Desain UI Perhitungan digunakan untuk proses perhitungan penilaian estimasi biaya perangkat lunak dengan menggunakan metode COCOMO II.



Gambar 3. 6 UI Perhitungan COCOMO II

#### E. Desain UI *History*

Berikut merupakan desain UI *history* dari aplikasi yang digunakan untuk menampilkan proyek yang sudah didaftarkan baik dari proyek yang sudah dihitung atau belum dihitung.

Value 1	Value 2	Value 3	Edit
Value 4	Value 5	Value 6	
Value 7	Value 8	Value 9	
			Delete
			Tambah

Gambar 3.7 UI *History*

### 3.4.3 Desain *Testing*

Desain *testing* adalah digunakan untuk mencoba semua fitur yang telah dikerjakan pada semua sprint untuk mengetahui apakah semua fitur tersebut bisa berfungsi dengan baik. Hasil desain testing akan di-*review* dan masuk ketahap *sprint review* setiap iterasi. Penjelasan lebih detail terkait desain *testing* terdapat pada lampiran 1.

#### A. Desain *Testing* Halaman Perhitungan Modul

Halaman perhitungan modul merupakan sebuah fitur untuk menentukan estimasi nilai total proyek. Pengujian pada tahap ini dilakukan untuk mengetahui apakah halaman *history* dapat berjalan dengan sukses dengan cara memasukan parameter sesuai yang dibutuhkan oleh metode COCOMO II pada aplikasi.

Tabel 3. 1 Desain *Testing* Perhitungan Modul

Nama <i>Field</i>	Data Masukan
UFP	Dataset
SF	Dataset
EM	Dataset
PM	Dataset
Salary	Dataset
Profit	Dataset
TAX	Dataset

Setelah itu dilakukan pengujian dengan *input*-an yang telah disepakati sesuai dengan tabel 3.1. Berikut merupakan langkah-langkah pengujian halaman perhitungan modul, dapat dilihat pada tabel 3.2.

Tabel 3. 2 Rancangan Pengujian Perhitungan *Modul*

No.	Deskripsi	Prosedur pengujian	Masukan	<i>Output</i> yang diharapkan
1	Pengujian tampilan halaman perhitungan modul	Melakukan pengecekan parameter yang harus diisi lalu	Memasukkan parameter sesuai tabel 3.1	Sistem akan melakukan otomatisasi lalu menghasilkan

No.	Deskripsi	Prosedur pengujian	Masukan	Output yang diharapkan
		melakukan otomatisasi perhitungan		nilai total proyek
2	Pengujian tampilan halaman perhitungan modul	Melakukan pengecekan parameter yang harus diisi lalu melakukan otomatisasi perhitungan	Melakukan pengecekan parameter yang harus diisi lalu melakukan otomatisasi perhitungan	Sistem tidak akan melakukan otomatisasi kalau inputannya kosong .

### 3.5 Pengembangan Perangkat Lunak

Tahap *software development* adalah proses pembuatan aplikasi dari awal hingga selesai dan menggunakan metode *SCRUM* dalam perancangan serta pembuatan aplikasi.

Dalam penelitian ini metode pengembangan perangkat lunak yang digunakan adalah *SCRUM* yang telah memiliki proses seperti *product backlog*, *sprint backlog*, *sprint*, dan *working increment of the software*. Tahapan dalam menggunakan metode *SCRUM* ini dapat dilihat pada gambar 3.1.

#### 3.5.1 Product Backlog Item

Tabel 3. 3 *Product Backlog Item*

P.ID	Product Backlog	Tingkat Kesulitan	Sprint	Time
1	Fitur <i>Login User</i>	<i>Low</i>		
2	Fitur <i>Register User</i>	<i>Low</i>		
3	Fitur CRUD Proyek	<i>Low</i>	1	2 Minggu
4	Fitur CRUD Modul	<i>Low</i>		
5	Otomatisasi Rumus Sesuai Inputan User	<i>Medium</i>	2	2 Minggu
6	Report Excel & PDF Proyek	<i>Medium</i>		
7	<i>Save History</i> Proyek User	<i>Medium</i>	3	2 Minggu
8	<i>Deploy</i> Aplikasi in VPS	<i>Height</i>	4	2 Minggu

Setelah melakukan identifikasi permasalahan dan memahami kebutuhan dari hasil indentifikasi permasalahan tersebut tahap selanjutnya adalah menyusun sebuah *Product Backlog Item* yang berisi item apa saja yang akan dikerjakan untuk merancang dan membangun aplikasi estimasi penentu estimasi biaya proyek pengembangan perangkat lunak menggunakan metode COCOMO II.

### **Keterangan:**

Tabel 3. 4 Tingkat Kesulitan

<b>Tingkat Kesulitan</b>	<b>Skala</b>
<i>Hight</i>	7 - 9
<i>Medium</i>	4 - 6
<i>Low</i>	1 - 3

### **3.5.2 Sprint Planning**

Pada *Sprint Planning* dilakukan disetiap awal *sprint* yang bertujuan untuk menentukan apakah telah selesai atau belum serta memilih *product backlog item* yang akan dikerjakan pada *sprint*. Telah selesai atau belum ditentukan pada tahap *sprint planning* ini. *Product backlog item* dikatakan telah selesai jika *Usecase system*, *Activity Diagram*, *Sequence Diagram* dan *class diagram* telah dibuat.

### **3.5.3 Sprint Backlog**

Tahap *Sprint Backlog* berisi *product backlog item* yang dipilih untuk dikerjakan pada *sprint*. Item pada backlog ditentukan oleh penulis dan *product owner*.

### **3.6.1 Sprint backlog Iterasi Pertama**

Hasil dari *sprint planning* untuk iterasi pertama adalah sebagai berikut.

Tabel 3. 5 *Sprint Backlog* Iterasi Pertama

<b>ID</b>	<b>Task</b>	<b>Deskripsi Pekerjaan</b>
1	Membuat Fitur <i>Login</i>	<i>Login</i> untuk pihak pemerintahan dan admin agar dapat
2	Membuat Fitur <i>Register</i>	<i>Register</i> untuk pihak pemerintahan agar mendapatkan akses masuk aplikasi
3	Membuat Fitur CRUD Proyek	<i>User</i> dapat menambah, mengedit dan menghapus proyek
4	Membuat Fitur CRUD Modul	<i>User</i> dapat menambah, mengedit dan menghapus modul

### 3.6.2 *Sprint Backlog* Iterasi Kedua

Hasil dari *sprint planning* untuk iterasi kedua adalah sebagai berikut.

Tabel 3. 6 *Sprint Backlog* Iterasi Kedua

<b>ID</b>	<b>Task</b>	<b>Deskripsi Pekerjaan</b>
1	Otomatisasi Rumus COCOMO II	<i>Developer</i> Mengintegrasikan Rumus COCOMO II terhadap inputan parameter oleh <i>user</i>

### 3.6.3 *Sprint Backlog* iterasi Ketiga

Hasil dari *sprint planning* untuk iterasi ketiga adalah sebagai berikut.

Tabel 3. 7 *Sprint Backlog* Iterasi Ketiga

<b>ID</b>	<b>Task</b>	<b>Deskripsi Pekerjaan</b>
1	<i>Report</i> Excel Proyek	<i>User</i> dapat mengekspor <i>history</i> perhitungan <i>detail</i> dalam bentuk excel
2	<i>Save History</i> Proyek	Sistem harus bisa menyimpan <i>history</i> proyek yang telah dihitung

### 3.6.4 *Sprint Backlog* Iterasi Keempat

Hasil dari *sprint planning* untuk iterasi keempat adalah sebagai berikut.

Tabel 3. 8 *Sprint Backlog* Iterasi Keempat

<b>ID</b>	<b>Task</b>	<b>Deskripsi Pekerjaan</b>
1	<i>Deployment</i> Aplikasi ke VPS ( <i>Virtual Private Server</i> )	<i>Developer</i> Mengupload proyek aplikasi kedalam server dengan domain tertentu agar dapat diakses oleh pihak luar.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Hasil *Sprint*

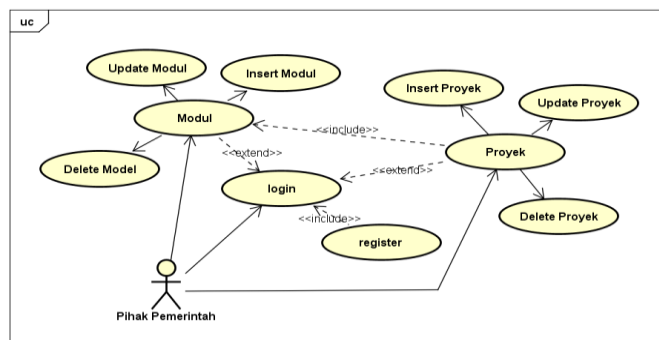
dari pengembangan perangkat lunak berdasarkan gambar 3.1 dan 3.2 yang dilakukan dengan pendekatan *SCRUM* dihasilkan aplikasi yang sesuai dengan kebutuhan dan dilakukan iterasi untuk improvisasi terkait dengan kemudahan pembuatan setiap fitur dalam aplikasi proses itu masuk pada tahap *sprint*.

