BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Penelitian dilakukan di Kantor Mitra Dinamika Konsultan, yang beralamat di Jalan Prof. Drg. Surya Sumantri 120 Kota Bandung, berikut penjelasan tentang objek penelitian.

3.1.1. Sejarah Singkat Perusahaan

Pada saat ini perkembangan teknologi dan sistem informasi pada tiap perusahaan menjadi hal yang penting dan menjadi keunggulan bersaing suatu perusahaan dengan perusahaan lainnya. Penggunaan teknologi dan sistem informasi yang efektif dan efisien akan mempengaruhi efektivitas dan efisiensi jaringan komunikasi antara setiap divisi dalam perusahaan. Jika sistem informasi perusahaan dapat menjangkau setiap divisi dalam perusahaan maka secara menyeluruh akan memudahkan setiap divisi untuk mencapai target maksimalnya. Jika sistem informasi yang ada tidak dapat menjawab kebutuhan setiap divisi, atau bahkan keakuratan data dan waktu tidak efektif, maka secara tidak langsung hal ini dapat berpengaruh pada keunggulan bersaing perusahaan, hal tersebut salah satunya dapat dikaitkan dengan kualitas software dan kualitas pelayanan SI yang diberikan perusahaan yang nantinya akan mempengaruhi kepuasan dari pelanggan. Jadi pelanggan merupakan kunci penting yang menjadi tolak ukur kualitas yang diberikan suatu perusahaan.

Oleh karena itu, berkembanglah banyak konsultan IT di Indonesia ini. Konsultan IT adalah seorang tenaga profesional yang menyediakan jasa kepenasihatan (*consultancy service*) dalam bidang informasi dan teknologi. Konsultan IT. Dimana Konsultan IT itu sendiri mempunyai fungsi dalam menangani konsultasi di biadang IT, meliputi saran bisnis, menyelesaikan masalah teknis maupun memperbaiki struktur dan efisiensi dalam sistem IT. Konsultan IT itu sendiri pada saat ini sudah berkembang cukup pesat di Indonesia seiring dengan kemajuan di bidang informasi dan teknologi, sehingga banyak perusahaan dan organisai yang beralih ke sistem komputerisasi.

Salah satu Konsultan IT yang sudah mulai berkembang adalah Mitra Dinamia Konsultan, suatu Konsultan IT yang didirikan pada bulan Oktober 2012 dimana Konsultan IT ini beralamatkan di Jalan Prof. Drg. Surya Sumantri 120 Kota Bandung. Mitra Dinamika Konsultan sudah menangani banyak proyek aplikasi *enterprise* dan proyek-proyek IT lainnya.

Terdiri dari karyawan-karyawan yang berpengalaman dan berdedikasi di bidangnya masing-masing, Mitra Dinamika Konsultan menargetkan lebih untuk aplikasi *enterprise* dalam mengembangkan perangkat lunak dengan pendekatan unik mereka dalam teknologi berbasis *web*.

Mitra Dinamika Konsultan menyediakan layanan-layanan berikut untuk memenuhi kebutuhan klien kami untuk teknologi informasi:

a) Kebutuhan Analisis dan Desain Sistem untuk kebutuhan perusahaan perangkat lunak solusi Anda.
b) Pengembangan Sistem Custom dan Outsource Build.

c) Implementasi Sistem dan Dukungan Produksi.

d) Business Intelligence Ekstensi untuk sistem saat ini.

e) Perusahaan Teknologi Informasi konsultasi dalam perencanaan, pengadaan, dan penyebaran untuk: Workstation, Server, LAN, WAN, Wireless, dll

Gambar 3.1. Logo Mitra Dinamika Konsultan

(Sumber : http://mitradinamikakonsultan.com, 2013)

3.1.2. Visi dan Misi Perusahaan

Adapun visi dan misi Mitra Dinamika Konsultan adalah sebagai beikut :

3.1.2.1. Visi Perusahaan

Visi Mitra Dinamika Konsultan adalah menjadi Konsultan IT yang terdepan dan selalu memberikan solusi yang terbaik untuk klien Mitra Dinamika Konsultan.

