

**PENGEMBANGAN SISTEM MONITORING PRODUKSI DAN  
PENJUALAN BATIK MADURA BERBASIS *WEBSITE*  
STUDI KASUS: CANTENG KONENG**

**PROPOSAL SKRIPSI**

Disusun oleh:  
Rezky Nurfahriza  
NIM: 165150207111005



TEKNIK INFORMATIKA  
TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2020

## DAFTAR ISI

DAFTAR ISI .....	ii
DAFTAR TABEL.....	iv
DAFTAR GAMBAR.....	v
DAFTAR LAMPIRAN .....	vi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penelitian .....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Penelitian Terdahulu.....	5
2.2 Rekayasa Perangkat Lunak .....	9
2.3 Pengembangan Perangkat Lunak .....	9
2.3.1 <i>Prototyping</i> .....	10
2.4 <i>Object Oriented</i> .....	13
2.5 UML( <i>Unified Modeling Language</i> ).....	13
2.5.1 <i>Use Case Diagram</i> .....	13
2.5.2 <i>Class Diagram</i> .....	15
2.5.3 <i>Use Case Scenario</i> .....	16
2.5.4 <i>Sequence Diagram</i> .....	16
2.6 Bahasa Pemrograman.....	19
2.6.1 PHP .....	19
2.6.2 Javascript .....	19
2.7 MySQL.....	19
2.8 <i>Framework</i> .....	20

2.8.1 <i>Codeigniter</i> .....	20
2.8.2 <i>Bootstrap</i> .....	20
2.9 MVC ( <i>Model View Controler</i> ) .....	20
2.10 Google Analytics .....	21
2.11 Pengujian Perangkat Lunak .....	21
2.11.1 <i>White-box Testing</i> .....	21
2.11.2 <i>Black-box Testing</i> .....	23
2.11.2 <i>Usability Testing</i> .....	23
BAB 3 METODOLOGI PENELITIAN .....	25
3.1 Studi Literatur .....	26
3.2 Analisis Kebutuhan .....	26
3.3 Perancangan Sistem.....	27
3.4 Evaluasi <i>Prototype</i> oleh <i>Client</i> .....	27
3.5 Membangun Sistem .....	27
3.6 Pengujian .....	28
3.7 Evaluasi Sistem oleh <i>Client</i> .....	28
3.8 Penutup.....	28
DAFTAR REFERENSI .....	29

## DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu .....	6
Tabel 2.1 Penelitian terdahulu (lanjutan) .....	7
Tabel 2.1 Penelitian terdahulu (lanjutan) .....	8
Tabel 2.1 Penelitian terdahulu (lanjutan) .....	9
Tabel 2.2 Simbol dan deskripsi <i>use case diagram</i> .....	14
Tabel 2.2 Simbol dan deskripsi <i>use case diagram</i> (lanjutan).....	15
Tabel 2.3 Simbol dan deskripsi <i>class diagram</i> .....	16
Tabel 2.4 Simbol dan deskripsi <i>sequence diagram</i> .....	17
Tabel 2.4 Simbol dan deskripsi <i>sequence diagram</i> (Lanjutan) .....	18
Tabel 2.4 Simbol dan deskripsi <i>sequence diagram</i> (Lanjutan) .....	19

## DAFTAR GAMBAR

Gambar 2.1 Model <i>prototyping</i> .....	11
Gambar 2.2 Model <i>evolutionary prototyping</i> .....	12
Gambar 2.3 Model <i>requirement prototyping</i> .....	12
Gambar 2.4 <i>Use case diagram</i> .....	14
Gambar 2.5 <i>Class diagram</i> .....	15
Gambar 2.6 <i>Sequence diagram</i> .....	17
Gambar 2.7 <i>Flow graph</i> .....	22
Gambar 2.7 Penilaian kualitas <i>usability</i> .....	23
Gambar 3.1 Diagram alir metodologi .....	25

## **DAFTAR LAMPIRAN**

LAMPIRAN A WAWANCARA PERTAMA.....	31
LAMPIRAN B DOKUMEN PENDUKUNG HASIL WAWANCARA PERTAMA.....	33

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Pada era globalisasi ekonomi yang didukung pesatnya perkembangan teknologi menyebabkan persaingan usaha yang semakin ketat sehingga memaksa para wirausahawan menghasilkan produk yang sesuai dengan minat konsumen. Produk merupakan faktor penting bagi perusahaan dalam melakukan pemasarannya untuk bersaing dengan perusahaan lain baik yang sejenis maupun tidak sejenis, sehingga perusahaan berusaha melakukan proses produksi dengan efektif dan efisien untuk menghasilkan produk yang layak untuk bersaing. Proses produksi menurut Heizer dan Reider(2011) merupakan rangkaian kegiatan yang menghasilkan nilai dalam bentuk barang dan jasa dengan mengubah *input* menjadi *output*. Proses produksi sangat bergantung pada baik dan buruknya sistem produksi yang direncanakan.

Batik merupakan salah satu produk yang sangat potensial untuk dikembangkan pada era globalisasi ekonomi saat ini. Menurut Kemenperin (2019) nilai ekspor dari industri batik nasional pada semester I tahun 2019 mencapai USD17,99 juta, sementara untuk sepanjang tahun 2018 tembuhs hingga USD52,44 juta. Industri batik yang merupakan salah satu industri padat karya memiliki kelebihan yaitu menggunakan banyak tenaga manusia sehingga dapat membantu mengurangi jumlah pengangguran. Hal tersebut didukung oleh pernyataan Kemenperin (2019) bahwa industri batik yang didominasi oleh industri kecil dan menengah (IKM) tersebar di 101 sentra di Indonesia, dengan jumlah sebanyak 47.000 unit usaha dan telah menyerap tenaga kerja lebih dari 200.000 orang. Selain itu, di Jawa Timur industri batik merupakan industri yang dijadikan keunggulan dalam bersaing pada masing-masing daerah. Salah satu daerah yang banyak memiliki pengrajin batik yaitu Madura. Batik tulis Madura memiliki ciri khas tersendiri dari segi karakter motif yang kuat dan warnanya yang terkenal berani. Canteng Koneng merupakan salah satu industri batik Madura yang banyak diminati konsumen. Canteng Koneng memiliki banyak konsumen baik dari daerah sekitar maupun dari luar Kabupaten Sumenep, sehingga industri batik ini memiliki permintaan yang banyak dan proses produksi yang padat (LAMPIRAN A). Industri batik ini bertempat di Kabupaten Sumenep. Motif-motif batik yang diproduksi di Canteng Koneng kebanyakan berbeda dari motif-motif batik tulis Madura pada umumnya, namun tidak keluar dari ciri khasnya yaitu motif yang kuat dan warnanya yang berani.

Seperti yang dijelaskan sebelumnya bahwa industri batik merupakan industri padat karya yang artinya menggunakan banyak tenaga kerja manusia. Canteng Koneng memiliki beberapa tenaga kerja yang juga tersebar di beberapa daerah di Kabupaten Sumenep. Tenaga kerja yang dimiliki oleh industri batik tentu harus yang sudah terlatih. Namun pada industri batik Canteng Koneng tenaga kerja yang terlatih masih kurang sehingga tidak sebanding dengan padatnya proses produksi (LAMPIRAN A). Selain itu industri batik ini melakukan produksi dengan penjadwalan kerja yang kurang baik. Penjadwalan pasokan bahan baku juga belum terjadwal dengan baik. Beberapa hal tersebut berdampak manajemen produksi kurang tertata dengan baik sehingga berpengaruh terhadap kualitas batik dan produktifitas produksi batik.

Proses pemesanan batik Canteng Koneng masih dilakukan melalui telepon, Whatsapp, ataupun datang langsung ke industri batik tersebut. Menurut salah satu pegawai disana hal tersebut kurang efisien karena para konsumen seringkali meminta foto beberapa stok produk tersedia yang membuat pegawai tersebut harus mengirimkan tiap produk tersedia ke beberapa konsumen yang berbeda. Selain itu pemesanan produk masih dicatat melalui nota, sehingga industri batik tersebut belum memiliki pembukuan yang bisa mencatat data para pelanggan, data pengeluaran biaya, dan data produk terjual. Beberapa permasalahan tersebut juga terjadi karena belum adanya sistem yang terintegrasi untuk membantu mengelola proses bisnis tersebut sehingga pimpinan dan para penanggung jawab bisa memonitoring jalannya produksi dengan baik.

Pada penelitian sebelumnya yang memiliki permasalahan seperti di atas, untuk menyelesaikan permasalahan dalam manajemen proses produksi menerapkan metode PMS (*Production Monitoring System*) yang merupakan bagian dari MES (*Manufacturing Execution System*) yang meliputi pengolahan sumber daya, pengumpulan dan akuisisi data, pelacakan barang, analisis kinerja produksi, dan pemantauan mesin (Snaktin, et al., 2015). Beberapa penelitian yang juga terkait dengan penelitian ini menerapkan model *prototyping* sebagai metode pengembangan perangkat lunaknya. Seperti pada penelitian Aziz dan Pranolo (2015) menerapkan model *prototyping* karena model ini memungkinkan *developer* untuk lebih memahami kebutuhan *user* sebab setiap perubahan dilakukan atas hasil evaluasi dari *user*.