#### 4.1.1 *Sprint* Iterasi Pertama

*Sprint* iterasi pertama diambil dari *sprint backlog* iterasi pertama sehingga untuk *Sprint* iterasi pertama menghasilkan fitur *login user*, fitur *register user*, fitur CRUD proyek, dan CRUD modul. Hasil *sprint* iterasi pertama akan di-*review* pada tahap *sprint review* iterasi pertama.

##### A. *Usecase System*

*Usecase* pada aplikasi estimasi penentu biaya proyek perangkat lunak untuk *sprint* iterasi pertama yang dijelaskan pada gambar 4.1. *Usecase* tersebut menjelaskan tentang beberapa proses utama dan sub proses yang ada dalam *sprint* iterasi pertama yang terdiri dari *activity login*, *activity register*, *activity* CRUD proyek, dan *activity* CRUD modul.

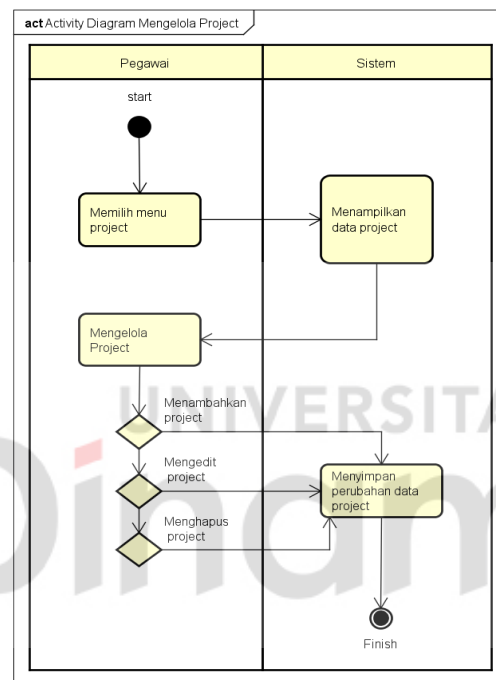


Gambar 4. 1 *Usecase System* Iterasi Pertama

## B. Activity Diagram

*Activity diagram* aplikasi estimasi penentuan estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi pertama yang dijelaskan pada gambar 4.2, 4.3, L2. 1, dan L2. 2. *Activity diagram* menjelaskan lebih detail terkait *usecase* yang telah dibuat. Penjelasan lebih detail terdapat lampiran 2.

### B.1 Activity Diagram CRUD Proyek



Gambar 4. 2 Activity Diagram CRUD Proyek

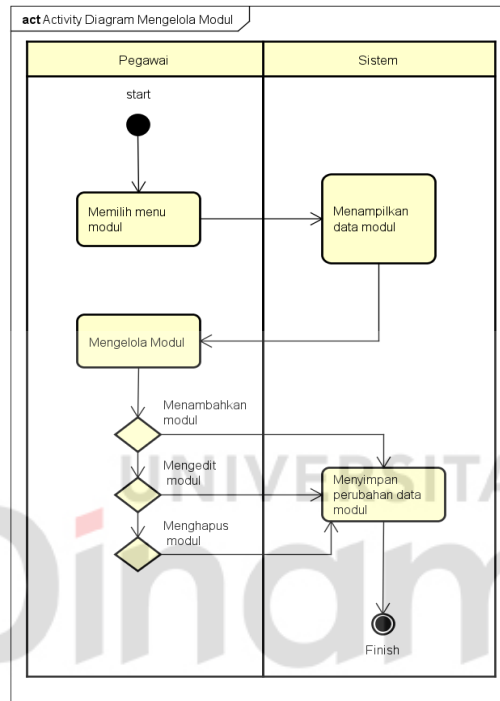
Tabel 4. 1 Flow of Event CRUD Proyek

Deskripsi	Usecase mengelola data proyek	
Kondisi Awal	Pegawai telah <i>login</i> pada aplikasi	
Kondisi Akhir	Pegawai dapat mengelola data proyek	
Aliran Kejadian Utama		
No.	User	Sistem
1.	User memilih menu proyek pada aplikasi	Sistem menampilkan halaman <i>list</i> data proyek
2.	User memilih pengelolaan yang akan dilakukan	<ol style="list-style-type: none"> <li>1. User memilih <i>Entry</i> data, sistem akan menampilkan <i>form entry</i> proyek.</li> <li>2. User memilih <i>edit</i> data, sistem akan menampilkan <i>form</i> yang berisi data yang sudah ada.</li> <li>3. User memilih <i>lihat</i>, sistem akan menampilkan data detail proyek</li> <li>4. User memilih <i>delete</i>, sistem akan menghapus data proyek yang dipilih pada aplikasi</li> </ol>



Pada gambar 4.2 akan dijelaskan lebih detail melalui tabel 4.1 yaitu tabel *flow of event* CRUD proyek yang menjelaskan bagaimana alur sistem mengelola data proyek pada aplikasi dan parameter apa saja yang akan dibutuhkan oleh *user* saat mengelola data proyek.

## B.2 Activity Diagram CRUD Proyek



Gambar 4. 3 Activity Diagram CRUD Modul

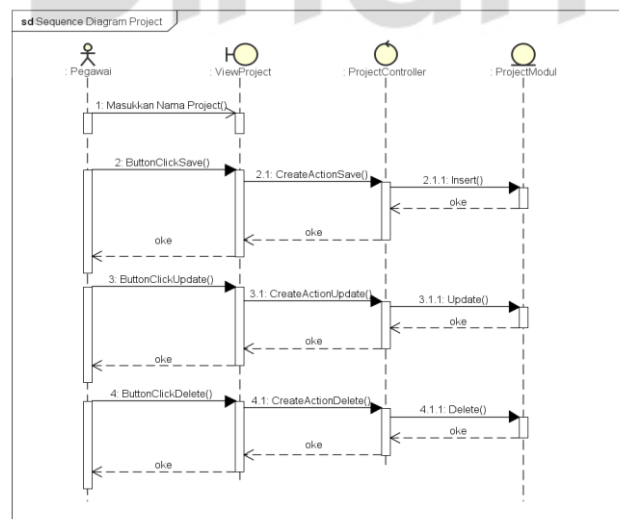
Pada gambar 4.3 akan dijelaskan lebih detail melalui tabel 4.2 yaitu tabel *flow of event* CRUD modul yang menjelaskan bagaimana alur sistem mengelola data modul pada aplikasi dan parameter apa saja yang akan dibutuhkan oleh *user* saat mengelola data modul.

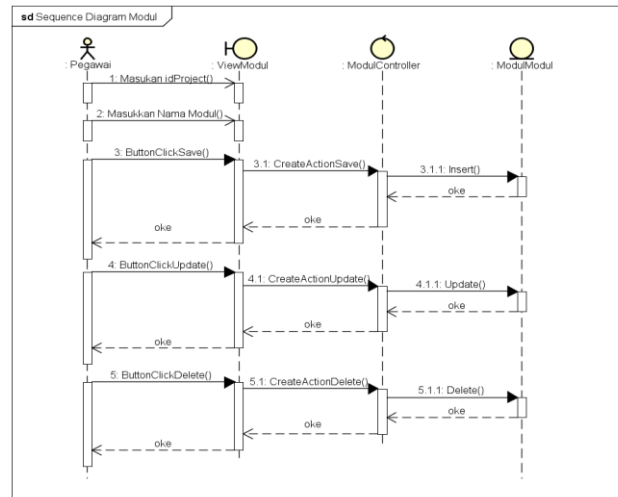
Tabel 4. 2 *Flow of Event* CRUD Modul

Deskripsi	<i>Usecase</i> mengelola data modul	
Kondisi Awal	Pegawai telah <i>login</i> pada aplikasi	
Kondisi Akhir	Pegawai dapat mengelola data modul	
Aliran Kejadian Utama		
No.	<i>User</i>	Sistem
1.	<i>User</i> memilih menu modul pada aplikasi	Sistem menampilkan halaman list data modul pada tiap proyek
2.	<i>User</i> memilih pengelolaan yang akan dilakukan	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih <i>Entry</i> data, sistem akan menampilkan <i>form entry</i> modul.</li> <li>2. <i>User</i> memilih <i>edit</i> data, sistem akan menampilkan <i>form</i> yang berisi data yang sudah ada.</li> <li>3. <i>User</i> memilih lihat, sistem akan menampilkan data detail modul</li> <li>4. <i>User</i> memilih <i>delete</i>, sistem akan menghapus data modul yang dipilih pada aplikasi</li> </ol>

### C. *Sequence Diagram*

*Sequence diagram* aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi pertama yang dijelaskan pada gambar 4.4, 4.5, L2. 3, dan L2. 4. *Sequence diagram* menjelaskan lebih detail lagi terkait *activity diagram* yang telah dibuat. Penjelasan lebih detail terdapat lampiran 2.