3.1.2.2. Misi Perusahaan

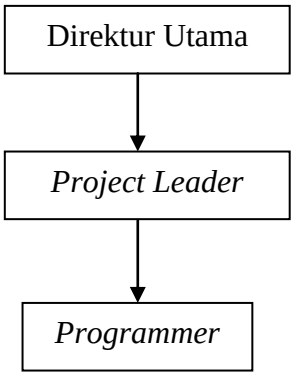
Misi Mitra Dinamika Konsultan adalah untuk memberikan solusi terbaik dalam memanfaatkan teknologi untuk para klien Mitra Dinamika Konsultan. Solusi Mitra Dinamika Konsultan adalah bukan hanya solusi teknologi yang seperti biasanya, tetapi harus memiliki manfaat nilai tambah dari segi produktivitas, optimalisasi biaya, dan skalabilitas untuk pertumbuhan bisnis klien Mitra Dinamika Konsultan, serta menawarkan pilihan *best-of-breed* teknologi

yang tersedia di pasar. Mitra Dinamika Konsultan dapat menjamin kepada klien kami bahwa solusi yang ditawarkan tidak akan sama dengan solusi Konsultan IT lainnya karena motto Mitra Dinamika Konsultan adalah: "*We will only do what we do could do best*".

3.1.3. Struktur Organisasi Perusahaan

Struktur organisasi kantor Mitra Dinamika Konsultan, terdiri dari :

1. Direktur utama
2. *Project Leader* : 1 orang
3. *Programmer* : 4 orang



Gambar 3.2. Struktur Organisasi Mitra Dinamika Konsultan

(Sumber : Mitra Dinamika Konsultan, 2013)

3.1.4. Deskripsi Tugas

1. Direktur Utama

Direktur utama Mitra Dinamika Konsultan mempunyai tugas pokok sebagai berikut :

a. Memimpin Mitra Dinamika Konsultan dengan membuat dan mengawasi kebijakan-kebijakan perusahaan.
b. Menyetujui anggaran perusahaan yang akan dikeluarkan.
c. Memilih, menetapkan dan mengawasi tugas dari karyawan-karyawan perusahaan.

2. *Project Leader*

Project Leader Mitra Dinamika Konsultan mempunyai tugas pokok sebagai berikut :

a. Memimpin perencanaan dan pelaksanaan proyek.
b. Melakukan koordinasi mengenai jadwal proyek secara keseluruhan.
c. Memastikan bahwa semua rencana proyek telah selesai.
d. Memenuhi setiap persyaratan kualitas dan waktu sesuai dengan perencanaan bisnis sebelumnya.
e. Melakukan perencanaan dan penjadwalan dalam rangka pencapaian target proyek dengan sukses.
f. Memimpin, memberikan arahan dan dorongan kepada para anggota tim kerja.
g. Menjaga hubungan baik dengan pelanggan.
h. Mengembangkan dan menyajikan laporan mengenai proyek kepada klien.
i. Melakukan pemantauan proyek secara terus-menerus.
j. Membuat laporan secara terperinci mengenai kemajuan proyek, jadwal, anggaran, risiko sampai solusi.
k. Melaporkan hasil kinerja mengenai pencapaian proyek.

3. *Programmer*

Programmer Mitra Dinamika Konsultan mempunyai tugas pokok sebagai berikut :

a. Membuat program seperti yang sudah diarahkan oleh *Project Leader.*
b. Membuat dokumentasi penerimaan tugas dari *Project Leader*.
c. Menguji program apakah alur kerja dan kebutuhannya sudah benar atau belum.
d. Membuat dokumentasi kepada *Project Leader* apabila programnya sudah terselesaikan.

3.2. Metode Penelitian

Metode penelitian merupakan cara penelitian yang digunakan untuk mendapatkan data agar mencapai tujuan tertentu. Metode penelitian diartikan sebagai cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu.

3.2.1. Desain Penelitian

Desain penelitian ini termasuk kedalam penelitian yang bersifat deskriptif. Penelitian deskriptif adalah penelitian yang bertujuan untuk memperoleh ciri-ciri variabel, dimana dalam penelitian ini adalah untuk memperoleh gambaran tentang kinerja program dirancang dan diimplementasikan kepada pengguna (*user*) dalam perusahaan.