Pada penelitian ini, industri batik Canteng Koneng membutuhkan sebuah sistem monitoring produksi dan penjualan batik berbasis *website* yang bisa membantu pihak penjual dalam melakukan proses produksi agar termanajemen dengan baik dan juga membantu kedua belah pihak yaitu penjual dan pembeli



dalam melakukan transaksi penjualan dan pemesanan produk batik. Sistem ini dibangun dengan melakukan pendekatan *prototyping*, karena model tersebut dapat mengevaluasi sistem baik dari sisi pengembang, *client* hingga pengguna, sehingga nantinya sistem yang dibangun sesuai dengan yang diharapkan dan dibutuhkan oleh *client* dan pengguna. Selain itu juga agar memperkecil kegagalan pengembangan sistem. Bahasa yang digunakan dalam pembangun *website* tersebut yaitu PHP dengan menggunakan *framework* Codeigniter dan *database* MySQL yang diharapkan dapat menyajikan UI dan UX yang menarik dan memudahkan pengguna. Dengan adanya Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* ini diharapkan dapat menyelesaikan permasalahan-permasalahan di atas.

## 1.2 Rumusan Masalah

Berdasar pada latar belakang yang sudah dijelaskan, maka dapat dirumuskan permasalahan sebagai berikut.

1. Bagaimana hasil analisis kebutuhan dari pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website*?
2. Bagaimana hasil perancangan dari pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website*?
3. Bagaimana hasil implementasi dari pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* berdasarkan hasil perancangan yang dibuat?
4. Bagaimana hasil pengujian dari pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* setelah dilakukan proses pembangunan sistem?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Melakukan analisis kebutuhan fungsional dan non fungsional dalam pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website*.
2. Merancang Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* dengan berdasar pada hasil analisis kebutuhan.
3. Mengimplementasikan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* berdasarkan hasil dari perancangan.
4. Menghasilkan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* yang memenuhi semua kebutuhan *client*.

## 1.4 Manfaat Penelitian

Sistem monitoring produksi dan penjualan batik berbasis *website* diharapkan bisa memberikan manfaat:

1. Membantu dalam memonitoring proses produksi dan penjualan batik untuk mengetahui status proses produksi dan prospek penjualan.
2. Melakukan manajemen data produk, data bahan baku, dan data penjualan.
3. Membantu agar manajemen pada industri Batik Canteng Koneng menjadi lebih produktif dan efisien.

## 1.5 Batasan Masalah

Agar tujuan penelitian bisa tercapai sesuai target maka diperlukan batasan masalah sebagai berikut:

1. Sistem monitoring produksi dan penjualan berbasis *website* sehingga untuk mengoperasikannya membutuhkan koneksi internet.
2. Metode pengembangan perangkat lunak yang digunakan yaitu model *prototyping (requirement prototyping)*.
3. Implementasi sistem monitoring produksi dan penjualan meliputi pengolahan data sebagai berikut:
  - Pengolahan data produk.
  - Pengolahan data penjualan produk.
  - Pengolahan data pengeluaran dan pendapatan.
  - Pengolahan data bahan baku.

## 1.6 Sistematika Penelitian

Penelitian ini disusun secara sistematis dalam beberapa bagian bab. Urutan dari penelitian ini dimulai dari BAB I sampai BAB VII.

### BAB I Pendahuluan

Pada bagian ini membahas tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penelitian yang mendasari Pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website*.

### BAB II Landasan Kepustakaan

Pada bagian ini membahas tentang teori serta tinjauan yang digunakan dalam Pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website*. Landasan-landasan yang dipakai berasal dari beberapa penelitian sebelumnya yang berkaitan dengan penelitian ini, buku, jurnal, dan sumber referensi lain.

### BAB III Metodologi Penelitian

Pada bagian ini membahas tentang langkah dalam melakukan Pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website*. Langkah-langkah tersebut diantaranya studi literatur, analisis kebutuhan sistem, perancangan sistem, evaluasi oleh *client*, implementasi program, pengujian, dan penutup.

### BAB IV Analisis Kebutuhan

Pada bagian ini membahas tentang analisis kebutuhan yaitu spesifikasi kebutuhan beserta pemodelan kebutuhannya yang akan diterapkan dalam Pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website*.

### BAB V Perancangan

Pada bagian ini membahas tentang perancangan Pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* berdasarkan hasil dari analisis kebutuhan yang sudah dilakukan sebelumnya. Pada bagian ini akan terdapat hasil dari evaluasi *client* terhadap *prototype* sistem, sehingga apabila perancangan tidak sesuai dengan yang dibutuhkan *client* maka akan terdapat proses iterasi pada bagian ini.

### BAB VI Implementasi

Pada bagian ini membahas tentang implementasi Pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* berdasarkan perancangan yang sudah dibuat sebelumnya dan sudah disetujui oleh *client*.

### BAB VII Pengujian

Pada bagian ini membahas tentang pengujian dari Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* untuk memvalidasi dan memverifikasi sistem agar tidak terjadi kesalahan dan agar sesuai dengan analisis kebutuhan. Pada bagian ini akan terdapat hasil dari evaluasi *client* terhadap sistem yang sudah dibangun, sehingga apabila sistem tidak sesuai dengan rancangan yang sudah disetujui *client* maka akan terdapat proses iterasi pada bagian ini.

### BAB VIII Penutup

Pada bagian ini berisi kesimpulan dari Pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* dan saran untuk penelitian selanjutnya.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Pada Kajian Pustaka dijelaskan penelitian terdahulu yang dijadikan acuan dalam penelitian ini agar memperkaya teori-teori yang digunakan dan memperbiki kekurangan-kekurangan yang masih ada. Dengan mengacu pada penelitian terdahulu bukan berarti hal yang diteliti saat ini sama persis dengan penelitian terdahulu, namun pada penelitian ini hanya mengangkat beberapa penelitian yang berkaitan dengan penelitian yang akan dilakukan dan juga menambahkan beberapa fitur yang masih belum ada di penelitian sebelumnya. Tidak ditemukan penelitian yang berjudul sama dengan penelitian yang akan dilakukan. Pada Tabel 2.1 beberapa penelitian terdahulu dalam beberapa jurnal yang berkaitan dengan penelitian yang akan dilakukan.

**Tabel 2.1 Penelitian terdahulu**

Penulis	Judul Penelitian	Hasil Penelitian	Pembeda
Snatkin, et al. (2015)	Production Monitoring System Development and Modification	<ul style="list-style-type: none"><li>- Menerapkan metode PMS (<i>Production Monitoring System</i>) dalam memonitor proses produksi pada perusahaan produksi kayu massal.</li><li>- PMS merupakan bagian dari MES (<i>Manufacturing Execution System</i>) yang terdiri dari proses pengumpulan dan akuisisi data, manajemen pemeliharaan, status sumber daya, pelacakan produk, dan analisis kinerja produksi, namun</li></ul>	Setelah melakukan kajian terhadap penelitian sebelumnya, maka pada penelitian ini akan dibangun Sistem Monitoring Produksi dan Penjualan Batik Madura yang akan dibangun dengan bahasa pemrograman PHP dengan <i>framework</i> Codeigniter dan <i>database</i> MySQL. Paradigma pemrograman yang diterapkan yaitu <i>object oriented</i> .

**Tabel 2.1 Penelitian terdahulu (lanjutan)**

Penulis	Judul Penelitian	Hasil Penelitian	Pembeda
		<p>pada PMS juga terdapat fungsi untuk pemantauan mesin.</p> <ul style="list-style-type: none"> <li>- Pengembangan sistem menggunakan teknologi AJAX (Asynchronous Javascript dan XML), bahasa pemrograman PHP, database MySQL, phpMyAdmin, dan server FileZilla, Apache.</li> <li>- Jika PMS dirancang dan diimplementasikan pada perusahaan berproduksi maka akan membantu dalam mendapatkan manfaat dari manufaktur. Selain itu sistem pemantauan yang diusulkan pada penelitian ini didasarkan pada <i>software</i> dan <i>hardware open-source</i> yang membuatnya lebih terjangkau.</li> </ul>	<p>Metode pengembangan perangkat lunak yang diterapkan pada penelitian ini yaitu model <i>requirement prototyping</i>.</p> <p>Pada penelitian sebelumnya, dalam melakukan monitoring terhadap proses produksi menggunakan metode PMS (<i>Production Monitoring System</i>) yang salah satu fungsinya yaitu untuk memantau mesin, dan pada penelitian ini mirip dengan metode tersebut namun pada penelitian ini tidak diperlukan fungsi pemantauan mesin.</p>
Aziz, Pranolo, dan	Developing Sales Information System Application using	<ul style="list-style-type: none"> <li>- Menerapkan model evolutionary prototyping.</li> </ul>	