Gambar 4. 4 *Sequence Diagram* Proyek

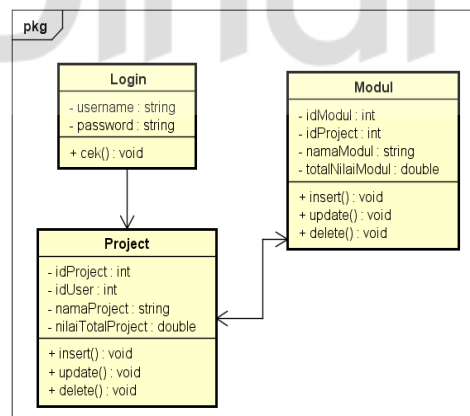


Gambar 4. 5 Sequence Diagram CRUD Modul

#### D. Class Diagram

*Class diagram* aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi pertama yang dijelaskan pada gambar 4.6.

*Class diagram* menjelaskan terkait function dan data yang dibutuhkan pada *sprint* iterasi pertama.



Gambar 4. 6 Class Diagram Iterasi Pertama

#### E. Hasil Desain I/O

Hasil Desain I/O adalah tampilan aplikasi yang telah dibuat pada aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak pada *sprint* iterasi pertama. Pejelasan lebih detail terkait hasil desain IO terdapat pada lampiran 3.

## E.1 Tampilan Proyek

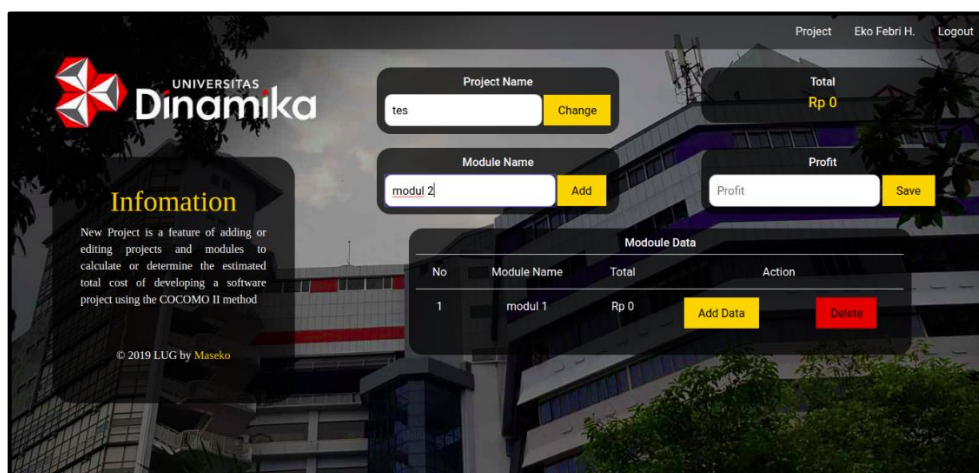
Halaman proyek merupakan halaman yang digunakan untuk mendaftarkan proyek sebelum mendaftar modul untuk dihitung menggunakan metode COCOMO II dan mendapatkan nilai total. Pada halaman proyek user harus mengisi nama proyek yang akan dibuat seperti gambar 4.8 berikut.



Gambar 4. 8 Tampilan Proyek

## E.2 Tampilan Modul

Halaman modul merupakan halaman yang digunakan untuk mendaftarkan modul dari proyek user untuk dihitung menggunakan metode COCOMO II dan mendapatkan nilai total modul seperti gambar 4.9 berikut.



Gambar 4. 9 Tampilan Modul

## F. *Sprint Review* Iterasi Pertama

Hasil *review* yang didapatkan setelah demo kepihak pemerintahan untuk *sprint* iterasi pertama dijelaskan pada tabel berikut.

Tabel 4. 3 Tabel *Sprint Review* Iterasi Pertama

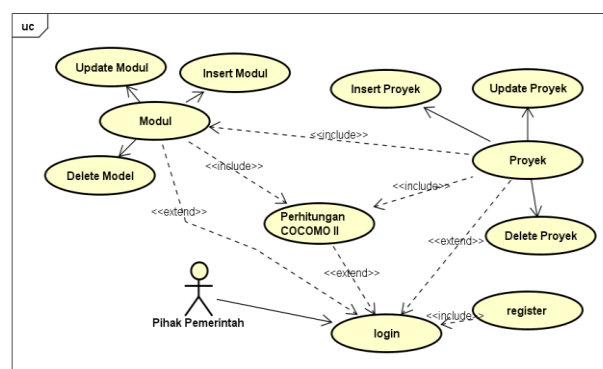
No	Fungsional	Hasil Review
1	Fitur <i>Login</i>	1. Fitur <i>login</i> berhasil dan tidak ada <i>bug / error</i> saat melakukan akse <i>login</i>
2	Fitur <i>Register</i>	2. Ftiru <i>register</i> berhasil dan tidak ada <i>bug / error</i> saat proses <i>register</i>
4	Fitur CRUD Proyek	1. Menambahkan proyek berhasil 2. Mengedit proyek berhasil 3. Menghapus proyek berhasil
5	Fitur CRUD Modul	1. Menambahkan modul sesuai proyek berhasil 2. Mengedit sesuai proyek berhasil 3. Menghapus modul sesuai proyek berhasil

### 4.1.2 *Sprint* Iterasi Kedua

*Sprint* iterasi kedua menghasilkan fitur implementasi rumus COCOMO II dan mengintegrasikan rumus COCOMO II dengan *input user*. Hasil *sprint* Iterasi pertama akan di-*review* pada *sprint* iterasi kedua.

#### A. *Usecase System*

*Usecase* pada aplikasi estimasi penentu biaya proyek perangkat lunak untuk *sprint* iterasi kedua yang dijelaskan pada gambar 4.10. *Usecase* tersebut menjelaskan tentang beberapa proses utama dan sub proses yang ada dalam *sprint* iterasi kedua yaitu otomatisasi rumus COCOMO II.



Gambar 4. 10 *Usecase System* Iterasi Kedua

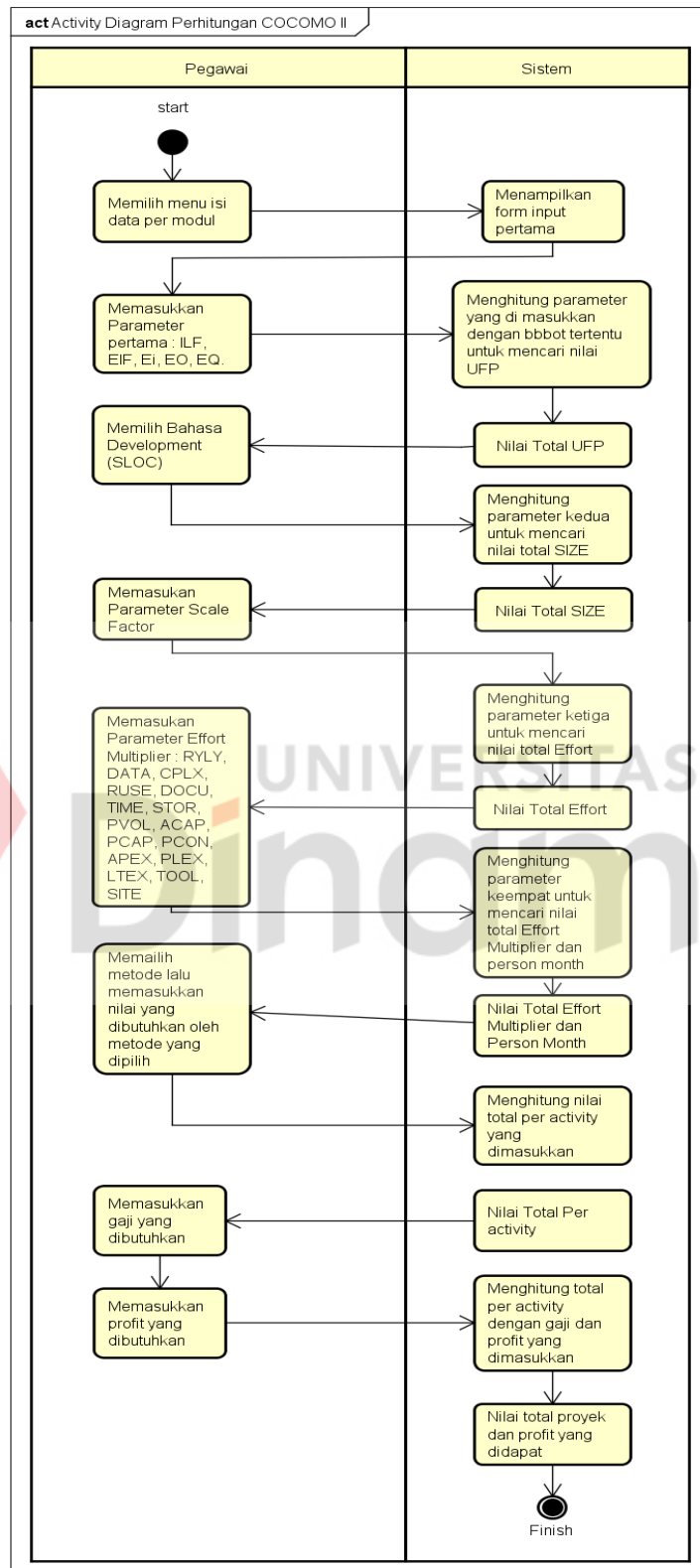
## **B. Activity Diagram**

*Activity diagram* aplikasi estimasi penentuan estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi kedua yang dijelaskan pada gambar 4.11. *Activity diagram* menjelaskan lebih detail terkait *usecase* yang telah dibuat.