Metode penelitian deskriptif (*descriptive research*) bertujuan mendeskripsikan suatu keadaan atau fenomena apa adanya. Dalam studi ini para

peneliti tidak melakukan manipulasi atau memberikan perlakuan tertentu terhadap objek penelitian. Semua kegiatan atau peristiwa berjalan seperti apa adanya. Penelitian deskriptif dapat berkenaan dengan kasus-kasus tertentu atau suatu populasi yang cukup luas. Kemudian, peneliti dapat menggunakan pendekatan kuantitatif, pendekatan kualitatif, atau penggambaran keadaan secara naratif-kualitatif. Penelitian deskriptif dapat dilakukan pada saat ini atau dalam kurun waktu yang singkat, tetapi dapat pula dilakukan dalam waktu yang cukup panjang. Penelitian yang berlangsung saat ini disebut penelitian deskriptif, sedangkan penelitian yang dilakukan dalam kurun waktu yang panjang disebut penelitian longitudinal.

3.2.2. Jenis dan Metode Pengumpulan Data

Jenis dan metode pengumpulan data digunakan penulis untuk mendapatkan data sebagai bahan kajian dalam penelitian ini dengan tujuan membuat suatu perancangan sistem informasi.

Metode pengumpulan data adalah teknik atau cara-cara yang dapat digunakan oleh peneliti untuk mengumpulkan data. Metode (cara atau teknik) menunjuk suatu kata yang abstrak dan tidak diwujudkan dalam benda, tetapi hanya dapat diperlihatkan penggunaannya melalui angket, wawancara, pengamatan ujian, dokumentasi, dan lainnya. Peneliti dapat menggunakan salah satu atau gabungan, tergantung pada masalah yang dihadapi

Dalam hal ini penulis menggunakan metode pengumpulan data berupa sumber data primer (observasi dan wawancara) dan sumber data sekunder (dokumentasi).

3.2.2.1. Data Primer

Sumber data atau informasi penelitian ini berdasarkan kepada jenis data yang diperlukan. Data primer yaitu data yang diperoleh dari responden secara langsung yang dikumpulkan melalui *survey* lapangan dengan menggunakan teknik pengumpulan data tertentu. Teknik pengumpulan data dalam rangka pengumpulan informasi mengenai objek penelitian ini, yaitu:

1. Metode Observasi

Peneliti langsung mengunjungi lokasi ke perusahaan yang dijadikan objek penelitian yaitu Kantor Mitra Dinamika Konsultan di Jalan Prof. Drg. Surya Sumantri 120 Kota Bandung. Hal ini dilakukan untuk melihat secara langsung masalah – masalah yang berhubungan dengan pokok bahasan penelitian.

2. Metode Wawancara

Wawancara merupakan suatu teknik pengumpulan data dengan cara bertatap muka antara pengumpul data dan narasumber yang dimaksud. Wawancara dilakukan kepada *Project Leader* dan bagian *Programmer* sehingga diperoleh sistem yang sedang berjalan di Mitra Dinamika Konsultan.

3.2.2.2. Data Sekunder

Jenis data sekunder yaitu data yang didapat dari dokumentasi – dokumentasi yang ada di perusahaan. Dilakukan dengan menelaah data sekunder yang digunakan untuk menunjang, melengkap idan menyempurnakan data primer, diperoleh dari dokumen perusahaan, internet, dan data – data dari perusahaan serta hasil penelitian yang ada hubungannya dengan bahasan ini.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

Metode adalah suatu cara / teknik yang sistematik untuk mengerjakan sesuatu. Metodologi pengembangan sistem yang ada biasanya dibuat atau diusulkan oleh : penulis buku, peneliti, konsultan, sistems house, pabrik software.

Metodologi adalah kesatuan metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang digunakan oleh suatu ilmu pengetahuan, seni atau disiplin lainnya.

3.2.3.1. Metode Pendekatan Sistem

Metode pendekatan sistem yang digunakan penulis adalah dengan menggunakan Pemodelan Berorientasi Objek yang ditandai dengan adanya diagram *Usecase Diagram*, *Activity Diagram, Sequential Diagram*, *Collaboration Diagram*, *Class Diagram*, *Component Diagram* dan *Deployment Diagram.*

3.2.3.2. Metode Pengembangan Sistem

Metode yang digunakan penulis dalam melakukan pengembangan sistem kepegawaian ini yaitu menggunakan model *prototype.* Model *prototype* adalah metode pengembangan sistem yang digunakan penulis dalam penelitian ini. Karena penulis memulai penelitian ini dengan mengumpulkan kebutuhan yang diperlukan dari pada sistem atau perangkat lunak yang akan dibuat.