**Tabel 2.1 Penelitian terdahulu (lanjutan)**

Penulis	Judul Penelitian	Hasil Penelitian	Pembeda
Agustin (2015)	Prototyping Model	<ul style="list-style-type: none"> <li>- Menghasilkan sistem informasi untuk laporan pembelian, penjualan, laporan manajemen, informasi inventaris.</li> <li>- Menggunakan paradigma pemrograman terstruktur, sehingga perancangan sistem digambarkan dalam bentuk DFD (Data Flow Diagram).</li> <li>- Dalam mengevaluasi sistem menggunakan metode kuisisioner, dan diperoleh hasil evaluasi dari sistem yang dibangun, antara lain: 1) 88% sangat setuju dan 11% setuju bahwa aplikasi dapat meningkatkan efektivitas dan efisien organisasi / perusahaan; 2) 33% sangat setuju, 62% setuju, dan 5% tidak setuju bahwa aplikasi dapat memenuhi kebutuhan organisasi / perusahaan.</li> </ul>	
Anthony, Tanaamah, dan	Analisis dan Perancangan Sistem Informasi	<ul style="list-style-type: none"> <li>- Menerapkan model evolutionary prototyping.</li> </ul>	

**Tabel 2.1 Penelitian terdahulu (lanjutan)**

Penulis	Judul Penelitian	Hasil Penelitian	Pembeda
Wijaya (2017)	Penjualan Berdasarkan Stok Gudang berbasis Client Server (Studi Kasus Toko Grosir “Restu Anda”)	<ul style="list-style-type: none"><li>- Menggunakan paradigma pemrograman terstruktur, sehingga perancangan sistem digambarkan dalam bentuk DFD (Data Flow Diagram).</li><li>- Mengelola data member, data pemasok, dan data penjualan barang yang dilakukan oleh pemilik toko.</li><li>- Pengujian menggunakan blackbox saja.</li></ul>	

## 2.2 Rekayasa Perangkat Lunak

Perangkat lunak (Software) merupakan program komputer beserta dokumentasi yang terkait. Rekayasa perangkat lunak merupakan prosedur dan serangkaian metode untuk membangun perangkat lunak yang berkualitas (Pressman, 2010). Rekayasa perangkat lunak membantu para *developer* untuk membangun sistem yang kompleks dengan kualitas yang baik secara tepat waktu. Menurut Sommerville (2011) rekayasa perangkat lunak merupakan disiplin ilmu yang berkaitan dengan keseluruhan aspek dalam membangun perangkat lunak yang mencakup beberapa teknik diantaranya spesifikasi, desain, dan evolusi program.

## 2.3 Pengembangan Perangkat Lunak

Pengembangan perangkat lunak merupakan serangkaian tahap rekayasa perangkat lunak. Tahap –tahap tersebut mengidentifikasi tugas-tugas yang harus diselesaikan, produk yang akan dikembangkan, jaminan kualitas produk, dan tujuan dari pengembangan produk tersebut. Beberapa model tahap rekayasa perangkat lunak diantaranya *linear process flow*, *iterative process flow*,

*evolutionary process flow, parallel process flow* (Pressman, 2010). Sommerville (2011) juga menyebutkan beberapa model pengembangan perangkat lunak yaitu *waterfall model, incremental development, reuse-oriented software engineering*. Model-model tersebut tidak sering digunakan bersama, terutama dalam pengembangan sistem yang besar.

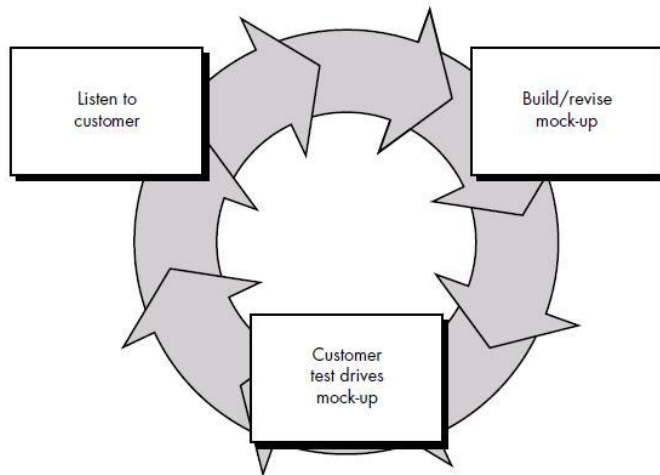
Tahap-tahap dalam proses pengembangan perangkat lunak, diantaranya:

1. Inisiasi dan perencanaan, pada tahap ini menyusun kerangka kerja untuk menjalankan tahapan-tahapan lainnya.
2. Analisis Kebutuhan, melakukan analisis kebutuhan baik fungsional maupun nonfungsional terhadap *user* dari sistem perangkat lunak yang akan dikembangkan.
3. Perancangan Sistem, menciptakan solusi dari permasalahan untuk memenuhi kebutuhan-kebutuhan perangkat lunak.
4. Pengembangan, pada tahap ini perancangan/desain yang sudah dibuat pada tahap sebelumnya dibangun menjadi sistem informasi yang lengkap.
5. Integrasi dan Pengujian Sistem, yaitu sistem perangkat lunak dijalankan untuk diuji apakah telah memenuhi kebutuhan yang di spesifikasikan pada dokumen kebutuhan fungsional dan nonfungsional.
6. Pemeliharaan Sistem, yaitu memelihara dan melakukan peninjauan terhadap sistem informasi pada lingkungan produksi.

### **2.3.1 Prototyping**

Menurut Sommerville (2011), *prototyping* merupakan rancangan awal dari konsep sistem perangkat lunak yang digunakan untuk memberikan opsi desain, dan mengetahui lebih lanjut tentang masalah yang mungkin ada. Model *prototyping* dapat membantu pengembang untuk menyambungkan ketidakpahaman *client* terhadap hal teknis dan memperjelas spesifikasi kebutuhan sesuai keinginan *client*. *Prototyping* berfungsi sebagai mekanisme untuk mengidentifikasi persyaratan perangkat lunak (Pressman, 2010). Penggambaran model *prototyping* dapat dilihat pada Gambar 2.1.



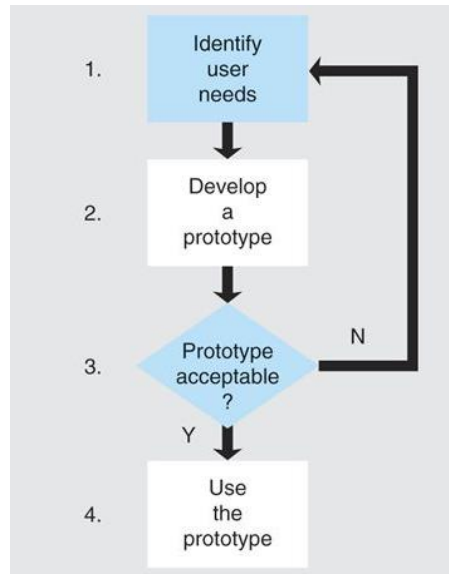


**Gambar 2.1 Model *prototyping***

Sumber : Pressman (2001)

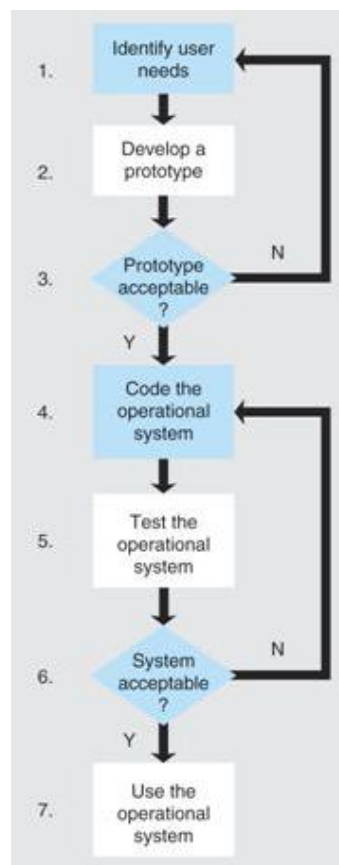
Pada Gambar 2.1 terdapat tiga tahap pada model *prototyping* yaitu *listen to customer* (mendengarkan pelanggan), *build/revise mockup* (membangun/merevisi *mockup*), dan *customer test drives mockup* (test *mockup* oleh pelanggan). Pada tahap pertama dimulai dengan mengumpulkan kebutuhan-kebutuhan *customer* dengan melakukan pertemuan antara pengembang dan *customer* untuk menetapkan tujuan dibangunnya sistem perangkat lunak, mengidentifikasi kebutuhan perangkat lunak, dan membicarakan bidang yang terkait. Selanjutnya pada tahap kedua dilakukan “*quick design*” untuk membangun sebuah *prototype* yang merepresentasikan aspek-aspek sistem perangkat lunak yang nantinya akan ditunjukkan kepada *customer*. Setelah *prototype* selesai dibuat, lalu dilakukan diberikan kepada *customer* untuk dievaluasi. Proses iterasi terjadi apabila *prototype* belum memenuhi kebutuhan *customer* dan pengembang juga ingin lebih memahami apa yang perlu diperbaiki pada *prototype*.