UNIVERSITAS  
**Dinamika**

**B.1 Diagram Activity Otomatisasi Rumus COCOMO II**



Gambar 4. 11 Activity Diagram CRUD Proyek

Pada gambar 4.11 akan dijelaskan lebih *detail* melalui tabel 4.4 yaitu tabel *flow of event* Otomatisasi Rumus COCOMO II yang menjelaskan bagaimana alus sistem untuk mencari nilai total proyek pada aplikasi dan parameter apa saja yang akan dibutuhkan oleh *user* saat menghitung menggunakan metode COCOMO II.

Tabel 4. 4 *Flow of Event* Otomatisasi Rumus COCOMO II

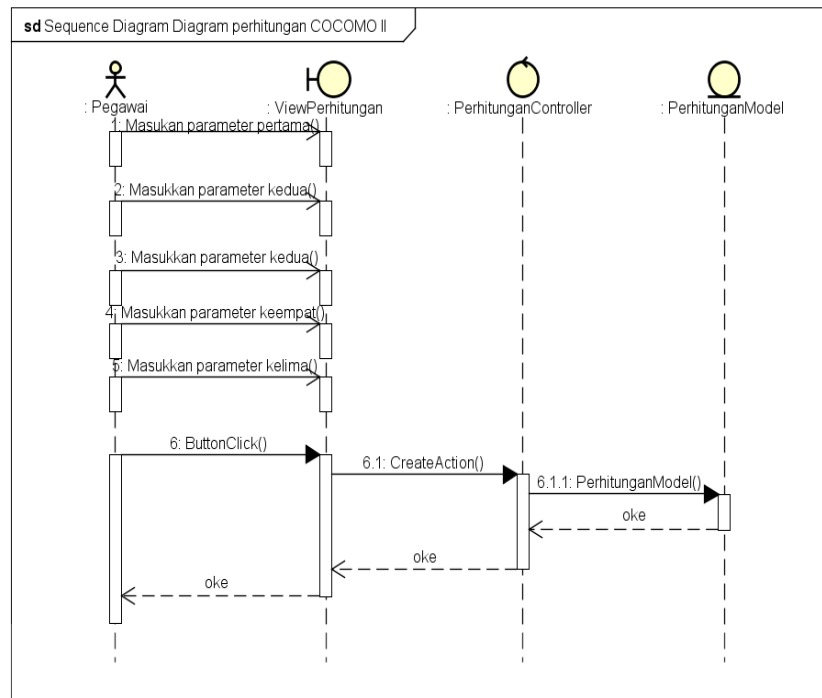
Deskripsi		Usecase perhitungan COCOMO II
Kondisi Awal		Pegawai telah <i>login</i> pada aplikasi
Kondisi Akhir		Pegawai dapat menghitung nilai total proyek
Aliran Kejadian Utama		
No.	User	Sistem
1.	<i>User</i> memilih menu proyek	Sistem menampilkan halaman proyek
2	<i>User</i> memilih modul yang akan dihitung	Sistem akan menampilkan <i>form entry</i> parameter (ILF, EIF, EI, EO, EQ)
3	<i>User</i> memasukkan nilai (ILF, EIF, EI, EO, EQ)	Sistem akan mentotal semua nilai yang dimasukkan oleh untuk mencari nilai total UFP
4	<i>User</i> memilih bahasa <i>development</i> (SLOC)	Sistem akan menghitung nilai yang telah dipilih oleh <i>user</i> untuk mencari nilai total SIZE
5	<i>User</i> memasukkan nilai Scale Factor	Sistem akan menghitung nilai total scale factor yang telah dimasukkan oleh user untuk mencari nilai total Effort
6	<i>User</i> memasukkan nilai effort multiplier (RELY, DATA, CPLX, RUSE, DOCU, TIME, STOR, PVOL, ACAP, PCAP, PCON, APEX, PLEX, LTEX, TOOL, SITE)	Sistem akan menghitung semua nilai yang dimasukkan oleh user untuk mencari nilai total effort multiplier dan person month.
7	<i>User</i> memilih metode yang digunakan dan memasukkan nilai yang dibutuhkan oleh metode tersebut	Sistem akan menghitung nilai total per <i>activity</i> yang telah dimasukkan oleh <i>user</i> untuk mendapatkan nilai total per <i>activity</i>
8	<i>User</i> memasukkan gaji yang dibutuhkan dan <i>profit</i> yang dibutuhkan	Sistem akan menghitung total per <i>activity</i> dengan gaji dan <i>profit</i> yang dimasukkan oleh <i>user</i> . sehingga sistem akan menghasilkan nilai total proyek dan <i>profit</i> yang didapatkan dari proyek yang telah dihitung.

### C. *Sequence Diagram*

*Sequence diagram* aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi kedua yang dijelaskan pada gambar 4.12. *Sequence diagram* menjelaskan lebih detail lagi terkait *activity diagram* yang telah dibuat.



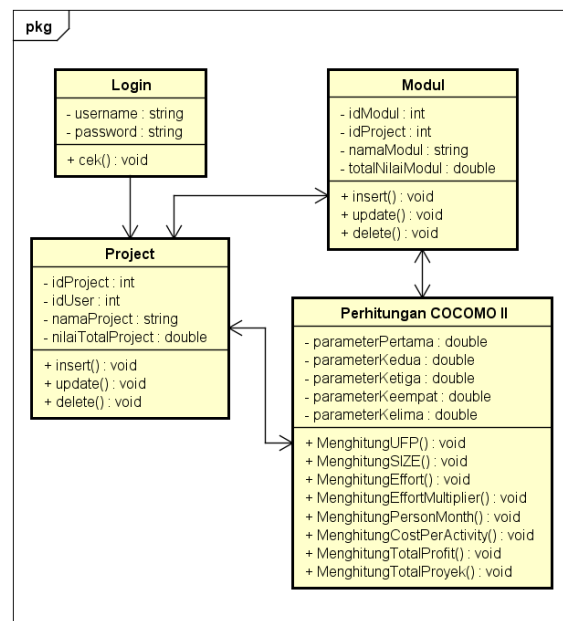
### C.1 Sequence Diagram Otomatisasi Rumus COCOMO II



Gambar 4. 12 Sequence Diagram Otomatisasi Rumus COCOMO II

#### D. Class Diagram

*Class diagram* aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi kedua yang dijelaskan pada gambar 4.3. *Class diagram* menjelaskan terkait *function* dan data yang dibutuhkan pada *sprint* iterasi kedua.



Gambar 4. 12 Class Diagram Iterasi kedua

## E. Hasil Desain I/O

Hasil desain I/O adalah tampilan aplikasi yang akan dibuat pada aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak pada *sprint* iterasi kedua. Penjelasan lebih detail terkait hasil desain I/O terdapat pada lampiran 3.

### E.1 Tampilan Perhitungan Langkah Awal

Halaman perhitungan COCOMO II langkah pertama merupakan halaman yang digunakan untuk menghitung nilai total UFP (Unadjusted Function Points) setiap modul. Langkah awal pengguna harus mengisi beberapa parameter khusus dari COCOMO II seperti pada gambar 4.13.

The screenshot shows the Universitas Dinamika software interface. On the left, there is a sidebar with the logo and the text 'UNIVERSITAS Dinamika', 'Module Project 1', and 'Infomation'. Below this, several metrics are listed: Total UFP (600), Size In SLOC (0), Scale Factor (0), Total Effort Multiplier (0), Person Month (0), and total Method (0). The main area contains a table for 'Function Type' and 'Complexity Weight' with columns for 'Low', 'Average', and 'Height'. The table has the following data:

Function Type	Complexity Weight		
	Low	Average	Height
Internal Logical Files	11	0	0
External Interfaces Files	47	0	0
External Inputs	54	7	0
External Outputs	12	4	0
External Inquiries	10	0	0
<b>TOTAL</b>	<b>600</b>		

There are 'Hitung' and 'Next' buttons at the bottom of the table. A copyright notice '© 2019 LUG by Maerka' is visible at the bottom left of the interface.

Gambar 4. 12 Perhitungan Langkah Pertama

## F. *Sprint Review* Iterasi Kedua

Hasil review yang didapatkan setelah demo kepihak pemerintahan untuk *sprint* iterasi kedua dijelaskan pada tabel berikut.