Secara umum tahapan pada model *prototyping* dapat dilihat pada gambar berikut :



Gambar 3.3. Mekanisme Pengembangan Sistem dengan *Prototype*

(Sumber : Abdul Kadir.2003. P*engenalan Sistem Informasi.* Andi, Yogyakarta)

Prototyping merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode *prototyping* ini pengembang dan *user* dapat saling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang *user* hanya mendefinisikan secara umum apa yang dikehendakinya tanpa menyebutkan secara *detail output* apa saja yang dibutuhkan, pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang kurang

memperhatikan efesiensi algoritma, kemampuan sistem operasi dan *interface* yang menghubungkan manusia dan komputer.

Berikut ini beberapa tahapan-tahapan dalam *Prototyping* yang akan dipakai oleh penulis dalam merancang sebuah sistem, yaitu sebagai berikut:

1. Pengumpulan kebutuhan, *User* dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Membangun *prototyping*, Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada *user* (misalnya dengan membuat *input* dan format *output*)
3. Evaluasi *protoptyping*, Evaluasi ini dilakukan oleh *user* apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginann *user*.
4. Mengkodekan sistem, Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai .
5. Menguji sistem, Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box*, *Black Box*, *Basis Path*, pengujian arsitektur dan lain-lain.
6. Evaluasi sistem, *User* mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan.
7. Menggunakan sistem, Perangkat lunak yang telah diuji dan diterima *user* siap untuk digunakan.

Jenis-Jenis *Prototyping* :

1. *Feasibility prototyping*. Digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk sistem informasi yang akan disusun.

2. *Requirement prototyping*. Digunakan untuk mengetahui kebutuhan aktivitas bisnis *user*. Misalnya dalam sebuah perusahaan terdapat *user* direktur, manajer, dan karyawan. Maka penggunaan sistem dapat dibedakan berdasarkan *user* tersebut sesuai dengan kebutuhannya.

3. *Desain Prototyping***.** Digunakan untuk mendorong perancangan sistem informasi yang akan digunakan.

4. *Implementation prototyping*. Merupakan lanjutan dari rancangan protipe, prototype ini langsung disusun sebagai suatu sistem informasi yang akan digunakan.

Dalam sistem ini, penulis menggunakan jenis *requirement prototyping* untuk pengembangannya.

Keunggulan metode *Prototyping***:**

1. Adanya komunikasi baik antara pengembang dengan pelanggan.

2. Pengembang dapat bekerja lebih baik untuk memenuhi kebutuhan pelanggan.

3. Pelanggan berperan aktif dalam pengembangan sistem.

4. Menghemat waktu dalam pengembangannya.

5. Penerapan lebih mudah karena pemakai akan mengetahui apa yang diharapkan.

Kelemahan metode *Prototyping* **:**

1. Kualitas sistem kurang baik karena hanya mengedepankan aspek kenyamanan user.

2. Pengembang kadang-kadang menggunakan implementasi yang sembarangan.

3. Tidak mencerminkan proses perancangan yang baik.

3.2.3.3. Alat Bantu Analisis dan Perancangan

Alat bantu yang digunakan dalam perancangan sistem adalah sebagai berikut:

1. *Usecase Diagram*

Sebuah bagian penting dari *Unified Modelling Language* (UML) adalah fasilitas yang memungkinkan kita untuk menggambar sebuah diagram, yaitu diagram use-case. *Use case* digunakan selama tahap analisis dalam sebuah proyek untuk mengidentifikasikan dan membagi fungsi dari sistem. Diagram ini membagi-bagi sistem menjadi *actor* dan *use cases*. Aktor mewakili peran-peran yang dapat dikerjakan oleh user di dalam sistem. *User-user* ini dapat berupa manusia, bagian *hardware* komputer atau bahkan *software* sistem yang lain. Salah satu kriteria adalah mereka harus berada di luar bagian dari *use case* sebuah sistem. Mereka harus menyediakan masukan (inputan) ke dalam sistem dan mereka akan mendapatkan keluaran (*output*) dari sistem tersebut.

