BAB II
TINJAUAN PUSTAKA

2.1 Unit Kegiatan Mahasiswa (UKM)

Unit Kegiatan Mahasiwa yang disingkat UKM adalah organisasi yang mempunyai tugas melakukan koordinasi ekstra-kurikuler mahasiswa di lingkungan kampus (Asri, Bawa & Suhendar (2016). UKM berfungsi sebagai wadah untuk penyaluran kreatifitas dan aktifitas para mahasiswa agar dapat :

a. Mengembangkan minat dan bakat.
b. Meningkatkan daya tahan dan wawasan nasional.
c. Membentuk watak dan kepribadian sesuai dengan tujuan lembaga.
d. Meningkatkan kualitas diri yang bersifat keahlian profesional dan edukatif.
e. Meningkatkan kesempatan mahasiswa untuk mengembangkan potensi diri.
f. Mengembangkan dan membantu proses pembelajarannya.

2.2 Pengertian Penarikan *(Recruitment)*

Rekrutmen merupakan salah satu kegiatan pengadaan pegawai (Rosidah, 2009:168) rekrutmen adalah proses mencari, menemukan dan menarik para pelamar untuk menjadi pegawai pada dan oleh organisasi tertentu. Rekrutmen *(recruitment)* juga dapat didefinisikan sebagai serangkaian aktivitas mencari dan memikat pelamar kerja dengan motivasi, kemampuan, keahlian, dan pengetahuan yang diperlukan guna menutupi kekurangan yang didefinisikan dalam perencanaan kepegawaian.

2.3 Maksud dan Tujuan *Recruitment*

Menurut Siagian dalam Rosidah (2009:169) diadakannya rekrutmen adalah untuk mendapatkan persediaan sebanyak mungkin calon-calon pelamar sehingga organisasi akan mempunyai kesempatan yang lebih besar untuk melakukan pilihan terhadap calon pegawai yang dianggap memenuhi standar kualifikasi organisasi. Program rekrutmen yang baik perlu melayani banyak tujuan.

Tujuan rekrutmen yang sering disebutkan adalah memikat sekumpulan besar pelamar kerja, namun kumpulan-kumpulan pelamar sedemikian besarnya sehingga sangat mahal biaya untuk memprosesnya.

2.4 PHP

PHP adalah bahasa (*scripting language*) yang dirancang secara khusus untuk penggunaan pada web. PHP adalah tool untuk pembuatan halaman web dinamis. Kaya akan *fiture* yang membuat perancangan web dan pemrograman lebih mudah, PHP digunakan pada 13 juta domain (menurut survai Netcraft pada www.php.net/usage.php). PHP kependekan dari *HyperText Preprocessor*. Pada awal pengembangannya oleh Rasmus Lerddorf, dia menyebutnya sebagai *tools Personal Home Page*. Seperti bahasa pemprograman lainnya PHP memperoses seluruh perintah yang berada dalam skrip PHP didalam *web server* dan menampilkan outputnya ke dalam web *browser client*. PHP adalah bahasa *scripting* yang menghasilkan output HTML ataupun output lain sesuai keinginan program (misalnya PDF) yang dijalankan pada server side (Hendrafi, 2017).