McLeod dan Schell (2007) menjelaskan bahwa terdapat 2 tipe model *prototype* yaitu *evolutionary prototype* dan *requirements prototype*. *Evolutionary prototype* yang digambarkan pada Gambar 2.2 merupakan *prototype* yang dikembangkan secara terus menerus hingga memenuhi fungsionalitas yang dibutuhkan oleh sistem. Sedangkan *requirement prototype* yang digambarkan pada Gambar 2.3 merupakan *prototype* yang digunakan ketika dalam menentukan kebutuhan *user* tidak dapat mengartikulasikan apa yang mereka inginkan. Pada penelitian ini menggunakan model *requirement prototype* karena *customer* belum bisa dengan pasti menggambarkan sistem yang akan dibangun, selain itu model *prototype* ini juga bisa membantu pengembang membangun sistem karena selalu melibatkan *customer* baik dalam membangun desain *prototype* hingga membangun desain tersebut menjadi sistem perangkat lunak.



**Gambar 2.2 Model *evolutionary prototyping***

Sumber : McLeod dan Schell (2007)



**Gambar 2.3 Model *requirement prototyping***

Sumber : McLeod dan Schell (2007)

## **2.4 Object Oriented**

*Object Oriented* merupakan paradigma pemrograman yang berorientasi objek. Data, fungsi, dan metode dibungkus ke dalam beberapa *class* atau objek-objek. *Object oriented* menggunakan objek sebagai domain masalah untuk mendorong dalam pembuatan struktur data dan metode yang akan diterapkan dalam pembangunan perangkat lunak (Pressman, 2010). UML adalah bentuk nyata dari standar untuk pemodelan *object oriented*, *use case* dan *use case* yang berbasis elisitasi banyak untuk elisitasi kebutuhan di era sekarang (Sommerville, 2011).

## **2.5 UML (Unified Modelling Language)**

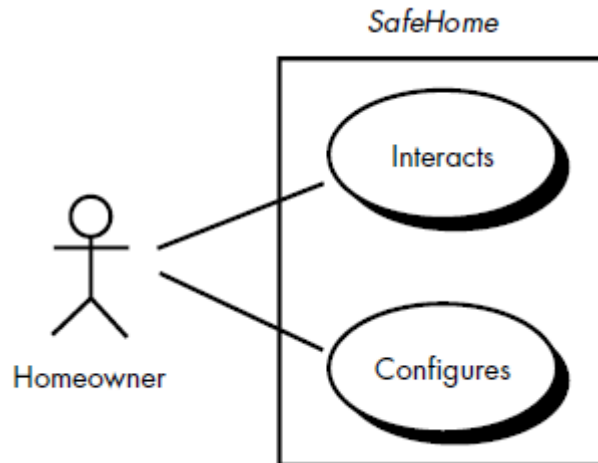
UML (*Unified Modelling Language*) adalah bahasa pemodelan yang berisi notasi-notasi untuk pemodelan dan pengembangan perangkat lunak dengan pendekatan *object-oriented analysis* (OOA). Karena menggunakan pendekatan OOA maka UML membuat analisis model perangkat lunak berdasarkan karakteristik representasi kelas dan hirarki kelas, model hubungan antar objek, dan model perilaku objek (Pressman, 2010). UML berfungsi menspesifikasikan, menggambarkan, membangun, dan mendokumentasi sistem perangkat lunak ke dalam pemodelan visual.

### **2.5.1 Use Case Diagram**

*Use case diagram* merupakan diagram yang digunakan untuk menjelaskan sebuah sistem tampak dari luar. *Use case diagram* menghubungkan antara *actor* sebagai orang yang berinteraksi dengan sistem dan *use case* yang merupakan proses yang dapat dilakukan oleh *actor* pada sebuah sistem. Menurut Pressman (2001) *use case* harus mencapai tujuan berikut:

1. Menentukan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario *user* yang disepakati oleh pengembang dan *user*.
2. Memberikan deskripsi interaksi antara *user* dan sistem dengan jelas dan tidak ambigu.
3. Memberikan dasar *validation testing*.

Penggambaran *use case diagram* dapat dilihat pada Gambar 2.4.



**Gambar 2.4 Use case diagram**


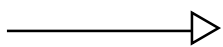
Sumber : Pressman (2001)

Simbol-simbol beserta deskripsinya yang biasa digunakan dalam *use case diagram* dapat dilihat pada Tabel 2.2.

**Tabel 2.2 Simbol dan deskripsi use case diagram**

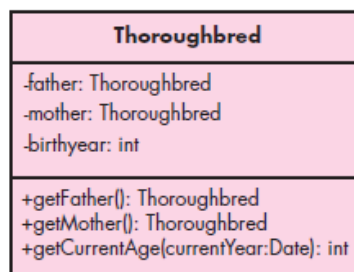
Notasi	Nama	Deskripsi
	<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal fase nama <i>use case</i> (Shalahuddin dan Rosa, 2015)
	Aktor	Aktor adalah yang berinteraksi dengan <i>use case</i> merepresentasikan sekumplan peranan yang yang berkorelasi di dalam sistem (Pressman, 2010)
<<include>> 	Menggunakan/ <i>include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> tanpa membutuhkan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini (Shalahuddin dan Rosa, 2015).
<<extend>> 	Ekstensi/ <i>Extended</i>	Relasi <i>use case</i> tambahkan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan (Shalahuddin dan Rosa, 2015)

**Tabel 2.2 Simbol dan deskripsi *use case diagram* (lanjutan)**

Notasi	Nama	Deskripsi
	Asosiasi/ <i>Association</i>	Komunikasi antar actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor (Shalahuddin dan Rosa, 2015)
	Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. (Shalahuddin dan Rosa, 2015)

### 2.5.2 Class Diagram

*Class diagram* menggambarkan tampilan statis atau struktural dari suatu sistem yang digunakan untuk memodelkan kelas, termasuk atribut, operasi dan hubungan antar kelas (Pressman, 2010). Pada Gambar 2.5 merupakan penggambaran *class diagram*.



**Gambar 2.5 Class diagram**

Sumber : Pressman (2010)

Menurut Shalahuddin dan Rosa (2015), kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel-variabel yang dimiliki kelas.
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Simbol-simbol beserta deskripsinya yang biasa digunakan dalam *class diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2.3 Simbol dan deskripsi *class diagram***

Notasi	Nama	Deskripsi
<div>nama_kelas</div> <div>+atribut</div> <div>+operasi()</div>	Kelas	Kelas pada struktur system
	Asosiasi/association	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
	Agregasi/aggregation	Relasi antarkelas dengan makna semua bagian

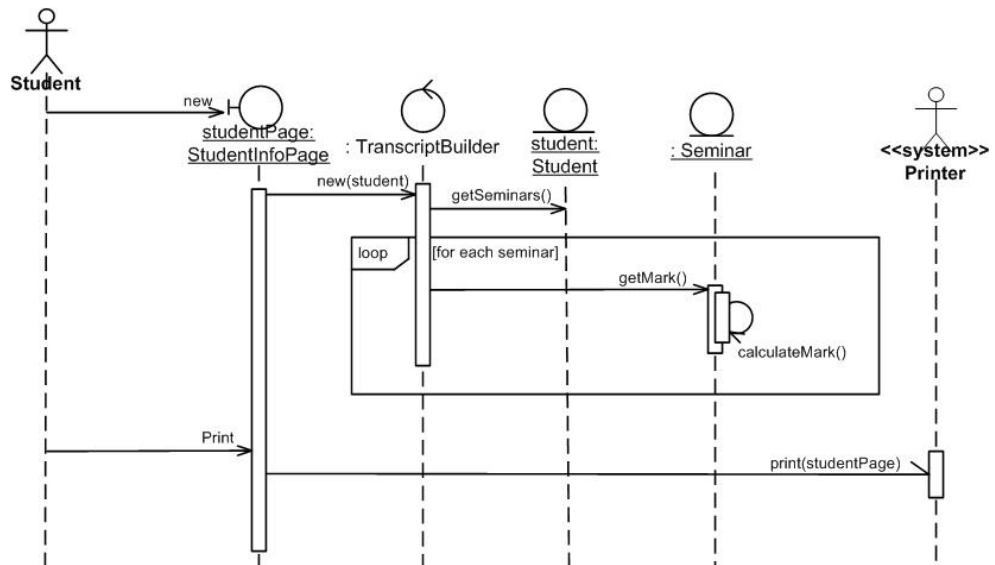
Sumber: Diadaptasi dari Shalahuddin dan Rosa (2015)

### 2.5.3 Use Case Scenario

Setiap *use case diagram* dilengkapi dengan *use case scenario*. *Use case scenario* digunakan untuk melatih *user interface design* (Pressman, 2001). Alur dari *use case* akan digambarkan pada *use case scenario* dari sisi aktor dan sistem. *Use case scenario* biasanya digambarkan dalam bentuk tabel yang di dalamnya dijelaskan kode dari kebutuhan, *objective* (tujuan), aktor, *pre-condition*, *main flow*, *alternative flow*, dan *post-condition*. *Pre-condition* dan *post-condition* menjelaskan kondisi sebelum dan sesudah kebutuhan tersebut dijalankan. *Main flow* menjelaskan alur dari jalannya sistem pada kebutuhan tersebut. *Alternative flow* menjelaskan skenario jika sistem tidak berjalan dengan normal. *Use case scenario* akan menjadi landasan pembuatan *sequence diagram*.