Tabel 4. 5 Tabel *Sprint Review* Iterasi Kedua

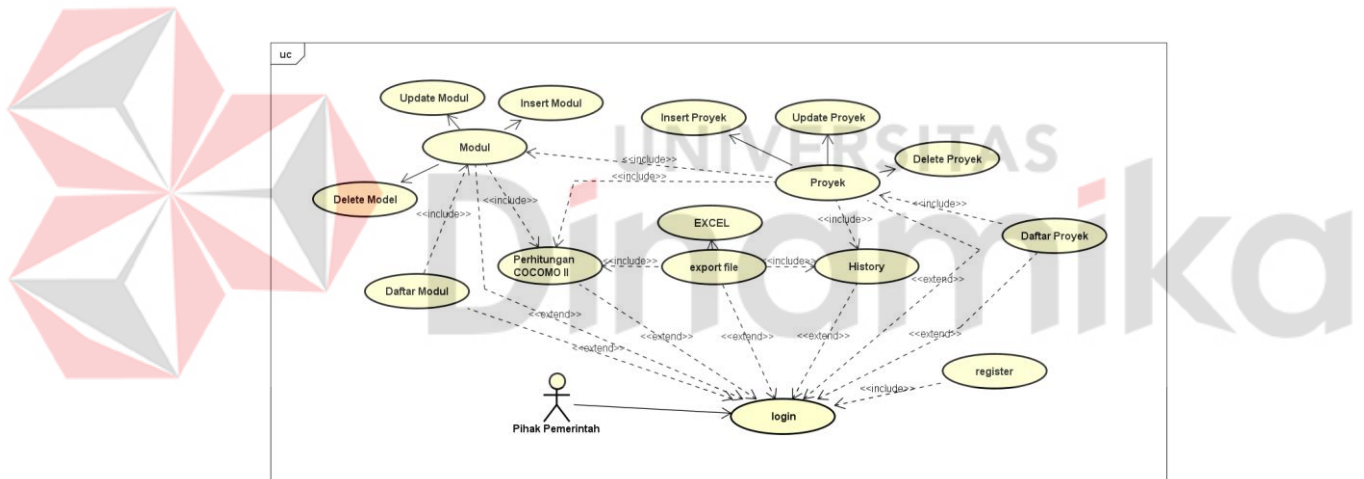
No	Fungsional	Hasil Review
1	Otomatisasi Rumus COCOMO II	<ol style="list-style-type: none"> <li>1. Langkah awal berhasil mendapatkan nilai total UFP</li> <li>2. Langkah kedua berhasil mendapatkan nilai KSLOC</li> <li>3. Langkah ketiga berhasil mendapatkan nilai <i>Scale Factor</i></li> <li>4. Langkah keempat berhasil mendapatkan nilai <i>Effort Multiplier</i></li> <li>5. Langkah kelima berhasil mendapatkan nilai total <i>activity</i></li> <li>6. keenam berhasil mendapatkan nilai total gaji personil</li> <li>7. Langkah ketujuh berhasil mendapatkan nilai total <i>personnel direct cost</i></li> <li>8. Langkah kedelapat berhasil mendapatkan nilai total <i>direct cost</i> dengan <i>profit</i> 10%</li> <li>9. Langkah ke sembilan berhasil mendapatkan nilai <i>non personel direct cost</i></li> <li>10. Langkah kesepuluh berhasil menghitung nilai total proyek dengan <i>tax</i> 10 %</li> </ol>

### 4.1.3 Sprint Iterasi Ketiga

*Sprint* iterasi ketiga menghasilkan fitur *Report Excel proyek*, dan *save history project user*. Hasil dari *sprint* iterasi ketiga akan di-*review* pada tahap *sprint review* iterasi ketiga.

#### A. Usecase System

*Usecase* pada aplikasi estimasi penentu biaya proyek perangkat lunak untuk *sprint* iterasi ketiga yang dijelaskan pada gambar 4.13. *Usecase* tersebut menjelaskan tentang beberapa proses utama dan sub proses yang ada dalam *sprint* iterasi ketiga yang terdiri dari *Activity Report excel project*, *Activity save history project user*.

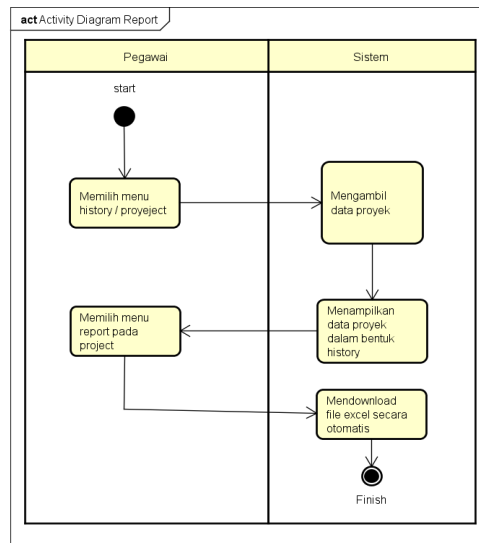


Gambar 4. 13 *Usecase System* Iterasi Ketiga

#### B. Activity Diagram

*Activity diagram* aplikasi estimasi penentuan estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi pertama yang dijelaskan pada gambar 4.13, dan L4. 1. *Activity diagram* menjelaskan lebih *detail* terkait *usecase* yang telah dibuat.

### B.1 Diagram Activity Report Excel Proyek



Gambar 4. 13 Activity Report Excel Proyek

Pada gambar 4.13 akan dijelaskan lebih detail melalui tabel 4.6 yaitu tabel *flow of event report project* yang menjelaskan bagaimana alur sistem *report* proyek pada aplikasi dan parameter apa saja yang akan dibutuhkan oleh *user*.

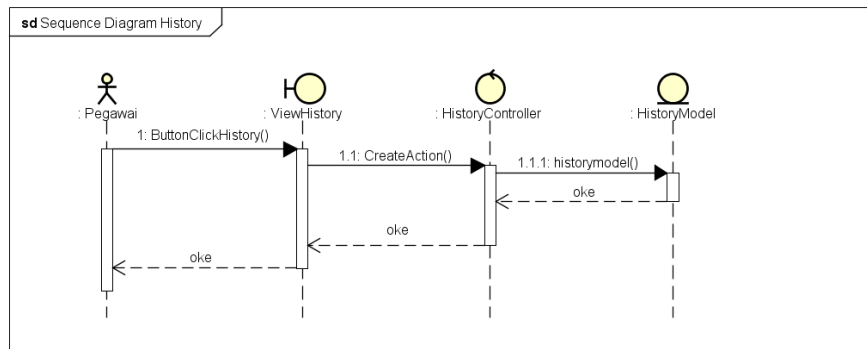
Tabel 4. 6 Flow of Event Report Project

Deskripsi	Login awal untuk masuk ke aplikasi	
Kondisi Awal	Pegawai sudah memiliki akun yang terdaftar	
Kondisi Akhir	Pegawai mendapatkan <i>report</i> excel	
Aliran Kejadian Utama		
No.	<i>User</i>	Sistem
1.	<i>User</i> memilih menu proyek kemudian menekan tombol <i>report</i>	Sistem melakukan menampilkan halaman report proyek dan kemudian sistem akan secara otomatis mendownload excel

### C. Sequence Diagram

*Sequence diagram* aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi pertama yang dijelaskan pada gambar 4.14, dan L4. 2. *Sequence diagram* menjelaskan lebih detail lagi terkait *activity diagram* yang telah dibuat.

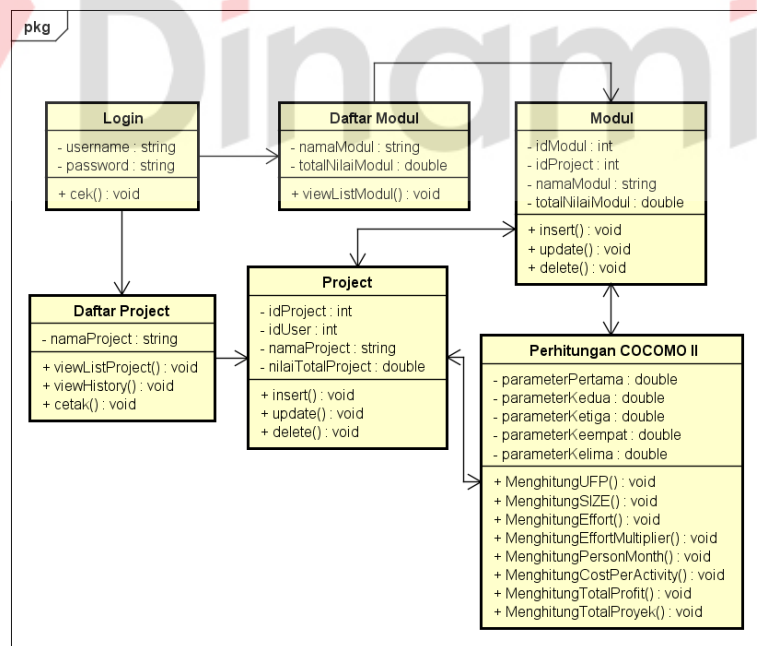
### C.1 Sequence Diagram Report Excel Proyek



Gambar 4. 15 Sequence Diagram Report Excel Proyek

### D. Class Diagram

*Class diagram* aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak untuk *sprint* iterasi ketiga yang dijelaskan pada gambar 4.16. *Class diagram* menjelaskan terkait *function* dan data yang dibutuhkan pada *sprint* iterasi ketiga.