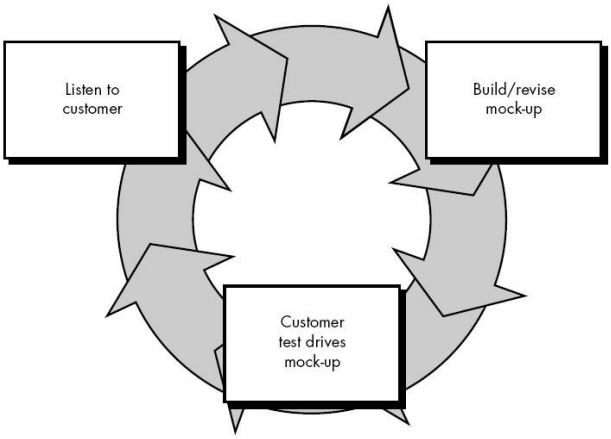
2.5 *Codeigniter*

Codeigniter adalah *framework* web untuk bahasah pemograman php, yang dibuat oleh Rick Ellis pada tahun 2006 yang mengiplementasikan pola desain atau arsitektur *Model*, *View*, *Controller* (MVC), yang memisahkan bagian kode untuk penganganan proses bisnis dengan kode untuk keperluan tampilan presentasi (tampilan). *Codeigniter* memiliki banyak fitur fasilitas yang membantu para pengembang php untuk dapat membantu aplikasi web secara mudah dan cepat (Raharjo, 2015).

2.6 Metode Pengembangan Sistem

Prototype merupakan metodologi pengembangan *software* yang menitik pada pendekatan aspek desain, fungsi dan *user-interface*. *Developer* dan *user* fokus pada *user-interface* dan bersama-sama mendefisikan spesifikasi, fungsi, desain dan bagaimana *software* bekerja. *Developer* dan *user* bertemu melakukan komunikasi dan menentukan tujuan umum, kebutuhan yang diketahui dan

gambaran bagian-bagian yang akan dibutuhkan. *Developer* mengumpulkan detail dari kebutuhan dan memberikan suatu gambaran dengan cetak biru (*prototype*). Dari proses tersebut akan diketahui detail-detail yang harus dikembangkan atau ditambahkan oleh *developer* terhadap cetak biru, atau menghapus detail-detail yang tidak diperlukan oleh *user*. Proses akan terjadi terus menerus sehingga produk sesuai dengan keinginan dari *user*.



Gambar 2.1 Metode Pengembangan Sistem *Prototype*
(Sumber : Nuryadi, 2014)

Pengembangan dari perancangan sistem ini dalam pelaksanaannya menggunakan tiga tahap siklus pengembangan model *prototype* yaitu:

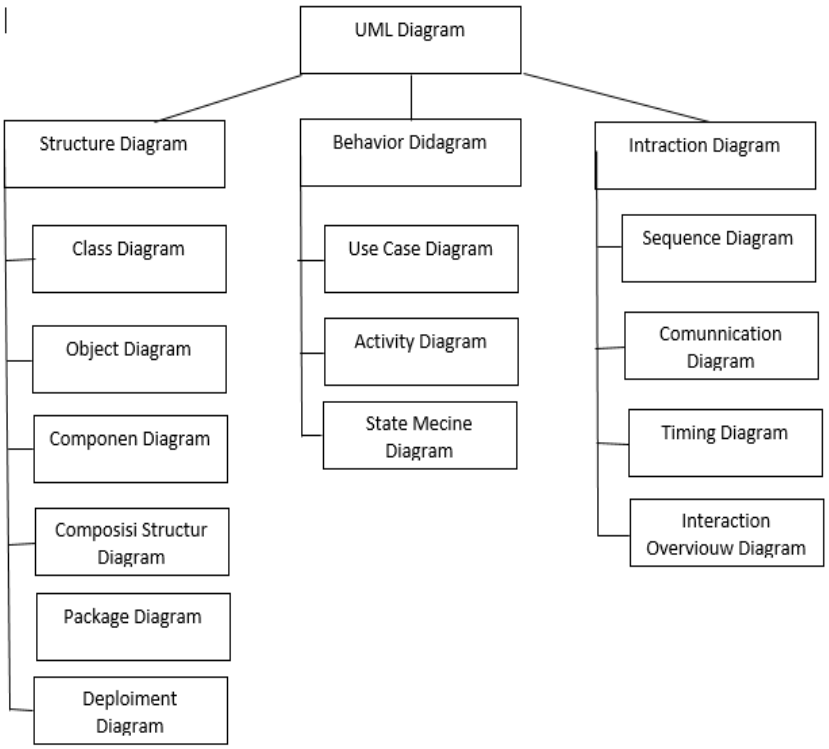
1. Mendengarkan Pelanggan *(Listen to Customer)* merupakan tahap pertama dalam merancang sebuah sistem. Pada tahap ini akan menentukan informasi- informasi yang dibutuhkan oleh pelanggan agar tercipta sebuah aplikasi sehingga mengarah pada tujuan dibuatnya aplikasi tersebut.
2. Membangun dan Memperbaiki prototipe *(Build/revice mockup)* dalam tahap ini dilakukan perancangan dan pengkodean untuk sistem yang diusulkan yang mana tahapannya meliputi perancangan proses-proses yang akan terjadi dalam sistem, perancangan diagram UML yang akan digunakan, perancangan antarmuka keluaran serta dilakukan tahap pengkodean terhadap rancangan- rancangan yang telah didefinisikan, kelengkapan *software* dan *hardware*.