### 2.5.4 Sequence Diagram

*Sequence diagram* adalah diagram UML yang digunakan untuk memodelkan interaksi antara aktor dengan objek dalam suatu sistem dan interaksi antara objek itu sendiri (Sommerville, 2011). *Sequence diagram* merupakan versi singkat dari *use case* yang mewakili perilaku dengan menggambarkan bagaimana suatu kelas berpindah dari satu *state* ke *state* yang lain (Pressman, 2010). Pada Gambar 2.6 merupakan penggambaran *sequence diagram*.



**Gambar 2.6 Sequence diagram**


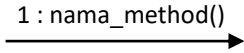
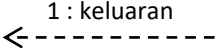



Sumber: Agile Modeling (2018)

Simbol-simbol beserta deskripsinya yang biasa digunakan dalam *sequence diagram* dapat dilihat pada Tabel 2.4.

**Tabel 2.4 Simbol dan deskripsi *sequence diagram***


Notasi	Nama	Deskripsi
	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama ackor (Shalahuddin dan Rosa, 2015).
	Garis hidup/ <i>Lifeline</i>	Menyatakan kehidupan suatu objek (Shalahuddin dan Rosa, 2015).

**Tabel 2.4 Simbol dan deskripsi *sequence diagram*(lanjutan)**

	Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya (Shalahuddin dan Rosa, 2015).
	Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi yang ada pada objek lain (Shalahuddin dan Rosa, 2015).
	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian (Shalahuddin dan Rosa, 2015).
	<i>Boundary</i>	<i>Boundary class</i> merupakan objek <i>stereotype</i> yang memodelkan beberapa batasan sistem (Sparxsystems, 2018).
	<i>Control</i>	Merupakan objek <i>stereotype</i> yang memodelkan entitas pengendali atau manajer. <i>Control</i> mengatur dan menjadwalkan aktivitas dan elemen lain (Sparxsystems, 2018).
	<i>Entity</i>	<i>Entity</i> adalah objek <i>stereotype</i> yang memodelkan mekanisme yang menangkap informasi dalam suatu sistem (Sparxsystems, 2018).



**Tabel 2.4 Simbol dan deskripsi *sequence diagram*(lanjutan)**

	<p><i>Combined Fragment</i></p>	<p>Fragmen gabungan mendefinisikan ekspresi fragmen interaksi. Fragmen gabungan didefinisikan oleh operator interaksi dan operan interaksi yang sesuai (Sparxsystems, 2018).</p>
---	---------------------------------	--

## 2.6 Bahasa Pemrograman

### 2.6.1 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman web sisi server (*server-side*) yang bersifat *open source*. PHP dikatakan *server side* karena pemrosesannya dilakukan di lingkungan *web server*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser* (Raharjo, Heryanto dan Enjang, 2012).

### 2.6.2 Javascript

Sebelumnya dijelaskan bahwa PHP merupakan bahasa pemrograman *server side*, untuk *client side* bahasa pemrograman yang menjadi pilihan yaitu Javascript, dimana pemrosesan dilakukan di komputer penggunanya sendiri. Javascript adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti Internet Explorer(IE), Mozilla Firefox, Netscape, dan Opera (Sunyoto, 2007). Javascript dapat memberikan fitur animasi, game, dan fitur interaktif lain pada dokumen HTML. Javascript juga mendukung penggunaan metode berorientasi objek. Untuk menyisipkan kode Javascript pada halaman web yaitu dengan menggunakan tag SCRIPT.

## 2.7 MySQL

MySQL merupakan salah satu *database server* yang didukung oleh bahasa pemrograman PHP dan banyak digunakan untuk pengembangan aplikasi web. Menurut Raharjo, Heryanto dan Enjang (2012) alasan MySQL banyak digunakan karena gratis, pengelolaan datanya sederhana, memiliki tingkat keamanan yang bagus, mudah diperoleh, dan lain-lain. Hidayatullah dan Kawistara (2017) juga

menjelaskan kelebihan dari MySQL adalah gratis, handal, selalu di-*update*, dan proses instalasinya mudah karena MySQL menjadi DBMS yang sering di *bundling* dengan *web server*.

## 2.8 **Framework**

*Framework* berisi berbagai fungsi dan konsep untuk membantu *programmer* dalam membuat aplikasi web agar web tersebut tersusun dan terstruktur dengan rapih. *Framework* merupakan kumpulan dari fungsi atau prosedur dan kelas untuk tujuan tertentu yang dapat digunakan sehingga bisa lebih mempermudah dan mempercepat dalam mengembangkan perangkat lunak (Basuki, 2014). Jenis *framework* yang populer diantaranya Codeigniter, Laravel, CakePHP, dan Symfony.

### 2.8.1 Codeigniter

Codeigniter merupakan salah satu *framework* PHP yang paling populer dalam pengembangan sebuah aplikasi web. Codeigniter adalah *framework* PHP yang paling powerful saat ini karena di dalamnya terdapat fitur lengkap aplikasi web dimana fitur-fitur tersebut sudah dikemas menjadi satu (Hidayatullah dan Kawistara, 2017). Codeigniter dibangun dengan konsep MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP yang dapat mempercepat pengembangan dalam membuat perangkat lunak berbasis web (Daqiqil, 2011).

### 2.8.2 Bootstrap

Bootstrap adalah sebuah pustaka *open source* yang merupakan *framework* CSS dan Javascript untuk membuat *website* yang responsif (Sulistiono, 2018). Paket Bootstrap berisi sekumpulan file CSS, font, dan Javascript yang siap diintegrasikan ke sebuah dokumen HTML menggunakan kaidah-kaidah tertentu (Rozi, et al., 2015). Bootstrap membantu mempermudah proses desain web untuk berbagai tingkat pengguna, baik dari tingkat pemula hingga yang berpengalaman karena *framework* ini mmbrikan kemudahan dan kepraktisan dalam penggunaannya.

## 2.9 MVC (*Model View Controler*)

MVC (*Model View Controler*) merupakan pola desain perangkat lunak yang dibangun dari tiga jenis komponen utama yang terdiri dari *model*, *view*, dan *controller* dalam bahasa pemrograman seperti PHP dengan fokus pada paradigma *object oriented programming* (OOP) (Pitt, 2012). *Model* mengelola basis data seperti MySQL ataupun Oracle RDMS, *view* merupakan bagian *user interface* atau

bagian yang nantinya merupakan tampilan untuk *end-user*, dan *controller* adalah penghubung antara *view* dan *model* karena *model* tidak dapat berhubungan langsung dengan *view* ataupun sebaliknya (Hidayatullah dan Kawistara, 2017).

## 2.10 Google Analytics

Google Analytics merupakan layanan yang disediakan oleh Google secara gratis untuk menampilkan statistik pengunjung pada situs web tertentu. Pemilik situs dapat membuat laporan tentang trafik dari web dalam jangka waktu baik harian sampai bulanan melalui Google Analytics. Dengan menggunakan Google Analytics pemilik situs dapat membuat keputusan yang lebih cerdas untuk meningkatkan kualitas bisnisnya.

Google Analytics memudahkan dalam memahami bagaimana pengunjung *website* terlibat dengan konten yang disediakan, sehingga pemilik situs tahu apakah konten tersebut berguna atau tidak. Selain itu *benefit* dari penggunaan Google Analytics yaitu dapat memahami pengunjung *website* untuk memeriksa kinerja pemasaran, konten, produk lebih baik; dapat memaksimalkan data; dan pemilik situs dapat menggunakan wawasannya untuk membantu menjangkau pelanggan yang tepat karena Google Analytics bekerja dengan produk periklanan dan penerbit Google (Google, 2020).

## 2.11 Pengujian Perangkat Lunak

Seringkali perangkat lunak mengalami *error* pada saat perangkat lunak tersebut sudah ada di tangan *user*. Untuk menghindari *error* tersebut diperlukan adanya tahap pengujian sebelum perangkat lunak sampai ke tangan *user*. Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan (Shalahuddin dan Rosa, 2015).

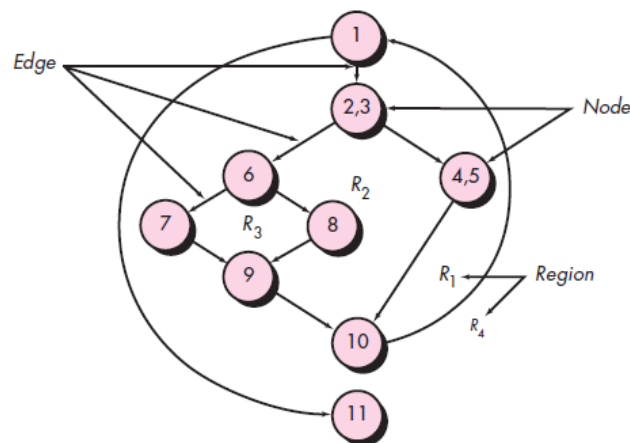
### 2.11.1 White Box Testing

*White box testing* yaitu pengujian yang dilakukan pada kode program. Serangkaian pengujian harus memastikan tiap *logical path* pada program dijalankan, dan setiap *statement* pada program dieksekusi setidaknya sekali (Sommerville, 2011). *White box testing* menguji apakah perangkat lunak mampu menghasilkan fungsi, *input*, dan *output* sesuai dengan spesifikasi kebutuhan.