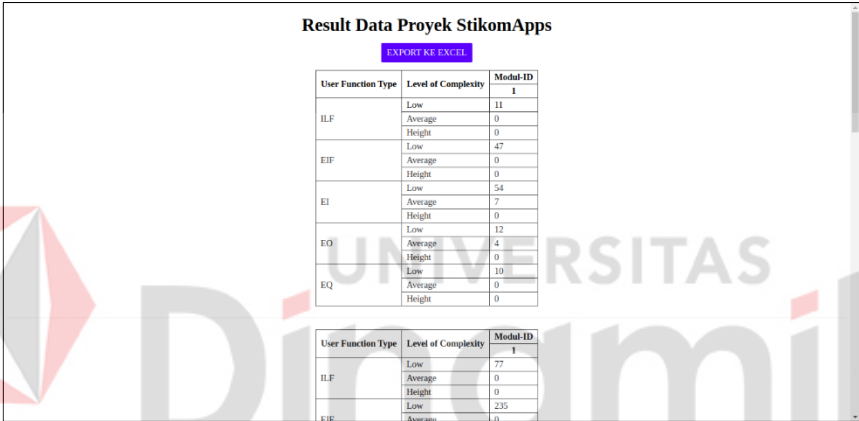
Gambar 4.16 Class Diagram Iterasi Ketiga

## E. Hasil Desain I/O

Hasil desain I/O adalah tampilan aplikasi yang akan dibuat pada aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak pada *sprint* iterasi ketiga. Penjelasan lebih detail terkait hasil desain I/O terdapat pada lampiran 4.

### E.1 Tampilan *Report* Excel dan PDF Proyek

Halaman *report* excel dan PDF proyek merupakan halaman yang digunakan untuk melihat *report detail* dalam bentuk tabel dan dapat *download* dalam bentuk *file* excel dan PDF seperti gambar 4.17.



The screenshot shows a web interface with a table titled "Result Data Proyek StikomApps". Above the table is a purple button labeled "EXPORT KE EXCEL". The table has three columns: "User Function Type", "Level of Complexity", and "Modul-ID". The data is organized into two sections, each with a sub-table header.

User Function Type	Level of Complexity	Modul-ID
		1
ILF	Low	11
	Average	0
	Height	0
EIF	Low	47
	Average	0
	Height	0
EI	Low	54
	Average	7
	Height	0
EG	Low	12
	Average	4
	Height	0
EQ	Low	10
	Average	0
	Height	0

User Function Type	Level of Complexity	Modul-ID
		1
ILF	Low	77
	Average	0
	Height	0
EIF	Low	235
	Average	0
	Height	0

Gambar 4. 17 Tampilan *Report* Excel Project

## F. *Sprint Review* Iterasi Ketiga

Hasil *review* yang didapatkan setelah demo ke pihak pemerintahan untuk *sprint* iterasi ketiga dijelaskan pada tabel berikut.

Tabel 4. 7 Tabel *Sprint Review* Iterasi Ketiga

No	Fungsional	Hasil Review
1	Fitur <i>export</i> excel & PDF	1. Excel & PDF dapat di- <i>download</i> dan bisa di buka dalam bentuk excel
2	Fitur <i>history</i> proyek	2. Fitur <i>history</i> berhasil menyimpan data <i>history</i> <i>project</i>

#### 4.1.4 *Sprint* Iterasi Keempat

##### A. *Deployment Project*

*Deployment project* adalah memasukkan hasil *production project* aplikasi penentu estimasi biaya proyek perangkat lunak ke dalam *Virtual Private Server* (VPS) dan dengan domain tertentu agar bisa di akses oleh banyak orang. Hasil dari tahap *deployment project* tersebut bentuk *link* berikut : <http://cocomoi.damartana.com>. Kebutuhan hardware dan software terdapat pada lampiran 5.

##### B. *Sprint Review* Iterasi Keempat

Hasil *review* yang didapatkan setelah demo kepihak pemerintahan untuk *sprint* iterasi keempat dijelaskan pada tabel berikut.

Tabel 4. 8 Tabel *Sprint Review* Iterasi Keempat

No	Fungsional	Hasil Review
1	<i>Deployment Project</i>	1. Aplikasi bisa diakses lewat <i>website</i> melalui <i>link</i> yang tertera. Dan semua fitur telah berfungsi

#### 4.2 Pembahasan *Result* Perhitungan COCOMO II

Tabel 4. 9 Dataset Pemerintahan

Modul-ID	Deskripsi Proyek	Provinsi	Baha Pemrogramman
1	E-SDM DPRKPCCKTR. Website of Public Housing and Residential Areas, Creative Works and Spatial Planning in Surabaya	East Java	PHP, MySQL
2	Archive of DLH. Website of Public Environmental Services in Surabaya	East Java	PHP, MySQL
3	Management of HRM. Website of Division IV of PT Wijaya Karya	East Java	PHP, MySQL
4	Building Intensity. Website of Public Housing and Residential Areas, Creative Works and Spatial Planning in Surabaya	East Java	PHP, MySQL



Dalam tahap pembahasan berikut menggunakan 4 dataset pemerintahan yang telah diuji coba atau dihitung menggunakan metode COCOMO II seperti tabel 4.9.

Tabel 4. 10 *Result input-an parameter UFP*

<i>User Function Type</i>	<i>Level of Complexity</i>	<b>Modul-ID</b>			
		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
ILF	<i>Low</i>	11	7	6	5
	<i>Average</i>	0	0	0	0
	<i>Height</i>	0	0	0	0
EIF	<i>Low</i>	47	18	26	22
	<i>Average</i>	0	0	0	0
	<i>Height</i>	0	0	0	0
EI	<i>Low</i>	54	44	36	23
	<i>Average</i>	7	2	1	0
	<i>Height</i>	0	0	0	0
EO	<i>Low</i>	12	11	5	5
	<i>Average</i>	4	8	3	3
	<i>Height</i>	0	0	0	1
EQ	<i>Low</i>	10	7	4	5
	<i>Average</i>	0	0	0	0
	<i>Height</i>	0	0	0	0

Tabel 4.10 adalah gambar *result input-an* untuk mencari nilai total UFP. Sebagai contoh modul 1 seperti pada gambar 4.10. User Function Types = ILF dan *Level of Complexity* = *low*, kemudian dikalikan dengan bobot yang telah di tentukan yaitu 7 pada ILF *level low* ( $11 \times 7 = 77$ ) seperti pada tabel 4.14.

Tabel 4. 11 *Result nilai total UFP*

<i>User Function Type</i>	<i>Level of Complexity</i>	<b>Modul-ID</b>			
		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
ILF	<i>Low</i>	77	49	42	35
	<i>Average</i>	0	0	0	0
	<i>Height</i>	0	0	0	0
EIF	<i>Low</i>	235	90	130	110
	<i>Average</i>	0	0	0	0
	<i>Height</i>	0	0	0	0
EI	<i>Low</i>	162	132	108	69
	<i>Average</i>	28	8	4	0
	<i>Height</i>	0	0	0	0
EO	<i>Low</i>	48	44	20	20
	<i>Average</i>	20	40	15	15
	<i>Height</i>	0	0	0	7
EQ	<i>Low</i>	30	21	12	15
	<i>Average</i>	0	0	0	0
	<i>Height</i>	0	0	0	0
<b>TOTAL (UFP)</b>		600	384	331	271

Tabel 4.11 adalah gambar *result* nilai total UFP yang telah dikalikan dengan bobot dari rumus COCOMO II sebagai contoh modul 1. *User Function Types* = ILF, kemudian  $ILF\ UFP = 77 + 0 + 0 = 77$  seperti tabel 4.11. Kemudian menjumlahkan semua nilai UFP.  $UFP = ILF\ UFP + EIF\ UFP + EI\ UFP + EO\ UFP + EQ\ UFP$ . Sebagai contoh modul 1 yaitu,  $UFP = 77 + 235 + 190 + 68 + 30 = 600$  seperti tabel 4.11.

Tabel 4. 12 *Result* nilai total KSLOC

Modul-ID	UFP	SLOC <i>Conversion Ratio</i>	Size in SLOC	Size in KSLOC
1	600	80	48000	48
2	384	80	30720	30.72
3	331	80	26480	26.48
4	271	80	21680	21.68

Tabel 4.11 adalah tabel *result* nilai total KSLOC dengan rumus *size* (SLOC) =  $UFP \times SLOC$  dan nilai SLOC sesuai *input-an* user yaitu *3rd generation language* (80). Sebagai contoh modul 1.  $SLOC = 600 \times 80 = 48,000$ . Kemudian nilai SLOC dikonversikan kedalam Kilo SLOC (KSLOC) =  $48,000 / 1000 = 48.00$ .

Tabel 4. 13 *Result* Nilai *Scale Factor*

<i>Scale Factor</i>	Modul-ID			
	1	2	3	4
PREC	4.96	3.72	2.48	3.72
FLEX	2.03	3.04	3.04	2.03
RESL	4.24	2.83	4.24	5.65
TEAM	3.29	4.38	3.29	3.29
PMAT	4.68	4.68	6.24	4.68

Tabel 4.13 adalah gambar *result input-an* user untuk *Scale Factor* dengan bobot dari rumus COCOMO II. Sebagai contoh modul 1 seperti pada tabel 4.16.