3. Pengujian *prototype* pada tahapan ini dilakukan pengujian terhadap sistem yang telah disusun dan melakukan pengenalan terhadap sistem yang telah diujikan serta evaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan.

2.7 *UML* (*Unified Modeling Language*)

Di perlukan adanya bahasa yang di gunakan untuk memodelkan perangkat lunak yang akan di buat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pemodelan perangkat lunak digunakan untuk mempermudah langkah berikutnya dalam pengembangan sistem operasi pada perkembangan teknik pemrograman berorientasi objek. Muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan pemrograman berorientasi objek yaitu *Unified Modeling Language* (UML).

2.7.1 *Use Case Diagram*



Gambar 2.2 *UML Diagram*

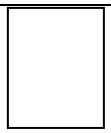
Sumber : Salahuddin (2013)

Use Case Diagram atau *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan di buat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan di buat. Ada dua hal utama pada *Use Case* yaitu pendefinisian apa yang desbut aktor dan *Use Case* :

- Aktor merupakan 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 gambar orang, tapi aktor belum tentu merupakan orang.
- *Use Case* merupakan fungsional yang di sediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.1 *Use Case Diagram*

Simbol	Fungsi	Keterangan
	Actor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat di luar sistem informasi yang akan di buat itu sendiri.
	Use Case	Fungsionalitas yang di sediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
	Asosiasi / *Association*	Komunikasi antara aktor dan *Use Case* yang berpartisipasi pada *Use Case* atau *Use Case* memiliki interaksi dengan aktor.
	Ekstensi / extend	Relasi *Use Case* tambahan ke sebuah *Use Case* dimana *Use Case* yang ditambahkan dapat berdiri sendiri walau tanpa *Use Case* tambahan itu.
	Generalisasi / *generalization*	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua sebuah *UseCase* dimana fungsi yang satu adlah yang lebih umum dari lainnya.

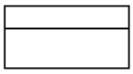
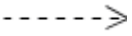
	Menggunakan / *Include/ uses*	Ada dua sudut pandang pada include pada *Use Case* pertama, *include* berarti *Use Case* yang ditambahkan akan selalu dipanggil saat *Use Case* tambahan di jalankan. kedua, interprestasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interprestasi yang dibutuhkan.
	*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

Sumber : Salahudin (2013)

2.7.2 *Class Diagram*

Diagram class atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. *Diagram class* dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Tabel 2.2 *Class Diagram*

Simbol	Fungsi	keterangan
	Kelas	Kelas pada struktur sistem
	Antarmuka / *interface*	Sama dengan konsep interface dalam pemograman berorientasi objek
	Asosiasi / *association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
	Asosiasi berdarah / *directed association*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
	Kebergantungan / *dependency*	relasi antar kelas dengan makna kebergantungan antar kelas
	Agregasi / *aggregation*	Relasi antar kelas dangan makna semua-bagian (whole-part)
	Generalisasi	Realsi antar kelas dengan makna generalisasi-spesialisasi (umum-khusu)

Sumber : Salahudin (2013)

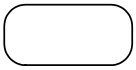
2.7.3 *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (Aliran Kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas juga bangak digunakan untuk mendefinisikan hal-hal berikut :