Pengujian *whitebox* memiliki beberapa teknik salah satunya adalah *basis path testing*. *Basis path testing* merupakan salah satu metode dari *white box testing*. *Basis path testing* bertujuan untuk mengetahui kompleksitas logika (*cyclomatic complexity*) dari sebuah program (Pressman, 2010). Pada *basis path testing* terdapat empat elemen elemen pengujian yaitu (Pressman, 2010):

1. *Flow graph*

*Flow graph* menggambarkan struktur logika yang ada pada program. Pada Gambar 2.7, bentuk lingkaran yang terdapat pada gambar merupakan penggambaran sebuah *node*. *Node* merupakan perwakilan dari satu atau lebih *procedural statements*. Tanda panah yang menghubungkan antar *node* disebut *edge* yang menggambarkan arah dari algoritme program pada *flow chart*. Wilayah yang dibatasi oleh *edge* dan *node* disebut *region*. Ketika menghitung *region*, area yang berada diluar *flow graph* dihitung sebagai satu *region*. Penggambaran *flow graph* dapat dilihat pada Gambar 2.7.



**Gambar 2.7 Flow graph**

Sumber: Pressman (2010)

## 2. Cyclomatic Complexity

*Cyclomatic complexity* merupakan pengukuran *software* yang digunakan untuk mengukur kompleksitas logika dari sebuah program. *Cyclomatic complexity*  $V(G)$  dapat dihitung dengan 3 cara:

1. Jumlah *region* dalam *flow graph*
2.  $V(G) = E \text{ (Edge)} - N \text{ (Node)} + 2$
3.  $V(G) = P \text{ (Predicate Node)} + 1$

## 3. Independent Path

*Independent path* merupakan semua jalur yang dilalui program yang menghasilkan setidaknya sebuah *statement* atau kondisi baru. *Independent path* harus melewati setidaknya satu *edge* yang belum dilewati oleh *independent path* sebelumnya.

## 4. Test Case

Untuk melakukan *test case* data harus dipilih sehingga kondisi pada node predikat ditetapkan dengan tepat saat setiap jalur diuji. Setiap *test case* dieksekusi dan dibandingkan dengan hasil yang diharapkan.

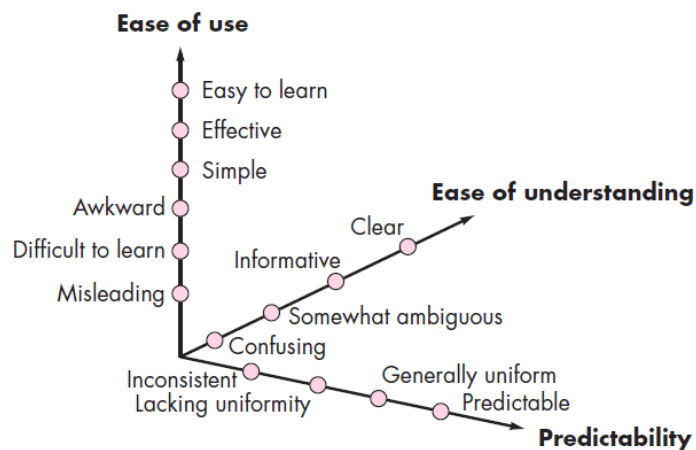
### 2.11.2 Black Box Testing

*Black box testing* digunakan untuk menguji perangkat lunak dari segi spesifikasi fungsionalnya tanpa menguji kode program. Pada pengujian ini tidak perlu pengetahuan cara kerja sistem, karena yang perlu dilakukan yaitu melihat kode program untuk menemukan test lain yang mungkin, misalnya menyertakan pengecualian untuk menangani input yang salah (Sommerville, 2011).

*Black box testing* mencoba untuk menemukan kesalahan pada suatu fungsi, kesalahan antarmuka, kesalahan dalam struktur data atau akses *database* eksternal, kesalahan kinerja, dan kesalahan inisialisasi dan terminasi (Pressman, 2010). Kasus uji pada *black box testing* harus dibuat dengan 2 kondisi, yaitu kondisi dengan kasus benar dan kasus salah.

### 2.11.3 Usability Testing

*Usability testing* merupakan metode pengujian yang mencerminkan kemudahan sistem. Menurut Pressman (2010) *usability testing* mengevaluasi tingkat efektifitas pengguna untuk berinteraksi dengan web dan sejauh mana web dapat memandu pengguna dalam menggunakannya, memberikan umpan balik, dan konsistensi. Pada Gambar 2.8 menggambarkan beberapa metode penilaian yang dapat dilakukan oleh pengguna.



Gambar 2.8 Penilaian kualitas *usability*

Sumber: Pressman (2010)

#### 2.11.3.1 PUPEU (*Perceived Usefulness and Perceived Ease of Use*)

*PUPEU* (*Perceived Usefulness and Perceived Ease of Use*) merupakan salah satu metode kuisioner yang digunakan untuk mengevaluasi aspek *usability* suatu *website*. Menurut Davis (1989) suatu sistem dengan persepsi *usefulness* adalah jika pengguna merasakan adanya relasi yang positif antara pengguna dan performa dari sistem. Persepsi *ease of use* yaitu meminimalkan effort yang

dibutuhkan pengguna dalam mengoperasikan sistem. Sebuah aplikasi akan lebih diterima oleh pengguna apabila aplikasi tersebut lebih mudah digunakan.

Skala pengukuran PUPEU, diantaranya:

*Perceived usefulness:*

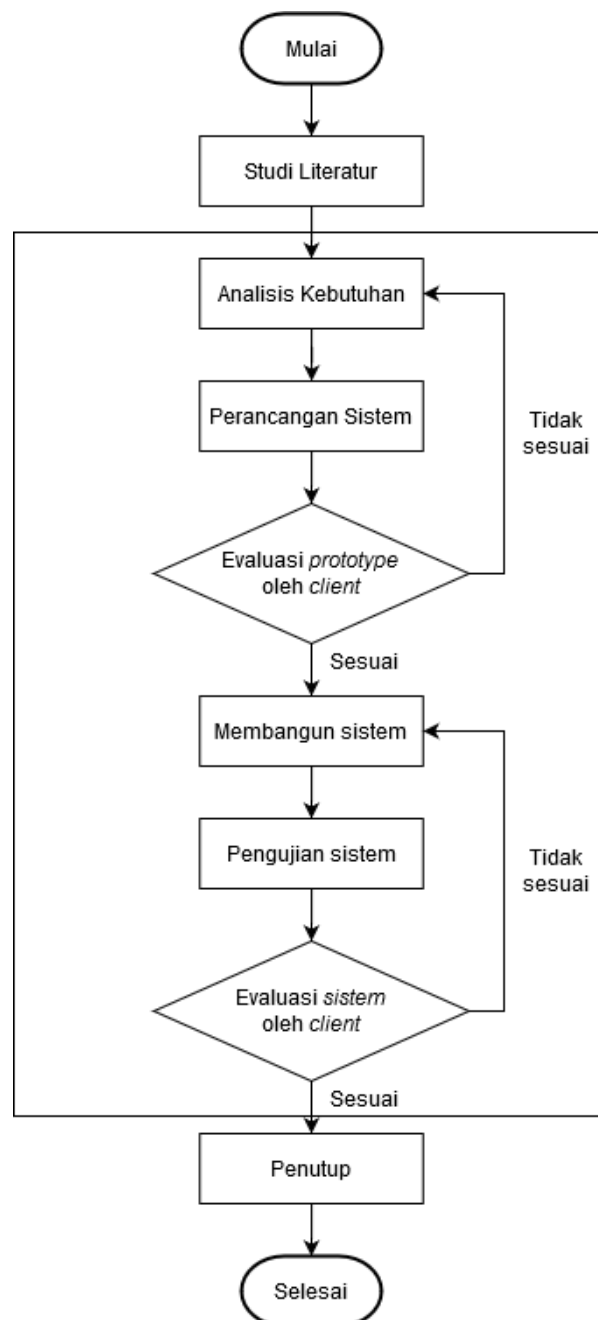
1. Menggunakan sistem dalam pekerjaan saya memungkinkan untuk saya menyelesaikan tugas lebih cepat.
2. Menggunakan sistem akan meningkatkan kinerja saya.
3. Menggunakan sistem akan meningkatkan produktivitas saya.
4. Menggunakan sistem akan meningkatkan efektivitas saya dalam bekerja.
5. Menggunakan sistem akan mempermudah saya dalam melakukan pekerjaan
6. Saya akan menemukan bahwa sistem ini berguna dalam pekerjaan saya.