Tabel 4. 14 *Result* Nilai *Effort*

Modul-ID	E
1	1.102
2	1.0965
3	1.1029

Modul-ID	E
4	1.1037

Tabel 4.13 adalah *result* nilai total E dengan rumus  $E = B + 0.01 \times \text{total SF}$ . Nilai B = 0.91 dari rumus COCOMO II. Sebagai contoh modul 1 untuk mengambil nilai SF seperti pada gambar 4.40.  $E = 0.91 + 0.01 \times (4.96 + 2.03 + 4.24 + 3.29 + 4.68) = 1.102$  hasil seperti tabel 4.13.

Tabel 4. 15 *Result* Input dan Total *Effort Multiplier*

Scale Factor	Modul-ID			
	1	2	3	4
RELY	1	1	1	1.1
DATA	0.9	1.28	0.9	1
CPLX	1	1.17	1.17	1
RUSE	1.07	1	1	0.95
DOCU	0.81	1	0.91	1
TIME	1	1.11	1.11	1.29
STOR	1	1	1	1
PVOL	1.15	1	1	0.87
ACAP	0.85	0.85	0.85	0.85
PCAP	1	0.88	1	1
PCON	0.81	0.81	0.81	0.9
APEX	0.81	0.88	0.81	0.88
PLEX	0.85	0.85	1	0.91
LTEX	0.84	0.91	0.91	0.91
TOOL	0.9	0.9	0.9	1
SITE	1	1.09	1	1.09
SCED	1	1	1	1
<b>TOTAL (EM)</b>	0.3215	0.6725	0.4858	0.7127

Tabel 4.14 adalah *input-an* dan total *effort multiplier*. Rumus untuk mendapatkan nilai total EM adalah dengan mengalikan semua *input-an* EM sebagai contoh modul 1.  $EM = 1 \times 0.9 \times 1 \times 1.07 \times 0.81 \times 1 \times 1 \times 1.15 \times 0.85 \times 1 \times 0.81 \times 0.81 \times 0.85 \times 0.84 \times 0.9 \times 1 \times 1 = 0.3215$  seperti pada tabel 4.14.

Tabel 4. 16 *Result* total PM

Modul-ID	Size (KSLOC)	E	Total EM	Hasil
1	48	1.102	0.3215	67.33
2	30.72	1.0965	0.6725	84.53
3	26.48	1.1029	0.4858	52.98
4	21.68	1.1037	0.7127	62.49

Tabel 4.15 adalah *result* nilai total TM yang didapatkan dari rumus  $PM = A \times KSLOC \text{ pangkat } E \times EM$ . Nilai  $A = 2.94$ . Sebagai contoh modul 1  $PM = 2.94 \times 48 \text{ pangkat } 1.102 \times 0.3215 = 67.33$  seperti tabel 4.15.

Tabel 4. 17 *Result Input* dan *Total Activity*

No	Activity	%	Modul-ID			
			1	2	3	4
1	<i>Requirement</i>	1.17	0.79	0.99	0.62	0.73
2	<i>Specifications</i>	6.75	4.54	5.71	3.58	4.22
3	<i>Design</i>	5.57	3.75	4.71	2.95	3.48
4	<i>Implementation</i>	55.65	37.47	47.04	29.48	34.78
5	<i>Integration Testing</i>	6.42	4.32	5.43	3.4	4.01
6	<i>Acceptance &amp; deployment</i>	5.6	3.77	4.73	2.97	3.5
7	<i>Project Management</i>	2.55	1.72	2.16	1.35	1.59
8	<i>Configuration Management</i>	3.58	2.41	3.03	1.9	2.24
9	<i>Quality Assurance</i>	0.66	0.44	0.56	0.35	0.41
10	<i>Documentations</i>	9.76	6.57	8.25	5.17	6.1
11	<i>Training &amp; Support</i>	0.62	0.42	0.52	0.33	0.39
12	<i>Evaluation &amp; Testing</i>	1.67	1.12	1.41	0.88	1.04
<b>Total</b>			67.33	84.53	52.98	62.49

Tabel 4.16 adalah gambar *result* dan *input-an Activity*. Sebagai contoh modul 1. *Activity* :  $Requirement = 1.17\% \times 67.33 = 0.79$ . 1.17 adalah *input-an user* seperti gambar 4.22, 67.33 adalah hasil PM dari modul 1 seperti pada tabel 4.19. dan seterusnya sampai *activity* ke 12. Kemudian mentotal nilai per *activity* yang hasilnya akan dibandingkan dengan nilai total PM. Dan harus nilai harus sama. Seperti tabel 4.16 dan tabel 4.15.

Tabel 4.17 adalah gambar *result* dan *input-an salary*. *Salary* dibagi menjadi 2 tipe yaitu *payment rate for private seactor* dan *payment rate for public sector*. Untuk *payment rate for private seactor* 100% tidak ada potongan apapun . karna *payment rate private* digunakan untuk internal pemerintahan. Sedangkan untuk *vendor payment rate public sector* terdapat potongan sebesar 30%. Jadi *vendor* mendapatkan 70% dari nominal awal yang telah dimasukkan.

Tabel 4. 18 Salary

<b>Position In Project Team</b>	<b>Minimum Requirement</b>	<b>Payment rate Public</b>	<b>Payment Private</b>	<b>Rate</b>
Project Management	Minimal Undergraduate and Experience > 4 years	18.250.000	12.775.000	
System Analyst	Minimal Undergraduate and Experience > 3 years	12.000.000	8.400.000	
Programmer	Experience 1- 2 years	11.000.000	7.700.000	
System Testing	Experience 1- 2 years	9.300.000	6.510.000	
Technical Support	Experience 1- 2 years	9.300.000	6.510.000	
Documenter	Experience 1- 2 years	5.000.000	3.500.000	

Tabel L6.1 adalah gambar *result* dan *input-an Cost per Activity*. Dalam tahap ini dibutuhkan hasil *total per activites* yang terletak pada tabel 4.16 dan membutuhkan data *salary (index)* pada tabel 4.17 dengan nama kolom *index*. Sebagai contoh modul 1. *Activity : Requirement = 12,000,000x 0.923 x 0.79 = 8,750,000*. 12,000,000 adalah *input-an salary* sesuai dengan dengan *jobdesc*, 0.923 adalah nilai *default* dari rumus COCOMO II, 0.79 adalah hasil *total per activity* seperti pada tabel 4.16 dari *activity* pertama dan seterusnya sampai *activity* ke 12. Kemudian menjumlahkan nilai semua *salary per activities*. Seperti tabel L6. 1.

Tabel 4. 19 *Personel Direct Cost*

<b>Project ID</b>	<b>Personel Direct Cost before Profit (IDR)</b>	<b>Profit (IDR)</b>	<b>Personel Direct Cost (IDR)</b>
1	653.429.543	65.342.954	718.772.497
2	820.579.305	82.057.931	902.637.236
3	359.949.026	35.994.903	395.943.929
4	424.560.488	42.456.049	467.016.537

Tabel 4.18 adalah *result* dari *personel direct cost* dan *input-an profit* sesuai keinginan user. dalam studi kasus kali ini menggunakan profit 10%. Sebagai contoh modul 1, total *personel direct project* 1 adalah 653,429,543 x 10% = 65,342,954.

Kemudian menjumlahkan total *personel direct cost* dan *profit* yang dihasilkan.  $653,429,543 + 65,342,954 = 718,772,497$ . Hasil seperti pada tabel 4.18.

Tabel L7. 1 adalah *result inputan direct cost non personel*. Dalam tahap ini adalah hanya menjumlahkan semua inputan biaya *non personnel* oleh *user* untuk mendapatkan nilai total biaya *non personnel*.

Tabel 4. 20 *Personnel With Profit*

<i>Modul</i>	<i>Personnel</i>	<i>Profit 10%</i>	<i>Personnel+Profit</i>
1	653.429.543	65.342.954	718.772.497
2	820.579.305	82.057.931	902.637.236
3	514.320.060	51.432.006	565.752.066
4	606.496.378	60.649.638	667.146.016

Tabel 4. 21 *Owner Estimate Cost*

<i>Non-Personnel</i>	<i>Cost Estimate</i>	<i>Tax 10%</i>	<i>Cost Estimate + Tax</i>
43.250.000	762.022.497	76.202.250	838.244.747
35.660.000	938.297.236	93.829.724	1.032.126.960
38.850.000	604.602.066	60.460.207	665.062.273
47.210.000	714.356.016	71.435.602	785.791.618

Tabel 4.19 dan 4.20 adalah result dari *input-an tax* dan total biaya proyek.

Dalam tahap ini menjumlahkan biaya *personnel*, *profit*, *non personnel* dan *tax*. *tax* pada *input-an user* adalah 10%. Bisa dibuat formula seperti berikut. *Cost estimate* = Biaya *Personel* dengan *profit* + *non personnel*. kemudian *tax* = *cost estimate* x 10%. *Cost estimate* ( $718,772,497 + 43,250,000 = 762,022,497$ ), *tax* ( $762,022,497 \times 10\% = 76,202,250$ ) dan yang terakhir adalah menjumlahkan *tax* dan *estimate cost*. Total proyek =  $762,022,497 + 76,202,250 = 838,244,747$ .

