|  |  |
| --- | --- |
| Nama Modul | preOrderStatistik |
| Deskripsi | Modul untuk menampilkan Statistik pada console secara PreOrder |
| Jenis | Procedure |
| Parameter | root : NodeTree, jmlFile : integer |
| Initial State | Tidak ada tampilan apapun pada *console* |
| Final State | Tampil Statistik kata pada *cosole*  secara pre-order |
| Kamus Data | |
| i : integer  temp : NodeStatistik | |
| Algoritma | |
| Begin  if(root != NULL)  {  printf("||%15s||", root->kata);  temp=root->headStatistik;  while(i<=jmlFile){  printf("%9d||", temp->jumlah);  temp=temp->next;  i++;  }    printf("\n");  preOrderStatistik(root->kiri, jmlFile);  preOrderStatistik(root->kanan, jmlFile);  }  end. | |

|  |  |
| --- | --- |
| Nama Modul | preOrderFileStatistik |
| Deskripsi | Modul untuk menampilkan Statistik pada file secara PreOrder |
| Jenis | Procedure |
| Parameter | root : NodeTree, jmlFile : integer, outfile : FILE |
| Initial State | Tidak ada tampilan apapun pada file outfile |
| Final State | Tampil Statistik kata pada file secara pre-order |
| Kamus Data | |
| i : integer  temp : NodeStatistik | |
| Algoritma | |
| Begin  if(root != NULL)  {  printf("||%15s||", root->kata);  temp=root->headStatistik;  while(i<=jmlFile){  printf("%9d||", temp->jumlah);    temp=temp->next;  i++;  }    printf("\n");  preOrderStatistik(root->kiri, jmlFile);  preOrderStatistik(root->kanan, jmlFile);  }  end. | |

|  |  |
| --- | --- |
| Nama Modul | preOrderPersentase |
| Deskripsi | Modul untuk menghitung persentase plagiarisme antar dua file |
| Jenis | Procedure |
| Parameter | root : NodeTree, jmlFile : integer, persentase : NodePersentase |
| Initial State | Node persentase kosong |
| Final State | Node persentase berisikan data |
| Kamus Data | |
| i, min : integer  temp : NodeStatistik | |
| Algoritma | |
| Begin  i=1;  min=0;  NodeStatistik\* temp;    if(root != NULL)  {  temp=root->headStatistik;  while(i < persentase->fileA){  temp=temp->next;  ++i;  }    min=temp->jumlah;  while(i<=jmlFile){  if(i==persentase->fileA || i==persentase->fileB){  persentase->totalWord += temp->jumlah; // Jumlahkan Seluruh kata  if(temp->jumlah<min){  min=temp->jumlah;  }  }  temp=temp->next;  i++;  }  persentase->totalSameWord += min\*2;    preOrderPresentase(root->kiri, jmlFile, persentase);  preOrderPresentase(root->kanan, jmlFile, persentase);  }  end. | |