*Perceived ease of use:*

7. Mudah bagi saya untuk belajar mengoperasikan sistem.
8. Mudah bagi sistem untuk melakukan apa yang ingin saya lakukan.
9. Interaksi saya dengan sistem jelas dan dapat dimengerti.
10. Sistem fleksibel dalam berinteraksi.
11. Saya menggunakan sistem dengan terampil
12. Sistem mudah untuk digunakan.

### BAB 3 METODOLOGI PENELITIAN

Pada bab metodologi penelitian akan dijelaskan metodologi yang akan dilakukan dalam pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website*. Penjelasan mengenai tahap-tahap dalam pengembangan Sistem Monitoring Produksi dan Penjualan Batik Madura berbasis *Website* akan diilustrasikan ke dalam diagram alir dalam Gambar 3.1:



Gambar 3.1 Diagram alir metodologi

### 3.1 Studi Literatur

Studi literatur menjelaskan sumber dari teori-teori yang relevan dengan penelitian ini, dan juga digunakan sebagai acuan yang akan digunakan untuk memperkaya wawasan penelitian, memperbaiki kekurangan-kekurangan yang masih ada, dan membantu penelitian ini dapat berjalan dengan baik dari tahap awal hingga akhir. Studi literatur bisa didapatkan dari buku, paper, dan dokumentasi dari internet. Studi literatur atau landasan teori yang relevan dengan penelitian ini sebagai berikut:

1. Kajian pustaka
2. Rekayasa perangkat lunak
3. Pengembangan perangkat lunak
  - a. *Prototyping*
4. *Object Oriented*
5. UML (*Unified Modelling Language*)
  - a. *Use case diagram*
  - b. *Class diagram*
  - c. *Use case scenario*
  - d. *Sequence diagram*
6. Bahasa Pemrograman
  - a. PHP
  - b. Javascript
7. MySQL
8. Framework
  - a. Codeigniter
  - b. Bootstrap
9. MVC
10. Google Analytics
11. Pengujian Perangkat Lunak
  - a. *White box testing*
  - b. *Black box testing*
  - c. *Usability testing* (Menggunakan *PUPEU* (Perceived Usefulness and Perceived Ease of Use))

### 3.2 Analisis Kebutuhan

Analisis kebutuhan sistem digunakan untuk menemukan, menganalisis mendefinisikan, dan mendokumentasikan kebutuhan sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan wawancara kepada pemangku kepentingan



yaitu salah satu pegawai yang merupakan bagian produksi di Batik Canteng Koneng dengan menggunakan pendekatan berorientasi objek. Pendekatan ini mengidentifikasi kebutuhan fungsional, kebutuhan non fungsional, dan aktor-aktor yang akan terlibat pada pengembangan sistem ini.

Setelah dilakukan analisis kebutuhan, maka akan dihasilkan deskripsi dari sistem, definisi kebutuhan sistem (fungsional dan non fungsional), dan spesifikasi kebutuhan sistem. Lalu akan dilakukan pemodelan kebutuhan yang terdiri dari *use case diagram* dan *use case scenario*. Pemodelan kebutuhan dilakukan untuk mengilustrasikan cara aktor berinteraksi dengan sistem.

### **3.3 Perancangan Sistem**

Perancangan sistem dilakukan berdasarkan hasil yang diperoleh dari analisis kebutuhan di tahap sebelumnya. Perancangan sistem digunakan sebagai acuan dalam tahap implementasi. Tahap perancangan sistem diantaranya yaitu *sequence diagram*, *class diagram*, dan perancangan antarmuka. *Sequence diagram* menggambarkan interaksi antar objek. *Class diagram* menggambarkan struktur kelas yang akan dibuat untuk membangun sistem. Perancangan antarmuka merupakan gambaran sistem (*prototype*) yang diimplementasikan dalam bentuk *mock-up*. Fungsi-fungsi yang ditampilkan pada *mock-up* masih terbatas dan akan menjadi kompleks seiring berjalannya proses pengembangan setelah dilakukan evaluasi oleh *client*.

### **3.4 Evaluasi *Prototype* oleh *Client***

Pada tahap ini, *prototype* yang sudah dibuat di tahap sebelumnya dievaluasi oleh para pemangku kepentingan sampai ditemukan spesifikasi yang sesuai dengan keinginan pemangku kepentingan. Jika hasil perancangan belum sesuai dengan permintaan pemangku kepentingan maka akan dilakukan kembali proses analisis kebutuhan dan perancangan sampai hasil perancangan sesuai dan disetujui oleh para pemangku kepentingan. Setelah disetujui, selanjutnya *prototype* akan dikembangkan ke dalam bentuk sistem yang sebenarnya.

### **3.5 Membangun Sistem**

Berdasarkan *prototype* yang sudah di evaluasi dan disetujui oleh para pemangku kepentingan, dibuatlah dibangunlah sistem yang mengimplementasikan komponen-komponen yang sebenarnya. Tahap ini akan menghasilkan sebuah web sistem monitoring yang dibangun menggunakan bahasa dan teknologi sebagai berikut:

1. Bahasa pemrograman PHP

2. *Framework Codeigniter* yang menerapkan metode MVC (*Model, View, Controller*)
3. *Database MySQL*

### 3.6 Pengujian

Setelah sistem dibangun, sebelum diberikan dan dievaluasi kembali oleh *client* dilakukan pengujian untuk memvalidasi dan memverifikasi agar tidak terjadi bug dan sistem sesuai dengan kebutuhan yang telah didefinisikan sebelumnya. Pada tahap pengujian akan menghasilkan hasil uji dari pengujian *white box*, pengujian *black box*, dan pengujian *usability*. Pada pengujian *white box* akan menerapkan metode *basis path testing*. Pengujian *white box* dilakukan untuk memeriksa logika pada kode program. Dari pengujian ini akan diketahui hasil kompleksitasnya. Pengujian *black box* dilakukan untuk mengetahui fungsi-fungsi, masukan, dan keluaran dari sistem sudah berjalan dengan baik atau belum. Pengujian *usability* diuji menggunakan metode kuisioner *PUPEU* (*Perceived Usefulness and Perceived Ease of Use*) untuk mengetahui sistem kemudahan pengoperasian sistem oleh pengguna dan kesesuaian sistem dengan kebutuhan yang sudah dideskripsikan.

### 3.7 Evaluasi Sistem oleh *Client*

Pada tahap ini, sistem yang sudah dibangun dievaluasi oleh para pemangku kepentingan, apakah sudah sesuai dengan keinginan pemangku kepentingan atau belum. Jika sistem belum sesuai dengan permintaan pemangku kepentingan maka akan dilakukan kembali proses pembangunan sistem sampai hasil sistem sesuai dan disetujui oleh para pemangku kepentingan. Setelah disetujui, maka sistem siap untuk digunakan.

### 3.8 Penutup

Setelah melakukan semua tahapan di atas maka pada akhir penelitian ini penulis akan menarik sebuah kesimpulan dan memberi saran. Kesimpulan harus dapat menjawab rumusan masalah yang sudah dijabarkan sebelumnya. Kesimpulan diambil berdasarkan hasil implementasi sistem yang telah dilakukan pengujian. Saran dibuat untuk para peneliti selanjutnya agar memperbaiki kekurangan yang terdapat pada penelitian ini sehingga penelitian ini dapat dikembangkan lagi.

## DAFTAR REFERENSI

- Agile Modeling, 2018. *UML 2 Sequence Diagrams: An Agile Introduction*. [online]. Tersedia di: <<http://www.agilemodeling.com/artifacts/sequenceDiagram.htm>> [Diakses 06 Februari 2020].
- Anthony, A., Tanaamah, A. R. & Wijaya, A. F., 2017. *Analisis dan Perancangan Sistem Informasi Penjualan Berdasarkan Stok Gudang Berbasis Client Server (Studi Kasus Toko Grosir "Restu Anda")*. [e-journal] 2(4), 136-147. Tersedia melalui: <<http://itiik.ub.ac.id/index.php/itiik/article/view/321/pdf>> [Diakses 29 Februari 2020].
- Aziz, M., Pranolo, A. & Agustin, C., 2015. *Developing Sales Information System Application using Prototyping Model*. [e-journal] 4(10), 782-785. Tersedia melalui: <<http://ijcatr.com/archives/volume4/issue10/ijcatr04101011.pdf>> [Diakses 29 Februari 2020].
- Basuki, A.P., 2014. *Proyek Membangun Website Berbasis PHP dengan Codeigniter*. Yogyakarta : Lokomedia.
- Daqiqil, I., 2011. *Framework Codeigniter – Sebuah Panduan dan Best Practice*. [e-book]. Tersedia di: Belajarphp.net <<https://belajarphp.net/ebook-tutorial-codeigniter-pdf/>> [Diakses 07 Februari 2020].
- Davis, F., 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, [e-journal] 13(3), 319-340. Tersedia melalui: <<https://www.jstor.org/stable/249008?seq=1>> [Diakses 2 Maret 2020].
- Google, 2020. *Google Analytics*. [online] Tersedia di: <<http://www.google.com/analytics>> [Diakses 2 Maret 2020].
- Heizer & Render. 2014. *Manajemen Operasi*. Jakarta: Salemba Empat.
- Hidayatullah, P. & Kawistara, J. K., 2017. *Pemrograman Web*. Bandung: Informatika.
- Kemenperin, 2019. *Selain Padat Karya, Industri Batik Punya Orientasi Ekspor*. [online] Tersedia di: <<https://kemenperin.go.id/artikel/21115/Selain-Padat-Karya,-Industri-Batik-Punya-Orientasi-Ekspor>> [Diakses 24 Februari 2020].

