**6.5. The Repository, Index and Working Areas.**

**Working Area:**

When starting the git repository, you can follow the procedures and type the commands to create a repository directory on the local machine. The following are some of the command you need to create a git repository in the directory.

*$mkdir testgit*

*$cd testgit*

*$git init*

The git repository is created in the testigit directory, but it has nothing in it yet. At this point there are many options to take depend on what you want to do, for example, you can branch, tag, or do many other things using the git commands. To fill in git repository in testgit directory we need to create a .git subdirectory using git init. We could go further and create files and directories in .git subdirectory. The following are just the default subdirectories in .git.

* HEAD
* Config
* Description
* Hooks
* Reference

These subdirectories may have more extended subdirectories, and files in local machine or working area. After work done on the local machine, we can move to the staging area.

**Index:**

Git Index is another important file on this process. It is used to stage the work done on the local machine before committing it to local repository. if there is specific change made on a file, it must be described in the commit message. You can use git command *git add -p* to specify a particular change on a file at staging area.

**Repository:**

We can move away from staging area to git repository by commit our files. The git checkout branch command executes branch check and move the HEAD reference to branch reference. The git add files command executes the function of shaping process and specified files committed from git index. The following sequence diagram summarized the transition in these three important steps.

Diagram

Description automatically generated

**6.6 The Object Database**

Git object database or object directory is a container that store all pointers to the local contents. The main four basic primitive objects in git are tree, blob, commit and tag. Each basic primitive object may have type, size, and content attributes. Every object in the git directory with no referral pointer can be collected by garbage collection, and the Git database is cleaned up.

* Tree: It is an object type that involved with the issue of storing the file name and allow the group of files to be stored together. The store contains in git resemble that of a UNIX filesystem although it is not a complicated structure. Many contents are store as tree or blob objects. A single tree may contain blob or subtree and resemble another tree structure.
* Blob: Blob stand for “binary large object”, and it is the first kind of object created by git when a file is committed. An example is README.md file which is created and commit before any file in the repository. A blob always stores the information of the file. The directory creates the hash of that file name and store it.
* Commit: unlike tag that point to commit, commit point to the tree represent the higher-level directory. You can create a new commit to represent advance Head and the current index to point to a new branch. Commit object can be created by the git and point it to its tree object.
* Tag: The tag object has some similarity to commit, except that it doesn’t points to tree but to commit. Tag objects contain tag name, object type, message, and tag pointer. The tag type contains the hash of the tag object. The two type of tags are annotated and lightweight. An annotated tag is a bit complicated then the light lightweight tag.

The SHA stand for “Secure Hash Algorithm”. It is used to store the information required to represent repo history and are referenced by 40-digit object name. the important of SHA is that git can determine the similarity of the two objects if they have the same SHA. Different SHAs mean unidentical objects. These features are used in many ways, for example tracking of merging tree. Like other hash tag if the object has been changed the SHA can identify that changes. Gits use this property to determine the change made to the file in repository.

Diagram

Description automatically generated