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan subah pengujian yang perlu didefinisikan kasus ujinya.
- Rancangan menu yang ditampilkan pada sebuah perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas

Tabel 2.3 Diagram Aktivitas

Simbol	Fungsi	Keterangan
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivias memiliki sebuah status awal
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan/ *Decision*	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

Sumber : Salahudin (2013)

2.8. Metode Pengujian *Black Box*

Menurut Rosa dan Shalahuddin (2016), *Black Box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah

fungsi–fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan sfesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkannya. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat
adalah :

1. Jika user memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika user memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.9 Penelitian Terkait

Pada penelitian (Rahmatulloh, Rachman, & Anwar, 2019) tentang Web Push *Notification* merupakan mekanisme pemberitahuan menggunakan *Javascript* pada web browser. Fitur ini tersedia dalam Push API HTML 5 dengan menggunakan *Push Service* atau *Messaging* server yang mengirim pemberitahuan ke web *browser* yang telah berlangganan tanpa membuka *website* sehingga dapat melakukan *broadcast message* dan *Notification* API HTML 5 tidak memerlukan *Push Service* atau *Messaging* server tetapi harus membuka *website*, tetapi belum didukung semua web *browser* sehingga pada makalah ini dibahas Implementasi Web *Push Notification* pada sistem informasi manajemen arsip menggunakan *Push JS*, metode pengembangan yang digunakan adalah *Rational Unified Proccess* (RUP).Teknologi pemberitahuan yang cocok untuk sistem informasi manajemen arsip berbasis web yaitu *Notification* API HTML 5 karena tidak akan mengirim pemberitahuan yang sama ke semua pengguna. Namun tidak ada proses di belakang layar sehingga tidak akan dijalankan secara otomatis, masalah tersebut diatasi dengan menggunakan *AJAX* dengan mengambil *JSON* kemudian dijalankan berulang-ulang pada web *browser* dan meminimalisir bentrokan antara script web *push notification* di multi tab window atau window web browser diatasi menggunakan *local Storage* dari *Web Storage* API HTML 5. Hasil uji

menunjukan bahwa penerapan teknologi Web *Push Notification* pada Sistem Informasi Manajemen Arsip dapat membantu para pengguna dalam mengelola arsip yang banyak serta penggunaan *AJAX* berpengaruh terhadap kecepatan akses web. Penelitian (Sauri, Haryono, Astuti, Khairina, & Cahyadi, 2015) tentang Universitas Mulawarman memiliki Unit Kegiatan Mahasiswa yang menaungi olahraga tersebut, bernama UKM sepak bola. Salah satu tujuan didirikannya Unit Kegiatan Mahasiswa (UKM) Sepak bola adalah untuk mengembangkan potensi dan menggali bakat khususnya mahasiswa Universitas Mulawarman dalam cabang olahraga khususnya sepak bola. Masalah lain yang dihadapi oleh para pengurus UKM sepak bola adalah bagaimana cara mengenalkan UKM sepak bola kepada mahasiswa yang baru masuk ataupun mahasiswa yang memiliki bakat sepak bola. Dampak yang paling terlihat dari permasalahan tersebut adalah kurangnya jumlah anggota UKM sepak bola di Universitas Mulawarman, sehingga dibutuhkan sebuah sistem informasi Unit Kegiatan Mahasiswa (UKM) sepakbola Universitas Mulawarman, untuk mempermudah mahasiswa mendapatkan informasi tentang UKM sepak bola yang ada di Universitas Mulawarman. Hasil dari penelitian ini adalah sebuah media informasi untuk mahasiswa yang ingin mengetahui dan dapat bergabung dalam kegiatan UKM sepak bola, yang dapat diakses melalui website. Layanan yang tersedia pada *website* dapat diakses oleh pengelola UKM diantaranya mendaftarkan *user*, *login user*, menginputkan informasi kegiatan, dan menginputkan modul yang dapat diakses dikunjungin oleh Mahasiswa untuk mendapatkan informasi.