- McLeod, R. & Schell, G., 2007. *Management Information System*. 10<sup>th</sup> Edition. New Jersey: Prentice-Hall.
- Pitt, C., 2012. *Pro Php Mvc*. [e-book] Apress. Tersedia di: IT Ebooks <<http://it-ebooks.info/search/?q=Pro+PHP+MVC&type=title>> [Diakses 07 Februari 2020].
- Pressman, R.S., 2001. *Software Engineering A Practitioner's Approach*. 5th ed. [e-book] Mc Graw Hill. Tersedia melalui: Qazin Islamic Azad University <<http://www.giau.ac.ir/teacher/files/911610/13-11-1387-17-31-03.pdf>> [Diakses 05 Februari 2020].
- Pressman, R.S., 2010. *Software Engineering A Practitioner's Approach*. 7th ed. [e-book] McGraw-Hill. Tersedia melalui: Universitas Dian Nuswantoro <[http://dinus.ac.id/repository/docs/ajar/RPL-7th ed software engineering a practitioners approach by roger s. presman .pdf](http://dinus.ac.id/repository/docs/ajar/RPL-7th%20ed%20software%20engineering%20a%20practitioners%20approach%20by%20roger%20s.%20presman.pdf)> [Diakses 05 Februari 2020].
- Raharjo, B., Heryanto, I., & Enjang, R. K., 2012. *Modul Pemrograman Web (HTML, PHP, & MySQL)*. Bandung: Modula.
- Rozi, Z., A., dan SmitDev Community, 2015. *Bootstrap Design Framework*. Jakarta: PT. Elex Media Komputindo.
- Shalahuddin M., dan Rosa A. S., 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Snatkin, A., Eiskop, T., Karjust, K. & Majak, J., 2015. *Production Monitoring System Development and Modification*. [e-journal] 64(4s), 567-580. Tersedia melalui:< [http://eap.ee/public/proceedings\\_pdf/2015/issue\\_4S/Proc-2015-4S-567-580.pdf](http://eap.ee/public/proceedings_pdf/2015/issue_4S/Proc-2015-4S-567-580.pdf)> [Diakses 29 Februari 2020].
- Sommerville, I., 2011. *Software Engineering*. 9th ed. London: Addison-Wesley.
- Sparxsystems, 2018. *UML 2 Tutorial – Sequence Diagram*. [online] Tersedia di: <<https://sparxsystems.com/resources/tutorials/uml2/sequence-diagram.html>> [Diakses 06 Februari 2020].
- Sulistiono, H., 2018. *Coding Mudah dengan CodeIgniter, JQuery, Bootstrap, dan Database*. Jakarta: PT. Elex Media Komputindo.
- Sunyoto, A., 2007. *Ajax Membangun Web dengan Teknologi Asynchrone JavaScript & XML*. Yogyakarta: CV. Andi Offset.

## LAMPIRAN A WAWANCARA PERTAMA

### Data Narasumber

Nama : Yudi Satria

Jabatan : bagian produksi

Usia : 30 tahun

Tanggal wawancara : 02 Februari 2020

Waktu wawancara : 09.30 – 10.30

### Hasil Wawancara

1. Bagaimana pembagian penugasan yang ada di Batik Canteng Koneng?

Jawab: Sebenarnya di Batik Canteng Koneng struktur pembagian tugasnya sudah ada, hanya saja orang-orangnya masih tidak lengkap, masih ada penugasan yang digabung yang artinya ada 1 orang pegawai yang ditugaskan lebih dari 1 tugas. Hal ini menyebabkan jika ada pemesanan yang melonjak Batik Canteng Koneng membutuhkan orang tambahan. Sehingga manajemen di Batik Canteng Koneng masih belum tertata dengan rapi. Kami sebenarnya sudah memiliki alur manajemennya, tapi karena kekurangan tenaga ahli jadi masih ada beberapa posisi yang kosong. Tapi dari kami ingin alur manajemen yang sudah dibuat benar-benar diterapkan di Batik Canteng Koneng.

2. Apakah batik yang dijual di Batik Canteng Koneng harus pesan terlebih dahulu atau ada batik yang sudah siap untuk dijual?

Jawab: Batik Canteng Koneng menyediakan stok untuk batik yang sudah jadi, namun masih dalam bentuk kain. Namun stok yang sudah jadi belum bisa memproduksi banyak karena banyaknya pesanan batik yang masuk, sehingga untuk memproduksi bati yang sudah jadi waktu yang dibutuhkan tidak cukup. Karena untuk menambah karyawan diperlukan keahlian, tidak sembarang karyawan.

3. Apakah di Batik Canteng Koneng bisa *me-request* motif?

Jawab: Untuk penentuan motif bebas, bisa mengikuti contoh batik yang sudah di produksi oleh Batik Canteng Koneng atau biasanya dari *customer* meminta jenis motif yang simpel saja, tambahan logo, ataupun motif dengan melihat kisaran harga.

4. Berapa lama proses pengerjaan pembuatan batik?

Jawab: 1 batik bisa sampai 4-6 hari tergantung tingkat kerumitan desain batik. Dan harga batik juga tergantung pada tingkat kerumitan motifnya dan hasil akhir dari batik tersebut. Namun sudah ditetapkan di awal patokan harga batik tersebut. Penentuan harga di dapat dari tingkat kerumitan motif dan jenis kain, sedangkan bahan pewarna sama.

5. Bagaimana penentuan penggajian pegawai?

Jawab: untuk Batik Canteng Koneng sendiri buka dari pagi-sore. Namun karyawan bekerja berdasarkan waktu yang dijadwalkan untuk mereka bertugas. Oleh karena itu penggajian pegawai berdasarkan waktu kerja mereka perhari, standardnya sehari Rp. 80.000,00.

6. Apakah pernah terjadi kekurangan bahan baku ketika sedang melakukan produksi?

Jawab: Jarang. Biasanya pegawai ketika bahan baku habis langsung melaporkan ke bagian penyedia bahan baku dan kebetulan penjual bahan bakunya dekat dengan tempat produksi.

7. Untuk pelunasan pembayaran biasanya dilakukan kapan?

Jawab: biasanya disini awal pemesanan membayar uang DP terlebih dahulu, terutama untuk pemesanan seragam.

8. Bagaimana proses pemesanan batik jika customer berada di luar daerah Sumenep?

Jawab: Biasanya pemesanan dilakukan melalui aplikasi Whatsapp. Namun kekurangan dari metode pemesanan ini ketika *customer* meminta untuk mengirimkan contoh motif, pemilik harus mengirimkan banyak gambar kepada *customer* tersebut yang menyebabkan hal tersebut kurang efektif. Saya sebenarnya ingin ada *website* akun, dimana para *customer* bisa melihat contoh motif yang tersedia, dan yang bisa mengakses *website* tersebut hanya orang-orang yang memiliki akses, agar tidak terjadi penjiplakan terhadap motif Batik Canteng Koneng.

Sumenep, 02 Februari 2020

Yudi Satria

## LAMPIRAN B DOKUMEN PENDUKUNG HASIL WAWANCARA PERTAMA

- Sebagian nota penjualan

Rumah Batik Tulis  
Canteng Koneg  
Jl. Kuning 88 - 100 - 01 - 2020  
P. Adi Mulyono

No	Jenis Batik	Harga	Bpk. Batang	Jumlah
	Batik Ck. A12	400.000	1	400.000

Jumlah Rp 400.000  
Dp -  
Total -

Didik Haryanto | 0823 3334 5556 | email : canteng\_koneg@yahoo.com

Rumah Batik Tulis  
Canteng Koneg  
Jl. Kuning 88 - 100 - 01 - 2020  
P. Adi Mulyono

No	Jenis Batik	Harga	Bpk. Batang	Jumlah
	Korik	400.000	1	400.000

Jumlah 400.000,-  
Dp -  
Total 400.000,-

Didik Haryanto | 0823 3334 5556 | email : canteng\_koneg@yahoo.com

Rumah Batik Tulis  
Canteng Koneg  
Jl. Kuning 88 - 100 - 01 - 2020  
P. Mulyono

No	Jenis Batik	Harga	Bpk. Batang	Jumlah
	Nafa Kota	700.000,-	1	700.000,-
	New Collection	1.500.000,-	3	4.500.000,-
	Sen. Prowasail	3.000.000,-	1	3.000.000,-

Jumlah 8.200.000,-  
Dp -  
Total 8.200.000,-

Didik Haryanto | 0823 3334 5556 | email : canteng\_koneg@yahoo.com



➤ Alur proses barang keluar



➤ Alur pemesanan batik