### 4.3 Evaluasi

Tahap evaluasi melakukan sebuah perbandingan dari sistem perhitungan yang telah dirancang dalam sebuah aplikasi dengan sistem perhitungan manual

(Excel). Perbandingan tersebut digunakan mengetahui apakah hasil perhitungan dari sistem yang telah dirancang dalam aplikasi sesuatu atau sama dengan perhitungan manual (Excel).

Input UFP		Output UFP			
Function Type	Complexity Weight			Total	
	Low	Average	High		
Internal Logical Files	11	0	0	0	
External Interfaces Files	47	0	0	0	
External Inputs	54	7	0	0	
External Outputs	12	4	0	0	
External Inquiries	10	0	0	0	
<b>Total</b>				<b>600</b>	

Input SLOC		Output SLOC	
Programming Language	80	UFP	600
		SLOC	80
		Size in SLOC	48000
		Size in KSLOC	48

Gambar 4.18 Manual COCOMO II

**Result Data Proyek Pemerintahan 2018**

EXPORT KE EXCEL

Modul	Personnel	Profit 10%	Personnel + Profit	Non-Personnel	Cost Estimate	Tax 10%	Cost Estimate + Tax
1	Rp 653.429.543	Rp 65.342.954	Rp 718.772.497	Rp 43.250.000	Rp 762.022.497	Rp 76.202.250	Rp 838.224.747
2	Rp 820.579.305	Rp 82.057.931	Rp 902.637.236	Rp 35.660.000	Rp 938.297.236	Rp 93.829.724	Rp 1.032.126.960

Position In Project Team	Minimum Requirement	Payment Rate For Private Sector in Person Month Rate(PMR) Requirement Modul 1	Payment Rate For Private Sector in Person Month Rate(PMR) Requirement Modul 2
Project Management	Minimal Undergraduate And Experience > 4 years	Rp 18.250.000	Rp 18.250.000
System Analyst	Minimal Undergraduate And Experience > 3 years	Rp 12.000.000	Rp 12.000.000
Programmer	Experience 1-2 years	Rp 11.000.000	Rp 11.000.000
System Testing	Experience 1-2 years	Rp 9.300.000	Rp 9.300.000
Technical Support	Experience 1-2 years	Rp 9.300.000	Rp 9.300.000
Documenter	Experience 1-2 years	Rp 5.000.000	Rp 5.000.000

Gambar 4.19 Sistem Aplikasi COCOMO II

Dari hasil uji coba pada gambar 4.18 dan 4.19 dapat disimpulkan bahwa hasil perhitungan antara aplikasi yang dirancang dan sistem perhitungan manual memiliki nilai *output* yang sama. Nilai yang dihasilkan dari kedua perhitungan tersebut adalah Rp. 838, 224,747 pada contoh modul.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Hasil dari tugas akhir ini adalah sebuah aplikasi penentu estimasi biaya proyek pengembangan perangkat lunak menggunakan metode COCOMO II berbasis website. Aplikasi ini menghasilkan nilai total proyek dan profit dari setiap proyek yang didaftar dan dihitung sebagai acuan sebelum proyek ditenderkan atau sebaliknya.

Aplikasi penentu estimasi biaya pengembangan proyek perangkat lunak menggunakan metode COCOMO II berbasis website ini telah diuji dengan pengujian *black box testing* dan dataset dari pihak pemerintahan. Hasil evaluasi aplikasi ini memiliki tingkat keberhasilan hingga 100% dari beberapa *test case*. Sehingga aplikasi telah sesuai dengan kebutuhan penggunaannya. Dan berdasarkan studi kasus yang telah diuji coba bahwa perhitungan excel dengan aplikasi sudah memunculkan nilai total proyek yang sama.

#### 5.1 Saran

Aplikasi penentu estimasi biaya pengembangan proyek perangkat lunak menggunakan metode COCOMO II berbasis website yang berguna untuk menghitung nilai total proyek sebelum proyek ditenderkan atau sebaliknya sebagai acuan harga estimasi.

Adapun saran untuk peneliti selanjutnya adalah :

1. Aplikasi masih dalam bentuk *website*, sehingga bisa dikembangkan lagi menjadi beberapa *platform* lain seperti android atau ios.



2. Aplikasi masih menggunakan 1 sub model COCOMO yaitu *Post Architecture*.  
Sehingga bisa dikembangkan lagi dengan menggunakan 3 sub model COCOMO II.



UNIVERSITAS  
**Dinamika**

## DAFTAR PUSTAKA

- Admin. (2019, juli 11). *Fresh Consultant*. Diambil kembali dari Penyusunan Harga Perkiraan Sendiri (Owner Estimate): <https://www.freshconsultant.co.id/2019/06/21/penyusunan-harga-perkiraan-sendiri-owner-estimate>
- Alexandra, J. (2017, Mei 8). *Agile Development Methods*. Diambil kembali dari [sis.binus.ac.id](https://sis.binus.ac.id): <https://sis.binus.ac.id/2017/05/08/agile-development-methods/>
- Aryadita, H. (2018, Februari 18). *Apakah yang dimaksud dengan metode SCRUM*. Diambil kembali dari DICTIO: <https://www.dictio.id/t/apakah-yang-dimaksud-dengan-metode-scrum/302>
- Barang/Jasa, B. P. (2019, Agustus 1). *Website Resmi Pemerintahan Kabupaten Buleleng*. Diambil kembali dari Pengertian HPS dan Cara Menyusunnya: <https://www.bulelengkab.go.id/detail/artikel/pengertian-hps-dan-cara-menyusunnya-23>
- Dillak, R. Y. (2012). Estimasi Biaya Perangkat Lunak Menggunakan Improved Backpropagation. *Seminar Nasional Informatika 2012*, 1979-2328.
- Fajrin, R. A. (2018, Agustus 23). *Evolusi Development Team dalam Scrum*. Diambil kembali dari [medium](https://medium.com/@rendyariesfajrin/evolusi-development-team-dalam-scrum-6fb96ab8a2cf): <https://medium.com/@rendyariesfajrin/evolusi-development-team-dalam-scrum-6fb96ab8a2cf>
- Febrina Putri Laksamana, A. R. (2018). Evaluasi Biaya Pengembangan Sistem Informasi Pengelolaan Arsip Surat (SIPAS) Menggunakan Function Point dan Object Point (Studi Kasus : PT Sekawan Media Informatika). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* , 2692-2701.
- Firdaus, M. A. (2017). Implementasi Kerangka Kerja SCRUM Pada Manajemen Pengembangan Sistem Informasi. *Seminar Nasional Teknologi Informasi dan Multimedia*, 1-5.
- Haviluddin. (2011). Memahami Penggunaan UML (Unified Modelling Language) . *Jurnal Informatika Mulawarman*, 3.
- Heriyanto, Y. (2018). PERANCANGAN SISTEM INFORMASI RENTAL MOBIL BERBASIS WEB PADA PT.APM RENT CAR . *Jurnal Intra-Tech* , 69.

Khoiro, M. N. (2018). Evaluasi Biaya Pengembangan Perangkat Lunak Dengan Menggunakan Metode Cocomo II (Studi Kasus: PT DOT Indonesia). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3220-3229.

Partogi, J. (2016, Agustus 1). *My Role As a Scrum Master*. Diambil kembali dari Modern Management: <https://medium.com/modern-management/scrum-master-bukan-manajer-proyek-ataupun-technical-leader-78a6339066ff>

Perpres. (2012, - -). *Peraturan Presiden Republik Indonesia No 70 Tahun 2012*. Diambil kembali dari BPKP: <http://www.bpkp.go.id/uu/filedownload/5/91/2253.bpkp>

Prasetya Muhamad Aditya, A. R. (2018). Perhitungan Biaya Pembangunan Sistem Informasi Pariwisata Kota. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4785-4793.

Purwanto, D. (2017). Aplikasi Penentu Harga Perkiraan Sendiri Proyek Pengembangan Perangkat Lunak Kepemerintahan Berbasis Website. -, -.

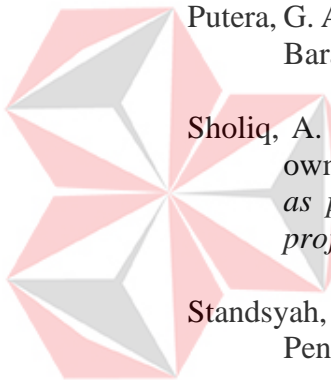
Putera, G. A. (2019). Model Penyusunan Harga Perkiraan Sendiri (HPS) Pengadaan Barang/Jasa Pemerintahan Di Provinsi Bali. *Jurnal Spektran*, 204-211.

Sholiq, A. P. (2018). Implementing COCOMO II as personnel direct cost in an owner estimate cost model for software project. *Implementing COCOMO II as personnel direct cost in an owner estimate cost model for software project*, 1-16.

Standisyah, R. E. (2017). Implementasi phpmyadmin pada Rancangan Sistem Pengadaan Administrasi. *Jurnal UJMC*, 39 - 40.

Suendri. (2018). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan) . *ALGORITMA: Jurnal Ilmu Komputer dan Informatika* , 2.

Zulkifli. (2018). Ancang bangun website e-learning dengan pemodelan uml (studi kasus di stkip muhammadiyah muara bungo) . *Journal of information technology and computer science (intecom)* , 168.



Dinamika