Use case menjelaskan tentang cara kerja dari sistem setelah mendapat inputan dari actor. Cara kerja ini dijelaskan secara rinci. *Use case* menjelaskan tentang hal-hal seperti, masukan dari seorang actor dan bagaimana keluaran yang diterima oleh *actor* lain, dan cara kerja yang dapat merubah suatu masukan menjadi keluaran.

Use case merupakan suatu diagram yang sangat berguna bagi pihak analis saat mengelompokkan fungsi-fungsi dari sistem. Hubungan-hubungan antara cases dan diagram-diagram yang berhubungan dapat membantu pihak analis untuk membuat struktur dari sebuah sistem. Tetapi *use cases* bukanlah sebuah alat desain, mereka tidak digunakan untuk memspesifikasikan sebuah software atau mereka tidak melibatkan sebuah keberadaan kelas-kelas dan obyek-obyek. *Use case* merupakan sebuah deskripsi tentang sebuah sistem yang secara murni terpisah dari tahap desain sebuah *software*.

2. *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas

dari level atas secara umum.Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

3. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/ metode dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message

4. *Collaboration Diagram*

Collaboration Diagram adalah suatu diagram yang memperlihatkan atau menampilkan pengorganisasian interaksi yang terdapat disekitar objek dan hubungannya terhadap yang lainnya. *Collaboration Diagram* lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan.

5. *Class Diagram*

Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. *Class diagram* memberikan gambaran sistem secara statis dan relasi antar mereka. Biasanya, dibua beberapa *class diagram* untuk sistem tunggal. Beberapa *diagram* akan menampilkan subset dari kelas-kelas dan relasinya. Dapat dibuat beberapa diagram sesuai dengan yang diinginkan untuk mendapatkan gambaran lengkap terhadap sistem yang dibangun.

6. *Component Diagram*

Component diagram adalah *diagram* UML yang menampilkan komponen dalam sistem dan hubungan antara mereka. Pada *component view*, akan difokuskan pada organisasi fisik sistem. Pertama, diputuskan bagaimana kelas-kelas akan diorganisasikan menjadi kode pustaka. Kemudian akan dilihat bagaimana perbedaan antara berkas eksekusi, berkas *dynamic link library* (DDL), dan berkas *runtime* lainnya dalam sistem.

7. *Deployment Diagram*

Deployment Diagram adalah *diagram* yang menggambarkan detail bagaimana komponen disebar kedalam *infrastruktur* sistem, dimana komponen akan terletak (pada mesin, *node*, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, *spesifikasi server*, dan hal-hal lain yang bersifat fisikal.

3.2.4. Pengujian *Software*

Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat

lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Metode yang digunakan dalam pengujian *software* ini adalah metode *Black Box Testing.*

Pengujian *black box* merupakan pendekatan komplementer dari teknik *white box*, karena pengujian *black box* diharapkan mampu mengungkap kelas kesalahan yang lebih luas dibandingkan teknik *white box*. Pengujian *black box* berfokus pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program.

Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan. Pengujian *black box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database eksternal.*
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Berbeda dengan pengujian *white box*, pengujian *black box* cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *black box* harus dapat menjawab pertanyaan sebagai berikut :

1. Bagaimana validitas fungsional diuji.

2. Kelas input apa yang akan membuat kasus pengujian menjadi lebih baik.

3. Apakah sistem akan sangat sensitive terhadap harga input tertentu.

4. Bagaimana batasan dari suatu data diisolasi.

5. Kecepatan data apa dan volume data apa yang akan ditoleransi oleh sistem.

6. Apa pengaruh kombinasi tertentu dari data terhadap sistem operasi.

Dilihat dari objek, fungsi, dan kegunaannya, *black box* testing sangat cocok digunakan untuk menguji apakah program / perangkat lunak sudah berfungsi dengan benar dan sesuai dengan keinginan pengguna. Maka dari itu, penulis menggunakan metode *black box* untuk menguji perangkat lunak untuk sistem informasi kepegawaian yang telah dibangun.